



AT&T

**UNIX[®] SYSTEM V/386
RELEASE 4**

*Product Overview and
Master Index*



UNIX Software Operation



AT&T

**UNIX[®] SYSTEM V/386
RELEASE 4**

*Product Overview and
Master Index*



UNIX Software Operation

**Copyright 1990, 1989, 1988, 1987, 1986, 1985, 1984 AT&T
All Rights Reserved
Printed in USA**

Published by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

No part of this publication may be reproduced or transmitted in any form or by any means—graphic, electronic, electrical, mechanical, or chemical, including photocopying, recording in any medium, taping, by any computer or information storage and retrieval systems, etc., without prior permissions in writing from AT&T.

ACKNOWLEDGEMENT

Portions of this book have been provided by Intel Corporation.

IMPORTANT NOTE TO USERS

While every effort has been made to ensure the accuracy of all information in this document, AT&T assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document, its updates, supplements, or special editions, whether such errors are omissions or statements resulting from negligence, accident, or any other cause. AT&T further assumes no liability arising out of the application or use of any product or system described herein; nor any liability for incidental or consequential damages arising from the use of this document. AT&T disclaims all warranties regarding the information contained herein, whether expressed, implied or statutory, *including implied warranties of merchantability or fitness for a particular purpose*. AT&T makes no representation that the interconnection of products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting or license to make, use or sell equipment constructed in accordance with this description.

AT&T reserves the right to make changes without further notice to any products herein to improve reliability, function, or design.

TRADEMARKS

MULTIBUS is a registered trademark of Intel Corporation.
OPEN LOOK is a trademark of AT&T.
PostScript is a registered trademark of Adobe Systems, Inc.
SunOS is a registered trademark of Sun Microsystems, Inc.
UNIX is a registered trademark of AT&T.
VAX is a trademark of Digital Equipment Corporation.
X11/NeWS is a registered trademark of Sun Microsystems, Inc.
XENIX is a registered trademark of Microsoft Corporation.
XWIN is a trademark of AT&T.

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-957515-4

P R E N T I C E H A L L

ORDERING INFORMATION

UNIX® SYSTEM V, RELEASE 4 DOCUMENTATION

To order single copies of UNIX® SYSTEM V, Release 4 documentation, please call (201) 767-5937.

ATTENTION DOCUMENTATION MANAGERS AND TRAINING DIRECTORS:

For bulk purchases in excess of 30 copies please write to:

Corporate Sales

Prentice Hall

Englewood Cliffs, N.J. 07632

Or call: (201) 592-2498

ATTENTION GOVERNMENT CUSTOMERS: For GSA and other pricing information please call (201) 767-5994.

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

AT&T UNIX® System V Release 4

General Use and System Administration

- *UNIX® System V/386 Release 4 PC-Interface Administrator's Guide
- *UNIX® System V/386 Release 4 Network User's and Administrator's Guide
- *UNIX® System V/386 Release 4 Product Overview and Master Index
- *UNIX® System V/386 Release 4 System Administrator's Reference Manual
- *UNIX® System V/386 Release 4 User's Reference Manual
- *UNIX® System V/386 Release 4 MULTIBUS® Reference Manual
- *UNIX® System V/386 Release 4 MULTIBUS® Installation and Configuration Guide
- *UNIX® System V/386 Release 4 Mouse Driver Administrator's Guide
- *UNIX® System V/386 Release 4 Transport Application Interface Guide
- UNIX® System V Release 4 User's Guide
- UNIX® System V Release 4 System Administrator's Guide

General Programmer's Series

- *UNIX® System V/386 Release 4 Programmer's Reference Manual
- *UNIX® System V/386 Release 4 Programmer's Guide: SCSI Driver Interface
- UNIX® System V Release 4 Programmer's Guide: ANSI C and Programming Support Tools
- UNIX® System V Release 4 Programmer's Guide: Character User Interface (FMLI and ETI)
- UNIX® System V Release 4 Programmer's Guide: Networking Interfaces
- UNIX® System V Release 4 Programmer's Guide: POSIX Conformance
- UNIX® System V Release 4 Programmer's Guide: Support Services and Application Packaging Tools

System Programmer's Series

- *UNIX® System V/386 Release 4 Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual
- *UNIX® System V/386 Release 4 Integrated Software Development Guide
- UNIX® System V Release 4 Programmer's Guide: STREAMS

Migration Series

- UNIX® System V Release 4 ANSI C Transition Guide
- UNIX® System V Release 4 BSD/XENIX® Compatibility Guide
- *UNIX® System V/386 Release 4 Migration Guide

Graphics Series

- UNIX® System V Release 4 OPEN LOOK™ Graphical User Interface Programmer's Reference Manual
- UNIX® System V Release 4 OPEN LOOK™ Graphical User Interface User's Guide
- UNIX® System V Release 4 Programmer's Guide: XWIN™ Graphical Windowing System Xlib—C Language Interface
- UNIX® System V Release 4 Programmer's Guide: OPEN LOOK™ Graphical User Interface
- UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System NeWS
- UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System Server Guide
- UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System tNt Technical Reference Manual
- UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System XVIEW™
- UNIX® System V Release 4 Programmer's Guide: XWIN™ Graphical Windowing System Addenda: Technical Papers
- UNIX® System V Release 4 Programmer's Guide: XWIN™ Graphical Windowing System The X Toolkit

*386 specific titles

Available from Prentice Hall



Contents

1	Product Overview	
	About This Manual	1-1
	Introduction to the UNIX Operating System	1-2
	Introduction to Release 4.0	1-8
	Product Description	1-26
	Related Products	1-44

2	Permuted Index	
	Using the Permuted Index	2-1
	Permuted Index	2-6

3	Master Subject Index	
	Using the Subject Index	3-1
	Master Subject Index	3-3

Index Reference Card

1. PRODUCT OVERVIEW

I. PRODUCT OVERVIEW

1 Product Overview

About This Manual 1-1

Introduction to the UNIX Operating System 1-2

A Brief History 1-2

The Structure of the UNIX System 1-3

- The Kernel 1-3

- The Shell 1-4

- The File System 1-4

Benefits of the UNIX System 1-5

- Power and Flexibility 1-5

- Portability 1-6

Recommended Reading 1-7

Introduction to Release 4 1-8

Standardization and Unification 1-8

- BSD Convergence 1-8

- XENIX System Compatibility 1-9

- POSIX Conformance 1-10

- Device-Kernel Interface/Device Driver Interface 1-10

- SVID Publication 1-10

Release 4 Features 1-11

- Operating System Enhancements 1-11

- File System Enhancements 1-15

- Networking 1-17

- System Administration and Maintenance 1-20

- Real-Time Processing 1-21

- Character-Based User Interface 1-22

- Graphical User Interface and Windowing System 1-23

- Internationalization 1-24

- C Language 1-24

■ Extended Terminal Interface	1-25
-------------------------------	------

Product Description	1-26
Utilities Packages	1-26
■ Archive XL Floppy Tape Utilities Package	1-27
■ Base System	1-28
■ BSD Compatibility Package	1-28
■ Cartridge Tape Utilities Package	1-29
■ Distributed File System Package	1-29
■ Editing Utilities Package	1-29
■ Form and Menu Language Interpreter Package	1-29
■ Framed Access Command Environment Package	1-30
■ Graphics End User System Package	1-30
■ Graphics Programmer's Package	1-30
■ Kernel Debugger Package	1-30
■ Line Printer Spooler Package	1-30
■ Mouse Support Package	1-31
■ Network File System Package	1-32
■ Network Support Utilities Package	1-32
■ OA&M Basic and Extended Package	1-32
■ PC586 Ethernet Package	1-32
■ PC-Interface Ethernet Service Package	1-33
■ PC-Interface Utilities & RS-232 Service Package	1-33
■ Remote File Sharing Utilities Package	1-33
■ Remote Procedure Call Package	1-33
■ Security Administration Package	1-34
■ Standard C Development Environment Package	1-34
■ Termcap Package	1-35
■ Terminfo Package	1-35
■ TCP/IP Package	1-35
■ Unlimited User Upgrade Package	1-35
■ Windowing Package	1-35
■ WD8003 Ethernet Driver	1-36
■ X11/NeWS Graphical Windowing System Package	1-36
■ X11/NeWS Toolkit Package	1-36
■ XENIX Compatibility Package	1-36
The Release 4 Document Set	1-36
■ Product Overview and Master Index	1-39
■ User's Guide	1-39

■ User's Reference Manual	1-39
■ System Administrator's Guide	1-39
■ System Administrator's Reference Manual	1-39
■ Network User's and Administrator's Guide	1-40
■ Mouse Driver Administrator's Guide	1-40
■ MULTIBUS Installation & Configuration Guide	1-40
■ PC-Interface Administrator's Guide	1-40
■ Programmer's Guide: ANSI C and Programming Support Tools	1-40
■ Programmer's Guide: System Services and Application Packaging Tools	1-40
■ Programmer's Guide: Character User Interface (FMLI and ETI)	1-41
■ Programmer's Guide: POSIX Conformance	1-41
■ Programmer's Guide: Networking Interfaces	1-41
■ Programmer's Reference Manual	1-41
■ Programmer's Guide: SCSI Driver Interface	1-41
■ Programmer's Guide: OPEN LOOK™ Graphical User Interface	1-41
■ OPEN LOOK™ Graphical User Interface Programmer's Reference Manual	1-42
■ OPEN LOOK™ Graphical User Interface User's Guide	1-42
■ Programmer's Guide: XWIN™ Graphical Windowing System	1-42
■ Programmer's Guide: X11/NeWS® Graphical Windowing System	1-42
■ Programmer's Guide: STREAMS	1-42
■ Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual	1-43
■ Integrated Software Development Guide	1-43
■ MULTIBUS Reference Manual	1-43
■ Transport Application Interface Guide	1-43
■ Migration Guide	1-43
■ ANSI C Transition Guide	1-44
■ Software Notes	1-44
■ BSD/XENIX® Compatibility Guide	1-44

Related Products

About This Manual

This volume is intended to be a starting point for exploring UNIX® System V and a central reference when using the UNIX System V Release 4 documentation. It includes a product overview, which introduces a new user to the UNIX system in general and specifically to Release 4.

The UNIX system is a vast, continually evolving system, and it is not possible to provide a new user with a complete description of the system in a few pages. The product overview in this volume is merely an introduction to the system, including a description of the software and documentation that make up the UNIX System V Release 4 product; you are directed to other documents for detailed descriptions of features, as well as user instructions. The overview does not assume prior knowledge of the UNIX system or of earlier releases of UNIX System V. If you are upgrading from an earlier release of UNIX System V, you should see the *Migration Guide* for a description of Release 4 that explains the differences between the new release and previous releases.

Introduction to the UNIX Operating System

UNIX System V is an interactive, multi-user, multi-tasking operating system. An *operating system* is a set of instructions that governs how the electronic components of your computer interpret and respond to commands and to coded instructions in your application software. Because UNIX System V is an *interactive* system, you enter commands from your keyboard, and the system responds by executing your command or by prompting you for additional input. Because it is a *multi-user* system, many people can use the system at the same time, working together, sharing information, and using common tools. Because the system is *multi-tasking*, it can perform several tasks for you at the same time, executing one task, or *process*, in the background while you move on to another task.

A Brief History

Originally developed at Bell Laboratories in the 1960s, the UNIX system was designed to serve the research community—specifically researchers working together on a common project, each with a vital need to share information and with a need to construct programs quickly and easily to perform specialized functions.

Prior to the 1980s, the UNIX system was used primarily in university and government research centers. Built around a set of small but powerful tools, the UNIX system soon proved to be well suited to the development of computer software and easily tailored to specific industry applications.

Because of AT&T's liberal licensing of the UNIX system, different versions of the system began to appear in the early 1970s. Universities as well as equipment vendors obtained licenses to use the system, then modified it to meet their special needs. The first commercial implementation of the system appeared in the mid-1970s, followed soon afterward by the introduction of Microsoft Corporation's UNIX system product—the XENIX® operating system.

In 1981, the University of California at Berkeley introduced an implementation, called Berkeley Software Distribution (BSD), that quickly became a standard on Digital Equipment Corporation's VAX™ line of computers. From BSD, many other versions of the UNIX system evolved.

By the early 1980s, the UNIX system was running in a wide range of environments on a wide variety of systems from a number of different equipment manufacturers—making it unique in the computing world.

In 1984, following a court-ordered divestiture that removed legal obstacles to its entry into the computer industry, AT&T began licensing the system commercially as UNIX System V.

The Structure of the UNIX System

UNIX System V is a powerful operating system built upon a few simple concepts. Typically, the UNIX system is described as consisting of three major parts. These are

- the UNIX operating system kernel
- the shell command interpreter and programming language
- the file system.

The following sections briefly describe these components of the UNIX system.

The Kernel

The UNIX system *kernel* is the component of the operating system that controls user processes (commands and programs) and manages system resources, such as the file system. It interacts directly with the hardware, passing information back and forth between the hardware and software programs. The kernel surrounds the hardware, insulating it and eliminating the need for a program to be aware of machine architecture. The fact that machine architecture is hidden makes it easier for software developers to write programs that run on different hardware.

The kernel manages the computer's processor so that processor time is allocated among multiple users and applications. Processor time is scheduled such that programs seem to execute simultaneously.

The Shell

The UNIX system *shell* is a command interpreter that recognizes and interprets commands entered by the user from a keyboard. While the shell is a simple interface for the novice user, it provides the experienced user with a great deal of power and flexibility.

The design of the shell makes it easy to manipulate files and the input and output of commands. Commands can take their input from either a terminal or a file, and the output of commands can be directed to a file or to a peripheral device other than the terminal. In addition, you can specify the output of one command as the input of another command—an action called “piping.” This capability allows users to build their own specialized functions easily.

Using the shell as a programming language, the user can write custom shell procedures to do simple or complex tasks, letting the shell handle details like opening and closing files. The shell allows users to enhance and build on UNIX system capabilities and to adapt the operating system to many user applications without using a compiler or link editor.

For more information about the shell as both a user interface and a programming language, see the UNIX System V Release 4 *User’s Guide*.

The File System

A *file system* is a collection of directories and files where users organize and store data.

A *file* is a location where you store data. Typically, a file is described as an electronic equivalent of a paper document. You give the file a name that identifies its contents, just as you would title a document.

A *directory* is where you organize related files. To continue the analogy, a directory is like a file folder, where you store similar kinds of documents. You give the directory a name that describes the general contents of all your files, just as you would label a file folder.

To complete the analogy, a file system (in simplest terms) is like a drawer of your file cabinet, which contains multiple file folders that contain multiple documents.

UNIX System V organizes files and directories hierarchically in a tree-like structure that makes it easy for users to locate and maintain their files. It also provides facilities for creating, accessing, copying, moving and processing files, directories, or sets of these in a simple, consistent way. Space for files is automatically allocated and deallocated when a file is created or deleted and as it grows and shrinks in size. This means you do not need to allocate space for a file before it is used.

The UNIX system supports multiple file systems and file systems of various types, such as the traditional UNIX System V file system, which stores data in blocks of 0.5K, 1K, or 2K bytes, and UFS, a file system type that stores data in blocks as large as 8K bytes. File systems are “mountable” and “unmountable,” which means that file systems can be stored on removable media and made available to a user or group only when needed. This gives users easy access to additional disk space.

The UNIX system treats all files identically because it hides the differences between file types. This is what makes it possible to use the output of one program as the input to another program.

Benefits of the UNIX System

What makes the UNIX system unique among operating systems is its power and flexibility—qualities that recall its origins in the research community. The design elements that make the UNIX system such a powerful yet flexible system also make it a highly portable one, which means that the UNIX system is compatible with a wide range of hardware and software products.

Power and Flexibility

The power and flexibility of the UNIX system are seen in its multi-user, multi-tasking capabilities. The multi-user capability allows you to work independently on the system while sharing resources and data easily with other users. The multi-tasking capability means that you can execute commands and run programs simultaneously, using several different methods. In addition to normal interactive mode, jobs can be submitted to run in the “background.” While one job is executing this way, you can use the terminal interactively on another job. If you need to run several sessions interactively on the same terminal, you can do so, using commands such as `sh1`.

The UNIX system's power and flexibility can be seen in features that encourage productivity (such as pattern-matching capabilities and the ability to execute a command on many files at once) and in its versatile command set. The system provides utilities for retrieving information, manipulating data, generating reports, and editing batch files. It also provides programs that identify a set of files based on their file system attributes.

The system's power and flexibility are also seen in a rich collection of programming tools—tools important not only to programmers but to *all* users who need to construct quick solutions to specialized problems. The shell, for example, is more than an interface. It's a mechanism that lets you build powerful functions out of existing, simpler functions by chaining commands together and using the output of one command as the input for another.

The flexibility of the UNIX system is well-known to software developers. UNIX System V is designed so that applications, as well as new devices and protocols, can be added easily, without modifying the source code. The availability of various commands, languages, and libraries of programs makes it easy to develop and maintain new application or system programs.

Portability

The UNIX system is a highly portable operating system, which means that it is easily modified to run on a wide range of hardware with a minimum of differences across implementations; once a user learns the UNIX system, he or she can use the system on different hardware without having to *relearn*. In addition, the structure of the kernel makes it possible to design application software with no knowledge of the underlying machine architecture, so that applications can run on a wide variety of hardware installations.

UNIX System V provides programmers with tools that support the development of portable applications, such as the `curses` library and the `terminfo` database. These tools allow user programs to produce output on many different types of terminals; applications can be terminal-independent but still use terminal-dependent specifications stored in a common database.

The portability of the UNIX system and the resulting compatibility between implementations on different manufacturers' equipment have contributed to the acceptance of the UNIX system as a *de facto* standard in the industry for portable operating systems. To preserve the UNIX system as a *de facto* standard, AT&T is committed to standardizing the system and unifying the various versions distributed by different equipment vendors into a single, common base

system. Toward that goal, AT&T publishes the *System V Interface Definition* (SVID), a set of documents that defines the minimum functionality a UNIX system product must deliver to be called UNIX System V. AT&T also provides the System V Verification Suite (SVVS) Release 4, a set of tests vendors can run against their UNIX system products to confirm that they conform to the SVID specifications.

In addition to defining its own standards, AT&T participates in the efforts of industry standards bodies, such as the IEEE. For more information about AT&T's standardization and unification efforts, see "Introduction to Release 4" later in this guide.

Recommended Reading

It is impossible to give more than a cursory overview of UNIX System V in these few pages. Throughout this overview, you will be referred to guides and manuals in the UNIX System V documentation set for more information on specific topics. A description of the entire set of documents available with UNIX System V appears in "Product Description" later in this overview.

In addition to the UNIX System V documentation set, the following mass market books provide the beginner with an introduction to the UNIX system:

- *The UNIX for Beginners Book: A Step-by-Step Introduction*, by Bryan Strong and Jay Hosler. New York: John Wiley & Sons, 1987.
- *Introducing UNIX System V*, by Rachel Morgan and Henry McGilton. New York: McGraw-Hill, 1987.

Introduction to Release 4

UNIX System V Release 4 is a major new release of the UNIX operating system, providing a wealth of new features and enhancements to the system. These features and enhancements are designed and implemented as part of AT&T's continuing commitment to the following goals:

- Standardization of the UNIX system and the unification of UNIX system derivatives.
- Introduction of new technology—specifically, new technology that extends networking capabilities, improves system administration and maintenance, and further internationalizes the UNIX system.

The following sections describe the features of Release 4 in terms of these goals. Throughout the discussions, you are referred to other documentation for feature information that goes beyond the scope of this product overview.

Standardization and Unification

UNIX System V is an *open* operating system. As an open system, manufacturers of computing equipment can license rights to use it and customize it to meet the specific needs of their markets and product lines. As a result of the open licensing, many versions of the system have diverged to the point where customers may be confused as to what the UNIX system is today. Release 4 represents a major step in AT&T's efforts to standardize the UNIX system and to unify the various UNIX system derivatives into a single, common base.

The goal of Release 4 is to merge the best of the UNIX systems based on Berkeley Software Distribution (BSD), such as SunOS®, with the best of UNIX System V, provide XENIX system compatibility, and conform to the standards set by the IEEE POSIX committee. To encourage continued progress toward a UNIX system industry standard, Release 4 has been released in conjunction with the System V Interface Definition, Third Edition (SVID3).

BSD Convergence

In an effort to converge on one standard UNIX system, AT&T is working to resolve differences between BSD-based versions of the UNIX operating system and UNIX System V. Toward that end, Release 4 provides:

- a library that implements the BSD sockets interface
- 4.3 BSD signal mechanisms
- BSD commands, integrated into the UNIX System V command set
- BSD Job Control
- the TCP/IP network protocols (the DARPA Internet Protocol Suite)
- Remote Procedure Call (RPC) and External Data Representation (XDR) libraries
- The Network File System file-sharing utility (Sun Microsystem's file distribution product, based on BSD)
- Virtual File System (VFS) Architecture—an architecture that merges AT&T's File System Switch and Sun Microsystem's Vnodes and provides a mechanism that allows multiple file system types to coexist within UNIX System V; VFS also incorporates BSD functionality and symbolic links
- UFS, an alternative file system type based on 4.2 Berkeley Fast File system
- Multiple groups and ownership changes.

For more information about BSD features, see "Release 4 Features" later in this guide.

XENIX System Compatibility

UNIX System V provides full source code compatibility with Microsoft's XENIX system. XENIX system features first supported in AT&T's implementation of UNIX System V Release 3.2 for the 386 processor are supported in the source code in Release 4, along with XENIX system file and record locking (via the `locking` function), shared memory, and semaphores. Existing XENIX system source code that uses these facilities compiles and runs in Release 4. (These facilities are provided strictly for compatibility with existing XENIX system source code. New applications should use SVID interfaces.)

For more information about XENIX system compatibility, see the *BSD/XENIX® Compatibility Guide*.

POSIX Conformance

With Release 4, AT&T continues its commitment to the Institute of Electrical and Electronics Engineers (IEEE) standards efforts. The IEEE P1003 standards committee recently published a standard for the UNIX operating system, known as POSIX (Portable Operating System Interface for Computer Environments). While the POSIX specification is close to AT&T's own standard definition as documented in SVID3, Release 4 provides the few changes and additional features required to bring UNIX System V into full conformance with the POSIX standard.

For more information on POSIX conformance, see the *Programmer's Guide: POSIX Conformance*.

Device-Kernel Interface/Device Driver Interface

A goal of Release 4 is to support popular interfaces to the UNIX system and to provide new interfaces that are consistent with industry standards. In Release 4, UNIX System V supports the Device-Kernel Interface (DKI), an interface between the UNIX System V kernel and device driver software. DKI makes it easier to port driver code across implementations of Release 4 for different hardware; if the driver writer conforms to the DKI where possible, the portability of the driver to other UNIX System V Release 4 implementations is greatly enhanced.

Release 4 also supports the Device Driver Interface (DDI), a superset of the DKI that is specific to the implementation of UNIX System V for computers using Intel 386-based processors with AT or MULTIBUS II bus architectures. The DDI enhances driver binary compatibility across releases of UNIX System V for the 386-based computers.

See the *Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual* for more information.

SVID Publication

The System V Interface Definition (SVID) is AT&T's published definition of interfaces to UNIX System V. It defines the operating system program interfaces that are consistent across all UNIX System V implementations, and specifies an operating system environment that allows users to create application software that is independent of any particular computer hardware.

The availability of Release 4 has been coordinated with the publication of the System V Interface Definition, Third Edition (SVID3), which documents the functionality of Release 4 and presents possible future directions for UNIX System V.

Release 4 Features

Release 4 introduces a wealth of new features to UNIX System V, some of which were introduced in BSD UNIX systems prior to the release of UNIX System V Release 4. These features are now supported in UNIX System V as part of AT&T's effort to unify the UNIX system derivatives into one standard product.

The new features introduced in Release 4, and the BSD features that have been incorporated in UNIX System V, are described in the following sections:

- Operating System Enhancements
- File System Enhancements
- Networking
- System Administration and Maintenance
- Real-Time Support
- Character User Interface
- Graphical User Interface
- Internationalization
- C Language
- Extended Terminal Interface

Operating System Enhancements

As part of the effort to unify the various UNIX systems into one standard, Release 4 introduces some fundamental changes to UNIX System V. These include the Virtual Memory (VM) memory management architecture, a system access mechanism called Service Access Facility, STREAMS-based terminal subsystem and pipes, BSD job control, Expanded Fundamental Types, and booting changes and enhancements.

Memory Management

Release 4 incorporates a third-generation memory management architecture based on the VM (Virtual Memory) architecture introduced in SunOS. VM replaces both the original UNIX system swapping architecture and the REGIONS demand-paged virtual memory architecture introduced in UNIX System V Release 2.

The benefits of a demand-paged virtual memory implementation are efficient use of the system's main memory and the capability to execute programs much larger than the physical memory provided by the system. The Release 4 VM architecture provides these same benefits, plus additional benefits of interest to particular groups of users (as indicated below in parentheses):

- Mapped files (application programmers)

A by-product of the VM architecture is a style of file I/O for user programs called "mapped files." This set of capabilities, provided by the `mmap` family of routines, allows a file to be mapped explicitly into the address space of a user program, where the file can be manipulated as if it were an array in primary memory. These capabilities make user programs easier to write, and they allow programs to execute more efficiently. For more information, see the *Programmer's Guide: System Services and Application Packaging Tools*.

- Shared memory (application programmers)

The mapped files provided by VM can be considered a form of shared memory. If several processes map a file simultaneously, the system maps the same copy of a file into the memory space of all processes.

Release 4 also supports the traditional UNIX System V shared memory facility (provided by the `shmat` family of routines) and XENIX system shared memory semantics.

- Flexible use of disk space for swapping (administrators)

UNIX System V Release 4 swaps pages of data from main memory to an ordinary file on one of the system's disks. Swapping to a file accommodates the needs of diskless systems and allows for the more efficient use of disk space.

- Portable implementation (porters)

The VM architecture isolates all hardware-dependent portions of the memory management subsystem in one block of C language source code. The remainder of the code is portable across different hardware and system architectures. Most of the hardware-dependent code comprises a well-defined Memory Management Unit (MMU) interface. This interface allows the architecture to be implemented on top of different MMU hardware. (This means that the Release 4 memory management subsystem is simple to port to other hardware, because the hardware-dependent code is isolated in one small part of the system and has a simple interface.)

Release 4 provides a “multi-level store” implementation of VM that allows a system to address extremely large ranges of virtual address space using only a 32-bit hardware address space—the industry standard for central processors and hardware memory management units. (This means that Release 4 is simple to port to standard processor architectures.)

For more information about VM, see the *Programmer's Guide: System Services and Application Packaging Tools*.

Service Access Facility

Service Access Facility (SAF) is a feature in Release 4 that standardizes the way UNIX System V handles connection requests from outside the system. The SAF acts as an umbrella over the system external access points and provides a consistent access mechanism. Services are administered through a consistent framework of commands and files, whether they are requested from the console, from a terminal via a tty line, or across a network.

For more information, see the *System Administrator's Guide* and the *Programmer's Guide: Networking Interfaces*.

STREAMS

STREAMS is a mechanism that provides a framework for character I/O in the UNIX system. STREAMS allows programmers to structure kernel software in a modular manner and define standard interfaces within the kernel.

STREAMS was introduced in an earlier release of UNIX System V as a mechanism for implementing network protocols in a way that allows application programmers to develop network applications that are protocol-independent. In

Release 4, the STREAMS mechanism is more pervasive. The terminal subsystem has been reimplemented using STREAMS, providing the following advantages:

- A decrease in complexity of the kernel
- Modularity and flexibility of code
- Seamless integration of user terminals and networking capabilities
- Reusable line discipline modules
- Customization
- Ease in adding new features to the terminal subsystem.

For consistency, other mechanisms such as “pipes” have also been implemented using STREAMS so that, in Release 4, the entire character I/O subsystem is STREAMS-based.

For more information about STREAMS, see the *Programmer's Guide: STREAMS*.

Job Control

Job control is a popular feature of the BSD operating system and an optional part of the IEEE P1003.1 POSIX standard. Job control allows a user to stop and later resume a job, whether it is executing in the foreground or the background. Job control also allows a user to move jobs back and forth between the background and the foreground.

With job control a user can

- stop a foreground job in order to perform a more pressing task
- put a foreground job in the background
- stop a job to satisfy a need of the job, such as looking up data for input or changing the name of an input file to match what was misspelled on the command line.

Job control capabilities are available through an optional shell called the job control shell (jsh). For more information about job control, see the *User's Guide*.

Expanded Fundamental Types

Release 4 supports the expansion in size of certain data types, such as user ID (uid), process ID (pid), and device ID. This feature, known as Expanded

Fundamental Types (EFT), makes it possible to remove constraints on certain fundamental system types that were imposed by the UNIX operating system's original hardware implementation.

File-System-Independent Booting and Autoconfiguration

File-system-independent booting allows an administrator to boot the system from any device that is readable by firmware.

In Release 4, it is possible to make the root file system either a standard UNIX System V file system or a UFS file system. Rather than code knowledge of the possible file system types into the boot program, a disk partition is created when Release 4 is installed that holds a file system of type `bfs` (Boot File System). This file system contains all the bootable programs necessary for the computer's boot process.

In Release 4, the autoconfiguration process—the process of detecting hardware and software changes on powerup or reboot and automatically generating a new bootable operating system—operates in the same file-system-independent manner as the boot program.

File System Enhancements

Release 4 introduces a number of significant changes related to file systems. Most important, it introduces Virtual File System, an architecture that makes it possible for file systems of various types to coexist on the same system; the traditional UNIX System V file system type is now just one of many supported file system types. Also, the layout of the directory tree has been reorganized in Release 4 to facilitate sharing files, directories, and file systems in a networked environment.

Virtual File System

Release 4 introduces to UNIX System V a file system switch architecture called Virtual File System (VFS). VFS provides a clearly defined, modular interface between file systems and the rest of the UNIX system kernel, and allows several different file system types to coexist on the same system—including file systems that have significantly different characteristics and internal formats. The modular nature of the architecture allows programmers to design and install new file system types in a clean, straight-forward manner.

New File System Types

By reimplementing the UNIX System V file system architecture, Release 4 can support a wide variety of file system types on the same system. In addition to providing a mechanism that allows new types to be defined and installed, Release 4 provides the following file system types as standard options.

- s5** the traditional file system type supported in earlier releases of UNIX System V. The s5 type stores data in disk blocks of 0.5K, 1K, or 2K bytes.
- ufs** a file system type based on the BSD system “fast file system”; it supports data storage in disk blocks as large as 8K bytes.
- rfs** a distributed file system type—one of two supported in Release 4; **rfs** supports Remote File Sharing capabilities (including the ability to share files across a network of UNIX System V systems).
- nfs** a distributed file system type originally introduced in the SunOS operating system (a BSD system derivative); **nfs** supports Network File System capabilities (including the ability to share files across a network in a heterogeneous environment).
- proc** a file system type useful for debuggers and similar utilities; it provides a mechanism for accessing the address space of running processes.
- fifofs** a file system type that provides common access to pipe files.
- specfs** a file system type that provides a common code interface to all device files.
- bfs** a file system type that provides support for file system independent booting; a **bfs** file system contains all the programs necessary for the boot process.

For more information about file system types in general, see the *System Administrator's Guide*. For more information about RFS and NFS, see the *Network User's and Administrator's Guide*.

Symbolic Links

Release 4 introduces support for symbolic links. A symbolic link is a file that contains the pathname of another file. References to the symbolic link are converted by the UNIX system kernel into references to the target file.

Symbolic links make it possible to link directories. This means that the logical structure of a system's file tree can be rearranged without changing the physical location of files. Symbolic links also allow links to a physical file to reside in a different physical file system. This means that names can be linked across file systems that reside on different computers on a network—allowing a computer to create a logical directory tree that includes directories and files that physically reside on many different computers.

For more information about symbolic links, see the *Programmer's Guide: System Services and Application Packaging Tools*.

The Directory Tree

In Release 4, the UNIX System V directory tree has been arranged to facilitate sharing files, directories, and file systems in a networked environment. Release 4 divides the system into four standard file systems: / (called root), /usr, /home, and /var. Within the standard file systems, files are organized by type, with each type occupying a separate branch of the tree. The directory tree defines three file types: machine private files, architecture dependent files, and architecture independent files. Because the directory tree separates private files from shareable files, a system administrator can easily mount a complete file system for distribution over a network.

For more information, see the *System Administrator's Guide*.

Networking

Earlier releases of UNIX System V provided tools for developing network protocols and applications, specifically STREAMS and the Transport Level Interface (TLI). STREAMS defines standard interfaces that allow networking architectures and higher-level protocols to be independent of underlying protocols, drivers, and media. TLI defines an interface between applications and transport level network protocols, relieving user programs of the need to know the characteristics of the transport protocol.

Early networking capabilities in the UNIX system were available through two optional utilities packages: Basic Networking Utilities, which provided basic file transfer capabilities between UNIX System V systems via the uucp family of commands, and Remote File Sharing, a service that makes it possible for a system to share files and devices with remote UNIX System V systems.



These tools and services continue to be important features of UNIX System V in Release 4. For information about STREAMS, see the *Programmer's Guide: STREAMS*. For information about TLI, see *Programmer's Guide: Networking Interfaces*. For information about uucp services, see the *User's Guide* and the *System Administrator's Guide*. For information about RFS, see the *Network User's and Administrator's Guide*.

New in Release 4 is support for TCP/IP and RPC, networking protocols that have become standards in the BSD networking environment, and Network File System, a SunOS file distribution facility. Convergence with BSD networking is also seen in UNIX System V through its support of *inet.d*, a BSD network daemon, and the sockets interface. In addition, Release 4 introduces new UNIX system networking technology in the form of Network Selection and Name-to-Address Mapping.

TCP/IP (the DARPA Internet Protocol Suite)

In Release 4, UNIX System V supports TCP/IP, a family of network protocols sometimes called the DARPA Internet Protocol Suite. TCP/IP is a comprehensive set of protocols and commands that support networking among systems of different types.

For more information about TCP/IP, see the *Network User's and Administrator's Guide*.

Remote Procedure Call/External Data Representation

Release 4 includes Remote Procedure Call (RPC), a protocol that makes it possible for a process to call a procedure on a remote machine of a different architecture, using the syntax and semantics of a local procedure call. RPC allows a server program to use procedures on remote machines like building blocks in complex applications.

RPC uses External Data Representation (XDR) to encode data passed between machines. XDR is used to represent data in a standard way, so that a local process can call a procedure residing on a machine of different architecture that is running the same or a different operating system.

See the *Programmer's Guide: Networking Interfaces* for more information about RPC and XDR.

Inetd

In Release 4, UNIX System V supports `inetd`, a port monitor (also called a daemon) that waits for service requests from computers on a TCP/IP network. `inetd` is similar to `listen`, a UNIX System V port monitor that listens for service requests from clients on any network that conforms to the Transport Provider Interface. When a connection request arrives over the network, `inetd` (like `listen`) spawns the server process and passes the network connection to it.

For more information about `inetd`, see the *Network User's and Administrator's Guide*.

Sockets

Sockets is a networking interface widely used in BSD systems. It is functionally similar to the UNIX System V Transport Level Interface (TLI) in that it defines how a process accesses the services of a transport protocol.

Sockets is provided in Release 4 in a library so that BSD applications that use sockets can migrate easily to UNIX System V.

For more information, see the *Programmer's Guide: Networking Interfaces*.

Network File System

Network File System (NFS) is a facility for sharing files in a heterogeneous environment of machines, operating systems, and networks. Sharing is accomplished by mounting a remote file system, then reading or writing files in place. Users are able to get directly to the files they want without knowing the network address of the data. To the user, all NFS-mounted file systems look like private disks. There are no apparent differences between reading or writing a file on a local disk, and reading or writing a file on a disk in another location.

See the *Network User's and Administrator's Guide* for more information.

Network Selection

Release 4 introduces a feature to UNIX System V called Network Selection. Network Selection allows network applications to choose the network over which they will communicate. An application can select a network from a directory of networks that are available to the system. If an application fails to connect with its first choice, it can select different networks until it finds one that meets its service requirements and allows the connection.

Network Selection frees an application from needing a network selection embedded in its code. This allows the application to run without modification on different systems that support different networks.

For more information about Network Selection, see the *System Administrator's Guide* and the *Programmer's Guide: Networking Interfaces*.

Name-to-Address Mapping

Name-to-Address Mapping is a mechanism that allows network clients to determine the addresses of servers on a network in a network independent manner. The mechanism makes it possible for clients to reach a server, even if the address on which the server is listening should change. Name-to-Address Mapping also allows a client to reach a server over different networks.

See the *Programmer's Guide: Networking Interfaces* for more information.

System Administration and Maintenance

One of the goals of Release 4 is to simplify system administration and maintenance, while at the same time adding functionality to meet the demands of increasingly complex computer and network configurations. Toward that goal, Release 4 provides new capabilities in backup and restore operations, and software installation. It also introduces an enhanced administration interface that makes it easier for a novice administrator to set up and maintain a system.

The system administration features new in Release 4 are described in the following sections. For detailed information about these features and for information about system administration in general, see the *System Administrator's Guide*.

Improved Backup and Restore Operations

Release 4 provides hardware independent backup and restore procedures (supporting multiple bus architectures and multiple destination types, such as tapes, floppy diskettes, and hard disks). Multiple commands used as part of the backup procedure have been integrated in Release 4 into a single backup service.

Key features of the backup and restore service include

- a backup history log

- on-line backups
- automated backup initiation
- mechanized restore requests.

Software Installation and Configuration Management Tools

A software installation facility in Release 4 provides a system administrator with installation procedures that are consistent across software packages, releases, and machines. It also provides tools and guidelines for developing add-on packages that take advantage of the standard software installation script and menu interface. Configuration tools allow an administrator easy access to information on system configuration.

System Administration Menus (*sysadm*)

Release 4 offers a new version of the menu interface for administration of a UNIX system. The System Administration Menus (accessed using the *sysadm* command) allow a user to administer a UNIX system without knowledge of UNIX system commands. Context-sensitive help messages make it possible for an administrator to explore the interface with a minimum of documentation.

Real-Time Processing

Historically, the UNIX system has been a general purpose time-sharing system. Today, however, the system is seen more and more in environments that require real-time processing.

To support real-time processing, Release 4 introduces a new process-scheduler architecture and high-resolution timing services. (For more information than that provided in the following descriptions, see the *System Administrator's Guide* and the *Programmer's Guide: System Services and Application Packaging Tools*).

User-Controlled Process Scheduler

A process scheduler is kernel code that determines what program will run, when, and for how long. The Release 4 architecture supports both the traditional scheduling policy and the new real-time scheduling policy. Each process has its own scheduler properties, which can be changed by users while the process is running. The scheduler runs each process for a configurable "time slice," then gives other processes a chance to run.

The time-sharing scheduler policy manages processes in a traditional manner: it dynamically adjusts time-sharing process priorities in an attempt to give good response to all interactive processes.

The real-time scheduler policy, on the other hand, never changes a real-time process priority except as the result of an explicit user request to change it. Moreover, all real-time processes run before any other processes. An application can perform its time-critical tasks in real-time processes and be assured that these processes will always get priority over all non-real-time processes.

Also to support real-time processing, Release 4 provides new pre-emption points in the kernel—points at which the scheduler may switch control of the CPU from one process to another. The additional pre-emption points improve system response time for high-priority processes.

Using real-time processes, an application can guarantee fast, deterministic response to its critical processes, on the order of microseconds rather than seconds.

High-Resolution Timers

For applications that deal with very short time intervals, Release 4 offers BSD system timing services, which give microsecond resolution. These services include alarms, interval timers, and a time-of-day clock. (Some of these services were available on earlier releases, but with lower timing resolution.)

Character-Based User Interface

An earlier release of UNIX System V introduced the Form and Menu Language Interpreter (FMLI)—a high-level language interpreter that allows developers to write user-friendly interfaces to their applications. Release 4 provides extensions to the Form and Menu Language, including a way to interrupt executables, a conditional statement (if-then-else), new built-in functions `test` and `expr`, and other improvements that give FMLI programmers more control over the appearance and behavior of their application interface.

Framed Access Command Environment (FACE), also introduced in an earlier release, is an interface that allows a user to see the UNIX system through frames containing menus and forms. In Release 4, FACE has been enhanced to be more consistent with the version of FACE developed for UNIX System V/386 Release 3.2, and adding applications to FACE has been made easier.

For more information about FMLI, see the *Programmer's Guide: Character User Interface (FMLI and ETI)*. For information about FACE, see the *User's Guide*.

Graphical User Interface and Windowing System

As part of the effort to encourage a standardization of the UNIX system, Release 4 offers programming tools to support the OPEN LOOK™ Graphical User Interface and the XWIN™ and X11/NeWS® graphical windowing systems.

XWIN Graphical Windowing System

The XWIN Graphical Windowing System is a portable, network-transparent windowing system. XWIN software creates a multi-layered server system on top of the UNIX system; it gives the user the ability to create multiple windows on a single display, with each window running different applications.

A windowing system keeps track of events and interfaces with client applications through a protocol system similar to the type of protocol system used by many network operations. XWIN software uses the X protocol for exchanging information between client applications and the server. The X protocol gives application programs running on different systems, and appearing in different windows, the ability to communicate with and use or display results from other application windows.

For more information, see the *Programmer's Guide: XWIN™ Graphical Windowing System*.

X11/NeWS® Graphical Windowing System

The X11/NeWS Graphical Windowing System runs applications written to the X11 and NeWS protocols. Although the protocols are different, X11/NeWS provides an integrated environment in which both are supported, with both working off a single window manager. This makes the GUI an extremely versatile graphical interface, offering more choices for both application developers and end users.

For more information, see the *Programmer's Guide: X11/NeWS® Graphical Windowing System*.

OPEN LOOK™ Graphical User Interface

The OPEN LOOK Graphical User Interface defines a standard for the appearance and function of the graphical user interface and provides developers with application programmer interface toolkits (APIs). APIs allow developers to manipulate windows and window-supported graphics to achieve the standard “look and feel” of OPEN LOOK Graphical User Interface applications. Two toolkits are provided as part of the OPEN LOOK software—one for writing applications that operate on the XWIN server, and one for writing X11/NeWS applications.

For more information, see the *Programmer's Guide: OPEN LOOK™ Graphical User Interface*, *OPEN LOOK™ Graphical User Interface Programmer's Reference Manual*, *OPEN LOOK™ Graphical User Interface User's Guide*.

Internationalization

With Release 4, AT&T continues to provide support for international environments in the UNIX system. New features in Release 4 that reflect the internationalization effort include

- support for multiple international code sets and multi-byte characters, and support for character (as opposed to byte) processing
- a message handling facility, which provides a means of translating system messages to the user's native language
- support for national conventions (forms and rules used to communicate information).

For information about features that support international environments, see the *Migration Guide*.

C Language

Release 4 incorporates enhancements to packages known by the following names in earlier releases: C Programming Language Utilities (CPLU), Advanced Programming Utilities (APU), and C Programmer Productivity Tools (CPPT) packages. The major features are:

- Compilation modes corresponding to degrees of compliance with ANSI C.
- Dynamic linking, which allows different processes to share library code at run time.
- Dynamic tables, which provide the necessary support for compilation of applications with very large tables (e.g. macros, symbols, etc.).
- Additional international support, allowing the international date and time to be set as a default, and support for customized internal character sets.
- A new transparent object file format, ELF (Executable and Linking Format), which supports extensions to the C language, such as new types. ELF is flexible enough to support other programming languages, particularly C++.
- Support tools for ELF and conversion tools for converting COFF (Common Object File Format) to ELF. All tools handle both COFF and ELF objects and archives.

See the *Programmer's Guide: ANSI C and Programming Support Tools*, the *ANSI C Transition Guide*, and the *Programmer's Reference Manual* for more information.

Extended Terminal Interface

The Extended Terminal Interface (ETI) is a standard programming interface provided in Release 4 for character mode screen management and text operations. ETI consists of three libraries, including the `curses` library and the `terminfo` data base, and the High Level Function library, which provides C subroutines for creating forms, menus, form editing functions, and panels.

See the *Programmer's Guide: Character User Interface (FMLI and ETI)* for more information.

Product Description

The UNIX System V/386, Release 4 operating system is made up of several software packages. These software packages logically separate the software into the Base System, consisting of the most basic components, and a set of add-on packages. Each add-on package provides a special set of features, such as networking facilities, programming tools, or editing utilities.

The vendor that produced your computer may not offer all of these packages or it may offer different packages. The UNIX system available for most 386-based computers, however, includes the Base System, the Standard C Development Environment Package (for software development), and a selection from the other packages.

The remainder of this section describes the UNIX System V/386, Release 4 software and the document set.

Utilities Packages

The utilities packages that make up the UNIX System V/386, Release 4 product are listed below. Following the list are descriptions of the software that make up each package.

- Archive XL Floppy Tape Utilities Package
- Base System
- BSD Compatibility Package
- Cartridge Tape Utilities Package
- Distributed File System Package
- Editing Utilities Package
- Forms and Menu Language Interpreter Package
- Framed Access Command Environment Package
- Graphics End User System Package
- Graphics Programmer's Package
- Kernel Debugger Package

- Line Printer Spooler Package
- Mouse Support Package
- Network File System Package
- Network Support Utilities Package
- OA&M Basic and Extended Package
- PC586 Ethernet Package
- PC-Interface Ethernet Service Package
- PC-Interface Utilities & RS-232 Service Package
- Remote File Sharing Utilities Package
- Remote Procedure Call Package
- Security Administration Package
- Standard C Development Environment Package
- Termcap Package
- Termino Package
- TCP/IP Package
- Unlimited User Upgrade Package
- WD8003 Ethernet Driver
- Windowing Package
- X11/NeWS Graphical Windowing System
- X11/NeWS Toolkit
- XENIX Compatibility Package

Archive XL Floppy Tape Utilities Package

The Archive XL Floppy Tape Utilities package supports Archive model 3800 streaming 1/4-inch mini cartridge tape drives, or any floppy tape drive that supports the QIC-107 physical and QIC-117 logical interface specifications and QIC-80 or QIC-40 recording format standards.

Base System

The Base System package consists of the operating system kernel, a minimum set of device drivers, and the most essential and commonly-used commands. The Base System also includes installation and configuration management utilities and message management and monitoring utilities.

The Base System has also bundled in components that were previously part of the Basic Networking Utilities package. Basic networking utilities enable a UNIX system to communicate with other systems in the following ways:

- transfer files and send electronic mail to other UNIX systems
- communicate interactively with others using UNIX systems or non-UNIX systems
- execute a restricted set of commands on a remote system without directly logging in on the remote system
- call and log in to a remote system
- call a remote terminal and allow the user of the terminal to log in on a UNIX system.

For additional information about the software in the Base System package and how to use it, see the *User's Guide*, the *System Administrator's Guide*, the Base System manual pages in the *User's Reference Manual* and *System Administrator's Reference Manual*.

BSD Compatibility Package

The BSD Compatibility package contains BSD commands and library routines that were not merged with the UNIX System V Base System. It also includes software for `sendmail` and the BSD versions of `nroff`, `troff`, and `lp`.

For more information about the BSD Compatibility package, see the *BSD/XENIX® Compatibility Guide*.

Cartridge Tape Utilities Package

The Cartridge Tape Utilities Package contains a cartridge tape driver that supports both Wangtek PC-36 and Archive SC499-R cartridge tape controllers.

See the `qt` manual page in the *System Administrator's Reference Manual* for more information.

Distributed File System Package

The Distributed File System package provides a common administrative interface to Remote File Sharing (RFS) and Network File System (NFS). Commands installed by the DFS Administration Utilities package allow an administrator to share and mount both RFS and NFS resources. DFS utilities must be installed to run RFS and NFS.

See the *Network User's and Administrator's Guide* for more information.

Editing Utilities Package

The Editing Utilities package provides three related editors: two line editors (`edit` and `ex`) and a screen editor (`vi`). The `edit` editor is mainly for novice users. The `ex` editor is an advanced version of `edit` and is for experienced users. The `vi` editor is intended for all users; it allows the user to view a screen of text at one time and move the terminal's cursor directly to any position in the file.

Editing Utilities also include the `spell` command. `spell` is used to check files for spelling errors.

For more information, see the editing tutorials in the *User's Guide* and the Editing Utilities manual pages in the *User's Reference Manual*.

Form and Menu Language Interpreter Package

Form and Menu Language Interpreter (FMLI) package provides the FMLI high-level language interpreter that allows developers to write user-friendly interfaces to their applications.

For more information about FMLI, see the *Programmer's Guide: Character User Interface (FMLI and ETI)*.

Framed Access Command Environment Package

Framed Access Command Environment (FACE) is a package that provides a character-based menu interface to the UNIX system intended for non-expert computer users. It provides a hierarchy of forms and menus through which a user can execute basic UNIX system commands for file editing and text manipulation. It also includes tools for creating menus of services through which a user can access application packages.

See the *User's Guide* for a tutorial that explains how to use FACE.

Graphics End User System Package

The Graphics End User System package contains all end-user facilities for the OPEN LOOK graphical user interface and the XWIN graphical windowing system.

For more information, see the *OPEN LOOK Graphical User Interface User's Guide*.

Graphics Programmer's Package

The Graphics Programmer's package contains all facilities needed to write OPEN LOOK and XWIN graphical application programs.

For more information, see the *OPEN LOOK Graphical User Interface Programmer's Reference Manual*, *Programmer's Guide: OPEN LOOK Graphical User Interface*, and *Programmer's Guide: XWIN™ Graphical Windowing System*.

Kernel Debugger Package

The Kernel Debugger package contains utilities and drivers needed to do low-level debugging of the UNIX Operating System. This package is not generally available to end users and, therefore, no end-user documentation is made available.

Line Printer Spooler Package

The Line Printer (LP) Spooling Utilities package provides the temporary storing (spooling) of data until the data can be printed.

With LP spooling, print requests can be entered at a terminal and printing can take place without tying up the terminal. Spooling also enables many users to share a printer, or several printers, efficiently. The commands in this package can perform the following functions:

- customize the system so that it will spool to a group of line printers
- group printers into logical classes to maximize the throughput of the printers
- queue print requests, thus allowing a print request (job) to get printed on the next available printer in the appropriate class
- cancel print requests, so a job that is no longer needed will not be printed
- start and stop the LP spooling software from processing requests
- change the configuration of printers
- find the status of the LP scheduler
- restart any printing that was not completed when the system was powered down
- connect multiple computers into a network so that the users of one computer can use any of the printers connected to the other computers in the network.

For information about using this package, see the *User's Guide* and the *User's Reference Manual*. For information about administering this package, see the *System Administrator's Guide* and the *System Administrator's Reference Manual*.

Mouse Support Package

The Mouse Support Package consists of the software necessary to use a mouse pointing device with UNIX® System V/386 Release 4. The package contains the device driver and administration and files to assist in administering mouse pointing devices. This new version of the mouse driver makes it possible to interface a mouse pointing device through the serial ports on any Fiber Optic Station (FOS) terminal or the system console.

For information about using this package, see the *Mouse Driver Administrator's Guide*.

Network File System Package

The Network File System package provides the facilities needed to share resources among computers in a heterogeneous environment. It requires the Networking Support Utilities package, UDP/IP (transport protocols provided by the Internet Utilities package), and the Distributed File System Administration Utilities package.

For more information about Network File System, see the *Network User's and Administrator's Guide*.

Network Support Utilities Package

The Network Support Utilities package installs the STREAMS mechanism as well as software support for Network Selection, the Service Access Facility, and Transport Interface (TI).

Service Access Facility utilities include the network daemon `listen`, as well as commands for administering it.

The Networking Support Utilities package must be installed to take advantage of services provided by the Remote File Sharing Utilities package and Network File System Utilities package.

OA&M Basic and Extended Package

The OA&M Basic and Extended package contains the facilities needed to administer the system through the `sysadm` command interface. The `sysadm` interface consists of a series of menus and forms intended to simplify administration of the UNIX system.

For more information on the `sysadm` interface, see the *System Administrator's Guide*.

PC586 Ethernet Package

The PC586 Ethernet package contains the device driver needed to use the Intel iMX-LAN/586 board.

PC-Interface Ethernet Service Package

The PC-Interface Ethernet Service package contains an Ethernet driver that can be used with the PC-Interface Utilities. The Ethernet driver can be used instead of the standard RS-232 interface that comes with the PC-Interface Utilities package.

For further information, see the PC-Interface Administrator's Guide.

PC-Interface Utilities & RS-232 Service Package

The PC-Interface Utilities package enables you to run DOS applications by using data from the UNIX system file system (host). It also allows you to store data and run DOS applications on the UNIX system host, just as you would on a local fixed disk.

For further information, see the PC-Interface Administrator's Guide.

Remote File Sharing Utilities Package

The Remote File Sharing Utilities package provides the facilities needed to share resources transparently among computers running UNIX System V. It requires the Networking Support Utilities package, the Distributed File System Administration Utilities, and a transport provider that conforms to the AT&T Transport Provider Interface, such as the Starlan Network or TCP/IP (provided in the Internet Utilities package).

For more information about Remote File Sharing, see the *Network User's and Administrator's Guide*.

Remote Procedure Call Package

The Remote Procedure Call package provides a remote operation capability. The package includes RPC, XDR, and sockets libraries.

For more information about RPC, see the *Programmer's Guide: Networking Interfaces*.

Security Administration Package

The Security Administration package provides an encryption mechanism for protecting information stored in the computer. Using the facilities of this package gives additional protection beyond that obtained through login IDs, passwords, and permission modes. When this package has been installed, the encryption mechanism is available by using the `crypt` command, as well as in the editors `ed`, `edit`, `ex`, and `vi`.

The Encryption Utilities package has restricted distribution and is provided only with computers sold within the United States.

For more information, see the `crypt(1)`, `ed(1)`, `edit(1)`, `ex(1)`, and `vi(1)` manual pages in the *User's Reference Manual*.

Standard C Development Environment Package

The Standard C Development Environment (SCDE) is a set of programming tools that help programmers develop C language programs. The main component is the compiler, called by the command `cc`. The `cc` command automatically calls the assembler and the link editor, as needed. SCDE provides a wide range of tools to aid C language programmers, such as profilers, a debugger, the `lint` program, and many more.

The Extended Terminal Interface (ETI) and the Source Code Control System (SCCS) are included as part of SCDE. ETI is a set of screen management library subroutines [built on `curses(3X)`] that promote fast development of application programs that manipulate windows, panels, menus, and forms. SCCS is a set of programs used to track evolving versions of source files and text files.

The major enhancements to the C programming language for this release fall into these categories: conformance with the American National Standards Institute (ANSI) X3.159-1989 C language standard; transition to dynamic linking; transition to ELF (Executable and Linking Format) from COFF (Common Object File Format); and new features added to aid in programming for international uses and environments.

See the *Programmer's Guide: ANSI C and Programming Support Tools*, the *Programmer's Guide: Character User Interface (FMLI and ETI)*, the *ANSI C Transition Guide*, and the *Programmer's Reference Manual* for more information.

Termcap Package

The Termcap Package contains the `termcap` file. The `termcap` file is required by some XENIX applications and utilities to define terminal settings.

Terminfo Package

The Terminfo package contains a database (`terminfo`) that allows programmers to write programs to manipulate screens (and parts of screens, such as function keys and soft labels) of video display terminals.

As delivered, the `terminfo` database contains descriptions of over 150 terminals.

For more information, see the *Programmer's Guide: Character User Interface (FMLI and ETI)*.

TCP/IP Package

The TCP/IP package installs the TCP/IP network protocols, also known as the DARPA Internet Protocol Suite, as well as commands that allow users to login to remote systems, execute commands on remote systems, and copy files from remote systems.

For information about TCP/IP, see the *Network User's and Administrator's Guide*.

Unlimited User Upgrade Package

Some computers running UNIX System V are configured to allow a limited number of users to access the system simultaneously. The Unlimited User Upgrade package is used to increase the number of simultaneous users allowed on a system.

Windowing Package

The Windowing package provides the software that is commonly required by AT&T windowing terminals (such as the AT&T 630 MTG) to create, delete, and manipulate terminal windows, query terminal window status, and display statistics about usage of windowing routines.

For more information about this package, see the `layers(1)` manual page in the *User's Reference Manual* and the `libwindows(3X)` manual page in the *Programmer's Reference Manual*.

WD8003 Ethernet Driver

The WD8003 Ethernet package contains the device driver needed to use the Western Digital WD8003 board.

X11/NeWS Graphical Windowing System Package

This package contains all end-user X11/NeWS facilities. These facilities allow an end user to run X11/NeWS applications.

X11/NeWS Toolkit Package

The X11/NeWS Toolkit Package contains facilities needed by programmers to create X11/NeWS applications.

For more information see the *Programmer's Guide: X11/NeWS Graphical Windowing System*.

XENIX Compatibility Package

The XENIX Compatibility Package contains commands that are specific to applications developed for XENIX systems and XENIX system commands that will not be supported beyond Release 4. These include the XENIX system file and record locking interface (`locking`) and system calls that support the XENIX system shared memory and semaphores (contained in a library called `libx`).

For more information about the XENIX Compatibility Package, see the *BSD/XENIX® Compatibility Guide*.

The Release 4 Document Set

A full set of documentation is available to end users, system administrators, and programmers for UNIX System V, Release 4 on the porting base processor—the AT&T 3B2/400 computer. Many of these documents have been tailored to suit computers based on the Intel 386 processor. All UNIX System documents described in this book are available from Prentice-Hall or one of the many computer vendors that reproduce the documents to suit their systems.

The Release 4 document set includes both guides and reference manuals. Guides contain conceptual and procedural information; they tell when and why to do something, as well as how to do it. A guide is usually the best introductory text (tutorials are often included), yet it is organized so that experienced users can easily skip to the information they need.

Reference manuals contain complete descriptions (traditionally known as “manual pages” or “man pages”) of commands, utilities, system calls, library functions, or system file formats. The assumption in a reference manual is that the user already knows when and why a task should be done. Although descriptions are complete, every attempt has been made to keep them to one page; as a result they are sometimes cryptic. A reference manual is an efficient source of information once you’ve become familiar with the product.

Below is a list of the volumes that make up the Release 4 document set.

GENERAL USE AND SYSTEM ADMINISTRATION

- Product Overview and Master Index
- User’s Guide
- User’s Reference Manual
- System Administrator’s Guide
- System Administrator’s Reference Manual
- Network User’s and Administrator’s Guide
- Mouse Driver Administrator’s Guide
- MULTIBUS Installation and Configuration Guide
- PC-Interface Administrator’s Guide

PROGRAMMING

General Programmer’s Series

- Programmer’s Guide: ANSI C and Programming Support Tools
- Programmer’s Guide: System Services and Application Packaging Tools

- Programmer's Guide: Character User Interface (FMLI and ETI)
- Programmer's Guide: POSIX Conformance
- Programmer's Guide: Networking Interfaces
- Programmer's Reference Manual
- Programmer's Guide: SCSI Driver Interface

Graphics Series

- Programmer's Guide: OPEN LOOK™ Graphical User Interface
- OPEN LOOK™ Graphical User Interface Programmer's Reference Manual
- OPEN LOOK™ Graphical User Interface User's Guide
- Programmer's Guide: XWIN™ Graphical Windowing System
- Programmer's Guide: X11/NeWS® Graphical Windowing System

System Programmer's Series

- Programmer's Guide: STREAMS
- Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual
- Integrated Software Development Guide
- MULTIBUS Reference Manual
- Transport Application Interface Guide

RELEASE-SPECIFIC INFORMATION

General

- Migration Guide
- ANSI C Transition Guide
- Software Notes
- BSD/XENIX® Compatibility Guide

Product Overview and Master Index

The product overview provides a brief introduction to UNIX System V and a summary of Release 4 features. There is also a master permuted index and a master subject index.

User's Guide

The *User's Guide* describes what you need to know to get started using the UNIX system, including how to use the basic commands to create and manipulate files. In addition, it includes tutorials describing *FACE*, *ed*, *vi*, the shell, *awk*, and how to send mail and files to other users. It also includes *vi* quick reference information.

User's Reference Manual

The *User's Reference Manual* describes each of the UNIX System V user commands. Each description includes a synopsis of the command syntax and an explanation of how it is used; also supplied where appropriate are diagnostic indications, warnings, examples of use, and where to find related information.

System Administrator's Guide

The *System Administrator's Guide* explains how to perform administrative tasks, including managing file systems and users, setting up basic networking, maintaining security, tuning the system to optimize performance, and administering printers. Tasks are organized by the menu layout of *sysadm*, the administration interface.

System Administrator's Reference Manual

The *System Administrator's Reference Manual* describes each of the UNIX System V administration commands. Each description includes a synopsis of the command syntax, an explanation of usage, and, where appropriate, diagnostic indications and warnings.

Network User's and Administrator's Guide

The *Network User's and Administrator's Guide* tells how to use and administer the network facilities provided with UNIX System V. It includes descriptions of administering and using Remote File Sharing (RFS), Network File Sharing (NFS), and TCP/IP.

Mouse Driver Administrator's Guide

The *Mouse Driver* documentation describes the installation, configuration, and administration of the AT&T Mouse Driver Package.

MULTIBUS Installation & Configuration Guide

This document explains how to install and configure MULTIBUS UNIX on MULTIBUS systems.

PC-Interface Administrator's Guide

The *PC-Interface Administrator's Guide* describes how to set up the server portion (UNIX) of an interface.

Programmer's Guide: ANSI C and Programming Support Tools

This guide discusses the UNIX system programming environment and utilities (e.g., compilers, debuggers) and provides details of the C language, file formats, the link editor, libraries, and tools.

Programmer's Guide: System Services and Application Packaging Tools

This guide tells how to develop an application package under UNIX System V. It describes how to use the UNIX system services provided by the kernel, such as file and record locking, interprocess communication, the process scheduler, and symbolic links. It also describes how to use libraries of routines in software development in general, and standard tools to package application software for easy installation on a running system.

Programmer's Guide: Character User Interface (FMLI and ETI)

This guide describes tools a programmer can use to interface with a user at a terminal that does not have graphics capabilities. These tools range from an interpretive language that makes it easy to develop forms and menus (FMLI) to libraries of routines (ETI/curses) that enable the programmer to work with high-level constructs, such as windows, or do low-level character placements. Routines remain independent of terminals by using a database that describes the capabilities of a terminal and the way it performs various operations.

Programmer's Guide: POSIX Conformance

In accordance with POSIX documentation requirements, this guide describes how UNIX System V conforms to POSIX and provides the implementation-specific details of that conformance.

Programmer's Guide: Networking Interfaces

This guide describes the Transport Level Interface (TLI), its capabilities and applications, and other network development tools, such as RPC and the Network Selection facility.

Programmer's Reference Manual

The *Programmer's Reference Manual* contains manual pages for UNIX System V programming commands, libraries, system calls, file formats, and miscellaneous information used by programmers.

Programmer's Guide: SCSI Driver Interface

This document describes the Small Computer Systems Interface (SCSI) Driver Interface, also known as SDI. SDI is a machine-independent mechanism for writing SCSI target drivers to access an SCSI device.

Programmer's Guide: OPEN LOOK™ Graphical User Interface

The programmer's guide provides overview information and a series of programs that acts as a guide to learning how to program with widgets. It also provides information about interoperability between clients and server.

OPEN LOOK™ Graphical User Interface Programmer's Reference Manual

The programmer's reference manual provides a manual page for each OPEN LOOK widget. Each page explains a widget function and the various resources which create that functionality. A description of different convenience routines for the application programmer to use while writing OPEN LOOK GUI applications is also provided.

OPEN LOOK™ Graphical User Interface User's Guide

In addition to end user information, the user's guide contains information valuable to installers and system administrators. The user's guide provides detailed descriptions of the various client programs which run on OPEN LOOK GUI as well as manual pages which describe the various demonstration packages that are shipped as part of the software product.

Programmer's Guide: XWIN™ Graphical Windowing System

Xlib is the C language interface and the lowest level interface to X programming. The document explains display functions, connecting to the X server, window functions, graphics functions, window manager functions, events handling, and application utility functions.

Programmer's Guide: X11/NeWS® Graphical Windowing System

This guide assumes the reader is reasonably familiar with the Adobe PostScript® language. It describes the initialization process and gives examples of how to modify an initialization routine by customizing PostScript language files. It also discusses different ways to run NeWS graphical windowing system programs.

Programmer's Guide: STREAMS

The *Programmer's Guide: STREAMS* describes the user-level STREAMS facilities and explains how to use STREAMS facilities to write UNIX System V kernel modules and device drivers. It includes a summary of kernel-level data structures, STREAMS message types, and specifications of kernel utility routines.

Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual

The *Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual* provides reference information for creating, modifying, and maintaining drivers that run on UNIX System V Release 4. It defines the entry point routines that must be written, the kernel functions that should be used, and the data structure with which the drivers interact.

Integrated Software Development Guide

This document addresses the needs of the independent software vendor, the value-added reseller, and system developers. It presents guidelines for software installation, device drivers, file system types, video interface, mouse interface, user interface, and internationalization.

MULTIBUS Reference Manual

This document provides manual page descriptions and device driver information for the MULTIBUS features.

Transport Application Interface Guide

This document describes the application interface to MULTIBUS II transport. The MULTIBUS II transport uses MULTIBUS II message passing.

Transport-application interface (TAI) is used by applications that need to communicate, via MULTIBUS II transport protocol, with remote applications.

Migration Guide

The *Migration Guide* presents an overview of Release 4 features and describes the major differences between Release 4 and previous releases of UNIX System V.

ANSI C Transition Guide

The *ANSI C Transition Guide* describes techniques for writing new and upgrading existing C code to comply with the ANSI C language specification.

Software Notes

Software Notes offers additional information about UNIX System V Release 4, including operating and programming tips and miscellaneous information that was not available in time to include in the Release 4 guides and manuals.

BSD/XENIX® Compatibility Guide

The *BSD/XENIX® Compatibility Guide* documents the supplemental BSD system and XENIX system commands and features that are provided by the BSD Compatibility Package and the XENIX Compatibility Package. The manual contains both user-level guide material and manual pages.

Related Products

AT&T provides a comprehensive collection of software products that work with UNIX System V Release 4. Products fall into a number of categories, including networking, programming, database management, and general applications.

For specific information about software products, see the *AT&T Computer Software Catalog*.

2. PERMUTED INDEX

COMPUTED INDEX

2 Permuted Index

Using the Permuted Index	2-1
How the Permuted Index Is Created	2-1

Permuted Index	2-6
-----------------------	-----

Using the Permuted Index

The comprehensive permuted index directs you to manual pages located throughout the UNIX System Documentation Set with the exception of the graphics books. Manual pages are concise reference pages that assume some familiarity with the information. Detailed discussions and procedures on select topics are covered in the guide volumes and can be found using the subject index.

How the Permuted Index Is Created

Many users find that understanding a few things about how the permuted index is created helps them to read it more effectively and clarifies what kind of information can and cannot be obtained from it.

The basic building block for the index is the one-line description given in the NAME line on the top of each manual page. For example, this is what the top of the `mountall(1M)` manual page looks like:

mountall(1M)	(Essential Utilities)	mountall(1M)
NAME		
<code>mountall</code> , <code>umountall</code> - mount, unmount multiple file systems		

Each NAME line includes:

- the command, file format, system call or other utility for which the manual page is named (this is the primary utility; `mountall` is the primary utility in the example)
- secondary utilities, which are also described on that manual page and do not have a separate manual page of their own (`umountall` is a secondary utility in the example)
- a brief description of the utility function(s)

For each manual page NAME line, the indexing software generates several index entries, generally one entry for each keyword in the phrase. The middle column of the index is alphabetized on these keywords.

For:

NAME

mountall, umountall - mount, unmount multiple file systems

This is generated:

mount, unmount multiple systems.	mountall, umountall: unmount multiple file systems.	/umountall:	mountall(1M)	SARM
mount, unmount multiple file systems.	mountall, umountall: mount, /umountall: mount, unmount multiple file systems.	mountall(1M)	SARM
mount, unmount multiple file multiple file/ mountall, mountall, umountall: mount,	mount, unmount multiple file systems.	mountall, umountall:	mountall(1M)	SARM
	mountall, umountall: mount, unmount multiple file systems.	mountall(1M)	SARM

How to Use the Index

Look in the middle column of the index for the word of interest. Then read the complete phrase by starting with the utility name, which may appear in the left or middle column. Utility names are followed by a colon. A period marks the end of the description from the NAME line. (See page 3-3 for information about what the section numbers mean.)

The NAME line phrase is contained in the two columns, with long phrases wrapping around to the beginning of the left column. The right column of the index provides the manual page name, section number, and acronym for the book where it is located.

A slash (/) sometimes appears in the index entry to indicate that space limitations were exceeded and one or more words from the phrase were deleted.

Book Acronyms Used in the Permuted Index

BSD	BSD Compatibility Guide
CHAR	Programmer's Guide: Character User Interface
DDRM	Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual
MBRMAN	MULTIBUS Reference Manual
MOUSE	Mouse Driver Administrator's Guide
NI	Programmer's Guide: Networking Interfaces
NUAG	Network User's and Administrator's Guide
PRM	Programmer's Reference Manual
SARM	System Administrator's Reference Manual
SCSI	Programmer's Guide: SCSI Driver Interface
SS	Programmer's Guide: System Services and Application Packaging Tools
STRM	Programmer's Guide: STREAMS
TAI	Transport Application Interface Guide
URM	User's Reference Manual
XNX	XENIX Compatibility Guide

Quick Reference to Manual Page Section Numbers

Section Number	Type of Utility
(1)	User Commands
(1C)	Basic Networking Commands

- (1F) FMLI Commands
- (1M) Administration Commands
- (2) System Calls
- (3) BSD Routines
- (3C) C Library Functions

Section Number	Type of Utility
(3E)	libelf Functions
(3G)	libgen Functions
(3M)	Math Library Functions
(3N)	Network Services Functions
(3S)	Standard I/O Functions
(3X)	Specialized Libraries
(4)	File Formats
(5)	Miscellaneous
(7)	Special Files
(8)	System Maintenance Procedures
(D2D)	DDI Driver Entry Points

- (D2DK) DDI/DKI Driver Entry Points
- (D3D) DDI Kernel Functions
- (D3DK) DDI/DKI Kernel Functions
- (D4D) DDI Data Structures
- (D4DK) DDI/DKI Data Structures

Permuted Index

I/O port. inw: read a
 read a 16 bit short word from a
 write a 16-bit short word to a
 I/O port. outw: write a
 I/O port. repoutsw: write
 buffer. repinsw: read
 disk device driver. i224a: iSBC
 tape device/ i224atp: iSBC
 controller device/ i410: iSBC
 device driver. i530: iSBC
 device driver. i214: iSBCR
 device driver. i214tp: iSBC
 Link driver for iNA961 Release
 inl: read a 32-bit word from a
 write a 32-bit long word to a
 port. outl: write a
 port. inl: read a
 buffer. repinsd: read
 I/O port. repoutsd: write
 i8251: console terminal/iSBXT
 device driver. i354: iSBX
 /u3b, u3b2, u3b5, u3b15, vax,
 device driver. i258: iSBC
 tape device/ i258tp: iSBC
 l3tol, ltol3: convert between
 comparison. diff3:
 compatibility/ tcompat: V7,
 communications interface/ ports:
 controller. i546: iSBC
 object downloader for the
 wd: Western Digital
 module formats. download: loads
 inb: read a byte from a
 outb: write a byte to an
 a message to be sent to initiate
 received request that is part of
 integer and base-64 ASCII/
 abort: generate an
 termination signal.
 absolute value.
 module/ download: loads 8086
 abs, labs: return integer
 /floor, ceiling, remainder,
 t_accept:
 accept:
 socket.
 /whether remote system can
 16 bit short word from a 16-bit inw(D3D) DDRM
 16-bit I/O port. inw: inw(D3D) DDRM
 16-bit I/O port. outw: outw(D3D) DDRM
 16-bit short word to a 16-bit outw(D3D) DDRM
 16-bit words from buffer to an repoutsw(D3D) DDRM
 16-bit words from I/O port to repinsw(D3D) DDRM
 186/224A peripheral controller i224a(1M) MBRMAN
 186/224A peripheral controller i224atp(1M) MBRMAN
 186/410 serial communications i410(1M) MBRMAN
 186/530 Ethernet controller i530(1M) MBRMAN
 214 peripheral controller disk i214(1M) MBRMAN
 214 peripheral controller tape i214tp(1M) MBRMAN
 3.0.. edlina: External Data edlina(1M) MBRMAN
 32-bit I/O port. inl(D3D) DDRM
 32-bit I/O port. outl: outl(D3D) DDRM
 32-bit long word to a 32-bit I/O outl(D3D) DDRM
 32-bit word from a 32-bit I/O inl(D3D) DDRM
 32-bit words from I/O port to repinsd(D3D) DDRM
 32-bit words from buffer to an repoutsd(D3D) DDRM
 351 driver. i8251(1M) MBRMAN
 354 dual channel serial-port i354(1M) MBRMAN
 386, u370: get processor type/ machid(1) URM
 386/258 peripheral controller i258(1M) MBRMAN
 386/258 peripheral controller i258tp(1M) MBRMAN
 3-byte integers and long/ l3tol(3C) PRM
 3-way differential file diff3(1) URM
 4BSD and XENIX STREAMS tcompat(7) SARM
 5 line asynchronous ports(7) STRM
 546 multi-port serial i546(1M) MBRMAN
 5620 DMD terminal. wtinit: wtinit(1M) SARM
 8003 Adapter Board. wd(7) SARM
 8086 absolute library and object download(1M) MBRMAN
 8-bit I/O port. inb(D3D) DDRM
 8-bit I/O port. outb(D3D) DDRM
 a. mps_mk_solrply: constructs mps_mk_solrply(D3D) DDRM
 a. /replies to a mps_AMPsend_reply(D3D) DDRM
 a64l, l64a: convert between long a64l(3C) PRM
 abnormal termination signal. abort(3C) PRM
 abort: generate an abnormal abort(3C) PRM
 abs, labs: return integer abs(3C) PRM
 absolute library and object download(1M) MBRMAN
 absolute value. abs(3C) PRM
 absolute value functions. floor(3M) PRM
 accept a connect request. t_accept(3N) NI
 accept a connection on a socket. accept(3N) NI
 accept: accept a connection on a accept(3N) NI
 accept binary messages. ckbinarsys(1M) SARM

accept, reject: accept or reject print requests. accept(1M) SARM
 print requests. accept(1M) SARM
 elf_rand: random archive member
 sdgetv: synchronize shared data
 files. settime: change the
 utime: set file
 a file. touch: update
 disk: random
 face: executable for the Framed
 sac: service
 sac: service
 administration. sacadm: service
 administration. sacadm: service
 cred:
 of a file.
 sequential archive member
 elf: object file
 get or set supplementary group
 the supplementary group
 sputl, sgetl:
 /icsslot, icssgetrec: utilities to
 fusage: disk
 fusage: disk
 sulogin:
 copy file systems for optimal
 copy s5 file systems for optimal
 close: relinquish
 open: gain
 start: start
 a/ /nbwaitsem: await and check
 sdenter, sdleave: synchronize
 useracc: verify whether user has
 pseudo-terminal/ grantpt: grant
 /setutent, endutent, utmpname:
 /getutmpx, updwtmp, updwtmpx:
 getvol: verifies device
 access: determine
 acct: enable or disable process
 runacct: run daily
 acctcon1, acctcon2: connect-time
 acctprc1, acctprc2: process
 diskusg: generate disk
 acct: per-process
 acctmerg: merge or add total
 search and print process
 command summary from per-process
 wtmpfix: manipulate connect
 access controller. sac(1M) NI
 access controller. sac(1M) SARM
 access controller sacadm(1M) NI
 access controller sacadm(1M) SARM
 access credential structure. cred(D4DK) DDRM
 access: determine accessibility access(2) PRM
 access. elf_next: elf_next(3E) PRM
 access library. elf(3E) PRM
 access list IDs. /setgroups: getgroups(2) PRM
 access list. /initialize initgroups(3C) PRM
 access long integer data in a/ sputl(3X) PRM
 access MULTIBUS II interconnect/ icprd(1M) MBRMAN
 access profiler. fusage(1M) NUAG
 access profiler. fusage(1M) SARM
 access single-user mode. sulogin(1M) SARM
 access time. dcopy (generic): dcopy(1M) SARM
 access time. dcopy (s5): dcopy(1M) SARM
 access to a device. close(D2DK) DDRM
 access to a device. open(D2DK) DDRM
 access to a device. start(D2D) DDRM
 access to a resource governed by waitsem(2) XNX
 access to a shared data segment. sdenter(2) XNX
 access to memory. useracc(D3DK) DDRM
 access to the slave grantpt(3C) STRM
 access utmp file entry. getut(3C) PRM
 access utmpx file entry. getutx(3C) PRM
 accessibility. getvol(1M) SARM
 accessibility of a file. access(2) PRM
 accounting. acct(2) PRM
 accounting. runacct(1M) SARM
 accounting. acctcon, acctcon(1M) SARM
 accounting. acctprc, acctprc(1M) SARM
 accounting data by user ID. diskusg(1M) SARM
 accounting file format. acct(4) SARM
 accounting files. acctmerg(1M) SARM
 accounting file(s). acctcom: acctcom(1) URM
 accounting records. acctcms: acctcms(1M) SARM
 accounting records. fwtmp, fwtmp(1M) SARM

Permuted Index

turnacct: shell procedures for
accton, acctwtmp.
accounting.
file format.
per-process accounting records.
process accounting file(s).
connect-time accounting.
accounting. acctcon,
accounting. acctcon, acctcon1,
acctwtmp. acct:
acct: acctdisk, acctdusg,
accounting files.
acct: acctdisk, acctdusg,
process accounting.
accounting. acctprc,
acctprc, acctprc1,
acctdisk, acctdusg, accton,
pkgchk: check
pkgchk: check
orderly release/ t_rcvrel:
/cosf, tan, tanf, asin, asinf,
/tan, tanf, asin, asinf, acos,
cosh, coshf, tanh, tanhf, asinh,
killall: kill all
channel. dma_stop: stop DMA
sag: system
event. sleep: suspend process
sar: sa1, sa2, sadc: system
sar: system
print current SCCS file editing
report process data and system
sdi_init: initialize the host
wd: Western Digital 8003
mvwaddch, echochar, wechochar:
installation/ installf:
curses/ /mvwaddstr, mvwaddnstr:
structure. ureadc:
definition on the/ groupadd:
device driver/ idinstall:
logger:
atexit:
/mvwaddchstr, mvwaddchnstr:
acctmerg: merge or
putenv: change or
bibliographic database.
mvwaddch, echochar,/ curs_addch:
curs_addchstr: addchstr,
accounting. /shutacct, startup, acctsh(1M) SARM
acct: acctdisk, acctdusg, acct(1M) SARM
acct: enable or disable process acct(2) PRM
acct: per-process accounting acct(4) SARM
acctcms: command summary from acctcms(1M) SARM
acctcom: search and print acctcom(1) URM
acctcon, acctcon1, acctcon2: acctcon(1M) SARM
acctcon1, acctcon2: connect-time acctcon(1M) SARM
acctcon2: connect-time acctcon(1M) SARM
acctdisk, acctdusg, accton, acct(1M) SARM
acctdusg, accton, acctwtmp. acct(1M) SARM
acctmerg: merge or add total acctmerg(1M) SARM
accton, acctwtmp. acct(1M) SARM
acctprc, acctprc1, acctprc2: acctprc(1M) SARM
acctprc1, acctprc2: process acctprc(1M) SARM
acctprc2: process accounting. acctprc(1M) SARM
acctwtmp. acct: acct(1M) SARM
accuracy of installation. pkgchk(1M) SARM
accuracy of installation. pkgchk(1M) SS
acknowledge receipt of an t_rcvrel(3N) NI
acos, acosf, atan, atanf, atan2,/ trig(3M) PRM
acosf, atan, atanf, atan2,/ trig(3M) PRM
acosh, atanh: hyperbolic/ /sinhf, sinh(3M) PRM
active processes. killall(1M) SARM
activity and release the dma_stop(D3D) DDRM
activity graph. sag(1) URM
activity pending execution of an sleep(D3DK) DDRM
activity report package. sar(1M) SARM
activity reporter. sar(1) URM
activity. sact: sact(1) PRM
activity. /time a command; timex(1) URM
adapter. sdi_init(D3I) SCSI
Adapter Board. wd(7) SARM
add a character (with/ /mvaddch, curs_addch(3X) CHAR
add a file to the software installf(1M) SS
add a string of characters to a curs_addstr(3X) CHAR
add character to a uio ureadc(D3DK) DDRM
add (create) a new group groupadd(1M) SARM
add, delete, update, or get idinstall(1M) SARM
add entries to the system log. logger(1) BSD
add program termination routine. atexit(3C) PRM
add string of characters (and/ curs_addchstr(3X) CHAR
add total accounting files. acctmerg(1M) SARM
add value to environment. putenv(3C) PRM
addbib: create or extend a addbib(1) BSD
addch, waddch, mvaddch, curs_addch(3X) CHAR
addchnstr, waddchstr,/ curs_addchstr(3X) CHAR

waddchnstr,/ curs_addchnstr: addchnstr, waddchnstr, curs_addchnstr(3X) CHAR
 mvaddstr,/ curs_addstr: addstr, waddstr, curs_addstr(3X) CHAR
 virtual address to physical address. /convert kernel kvtophys(D3D) DDRM
 get page frame number for kernel address. hat_getkpfnum: hat_getkpfnum(D3K) DDRM
 bcopy: copy data between address locations in the kernel. bcopy(D3DK) DDRM
 /inet_netof, inet_ntoa: Internet address manipulation. inet(3N) NI
 physmap: obtain virtual address mapping for physical/ physmap(D3DK) DDRM
 ethers: Ethernet address mapping operations. ethers(3N) NI
 ethers: Ethernet address mapping operations. ethers(3N) NUAG
 object. dlsym: get the address of a symbol in shared dlsym(3X) PRM
 control. arp: address resolution display and arp(1M) NUAG
 control. arp: address resolution display and arp(1M) SARM
 ARP: Address Resolution Protocol. ARP(7) NUAG
 ARP: Address Resolution Protocol. ARP(7) SARM
 server. rarpd: DARPA Reverse Address Resolution Protocol rarpd(1M) NUAG
 server. rarpd: DARPA Reverse Address Resolution Protocol rarpd(1M) SARM
 bp_mapin: allocate virtual address space. bp_mapin(D3DK) DDRM
 bp_mapout: deallocate virtual address space. bp_mapout(D3DK) DDRM
 mlockall: lock or unlock address space. mlockall, mlockall(3C) PRM
 t_bind: bind an address to a transport endpoint. t_bind(3N) NI
 domain. ethers: Ethernet address to hostname database or ethers(4) NUAG
 domain. ethers: Ethernet address to hostname database or ethers(4) SARM
 kvtophys: convert kernel virtual address to physical address. kvtophys(D3D) DDRM
 convert virtual to physical address. vtop: vtop(D3D) DDRM
 sdi_name: get name of addressed controller. sdi_name(D3I) SCSI
 aliases, addresses, forward: addresses and aliases for/ aliases(4) BSD
 and aliases for/ aliases: addresses, forward: addresses aliases(4) BSD
 address mapping for physical addresses. /obtain virtual physmap(D3DK) DDRM
 translate scb virtual addresses. sdi_translate: sdi_translate(D3I) SCSI
 mapper. rpcbind: universal addresses to RPC program number rpcbind(1M) NI
 mapper. rpcbind: universal addresses to RPC program number rpcbind(1M) NUAG
 severity levels for an/ addseverity: build a list of addseverity(3C) PRM
 waddchnstr,/ curs_addstr: addstr, addchnstr, waddstr, curs_addstr(3X) CHAR
 message. adjmsg: trim bytes from a adjmsg(D3DK) DDRM
 allow synchronization of the/ adjtime: correct the time to adjtime(2) PRM
 d258: i82258 ADMA device driver. d258(1M) MBRMAN
 SCCS files. admin: create and administer admin(1) PRM
 file. admin: installation defaults admin(4) SARM
 the system. useradd: administer a new user login on useradd(1M) SARM
 LP print service. lpfilter: administer filters used with the lpfilter(1M) SARM
 LP print service. lpforms: administer forms used with the lpforms(1M) SARM
 admin: create and administer SCCS files. admin(1) PRM
 Administration. SA: devices administered by System sa(7) SARM
 dispadmin: process scheduler administration. dispadmin(1M) SARM
 mouseadmin: mouse administration. mouseadmin(1) MOUSE
 pmadm: port monitor administration. pmadm(1M) NI
 pmadm: port monitor administration. pmadm(1M) SARM

network listener service
 Remote File Sharing domain
 Remote File Sharing domain
 devices administered by System
 service access controller
 service access controller
 interface to perform system
 uadmin:
 uadmin:
 sad: STREAMS
 swap: swap
 to a curses window and
 to a curses window and
 compile/ regexp: compile, step,
 compile/ regexp: compile, step,
 compile/ regexp: compile, step,
 fumount: forced unmount of an
 fumount: forced unmount of an
 dumbplot, gigipLOT, / plot,
 forms field has off-screen data
 alarm: set a process
 clock.
 /display application specific
 pathalias:
 mailalias: translate mail
 command; display its pathname or
 services: Internet services and
 services: Internet services and
 addresses and aliases for/
 the data base for the mail
 /forward: addresses and
 mode. /Confirm data is
 t_alloc:
 allocb:
 shared buffer. esballoc:
 target driver. sdi_getblk:
 kernel free/ kmem_zalloc:
 space management map. rmalloc:
 memory. kmem_alloc:
 Buffer Descriptor. dma_get_buf:
 Command Block. dma_get_cb:
 bp_mapin:
 kmem_free: free previously
 /release a previously
 mps_free_tid: frees a previously
 mps_get_msgbuf:
 administration. nlsadmin: nlsadmin(1M) SARM
 administration. rfadmin: rfadmin(1M) NUAG
 administration. rfadmin: rfadmin(1M) SARM
 Administration. SA: sa(7) SARM
 administration. sacadm: sacadm(1M) NI
 administration. sacadm: sacadm(1M) SARM
 administration. sysadm: visual sysadm(1M) SARM
 administrative control. uadmin(1M) SARM
 administrative control. uadmin(2) PRM
 Administrative Driver. sad(7) STRM
 administrative interface. swap(1M) SARM
 advance cursor. /attributes) curs_addch(3X) CHAR
 advance cursor. /of characters curs_addstr(3X) CHAR
 advance: regular expression regexp(5) PRM
 advance: regular expression regexp(5) SARM
 advance: regular expression regexp(3G) PRM
 advertised resource. fumount(1M) NUAG
 advertised resource. fumount(1M) SARM
 aedplot, bgplot, crtplot, plot(1G) BSD
 ahead or behind. /tell if form_data(3X) CHAR
 alarm clock. alarm(2) PRM
 alarm: set a process alarm alarm(2) PRM
 alarms and/or the "working"/ indicator(1F) CHAR
 alias file for FACE. pathalias(4) SARM
 alias names. mailalias(1) URM
 alias. which: locate a which(1) BSD
 aliases. services(4) NUAG
 aliases. services(4) SARM
 aliases, addresses, forward: aliases(4) BSD
 aliases file. /rebuild newaliases(1M) BSD
 aliases for sendmail. aliases(4) BSD
 aligned for efficient flyby dma_get_best_mode(D3D) DDRM
 alloca: memory allocator. alloca(3) BSD
 allocate a library structure. t_alloc(3N) NI
 allocate a message block. allocb(D3DK) DDRM
 allocate a message block using a esballoc(D3DK) DDRM
 allocate a SCSI block for the sdi_getblk(D3I) SCSI
 allocate and clear space from kmem_zalloc(D3DK) DDRM
 allocate space from a private rmalloc(D3DK) DDRM
 allocate space from kernel free kmem_alloc(D3DK) DDRM
 Allocate storage for a DMA dma_get_buf(D3D) DDRM
 Allocate storage for a DMA dma_get_cb(D3D) DDRM
 allocate virtual address space. bp_mapin(D3DK) DDRM
 allocated kernel memory. kmem_free(D3DK) DDRM
 allocated SCSI block. sdi_freeblk(D3I) SCSI
 allocated transaction id.. mps_free_tid(D3D) DDRM
 allocates a message buffer.. mps_get_msgbuf(D3D) DDRM

mps_get_tid: allocates transaction ids. mps_get_tid(D3D) DDRM
 sbrk: change data segment space allocation. brk, brk(2) PRM
 alloca: memory allocator. alloca(3) BSD
 memalign, valloc,: memory allocator. /realloc, calloc, malloc(3C) PRM
 mallopt, mallinfo: memory allocator. /realloc, calloc, malloc(3X) PRM
 block. allocb: allocate a message allocb(D3DK) DDRM
 system calls. siginterrupt: allow signals to interrupt siginterrupt(3) BSD
 adjtime: correct the time to allow synchronization of the/ adjtime(2) PRM
 scandir, alter priority of running scandir(3) BSD
 processes. renice: alter priority of running renice(1M) BSD
 sigaltstack: set or get signal alternate stack context. sigaltstack(2) PRM
 data that corresponds to an. /receives solicited mps_AMPreceive(D3D) DDRM
 fsba: file system block analyzer. fsba(1M) SARM
 /get a string of characters (and attributes) from a curses/ curs_inchstr(3X) CHAR
 /add string of characters (and attributes) to a curses/ curs_addchstr(3X) CHAR
 sigstack: set and/or get signal stack context. sigstack(3) BSD
 sort: sort and/or merge files. sort(1) URM
 application specific alarms and/or the "working"/ /display indicator(1F) CHAR
 files. ckperms: set and/or verify permissions on ckperms(1M) MBRMAN
 from one set of volumes to another. /move an archive migration(1M) SARM
 verify and return a string answer. /display a prompt; ckstr(1) SARM
 verify and return a string answer. /display a prompt; ckstr(1) SS
 pkgask: stores answers to a request script. pkgask(1M) SARM
 pkgask: stores answers to a request script. pkgask(1M) SS
 data that is not part of any. /sends solicited mps_AMPsend_data(D3D) DDRM
 Linking Format) files. a.out: ELF (Executable and a.out(4) PRM
 /field_just: format the general appearance of forms. form_field_just(3X) CHAR
 with the controlling FMLI application. /a co-process vsig(1F) CHAR
 panel. /panel_userptr: associate application data with a panels panel_userptr(3X) CHAR
 /field_userptr: associate application data with forms. form_field_userptr(3X) CHAR
 /form_userptr: associate application data with forms. form_userptr(3X) CHAR
 items. /item_userptr: associate application data with menus menu_item_userptr(3X) CHAR
 /menu_userptr: associate application data with menus. menu_userptr(3X) CHAR
 /a list of severity levels for an application for use with ffmtmsg. addseverity(3C) PRM
 /to maintenance commands and application programs. intro(1M) SARM
 introduction to commands and application programs. intro: intro(1) URM
 and/or the/ indicator: display application specific alarms indicator(1F) CHAR
 /coordinate ELF library and application versions. elf_version(3E) PRM
 for/ /field_term: assign application-specific routines form_hook(3X) CHAR
 /set_menu_term, menu_term: assign application-specific routines/ menu_hook(3X) CHAR
 keyword lookup. apropos: locate commands by apropos(1) BSD
 ar: archive file format. ar(4) PRM
 ar: archive file format. ar(4) SARM
 library. ar: maintain portable archive or ar(1) PRM
 library. ar: maintain portable archive or ar(1) URM
 language. bc: arbitrary-precision arithmetic bc(1) URM
 of the current host. arch: display the architecture arch(1) BSD

Permuted Index

.ott: FACE object	architecture information.	ott(4) SARM
host. arch: display the	architecture of the current	arch(1) BSD
an incremental filesystem	archive. /create, restore	infile(1M) SARM
restore from, a full file system	archive. fdp: create, or	fdp(1M) SARM
restore from, a full file system	archive. ffile: create, or	ffile(1M) SARM
ar:	archive file format.	ar(4) PRM
ar:	archive file format.	ar(4) SARM
convert: convert	archive files to common formats.	convert(1) PRM
to another. migration: move an	archive from one set of volumes	migration(1M) SARM
elf_next: sequential	archive member access.	elf_next(3E) PRM
elf_rand: random	archive member access.	elf_rand(3E) PRM
elf_getarhdr: retrieve	archive member header.	elf_getarhdr(3E) PRM
fimage: create, restore an image	archive of a filesystem.	fimage(1M) SARM
ar: maintain portable	archive or library.	ar(1) PRM
ar: maintain portable	archive or library.	ar(1) URM
elf_getarsym: retrieve	archive symbol table.	elf_getarsym(3E) PRM
tar: tape file	archiver.	tar(1) URM
cpio: copy file	archives: device header file.	archives(4) SARM
cpout: copy out file	archives in and out.	cpio(1) URM
stdarg: handle variable	archives to multiple volumes.	cpout(1M) MBRMAN
varargs: handle variable	argument list.	stdarg(5) PRM
formatted output of a variable	argument list.	varargs(5) PRM
command. xargs: construct	argument list. /vsprintf: print	vprintf(3S) PRM
getopt: get option letter from	argument list(s) and execute	xargs(1) URM
echo: echo	argument vector.	getopt(3C) PRM
echo: echo	arguments.	echo(1) BSD
expr: evaluate	arguments.	echo(1) URM
fmlexpr: evaluate	arguments as an expression.	expr(1) URM
message: puts its	arguments as an expression.	fmlexpr(1F) CHAR
miscellaneous functions for IEEE	arguments on FMLI message line.	message(1F) CHAR
bc: arbitrary-precision	arithmetic. /copysign, scalbn:	ieee_functions(3M) BSD
multiple precision integer	arithmetic language.	bc(1) URM
Protocol.	arithmetic. /xtom, mtox, mfree:	mp(3X) BSD
Protocol.	ARP: Address Resolution	ARP(7) NUAG
and control.	ARP: Address Resolution	ARP(7) SARM
and control.	arp: address resolution display	arp(1M) NUAG
notify: notify user of the	arp: address resolution display	arp(1M) SARM
arrival of new mail.	arrival of new mail.	notify(1) URM
as: assembler.	as: assembler.	as(1) PRM
ascftime: convert date and time	ascftime: convert date and time	strftime(3C) PRM
ASCII character set.	ASCII character set.	ascii(5) PRM
ASCII character set.	ASCII character set.	ascii(5) SARM
ascii: map of ASCII character	ascii: map of ASCII character	ascii(5) PRM
ascii: map of ASCII character	ascii: map of ASCII character	ascii(5) SARM
ASCII representation. /encode	ASCII representation. /encode	uuencode(1C) URM
ASCII string. /l64a: convert	ASCII string. /l64a: convert	a64l(3C) PRM
asctime, tzset: convert date and	asctime, tzset: convert date and	ctime(3C) PRM

/sin, sinf, cos, cosf, tan, tanf, asin, asinf, acos, acosf, atan, atanf, /
 /cos, cosf, tan, tanf, asin, asinh, cosh, coshf, tanh, tanhf, numbers or SCCS commands. help: as: assertion. assert: verify program /set_field_term, field_term: /set_menu_term, menu_term: /setbuffer, setlinebuf, setvbuf: setbuf, setvbuf: setbuffer, setlinebuf: forms. /field_userptr: /set_form_userptr, form_userptr: /set_item_userptr, item_userptr: /set_menu_userptr, menu_userptr: a panels panel. /panel_userptr: /write or erase forms from /write or erase menus from /forms window and subwindow /menus window and subwindow interface STREAMS/ ports: 5 line mb2a_getinfo: Gets specific /Sends a broadcast message in a transactionless message in mb2a_closeport: Closes an mb2a_openport: Opens an asy: script device driver. atcs: driver. iasy: /Sends a cancel message /Sends a reply message later time. /tanf, asin, asinf, acos, acosf, /asinf, acos, acosf, atan, atanf, /acos, acosf, atan, atanf, atan2, asin, asinf, acos, acosf, atan, /tanh, tanhf, asinh, acosh, controller script device/ routine. double-precision/ strtod, strtol, strtoul, atol, integer. strtol, strtoul, run at specified times. or batch. asin, asinf, acos, acosf, atan, / asinf, acos, acosf, atan, atanf, / asinh, acosh, atanh: hyperbolic/ ask for help with message help(1) PRM assembler. as(1) PRM assert: verify program assert(3X) PRM assertion. assert(3X) PRM assign application-specific/ form_hook(3X) CHAR assign application-specific/ menu_hook(3X) CHAR assign buffering to a stream. setbuf(3S) BSD assign buffering to a stream. setbuffer(3S) PRM assign buffering to a stream. setbuffer(3S) BSD associate application data with form_field_userptr(3X) CHAR associate application data with/ form_userptr(3X) CHAR associate application data with/ menu_item_userptr(3X) CHAR associate application data with/ menu_userptr(3X) CHAR associate application data with panel_userptr(3X) CHAR associated subwindows. form_post(3X) CHAR associated subwindows. menu_post(3X) CHAR association routines. form_win(3X) CHAR association routines. menu_win(3X) CHAR asy: asynchronous serial port. asy(7) SARM asynchronous communications ports(7) STRM asynchronous endpoint/ mb2a_getinfo(3tai) TAI asynchronous mode.. mb2a_brdcst(3tai) TAI asynchronous mode. /Sends mb2a_send(3tai) TAI asynchronous MULTIBUS II/ mb2a_closeport(3tai) TAI asynchronous MULTIBUS II/ mb2a_openport(3tai) TAI asynchronous serial port. asy(7) SARM asynchronous terminal controller atcs(1M) MBRMAN asynchronous (terminal) device iasy(1M) MBRMAN asynchronously.. mb2a_sendcancel(3tai) TAI asynchronously.. mb2a_sendreply(3tai) TAI at, batch: execute commands at a at(1) URM atan, atanf, atan2, atan2f:/ trig(3M) PRM atan2, atan2f: trigonometric/ trig(3M) PRM atan2f: trigonometric functions. trig(3M) PRM atanf, atan2, atan2f:/ /tanf, trig(3M) PRM atanh: hyperbolic functions. sinh(3M) PRM atcs: asynchronous terminal atcs(1M) MBRMAN atexit: add program termination atexit(3C) PRM atof: convert string to strtod(3C) PRM atoi: convert string to integer. strtol(3C) PRM atol, atoi: convert string to strtol(3C) PRM atq: display the jobs queued to atq(1) URM atrm: remove jobs spooled by at atrm(1) URM

supporting Bus, serial, and descriptor to an/ `fattach`:
 segment. `sdget`, `sdfree`:
 resources. `rmntry`:
 resources. `rmntry`:
 loginlog: log of failed login
 tunable parameter. `idttune`:
 /curses character and window
 `devattr`: lists device
 /get a string of characters (and
 /mvwinch: get a character and its
 format the general display
 login password and password
 set and get forms field
 menu_pad: control menus display
 /wechochar: add a character (with
 /add string of characters (and
 `wattron`, `attrset`,/ `curs attr`:
 `curs attr`: `attroff`, `wattroff`,
 /wattroff, `attron`, `wattron`,
 secure_rpc: `authdes_seccreate`,
 `authdes_getucred`,/ `secure_rpc`:
 `authsys_create`,/ `rpc_clnt_auth`:
 side remote procedure call
 `rpc_clnt_auth`: `auth_destroy`,
 `auth_destroy`, `authnone_create`,
 /authnone_create, `authsys_create`,
 /routines for
 vacation: reply to mail
 systems. `automount`:
 systems. `automount`:
 autopush: configure lists of
 autopush: configure lists of
 incoming mail/ `vacation`:
 signals and wait for/ `sigpause`:
 NFS file systems.
 NFS file systems.
 automatically pushed STREAMS/
 automatically pushed STREAMS/
 a function when a buffer becomes
 `testb`: check for an
 call function when buffer is
 share: make local NFS resource
 share: make local resource
 remote systems. `dfshares`: list
 remote systems. `dfshares`: list
 /list of service grades that are
 AT&T. /mouse device driver `mouse(7)` MOUSE
 attach a STREAMS-based file `fattach(3C)` STRM
 attach and detach a shared data `sdget(2)` XNX
 attempt to mount queued remote `rmntry(1M)` NUAG
 attempt to mount queued remote `rmntry(1M)` SARM
 attempts. `loginlog(4)` SARM
 attempts to set value of a `idttune(1M)` SARM
 attribute control routines. `curs_attr(3X)` CHAR
 attributes. `devattr(1M)` SARM
 (attributes) from a curses/ `curs_inchstr(3X)` CHAR
 attributes from a curses window. `curs_inch(3X)` CHAR
 attributes of forms. /field_pad: `form_field_attributes(3X)` CHAR
 attributes. `passwd`: change `passwd(1)` URM
 attributes. /set_max_field: `form_field_buffer(3X)` CHAR
 attributes. /set_menu_pad, `menu_attributes(3X)` CHAR
 (attributes) to a curses window/ `curs_addch(3X)` CHAR
 (attributes) to a curses window. `curs_addchstr(3X)` CHAR
 `attroff`, `wattroff`, `attron`, `curs_attr(3X)` CHAR
 `attron`, `wattron`, `attrset`,/ `curs_attr(3X)` CHAR
 `attrset`, `wattrset`, `standend`,/ `curs_attr(3X)` CHAR
 `authdes_getucred`, `getnetname`,/ `secure_rpc(3N)` NI
 `authdes_seccreate`, `secure_rpc(3N)` NI
 `auth_destroy`, `authnone_create`, `rpc_clnt_auth(3N)` NI
 authentication. /for client `rpc_clnt_auth(3N)` NI
 `authnone_create`, `authsys_create`,/ `rpc_clnt_auth(3N)` NI
 `authsys_create`,/ `rpc_clnt_auth`: `rpc_clnt_auth(3N)` NI
 `authsys_create_default`: library/ `rpc_clnt_auth(3N)` NI
 automatic invocation by menus. `menu_hook(3X)` CHAR
 automatically. `vacation(1)` BSD
 automatically mount NFS file `automount(1M)` NUAG
 automatically mount NFS file `automount(1M)` SARM
 automatically pushed STREAMS/ `autopush(1M)` SARM
 automatically pushed STREAMS/ `autopush(1M)` STRM
 automatically respond to `vacation(1)` URM
 automatically release blocked `sigpause(3)` BSD
 `automount`: automatically mount `automount(1M)` NUAG
 `automount`: automatically mount `automount(1M)` SARM
 `autopush`: configure lists of `autopush(1M)` SARM
 `autopush`: configure lists of `autopush(1M)` STRM
 available. `bufcall`: call `bufcall(D3DK)` DDRM
 available buffer. `testb(D3DK)` DDRM
 available. `esbcall`: `esbcall(D3DK)` DDRM
 available for mounting by remote/ `share(1M)` NUAG
 available for mounting by remote/ `share(1M)` SARM
 available NFS resources from `dfshares(1M)` NUAG
 available NFS resources from `dfshares(1M)` SARM
 available on this UNIX system. `uuglist(1C)` URM

or local/ dfshares: list
 or local/ dfshares: list
 remote systems. dfshares: list
 remote systems. dfshares: list
 resource/ waitsem, nbwaitsem:
 wait:
 processing language.
 /mvwgetch, ungetch: get (or push
 /bkgd, wbkgd: curses window
 behind the current queue.
 backup: perform
 system backup session.
 bkhistory: report on completed
 bkstatus: display the status of
 media/ bkoper: interact with
 functions.
 or display the contents of a
 ckbupscd: check file system
 initiate or control a system
 control in specified priority
 for a specified priority
 information about a queue or
 get information about a queue or

 hosts: host name data
 hosts: host name data
 hosts: host name data
 networks: network name data
 networks: network name data
 protocols: protocol name data
 protocols: protocol name data
 rpc: rpc program number data
 ypcat: print values in a YP data
 newaliases: rebuild the data
 text string from a message data
 elf_getbase: get the
 signal:
 signal:
 dbm_open, dbm_store: data
 delete, firstkey, nextkey: data
 terminal capability data
 terminal capability data
 convert between long integer and
 forms: character
 menus: character

 available resources from remote dfshares(1M) NUAG
 available resources from remote dfshares(1M) SARM
 available RFS resources from dfshares(1M) NUAG
 available RFS resources from dfshares(1M) SARM
 await and check access to a waitsem(2) XNX
 await completion of process. wait(1) URM
 awk: pattern scanning and awk(1) URM
 back characters from curses/ curs_getch(3X) CHAR
 background manipulation/ curs_bkgd(3X) CHAR
 backq: get pointer to the queue backq(D3DK) DDRM
 backup functions. backup(1) XNX
 backup: initiate or control a backup(1M) SARM
 backup operations. bkhistory(1M) SARM
 backup operations. bkstatus(1M) SARM
 backup operations to service bkoper(1M) SARM
 backup: perform backup backup(1) XNX
 backup register. bkreg: change bkreg(1M) SARM
 backup schedule. ckbupscd(1M) SARM
 backup session. backup: backup(1M) SARM
 band. bcanput: test for flow bcanput(D3DK) DDRM
 band. flushband: flush messages flushband(D3DK) DDRM
 band of the queue. /change bkqset(D3DK) DDRM
 band of the queue. strqget: strqget(D3DK) DDRM
 banner: make posters. banner(1) URM
 base. hosts(4) NI
 base. hosts(4) NUAG
 base. hosts(4) SARM
 base. networks(4) NUAG
 base. networks(4) SARM
 base. protocols(4) NUAG
 base. protocols(4) SARM
 base. rpc(4) NI
 base. ypcat(1) NI
 base for the mail aliases file. newaliases(1M) BSD
 base. gettxt: retrieve a gettxt(1) URM
 base offset for an object file. elf_getbase(3E) PRM
 base signals. signal(5) PRM
 base signals. signal(5) SARM
 base subroutines. /dbm_nextkey, ndbm(3) BSD
 base subroutines. /fetch, store, dbm(3X) BSD
 base. terminfo: terminfo(4) CHAR
 base. terminfo: terminfo(4) SARM
 base-64 ASCII string. /l64a: a64l(3C) PRM
 based forms package. forms(3X) CHAR
 based menus package. menus(3X) CHAR

getdev: lists devices
 (visual) display editor
 panels: character
 portions of path names.
 pathnames.
 element of a path name.
 a text string in, message data
 remove jobs spooled by at or
 later time. at,
 has il,/ curs termattr:
 arithmetic language.
 in specified priority band.
 procedures. brc,
 string/ bstring: bcopy,
 byte string/ bstring:
 locations in the kernel.

 cb: C program
 user. su:
 user. su:
 call a function when a buffer
 screen flash/ curs beep:
 has off-screen data ahead or
 backq: get pointer to the queue
 curs beep: beep, flash: curses
 bessel: j0, j1, jn, y0, y1, yn:
 Bessel functions.
 /srandom, initstate, setstate:
 brelse: return buffer to the

 systems. fsck
 system. mkfs
 fs (bfs): format of the
 fsck (bfs): check and repair
 mount (bfs): mount
 inode
 system volume. fs
 inode (bfs): format of a
 mount
 delimiter.
 gigiplot,/ plot, aedplot,
 addbib: create or extend a
 /create an inverted index to a
 lookbib: find references in a
 roffbib: format and print a
 sortbib: sort a
 and insert references from a
 based on criteria. getdev(1M) SARM
 based on ex. /screen-oriented vi(1) URM
 based panels package. panels(3X) CHAR
 basename, dirname: deliver basename(1) URM
 basename: display portions of basename(1) BSD
 basename: return the last basename(3G) PRM
 bases. /of, or search for srchtxt(1) URM
 batch. atrm: atrm(1) URM
 batch: execute commands at a at(1) URM
 baudrate, erasechar, has_ic, curs_termattr(3X) CHAR
 bc: arbitrary-precision bc(1) URM
 bcanput: test for flow control bcanput(D3DK) DDRM
 bcheckrc: system initialization brc(1M) SARM
 bcmp, bzero, ffs: bit and byte bstring(3) BSD
 bcopy, bcmp, bzero, ffs: bit and bstring(3) BSD
 bcopy: copy data between address bcopy(D3DK) DDRM
 bdiff: big diff. bdiff(1) URM
 beautifier. cb(1) PRM
 become super-user or another su(1M) SARM
 become super-user or another su(1M) URM
 becomes available. bufcall: bufcall(D3DK) DDRM
 beep, flash: curses bell and curs_beep(3X) CHAR
 behind. /tell if forms field form_data(3X) CHAR
 behind the current queue. backq(D3DK) DDRM
 bell and screen flash routines. curs_beep(3X) CHAR
 Bessel functions. bessel(3M) PRM
 bessel: j0, j1, jn, y0, y1, yn: bessel(3M) PRM
 better random number generator;/ random(3) BSD
 bfreelist. brelse(D3DK) DDRM
 bfs: big file scanner. bfs(1) URM
 (bfs): check and repair bfs file fsck(1M) SARM
 (bfs): construct a boot file mkfs(1M) SARM
 bfs file system volume. fs(4) SARM
 bfs file systems. fsck(1M) SARM
 bfs file systems. mount(1M) SARM
 (bfs): format of a bfs i-node. inode(4) SARM
 (bfs): format of the bfs file fs(4) SARM
 bfs i-node. inode(4) SARM
 (bfs): mount bfs file systems. mount(1M) SARM
 bgets: read stream up to next bgets(3G) PRM
 bgplot, crtplot, dumbplot, plot(1G) BSD
 bibliographic database. addbib(1) BSD
 bibliographic database. indxbib(1) BSD
 bibliographic database. lookbib(1) BSD
 bibliographic database. roffbib(1) BSD
 bibliographic database. sortbib(1) BSD
 bibliographic database. /expand refer(1) BSD

mail messages. **biff**: give notice of incoming **biff(1)** BSD
comsat, **in.comsat**: **biff** server. **comsat(1M)** NUAG
comsat, **in.comsat**: **biff** server. **comsat(1M)** SARM
bdiff: **big** diff. **bdiff(1)** URM
bfs: **big** file scanner. **bfs(1)** URM
information for the **ckbinarsys/** **binarsys**: remote system **binarsys(4)** SARM
uuencode, **uudecode**: encode a **binary** file, or decode its ASCII/ **uuencode(1C)** URM
fread, **fwrite**: **binary** input/output. **fread(3S)** PRM
whether remote system can accept **binary** messages. /determine **ckbinarsys(1M)** SARM
bsearch: **binary** search a sorted table. **bsearch(3C)** PRM
tfind, **tdelete**, **twalk**: manage **binary** search trees. **tsearch**, **tsearch(3C)** PRM
create an instance of a **binary** semaphore. **creatsem**: **creatsem(2)** XNX
strings in an object file or **binary**. strings: find printable **strings(1)** URM
bind: bind a name to a socket. **bind(3N)** NI
endpoint. **t_bind**: bind an address to a transport **t_bind(3N)** NI
bind: bind a name to a socket. **bind(3N)** NI
library routines for RPC **bind** service. /**rpcb_unset**: **rpcbind(3N)** NI
ypserv, **ypbind**: YP server and binder processes. **ypserv(1M)** NI
line on the/ **ccbind**: binds a line discipline to a **ccbind(1M)** MBRMAN
biod: NFS daemon. **biod(1M)** NUAG
biod: NFS daemon. **biod(1M)** SARM
block I/O and wakeup processes. **biodone**: release buffer after **biodone(D3DK)** DDRM
pending completion of block/ **biowait**: suspend processes **biowait(D3DK)** DDRM
ffs: find first set **bit**. **ffs(3C)** PRM
/bcopy, **bcmp**, **bzero**, **ffs**: **bit** and byte string operations. **bstring(3)** BSD
I/O port. **inw**: read a 16-bit short word from a 16-bit **inw(D3D)** DDRM
exception list. **bkexcept**: change or display an **bkexcept(1M)** SARM
cursor_bkgd: **bkgdset**, **wbkgdset**, **bkgd**, **wbkgd**: **cursor_bkgd(3X)** CHAR
cursor_bkgd: **cursor_bkgd(3X)** CHAR
cursor_bkgd: **cursor_bkgd(3X)** CHAR
bkhistory: report on completed **bkhistory(1M)** SARM
operations to service media/ **bkoper**: interact with backup **bkoper(1M)** SARM
contents of a backup register. **bkreg**: change or display the **bkreg(1M)** SARM
backup operations. **bkstatus**: display the status of **bkstatus(1M)** SARM
allocb: allocate a message **block**. **allocb(D3DK)** DDRM
copyb: copy a message **block**. **copyb(D3DK)** DDRM
freeb: free a message **block**. **freeb(D3DK)** DDRM
sync: update the super **block**. **sync(1M)** SARM
sync: update the super **block**. **sync(1M)** URM
sync: update the super **block**. **sync(2)** PRM
fsba: file system **block** analyzer. **fsba(1M)** SARM
sum: print checksum and **block** count of a file. **sum(1)** URM
dupb: duplicate a message **block** descriptor. **dupb(D3DK)** DDRM
prtvtoc: print the VTOC of a **block** device. **prtvtoc(1M)** SARM
Free storage for a DMA Command **Block**. **dma_free_cb**: **dma_free_cb(D3D)** DDRM
storage for a DMA Command **Block**. **dma_get_cb**: Allocate **dma_get_cb(D3D)** DDRM
sdi_getblk: allocate a SCSI **block** for the target driver. **sdi_getblk(D3I)** SCSI
rmvb: remove a message **block** from a message. **rmvb(D3DK)** DDRM

unlinkb: remove a message strategy: perform	block from the head of a/	unlinkb(D3DK) DDRM
biodone: release buffer after processes pending completion of structure. buf: a previously allocated SCSI sigblock, sigmask: msgb: STREAMS message esballoc: allocate a message spl: /examine signals that are sigpouse: automatically release linkb: concatenate two message report number of free disk report number of free disk /s5: report number of free disk freemsg: free all message du: display the number of disk wd: Western Digital 8003 Adapter interconnect registers of the interconnect registers of the the interconnect register of the the interconnect register of the the specified register of the slot. /checks for certain	block I/O. block I/O and wakeup processes. block I/O. biowait: suspend block I/O data transfer block. sdi_freeblk: release block signals. block structure. block using a shared buffer. block/allow interrupts. blocked and pending. blocked signals and wait for/ blocks. blocks and files. df (generic): blocks and files. df (generic): blocks and i-nodes for s5 file/ blocks in a message. blocks used per directory or/ Board. board and returns the. /reads board. ics_write: writes into board in the. ics_read: reads board in the. /_rec: reads board in. /writes a value into board types in the designated boot. boot file system. boot parameter server. boot parameter server. boot program. boot service daemon. boot: UNIX system boot program. bootparamd: boot parameter bootparamd: boot parameter bootserver: starts a MULTIBUS II server. server. boot service daemon. sgib: install MULTIBUS driver. bps: the/ initbp: initializes the wvline: create/ curs_border: /whline, wvline: create curs panel_top: top_panel, curs_border: border, wborder, address space. address space. driver. initialization procedures. manageable units. dma_pageio:	strategy(D2DK) DDRM biodone(D3DK) DDRM biowait(D3DK) DDRM buf(D4DK) DDRM sdi_freeblk(D3I) SCSI sigblock(3) BSD msgb(D4DK) DDRM esballoc(D3DK) DDRM spl(D3D) DDRM sigpending(2) PRM sigpouse(3) BSD linkb(D3DK) DDRM df(1M) SARM df(1M) URM df(1M) SARM freemsg(D3DK) DDRM du(1M) BSD wd(7) SARM ics_read(1M) MBRMAN ics_write(1M) MBRMAN ics_read(D3D) DDRM ics_find_rec(D3D) DDRM ics_write(D3D) DDRM ics_agent_cmp(D3D) DDRM boot(4) SARM mkfs(1M) SARM bootparamd(1M) NI bootparamd(1M) NUAG boot(1M) SARM bootserver(1M) MBRMAN boot(1M) SARM bootparamd(1M) NI bootparamd(1M) NUAG bootserver(1M) MBRMAN sgib(1M) MBRMAN bps(1M) MBRMAN initbp(1M) MBRMAN curs_border(3X) CHAR curs_border(3X) CHAR panel_top(3X) CHAR curs_border(3X) CHAR bp_mapin: allocate virtual bp_mapout: deallocate virtual bps: Bootstrap Parameter String brc: bcheckrc: system brc(1M) SARM dma_pageio(D3DK) DDRM

bfreelist.
 space allocation.
 mb2a_brdcst: Sends a
 mode.. mb2s_brdcst: Sends a
 mps_mk_brdcst: constructs a
 file. more, page:
 table.
 ffs: bit and byte string/
 size in pages (round down).
 size in pages (round up).
 structure.
 buffer becomes available.
 clrbuf: erase the contents of a
 testb: check for an available
 wakeup/ biodone: release
 pool. mps_free_msgbuf: puts a

 bufcall: call a function when a
 a buffer reject in response to a
 user data from the message
 user data from the message
 from a user program to a driver
 /Free storage for a DMA
 Allocate storage for a DMA
 /frees a list of data
 a pointer to a list of data
 a message block using a shared
 mps_mk_bgrant: construct a
 freerbuf: free a raw
 getrbuf: get a raw
 bufsplit: split
 esbcall: call function when
 determine whether a character
 set and get menus pattern match
 allocates a message
 mps_mk_breject: construct a
 read bytes from I/O port to
 32-bit words from I/O port to
 16-bit words from I/O port to
 a buffer grant in response to a
 repoutsb: write bytes from
 /write 32-bit words from
 /write 16-bit words from
 brelse: return
 stdio: standard
 /setlinebuf, setvbuf: assign
 setbuf, setvbuf: assign

 brelse: return buffer to the brelse(D3DK) DDRM
 brk, sbrk: change data segment brk(2) PRM
 broadcast message in/ mb2a_brdcst(3tai) TAI
 broadcast message in synchronous mb2s_brdcst(3tai) TAI
 broadcast message to be sent.. mps_mk_brdcst(D3D) DDRM
 browse or page through a text more(1) URM
 bsearch: binary search a sorted bsearch(3C) PRM
 bstring: bcopy, bcmp, bzero, bstring(3) BSD
 btop: convert size in bytes to btop(D3DK) DDRM
 btopr: convert size in bytes to btopr(D3DK) DDRM
 buf: block I/O data transfer buf(D4DK) DDRM
 bufcall: call a function when a bufcall(D3DK) DDRM
 buffer. cirbuf(D3DK) DDRM
 buffer. testb(D3DK) DDRM
 buffer after block I/O and biodone(D3DK) DDRM
 buffer back into the free memory
 mps_free_msgbuf(D3D) DDRM
 buffer becomes available. bufcall(D3DK) DDRM
 buffer. /construct mps_mk_breject(D3D) DDRM
 buffer.. /copies mps_get_soldata(D3D) DDRM
 buffer.. /copies mps_get_unsoldata(D3D) DDRM
 buffer. copyin: copy data copyin(D3DK) DDRM
 Buffer Descriptor. dma_get_buf(D3D) DDRM
 Buffer Descriptor. dma_get_buf: dma_get_buf(D3D) DDRM
 buffer descriptors.. mps_free_dmabuf(D3D) DDRM
 buffer descriptors.. /returns mps_get_dmabuf(D3D) DDRM
 buffer. esballoc: allocate esballoc(D3DK) DDRM
 buffer grant in response to a/ mps_mk_bgrant(D3D) DDRM
 buffer header. freerbuf(D3DK) DDRM
 buffer header. getrbuf(D3DK) DDRM
 buffer into fields. bufsplit(3G) PRM
 buffer is available. esbcall(D3DK) DDRM
 buffer is encrypted. isencrypt: isencrypt(3G) PRM
 buffer. /menu_pattern: menu_pattern(3X) CHAR
 buffer.. mps_get_msgbuf: mps_get_msgbuf(D3D) DDRM
 buffer reject in response to a/ mps_mk_breject(D3D) DDRM
 buffer. repinsb: repinsb(D3D) DDRM
 buffer. repinsd: read repinsd(D3D) DDRM
 buffer. repinsw: read repinsw(D3D) DDRM
 buffer request.. /construct mps_mk_bgrant(D3D) DDRM
 buffer to an I/O port. repoutsb(D3D) DDRM
 buffer to an I/O port. repoutsd(D3D) DDRM
 buffer to an I/O port. repoutsw(D3D) DDRM
 buffer to the bfreelist. brelse(D3DK) DDRM
 buffered input/output package. stdio(3S) PRM
 buffering to a stream. setbuf(3S) BSD
 buffering to a stream. setbuf(3S) PRM

Permuted Index

setbuffer, setlinebuf: assign fields. `setbuffer(3S)` BSD
 for an application/ addseverity: `bufsplit(3G)` PRM
 return a menu item. `ckitem(1)` SARM
 return a menu item. `ckitem(1)` SS
 `ypinit(1M)` NI
 `idbuild(1M)` SARM
 `idmkunix(1M)` SARM
 `disk(7)` SARM
 mouse device driver supporting Bus, serial, and AT&T. `mouse(7)` MOUSE
 interval. `drv_usecwait(D3DK)` DDRM
 `elf_fill(3E)` PRM
 `inb(D3D)` DDRM
 values between host and network byte order. `/ntohs: convert` `byteorder(3N)` NI
 /bcopy, bcmp, bzero, ffs: bit and byte string operations. `bstring(3)` BSD
 `outb(D3D)` DDRM
 `byteorder, htonl, htons, ntohl, byteorder(3N)` NI
 `swab(3C)` PRM
 memory for a given number of bytes. `bzero(D3DK)` DDRM
 `adjmsg(D3DK)` DDRM
 `port. repoutsb(D3D)` DDRM
 `repinsb(D3D)` DDRM
 `msgdsize(D3DK)` DDRM
 `pullupmsg(D3DK)` DDRM
 `size(1)` PRM
 `ptob(D3DK)` DDRM
 `bytes to size in pages (round btop(D3DK)` DDRM
 `bytes to size in pages (round btopr(D3DK)` DDRM
 `bzero: clear memory for a given bzero(D3DK)` DDRM
 `bzero, ffs: bit and byte string/ bstring(3)` BSD
 `cc(1)` BSD
 `cc(1)` PRM
 `C flowgraph. cflow(1)` PRM
 `C program. cscope(1)` PRM
 `C program beautifier. cb(1)` PRM
 `C program checker. lint(1)` PRM
 `C program cross-reference. cxref(1)` PRM
 `C program debugger. ctrace(1)` PRM
 `C program source. /resolve unifdef(1)` BSD
 `cal: print calendar. cal(1)` URM
 `calculate a checksum for a file. sum(1)` BSD
 `dc: desk calculator. dc(1)` URM
 `cal: print calendar. cal(1)` URM
 `calendar: reminder service. calendar(1)` URM
 `calendar time. mktime(3C)` PRM
 `calendar times. /computes difftime(3C)` PRM
 `call. syscall(3)` BSD
 `and remove ifdef'ed lines from`
 `sum:`
 `dc: desk`
 `cal: print`
 `converts a tm structure to a`
 `the difference between two`
 `syscall: indirect system`

becomes available. `bufcall`: call a function when a buffer `bufcall(D3DK)` DDRM
 `cu`: call another UNIX system. `cu(1C)` URM
 for client side remote procedure call authentication. `/routines` `rpc_dnt_auth(3N)` NI
 for server side remote procedure call errors. `/library routines` `rpc_svc_err(3N)` NI
 available. `esbcall`: call function when buffer is `esbcall(D3DK)` DDRM
 data returned by stat system call. `stat`: `stat(4)` XNX
 data returned by stat system call. `stat`: `stat(5)` PRM
 data returned by stat system call. `stat`: `stat(5)` SARM
 previous timeout(D3DK) function call. `untimeout: cancel` `untimeout(D3DK)` DDRM
 memory/ `malloc`, `free`, `realloc`, `calloc`, `mallopt`, `mallinfo`: `malloc(3X)` PRM
 memory/ `malloc`, `free`, `realloc`, `calloc`, `memalign`, `valloc`: `malloc(3C)` PRM
 `intro`: introduction to system calls and error numbers. `intro(2)` PRM
 `truss`: trace system calls and signals. `truss(1)` URM
 for secure remote procedure calls. `/library routines` `secure_rpc(3N)` NI
 routines for remote procedure calls. `rpc: library` `rpc(3N)` NI
 library routines for client side calls. `/rpc broadcast`, `rpc_call`: `rpc_dnt_calls(3N)` NI
 signals to interrupt system calls. `siginterrupt: allow` `siginterrupt(3)` BSD
 routines for remote procedure calls. `/XDR library` `rpc_xdr(3N)` NI
 `unbufcall: cancel a pending call-when-buffer-available/` `unbufcall(D3DK)` DDRM
 `unbufcall`: cancel a pending/ `unbufcall(D3DK)` DDRM
 `mb2a_sendcancel`: Sends a cancel message asynchronously.. `mb2a_sendcancel(3tai)` TAI
 `mb2s_sendcancel`: Sends a cancel message in synchronous. `mb2s_sendcancel(3tai)` TAI
 function call. `untimeout: cancel previous timeout(D3DK)` `untimeout(D3DK)` DDRM
 request. `rumount: cancel queued remote resource` `rumount(1M)` NUAG
 request. `rumount: cancel queued remote resource` `rumount(1M)` SARM
 an LP print service. `lp`, cancel: send/cancel requests to `lp(1)` URM
 transaction. `mps_AMPcancel`: cancels an ongoing rsvp `mps_AMPcancel(D3D)` DDRM
 `/init_color`, `has_colors`, `can_change_color`, `color_content`,/ `curl_color(3X)` CHAR
 message queue. `canput: test for room in a` `canput(D3DK)` DDRM
 `terminfo`: terminal capability data base. `terminfo(4)` CHAR
 `terminfo`: terminal capability data base. `terminfo(4)` SARM
 description into a `termcap` `captainfo: convert a termcap` `captainfo(1M)` CHAR
 description into a `termcap` `captainfo: convert a termcap` `captainfo(1M)` SARM
 `pnch`: file format for card images. `pnch(4)` SARM
 interface. `qt`: QIC cartridge magnetic tape streamer `qt(7)` SARM
 text editor (variant of `ex` for casual users). `edit`: `edit(1)` URM
 files. `cat: concatenate and print` `cat(1)` URM
 `catman`: create the cat files for the manual. `catman(1M)` BSD
 `catclose: open/close a message catalogue`. `catopen`, `catopen(3C)` PRM
 generate a formatted message catalogue. `gencat`: `gencat(1)` URM
 catalogue. `catopen: catclose: open/close a message` `catopen(3C)` PRM
 catalogue. `catgets: read a program message`. `catgets(3C)` PRM
 the manual. `catman: create the cat files for` `catman(1M)` BSD
 message catalogue. `catopen, catclose: open /close a` `catopen(3C)` PRM
 `cb`: C program beautifier. `cb(1)` URM
 `halfdelay`,/ `curl_inopts: cbreak`, `nocbreak`, `echo`, `noecho`, `curl_inopts(3X)` CHAR
 `pow`, `powf`, `sqrt`,/ `exp`, `expf`, `cbrt`, `log`, `logf`, `log10`, `log10f`, `exp(3M)` PRM

gencc: create a front-end to the interface device driver.
 communications/ ccisrvinfo: to a subchannel on the/ to a line on the communication/ host with a subchannel on a/ program on the communications/ information on the/ on the communications/ the communications controller. program on to the/ information on the/ information on the/ on a line to a new host on the/ discipline from a line on the/ an SCCS delta.
 fmodf, fabs,/ floor, floorf, fabs,/ floor, floorf, ceil, /fabsf, rint, remainder: floor, ics_agent_cmp: checks for /tclflush, tclflow, cfgetospeed, /tcdrain, tclflush, tclflow, /cfgetospeed, cfgetispeed, /cfgetispeed, cfsetispeed, and time to string. strftime, flags. configs: allocation. brk, sbrk: chmod: chown: chown: or band of the queue. strqset: password attributes. passwd: chmod, fchmod: environment. putenv: list. bkexcept: of a backup register. bkgreg: sigprocmask: configuration. strchg, strconf: configuration. strchg, strconf: file. chown, lchown, fchown: time-sharing process. nice: nice: password. rfpasswd: cc: C compiler. cc(1) BSD cc: C compiler. cc(1) PRM cc command. gencc(1M) SARM cci: communications controller cci(1M) MBRMAN CCI server information on the ccisrvinfo(1M) MBRMAN cciattach: Creates a connection cciattach(1M) MBRMAN ccbind: binds a line discipline ccbind(1M) MBRMAN ccidetach: disassociates this ccidetach(1M) MBRMAN ccifree: frees a line discipline ccifree(1M) MBRMAN ccildinfo: line discipline ccildinfo(1M) MBRMAN ccildlist: line discipline list ccildlist(1M) MBRMAN ccilinfo: line information on ccilinfo(1M) MBRMAN cciload: load a line discipline cciload(1M) MBRMAN ccisrvinfo: CCI server ccisrvinfo(1M) MBRMAN ccisubinfo: subchannel ccisubinfo(1M) MBRMAN cciswitch: switch a subchannel cciswitch(1M) MBRMAN cciunbind: unbinds a line cciunbind(1M) MBRMAN cd: change working directory. cd(1) URM cdc: change the delta comment of cdc(1) PRM ceil, ceilf, copysign, fmod, floor(3M) PRM ceilf, copysign, fmod, fmodf, floor(3M) PRM ceiling, remainder, absolute/ floor(3M) PRM certain board types in the/ ics_agent_cmp(D3D) DDRM cfgetispeed, cfsetispeed,/ termios(2) PRM cfgetospeed, cfgetispeed,/ termios(2) PRM cflow: generate C flowgraph. cflow(1) PRM cfsetispeed, cfsetospeed,/ termios(2) PRM cfsetospeed, tcgetpgrp,/ termios(2) PRM cftime, ascftime: convert date strftime(3C) PRM change and display console configs(1M) SARM change data segment space brk(2) PRM change file mode. chmod(1) URM change file owner. chown(1) BSD change file owner. chown(1) URM change information about a queue strqset(D3DK) DDRM change login password and passwd(1) URM change mode of file. chmod(2) PRM change or add value to putenv(3C) PRM change or display an exception bkexcept(1M) SARM change or display the contents bkgreg(1M) SARM change or examine signal mask. sigprocmask(2) PRM change or query stream strchg(1) STRM change or query stream strchg(1) URM change owner and group of a chown(2) PRM change priority of a nice(2) PRM change priority of a process. nice(3C) BSD change Remote File Sharing host rfpasswd(1M) NUAG

password. rpasswd: change Remote File Sharing host rpasswd(1M) SARM
 chroot: change root directory. chroot(2) PRM
 command. chroot: change root directory for a chroot(1M) SARM
 wait for child process to change state. waitid: waitid(2) PRM
 wait for child process to change state. waitpid: waitpid(2) PRM
 shutdown: shut down system, change system state. shutdown(1M) SARM
 modification dates of/ settime: change the access and settime(1) XNX
 SCCS delta. cdc: change the delta comment of an cdc(1) PRM
 file. newform: change the format of a text newform(1) URM
 file. chgrp: change the group ownership of a chgrp(1) URM
 rename: change the name of a file. rename(2) PRM
 chsize: change the size of a file. chsize(2) XNX
 delta: make a delta (change) to an SCCS file. delta(1) PRM
 chkey: change user encryption key. chkey(1) NI
 chkey: change user encryption key. chkey(1) NUAG
 cd: change working directory. cd(1) URM
 chdir, fchdir: change working directory. chdir(2) PRM
 yppush: force propagation of changed YP map. yppush(1M) NI
 setuname: changes machine information. setuname(1M) SARM
 yp_update: changes yp information. yupdate(3N) NI
 number generator; routines for changing generators. /random random(3) BSD
 ypupdated: server for changing YP information. ypupdated(1M) NI
 mps_open_chan: opens a channel. mps_open_chan(D3D) DDRM
 pipe: create an interprocess channel. pipe(2) PRM
 requests on a DMA Controller channel. dma_disable: Disable dma_disable(D3D) DDRM
 requests on a DMA Controller channel. dma_enable: Enable dma_enable(D3D) DDRM
 DMA activity and release the channel. dma_stop: stop dma_stop(D3D) DDRM
 dma_prog: Programming a Channel for a hardware request. dma_prog(D3D) DDRM
 dma_swsetup: Setting a channel for software request. dma_swsetup(D3D) DDRM
 closes a previously opened channel. mps_close_chan: mps_close_chan(D3D) DDRM
 driver. i354: iSBX 354 dual channel serial-port device i354(1M) MBRMAN
 driver. xtproto: multiplexed channels protocol used by xt xtproto(5) PRM
 driver. xtproto: multiplexed channels protocol used by xt xtproto(5) SARM
 /winch, mvinch, mvwinch: get a character and its attributes/ curs_inch(3X) CHAR
 /standout, wstandout: curses character and window attribute/ curs_attr(3X) CHAR
 stream. ungetc: push character back onto input ungetc(3S) PRM
 forms: character based forms package. forms(3X) CHAR
 menus: character based menus package. menus(3X) CHAR
 panels: character based panels package. panels(3X) CHAR
 /mvinsch, mvwinsch: insert a character before the character/ curs_insch(3X) CHAR
 isencrypt: determine whether a character buffer is encrypted. isencrypt(3G) PRM
 conversion/ chrtbl: generate character classification and chrtbl(1M) SARM
 eqnchar: special character definitions for eqn. eqnchar(7) BSD
 ioctl: control a character device. ioctl(D2DK) DDRM
 entry point for a non-STREAMS character driver. chpoll: poll chpoll(D2DK) DDRM
 uwritec: remove a character from a uio structure. uwritec(D3DK) DDRM
 isprint, isgraph, isascii: character handling. /ispunct, ctype(3C) PRM

mbtowc, mblen, wctomb: multibyte user. cuserid: get
 getc, getchar, fgetc, getw: get putc, putchar, fputc, putw: put
 ascii: map of ASCII character set. ascii(5) PRM
 fgrep: search a file for a character string. fgrep(1) URM
 fgrep: search a file for a character string. fgrep(1) XNX
 /mvwgetstr, wgetnstr: get character strings from curses/ curs_getstr(3X) CHAR
 ureadc: add character to a uio structure. ureadc(D3DK) DDRM
 /mvdelch, mvwdelch: delete character under cursor in a/ curs_delch(3X) CHAR
 /insert a character before the character under the cursor in a/ curs_insch(3X) CHAR
 /mvwinsnstr: insert string before character under the cursor in a/ curs_insstr(3X) CHAR
 /echochar, wechochar: add a character (with attributes) to a/ curs_addch(3X) CHAR
 /get forms field characteristics. form_field_info(3X) CHAR
 pkginfo: package characteristics file. pkginfo(4) SS
 establish or restore terminal characteristics. tset, reset: tset(1) BSD
 tr: translate characters. tr(1) BSD
 tr: translate characters. tr(1) URM
 a/ /mvwinchnstr: get a string of characters (and attributes) from curs_inchstr(3X) CHAR
 /mvwaddchnstr: add string of characters (and attributes) to a/ curs_addchstr(3X) CHAR
 /mvwinstr: get a string of characters from a curses window. curs_instr(3X) CHAR
 /ungetch: get (or push back) characters from curses terminal/ curs_getch(3X) CHAR
 and/ /mvwaddnstr: add a string of characters to a curses window curs_addstr(3X) CHAR
 _tolower, toascii: translate characters. /tolower, toupper, conv(3C) PRM
 lastlogin, monacct, nulladm,/ chargefee, ckpacct, dodisk, acctsh(1M) SARM
 directory. chdir, fchdir: change working chdir(2) PRM
 checkfsys: check a file system. checkfsys(1M) SARM
 waitsem, nwaitsem: await and check access to a resource/ waitsem(2) XNX
 pkgchk: check accuracy of installation. pkgchk(1M) SARM
 pkgchk: check accuracy of installation. pkgchk(1M) SS
 /(ufs): file system consistency check and interactive repair. fsck(1M) SARM
 systems. fsck (bfs): check and repair bfs file fsck(1M) SARM
 fsck: check and repair file systems. fsck(1M) SARM
 systems. fsck (s5): check and repair s5 file fsck(1M) SARM
 filesystems. xfsck: check and repair XENIX xfsck(1M) SARM
 schedule. ckbupscd: check file system backup ckbupscd(1M) SARM
 testb: check for an available buffer. testb(D3DK) DDRM
 grpck: check group database entries. grpck(1M) BSD
 files; report possible/ checknr: check nroff and troff input checknr(1) BSD
 pwck: check password database entries. pwck(1M) BSD
 spray: scatter data in order to check the network. spray(3N) NI
 permissions file. uucheck: check the uucp directories and uucheck(1M) SARM
 be read. rdchk: check to see if there is data to rdchk(2) XNX
 mapped device. mmap: check virtual mapping for memory mmap(D2K) DDRM
 eqn, neqn, checkeq: typeset mathematics. eqn(1) BSD
 lint: a C program checker. lint(1) PRM

file system quota consistency
 pwck, grpck: password/group file

reboot/halt the system without
 input files; report possible/
 in the/ ics_agent_cmp:
 file. sum: print
 sum: calculate a
 ownership of a file.
 times: get process and
 waitid: wait for
 waitpid: wait for
 terminate. wait: wait for
 key.
 key.
 file.

owner and group of a file.
 non-STREAMS character driver.
 for a command.

classification and conversion/
 file.
 system information for the
 remote system can accept binary/
 backup schedule.
 valdate: prompts for and/
 valdate: prompts for and/
 prompts for and validates a/
 prompts for and validates a/
 and return an integer value.
 and return an integer value.
 and return a menu item.
 and return a menu item.
 validates a keyword.
 validates a keyword.
 monacct, nulladm,/ chargefee,
 and return a pathname.
 and return a pathname.
 permissions on files.
 validates an integer.
 validates an integer.
 and return a string answer.
 and return a string answer.
 and return a time of day.

checker. quotacheck: quotacheck(1M) SARM
 checkers. pwck(1M) SARM
 checkfsys: check a file system. checkfsys(1M) SARM
 checking the disks. /fasthalt: fastboot(1M) BSD
 checknr: check nroff and troff checknr(1) BSD
 checks for certain board types ics_agent_cmp(D3D) DDRM
 checksum and block count of a sum(1) URM
 checksum for a file. sum(1) BSD
 chgrp: change the group chgrp(1) URM
 child process times. times(2) PRM
 child process to change state. waitid(2) PRM
 child process to change state. waitpid(2) PRM
 child process to stop or wait(2) PRM
 chkey: change user encryption chkey(1) NI
 chkey: change user encryption chkey(1) NUAG
 chmod: change file mode. chmod(1) URM
 chmod, fchmod: change mode of chmod(2) PRM
 chown: change file owner. chown(1) BSD
 chown: change file owner. chown(1) URM
 chown, lchown, fchown: change chown(2) PRM
 chpoll: poll entry point for a chpoll(D2DK) DDRM
 chroot: change root directory chroot(1M) SARM
 chroot: change root directory. chroot(2) PRM
 chrtbl: generate character chrtbl(1M) SARM
 chsize: change the size of a chsize(2) XNX
 ckbinarsys command. /remote binarsys(4) SARM
 ckbinarsys: determine whether ckbinarsys(1M) SARM
 ckbupscd: check file system ckbupscd(1M) SARM
 ckdate, errdate, helpdate, ckdate(1) SARM
 ckdate, errdate, helpdate, ckdate(1) SS
 ckgid, errgid, helpgid, valgid: ckgid(1) SARM
 ckgid, errgid, helpgid, valgid: ckgid(1) SS
 ckint: display a prompt; verify ckint(1) SARM
 ckint: display a prompt; verify ckint(1) SS
 ckitem: build a menu; prompt for ckitem(1) SARM
 ckitem: build a menu; prompt for ckitem(1) SS
 ckkeywd: prompts for and ckkeywd(1) SARM
 ckkeywd: prompts for and ckkeywd(1) SS
 ckpacct, dodisk, lastlogin, acctsh(1M) SARM
 ckpath: display a prompt; verify ckpath(1) SARM
 ckpath: display a prompt; verify ckpath(1) SS
 ckperms: set and/or verify ckperms(1M) MBRMAN
 ckrange: prompts for and ckrange(1) SARM
 ckrange: prompts for and ckrange(1) SS
 ckstr: display a prompt; verify ckstr(1) SARM
 ckstr: display a prompt; verify ckstr(1) SS
 cktime: display a prompt; verify cktime(1) SARM

and return a time of day.
 a user ID.
 a user ID.
 validates yes/no.
 validates yes/no.
 elf32_xlatetof, elf32_xlatetom:
 header. /elf32_newehdr: retrieve
 table. /elf32_newphdr: retrieve
 /elf32_getshdr: retrieve
 chrtrl: generate character
 uucleanup: uucp spool directory
 strclean: STREAMS error logger
 strclean: STREAMS error logger
 /wclrtobot, clrtoeol, wclrtoeol:
 screen.
 of bytes. bzero:
 kmem_zalloc: allocate and
 clear:
 curs_clear: erase, werase,
 inquiries. ferror, feof,
 leaveok, / curs_outopts:
 creation and manipulation of
 yperr_string, ypprot_err: YP
 /rpc_call: library routines for
 call/ /library routines for
 listener. nlsgetcall: get
 shell command interpreter with a
 clnt_geterr,/ rpc_clnt_calls:
 clnt_destroy,/ rpc_clnt_create:
 rpc_clnt_create: clnt_control,
 /clnt_control, clnt_create,
 clnt_create, clnt_destroy,
 rpc_clnt_calls: clnt_call,
 /clnt_call, clnt_freeres,
 clnt_destroy, clnt_dg_create,
 /clnt_freeres, clnt_geterr,
 /clnt_geterr, clnt_perrno,
 /clnt_pcreateerror,
 /clnt_raw_create,
 /clnt_perrno, clnt_perror,
 /clnt_perror, clnt_sperrno,
 /clnt_screateerror,
 library/ /clnt_tli_create,
 /clnt_tli_create, clnt_tp_create,
 alarm: set a process alarm
 cron:
 rtc: MULTIBUS
 cktime: display a prompt; verify cktime(1) SS
 ckuid: prompts for and validates ckuid(1) SARM
 ckuid: prompts for and validates ckuid(1) SS
 ckyorn: prompts for and ckyorn(1) SARM
 ckyorn: prompts for and ckyorn(1) SS
 class-dependent data/ elf_xlate: elf_xlate(3E) PRM
 class-dependent object file elf_getehdr(3E) PRM
 class-dependent program header elf_getphdr(3E) PRM
 class-dependent section header. elf_getshdr(3E) PRM
 classification and conversion/ chrtrl(1M) SARM
 clean-up. uucleanup(1M) SARM
 cleanup program. strclean(1M) SARM
 cleanup program. strclean(1M) STRM
 clear all or part of a curses/ curs_clear(3X) CHAR
 clear: clear the terminal clear(1) URM
 clear memory for a given number bzero(D3DK) DDRM
 clear space from kernel free/ kmem_zalloc(D3DK) DDRM
 clear the terminal screen. clear(1) URM
 clear, wclear, clrtoeol,/ curs_clear(3X) CHAR
 clearerr, fileno: stream status ferror(3S) PRM
 clearok, idlok, idcok immedok, curs_outopts(3X) CHAR
 CLIENT handles. /dealing with rpc_clnt_create(3N) NI
 client interface. /yp_master, ypcnt(3N) NI
 client side calls. rpc_clnt_calls(3N) NI
 client side remote procedure rpc_clnt_auth(3N) NI
 client's data passed via the nlsgetcall(3N) NI
 C-like syntax. csh: csh(1) URM
 clnt_call, clnt_freeres, rpc_clnt_calls(3N) NI
 clnt_control, clnt_create, rpc_clnt_create(3N) NI
 clnt_create, clnt_destroy,/ rpc_clnt_create(3N) NI
 clnt_destroy, clnt_dg_create,/ rpc_clnt_create(3N) NI
 clnt_dg_create,/ /clnt_control, rpc_clnt_create(3N) NI
 clnt_freeres, clnt_geterr,/ rpc_clnt_calls(3N) NI
 clnt_geterr, clnt_perrno,/ rpc_clnt_calls(3N) NI
 clnt_pcreateerror,/ /clnt_create, rpc_clnt_create(3N) NI
 clnt_perrno, clnt_perror,/ rpc_clnt_calls(3N) NI
 clnt_perror, clnt_sperrno,/ rpc_clnt_calls(3N) NI
 clnt_raw_create,/ rpc_clnt_create(3N) NI
 clnt_screateerror,/ rpc_clnt_create(3N) NI
 clnt_sperrno, clnt_sperror,/ rpc_clnt_calls(3N) NI
 clnt_sperror, rpc_broadcast,/ rpc_clnt_calls(3N) NI
 clnt_tli_create, clnt_tp_create,/ rpc_clnt_create(3N) NI
 clnt_tp_create, clnt_vc_create: rpc_clnt_create(3N) NI
 clnt_vc_create: library routines/ rpc_clnt_create(3N) NI
 clock. alarm(2) PRM
 clock daemon. cron(1M) SARM
 clock driver. rtc(1M) MBRMAN

rtc: real time
 set system time from hardware
 synchronization of the system
 convert microseconds to
 for a specified number of
 drv_hztousec: convert
 device pair on a STREAMS/
 close:
 dlclose:
 t_close:
 time. shutdown:
 fclose, fflush:
 command. p2open, p2close: open,
 device.
 /telldir, seekdir, rewinddir,
 system log. syslog, openlog,
 channel. mps_close_chan:
 transport. mb2s_closeport:
 II transport/ mb2a_closeport:
 buffer.
 /erase, werase, clear, wclear,
 or/ /wclear, clrtoobot, wclrtoobot,
 message or panic the system.
 cram-
 communicate/ ccreate, cosend,
 coreceive, codestroy:/
 handling for specific SIGFPE
 /cosend, cocheck, coreceive,
 translation.
 translation. cof2elf:
 strcoll: string
 colltbl: create
 mailstats: print statistics
 database.
 setcolor: redefine or create a
 /pair_content: curses
 /has_colors, can_change_color,
 get maximum numbers of rows and
 comb:
 common to two sorted files.
 rexec: return stream to a remote
 rexec: return stream to a remote
 clock interface. rtc(7) SARM
 clock: report CPU time used. clock(3C) PRM
 clock. setclk: setclk(1M) SARM
 clock. /the time to allow adjtime(2) PRM
 clock ticks. drv_usectohz: drv_usectohz(D3DK) DDRM
 clock ticks. /process execution delay(D3DK) DDRM
 clock ticks to microseconds. drv_hztousec(D3DK) DDRM
 clone: open any major/minor clone(7) STRM
 close a file descriptor. close(2) PRM
 close a shared object. dlclose(3X) PRM
 close a transport endpoint. t_close(3N) NI
 close: close a file descriptor. close(2) PRM
 close down the system at a given shutdown(1M) BSD
 close or flush a stream. fclose(3S) PRM
 close pipes to and from a p2open(3G) PRM
 close: relinquish access to a close(D2DK) DDRM
 closedir: directory operations. directory(3C) PRM
 closelog, setlogmask: control syslog(3) BSD
 closes a previously opened mps_close_chan(D3D) DDRM
 Closes a synchronous MULTIBUS II mb2s_closeport(3tai) TAI
 Closes an asynchronous MULTIBUS mb2a_closeport(3tai) TAI
 clrbuf: erase the contents of a clrbuf(D3DK) DDRM
 clrtoobot, wclrtoobot, clrtoeol,/ curs_clear(3X) CHAR
 clrtoeol, wclrtoeol: clear all curs_clear(3X) CHAR
 cmn_err: display an error cmn_err(D3DK) DDRM
 CMOS RAM interface. cram(7) SARM
 cmp: compare two files. cmp(1) URM
 cocheck, coreceive, codestroy: coproc(1F) CHAR
 ccreate, cosend, cocheck, coproc(1F) CHAR
 codes. sigfpe: signal sigfpe(3) BSD
 codestroy: communicate with a/ coproc(1F) CHAR
 cof2elf: COFF to ELF object file cof2elf(1) PRM
 COFF to ELF object file cof2elf(1) PRM
 col: filter reverse line-feeds. col(1) URM
 collation. strcoll(3C) PRM
 collation database. colltbl(1M) SARM
 collected by sendmail. mailstats(1M) BSD
 colltbl: create collation colltbl(1M) SARM
 color. setcolor(1F) CHAR
 color manipulation routines. curs_color(3X) CHAR
 color_content, pair_content:/ curs_color(3X) CHAR
 columns in menus. /set and menu_format(3X) CHAR
 comb: combine SCCS deltas. comb(1) PRM
 combine SCCS deltas. comb(1) PRM
 comm: select or reject lines comm(1) URM
 command. rexec(3N) NI
 command. rexec(3N) NUAG

Permuted Index

system: issue a shell
test: condition evaluation
test: condition evaluation
test: condition evaluation
time: time a
KornShell, a standard/restricted
nice: run a
Free storage for a DMA
Allocate storage for a DMA
change root directory for a
alias. which: locate a
/executable for the Framed Access
env: set environment for
uux: UNIX-to-UNIX system
mail_pipe: invoke recipient
create a front-end to the cc
quits. nohup: run a
job control, and restricted
C-like syntax. csh: shell
getopt: parse
getopts, getoptcv: parse
open, close pipes to and from a
subsystem. form_driver:
subsystem. menu_driver:
information for the ckbinarsys
system activity. timex: time a
uuxqt: execute remote
accounting records. acctcms:
sdi_send: send SCSI
shell: run a
argument list(s) and execute
install: install
install: install
streamio: STREAMS ioctl
xinstall: install
intro: introduction to
introduction to maintenance
at, batch: execute
apropos: locate
order. lastcomm: show the last
transport/ mailsurr: surrogate
dfstab: file containing
dfstab: file containing
with message numbers or SCCS
introduction to programming
multi-user/ rc2: run
reboot the operating/ rc6: run
command. system(3S) PRM
command. test(1) BSD
command. test(1F) CHAR
command. test(1) URM
command. time(1) URM
command and programming/ /rksh: ksh(1) URM
command at low priority. nice(1) URM
Command Block. dma_free_cb: dma_free_cb(D3D) DDRM
Command Block. dma_get_cb: dma_get_cb(D3D) DDRM
command. chroot: chroot(1M) SARM
command; display its pathname or which(1) BSD
Command Environment Interface. face(1) URM
command execution. env(1) URM
command execution. uux(1C) URM
command for incoming mail. mail_pipe(1M) SARM
command. gencc: gencc(1M) SARM
command immune to hangups and nohup(1) URM
command interpreter. /standard, sh(1) URM
command interpreter with a csh(1) URM
command options. getopt(1) URM
command options. getopts(1) URM
command. p2open, p2close: p2open(3G) PRM
command processor for the forms form_driver(3X) CHAR
command processor for the menus menu_driver(3X) CHAR
command. /remote system binarsys(4) SARM
command; report process data and timex(1) URM
command requests. uuxqt(1M) SARM
command summary from per-process acctcms(1M) SARM
command to the controller. sdi_send(D3I) SCSI
command using shell. shell(1F) CHAR
command. xargs: construct xargs(1) URM
commands. install(1M) PRM
commands. install(1M) SARM
commands. streamio(7) STRM
commands. xinstall(1M) XNX
commands and application/ intro(1) URM
commands and application/ intro: intro(1M) SARM
commands at a later time. at(1) URM
commands by keyword lookup. apropos(1) BSD
commands executed, in reverse lastcomm(1) BSD
commands for routing and mailsurr(4) SARM
commands for sharing resources. dfstab(4) NUAG
commands for sharing resources. dfstab(4) SARM
commands. help: ask for help help(1) PRM
commands. intro: intro(1) PRM
commands performed for rc2(1M) SARM
commands performed to stop and rc6(1M) SARM

operating system. rc0: run
 cdc: change the delta
 file. mcs: manipulate the
 prt: display the delta and
 convert archive files to
 filehdr: file header for
 comm: select or reject lines
 /cocheck, coreceive, codestroy:
 socket: create an endpoint for
 line discipline to a line on the
 ipcs: report inter-process
 ftok: standard interprocess
 to a subchannel on the
 interface device driver. cci:
 /a line discipline program on the
 /discipline information on the
 /line discipline list on the
 /line information on the
 /discipline program on to the
 /CCI server information on the
 /subchannel information on the
 /on a line to a new host on the
 /discipline from a line on the
 /a line to a new host on the
 i410: iSBC 186/410 serial
 ports: 5 line asynchronous
 users: display a
 diff: differential file
 descriptions. infocmp:
 descriptions. infocmp:
 cmp:
 file. scsdiff:
 diff3: 3-way differential file
 dircmp: directory
 V7, 4BSD and XENIX STREAMS
 compver:
 regcmp: regular expression
 expression. regcmp, regex:
 /advance: regular expression
 /advance: regular expression
 /advance: regular expression
 expression compile and/ regexp:
 expression compile and/ regexp:
 expression compile and/ regexpr:
 term: format of
 term: format of
 cc: C
 commands performed to stop the rc0(1M) SARM
 comment of an SCCS delta. cdc(1) PRM
 comment section of an object mcs(1) PRM
 commentary history of an SCCS/ prt(1) BSD
 common formats. convert: convert(1) PRM
 common object files. filehdr(4) SARM
 common to two sorted files. comm(1) URM
 communicate with a process. coproc(1F) CHAR
 communication. socket(3N) NI
 communication controller.. /a ccibind(1M) MBRMAN
 communication facilities status. ipcs(1) URM
 communication package. stdipc: stdipc(3C) PRM
 communications. /a connection cciattach(1M) MBRMAN
 communications controller cci(1M) MBRMAN
 communications controller.. ccifree(1M) MBRMAN
 communications controller.. ccildinfo(1M) MBRMAN
 communications controller.. ccildlist(1M) MBRMAN
 communications controller. ccilinfo(1M) MBRMAN
 communications controller.. cciload(1M) MBRMAN
 communications controller.. ccisrvinfo(1M) MBRMAN
 communications controller.. ccisubinfo(1M) MBRMAN
 communications controller.. cciswitch(1M) MBRMAN
 communications controller.. cciunbind(1M) MBRMAN
 communications controller. ttyswitch(1M) MBRMAN
 communications controller device/ i410(1M) MBRMAN
 communications interface STREAMS/ ports(7) STRM
 compact list of users logged in. users(1) BSD
 comparator. diff(1) URM
 compare or print out terminfo infocmp(1M) CHAR
 compare or print out terminfo infocmp(1M) SARM
 compare two files. cmp(1) URM
 compare two versions of an SCCS scsdiff(1) PRM
 comparison. diff3(1) URM
 comparison. dircmp(1) URM
 compatibility module. tcompat: tcompat(7) SARM
 compatible versions file. compver(4) SS
 compile. regcmp(1) PRM
 compile and execute regular regcmp(3G) PRM
 compile and match routines. regexp(5) PRM
 compile and match routines. regexp(5) SARM
 compile and match routines. regexp(3G) PRM
 compile, step, advance: regular regexp(5) PRM
 compile, step, advance: regular regexp(5) SARM
 compile, step, advance: regular regexpr(3G) PRM
 compiled term file. term(4) CHAR
 compiled term file. term(4) SARM
 compiler. cc(1) BSD

cc: C	compiler.	cc(1) PRM
rpcgen: an RPC protocol	compiler.	rpcgen(1) NI
tic: terminfo	compiler.	tic(1M) CHAR
tic: terminfo	compiler.	tic(1M) SARM
zic: time zone	compiler.	zic(1M) SARM
yacc: yet another	compiler-compiler.	yacc(1) PRM
erf, erf: error function and	complementary error function.	erf(3M) PRM
bkhistory: report on	completed backup operations.	bkhistory(1M) SARM
/suspend processes pending	completion of block I/O.	biowait(D3DK) DDRM
wait: await	completion of process.	wait(1) URM
entry corresponding to NETPATH	component. /get /etc/netconfig	getnetpath(3N) NI
pack, pcat, unpack:	compress and expand files.	pack(1) URM
compress, uncompress, zcat:	compress, expand or display/	compress(1) URM
compress, expand or display/	compress, uncompress, zcat:	compress(1) URM
strcadd, strccpy: copy strings,	compressing. strccpy: streadd,	strccpy(3G) PRM
elf_hash:	compute hash value.	elf_hash(3E) PRM
remainder. div, ldiv:	compute the quotient and	div(3C) PRM
two calendar times. difftime:	computes the difference between	difftime(3C) PRM
file.	compver: compatible versions	compver(4) SS
	comsat, in.comsat: biff server.	comsat(1M) NUAG
	comsat, in.comsat: biff server.	comsat(1M) SARM
	concatenate and print files.	cat(1) URM
cat:	concatenate bytes in a message.	pullupmsg(D3DK) DDRM
pullupmsg:	concatenate two message blocks.	linkb(D3DK) DDRM
linkb:	condition evaluation command.	test(1) BSD
test:	condition evaluation command.	test(1F) CHAR
test:	condition evaluation command.	test(1) URM
configuration file format.	config: MULTIBUS II	config(1M) MBRMAN
fpathconf, pathconf: get	configurable pathname variables.	fpathconf(2) PRM
sysconf: get	configurable system variables.	sysconf(3C) PRM
idconfig: produce a new kernel	configuration.	idconfig(1M) SARM
update, or get device driver	configuration data. /delete,	idinstall(1M) SARM
netconfig: network	configuration database.	netconfig(4) NI
netconfig: network	configuration database.	netconfig(4) SARM
getnetconfig: get network	configuration database entry.	getnetconfig(3N) NI
server routines. resolv.conf:	configuration file for name	resolv.conf(4) NUAG
server routines. resolv.conf:	configuration file for name	resolv.conf(4) SARM
TCP/IP. strcf: STREAMS	Configuration File for STREAMS	strcf(4) NUAG
TCP/IP. strcf: STREAMS	Configuration File for STREAMS	strcf(4) SARM
system log daemon. syslog.conf:	configuration file for syslogd	syslog.conf(4) BSD
updating. updaters:	configuration file for YP	updaters(4) NI
config: MULTIBUS II	configuration file format.	config(1M) MBRMAN
doconfig: execute a	configuration script.	doconfig(3N) NI
strconf: change or query stream	configuration. strchg,	strchg(1) STRM
strconf: change or query stream	configuration. strchg,	strchg(1) URM
pushed STREAMS/ autopush:	configure lists of automatically	autopush(1M) SARM
pushed STREAMS/ autopush:	configure lists of automatically	autopush(1M) STRM

parameters. ifconfig: configure network interface ifconfig(1M) NUAG
 parameters. ifconfig: configure network interface ifconfig(1M) SARM
 lpadmin: configure the LP print service. lpadmin(1M) SARM
 efficient/ dma_get_best_mode: Confirm data is aligned for dma_get_best_mode(D3D) DDRM
 t_rcvconnect: receive the confirmation from a connect/ t_rcvconnect(3N) NI
 console flags. confgls: change and display confgls(1M) SARM
 fwtmp, wtmpfix: manipulate connect accounting records. fwtmp(1M) SARM
 and/ /menu_items, item_count: connect and disconnect items to menu_items(3X) CHAR
 /field_count, move_field: connect fields to forms. form_field(3X) CHAR
 on a socket. connect: initiate a connection connect(3N) NI
 t_accept: accept a connect request. t_accept(3N) NI
 t_listen: listen for a connect request. t_listen(3N) NI
 receive the confirmation from a connect request. t_rcvconnect: t_rcvconnect(3N) NI
 getpeername: get name of connected peer. getpeername(3N) NI
 socketpair: create a pair of connected sockets. socketpair(3N) NI
 an outgoing terminal line connection. dial: establish dial(3C) NI
 accept: accept a connection on a socket. accept(3N) NI
 connect: initiate a connection on a socket. connect(3N) NI
 shut down part of a full-duplex connection. shutdown: shutdown(3N) NI
 the/ cciattach: Creates a connection to a subchannel on cciattach(1M) MBRMAN
 or expedited data sent over a connection. t_rcv: receive data t_rcv(3N) NI
 data or expedited data over a connection. t_snd: send t_snd(3N) NI
 t_connect: establish a connection with another/ t_connect(3N) NI
 discipline for unique stream connections. connld: line connld(7) STRM
 listen: listen for connections on a socket. listen(3N) NI
 acctcon, acctcon1, acctcon2: connect-time accounting. acctcon(1M) SARM
 unique stream connections. connld: line discipline for connld(7) STRM
 fsck (ufs): file system consistency check and/ fsck(1M) SARM
 quotacheck: file system quota consistency checker. quotacheck(1M) SARM
 driver. console: console port device console(1M) MBRMAN
 display: system console display. display(7) SARM
 confgls: change and display console flags. confgls(1M) SARM
 a message on stderr or system console. fmtmsg: display fmtmsg(1) URM
 a message on stderr or system console. fmtmsg: display fmtmsg(3C) PRM
 console: STREAMS-based console interface. console(7) STRM
 keyboard: system console keyboard. keyboard(7) SARM
 console: console port device driver. console(1M) MBRMAN
 a driver message on system console. print: display print(D2DK) DDRM
 interface. console: STREAMS-based console console(7) STRM
 driver. i8251: console terminal/iSBXT 351 i8251(1M) MBRMAN
 rci: debug console/rci protocol driver. rci(1M) MBRMAN
 langinfo: language information constants. langinfo(5) PRM
 langinfo: language information constants. langinfo(5) SARM
 math: math functions and constants. math(5) PRM
 unistd: header file for symbolic constants. unistd(4) SARM
 file for implementation-specific constants. limits: header limits(4) PRM
 file for implementation-specific constants. limits: header limits(4) SARM

Permuted Index

mkfs (bfs):	construct a boot file system.	mkfs(1M) SARM
response to a/ mps_mk_bgrant:	construct a buffer grant in	mps_mk_bgrant(D3D) DDRM
response to a/ mps_mk_breject:	construct a buffer reject in	mps_mk_breject(D3D) DDRM
mkfs (generic):	construct a file system.	mkfs(1M) SARM
newfs:	construct a new file system.	newfs(1M) BSD
mkfs (ufs):	construct a ufs file system.	mkfs(1M) SARM
mkfs (s5):	construct an s5 file system.	mkfs(1M) SARM
execute command. xargs:	construct argument list(s) and	xargs(1) URM
to be sent.. mps_mk_brdcst:	constructs a broadcast message	mps_mk_brdcst(D3D) DDRM
to initiate a/ mps_mk_sol:	constructs a message to be sent	mps_mk_sol(D3D) DDRM
to initiate a. mps_mk_solrply:	constructs a message to be sent ...	mps_mk_solrply(D3D) DDRM
message to be/ mps_mk_unsolrply:	constructs a unsolicited reply ...	mps_mk_unsolrply(D3D) DDRM
message to be/ mps_mk_unsol:	constructs an unsolicited	mps_mk_unsol(D3D) DDRM
remove nroff, troff, tbl and eqn	constructs. deroff:	deroff(1) BSD
remove nroff/troff, tbl, and eqn	constructs. deroff:	deroff(1) URM
control maximum system resource	consumption. /setrlimit:	getrlimit(2) PRM
debugging on. Uutry: try to	contact remote system with	Uutry(1M) SARM
/lists device groups which	contain devices that match/	getdgrp(1M) SARM
resources. dfstab: file	containing commands for sharing	dfstab(4) NUAG
resources. dfstab: file	containing commands for sharing	dfstab(4) SARM
idmkinit: reads files	containing specifications.	idmkinit(1M) SARM
information for/ ttydefs: file	contains terminal line settings	ttydefs(4) NI
pkgmap: package	contents description file.	pkgmap(4) SS
retrieve uninterpreted file	contents. elf_rawfile:	elf_rawfile(3E) PRM
bkgreg: change or display the	contents of a backup register.	bkgreg(1M) SARM
clrbuf: erase the	contents of a buffer.	clrbuf(D3DK) DDRM
ls: list the	contents of a directory.	ls(1) BSD
ls: list	contents of directory.	ls(1) URM
ls, lc: list	contents of directory.	ls(1) XNX
text string/ srchtxt: display	contents of, or search for a	srchtxt(1) URM
ucontext: user	context.	ucontext(5) PRM
get and set current user	context. /setcontext:	getcontext(2) PRM
or get signal alternate stack	context. sigaltstack: set	sigaltstack(2) PRM
set and/or get signal stack	context. sigstack:	sigstack(3) BSD
csplit:	context split.	csplit(1) URM
swapcontext: manipulate user	contexts. makecontext,	makecontext(3C) PRM
fcntl: file	control.	fcntl(2) PRM
mctl: memory management	control.	mctl(3) BSD
memcntl: memory management	control.	memcntl(2) PRM
mt: magnetic tape	control.	mt(1) BSD
priocntl: process scheduler	control.	priocntl(1) URM
priocntl: process scheduler	control.	priocntl(2) PRM
uadmin: administrative	control.	uadmin(1M) SARM
uadmin: administrative	control.	uadmin(2) PRM
vc: version	control.	vc(1) PRM
ioctl:	control a character device.	ioctl(D2DK) DDRM
elf_cntl:	control a file descriptor.	elf_cntl(3E) PRM

backup: initiate or control a system backup session. backup(1M) SARM
 /rsh: shell, the standard, job control, and restricted command/ sh(1) URM
 address resolution display and control. arp: arp(1M) NUAG
 address resolution display and control. arp: arp(1M) SARM
 ioctl: control device. ioctl(2) PRM
 IEEE floating-point environment control. /fpsetsticky: fpgetround(3C) PRM
 band. bcanput: test for flow control in specified priority bcanput(D3DK) DDRM
 qband: STREAMS queue flow control information structure. qband(D4DK) DDRM
 init, telinit: process control initialization. init(1M) SARM
 getrlimit, setrlimit: control maximum system resource/ getrlimit(2) PRM
 /set_menu_pad, menu_pad: control menus display/ menu_attributes(3X) CHAR
 ICMP: Internet Control Message Protocol. ICMP(7) NI
 ICMP: Internet Control Message Protocol. ICMP(7) NUAG
 ICMP: Internet Control Message Protocol. ICMP(7) SARM
 putctl: send a control message to a queue. putctl(D3DK) DDRM
 parameter to a/ putctl1: send a control message with a one-byte putctl1(D3DK) DDRM
 jagent: host control of windowing terminal. jagent(5) PRM
 jagent: host control of windowing terminal. jagent(5) SARM
 msgctl: message control operations. msgctl(2) PRM
 semctl: semaphore control operations. semctl(2) PRM
 shmctl: shared memory control operations. shmctl(2) PRM
 fcntl: file control options. fcntl(5) PRM
 fcntl: file control options. fcntl(5) SARM
 generalized process scheduler control. pricntlset: pricntlset(2) PRM
 lpc: line printer control program. lpc(1M) BSD
 TCP: Internet Transmission Control Protocol. TCP(7) NI
 TCP: Internet Transmission Control Protocol. TCP(7) NUAG
 TCP: Internet Transmission Control Protocol. TCP(7) SARM
 /is_wintouched: curses refresh control routines. curs_touch(3X) CHAR
 character and window attribute control routines. /curses curs_attr(3X) CHAR
 curses terminal output option control routines. /nl, nonl: curs_outopts(3X) CHAR
 curses terminal input option control routines. /typeahead: curs_inopts(3X) CHAR
 openlog, closelog, setlogmask: control system log. syslog, syslog(3) BSD
 front end for the Source Code Control System (SCCS). sccs: sccs(1) BSD
 uuucp status inquiry and job control. uustat: uustat(1C) URM
 i546: iSBC 546 multi-port serial controller. i546(1M) MBRMAN
 sac: service access controller. sac(1M) NI
 sac: service access controller. sac(1M) SARM
 sdi_name: get name of addressed controller. sdi_name(D3I) SCSI
 to a line on the communication controller.. /a line discipline ccbind(1M) MBRMAN
 program on the communications controller.. /a line discipline ccifree(1M) MBRMAN
 program on to the communications controller.. /a line discipline cciload(1M) MBRMAN
 sacadm: service access controller administration. sacadm(1M) NI
 sacadm: service access controller administration. sacadm(1M) SARM
 /Disable requests on a DMA Controller channel. dma_disable(D3D) DDRM
 Enable requests on a DMA Controller channel. dma_enable: dma_enable(D3D) DDRM
 i258: iSBC 386/258 peripheral controller device driver. i258(1M) MBRMAN

i530: iSBC 186/530 Ethernet controller device driver. i530(1M) MBRMAN
 186/410 serial communications controller device driver. /iSBC i410(1M) MBRMAN
 a line on the communications controller.. /discipline from cciunbind(1M) MBRMAN
 i214: iSBCR 214 peripheral controller disk device driver. i214(1M) MBRMAN
 i224a: iSBC 186/224A peripheral controller disk device driver. i224a(1M) MBRMAN
 resets and reloads the ethernet controller. enetload: enetload(1M) MBRMAN
 on the communications controller.. /information ccildinfo(1M) MBRMAN
 on the communications controller.. /information ccisubinfo(1M) MBRMAN
 on an ethernet node's ethernet controller. /information enetinfo(1M) MBRMAN
 driver. cci: communications controller interface device cci(1M) MBRMAN
 list on the communications controller.. /line discipline cchildlist(1M) MBRMAN
 on the communications controller. /line information ccilinfo(1M) MBRMAN
 a new host on the communications controller.. /on a line to cciswitch(1M) MBRMAN
 atcs: asynchronous terminal controller script device driver. atcs(1M) MBRMAN
 send SCSI command to the controller. sdi_send: sdi_send(D3I) SCSI
 on the communications controller.. /server information ccisrvinfo(1M) MBRMAN
 a new host on the communications controller. /switches a line to ttyswitch(1M) MBRMAN
 i214tp: iSBC 214 peripheral controller tape device driver. i214tp(1M) MBRMAN
 /iSBC 186/224A peripheral controller tape device driver. i224atp(1M) MBRMAN
 i258tp: iSBC 386/258 peripheral controller tape device driver. i258tp(1M) MBRMAN
 /a co-process with the controlling FMLI application. vsig(1F) CHAR
 tty: controlling terminal interface. tty(7) SARM
 _toupper, _tolower, toascii:/ conv: toupper, tolower, conv(3C) PRM
 terminals. term: conventional names for term(5) CHAR
 terminals. term: conventional names for term(5) SARM
 units: conversion program. units(1) URM
 sfconvert, sgconvert: output conversion. /seconvert, econvert(3) BSD
 iconv: code set conversion tables. iconv(5) SARM
 character classification and conversion tables. /generate chrtrbl(1M) SARM
 iconv: code set conversion utility. iconv(1) URM
 vsprintf: formatted output conversion. /vprintf, vfprintf, printf(3S) BSD
 into a terminfo/ captainfo: convert a termcap description captainfo(1M) CHAR
 into a terminfo/ captainfo: convert a termcap description captainfo(1M) SARM
 dd: convert and copy a file. dd(1M) SARM
 dd: convert and copy a file. dd(1M) URM
 formats. convert: convert archive files to common convert(1) PRM
 and long/ l3tol, ltol3: convert between 3-byte integers l3tol(3C) PRM
 base-64 ASCII/ a64l, l64a: convert between long integer and a64l(3C) PRM
 microseconds. drv_hztousec: convert clock ticks to drv_hztousec(D3DK) DDRM
 to common formats. convert: convert archive files convert(1) PRM
 /gmtime, asctime, tzset: convert date and time to string. ctime(3C) PRM
 strftime, cftime, ascftime: convert date and time to string. strftime(3C) PRM
 /decimal_to_extended: convert decimal record to/ decimal_to_floating(3) BSD
 pass-through device/ sdi_getdev: convert device number to sdi_getdev(D3I) SCSI
 table. htable: convert DoD Internet format host htable(1M) NUAG
 table. htable: convert DoD Internet format host htable(1M) SARM
 string. ecvt, fcvt, gcvt: convert floating-point number to ecvt(3C) PRM

decimal/ /extended_to_decimal: convert floating-point value to floating_to_decimal(3) BSD
 scanf, fscanf, sscanf: convert formatted input. scanf(3S) PRM
 /mvscanw, mvwscanw, vwscanw: convert formatted input from a/ curs_scanw(3X) CHAR
 to physical address. kvtophys: convert kernel virtual address kvtophys(D3D) DDRM
 ticks. drv_usecstohz: convert microseconds to clock drv_usecstohz(D3DK) DDRM
 page structure. page_numtopp: convert page frame number to page_numtopp(D3DK) DDRM
 frame number. page_pptonum: convert page structure to page page_pptonum(D3DK) DDRM
 pages (round down). btop: convert size in bytes to size in btop(D3DK) DDRM
 pages (round up). btopr: convert size in bytes to size in btopr(D3DK) DDRM
 bytes. ptob: convert size in pages to size in ptob(D3DK) DDRM
 double-precision/ strtod, atof: convert string to strtod(3C) PRM
 strtol, strtoul, atol, atoi: convert string to integer. strtol(3C) PRM
 mpscncv: Convert transport code. mpscncv(1M) MBRMAN
 time. getdate: convert user format date and getdate(3C) PRM
 /htonl, htons, ntohl, ntohs: convert values between host and/ byteorder(3N) NI
 address. vtop: convert virtual to physical vtop(D3D) DDRM
 calendar time. mktime: converts a tm structure to a mktime(3C) PRM
 timod: Transport Interface cooperating STREAMS module. timod(7) STRM
 application/ elf_version: coordinate ELF library and elf_version(3E) PRM
 get curses cursor and window coordinates. /getmaxyx: curs_getyx(3X) CHAR
 message/ mps_get_soldata: copies user data from the mps_get_soldata(D3D) DDRM
 message/ mps_get_unsoldata: copies user data from the mps_get_unsoldata(D3D) DDRM
 FMLI/ vsig: synchronize a co-process with the controlling vsig(1F) CHAR
 rcp: remote file copy. rcp(1) NUAG
 rcp: remote file copy. rcp(1) URM
 dd: convert and copy a file. dd(1M) SARM
 dd: convert and copy a file. dd(1M) URM
 copylist: copy a file into memory. copylist(3G) PRM
 tcopy: copy a magnetic tape. tcopy(1) BSD
 copymsg: copy a message. copymsg(D3DK) DDRM
 copyb: copy a message block. copyb(D3DK) DDRM
 copy: copy groups of files. copy(1) XNX
 locations in the kernel. bcopy: copy data between address bcopy(D3DK) DDRM
 user program. copyout: copy data from a driver to a copyout(D3DK) DDRM
 a driver buffer. copyin: copy data from a user program to copyin(D3DK) DDRM
 cpio: copy file archives in and out. cpio(1) URM
 access time. dcopy (generic): copy file systems for optimal dcopy(1M) SARM
 cp: copy files. cp(1) URM
 copy: copy groups of files. copy(1) XNX
 structure. uiomove: copy kernel data using uiop(D4DK) uiomove(D3DK) DDRM
 volcopy (ufs): make a literal copy of a ufs file system. volcopy(1M) SARM
 volcopy (s5): make a literal copy of an s5 file system. volcopy(1M) SARM
 volcopy (generic): make literal copy of file system. volcopy(1M) SARM
 multiple volumes. cpout: copy out file archives to cpout(1M) MBRMAN
 access time. dcopy (s5): copy s5 file systems for optimal dcopy(1M) SARM
 /stredd, strcadd, streccpy: copy strings, compressing. streccpy(3G) PRM
 uuname: UNIX-to-UNIX system copy. uuap, uulog, uuap(1C) URM

public UNIX-to-UNIX system file	copy. uuto, uupick: uuto(1C) URM
program to a driver buffer.	copyb: copy a message block. copyb(D3DK) DDRM
memory.	copyin: copy data from a user copyin(D3DK) DDRM
to a user program.	copylist: copy a file into copylist(3G) PRM
file.	copymsg: copy a message. copymsg(D3DK) DDRM
floor, floorf, ceil, ceilf,	copyout: copy data from a driver copyout(D3DK) DDRM
ieee_functions, fp_class, isnan,	copyright: copyright information copyright(4) SS
overlapped/ /overlay, overwrite,	copysign, fmod, fmodf, fabs,/ floor(3M) PRM
core:	copysign, scalbn: miscellaneous/ ieee_functions(3M) BSD
core:	copywin: overlap and manipulate curs_overlay(3X) CHAR
core:	core: core image file. core(4) PRM
core:	core: core image file. core(4) SARM
core:	core image file. core(4) PRM
core:	core image file. core(4) SARM
processes. gcore: get	core images of running gcore(1) URM
mem, kmem:	core memory. mem(7) SARM
cocreate, cosend, cocheck,	coreceive, codestroy:/ coproc(1F) CHAR
permissions and/ fixperm:	correct or initialize file fixperm(1) XNX
permissions and/ fixperm:	correct or initialize XENIX file fixperm(1M) SARM
synchronization of the/ adjtime:	correct the time to allow adjtime(2) PRM
menu_cursor: pos_menu cursor:	correctly position a menus/ menu_cursor(3X) CHAR
/get /etc/netconfig entry	corresponding to NETPATH/ getnetpath(3N) NI
/receives solicited data that	corresponds to an. mps_AMPreceive(D3D) DDRM
asinf, acos,/ trig: sin, sinf,	cos, cosf, tan, tanf, asin, trig(3M) PRM
codestroy:/ cocreate,	cosend, cocheck, coreceive, coproc(1F) CHAR
acos,/ trig: sin, sinf, cos,	cosf, tan, tanf, asin, asinf, trig(3M) PRM
acosh, atanh:/ sinh, sinhf,	cosh, coshf, tanh, tanhf, asinh, sinh(3M) PRM
acosh,/ sinh, sinhf, cosh,	coshf, tanh, tanhf, asinh, sinh(3M) PRM
wc: word	count. wc(1) URM
sum: print checksum and block	count of a file. sum(1) URM
display line-by-line execution	count profile data. lprof: lprof(1) PRM
out.	cp: copy files. cp(1) URM
multiple volumes.	cpio: copy file archives in and cpio(1) URM
clock: report	cpout: copy out file archives to cpout(1M) MBRMAN
rewrite an existing one.	CPU time used. clock(3C) PRM
setcolor: redefine or	cram- CMOS RAM interface. cram(7) SARM
makesys:	crash: examine system images. crash(1M) SARM
command. gcc:	creat: create a new file or creat(2) PRM
file. tmpnam, tempnam:	create a color. setcolor(1F) CHAR
mkfifo:	create a file system. makesys(1M) SARM
existing one. creat:	create a front-end to the cc gcc(1M) SARM
on the system. groupadd: add	create a name for a temporary tmpnam(3S) PRM
publickey database. newkey:	create a new FIFO. mkfifo(3C) PRM
publickey database. newkey:	create a new file or rewrite an creat(2) PRM
	(create) a new group definition groupadd(1M) SARM
	create a new key in the newkey(1M) NI
	create a new key in the newkey(1M) NUAG

fork: create a new process. fork(2) PRM
 sockets. socketpair: create a pair of connected socketpair(3N) NI
 vi. ctags: create a tags file for use with ctags(1) URM
 tmpfile: create a temporary file. tmpfile(3S) PRM
 communication. socket: create an endpoint for socket(3N) NI
 semaphore. creatsem: create an instance of a binary creatsem(2) XNX
 pipe: create an interprocess channel. pipe(2) PRM
 bibliographic/ indxbib: create an inverted index to a indxbib(1) BSD
 files. admin: create and administer SCCS admin(1) PRM
 form_new: new_form, free_form: create and destroy forms. form_new(3X) CHAR
 /link_field, free_field: create and destroy forms fields. form_field_new(3X) CHAR
 menu_new: new_menu, free_menu: create and destroy menus. menu_new(3X) CHAR
 /new_item, free_item: create and destroy menus items. menu_item_new(3X) CHAR
 /new_panel, del_panel: create and destroy panels. panel_new(3X) CHAR
 /pnoutrefresh, pechochar: create and display curses pads. curs_pad(3X) CHAR
 colltbl: create collation database. colltbl(1M) SARM
 wborder, box, whline, wvline: create curses borders,/ /border, curs_border(3X) CHAR
 syncok, wcursyncup, wsyncdown: create curses windows. /wsyncup, curs_window(3X) CHAR
 gettxt. mkmsgs: create message files for use by mkmsgs(1) URM
 montbl: create monetary database. montbl(1M) SARM
 database. addbib: create or extend a bibliographic addbib(1) BSD
 file system archive. fdp: create, or restore from, a full fdp(1M) SARM
 file system archive. ffile: create, or restore from, a full ffile(1M) SARM
 path. mkdirp, rmdirp: create, remove directories in a mkdirp(3G) PRM
 of a filesystem. fimage: create, restore an image archive fimage(1M) SARM
 filesystem archive. incfile: create, restore an incremental incfile(1M) SARM
 manual. catman: create the cat files for the catman(1M) BSD
 subchannel on the/ cciattach: Creates a connection to a cciattach(1M) MBRMAN
 /routines for dealing with creation and manipulation of/ rpc_dnt_create(3N) NI
 umask: set and get file creation mask. umask(2) PRM
 /routines for dealing with the creation of server handles. rpc_svc_create(3N) NI
 data representation stream creation. /routines for external xdr_create(3N) NI
 a binary semaphore. creatsem: create an instance of creatsem(2) XNX
 structure. cred: access credential cred(D4DK) DDRM
 cred: access credential structure. cred(D4DK) DDRM
 getdev: lists devices based on criteria. getdev(1M) SARM
 pathconv: search FMLI criteria for filename. pathconv(1F) CHAR
 which contain devices that match criteria. /lists device groups getdgrp(1M) SARM
 cron. cron(4) SARM
 cron: clock daemon. cron(1M) SARM
 crontab file. crontab(1) URM
 crontab: user crontab file. crontab(1) URM
 cxref: generate C program cross-reference. cxref(1) PRM
 optimization package. curses: CRT screen handling and curses(3X) CHAR
 hplot,/ plot, aedplot, bgplot, crtplot, dumbplot, gigiplot, plot(1G) BSD
 pg: file perusal filter for CRTs. pg(1) URM
 crypt: encode/decode. crypt(1) URM

encryption functions. **crypt: password and file** **crypt(3X) PRM**
 encryption. **crypt, setkey, encrypt: generate** **crypt(3C) PRM**
 C program. **cscope: interactively examine a** **cscope(1) PRM**
 with a C-like syntax. **csh: shell command interpreter** **csh(1) URM**
 terminal. **csplit: context split.** **csplit(1) URM**
 use with vi. **ct: spawn login to a remote** **ct(1C) URM**
 terminal. **ctags: create a tags file for** **ctags(1) URM**
asctime, tzset: convert date/ **ctermid: generate file name for** **ctermid(3S) PRM**
islower, isupper, isalpha,/ **ctime, localtime, gmtime,** **ctime(3C) PRM**
endpoint. t_look: look at the **ctrace: C program debugger.** **ctrace(1) PRM**
 default/ reset: reset the **ctype: isdigit, isxdigit,** **ctype(3C) PRM**
 getfrm: returns the **cu: call another UNIX system.** **cu(1C) URM**
display the architecture of the **current event on a transport** **t_look(3N) NI**
 get unique identifier of **current form field to its** **reset(1F) CHAR**
 sethostname: get/set name of **current frameID number.** **getfrm(1F) CHAR**
 the numeric identifier of the **current host. arch:** **arch(1) BSD**
 the processor type of the **current host. gethostid:** **gethostid(3) BSD**
 hostname: set or print name of **current host. gethostname,** **gethostname(3) BSD**
 rename login entry to show **current host. hostid: print** **hostid(1) BSD**
 rename login entry to show **current host. mach: display** **mach(1) BSD**
/top_row, item_index: set and get **current host system.** **hostname(1) BSD**
 /field_index: set forms **current layer. relogin:** **relogin(1M) SARM**
 ps: display the status of **current layer. relogin:** **relogin(1M) URM**
pointer to the queue behind the **current menus items.** **menu_item_current(3X) CHAR**
 activity. sact: print **current page and field.** **form_page(3X) CHAR**
domainname: get/set name of **current processes.** **ps(1) BSD**
 sigsetmask: set **current queue. backq: get** **backq(D3DK) DDRM**
 t_getstate: get the **current SCCS file editing** **sact(1) PRM**
 uname: print name of **current secure RPC domain.** **domainname(1M) NI**
 uname: get name of **current signal mask.** **sigsetmask(3) BSD**
 /setcontext: get and set **current state.** **t_getstate(3N) NI**
 the slot in the utmp file of the **current UNIX system.** **uname(1) URM**
 whoami: display the effective **current UNIX system.** **uname(2) PRM**
YP server host. yppoll: return **current user context.** **getcontext(2) PRM**
 /replace_panel: get or set the **current user. ttyslot: find** **ttyslot(3C) PRM**
 getcwd: get pathname of **current username.** **whoami(1) BSD**
 pathname. getwd: get **current version of a YP map at a** **yppoll(1M) NI**
 /form_page, set current field, **current window of a panels/** **panel_window(3X) CHAR**
 top_row,/ /set_current_item, **current working directory.** **getcwd(3C) PRM**
 getitem: return a list of **current working directory** **getwd(3) BSD**
 display environment variables **current field, field_index: set/** **form_page(3X) CHAR**
 mvaddch, mvwaddch, echochar,/ **current_item, set_top_row,** **menu_item_current(3X) CHAR**
 addchnstr, waddchnstr,/ **currently marked menu items.** **getitem(1F) CHAR**
 waddstr, waddnstr, mvaddstr,/ **currently set. printenv:** **printenv(1) BSD**
 crypt: password and file **crypt(3X) PRM**
 crypt, setkey, encrypt: generate **crypt(3C) PRM**
 cscope: interactively examine a **cscope(1) PRM**
 csh: shell command interpreter **csh(1) URM**
 csplit: context split. **csplit(1) URM**
 ct: spawn login to a remote **ct(1C) URM**
 ctags: create a tags file for **ctags(1) URM**
 ctermid: generate file name for **ctermid(3S) PRM**
 ctime, localtime, gmtime, **ctime(3C) PRM**
 ctrace: C program debugger. **ctrace(1) PRM**
 ctype: isdigit, isxdigit, **ctype(3C) PRM**
 cu: call another UNIX system. **cu(1C) URM**
 current event on a transport **t_look(3N) NI**
 current form field to its **reset(1F) CHAR**
 current frameID number. **getfrm(1F) CHAR**
 current host. arch: **arch(1) BSD**
 current host. gethostid: **gethostid(3) BSD**
 current host. gethostname, **gethostname(3) BSD**
 current host. hostid: print **hostid(1) BSD**
 current host. mach: display **mach(1) BSD**
 current host system. **hostname(1) BSD**
 current layer. relogin: **relogin(1M) SARM**
 current layer. relogin: **relogin(1M) URM**
 current menus items. **menu_item_current(3X) CHAR**
 current page and field. **form_page(3X) CHAR**
 current processes. **ps(1) BSD**
 current queue. backq: get **backq(D3DK) DDRM**
 current SCCS file editing **sact(1) PRM**
 current secure RPC domain. **domainname(1M) NI**
 current signal mask. **sigsetmask(3) BSD**
 current state. **t_getstate(3N) NI**
 current UNIX system. **uname(1) URM**
 current UNIX system. **uname(2) PRM**
 current user context. **getcontext(2) PRM**
 current user. ttyslot: find **ttyslot(3C) PRM**
 current username. **whoami(1) BSD**
 current version of a YP map at a **yppoll(1M) NI**
 current window of a panels/ **panel_window(3X) CHAR**
 current working directory. **getcwd(3C) PRM**
 current working directory **getwd(3) BSD**
 current field, field_index: set/ **form_page(3X) CHAR**
 current_item, set_top_row, **menu_item_current(3X) CHAR**
 currently marked menu items. **getitem(1F) CHAR**
 currently set. printenv: **printenv(1) BSD**
 curs_addch: addch, waddch, **curs_addch(3X) CHAR**
 curs_addchnstr: addchnstr, **curs_addchnstr(3X) CHAR**
 curs_addstr: addstr, addnstr, **curs_addstr(3X) CHAR**

attron, wattron, attrset,/
 bell and screen flash routines.
 bkgd, wbkgd: curses window/
 box, whline, wvline: create/
 clear, wclear, clrtoeol,/
 init_pair, init_color,/
 mvdelch, mvwdelch: delete/
 wdeleteln, insdelln,/
 curs_beep: beep, flash:
 /box, whline, wvline: create
 /wstandend, standout, wstandout:
 /color_content, pair_content:
 optimization package.
 /getbegyx, getmaxyx: get
 /longname, termattrs, termname:
 /tgetnum, tgetstr, tgoto, tputs:
 /tigetflag, tigetnum, tigetstr:
 pechochar: create and display
 /is_linetouched, is_wintouched:
 curs_set, napms: low-level
 /scr_set: read (write) a
 /isendwin, set_term, delscreen:
 /slk_attrset, slk_atroff:
 /timeout, wtimeout, typeahead:
 /get character strings from
 (or push back) characters from
 /wsetscreg, scrollok, nl, nonl:
 /flushinp: miscellaneous
 convert formatted input from a
 characters (and attributes) to a
 /add a string of characters to a
 character (with attributes) to a
 /bkgdset, wbkgdset, bkgd, wbkgd:
 character under the cursor in a
 curs_move: move, wmove: move
 scroll, srcl, wscl: scroll a
 and its attributes from a
 character under cursor in a
 a string of characters from a
 (and attributes) from a
 character under the cursor in a
 clear all or part of a
 delete and insert lines in a
 /redrawwin, wredrawln: refresh
 and manipulate overlapped
 wcursyncup, wsyncdown: create
 print formatted output in
 curs_attr: attroff, wattroff, curs_attr(3X) CHAR
 curs_beep: beep, flash: curses curs_beep(3X) CHAR
 curs_bkgd: bkgdset, wbkgdset, curs_bkgd(3X) CHAR
 curs_border: border, wborder, curs_border(3X) CHAR
 curs_clear: erase, werase, curs_clear(3X) CHAR
 curs_color: start_color, curs_color(3X) CHAR
 curs_delch: delch, wdelch, curs_delch(3X) CHAR
 curs_deleteln: deleteln, curs_deleteln(3X) CHAR
 curses bell and screen flash/ curs_beep(3X) CHAR
 curses borders, horizontal and/ curs_border(3X) CHAR
 curses character and window/ curs_attr(3X) CHAR
 curses color manipulation/ curs_color(3X) CHAR
 curses: CRT screen handling and curses(3X) CHAR
 curses cursor and window/ curs_getyx(3X) CHAR
 curses environment query/ curs_termattrs(3X) CHAR
 curses interfaces (emulated) to/ curs_termcap(3X) CHAR
 curses interfaces to terminfo/ curs_terminfo(3X) CHAR
 curses pads. /pnoutrefresh, curs_pad(3X) CHAR
 curses refresh control routines. curs_touch(3X) CHAR
 curses routines. /ripoffline, curs_kernel(3X) CHAR
 curses screen from (to) a file. curs_scr_dump(3X) CHAR
 curses screen initialization and/ curs_initscr(3X) CHAR
 curses soft label routines. curs_slk(3X) CHAR
 curses terminal input option/ curs_inopts(3X) CHAR
 curses terminal keyboard. curs_getstr(3X) CHAR
 curses terminal keyboard. /get curs_getch(3X) CHAR
 curses terminal output option/ curs_outopts(3X) CHAR
 curses utility routines. curs_util(3X) CHAR
 curses widow. /vwscanw: curs_scanw(3X) CHAR
 curses window. /add string of curs_addchstr(3X) CHAR
 curses window and advance/ curs_addstr(3X) CHAR
 curses window and advance/ /add a curs_addch(3X) CHAR
 curses window background/ curs_bkgd(3X) CHAR
 curses window. /before the curs_insch(3X) CHAR
 curses window cursor. curs_move(3X) CHAR
 curses window. curs_scroll: curs_scroll(3X) CHAR
 curses window. /get a character curs_inch(3X) CHAR
 curses window. /mvwdelch: delete curs_delch(3X) CHAR
 curses window. /mvwinstr: get curs_instr(3X) CHAR
 curses window. /of characters curs_inchstr(3X) CHAR
 curses window. /string before curs_insstr(3X) CHAR
 curses window. /wclrtoeol: curs_clear(3X) CHAR
 curses window. /winsertln: curs_deleteln(3X) CHAR
 curses windows and lines. curs_refresh(3X) CHAR
 curses windows. /overlap curs_overlay(3X) CHAR
 curses windows. /syncok, curs_window(3X) CHAR
 curses windows. /vwprintw: curs_printw(3X) CHAR

Permuted Index

mvgetch, mvwgetch, ungetch: get/
mvgetstr, mvwgetstr, wgetstr:/
getbegyx, getmaxyx: get curses/
mvwinch: get a character and/
inchstr, winchstr, winchnstr,/
endwin, isendwin, set_term,/
echo, noecho, halfdelay,/
mvinsch, mvwinsch: insert a/
winsstr, winsnstr, mvinsstr,/
winstr, winnstr, mvinsstr,/
def_shell_mode,/
curses window cursor.
to a curses window and advance
/getbegyx, getmaxyx: get curses
move, wmove: move curses window
/mvwdelch: delete character under
/before the character under the
/before character under the
position forms window
correctly position a menu
to a curses window and advance
idcok immedok, leaveok,/
overwrite, copywin: overlap and/
prefresh, pnoutrefresh,/
mvprintw, mvwprintw, vwprintw:/
wrefresh, wnoutrefresh,/
mvscanw, mvwscanw, vwscanw:/
scr_restore, scr_init, scr_set:/
wscr: scroll a curses window.
/getsyx, setsyx, ripoffline,
silk_refresh, silk_noutrefresh,/
erasechar, has_ic, has_il,/
tgetflag, tgetnum, tgetstr,/
setterm, set_curterm,/
touchline, untouchwin,/
filter, use_env, putwin,/
mvwin, subwin, derwin,/
name of the user.
portions of a UNIX package.
portions of a UNIX package.
each line of a file.
line of a file. cut:
line of a file. fmlcut:
cross-reference.

biod: NFS
biod: NFS

curs_getch: getch, wgetch, curs_getch(3X) CHAR
curs_getstr: getstr, wgetstr, curs_getstr(3X) CHAR
curs_getyx: getyx, getparyx, curs_getyx(3X) CHAR
curs_inch: inch, winch, mvinch, curs_inch(3X) CHAR
curs_inchstr: inchstr, curs_inchstr(3X) CHAR
curs_initscr: initscr, newterm, curs_initscr(3X) CHAR
curs_inopts: cbreak, nocbreak, curs_inopts(3X) CHAR
curs_insch: insch, winsch, curs_insch(3X) CHAR
curs_instr: insstr, insnstr, curs_instr(3X) CHAR
curs_instr: instr, innstr, curs_instr(3X) CHAR
curs_kernel: def_prog_mode, curs_kernel(3X) CHAR
curs_move: move, wmove: move curs_move(3X) CHAR
cursor. /a string of characters curs_addstr(3X) CHAR
cursor and window coordinates. curs_getyx(3X) CHAR
cursor. curs_move: curs_move(3X) CHAR
cursor in a curses window. curs_delch(3X) CHAR
cursor in a curses window. curs_insch(3X) CHAR
cursor in a curses window. curs_instr(3X) CHAR
cursor. /pos_form_cursor: form_cursor(3X) CHAR
cursor. /pos_menu_cursor: menu_cursor(3X) CHAR
cursor. /(with attributes) curs_addch(3X) CHAR
curs_outopts: clearok, idlok, curs_outopts(3X) CHAR
curs_overlay: overlay, curs_overlay(3X) CHAR
curs_pad: newpad, subpad, curs_pad(3X) CHAR
curs_printw: printw, wprintw, curs_printw(3X) CHAR
curs_refresh: refresh, curs_refresh(3X) CHAR
curs_scanw: scanw, wscanw, curs_scanw(3X) CHAR
curs_scr_dump: scr_dump, curs_scr_dump(3X) CHAR
curs_scroll: scroll, srcl, curs_scroll(3X) CHAR
curs_set, napms: low-level/ curs_kernel(3X) CHAR
curs_slk: slk_init, slk_set, curs_slk(3X) CHAR
curs_termattrs: baudrate, curs_termattrs(3X) CHAR
curs_termcap: tgetent, curs_termcap(3X) CHAR
curs_terminfo: setupterm, curs_terminfo(3X) CHAR
curs_touch: touchwin, curs_touch(3X) CHAR
curs_util: unctrl, keyname, curs_util(3X) CHAR
curs_window: newwin, delwin, curs_window(3X) CHAR
cuserid: get character login cuserid(3S) PRM
custom: install specific custom(1M) SARM
custom: install specific custom(1) XNX
cut: cut out selected fields of cut(1) URM
cut out selected fields of each cut(1) URM
cut out selected fields of each fmlcut(1F) CHAR
cxref: generate C program cxref(1) PRM
d258: i82258 ADMA device driver. d258(1M) MBRMAN
daemon. biod(1M) NUAG
daemon. biod(1M) SARM

cron: clock
 inetd: Internet services
 inetd: Internet services
 inetd: Internet services
 listen: network listener
 lockd: network lock
 nfsd: NFS
 nfsd: NFS
 routed: network routing
 routed: network routing
 strerr: STREAMS error logger
 strerr: STREAMS error logger
 a MULTIBUS II boot service
 file for syslogd system log
 rfudaemon: Remote File Sharing
 rfudaemon: Remote File Sharing
 runacct: run
 Protocol server. rarpd:
 Protocol server. rarpd:
 telnetd:
 telnetd:
 Protocol server. tftpd:
 Protocol server. tftpd:
 tnamed, in.tnamed:
 tnamed, in.tnamed:
 netrc: file for ftp remote login
 netrc: file for ftp remote login
 prof: display profile
 sdgetv: synchronize shared
 get device driver configuration
 if forms field has off-screen
 /time a command; report process
 hosts: host name
 hosts: host name
 hosts: host name
 networks: network name
 networks: network name
 protocols: protocol name
 protocols: protocol name
 rpc: rpc program number
 terminfo: terminal capability
 terminfo: terminal capability
 ypcat: print values in a YP
 file. newaliases: rebuild the
 a text string from a message
 daemon. cron(1M) SARM
 daemon. inetd(1M) NI
 daemon. inetd(1M) NUAG
 daemon. inetd(1M) SARM
 daemon. listen(1M) SARM
 daemon. lockd(1M) NUAG
 daemon. nfsd(1M) NUAG
 daemon. nfsd(1M) SARM
 daemon. routed(1M) NUAG
 daemon. routed(1M) SARM
 daemon. strerr(1M) SARM
 daemon. strerr(1M) STRM
 daemon. bootserver: starts bootserver(1M) MBRMAN
 daemon. /configuration syslog.conf(4) BSD
 daemon process. rfudaemon(1M) NUAG
 daemon process. rfudaemon(1M) SARM
 daily accounting. runacct(1M) SARM
 DARPA Reverse Address Resolution rarpd(1M) NUAG
 DARPA Reverse Address Resolution rarpd(1M) SARM
 DARPA TELNET protocol server. telnetd(1M) NUAG
 DARPA TELNET protocol server. telnetd(1M) SARM
 DARPA Trivial File Transfer tftpd(1M) NUAG
 DARPA Trivial File Transfer tftpd(1M) SARM
 DARPA trivial name server. tnamed(1M) NUAG
 DARPA trivial name server. tnamed(1M) SARM
 data. netrc(4) NUAG
 data. netrc(4) SARM
 data. prof(1) PRM
 data access. sdgetv(2) XNX
 data. /add, delete, update, or idinstall(1M) SARM
 data ahead or behind. /tell form_data(3X) CHAR
 data and system activity. timex(1) URM
 data base. hosts(4) NI
 data base. hosts(4) NUAG
 data base. hosts(4) SARM
 data base. networks(4) NUAG
 data base. networks(4) SARM
 data base. protocols(4) NUAG
 data base. protocols(4) SARM
 data base. rpc(4) NI
 data base. terminfo(4) CHAR
 data base. terminfo(4) SARM
 data base. ypcat(1) NI
 data base for the mail aliases newaliases(1M) BSD
 data base. gettxt: retrieve gettxt(1) URM

Permuted Index

<p> /dbm_open, dbm_store: delete, firstkey, nextkey: for a text string in, message in the kernel. bcopy: copy mps_free_dmabuf: frees a list of /returns a pointer to a list of generate disk accounting execution count profile elf_rawdata: get section retrieve file identification t_rcvuderr: receive a unit mb2a_getreqfrag: Receives a mb2s_getreqfrag: Receives a read: read program. copyout: copy driver buffer. copyin: copy mps_get_soldata: copies user mps_get_unsoldata: copies user /sgetl: access long integer /receives solicited network. spray: scatter dma_get_best_mode: Confirm reply.. mps_get_reply_len: get Release 3.0.. edlina: External test whether a message is a connection. t_snd: send a connection. t_rcv: receive t_snd: send data or expedited /restores of filesystems, nlsgetcall: get client's or unlock process, text, or /library routines for external /library routines for external /library routines for external library routines for external library routines for external call. stat: call. stat: call. stat: synchronize access to a shared attach and detach a shared brk, sbrk: change t_rcv: receive data or expedited using DMA. dma_buf: using DMA. dma_cb: using uio(D4DK). iovec: </p>	<p> data base subroutines. ndbm(3) BSD data base subroutines. /store, dbm(3X) BSD data bases. /of, or search srchtxt(1) URM data between address locations bcopy(D3DK) DDRM data buffer descriptors.. mps_free_dmabuf(D3D) DDRM data buffer descriptors.. mps_get_dmabuf(D3D) DDRM data by user ID. diskusg: diskusg(1M) SARM data. /display line-by-line lprof(1) PRM data. elf_getdata, elf_newdata, elf_getdata(3E) PRM data. elf_getident: elf_getident(3E) PRM data error indication. t_rcvuderr(3N) NI data fragment of a request/ mb2a_getreqfrag(3tai) TAI data fragment of a request/ mb2s_getreqfrag(3tai) TAI data from a device. read(D2DK) DDRM data from a driver to a user copyout(D3DK) DDRM data from a user program to a copyin(D3DK) DDRM data from the message buffer.. mps_get_soldata(D3D) DDRM data from the message buffer.. mps_get_unsoldata(D3D) DDRM data in a machine-independent/ sputl(3X) PRM data in fragments when. mps_AMPreceive_frag(D3D) DDRM data in order to check the spray(3N) NI data is aligned for efficient/ ... dma_get_best_mode(D3D) DDRM data length for a solicited mps_get_reply_len(D3D) DDRM Data Link driver for iNA961 edlina(1M) MBRMAN data message. datamsq: datamsq(D3DK) DDRM data or expedited data over a t_snd(3N) NI data or expedited data sent over t_rcv(3N) NI data over a connection. t_snd(3N) NI data partitions, or disks. restore(1M) SARM data passed via the listener. nlsgetcall(3N) NI data. plock: lock into memory plock(2) PRM data representation. xdr_admin(3N) NI data representation. xdr_complex(3N) NI data representation stream/ xdr_create(3N) NI data representation. xdr: xdr(3N) NI data representation. /xdr_void: xdr_simple(3N) NI data returned by stat system stat(4) XNX data returned by stat system stat(5) PRM data returned by stat system stat(5) SARM data segment. sdenter, sdleave: sdenter(2) XNX data segment. sdget, sdfree: sdget(2) XNX data segment space allocation. brk(2) PRM data sent over a connection. t_rcv(3N) NI data storage structure for I/O dma_buf(D4D) DDRM data storage structure for I/O dma_cb(D4D) DDRM data storage structure for I/O iovec(D4DK) DDRM </p>
--	--

datab: STREAMS message
 /receives solicited
 /sends solicited
 write: write
 rdchk: check to see if there is
 buf: block I/O
 /elf32_xlatetom: class-dependent
 /field_arg: forms field
 nl_types: native language
 nl_types: native language
 types: primitive system
 t_rcvudata: receive a
 t_sndudata: send a
 uiomove: copy kernel
 /associate application
 associate application
 associate application
 /associate application
 associate application
 forms field has/ form_data:
 structure.
 colltbl: create collation
 inetd.conf: Internet servers
 inetd.conf: Internet servers
 montbl: create monetary
 netconfig: network configuration
 netconfig: network configuration
 publickey: public key
 publickey: public key
 sortbib: sort a bibliographic
 ypinit: build and install YP
 ypmake: rebuild YP
 create or extend a bibliographic
 structure. ypfiles: the YP
 index to a bibliographic
 grpck: check group
 pwck: check password
 get network configuration
 references from a bibliographic
 to the software installation
 references in a bibliographic
 a new key in the publickey
 a new key in the publickey
 join: relational
 Ethernet address to hostname
 Ethernet address to hostname
 remove a file from software
 data structure. datab(D4DK) DDRM
 data that corresponds to an. mps_AMPreceive(D3D) DDRM
 data that is not part of any. mps_AMPsend_data(D3D) DDRM
 data to a device. write(D2DK) DDRM
 data to be read. rdchk(2) XNX
 data transfer structure. buf(D4DK) DDRM
 data translation. elf_xlate(3E) PRM
 data type validation. form_field_validation(3X) CHAR
 data types. nl_types(5) PRM
 data types. nl_types(5) SARM
 data types. types(5) PRM
 data unit. t_rcvudata(3N) NI
 data unit. t_sndudata(3N) NI
 data using uio(D4DK) structure. uiomove(D3DK) DDRM
 data with a panels panel. panel_userptr(3X) CHAR
 data with forms. /field_userptr: .. form_field_userptr(3X) CHAR
 data with forms. /form_userptr: form_userptr(3X) CHAR
 data with menus items. menu_item_userptr(3X) CHAR
 data with menus. /menu_userptr: menu_userptr(3X) CHAR
 data ahead, data behind: tell if form_data(3X) CHAR
 datab: STREAMS message data datab(D4DK) DDRM
 database. colltbl(1M) SARM
 database. inetd.conf(4) NUAG
 database. inetd.conf(4) SARM
 database. montbl(1M) SARM
 database. netconfig(4) NI
 database. netconfig(4) SARM
 database. publickey(4) NI
 database. publickey(4) NUAG
 database. sortbib(1) BSD
 database. ypinit(1M) NI
 database. ypmake(1M) NI
 database. addbib: addbib(1) BSD
 database and directory ypfiles(4) NI
 database. /create an inverted indxbib(1) BSD
 database entries. grpck(1M) BSD
 database entries. pwck(1M) BSD
 database entry. getnetconfig: getnetconfig(3N) NI
 database. /expand and insert refer(1) BSD
 database. installf: add a file installf(1M) SS
 database. lookbib: find lookbib(1) BSD
 database. newkey: create newkey(1M) NI
 database. newkey: create newkey(1M) NUAG
 database operator. join(1) URM
 database or domain. ethers: ethers(4) NUAG
 database or domain. ethers: ethers(4) SARM
 database. removef: removef(1M) SS

Permuted Index

format and print a bibliographic
 delete, firstkey, nextkey:
 curses interfaces to terminfo
 a terminal or query terminfo
 has/ form_data: data ahead,
 UDP: Internet User
 UDP: Internet User
 UDP: Internet User
 is a data message.
 date: print and set the
 ftime: get time and
 ftime: get
 getdate: convert user format
 settimeofday: get or set the
 settimeofday: get or set the
 /gmtime, asctime, tzset: convert
 /cftime, ascftime: convert
 rdate: set system
 rdate: set system
 prompts for and validates a
 prompts for and validates a

 the access and modification
 verify and return a time of
 verify and return a time of
 store, delete, firstkey,/br/>
 store, delete, firstkey,/br/>
 makedbm: make a YP
 dbm_delete, dbm_error,/ ndbm:
 dbm_error,/ ndbm: dbm_clearerr,
 firstkey,/ dbm, dbminit,
 firstkey,/ dbm: dbminit,
 ndbm: dbm_clearerr, dbm_close,
 /dbm_close, dbm_delete,
 /dbm_delete, dbm_error,
 dbm_open,/ /dbm_error, dbm_fetch,
 delete, firstkey, nextkey:/ dbm,
 delete, firstkey, nextkey:/ dbm:
 /dbm_fetch, dbm_firstkey,
 /dbm_firstkey, dbm_nextkey,
 /dbm_nextkey, dbm_open,
 panel message delivery.

 systems for optimal access/
 for optimal access time.

database. roffbib: roffbib(1) BSD
 database subroutines. /store, dbm(3) NI
 database. /tigetnum, tigetstr: curs_terminfo(3X) CHAR
 database. tput: initialize tput(1) URM
 data behind: tell if forms field form_data(3X) CHAR
 Datagram Protocol. UDP(7) NI
 Datagram Protocol. UDP(7) NUAG
 Datagram Protocol. UDP(7) SARM
 datamsq: test whether a message datamsq(D3DK) DDRM
 date. date(1) URM
 date. ftime(2) XNX
 date and time. ftime(3C) BSD
 date and time. getdate(3C) PRM
 date and time. gettimeofday, gettimeofday(3) BSD
 date and time. gettimeofday, gettimeofday(3C) PRM
 date and time to string. ctime(3C) PRM
 date and time to string. strptime(3C) PRM
 date from a remote host. rdate(1M) NUAG
 date from a remote host. rdate(1M) SARM
 date. /helpdate, valdate: ckdate(1) SARM
 date. /helpdate, valdate: ckdate(1) SS
 date: print and set the date. date(1) URM
 dates of files. settime: change settime(1) XNX
 day. cktime: display a prompt; cktime(1) SARM
 day. cktime: display a prompt; cktime(1) SS
 dbm: dbminit, dbmclose, fetch, dbm(3X) BSD
 dbm, dbminit, dbmclose, fetch, dbm(3) NI
 dbm file. makedbm(1M) NI
 dbm_clearerr, dbm_close, ndbm(3) BSD
 dbm_close, dbm_delete, ndbm(3) BSD
 dbmclose, fetch, store, delete, dbm(3) NI
 dbmclose, fetch, store, delete, dbm(3X) BSD
 dbm_delete, dbm_error,/ ndbm(3) BSD
 dbm_error, dbm_fetch,/ ndbm(3) BSD
 dbm_fetch, dbm_firstkey,/ ndbm(3) BSD
 dbm_firstkey, dbm_nextkey, ndbm(3) BSD
 dbminit, dbmclose, fetch, store, dbm(3) NI
 dbminit, dbmclose, fetch, store, dbm(3X) BSD
 dbm_nextkey, dbm_open,/ ndbm(3) BSD
 dbm_open, dbm_store: data base/ ndbm(3) BSD
 dbm_store: data base/ ndbm(3) BSD
 dbon: sets target for front dbon(1M) MBRMAN
 dc: desk calculator. dc(1) URM
 dcopy (generic): copy file dcopy(1M) SARM
 dcopy (s5): copy s5 file systems dcopy(1M) SARM
 dd: convert and copy a file. dd(1M) SARM
 dd: convert and copy a file. dd(1M) URM

/library routines for server/ /library routines for space. bp_mapout: driver. rci: ctrace: C program fsdb (generic): file system sdb: symbolic strip: strip symbol table, to contact remote system with /convert floating-point value to /decimal_to_extended: convert /decimal_to_single, decimal/ /decimal_to_double, decimal_to_single,/ decimal_to_floating: /hide_panel, panel_hidden: panels /top_panel, bottom_panel: panels /panel_above, panel_below: panels streamtab: STREAMS entity /encode a binary file, or keylogin: keylogin: kill: terminate a process by login: login timezone: set timezone: set the current form field to its vfstab: table of file system admin: installation write the manufacturer's groupdel: delete a group /add (create) a new group groupmod: modify a group IEEE floating point eqnchar: special character reset_prog_mode,/ curs_kernel: curs_kernel: def_prog_mode, for a specified number of clock/ specified number of/ delay: /filter, use_env, putwin, getwin, mvwdelch: delete/ curs_delch: /setupterm, setterm, set_curterm, the system. groupdel: system. userdel: curses/ /insertln, winsertln a/ /wdelch, mvdelch, mvwdelch: /dbmunit, dbmclose, fetch, store, dealing with creation and/ rpc_clnt_create(3N) NI dealing with the creation of rpc_svc_create(3N) NI deallocate virtual address bp_mapout(D3DK) DDRM debug console/rci protocol rci(1M) MBRMAN debugger. ctrace(1) PRM debugger. fsdb(1M) SARM debugger. sdb(1) PRM debugging and line. strip(1) PRM debugging on. Uutry: try Uutry(1M) SARM decimal record. floating_to_decimal(3) BSD decimal record to floating-point/ decimal_to_floating(3) BSD decimal_to_double,/ decimal_to_floating(3) BSD decimal_to_extended: convert decimal_to_floating(3) BSD decimal_to_floating: decimal_to_floating(3) BSD decimal_to_single,/ decimal_to_floating(3) BSD deck manipulation routines. panel_show(3X) CHAR deck manipulation routines. panel_top(3X) CHAR deck traversal primitives. panel_above(3X) CHAR declaration structure. streamtab(D4DK) DDRM decode its ASCII representation. uuencode(1C) URM decrypt and store secret key. keylogin(1) NI decrypt and store secret key. keylogin(1) NUAG default. kill(1) URM default file. login(4) SARM default system time zone. timezone(4) PRM default system time zone. timezone(4) SARM default values. reset: reset reset(1F) CHAR defaults. vfstab(4) SARM defaults file. admin(4) SARM defect list.. /and optionally mdl(1M) MBRMAN definition from the system. groupdel(1M) SARM definition on the system. groupadd(1M) SARM definition on the system. groupmod(1M) SARM definitions. floatingpoint: floatingpoint(3) BSD definitions for eqn. eqnchar(7) BSD def_prog_mode, def_shell_mode, curs_kernel(3X) CHAR def_shell_mode, reset_prog_mode,/ curs_kernel(3X) CHAR delay: delay process execution delay(D3DK) DDRM delay process execution for a delay(D3DK) DDRM delay_output, flushinp:/ curs_util(3X) CHAR delch, wdelch, mvdelch, curs_delch(3X) CHAR del_curterm, restartterm, tparm,/ curs_terminfo(3X) CHAR delete a group definition from groupdel(1M) SARM delete a user's login from the userdel(1M) SARM delete and insert lines in a curs_deleteln(3X) CHAR delete character under cursor in curs_delch(3X) CHAR delete, firstkey, nextkey: data/ dbm(3X) BSD

Permuted Index

dbmunit, dbmclose, fetch, store, delete, firstkey, nextkey:/ dbm, dbm(3) NI
 driver/ idinstall: add, delete, update, or get device idinstall(1M) SARM
 winsdelln,/ curs_deleteln: deleteln, wdeleteln, insdelln, curs_deleteln(3X) CHAR
 bgets: read stream up to next delimiter. bgets(3G) PRM
 basename, dirname: deliver portions of path names. basename(1) URM
 tail: deliver the last part of a file. tail(1) URM
 smtpqer: queue mail for delivery by SMTP. smtpqer(1M) SARM
 target for front panel message delivery. dbon: sets dbon(1M) MBRMAN
 panels. panel_new: new_panel, del_panel: create and destroy panel_new(3X) CHAR
 /endwin, isendwin, set_term, delscreen: curses screen/ curs_initscr(3X) CHAR
 or task removal tool. delsysadm: sysadm interface menu delsysadm(1M) SARM
 or task removal tool. delsysadm: sysadm interface menu delsysadm(1M) SS
 an SCCS file. prt: display the delta and commentary history of prt(1) BSD
 the delta comment of an SCCS delta. cdc: change cdc(1) PRM
 delta: make a delta (change) to an SCCS file. delta(1) PRM
 cdc: change the delta comment of an SCCS delta. cdc(1) PRM
 rmdel: remove a delta from an SCCS file. rmdel(1) PRM
 an SCCS file. delta: make a delta (change) to delta(1) PRM
 comb: combine SCCS deltas. comb(1) PRM
 mvderwin,/ curs_window: newwin, delwin, mvwin, subwin, derwin, curs_window(3X) CHAR
 msg: permit or deny messages. msg(1) URM
 files. depend: software dependencies depend(4) SS
 ldd: list dynamic dependencies. ldd(1) PRM
 depend: software dependencies files. depend(4) SS
 and eqn constructs. deroff: remove nroff, troff, tbl deroff(1) BSD
 and eqn constructs. deroff: remove nroff/troff, tbl, deroff(1) URM
 /newwin, delwin, mvwin, subwin, derwin, mvderwin, dupwin,/ curs_window(3X) CHAR
 description into a terminfo description. /convert a termcap captainfo(1M) CHAR
 description into a terminfo description. /convert a termcap captainfo(1M) SARM
 pkgmap: package contents description file. pkgmap(4) SS
 captainfo: convert a termcap description into a terminfo/ captainfo(1M) CHAR
 captainfo: convert a termcap description into a terminfo/ captainfo(1M) SARM
 get menus item name and description. /item_description: menu_item_name(3X) CHAR
 compare or print out terminfo descriptions. infocmp: infocmp(1M) CHAR
 compare or print out terminfo descriptions. infocmp: infocmp(1M) SARM
 close: close a file descriptor. close(2) PRM
 dup2: duplicate an open file descriptor. dup2(3C) PRM
 dup: duplicate an open file descriptor. dup(2) PRM
 dupb: duplicate a message block descriptor. dupb(D3DK) DDRM
 elf_begin: make a file descriptor. elf_begin(3E) PRM
 elf_cntl: control a file descriptor. elf_cntl(3E) PRM
 elf_update: update an ELF descriptor. elf_update(3E) PRM
 isastream: test a file descriptor. isastream(3C) STRM
 storage for a DMA Buffer Descriptor. /Allocate dma_get_buf(D3D) DDRM
 Free storage for a DMA Buffer descriptor. dma_free_buf: dma_free_buf(D3D) DDRM
 a name from a STREAMS-based file descriptor. fdetach: detach fdetach(1M) STRM
 a name from a STREAMS-based file descriptor. fdetach: detach fdetach(3C) STRM

/dev/fd: file descriptor files. fd(4) SARM
 getdtablesize: get descriptor table size. getdtablesize(3) BSD
 /attach a STREAMS-based file descriptor to an object. fattach(3C) STRM
 frees a list of data buffer descriptors.. mps_free_dmabuf: . mps_free_dmabuf(D3D) DDRM
 pointer to a list of data buffer descriptors.. /returns a mps_get_dmabuf(D3D) DDRM
 for certain board types in the designated slot. /checks ics_agent_cmp(D3D) DDRM
 dc: desk calculator. dc(1) URM
 /free_field,: create and destroy forms fields. form_field_new(3X) CHAR
 new_form, free_form: create and destroy forms. form_new: form_new(3X) CHAR
 /new_item, free_item: create and destroy menus items. menu_item_new(3X) CHAR
 new_menu, free_menu: create and destroy menus. menu_new: menu_new(3X) CHAR
 new_panel, del_panel: create and destroy panels. panel_new: panel_new(3X) CHAR
 STREAMS-based file/ fdetach: detach a name from a fdetach(1M) STRM
 STREAMS-based file/ fdetach: detach a name from a fdetach(3C) STRM
 sdget, sdfree: attach and detach a shared data segment. sdget(2) XNX
 sigaction: detailed signal management. sigaction(2) PRM
 file. access: determine accessibility of a access(2) PRM
 drv_priv: determine driver privilege. drv_priv(D3DK) DDRM
 fstyp (generic): determine file system type. fstyp(1M) SARM
 elf_kind: determine file type. elf_kind(3E) PRM
 file: determine file type. file(1) URM
 pages. mincore: determine residency of memory mincore(2) PRM
 /finite, fpclass, unordered: determine type of floating-point/ isnan(3C) PRM
 buffer is encrypted. isencrypt: determine whether a character isencrypt(3G) PRM
 can accept binary/ ckbinarsys: determine whether remote system ckbinarsys(1M) SARM
 attributes. devattr: lists device devattr(1M) SARM
 /dev/fd: file descriptor files. fd(4) SARM
 devfree: release devices from devfree(1M) SARM
 exclusive use. device. close(D2DK) DDRM
 close: relinquish access to a device. init(D2D) DDRM
 init: initialize a device. ioctl(2) PRM
 ioctl: control device. ioctl(D2DK) DDRM
 ioctl: control a character device. open(D2DK) DDRM
 open: gain access to a device. read(D2DK) DDRM
 read: read data from a device. size(D2DK) DDRM
 size: return size of logical device. start(D2D) DDRM
 start: start access to a device. write(D2DK) DDRM
 write: write data to a device accessibility. getvol(1M) SARM
 getvol: verifies device attributes. devattr(1M) SARM
 devattr: lists device driver. console(1M) MBRMAN
 console: console port device driver. d258(1M) MBRMAN
 d258: i82258 ADMA device driver. dma(1M) MBRMAN
 dma: DMA device driver. iasy(1M) MBRMAN
 iasy: asynchronous (terminal) device driver. ics(1M) MBRMAN
 ics: Interconnect Space device driver. mpc(1M) MBRMAN
 mpc: MPC device driver. mps(1M) MBRMAN
 mps: Message Passing Space device driver. /asynchronous atcs(1M) MBRMAN
 terminal controller script

Permuted Index

controller interface device driver. /communications cci(1M) MBRMAN
 /add, delete, update, or get device driver configuration/ idinstall(1M) SARM
 214 peripheral controller disk device driver. i214: iSBCR i214(1M) MBRMAN
 214 peripheral controller tape device driver. i214tp: iSBC i214tp(1M) MBRMAN
 386/258 peripheral controller device driver. i258: iSBC i258(1M) MBRMAN
 354 dual channel serial-port device driver. i354: iSBX i354(1M) MBRMAN
 iSBC 186/530 Ethernet controller device driver. i530: i530(1M) MBRMAN
 peripheral controller disk device driver. /iSBC 186/224A i224a(1M) MBRMAN
 peripheral controller tape device driver. /iSBC 186/224A i224atp(1M) MBRMAN
 serial communications controller device driver. /iSBC 186/410 i410(1M) MBRMAN
 peripheral controller tape device driver. /iSBC 386/258 i258tp(1M) MBRMAN
 serial, and AT&T. mouse: mouse device driver supporting Bus, mouse(7) MOUSE
 OSI Transport Service (ots) device driver. /System V/386 ots(1M) MBRMAN
 fddd: MULTIBUS flexible disk device drivers. fddd(1M) MBRMAN
 hddd: MULTIBUS hard disk device drivers. hddd(1M) MBRMAN
 tdd: MULTIBUS tape device drivers. tdd(1M) MBRMAN
 to the slave pseudo-terminal device. grantpt: grant access grantpt(3C) STRM
 listdgrp: lists members of a device group. listdgrp(1M) SARM
 putdgrp: edits device group table. putdgrp(1) SARM
 devices that/ getdgrp: lists device groups which contain getdgrp(1M) SARM
 archives: device header file. archives(4) SARM
 intr: process a device interrupt. intr(D2D) DDRM
 segmap: map device memory into user space. segmap(D2K) DDRM
 mapping for memory mapped device. mmap: check virtual mmap(D2K) DDRM
 devnm: device name. devnm(1M) SARM
 madev, major, minor: manage a device number. madev(3C) PRM
 device number to pass-through device number. /convert sdi_getdev(D3I) SCSI
 minor/ madevice: make device number from major and madevice(D3DK) DDRM
 get major or internal major device number. getmajor: getmajor(D3DK) DDRM
 get minor or internal minor device number. getminor: getminor(D3DK) DDRM
 device/ sdi_getdev: convert device number to pass-through sdi_getdev(D3I) SCSI
 clone: open any major/minor device pair on a STREAMS driver. clone(7) STRM
 print the VTOC of a block device. prtvtoc: prtvtoc(1M) SARM
 of the slave pseudo-terminal device. ptsname: get name ptsname(3C) STRM
 putdev: edits device table. putdev(1) SARM
 Administration. SA: devices administered by System sa(7) SARM
 getdev: lists devices based on criteria. getdev(1M) SARM
 devreserv: reserves devices for exclusive use. devreserv(1M) SARM
 devfree: release devices from exclusive use. devfree(1M) SARM
 /device groups which contain devices that match criteria. getdgrp(1M) SARM
 devnm: device name. devnm(1M) SARM
 devreserv: reserves devices for devreserv(1M) SARM
 exclusive use. devreserv: reserves devices for devreserv(1M) SARM
 free disk blocks and files. df (generic): report number of df(1M) SARM
 free disk blocks and files. df (generic): report number of df(1M) URM
 file systems. df: report free disk space on df(1) BSD
 disk blocks and i-nodes for s5/ df (s5): report number of free df(1M) SARM
 on ufs file systems. df (ufs): report free disk space df(1M) SARM

resource information.
 resource information.
 resource information.
 resource information.
 resource information.
 resources from remote or local/
 resources from remote or local/
 resources from remote systems.
 resources from remote systems.
 resources from remote systems.
 resources from remote systems.
 for sharing resources.
 for sharing resources.
 dlerror: get
 terminal line connection.
 look: find words in the system
 bdiff: big
 comparator.
 comparison.
 times. difftime: computes the
 a troff input/ diffmk: mark
 sdiff: print file
 diff:
 diff3: 3-way
 versions of a troff input file.
 difference between two calendar/
 wd: Western
 directories.
 directories.
 on a stream in the reverse
 sockio: ioctl's that operate
 dir (ufs): format of ufs
 mkdir: make
 rm, rmdir: remove files or
 file. uucheck: check the uucp
 mkdirp, rmdirp: create, remove
 link and unlink files and
 search for named file in named
 cd: change working
 chdir, fchdir: change working
 chroot: change root
 ls: list the contents of a
 ls: list contents of
 ls, lc: list contents of
 mkdir: make a
 dfmounts: display mounted dfmounts(1M) NUAG
 dfmounts: display mounted dfmounts(1M) SARM
 dfmounts: display mounted NFS dfmounts(1M) NUAG
 dfmounts: display mounted NFS dfmounts(1M) SARM
 dfmounts: display mounted RFS dfmounts(1M) NUAG
 dfmounts: display mounted RFS dfmounts(1M) SARM
 dfshares: list available dfshares(1M) NUAG
 dfshares: list available dfshares(1M) SARM
 dfshares: list available NFS dfshares(1M) NUAG
 dfshares: list available NFS dfshares(1M) SARM
 dfshares: list available RFS dfshares(1M) NUAG
 dfshares: list available RFS dfshares(1M) SARM
 dfstab: file containing commands dfstab(4) NUAG
 dfstab: file containing commands dfstab(4) SARM
 diagnostic information. dlerror(3X) PRM
 dial: establish an outgoing dial(3C) NI
 dictionary or lines in a sorted/ look(1) BSD
 diff. bdiff(1) URM
 diff: differential file diff(1) URM
 diff3: 3-way differential file diff3(1) URM
 difference between two calendar difftime(3C) PRM
 differences between versions of diffmk(1) BSD
 differences side-by-side. sdiff(1) URM
 differential file comparator. diff(1) URM
 differential file comparison. diff3(1) URM
 diffmk: mark differences between diffmk(1) BSD
 difftime: computes the difftime(3C) PRM
 Digital 8003 Adapter Board. wd(7) SARM
 dir (s5): format of s5 dir(4) SARM
 dir (ufs): format of ufs dir(4) SARM
 dircmp: directory comparison. dircmp(1) URM
 direction. /send a message qreply(D3DK) DDRM
 directly on sockets. sockio(7) NI
 directories. dir(4) SARM
 directories. mkdir(1) URM
 directories. rm(1) URM
 directories and permissions uucheck(1M) SARM
 directories in a path. mkdirp(3G) PRM
 directories. link, unlink: link(1M) SARM
 directories. pathfind: pathfind(3G) PRM
 directory. cd(1) URM
 directory. chdir(2) PRM
 directory. chroot(2) PRM
 directory. ls(1) BSD
 directory. ls(1) URM
 directory. ls(1) XNX
 directory. mkdir(2) PRM

mvdirc: move a	directory.	mvdirc(1M) SARM
rmdir: remove a	directory.	rmdir(2) PRM
scandir, alphasort: scan a	directory.	scandir(3) BSD
uucleanup: uucp spool	directory clean-up.	uucleanup(1M) SARM
dircmp:	directory comparison.	dircmp(1) URM
file. getdents: read	directory entries and put in a	getdents(2) PRM
dirent: file system independent	directory entry.	dirent(4) SARM
unlink: remove	directory entry.	unlink(2) PRM
chroot: change root	directory for a command.	chroot(1M) SARM
get pathname of current working	directory. getcwd:	getcwd(3C) PRM
pwd: working	directory name.	pwd(1) URM
dirname: report the parent	directory name of a file path/	dirname(3G) PRM
telldir, seekdir, rewinddir, /	directory: opendir, readdir,	directory(3C) PRM
seekdir, rewinddir, closedir:	directory operations. /telldir,	directory(3C) PRM
ordinary file. mknod: make a	directory, or a special or	mknod(2) PRM
ordinary file. mknod: make a	directory, or a special or	mknod(2) XNX
number of disk blocks used per	directory or file. /display the	du(1M) BSD
getwd: get current working	directory pathname.	getwd(3) BSD
restore file to original	directory. restore:	restore(1) XNX
ttyname. ttysrch:	directory search list for	ttysrch(4) SARM
whois: Internet user name	directory service.	whois(1) NUAG
whois: Internet user name	directory service.	whois(1) URM
ypfiles: the YP database and	directory structure.	ypfiles(4) NI
directory entry.	dirent: file system independent	dirent(4) SARM
path names. basename,	dirname: deliver portions of	basename(1) URM
directory name of a file path /	dirname: report the parent	dirname(3G) PRM
t_unbind:	dis: object code disassembler.	dis(1) PRM
printers. enable,	disable a transport endpoint.	t_unbind(3N) NI
acct: enable or	disable: enable/disable LP	enable(1) URM
Controller/ dma_disable:	disable process accounting.	acct(2) PRM
dis: object code	Disable requests on a DMA	dma_disable(D3D) DDRM
subchannel on a line/ ccidetach:	disassembler.	dis(1) PRM
connections. connld: line	disassociates this host with a	ccidetach(1M) MBRMAN
ccibunbind: unbinds a line	discipline for unique stream	connld(7) STRM
type, modes, speed, and line	discipline from a line on the/	ccibunbind(1M) MBRMAN
communications/ ccldinfo: line	discipline. getty: set terminal	getty(1M) SARM
communications/ ccldlist: line	discipline information on the	ccldinfo(1M) MBRMAN
standard STREAMS terminal line	discipline list on the	ccldlist(1M) MBRMAN
ccifree: frees a line	discipline module. ldterm:	ldterm(7) STRM
cciload: load a line	discipline program on the/	ccifree(1M) MBRMAN
ccibind: binds a line	discipline program on to the/	cciload(1M) MBRMAN
menus. /item_count: connect and	discipline to a line on the/	ccibind(1M) MBRMAN
t_snddis: send user-initiated	disconnect items to and from	menu_items(3X) CHAR
retrieve information from	disconnect request.	t_snddis(3N) NI
fd: diskette (floppy	disconnect. t_rcvdis:	t_rcvdis(3N) NI
hd: hard (fixed)	disk).	fd(7) SARM
	disk.	hd(7) SARM

fusage: disk access profiler. fusage(1M) NUAG
 fusage: disk access profiler. fusage(1M) SARM
 diskusg: generate disk accounting data by user ID. diskusg(1M) SARM
 (generic): report number of free disk blocks and files. df df(1M) SARM
 (generic): report number of free disk blocks and files. df df(1M) URM
 df (s5): report number of free disk blocks and i-nodes for s5/ df(1M) SARM
 or/ du: display the number of disk blocks used per directory du(1M) BSD
 iSBCR 214 peripheral controller disk device driver. i214: i214(1M) MBRMAN
 186/224A peripheral controller disk device driver. i224a: iSBC i224a(1M) MBRMAN
 fddd: MULTIBUS flexible disk device drivers. fddd(1M) MBRMAN
 hddd: MULTIBUS hard disk device drivers. hddd(1M) MBRMAN
 ramd: MULTIBUS RAM disk driver. ramd(1M) MBRMAN
 hdformat: low level hard disk formatter. hdformat(1M) MBRMAN
 mkpart: disk maintenance utility. mkpart(1M) SARM
 quota: display a user's disk quota and usage. quota(1M) SARM
 medium. disk: random access bulk storage disk(7) SARM
 /f4diskadd/fl: disk set up utility. diskadd(1M) SARM
 disksetup: disk set up utility. disksetup(1M) SARM
 df: report free disk space on file systems. df(1) BSD
 df (ufs): report free disk space on ufs file systems. df(1M) SARM
 space: disk space requirement file. space(4) SS
 du: summarize disk usage. du(1M) SARM
 du: summarize disk usage. du(1M) URM
 fd: diskette (floppy disk). fd(7) SARM
 load system dump from floppy diskettes. ldsysdump: ldsysdump(1M) SARM
 the system without checking the disks. /fasthalt: reboot/halt fastboot(1M) BSD
 filesystems, data partitions, or disks. /initiate restores of restore(1M) SARM
 accounting data by user ID. disksetup: disk set up utility. disksetup(1M) SARM
 administration. diskusg: generate disk diskusg(1M) SARM
 rt_dptbl: real-time dispatcher parameter table. rt_dptbl(4) SARM
 ts_dptbl: time-sharing dispatcher parameter table. ts_dptbl(4) SARM
 valid group names. dispgid: displays a list of all dispgid(1) SARM
 valid group names. dispgid: displays a list of all dispgid(1) SS
 display: system console display. display(7) SARM
 logged in. users: display a compact list of users users(1) BSD
 system console. print: display a driver message on print(D2DK) DDRM
 system console. fmtmsg: display a message on stderr or fmtmsg(1) URM
 system console. fmtmsg: display a message on stderr or fmtmsg(3C) PRM
 a keyword. whatis: display a one-line summary about whatis(1) BSD
 return an integer value. ckint: display a prompt; verify and ckint(1) SARM
 return an integer value. ckint: display a prompt; verify and ckint(1) SS
 return a pathname. ckpath: display a prompt; verify and ckpath(1) SARM
 return a pathname. ckpath: display a prompt; verify and ckpath(1) SS
 return a string answer. ckstr: display a prompt; verify and ckstr(1) SARM
 return a string answer. ckstr: display a prompt; verify and ckstr(1) SS
 return a time of day. cktime: display a prompt; verify and cktime(1) SARM

return a time of day. cktime: display a prompt; verify and cktime(1) SS
 usage. quota: display a user's disk quota and quota(1M) SARM
 memberships. groups: display a user's group groups(1) BSD
 panic the system. cmn_err: display an error message or cmn_err(D3DK) DDRM
 bkexcept: change or display an exception list. bkexcept(1M) SARM
 arp: address resolution display and control. arp(1M) NUAG
 arp: address resolution display and control. arp(1M) SARM
 alarms and/or the/ indicator: display application specific indicator(1F) CHAR
 /menu_pad: control menus display attributes. menu_attributes(3X) CHAR
 /field_pad: format the general display attributes of forms. form_field_attributes(3X) CHAR
 conflgs: change and display console flags. conflgs(1M) SARM
 for a text string in,/ srchtxt: display contents of, or search srchtxt(1) URM
 /pechochar: create and display curses pads. curs_pad(3X) CHAR
 vi: screen-oriented (visual) display editor based on ex. vi(1) URM
 currently set. printenv: display environment variables printenv(1) BSD
 /zcat: compress, expand or display expanded files. compress(1) URM
 format. hd: display files in hexadecimal hd(1) XNX
 files. head: display first few lines of head(1) URM
 ff (s5): display i-list information. ff(1M) SARM
 and remote users. finger: display information about local finger(1) NUAG
 and remote users. finger: display information about local finger(1) URM
 which: locate a command; display its pathname or alias. which(1) BSD
 count profile data. lprof: display line-by-line execution lprof(1) PRM
 information. dfmounts: display mounted NFS resource dfmounts(1M) NUAG
 information. dfmounts: display mounted NFS resource dfmounts(1M) SARM
 information. dfmounts: display mounted resource dfmounts(1M) NUAG
 information. dfmounts: display mounted resource dfmounts(1M) SARM
 information. rmntstat: display mounted resource rmntstat(1M) NUAG
 information. rmntstat: display mounted resource rmntstat(1M) SARM
 information. dfmounts: display mounted RFS resource dfmounts(1M) NUAG
 information. dfmounts: display mounted RFS resource dfmounts(1M) SARM
 nroff: format documents for display or line-printer. nroff(1) BSD
 basename: display portions of pathnames. basename(1) BSD
 prof: display profile data. prof(1) PRM
 find reference pages by/ man: display reference manual pages; man(1) BSD
 information. pkginfo: display software package pkginfo(1) SARM
 information. pkginfo: display software package pkginfo(1) SS
 display: system console display. display(7) SARM
 current host. arch: display the architecture of the arch(1) BSD
 register. bkgreg: change or display the contents of a backup bkgreg(1M) SARM
 history of an SCCS file. prt: display the delta and commentary prt(1) BSD
 username. whoami: display the effective current whoami(1) BSD
 at specified times. atq: display the jobs queued to run atq(1) URM
 blocks used per directory/ du: display the number of disk du(1M) BSD
 the current host. mach: display the processor type of mach(1) BSD
 jobs. lpq: display the queue of printer lpq(1) BSD
 memory. pagesize: display the size of a page of pagesize(1) BSD

operations. bkstatus: display the status of backup bkstatus(1M) SARM
 processes. ps: display the status of current ps(1) BSD
 group names. dispgid: displays a list of all valid dispgid(1) SARM
 group names. dispgid: displays a list of all valid dispgid(1) SS
 user names. dispuid: displays a list of all valid dispuid(1) SARM
 user names. dispuid: displays a list of all valid dispuid(1) SS
 values. pkgparam: displays package parameter pkgparam(1) SARM
 values. pkgparam: displays package parameter pkgparam(1) SS
 valid user names. dispuid: displays a list of all dispuid(1) SARM
 valid user names. dispuid: displays a list of all dispuid(1) SS
 hypot: Euclidean distance function. hypot(3M) PRM
 fstypes: file that registers distributed file system/ fstypes(4) NUAG
 fstypes: file that registers distributed file system/ fstypes(4) SARM
 /lcong48: generate uniformly distributed pseudo-random/ drand48(3C) PRM
 and remainder. div, ldiv: compute the quotient div(3C) PRM
 information. dlclose: close a shared object. dlclose(3X) PRM
 derror: get diagnostic derror(3X) PRM
 dlopen: open a shared object. dlopen(3X) PRM
 dlsym: get the address of a dlsym(3X) PRM
 symbol in shared object. DMA activity and release the dma_stop(D3D) DDRM
 channel. dma_stop: stop DMA Buffer Descriptor. dma_free_buf(D3D) DDRM
 dma_free_buf: Free storage for a DMA Buffer Descriptor. dma_get_buf(D3D) DDRM
 /Allocate storage for a DMA Command Block. dma_free_cb(D3D) DDRM
 dma_free_cb: Free storage for a DMA Command Block. dma_get_cb: .. dma_get_cb(D3D) DDRM
 Allocate storage for a DMA Controller channel. dma_disable(D3D) DDRM
 /Disable requests on a DMA Controller channel. dma_enable(D3D) DDRM
 dma_enable: Enable requests on a DMA device driver. dma(1M) MBRMAN
 dma: DMA device driver. dma(1M) MBRMAN
 DMA. dma_buf: data dma_buf(D4D) DDRM
 DMA. dma_cb: data dma_cb(D4D) DDRM
 dma_buf: data storage structure dma_buf(D4D) DDRM
 dma_cb: data storage structure dma_cb(D4D) DDRM
 dma_disable: Disable requests on dma_disable(D3D) DDRM
 dma_enable: Enable requests on a dma_enable(D3D) DDRM
 dma_free_buf: Free storage for a dma_free_buf(D3D) DDRM
 dma_free_cb: Free storage for a dma_free_cb(D3D) DDRM
 dma_get_best_mode: Confirm data dma_get_best_mode(D3D) DDRM
 dma_get_best_mode(D3D) DDRM
 dma_get_buf: Allocate storage dma_get_buf(D3D) DDRM
 dma_get_cb: Allocate storage for dma_get_cb(D3D) DDRM
 request into manageable units. dma_pageio: break up an I/O dma_pageio(D3DK) DDRM
 for a hardware request. dma_prog: Programming a Channel dma_prog(D3D) DDRM
 release the channel. dma_stop: stop DMA activity and dma_stop(D3D) DDRM
 for software request. dma_swsetup: Setting a channel dma_swsetup(D3D) DDRM
 Software-programmed Transfer. dma_swstart: to start a dma_swstart(D3D) DDRM
 object downloader for the 5620 DMD terminal. wtinit: wtinit(1M) SARM
 domain and network names. dname: print Remote File Sharing dname(1M) NUAG

Permuted Index

domain and network names. `dnname: print Remote File Sharing` `dnname(1M)` SARM
`/res_mkquery, res_send, res_init, dn_comp, dn_expand: resolver/` `resolver(3N)` NI
`/res_mkquery, res_send, res_init, dn_comp, dn_expand: resolver/` `resolver(3N)` NUAG
`/res_send, res_init, dn_comp, dn_expand: resolver routines.` `resolver(3N)` NI
`/res_send, res_init, dn_comp, dn_expand: resolver routines.` `resolver(3N)` NUAG
 configuration script. `doconfig: execute a` `doconfig(3N)` NI
`troff: typeset or format documents.` `troff(1)` BSD
`line-printer. nroff: format documents for display or` `nroff(1)` BSD
`from a host. gettable: get DoD Internet format host table` `gettable(1M)` NUAG
`from a host. gettable: get DoD Internet format host table` `gettable(1M)` SARM
`htable: convert DoD Internet format host table.` `htable(1M)` NUAG
`htable: convert DoD Internet format host table.` `htable(1M)` SARM
`nulladm,/ chargefee, ckpact, dodisk, lastlogin, monacct,` `acctsh(1M)` SARM
`is logged in, and what are they doing. w: who` `w(1)` BSD
`whodo: who is doing what.` `whodo(1M)` SARM
`rfadmin: Remote File Sharing domain administration.` `rfadmin(1M)` NUAG
`rfadmin: Remote File Sharing domain administration.` `rfadmin(1M)` SARM
`dnname: print Remote File Sharing domain and network names.` `dnname(1M)` NUAG
`dnname: print Remote File Sharing domain and network names.` `dnname(1M)` SARM
`name of current secure RPC domain. domainname: get/set` `domainname(1M)` NI
`address to hostname database or domain. ethers: Ethernet` `ethers(4)` NUAG
`address to hostname database or domain. ethers: Ethernet` `ethers(4)` SARM
`named, in.named: Internet domain name server.` `named(1M)` NUAG
`named, in.named: Internet domain name server.` `named(1M)` SARM
`current secure RPC domain. domainname: get/set name of` `domainname(1M)` NI
`strtod, atof: convert string to double-precision number.` `strtod(3C)` PRM
`/single_to_decimal, double_to_decimal,/` `floating_to_decimal(3)` BSD
`/refresh, wrefresh, wnoutrefresh, doupdate, redrawwin, wredrawn:/` `curs_refresh(3X)` CHAR
`in bytes to size in pages (round down). btop: convert size` `btop(D3DK)` DDRM
`library and object module/ download: loads 8086 absolute` `download(1M)` MBRMAN
`terminal. wtinit: object downloader for the 5620 DMD` `wtinit(1M)` SARM
`nrand48, mrand48, jrand48,/ drand48, erand48, lrand48,` `drand48(3C)` PRM
`bps: Bootstrap Parameter String driver.` `bps(1M)` MBRMAN
`console: console port device driver.` `console(1M)` MBRMAN
`d258: i82258 ADMA device driver.` `d258(1M)` MBRMAN
`dma: DMA device driver.` `dma(1M)` MBRMAN
`ics: Interconnect Space device driver.` `ics(1M)` MBRMAN
`mpc: MPC device driver.` `mpc(1M)` MBRMAN
`ramd: MULTIBUS RAM device driver.` `ramd(1M)` MBRMAN
`rci: debug console/rci protocol driver.` `rci(1M)` MBRMAN
`rtc: MULTIBUS clock driver.` `rtc(1M)` MBRMAN
`sad: STREAMS Administrative Driver.` `sad(7)` STRM
`sxt: pseudo-device driver.` `sxt(7)` SARM
`sxt: pseudo-device driver.` `sxt(7)` STRM
`controller disk device /186/224A peripheral` `i224a(1M)` MBRMAN
`controller tape device driver. /186/224A peripheral` `i224atp(1M)` MBRMAN
`communications interface STREAMS driver. /5 line asynchronous` `ports(7)` STRM

controller script device
 data from a user program to a
 controller interface device
 delete, update, or get device
 STREAMS-based multiplexed tty
 STREAMS-based multiplexed tty
 edlina: External Data Link
 peripheral controller device
 dual channel serial-port device
 Ethernet controller device
 console terminal/iSBXT 351
 asynchronous (terminal) device
 value/ module_info: STREAMS
 communications controller device
 controller tape device
 controller tape device
 controller disk device
 console. print: display a
 Message Passing Space device
 device pair on a STREAMS
 Transport Service (ots) device
 xtt: extract and print xt
 for a non-STREAMS character
 drv_priv: determine
 a SCSI block for the target
 xts: extract and print xt
 submit messages to the log
 and AT&T. mouse: mouse device
 copyout: copy data from a
 shutdown. halt: shutdown the
 write queue for this module or
 channels protocol used by xt
 channels protocol used by xt
 hddd: MULTIBUS hard disk device
 tdd: MULTIBUS tape device
 MULTIBUS flexible disk device
 /structure that specifies a
 state information.
 ticks to microseconds.
 privilege.
 microseconds to clock ticks.
 specified interval.
 blocks used per directory or/

 driver. i354: iSBX 354
 plot, aedplot, bgplot, crtplot,
 driver. /asynchronous terminal atcs(1M) MBRMAN
 driver buffer. copyin: copy copyin(D3DK) DDRM
 driver. cci: communications cci(1M) MBRMAN
 driver configuration data. /add, idinstall(1M) SARM
 driver for AT&T windowing. xt: xt(7) SARM
 driver for AT&T windowing. xt: xt(7) STRM
 driver for iNA961 Release 3.0.. edlina(1M) MBRMAN
 driver. i258: iSBC 386/258 i258(1M) MBRMAN
 driver. i354: iSBX 354 i354(1M) MBRMAN
 driver. i530: iSBC 186/530 i530(1M) MBRMAN
 driver. i8251: i8251(1M) MBRMAN
 driver. iasy: iasy(1M) MBRMAN
 driver identification and limit module_info(D4DK) DDRM
 driver. /iSBC 186/410 serial i410(1M) MBRMAN
 driver. /iSBC 214 peripheral i214tp(1M) MBRMAN
 driver. /iSBC 386/258 peripheral i258tp(1M) MBRMAN
 driver. /iSBCR 214 peripheral i214(1M) MBRMAN
 driver message on system print(D2DK) DDRM
 driver. mps: mps(1M) MBRMAN
 driver. /open any major/minor clone(7) STRM
 driver. ots: System V/386 OSI ots(1M) MBRMAN
 driver packet traces. xtt(1M) SARM
 driver. /poll entry point chpoll(D2DK) DDRM
 driver privilege. drv_priv(D3DK) DDRM
 driver. sdi_getblk: allocate sdi_getblk(D3I) SCSI
 driver statistics. xts(1M) SARM
 driver. strlog: strlog(D3DK) DDRM
 driver supporting Bus, serial, mouse(7) MOUSE
 driver to a user program. copyout(D3DK) DDRM
 driver when the system halt(D2D) DDRM
 driver. WR: get pointer to the WR(D3DK) DDRM
 driver. xtproto: multiplexed xtproto(5) PRM
 driver. xtproto: multiplexed xtproto(5) SARM
 drivers. hddd(1M) MBRMAN
 drivers. tdd(1M) MBRMAN
 drivers. fddd: fddd(1M) MBRMAN
 driver's message freeing/ free_rtn(D4DK) DDRM
 drv_getparm: retrieve kernel drv_getparm(D3DK) DDRM
 drv_hztousec: convert clock drv_hztousec(D3DK) DDRM
 drv_priv: determine driver drv_priv(D3DK) DDRM
 drv_usectohz: convert drv_usectohz(D3DK) DDRM
 drv_usecwait: busy-wait for drv_usecwait(D3DK) DDRM
 du: display the number of disk du(1M) BSD
 du: summarize disk usage. du(1M) SARM
 du: summarize disk usage. du(1M) URM
 dual channel serial-port device i354(1M) MBRMAN
 dumbplot, gigipplot, hpplot,/ plot(1G) BSD

od: octal
 ufsdump: incremental file system object file.
 ldsysdump: load system file. dump:
 zdump: time zone descriptor.
 descriptor. descriptor.
 form_field_new: new_field, dupmsg:
 descriptor. dupb:
 descriptor. dup2:
 descriptor. dup:
 /mvwin, subwin, derwin, mvderwin,
 ldd: list
 ld: link editor,
 form_field_info: field_info,
 echo:
 echo:
 curs_inopts: cbreak, nocbreak,
 output.
 /waddch, mvaddch, mvwaddch,
 hosts. ping: send ICMP
 hosts. ping: send ICMP
 seconvert, sfconvert,/
 floating-point number to/
 program. end, etext,
 for casual users).
 edquota:
 sact: print current SCCS file
 edsysadm: sysadm interface
 edsysadm: sysadm interface
 edvtoc: VTOC
 ed, red: text
 ex: text
 ld: link editor, dynamic link
 sed: stream
 screen-oriented (visual) display
 ld: link
 ld: link
 users). edit: text
 dump. dump(4) SARM
 dump. od(1) URM
 dump. ufsdump(1M) SARM
 dump: dump selected parts of an dump(1) PRM
 dump from floppy diskettes. ldsysdump(1M) SARM
 dump selected parts of an object dump(1) PRM
 dumper. zdump(1M) SARM
 dup: duplicate an open file dup(2) PRM
 dup2: duplicate an open file dup2(3C) PRM
 dupb: duplicate a message block dupb(D3DK) DDRM
 dup_field, link_field,/ form_field_new(3X) CHAR
 duplicate a message. dupmsg(D3DK) DDRM
 duplicate a message block dupb(D3DK) DDRM
 duplicate an open file dup2(3C) PRM
 duplicate an open file dup(2) PRM
 dupmsg: duplicate a message. dupmsg(D3DK) DDRM
 dupwin, wsyncup, syncok,/ curs_window(3X) CHAR
 dynamic dependencies. ldd(1) PRM
 dynamic link editor. ld(1) BSD
 dynamic_field_info: get forms/ form_field_info(3X) CHAR
 echo arguments. echo(1) BSD
 echo arguments. echo(1) URM
 echo: echo arguments. echo(1) BSD
 echo: echo arguments. echo(1) URM
 echo, noecho, halfdelay,/ curs_inopts(3X) CHAR
 echo: put string on virtual echo(1F) CHAR
 echochar, wechochar: add a/ curs_addch(3X) CHAR
 ECHO_REQUEST packets to network ping(1M) NUAG
 ECHO_REQUEST packets to network ping(1M) SARM
 econvert, fconvert, gconvert, econvert(3) BSD
 ecvt, fcvt, gcvt: convert ecvt(3C) PRM
 ed, red: text editor. ed(1) URM
 edata: last locations in end(3C) PRM
 edit: text editor (variant of ex edit(1) URM
 edit user quotas. edquota(1M) SARM
 editing activity. sact(1) PRM
 editing tool. edsysadm(1M) SARM
 editing tool. edsysadm(1M) SS
 editing utility. edvtoc(1M) SARM
 editor. ed(1) URM
 editor. ex(1) URM
 editor. ld(1) BSD
 editor. sed(1) URM
 editor based on ex. vi: vi(1) URM
 editor, dynamic link editor. ld(1) BSD
 editor for object files. ld(1) PRM
 editor (variant of ex for casual edit(1) URM

putdgrp: edits device group table. putdgrp(1) SARM
 putdev: edits device table. putdev(1) SARM
 driver for iNA961 Release 3.0.. edlina: External Data Link edlina(1M) MBRMAN
 edquota: edit user quotas. edquota(1M) SARM
 editing tool. edsysadm: sysadm interface edsysadm(1M) SARM
 editing tool. edsysadm: sysadm interface edsysadm(1M) SS
 edvtoc: VTOC editing utility. edvtoc(1M) SARM
 effective current username. whoami(1) BSD
 effective group IDs. setregid(3) BSD
 effective group IDs. /real user, getuid(2) PRM
 effective user IDs. setreuid(3) BSD
 effective user, real group, and/ getuid(2) PRM
 efficient flyby mode. dma_get_best_mode(D3D) DDRM
 efficient way. vfork: spawn vfork(2) PRM
 egrep: search a file for a egrep(1) URM
 egrep: search a file for a egrep(1) XNX
 element from a queue. insque(3C) PRM
 element of a path name. basename(3G) PRM
 ELF descriptor. elf_update(3E) PRM
 ELF (Executable and Linking a.out(4) PRM
 ELF library and application/ elf_version(3E) PRM
 elf: object file access library. elf(3E) PRM
 ELF object file translation. cof2elf(1) PRM
 elf32_fsize: return the size of elf_fsize(3E) PRM
 elf32_getehdr, elf32_newehdr: elf_getehdr(3E) PRM
 elf32_getphdr, elf32_newphdr: elf_getphdr(3E) PRM
 elf32_getshdr: retrieve elf_getshdr(3E) PRM
 elf32_newehdr: retrieve/ elf_getehdr(3E) PRM
 elf32_newphdr: retrieve/ elf_getphdr(3E) PRM
 elf32_xlatetof, elf32_xlatetom: elf_xlate(3E) PRM
 elf32_xlatetom: class-dependent elf_xlate(3E) PRM
 elf_begin: make a file elf_begin(3E) PRM
 elf_cntl: control a file elf_cntl(3E) PRM
 elf_end: finish using an object elf_end(3E) PRM
 elf_errmsg, elf_errno: error elf_error(3E) PRM
 elf_errno: error handling. elf_error(3E) PRM
 elf_fill: set fill byte. elf_fill(3E) PRM
 elf_flagdata, elf_flagehdr, elf_flag(3E) PRM
 elf_flagehdr, elf_flagelf, elf_flag(3E) PRM
 elf_flagelf, elf_flagphdr,/ elf_flag(3E) PRM
 elf_flagphdr, elf_flagdata, elf_flag(3E) PRM
 /elf_flagehdr, elf_flagelf, elf_flag(3E) PRM
 /elf_flagelf, elf_flagphdr, elf_flag(3E) PRM
 /elf_flagphdr, elf_flagscn, elf_flag(3E) PRM
 elf_flagshdr: manipulate flags. elf_flag(3E) PRM
 elf_fsize: elf32_fsize: return elf_fsize(3E) PRM
 elf_getarhdr: retrieve archive elf_getarhdr(3E) PRM
 elf_getarsym: retrieve archive elf_getarsym(3E) PRM
 elf_getbase: get the base offset elf_getbase(3E) PRM

elf_rawdata: get section data.	elf_getdata, elf_newdata,	elf_getdata(3E) PRM
elf32_newehdr: retrieve/	elf_getehdr: elf32_getehdr,	elf_getehdr(3E) PRM
identification data.	elf_getident: retrieve file	elf_getident(3E) PRM
elf32_newphdr: retrieve/	elf_getphdr: elf32_getphdr,	elf_getphdr(3E) PRM
elf_newscn, elf_nextscn: get/	elf_getscn, elf_ndxscn,	elf_getscn(3E) PRM
retrieve class-dependent/	elf_getshdr: elf32_getshdr:	elf_getshdr(3E) PRM
	elf_hash: compute hash value.	elf_hash(3E) PRM
	elf_kind: determine file type.	elf_kind(3E) PRM
elf_nextscn: get/ elf_getscn,	elf_ndxscn, elf_newscn,	elf_getscn(3E) PRM
section data. elf_getdata,	elf_newdata, elf_rawdata: get	elf_getdata(3E) PRM
section/ elf_getscn, elf_ndxscn,	elf_newscn, elf_nextscn: get	elf_getscn(3E) PRM
member access.	elf_next: sequential archive	elf_next(3E) PRM
/elf_ndxscn, elf_newscn,	elf_nextscn: get section/	elf_getscn(3E) PRM
access.	elf_rand: random archive member	elf_rand(3E) PRM
elf_getdata, elf_newdata,	elf_rawdata: get section data.	elf_getdata(3E) PRM
uninterpreted file contents.	elf_rawfile: retrieve	elf_rawfile(3E) PRM
pointer.	elf_strptr: make a string	elf_strptr(3E) PRM
descriptor.	elf_update: update an ELF	elf_update(3E) PRM
library and application/	elf_version: coordinate ELF	elf_version(3E) PRM
elf32_xlatetom: class-dependent/	elf_xlate: elf32_xlatetof,	elf_xlate(3E) PRM
nroff or/ soelim: resolve and	eliminate .so requests from	soelim(1) BSD
/tgoto, tputs: curses interfaces	(emulated) to the termcap/	termcap(3X) CHAR
ptem: STREAMS Pseudo Terminal	Emulation module.	ptem(7) STRM
qenable:	enable a queue.	qenable(D3DK) DDRM
LP printers.	enable, disable: enable/disable	enable(1) URM
accounting. acct:	enable or disable process	acct(2) PRM
Controller channel. dma_enable:	Enable requests on a DMA	dma_enable(D3D) DDRM
enable, disable:	enable/disable LP printers.	enable(1) URM
service.	enableok: reschedule a queue for	enableok(D3DK) DDRM
its ASCII/ uuencode, uudecode:	encode a binary file, or decode	uuencode(1C) URM
crypt:	encode/decode.	crypt(1) URM
crypt, setkey,	encrypt: generate encryption.	crypt(3C) PRM
whether a character buffer is	encrypted. isencrypt: determine	isencrypt(3G) PRM
crypt, setkey, encrypt: generate	encryption.	crypt(3C) PRM
crypt: password and file	encryption functions.	crypt(3X) PRM
chkey: change user	encryption key.	chkey(1) NI
chkey: change user	encryption key.	chkey(1) NUAG
makekey: generate	encryption key.	makekey(1) URM
locations in program.	end, etext, edata: last	end(3C) PRM
System (SCCS). scs: front	end for the Source Code Control	scs(1) BSD
/getgrgid, getgrnam, setgrent,	endgrent, fgetgrent: get group/	getgrent(3C) PRM
/gethostbyname, sethostent,	endhostent: get network host/	gethostent(3N) NI
/getnetbyname, setnetent,	endnetent: get network entry.	getnetent(3N) NI
t_close: close a transport	endpoint.	t_close(3N) NI
t_open: establish a transport	endpoint.	t_open(3N) NI
t_unbind: disable a transport	endpoint.	t_unbind(3N) NI
MULTIBUS II transport	endpoint.. /an asynchronous	mb2a_closeport(3tai) TAI

MULTIBUS II transport
 socket: create an
 /Gets specific asynchronous
 /Gets specific synchronous
 MULTIBUS II transport
 bind an address to a transport
 the current event on a transport
 manage options for a transport
 /getprotobyname, setprotoent,
 /getpwuid, getpwnam, setpwent,
 /getservbyname, setservent,
 getspent, getspsnam, setspent,
 getusershell, setusershell,
 /getutline, pututline, setutent,
 /pututxline, setutxent,
 curs_initscr: initscr, newterm,
 on an ethernet node's ethernet/
 ethernet controller.
 streamtab: STREAMS
 grpck: check group database
 pwck: check password database
 getdents: read directory
 nlist: get
 nlist: get
 logger: add
 putpwent: write password file
 unlink: remove directory
 getnetpath: get /etc/netconfig
 system independent directory
 manipulate password file
 utmp, wtmp: utmp and wtmp
 utmp, wtmp: utmp and wtmp
 utmpx, wtmpx: utmpx and wtmpx
 utmpx, wtmpx: utmpx and wtmpx
 getmntany: get mnttab file
 endnetent: get network
 network configuration database
 updwtmpx: access utmpx file
 getvfsany: get vfstab file
 manipulate shadow password file
 character driver. chpoll: poll
 write shadow password file
 fgetgrent: get group file
 endhostent: get network host
 endprotoent: get protocol
 endservent: get service
 utmpname: access utmp file
 endpoint.. /an asynchronous mb2a_openport(3tai) TAI
 endpoint for communication. socket(3N) NI
 endpoint information.. mb2a_getinfo(3tai) TAI
 endpoint information.. mb2s_getinfo(3tai) TAI
 endpoint.. /Opens a synchronous mb2s_openport(3tai) TAI
 endpoint. t_bind: t_bind(3N) NI
 endpoint. t_look: look at t_look(3N) NI
 endpoint. t_optmgmt: t_optmgmt(3N) NI
 endprotoent: get protocol entry. getprotoent(3N) NI
 endpwent, fgetpwent: manipulate/ getpwent(3C) PRM
 endservent: get service entry. getservent(3N) NI
 endspent, fgetspent, lckpword,/ getspent(3C) PRM
 endusershell: get legal user/ getusershell(3) BSD
 endutent, utmpname: access utmp/ getut(3C) PRM
 endutxent, utmpxname, getutmp,/ getutx(3C) PRM
 endwin, isendwin, set_term,/ curs_initscr(3X) CHAR
 enetinfo: provides information enetinfo(1M) MBRMAN
 enetload: resets and reloads the enetload(1M) MBRMAN
 entity declaration structure. streamtab(D4DK) DDRM
 entries. grpck(1M) BSD
 entries. pwck(1M) BSD
 entries and put in a file. getdents(2) PRM
 entries from name list. nlist(3E) PRM
 entries from symbol table. nlist(3) BSD
 entries to the system log. logger(1) BSD
 entry. putpwent(3C) PRM
 entry. unlink(2) PRM
 entry corresponding to NETPATH/ getnetpath(3N) NI
 entry. dirent: file dirent(4) SARM
 entry. /endpwent, fgetpwent: getpwent(3C) PRM
 entry formats. utmp(4) PRM
 entry formats. utmp(4) SARM
 entry formats. utmpx(4) PRM
 entry formats. utmpx(4) SARM
 entry. getmntent, getmntent(3C) PRM
 entry. /getnetbyname, setnetent, getnetent(3N) NI
 entry. getnetconfig: get getnetconfig(3N) NI
 entry. /getutmpx, updwtmp, getutx(3C) PRM
 entry. /getvfile, getvfsspec, getvfile(3C) PRM
 entry. /lckpword, ulckpword: getspent(3C) PRM
 entry point for a non-STREAMS chpoll(D2DK) DDRM
 entry. putsent: putsent(3C) PRM
 entry. /setgrent, endgrent, getgrent(3C) PRM
 entry. /sethostent, gethostent(3N) NI
 entry. /setprotoent, getprotoent(3N) NI
 entry. /setservent, getservent(3N) NI
 entry. /setutent, endutent, getut(3C) PRM

relogin: rename login
 relogin: rename login execution.
 user-preference variable files/

 environ: user environment.
 environ: user environment.
 environ: user environment.
 putenv: change or add value to profile: setting up an /fpsetsticky: IEEE floating-point execution. env: set /for the Framed Access Command getenv: return value for /termattrs, termname: curses performed for multi-user stop the Remote File Sharing stop the Remote File Sharing set. printenv: display set and unset local or global remove nroff, troff, tbl and remove nroff/troff, tbl, and character definitions for mathematics. definitions for eqn. mrand48, jrand48, / drand48, /post_form, unpost_form: write or /post_menu, unpost_menu: write or clrbuf: dlrbot,/ curs_clear: curs_termattrs: baudrate, complementary error function. complementary error / erf, prompts for and/ ckdate, prompts for and/ ckdate, for and validates a/ ckgid, for and validates a/ ckgid, geterror: return I/O error function. erf, erfc: error function and complementary elf_errmsg, elf_errno: t_rcvuderr: receive a unit data strclean: STREAMS strclean: STREAMS strerr: STREAMS
 entry to show current layer. relogin(1M) SARM
 entry to show current layer. relogin(1M) URM
 env: set environment for command env(1) URM
 .environ, .pref, .variables: environ(4) SARM
 environ: user environment. environ(5) NI
 environ: user environment. environ(5) PRM
 environ: user environment. environ(5) SARM
 environment. environ(5) NI
 environment. environ(5) PRM
 environment. environ(5) SARM
 environment. putenv(3C) PRM
 environment at login time. profile(4) SARM
 environment control. fpgetround(3C) PRM
 environment for command env(1) URM
 Environment Interface. face(1) URM
 environment name. getenv(3C) PRM
 environment query routines. curs_termattrs(3X) CHAR
 environment. rc2: run commands rc2(1M) SARM
 environment. rfstop: rfstop(1M) NUAG
 environment. rfstop: rfstop(1M) SARM
 environment variables currently printenv(1) BSD
 environment variables. /unset: set(1F) CHAR
 eqn constructs. deroff: deroff(1) BSD
 eqn constructs. deroff: deroff(1) URM
 eqn. eqnchar: special eqnchar(7) BSD
 eqn, neqn, checkeq: typeset eqn(1) BSD
 eqnchar: special character eqnchar(7) BSD
 erand48, lrand48, nrand48, drand48(3C) PRM
 erase forms from associated/ form_post(3X) CHAR
 erase menus from associated/ menu_post(3X) CHAR
 erase the contents of a buffer. clrbuf(D3DK) DDRM
 erase, werase, clear, wclear, curs_clear(3X) CHAR
 erasechar, has_ic, has_il, / curs_termattrs(3X) CHAR
 erf, erfc: error function and erf(3M) PRM
 erfc: error function and erf(3M) PRM
 errdate, helpdate, valdate: ckdate(1) SARM
 errdate, helpdate, valdate: ckdate(1) SS
 errgid, helpgid, valgid: prompts ckgid(1) SARM
 errgid, helpgid, valgid: prompts ckgid(1) SS
 error. geterror(D3DK) DDRM
 error function and complementary erf(3M) PRM
 error function. erf, erfc: erf(3M) PRM
 error handling. elf_error(3E) PRM
 error indication. t_rcvuderr(3N) NI
 error logger cleanup program. strclean(1M) SARM
 error logger cleanup program. strclean(1M) STRM
 error logger daemon. strerr(1M) SARM

strerr: STREAMS error logger daemon. strerr(1M) STRM
 log: interface to STREAMS error logging and event tracing. log(7) STRM
 t_error: produce error message. t_error(3N) NI
 system. cmn_err: display an error message or panic the cmn_err(D3DK) DDRM
 strerror: get error message string. strerror(3C) PRM
 perror: print system error messages. perror(3C) PRM
 introduction to system calls and error numbers. intro: intro(2) PRM
 matherr: error-handling function. matherr(3M) PRM
 input files; report possible errors. /check nroff and troff checknr(1) BSD
 hashcheck: find spelling errors. /hashmake, spellin, spell(1) URM
 side remote procedure call errors. /routines for server rpc_svc_err(3N) NI
 block using a shared buffer. esballoc: allocate a message esballoc(D3DK) DDRM
 buffer is available. esbcall: call function when esbcall(D3DK) DDRM
 another transport/ t_connect: establish a connection with t_connect(3N) NI
 t_open: establish a transport endpoint. t_open(3N) NI
 line connection. dial: establish an outgoing terminal dial(3C) NI
 setmnt: establish mount table. setmnt(1M) SARM
 characteristics. tset, reset: establish or restore terminal tset(1) BSD
 corresponding/ getnetpath: get /etc/netconfig entry getnetpath(3N) NI
 program. end, etext, edata: last locations in end(3C) PRM
 operations. ethers: Ethernet address mapping ethers(3N) NI
 operations. ethers: Ethernet address mapping ethers(3N) NUAG
 database or domain. ethers: Ethernet address to hostname ethers(4) NUAG
 database or domain. ethers: Ethernet address to hostname ethers(4) SARM
 /on an ethernet node's ethernet controller. enetinfo(1M) MBRMAN
 enetload: resets and reloads the ethernet controller. enetload(1M) MBRMAN
 driver. i530: iSBC 186/530 Ethernet controller device i530(1M) MBRMAN
 /provides information on an ethernet node's ethernet/ enetinfo(1M) MBRMAN
 operations. ethers: Ethernet address mapping ethers(3N) NI
 operations. ethers: Ethernet address mapping ethers(3N) NUAG
 hostname database or domain. ethers: Ethernet address to ethers(4) NUAG
 hostname database or domain. ethers: Ethernet address to ethers(4) SARM
 hypot: Euclidean distance function. hypot(3M) PRM
 expression. expr: evaluate arguments as an expr(1) URM
 expression. fmlexpr: evaluate arguments as an fmlexpr(1F) CHAR
 test: condition evaluation command. test(1) BSD
 test: condition evaluation command. test(1F) CHAR
 test: condition evaluation command. test(1) URM
 inform a process that an event has occurred. pollwakeup: pollwakeup(D3DK) DDRM
 t_look: look at the current event on a transport endpoint. t_look(3N) NI
 activity pending execution of an event. sleep: suspend process sleep(D3DK) DDRM
 to STREAMS error logging and event tracing. log: interface log(7) STRM
 edit: text editor (variant of ex for casual users). edit(1) URM
 (visual) display editor based on ex: text editor. ex(1) URM
 cscope: interactively examine a C program. cscope(1) PRM
 sigprocmask: change or examine signal mask. sigprocmask(2) PRM

Permuted Index

and pending. sigpending: examine signals that are blocked sigpending(2) PRM
 crash: examine system images. crash(1M) SARM
 bkexcept: change or display an exception list. bkexcept(1M) SARM
 ieee_handler: IEEE exception trap handler function. ieee_handler(3M) BSD
 devfree: release devices from exclusive use. devfree(1M) SARM
 devreserv: reserves devices for exclusive use. devreserv(1M) SARM
 execve, execlp, execvp: execute/ exec: execute a/ exec: execute a/ exec: execl, execv, /execl, execv, execl, execve, run: run an files. a.out: ELF
 Command Environment/ face: executable for the Framed Access face(1) URM
 doconfig: execute a configuration script. doconfig(3N) NI
 execl, execve, execlp, execvp: execute a file. /execl, execv, exec(2) PRM
 specified length of/ timeout: execute a function after a timeout(D3DK) DDRM
 construct argument list(s) and execute command. xargs: xargs(1) URM
 time. at, batch: execute commands at a later at(1) URM
 regcmp, regex: compile and execute regular expression. regcmp(3G) PRM
 uuxqt: execute remote command requests. uuxqt(1M) SARM
 lastcomm: show the last commands executed, in reverse order. lastcomm(1) BSD
 env: set environment for command execution. env(1) URM
 uux: UNIX-to-UNIX system command execution. uux(1C) URM
 wakeup: resume suspended process execution. wakeup(D3DK) DDRM
 lprof: display line-by-line execution count profile data. lprof(1) PRM
 nap: suspend execution for a short interval. nap(2) NXN
 of clock/ delay: delay process execution for a specified number delay(D3DK) DDRM
 sleep: suspend sleep: suspend execution for an interval. sleep(1) URM
 sleep: suspend sleep: suspend execution for interval. sleep(3) BSD
 microseconds. usleep: suspend execution for interval. usleep(3) PRM
 suspend process activity pending execution of an event. sleep: sleep(D3DK) DDRM
 monitor: prepare execution profile. monitor(3C) PRM
 rexecd: remote execution server. rexecd(1M) NUAG
 rexecd: remote execution server. rexecd(1M) SARM
 profil: execution time profile. profil(2) PRM
 execvp: execute a/ exec: execl, execv, execl, execve, execlp, exec(2) PRM
 a/ exec: execl, execv, execl, execve, execlp, execvp: execute exec(2) PRM
 execv, execl, execve, execlp, execvp: execute a file. /execl, exec(2) PRM
 tunefs: tune up an existing file system. tunefs(1M) SARM
 create a new file or rewrite an existing one. creat: creat(2) PRM
 exit, _exit: terminate process. exit(2) PRM
 _log10, log10f, pow, powf, sqrt, _exit: terminate process. exit(2) PRM
 refer: from a bibliographic/ refer: exp, expf, cbrt, log, logf, exp(3M) PRM
 pack, pcat, unpack: compress and expand and insert references refer(1) BSD
 /uncompress, zcat: compress, expand files. pack(1) URM
 expand or display expanded/ compress(1) URM

compress, expand or display
 connection. `t_snd`: send data or
 `t_rcv`: receive data or
 `log10f, pow, powf, sqrt, / exp,`
 `/log10f, pow, powf, sqrt, sqrtf:`
 expression.
 expr: evaluate arguments as an
 `regcmp`: regular
 /compile, step, advance: regular
 /compile, step, advance: regular
 /compile, step, advance: regular
 evaluate arguments as an
 regex, re_comp, re_exec: regular
 compile and execute regular
 for a pattern using full regular
 for a pattern using full regular
 source files.
 `addbib`: create or
 interface. `termiox`:
 /float_to_decimal,
 iNA961 Release 3.0.. `edlina`:
 `xdr`: library routines for
 /xdr_setpos: library routines for
 /library routines for
 stream/ /library routines for
 /xdr_void: library routines for
 statistics. `xts`:
 packet traces. `xtt`:
 files. `exstr`:
 utility.
 /ceilf, copysign, fmod, fmodf,
 /copysign, fmod, fmodf, fabs,
 pathalias: alias file for
 Access Command Environment/
 information. `.ott`:
 variable files for AT&T
 `sigvec`: software signal
 simplified software signal
 inter-process communication
 of a number.
 factor: obtain the prime
 `loginlog`: log of
 true,
 inet: Internet protocol
 inet: Internet protocol
 data in a machine-independent
 the system without checking the/
 expanded files. `/zcat`:
 expedited data over a
 expedited data sent over a/
 `expf, cbrt, log, logf, log10,`
 exponential, logarithm, power,/
 expr: evaluate arguments as an
 expression.
 expression compile.
 expression compile and match/
 expression compile and match/
 expression compile and match/
 expression. `fmlexpr`:
 expression handler.
 expression. `regcmp, regex`:
 expressions. /search a file
 expressions. /search a file
 `exstr`: extract strings from
 extend a bibliographic database.
 extended general terminal
 extended_to_decimal: convert/
 External Data Link driver for
 external data representation.
 external data representation.
 external data representation.
 external data representation
 external data representation.
 extract and print xt driver
 extract and print xt driver
 extract strings from source
 /f4diskadd/fl: disk set up
 fabs, fabsf, rint, remainder:/
 fabsf, rint, remainder: floor,/
 FACE.
 face: executable for the Framed
 FACE object architecture
 FACE. /user-preference
 facilities.
 facilities. signal:
 facilities status. `ipcs`: report
 factor: obtain the prime factors
 factors of a number.
 failed login attempts.
 false: provide truth values.
 family.
 family.
 fashion. /access long integer
 fastboot, fasthalt: reboot/halt

without checking the/ fastboot,
file descriptor to an object.
 directory. chdir,
 chmod,
of a file. chown, lchown,
 stream.

sfconvert, sgconvert:/ econvert,
floating-point number to/ ecvt,
 device drivers.
STREAMS-based file descriptor.
STREAMS-based file descriptor.
 fopen, freopen,
 fopen, freopen,
full file system archive.
status inquiries. ferror,
stream status inquiries.
dbm, dbmunit, dbmclose,
dbm: dbmunit, dbmclose,
 head: display first
and statistics for a file/
 information.
statistics for a ufs file/
a full file system archive.
 fclose,
bstring: bcopy, bcmp, bzero,
 word from a/ getc, getchar,
/getgrnam, setgrent, endgrent,
pointer in a stream. fsetpos,
/getpwnam, setpwent, endpwent,
 stream. gets,
/getspnam, setspent, endspent,
 character string.
 character string.
set_max_field: set and get forms
/dynamic_field_info: get forms
/field_type, field_arg: forms
set forms current page and
or/ /data_behind: tell if forms
ics_hostid: returns the host id
 /field_opts: forms
reset: reset the current form

fasthalt: reboot/halt the system fastboot(1M) BSD
fattach: attach a STREAMS-based fattach(3C) STRM
fchdir: change working chdir(2) PRM
fchmod: change mode of file. chmod(2) PRM
fchown: change owner and group chown(2) PRM
fclose, fflush: close or flush a fclose(3S) PRM
fcntl: file control. fcntl(2) PRM
fcntl: file control options. fcntl(5) PRM
fcntl: file control options. fcntl(5) SARM
fconvert, gconvert, seconvert, econvert(3) BSD
fcvt, gcvt: convert ecvt(3C) PRM
fd: diskette (floppy disk). fd(7) SARM
fddd: MULTIBUS flexible disk fddd(1M) MBRMAN
fdetach: detach a name from a fdetach(1M) STRM
fdetach: detach a name from a fdetach(3C) STRM
fdopen: open a stream. fopen(3S) BSD
fdopen: open a stream. fopen(3S) PRM
fdp: create, or restore from, a fdp(1M) SARM
feof, clearerr, fileno: stream ferror(3S) PRM
ferror, feof, clearerr, fileno: ferror(3S) PRM
fetch, store, delete, firstkey,/ dbm(3) NI
fetch, store, delete, firstkey,/ dbm(3X) BSD
few lines of files. head(1) URM
ff (generic): list file names ff(1M) SARM
ff (s5): display i-list ff(1M) SARM
ff (ufs): list file names and ff(1M) SARM
ffile: create, or restore from, ffile(1M) SARM
fflush: close or flush a stream. fclose(3S) PRM
ffs: bit and byte string/ bstring(3) BSD
ffs: find first set bit. ffs(3C) PRM
fgetc, getw: get character or getc(3S) PRM
fgetgrent: get group file entry. getgrent(3C) PRM
fgetpos: reposition a file fsetpos(3C) PRM
fgetpwent: manipulate password/ getpwent(3C) PRM
fgets: get a string from a gets(3S) PRM
fgetspent, lckpwwdf, ulckpwwdf:/ getspent(3C) PRM
fgrep: search a file for a fgrep(1) URM
fgrep: search a file for a fgrep(1) XNX
field attributes. /field_status, form_field_buffer(3X) CHAR
field characteristics. form_field_info(3X) CHAR
field data type validation. form_field_validation(3X) CHAR
field. /field_index: form_page(3X) CHAR
field has off-screen data ahead form_data(3X) CHAR
field of the HOST ID record in/ ics_hostid(D3D) DDRM
field option routines. form_field_opts(3X) CHAR
field to its default values. reset(1F) CHAR

/set_field_type, field_type, field_arg: forms field data type/
 form_field_validation(3X) CHAR
 /field_fore, set_field_back, field_back, set_field_pad,/ form_field_attributes(3X) CHAR
 field_status,/ /set_field_buffer, field_buffer, set_field_status, form_field_buffer(3X) CHAR
 /set_form_fields, form_fields, field_count, move_field: connect/ form_field(3X) CHAR
 field_back,/ /set_field_fore, field_fore, set_field_back, form_field_attributes(3X) CHAR
 page and field. /current_field, field_index: set forms current form_page(3X) CHAR
 get forms/ form_field_info, field_info, dynamic_field_info: form_field_info(3X) CHAR
 /form_term, set_field_init, field_init, set_field_term,/ form_hook(3X) CHAR
 appearance of/ /set_field_just, field_just: format the general form_field_just(3X) CHAR
 /field_opts_on, field_opts_off, field_opts: forms field option/ form_field_opts(3X) CHAR
 /set_field_opts, field_opts_on, field_opts_off, field_opts:/ form_field_opts(3X) CHAR
 field_opts:/ /set_field_opts, field_opts_on, field_opts_off, form_field_opts(3X) CHAR
 /field_back, set_field_pad, field_pad: format the general/ .. form_field_attributes(3X) CHAR
 bufsplit: split buffer into fields. bufsplit(3G) PRM
 create and destroy forms fields. /free field,: form_field_new(3X) CHAR
 cut: cut out selected fields of each line of a file. cut(1) URM
 fmlcut: cut out selected fields of each line of a file. fmlcut(1F) CHAR
 field_count, move_field: connect fields to forms. /form fields, form_field(3X) CHAR
 /field_buffer, set_field_status, field_status, set_max_field: set/ form_field_buffer(3X) CHAR
 /field_init, set_field_term, field_term: assign/ form_hook(3X) CHAR
 field data type/ /set_field_type, field_type, field_arg: forms form_field_validation(3X) CHAR
 /link_fielddtype: forms fieldtype routines. form_fielddtype(3X) CHAR
 application/ /set_field_userptr, field_userptr: associate form_field_userptr(3X) CHAR
 mkfifo: create a new FIFO. mkfifo(3C) PRM
 mkfifo: make FIFO special file. mkfifo(1M) SARM
 admin: installation defaults file. admin(4) SARM
 archives: device header file. archives(4) SARM
 chmod, fchmod: change mode of file. chmod(2) PRM
 chsize: change the size of a file. chsize(2) XNX
 compver: compatible versions file. compver(4) SS
 copyright: copyright information file. copyright(4) SS
 core: core image file. core(4) PRM
 core: core image file. core(4) SARM
 crontab: user crontab file. crontab(1) URM
 dd: convert and copy a file. dd(1M) SARM
 dd: convert and copy a file. dd(1M) URM
 elf_end: finish using an object file. elf_end(3E) PRM
 get: get a version of an SCCS file. get(1) PRM
 group: group file. group(4) SARM
 issue: issue identification file. issue(4) SARM
 link: link to a file. link(2) PRM
 login: login default file. login(4) SARM
 makedbm: make a YP dbm file. makedbm(1M) NI
 mkfifo: make FIFO special file. mkfifo(1M) SARM
 mknod: make a special file. mknod(1M) SARM
 nm: print name list of an object file. nm(1) PRM

Permuted Index

null: the null	file.	null(7) SARM
passwd: password	file.	passwd(4) SARM
pkginfo: package characteristics	file.	pkginfo(4) SS
pkgproto: generate a prototype	file.	pkgproto(1) SS
prototype: package information	file.	prototype(4) SS
prs: print an SCCS	file.	prs(1) PRM
read: read from	file.	read(2) PRM
reinit: runs an initialization	file.	reinit(1F) CHAR
remove: remove	file.	remove(3C) PRM
rename: change the name of a	file.	rename(2) PRM
scsfile: format of SCCS	file.	scsfile(4) PRM
scsfile: format of SCCS	file.	scsfile(4) SARM
shadow: shadow password	file.	shadow(4) SARM
space: disk space requirement	file.	space(4) SS
sum: calculate a checksum for a	file.	sum(1) BSD
tail: deliver the last part of a	file.	tail(1) URM
term: format of compiled term	file.	term(4) CHAR
term: format of compiled term	file.	term(4) SARM
tmpfile: create a temporary	file.	tmpfile(3S) PRM
uniq: report repeated lines in a	file.	uniq(1) URM
val: validate an SCCS	file.	val(1) PRM
write, writev: write on a	file.	write(2) PRM
determine accessibility of a	file. access:	access(2) PRM
times. utime: set	file access and modification	utime(2) PRM
elf: object	file access library.	elf(3E) PRM
tar: tape	file archiver.	tar(1) URM
cpio: copy	file archives in and out.	cpio(1) URM
volumes. cpout: copy out	file archives to multiple	cpout(1M) MBRMAN
pwck, grpck: password/group	file checkers.	pwck(1M) SARM
change the group ownership of a	file. chgrp:	chgrp(1) URM
change owner and group of a	file. chown, lchown, fchown:	chown(2) PRM
diff: differential	file comparator.	diff(1) URM
diff3: 3-way differential	file comparison.	diff3(1) URM
sharing resources. dfstab:	file containing commands for	dfstab(4) NUAG
sharing resources. dfstab:	file containing commands for	dfstab(4) SARM
settings information/ ttydefs:	file contains terminal line	ttydefs(4) NI
retrieve uninterpreted	file contents. elf_rawfile:	elf_rawfile(3E) PRM
fcntl:	file control.	fcntl(2) PRM
fcntl:	file control options.	fcntl(5) PRM
fcntl:	file control options.	fcntl(5) SARM
rcp: remote	file copy.	rcp(1) NUAG
rcp: remote	file copy.	rcp(1) URM
public UNIX-to-UNIX system	file copy. uuto, uupick:	uuto(1C) URM
umask: set and get	file creation mask.	umask(2) PRM
fields of each line of a	file. cut: cut out selected	cut(1) URM
make a delta (change) to an SCCS	file. delta:	delta(1) PRM
close: close a	file descriptor.	close(2) PRM

dup2: duplicate an open	file descriptor. dup2(3C) PRM
dup: duplicate an open	file descriptor. dup(2) PRM
elf_begin: make a	file descriptor. elf_begin(3E) PRM
elf_cntl: control a	file descriptor. elf_cntl(3E) PRM
isastream: test a	file descriptor. isastream(3C) STRM
a name from a STREAMS-based	file descriptor. /detach fdetach(1M) STRM
a name from a STREAMS-based	file descriptor. /detach fdetach(3C) STRM
/dev/fd:	file descriptor files. fd(4) SARM
fattach: attach a STREAMS-based	file descriptor to an object. fattach(3C) STRM
	file: determine file type. file(1) URM
sdiff: print	file differences side-by-side. sdiff(1) URM
commentary history of an SCCS	file. /display the delta and prt(1) BSD
dump selected parts of an object	file. dump: dump(1) PRM
sact: print current SCCS	file editing activity. sact(1) PRM
the base offset for an object	file. elf_getbase: get elf_getbase(3E) PRM
crypt: password and	file encryption functions. crypt(3X) PRM
getmntent, getmntany: get mnttab	file entry. getmntent(3C) PRM
putpwent: write password	file entry. putpwent(3C) PRM
putspent: write shadow password	file entry. putspent(3C) PRM
updwtmp, updwtmpx: access utmpx	file entry. /getutmp, getutmpx, getutx(3C) PRM
getvfsany: get vfstab	file entry. /getvfsspec, getvfsent(3C) PRM
manipulate shadow password	file entry. /lckpwwdf, ulckpwwdf: getsptent(3C) PRM
endgrent, fgetgrent: get group	file entry. /setgrent, getgrent(3C) PRM
fgetpwent: manipulate password	file entry. /setpwent, endpwent, getpwent(3C) PRM
endutent, utmpname: access utmp	file entry. /setutent, getut(3C) PRM
execlp, execvp: execute a	file. /execv, execl, execve, exec(2) PRM
fields of each line of a	file. fmlcut: cut out selected fmlcut(1F) CHAR
fgrep: search a	file for a character string. fgrep(1) URM
fgrep: search a	file for a character string. fgrep(1) XNX
fmlgrep: search a	file for a pattern. fmlgrep(1F) CHAR
grep: search a	file for a pattern. grep(1) URM
grep: search a	file for a pattern. grep(1) XNX
regular/ egrep: search a	file for a pattern using full egrep(1) URM
regular/ egrep: search a	file for a pattern using full egrep(1) XNX
pathalias: alias	file for FACE. pathalias(4) SARM
netrc:	file for ftp remote login data. netrc(4) NUAG
netrc:	file for ftp remote login data. netrc(4) SARM
constants. limits: header	file for implementation-specific limits(4) PRM
constants. limits: header	file for implementation-specific limits(4) SARM
resolv.conf: configuration	file for name server routines. resolv.conf(4) NUAG
resolv.conf: configuration	file for name server routines. resolv.conf(4) SARM
strcf: STREAMS Configuration	File for STREAMS TCP/IP. strcf(4) NUAG
strcf: STREAMS Configuration	File for STREAMS TCP/IP. strcf(4) SARM
unistd: header	file for symbolic constants. unistd(4) SARM
syslog.conf: configuration	file for syslogd system log/ syslog.conf(4) BSD
ctags: create a tags	file for use with vi. ctags(1) URM
updaters: configuration	file for YP updating. updaters(4) NI

acct: per-process accounting
 ar: archive
 ar: archive
 mdevice:
 mfsys:
 mtune:
 sdevice:
 sfsys:
 stune:
 MULTIBUS II configuration
 pnch:
 intro: introduction to
 intro: introduction to
 removef: remove a
 directory entries and put in a
 readfile, longline: reads
 retrieve class-dependent object
 files. filehdr:
 elf_getident: retrieve
 pathfind: search for named
 copylist: copy a
 split: split a
 versions of a troff input
 the comment section of an object
 or a special or ordinary
 or a special or ordinary
 chmod: change
 browse or page through a text
 mkstemp: make a unique
 mktemp: make a unique
 realpath: returns the real
 ctermid: generate
 ufs file system. ff (ufs): list
 data base for the mail aliases
 change the format of a text
 /find the slot in the utmp
 printable strings in an object
 /uuencode: encode a binary
 identify processes using a
 creat: create a new
 chown: change
 chown: change
 the parent directory name of a
 /correct or initialize XENIX
 fixperm: correct or initialize
 pg:
 package contents description
 file format. acct(4) SARM
 file format. ar(4) PRM
 file format. ar(4) SARM
 file format.. mdevice(4) SARM
 file format.. mfsys(4) SARM
 file format.. mtune(4) SARM
 file format.. sdevice(4) SARM
 file format.. sfsys(4) SARM
 file format.. stune(4) SARM
 file format. config: config(1M) MBRMAN
 file format for card images. pnch(4) SARM
 file formats. intro(4) PRM
 file formats. intro(4) SARM
 file from software database. removef(1M) SS
 file. getdents: read getdents(2) PRM
 file, gets longest line. readfile(1F) CHAR
 file header. /elf32_newehdr: elf_getehdr(3E) PRM
 file header for common object filehdr(4) SARM
 file identification data. elf_getident(3E) PRM
 file in named directories. pathfind(3G) PRM
 file into memory. copylist(3G) PRM
 file into pieces. split(1) URM
 file. /mark differences between diffmk(1) BSD
 file. mcs: manipulate mcs(1) PRM
 file. mknod: make a directory, mknod(2) PRM
 file. mknod: make a directory, mknod(2) XNX
 file mode. chmod(1) URM
 file. more, page: more(1) URM
 file name. mkstemp(3) BSD
 file name. mktemp(3C) PRM
 file name. realpath(3C) PRM
 file name for terminal. ctermid(3S) PRM
 file names and statistics for a ff(1M) SARM
 file. newaliases: rebuild the newaliases(1M) BSD
 file. newform: newform(1) URM
 file of the current user. ttyslot(3C) PRM
 file or binary. strings: find strings(1) URM
 file, or decode its ASCII/ uuencode(1C) URM
 file or file structure. fuser: fuser(1M) SARM
 file or rewrite an existing one. creat(2) PRM
 file owner. chown(1) BSD
 file owner. chown(1) URM
 file path name. dirname: report dirname(3G) PRM
 file permissions and ownership. fixperm(1M) SARM
 file permissions and ownership. fixperm(1) XNX
 file perusal filter for CRTs. pg(1) URM
 file. pkgmap: pkgmap(4) SS

lseek: move read/write
 /rewind, ftell: reposition a
 fsetpos, fgetpos: reposition a
 locking: lock or unlock a
 File Sharing name server master
 File Sharing name server master
 remove a delta from an SCCS
 files or subsequent lines of one
 bfs: big
 compare two versions of an SCCS
 a curses screen from (to) a
 rfstart: start Remote
 rfstart: start Remote
 rfudaemon: Remote
 rfudaemon: Remote
 administration. rfadmin: Remote
 administration. rfadmin: Remote
 names. dname: print Remote
 names. dname: print Remote
 rfstop: stop the Remote
 rfstop: stop the Remote
 rfpasswd: change Remote
 rfpasswd: change Remote
 file. rfmaster: Remote
 file. rfmaster: Remote
 nsquery: Remote
 nsquery: Remote
 script. rfadmin: Remote
 script. rfadmin: Remote
 /mount, unmount Remote
 /mount, unmount Remote
 mapping. idload: Remote
 mapping. idload: Remote
 stat, lstat, fstat: get
 stat, lstat, fstat: get
 processes using a file or
 checksum and block count of a
 make a symbolic link to a
 checkfsys: check a
 makesys: create a
 mkfs (s5): construct an s5
 mount (s5): mount an s5
 mount: mount a
 newfs: construct a new
 /proc: process
 repquota: summarize quotas for a
 tunefs: tune up an existing
 file pointer. lseek(2) PRM
 file pointer in a stream. fseek(3S) PRM
 file pointer in a stream. fsetpos(3C) PRM
 file region for reading or/ locking(2) XNX
 file. rfmaster: Remote rfmaster(4) NUAG
 file. rfmaster: Remote rfmaster(4) SARM
 file. rmdel: rmdel(1) PRM
 file. /same lines of several paste(1) URM
 file scanner. bfs(1) URM
 file. sccsdiff: sccsdiff(1) PRM
 file. /scr_set: read (write) curs_scr_dump(3X) CHAR
 File Sharing. rfstart(1M) NUAG
 File Sharing. rfstart(1M) SARM
 File Sharing daemon process. rfudaemon(1M) NUAG
 File Sharing daemon process. rfudaemon(1M) SARM
 File Sharing domain rfadmin(1M) NUAG
 File Sharing domain rfadmin(1M) SARM
 File Sharing domain and network dname(1M) NUAG
 File Sharing domain and network dname(1M) SARM
 File Sharing environment. rfstop(1M) NUAG
 File Sharing environment. rfstop(1M) SARM
 File Sharing host password. rfpasswd(1M) NUAG
 File Sharing host password. rfpasswd(1M) SARM
 File Sharing name server master rfmaster(4) NUAG
 File Sharing name server master rfmaster(4) SARM
 File Sharing name server query. nsquery(1M) NUAG
 File Sharing name server query. nsquery(1M) SARM
 File Sharing notification shell rfadmin(1M) NUAG
 File Sharing notification shell rfadmin(1M) SARM
 File Sharing resources. rmountall(1M) NUAG
 File Sharing resources. rmountall(1M) SARM
 File Sharing user and group idload(1M) NUAG
 File Sharing user and group idload(1M) SARM
 file status. stat(2) PRM
 file status. stat(2) XNX
 file structure. fuser: identify fuser(1M) SARM
 file. sum: print sum(1) URM
 file. symlink: symlink(2) PRM
 file system. checkfsys(1M) SARM
 file system. makesys(1M) SARM
 file system. mkfs(1M) SARM
 file system. mount(1M) SARM
 file system. mount(2) PRM
 file system. newfs(1M) BSD
 file system. proc(4) SARM
 file system. repquota(1M) SARM
 file system. tunefs(1M) SARM

Permuted Index

umount: unmount a file system. umount(2) PRM
 create, or restore from, a full file system archive. fdp: fdp(1M) SARM
 create, or restore from, a full file system archive. ffile: ffile(1M) SARM
 ckbupscd: check file system backup schedule. ckbupscd(1M) SARM
 fsba: file system block analyzer. fsba(1M) SARM
 and interactive/ fsck (ufs): file system consistency check fsck(1M) SARM
 fsdb (s5): s5 file system debugger. fsdb(1M) SARM
 vfstab: table of file system defaults. vfstab(4) SARM
 ufsdump: incremental file system dump. ufsdump(1M) SARM
 file names and statistics for a file system. ff (generic): list ff(1M) SARM
 directory entry. dirent: file system independent dirent(4) SARM
 statvfs, fstatvfs: get file system information. statvfs(2) PRM
 umountfsys: mount, unmount a file system. mountfsys, mountfsys(1M) SARM
 filesystem: file system organization. filesystem(7) SARM
 quot: summarize file system ownership. quot(1M) SARM
 file that registers distributed file system packages. fstypes: fstypes(4) NUAG
 file that registers distributed file system packages. fstypes: fstypes(4) SARM
 checker. quotacheck: file system quota consistency quotacheck(1M) SARM
 quotaon, quotaoff: turn file system quotas on and off. quotaon(1M) SARM
 ufsrestore: incremental file system restore. ufsrestore(1M) SARM
 ustat: get file system statistics. ustat(2) PRM
 mnttab: mounted file system table. mnttab(4) SARM
 sharetab: shared file system table. sharetab(4) NUAG
 sharetab: shared file system table. sharetab(4) SARM
 fstyp (generic): determine file system type. fstyp(1M) SARM
 sysfs: get file system type information. sysfs(2) PRM
 names and statistics for a ufs file system. /(ufs): list file ff(1M) SARM
 make a literal copy of an s5 file system. volcopy (s5): volcopy(1M) SARM
 make a literal copy of a ufs file system. volcopy (ufs): volcopy(1M) SARM
 (generic): make literal copy of file system. volcopy volcopy(1M) SARM
 fs (s5): format of s5 file system volume. fs(4) SARM
 df: report free disk space on file systems. df(1) BSD
 fsck: check and repair file systems. fsck(1M) SARM
 mount (bfs): mount bfs file systems. mount(1M) SARM
 mount, umount: mount or unmount file systems and remote/ mount(1M) SARM
 automatically mount NFS file systems. automount: automount(1M) NUAG
 automatically mount NFS file systems. automount: automount(1M) SARM
 report free disk space on ufs file systems. df (ufs): df(1M) SARM
 time. dcopy (s5): copy s5 file systems for optimal access dcopy(1M) SARM
 names versus i-numbers for s5 file systems. /generate path ncheck(1M) SARM
 (ufs): provide labels for ufs file systems. labelit labelit(1M) SARM
 disk blocks and i-nodes for s5 file systems. /number of free df(1M) SARM
 versus i-numbers for ufs file systems. /pathnames ncheck(1M) SARM
 mount, unmount multiple file systems. /umountall: mountall(1M) SARM
 file system packages. fstypes: file that registers distributed fstypes(4) NUAG
 file system packages. fstypes: file that registers distributed fstypes(4) SARM
 blocks used per directory or file. /the number of disk du(1M) BSD

utimes: set	file times.	utimes(3) BSD
create a name for a temporary	file. tmpnam, tmpnam:	tmpnam(3S) PRM
truncate, ftruncate: set a	file to a specified length.	truncate(3C) PRM
restore: restore	file to original directory.	restore(1) XNX
installation/ install: add a	file to the software	installf(1M) SS
and modification times of a	file. touch: update access	touch(1) URM
ftp:	file transfer program.	ftp(1) NUAG
ftp:	file transfer program.	ftp(1) URM
tftp: trivial	file transfer program.	tftp(1) NUAG
tftp: trivial	file transfer program.	tftp(1) URM
ftpd:	file transfer protocol server.	ftpd(1M) NUAG
ftpd:	file transfer protocol server.	ftpd(1M) SARM
tftpd: DARPA Trivial	File Transfer Protocol server.	tftpd(1M) NUAG
tftpd: DARPA Trivial	File Transfer Protocol server.	tftpd(1M) SARM
cof2elf: COFF to ELF object	file translation.	cof2elf(1) PRM
/the scheduler for the uucp	file transport program.	uusched(1M) SARM
uucp system. uucico:	file transport program for the	uucico(1M) SARM
ftw, nftw: walk a	file tree.	ftw(3C) PRM
elf_kind: determine	file type.	elf_kind(3E) PRM
file: determine	file type.	file(1) URM
return the size of an object	file type. /elf32_fsize:	elf_fsize(3E) PRM
undo a previous get of an SCCS	file. unget:	unget(1) PRM
uucp directories and permissions	file. uucheck: check the	uucheck(1M) SARM
umask: set	file-creation mode mask.	umask(1) URM
object files.	filehdr: file header for common	filehdr(4) SARM
search FMLI criteria for	filename. pathconv:	pathconv(1F) CHAR
ferro, feof, clearerr,	fileno: stream status inquiries.	ferro(3S) PRM
cat: concatenate and print	files.	cat(1) URM
cmp: compare two	files.	cmp(1) URM
copy: copy groups of	files.	copy(1) XNX
cp: copy	files.	cp(1) URM
depend: software dependencies	files.	depend(4) SS
/dev/fd: file descriptor	files.	fd(4) SARM
find: find	files.	find(1) URM
head: display first few lines of	files.	head(1) URM
install: install	files.	install(1) BSD
intro: introduction to special	files.	intro(7) SARM
ld: link editor for object	files.	ld(1) PRM
ln: link	files.	ln(1) URM
lockf: record locking on	files.	lockf(3C) PRM
mv: move	files.	mv(1) URM
pr: print	files.	pr(1) URM
sort: sort and/or merge	files.	sort(1) URM
and print process accounting	file(s). acctcom: search	acctcom(1) URM
merge or add total accounting	files. acctmerg:	acctmerg(1M) SARM
create and administer SCCS	files. admin:	admin(1) PRM
link, unlink: link and unlink	files and directories.	link(1M) SARM

(Executable and Linking Format)
 set and/or verify permissions on
 lines common to two sorted
 idmkinit: reads
 number of free disk blocks and
 number of free disk blocks and
 extract strings from source
 file header for common object
 /user-preference variable
 catman: create the cat
 mkmsgs: create message
 format specification in text
 hd: display
 on the/ fsync: synchronize a
 make hard or symbolic links to
 passmgmt: password
 rm, rmdir: remove
 /merge same lines of several
 unpack: compress and expand
 /check nroff and troff input
 access and modification dates of
 section sizes in bytes of object
 convert: convert archive
 expand or display expanded
 create, restore an incremental
 organization.
 restore an image archive of a
 invoke XENIX incremental
 xfsck: check and repair XENIX
 restore: initiate restores of
 elf_fill: set
 nl: line numbering
 pg: file perusal
 col:
 curs_util: unctrl, keyname,
 /t4013, t450, tek: graphics
 service. lpfiler: administer
 archive of a filesystem.
 find:
 ffs:
 ttyname, isatty:
 object library. lorder:
 object file or binary. strings:
 /display reference manual pages;
 bibliographic/ lookbib:
 hashmake, spellin, hashcheck:
 files. a.out: ELF a.out(4) PRM
 files. ckperms: ckperms(1M) MBRMAN
 files. comm: select or reject comm(1) URM
 files containing specifications. idmkinit(1M) SARM
 files. df (generic): report df(1M) SARM
 files. df (generic): report df(1M) URM
 files. exstr: exstr(1) URM
 files. filehdr: filehdr(4) SARM
 files for AT&T FACE. environ(4) SARM
 files for the manual. catman(1M) BSD
 files for use by gettxt. mkmsgs(1) URM
 files. fspec: fspec(4) SARM
 files in hexadecimal format. hd(1) XNX
 file's in-memory state with that fsync(2) PRM
 files. ln: ln(1) BSD
 files management. passmgmt(1M) SARM
 files or directories. rm(1) URM
 files or subsequent lines of one/ paste(1) URM
 files. pack, pcat, pack(1) URM
 files; report possible errors. checknr(1) BSD
 files. settime: change the settime(1) XNX
 files. size: print size(1) PRM
 files to common formats. convert(1) PRM
 files. /zcat: compress, compress(1) URM
 filesystem archive. incfile: incfile(1M) SARM
 filesystem: file system filesystem(7) SARM
 filesystem. fimage: create, fimage(1M) SARM
 filesystem restorer. /xrestor: xrestor(1M) SARM
 filesystems. xfsck(1M) SARM
 filesystems, data partitions, or/ restore(1M) SARM
 fill byte. elf_fill(3E) PRM
 filter. nl(1) URM
 filter for CRTs. pg(1) URM
 filter reverse line-feeds. col(1) URM
 filter, use_env, putwin, getwin,/ curs_util(3X) CHAR
 filters for various plotters. plot(1G) BSD
 filters used with the LP print lpfiler(1M) SARM
 fimage: create, restore an image fimage(1M) SARM
 find files. find(1) URM
 find: find files. find(1) URM
 find first set bit. ffs(3C) PRM
 find name of a terminal. ttyname(3C) PRM
 find ordering relation for an lorder(1) PRM
 find printable strings in an strings(1) URM
 find reference pages by keyword. man(1) BSD
 find references in a lookbib(1) BSD
 find spelling errors. spell, spell(1) URM

queue. qsize: find the number of messages on a qsize(D3DK) DDRM
 of the current user. ttyslot: find the slot in the utmp file ttyslot(3C) PRM
 dictionary or lines in a/ look: find words in the system look(1) BSD
 interconnect/ ics_find_rec: finds a specific record in the ics_find_rec(1M) MBRMAN
 about local and remote users. finger: display information finger(1) NUAG
 about local and remote users. finger: display information finger(1) URM
 information server. fingerd, in.fingerd: remote user fingerd(1M) NUAG
 information server. fingerd, in.fingerd: remote user fingerd(1M) SARM
 elf_end: finish using an object file. elf_end(3E) PRM
 isnan, isnand, isnanf, finite, fpclass, unordered:/ isnan(3C) PRM
 /dbmclose, fetch, store, delete, firstkey, nextkey: data base/ dbm(3X) BSD
 /dbmclose, fetch, store, delete, firstkey, nextkey: database/ dbm(3) NI
 tee: pipe fitting. tee(1) URM
 hd: hard (fixed) disk. hd(7) SARM
 XENIX file permissions and/ fixperm: correct or initialize fixperm(1M) SARM
 file permissions and ownership. fixperm: correct or initialize fixperm(1) XNX
 rmsetwant: set the map's wait flag for a wakeup. rmsetwant(D3DK) DDRM
 change and display console flags. configs: configs(1M) SARM
 elf_flagshdr: manipulate flags. /elf_flagscn, elf_flag(3E) PRM
 flash/ curs beep: beep, flash: curses bell and screen curs_beep(3X) CHAR
 flash routines. /beep, curs_beep(3X) CHAR
 fddd: MULTIBUS flexible disk device drivers. fddd(1M) MBRMAN
 floatingpoint: IEEE floating point definitions. floatingpoint(3) BSD
 /fpgetsticky, fpsetsticky: IEEE floating-point environment/ fpgetround(3C) PRM
 point definitions. floatingpoint: IEEE floating floatingpoint(3) BSD
 unordered: determine type of floating-point number. /fpclass, isnan(3C) PRM
 ecvt, fcvt, gcvt: convert floating-point number to string. ecvt(3C) PRM
 /convert decimal record to floating-point value. decimal_to_floating(3) BSD
 /extended_to_decimal: convert floating-point value to decimal/ floating_to_decimal(3) BSD
 single_to_decimal, floating_to_decimal: floating_to_decimal(3) BSD
 /fabs, fabsf, rint, remainder: floor, ceiling, remainder,/ floor(3M) PRM
 copysign, fmod, fmodf, fabs,/ floor, floor, floorf, ceil, ceilf, floor(3M) PRM
 fmod, fmodf, fabs,/ floor, floorf, ceil, ceilf, copysign, floor(3M) PRM
 fd: diskette (floppy disk). fd(7) SARM
 ldsysdump: load system dump from floppy diskettes. ldsysdump(1M) SARM
 priority/ bcanput: test for flow control in specified bcanput(D3DK) DDRM
 structure. qband: STREAMS queue flow control information qband(D4DK) DDRM
 cflow: generate C flowgraph. cflow(1) PRM
 fclose, fflush: close or flush a stream. fclose(3S) PRM
 priority band. flushband: flush messages for a specified flushband(D3DK) DDRM
 specified priority band. flushband: flush messages for a flushband(D3DK) DDRM
 /putwin, getwin, delay_output, flushinp: miscellaneous curses/ curs_util(3X) CHAR
 queue. flushq: remove messages from a flushq(D3DK) DDRM
 data is aligned for efficient flyby mode. /Confirm dma_get_best_mode(D3D) DDRM
 of each line of a file. fmlcut: cut out selected fields fmlcut(1F) CHAR
 an expression. fmlexpr: evaluate arguments as fmlexpr(1F) CHAR
 pattern. fmlgrep: search a file for a fmlgrep(1F) CHAR

Permuted Index

fml: invoke	FMLI	fml(1) CHAR
fml: invoke	FMLI	fml(1) URM
co-process with the controlling	FMLI application. /synchronize a	vmsg(1F) CHAR
pathconv: search	FMLI criteria for filename.	pathconv(1F) CHAR
	fml: invoke FMLI.	fml(1) CHAR
	fml: invoke FMLI.	fml(1) URM
message: puts its arguments on	FMLI message line.	message(1F) CHAR
/floorf, ceil, ceilf, copysign,	fmod, fmodf, fabs, fabsf, rint,/	floor(3M) PRM
/ceil, ceilf, copysign, fmod,	fmodf, fabs, fabsf, rint,/	floor(3M) PRM
	fnt: simple text formatters.	fnt(1) URM
stderr or system console.	fntmsg: display a message on	fntmsg(1) URM
stderr or system console.	fntmsg: display a message on	fntmsg(3C) PRM
for an application for use with	fntmsg. /list of severity levels	addseverity(3C) PRM
	fold: fold long lines.	fold(1) URM
fold:	fold long lines.	fold(1) URM
stream.	fopen, freopen, fdopen: open a	fopen(3S) BSD
stream.	fopen, freopen, fdopen: open a	fopen(3S) PRM
YP map. yppush:	force propagation of a changed	yppush(1M) NI
resource. fumount:	forced unmount of an advertised	fumount(1M) NUAG
resource. fumount:	forced unmount of an advertised	fumount(1M) SARM
tsetpgrp: set terminal	foreground process group id.	tsetpgrp(3C) PRM
	fork: create a new process.	fork(2) PRM
reset: reset the current	form field to its default/	reset(1F) CHAR
ar: archive file	format.	ar(4) PRM
ar: archive file	format.	ar(4) SARM
hd: display files in hexadecimal	format.	hd(1) XNX
mdevice: file	format..	mdevice(4) SARM
mfsys: file	format..	mfsys(4) SARM
mtune: file	format..	mtune(4) SARM
pkgtrans: translate package	format.	pkgtrans(1) SARM
pkgtrans: translate package	format.	pkgtrans(1) SS
sdevice: file	format..	sdevice(4) SARM
sfsys: file	format..	sfsys(4) SARM
stune: file	format..	stune(4) SARM
per-process accounting file	format. acct:	acct(4) SARM
monitor-specific/ ttyadm:	format and output port	ttyadm(1M) SARM
database. roffbib:	format and print a bibliographic	roffbib(1) BSD
request message. nlsrequest:	format and send listener service	nlsrequest(3N) NI
MULTIBUS II configuration file	format. config:	config(1M) MBRMAN
getdate: convert user	format date and time.	getdate(3C) PRM
troff: typeset or	format documents.	troff(1) BSD
line-printer. nroff:	format documents for display or	nroff(1) BSD
ELF (Executable and Linking	Format files. a.out:	a.out(4) PRM
pnc: file	format for card images.	pnc(4) SARM
htable: convert DoD Internet	format host table.	htable(1M) NUAG
htable: convert DoD Internet	format host table.	htable(1M) SARM
gettable: get DoD Internet	format host table from a host.	gettable(1M) NUAG

gettable: get DoD Internet format host table from a host. gettable(1M) SARM
 inode (bfs): format of a bfs i-node. inode(4) SARM
 newform: change the format of a text file. newform(1) URM
 inode (ufs): format of a ufs inode. inode(4) SARM
 inode (s5): format of an s5 i-node. inode(4) SARM
 term: format of compiled term file. term(4) CHAR
 term: format of compiled term file. term(4) SARM
 dir (s5): format of s5 directories. dir(4) SARM
 fs (s5): format of s5 file system volume. fs(4) SARM
 sccsfile: format of SCCS file. sccsfile(4) PRM
 sccsfile: format of SCCS file. sccsfile(4) SARM
 volume. fs (bfs): format of the bfs file system fs(4) SARM
 dir (ufs): format of ufs directories. dir(4) SARM
 volume. fs (ufs): format of ufs file system fs(4) SARM
 man: macros to format Reference Manual pages. man(7) BSD
 files. fspec: format specification in text fspec(4) SARM
 troff. tbl: format tables for nroff or tbl(1) BSD
 /set_field_just, field_just: format the general appearance of/ form_field_just(3X) CHAR
 /set_field_pad, field_pad: format the general display/ form_field_attributes(3X) CHAR
 intro: introduction to file formats. intro(4) PRM
 intro: introduction to file formats. intro(4) SARM
 utmp, wtmp: utmp and wtmp entry formats. utmp(4) PRM
 utmp, wtmp: utmp and wtmp entry formats. utmp(4) SARM
 convert archive files to common formats. convert: convert(1) PRM
 library and object module formats. /loads 8086 absolute download(1M) MBRMAN
 wtmpx: utmpx and wtmpx entry formats. utmpx, utmpx(4) PRM
 wtmpx: utmpx and wtmpx entry formats. utmpx, utmpx(4) SARM
 scanf, fscanf, sscanf: convert formatted input. scanf(3S) PRM
 /mvwscanw, vwscanw: convert formatted input from a curses/ curs_scanw(3X) CHAR
 gencat: generate a formatted message catalogue. gencat(1) URM
 printf: print formatted output. printf(1) URM
 printf, fprintf, sprintf: print formatted output. printf(3S) PRM
 /vprintf, vfprintf, vsprintf: formatted output conversion. printf(3S) BSD
 /mvwprintw, vwprintw: print formatted output in curses/ curs_printw(3X) CHAR
 /vfprintf, vsprintf: print formatted output of a variable/ vprintf(3S) PRM
 hdformat: low level hard disk formatter.. hdformat(1M) MBRMAN
 fmt: simple text formatters. fmt(1) URM
 localeconv: get numeric formatting information. localeconv(3C) PRM
 ms: text formatting macros. ms(7) BSD
 me: macros for formatting papers. me(7) BSD
 position forms window cursor. form_cursor: pos_form_cursor: form_cursor(3X) CHAR
 data_behind: tell if forms/ form_data: data_ahead, form_data(3X) CHAR
 for the forms subsystem. form_driver: command processor form_driver(3X) CHAR
 form_fields, field_count,/ form_field: set_form_fields, form_field(3X) CHAR
 set_field_fore, field_fore,/ form_field_attributes: form_field_attributes(3X) CHAR
 set_field_buffer,/ form_field_buffer: form_field_buffer(3X) CHAR
 dynamic_field_info: get forms/ form_field_info: field_info, form_field_info(3X) CHAR

Permuted Index

set_field_just, field_just:/ dup_field, link_field,/ set_field_opts, field_opts_on,/ form_field: set_form_fields, free_fieldtype,/ set_field_userptr,/ set_field_type, field_type,/ form_init, set_form_term,/ form_hook: set_form_init, create and destroy forms. new_page: forms pagination. form_opts_on, form_opts_off,/ /form_opts_on, form_opts_off, /set_form_opts, form_opts_on, form_opts: set_form_opts,/ form_page, set_current_field,/ form_page: set_form_page, unpost_form: write or erase/ routines for invocation by package. /current_field, field_index: set /set_max_field: set and get /dynamic_field_info: get /field_type, field_arg: /data_ahead, data_behind: tell if /field_opts_off, field_opts: move_field: connect fields to format the general appearance of general display attributes of free_field,: create and destroy /link_fieldtype: associate application data with free_form: create and destroy associate application data with /unpost_form: write or erase /form_opts_off, form_opts: forms: character based /set_new_page, new_page: command processor for the service. lpforms: administer /form_sub, scale_form: /pos_form_cursor: position window/ /form_win, set_form_sub, /form_init, set_form_term, form_userptr: associate/ form_userptr: set_form_userptr, form_win, set_form_sub,/ form_field_just: form_field_just(3X) CHAR form_field_new: new_field, form_field_new(3X) CHAR form_field_opts: form_field_opts(3X) CHAR form_fields, field_count, / form_field(3X) CHAR form_fieldtype: new_fieldtype, form_fieldtype(3X) CHAR form_field_userptr: form_field_userptr(3X) CHAR form_field_validation: form_field_validation(3X) CHAR form_hook: set_form_init, form_hook(3X) CHAR form_init, set_form_term, / form_hook(3X) CHAR form_new: new_form, free_form: form_new(3X) CHAR form_new_page: set_new_page, form_new_page(3X) CHAR form_opts: set_form_opts, form_opts(3X) CHAR form_opts: forms option / form_opts(3X) CHAR form_opts_off, form_opts: forms / form_opts(3X) CHAR form_opts_on, form_opts_off, / form_opts(3X) CHAR form_page: set_form_page, form_page(3X) CHAR form_page, set_current_field, / form_page(3X) CHAR form_post: post_form, form_post(3X) CHAR forms. /application-specific form_hook(3X) CHAR forms: character based forms forms(3X) CHAR forms current page and field. form_page(3X) CHAR forms field attributes. form_field_buffer(3X) CHAR forms field characteristics. form_field_info(3X) CHAR forms field data type / form_field_validation(3X) CHAR forms field has off-screen data / form_data(3X) CHAR forms field option routines. form_field_opts(3X) CHAR forms. /field_count, form_field(3X) CHAR forms. /field_just: form_field_just(3X) CHAR forms. /field_pad: format the ... form_field_attributes(3X) CHAR forms fields. /link_field, form_field_new(3X) CHAR forms fieldtype routines. form_fieldtype(3X) CHAR forms. /field_userptr: form_field_userptr(3X) CHAR forms. form_new: new_form, form_new(3X) CHAR forms. /form_userptr: form_userptr(3X) CHAR forms from associated / form_post(3X) CHAR forms option routines. form_opts(3X) CHAR forms package. forms(3X) CHAR forms pagination. form_new_page(3X) CHAR forms subsystem. form_driver: form_driver(3X) CHAR forms used with the LP print lpforms(1M) SARM forms window and subwindow / form_win(3X) CHAR forms window cursor. form_cursor(3X) CHAR form_sub, scale_form: forms form_win(3X) CHAR form_term, set_field_init, / form_hook(3X) CHAR form_userptr: set_form_userptr, form_userptr(3X) CHAR form_userptr: associate / form_userptr(3X) CHAR form_win: set_form_win, form_win(3X) CHAR
--

form_win: set_form_win, for/ aliases, addresses, configurable pathname/ scalbn:/ ieee_functions, isnan, isnand, isnanf, finite, fpgetround, fpsetround, fpgetmask, fpsetmask,/ /fpgetmask, fpsetmask, formatted output. printf, vfprintf, vsprintf:/ printf, /fpsetround, fpgetmask, fpsetmask,/ fpgetround, /fpsetmask, fpgetsticky, word on a/ putc, putchar, puts, mb2a_getreqfrag: Receives a data mb2s_getreqfrag: Receives a data /receives solicited data in hat_getkpfnum: get page convert page structure to page page_numtopp: convert page
 face: executable for the getfrm: returns the current input/output. t free: freeb: freerbuf: message. freemsg: df (generic): report number of df (generic): report number of s5/ df (s5): report number of df: report systems. df (ufs): report rmwant: wait for and clear space from kernel allocate space from kernel /puts a buffer back into the memory. kmem_free: mallinfo: memory/ malloc, valloc,: memory/ malloc, idspace: investigates space management map. rmfree: Descriptor. dma_free_buf: Block. dma_free_cb: forms/ /dup_field, link_field, form_win, set_form_sub,/ form_win(3X) CHAR forward: addresses and aliases aliases(4) BSD fpathconf, pathconf: get fpathconf(2) PRM fp_class, isnan, copysign, ieee_functions(3M) BSD fpclass, unordered: determine/ isnan(3C) PRM fpgetmask, fpsetmask,/ fpgetround(3C) PRM fpgetround, fpsetround, fpgetround(3C) PRM fpgetsticky, fpsetsticky: IEEE/ fpgetround(3C) PRM fprintf, sprintf: print printf(3S) PRM fprintf, sprintf, vprintf, printf(3S) BSD fpsetmask, fpgetsticky,/ fpgetround(3C) PRM fpsetround, fpgetmask, fpgetround(3C) PRM fpsetsticky: IEEE floating-point/ fpgetround(3C) PRM fputc, putw: put character or putc(3S) PRM fputs: put a string on a stream. puts(3S) PRM fragment of a request message. mb2a_getreqfrag(3tai) TAI fragment of a request message.. mb2s_getreqfrag(3tai) TAI fragments when. mps_AMPreceive_frag(D3D) DDRM frame number for kernel address. ... hat_getkpfnum(D3K) DDRM frame number. page_pptonum: ... page_pptonum(D3DK) DDRM frame number to page structure. page_numtopp(D3DK) DDRM Framed Access Command/ face(1) URM frameID number. getfrm(1F) CHAR fread, fwrite: binary fread(3S) PRM free a library structure. t_free(3N) NI free a message block. freeb(D3DK) DDRM free a raw buffer header. freerbuf(D3DK) DDRM free all message blocks in a freemsg(D3DK) DDRM free disk blocks and files. df(1M) SARM free disk blocks and files. df(1M) URM free disk blocks and i-nodes for df(1M) SARM free disk space on file systems. df(1) BSD free disk space on ufs file df(1M) SARM free memory. rmwant(D3DK) DDRM free memory. /allocate kmem_zalloc(D3DK) DDRM free memory. kmem_alloc: kmem_alloc(D3DK) DDRM free memory pool. mps_free_msgbuf(D3D) DDRM free previously allocated kernel kmem_free(D3DK) DDRM free, realloc, calloc, malloc, malloc(3X) PRM free, realloc, calloc, memalign, malloc(3C) PRM free space. idspace(1M) SARM free space back into a private rmfree(D3DK) DDRM Free storage for a DMA Buffer dma_free_buf(D3D) DDRM Free storage for a DMA Command dma_free_cb(D3D) DDRM freeb: free a message block. freeb(D3DK) DDRM free_field,: create and destroy form_field_new(3X) CHAR

form_fieldtype: new_fieldtype,
 forms. form_new: new_form,
 specifies a driver's message
 menus/ menu_item_new: new_item,
 menus. menu_new: new_menu,
 in a message.
 header.
 specifies a driver's message/
 on the communications/ ccifree:
 descriptors.. mps_free_dmabuf:
 transaction id.. mps_free_tid:
 fopen,
 fopen,
 nextafter, scalb: manipulate.
 fdp: create, or restore
 ffile: create, or restore
 from SMTP.
 Control System (SCCS). sccs:
 dbon: sets target for
 gencc: create a
 system volume.
 system volume.
 system volume.
 analyzer.
 formatted input. scanf,
 file systems.
 systems.
 file systems.
 consistency check and/
 debugger.
 debugger.
 debugger.
 a file pointer in a stream.
 file pointer in a stream.
 generation numbers.
 text files.
 stat, lstat,
 stat, lstat,
 information. statvfs,
 system type.
 distributed file system/
 distributed file system/
 in-memory state with that on/
 in a stream. fseek, rewind,
 free_fieldtype:/ form_fieldtype(3X) CHAR
 free_form: create and destroy form_new(3X) CHAR
 freeing routine. /structure that free_rtn(D4DK) DDRM
 free_item: create and destroy menu_item_new(3X) CHAR
 free_menu: create and destroy menu_new(3X) CHAR
 freemsg: free all message blocks freemsg(D3DK) DDRM
 freerbuf: free a raw buffer freerbuf(D3DK) DDRM
 free_rtn: structure that free_rtn(D4DK) DDRM
 frees a line discipline program ccifree(1M) MBRMAN
 frees a list of data buffer mps_free_dmabuf(D3D) DDRM
 frees a previously allocated mps_free_tid(D3D) DDRM
 freopen, fdopen: open a stream. fopen(3S) BSD
 freopen, fdopen: open a stream. fopen(3S) PRM
 frep, lexp, logb, modf, modff, frexp(3C) PRM
 from, a full file system/ fdp(1M) SARM
 from, a full file system/ ffile(1M) SARM
 fromsmtp: receive RFC822 mail fromsmtp(1M) SARM
 front end for the Source Code sccs(1) BSD
 front panel message delivery. dbon(1M) MBRMAN
 front-end to the cc command. gencc(1M) SARM
 fs (bfs): format of the bfs file fs(4) SARM
 fs (s5): format of s5 file fs(4) SARM
 fs (ufs): format of ufs file fs(4) SARM
 fsba: file system block fsba(1M) SARM
 fscanf, scanf: convert scanf(3S) PRM
 fsck (bfs): check and repair bfs fsck(1M) SARM
 fsck: check and repair file fsck(1M) SARM
 fsck (s5): check and repair s5 fsck(1M) SARM
 fsck (ufs): file system fsck(1M) SARM
 fsdb (generic): file system fsdb(1M) SARM
 fsdb (s5): s5 file system fsdb(1M) SARM
 fsdb (ufs): ufs file system fsdb(1M) SARM
 fseek, rewind, ftell: reposition fseek(3S) PRM
 fsetpos, fgetpos: reposition a fsetpos(3C) PRM
 fsirand: install random inode fsirand(1) BSD
 fspec: format specification in fspec(4) SARM
 fstat: get file status. stat(2) PRM
 fstat: get file status. stat(2) XNX
 fstatvfs: get file system statvfs(2) PRM
 fstyp (generic): determine file fstyp(1M) SARM
 fstypes: file that registers fstypes(4) NUAG
 fstypes: file that registers fstypes(4) SARM
 fsync: synchronize a file's fsync(2) PRM
 ftell: reposition a file pointer fseek(3S) PRM
 ftime: get date and time. ftime(3C) BSD
 ftime: get time and date. ftime(2) XNX
 ftok: standard interprocess stdipc(3C) PRM
 communication package. stdipc:

netrc: file for server. server. specified length. truncate,
 fdp: create, or restore from, a /create, or restore from, a /a file for a pattern using /a file for a pattern using
 shutdown: shut down part of a advertised resource. advertised resource.
 gamma, lgamma: log gamma
 hypot: Euclidean distance
 matherr: error-handling
 prof: profile within a length of/ timeout: execute a function. erf, erfc: error cancel previous timeout(D3DK)
 function and complementary error IEEE exception trap handler
 libwindows: windowing terminal available. bufcall: call a available. esbcall: call backup: perform backup sysi86: machine specific atan2, atan2f: trigonometric math: math intro: introduction to j0, j1, jn, y0, y1, yn: Bessel asinh, acosh, atanh: hyperbolic password and file encryption remainder, absolute value /copysign, scalbn: miscellaneous wctombs: multibyte string logarithm, power, square root
 a file or file structure. fread, connect accounting records. open: gamma, lgamma: log function.
 ftp: file transfer program. ftp(1) NUAG
 ftp: file transfer program. ftp(1) URM
 ftp remote login data. netrc(4) NUAG
 ftp remote login data. netrc(4) SARM
 ftpd: file transfer protocol ftpd(1M) NUAG
 ftpd: file transfer protocol ftpd(1M) SARM
 ftruncate: set a file to a truncate(3C) PRM
 ftw, nftw: walk a file tree. ftw(3C) PRM
 full file system archive. fdp(1M) SARM
 full file system archive. ffile(1M) SARM
 full regular expressions. egrep(1) URM
 full regular expressions. egrep(1) XNX
 full-duplex connection. shutdown(3N) NI
 fumount: forced unmount of an fumount(1M) NUAG
 fumount: forced unmount of an fumount(1M) SARM
 function. gamma(3M) PRM
 function. hypot(3M) PRM
 function. matherr(3M) PRM
 function. prof(5) PRM
 function after a specified timeout(D3DK) DDRM
 function and complementary error erf(3M) PRM
 function call. untimeout: untimeout(D3DK) DDRM
 function. erf, erfc: error erf(3M) PRM
 function. ieee_handler: ieee_handler(3M) BSD
 function library. libwindows(3X) PRM
 function when a buffer becomes bufcall(D3DK) DDRM
 function when buffer is esbcall(D3DK) DDRM
 functions. backup(1) XNX
 functions. sysi86(2) PRM
 functions. /acosf, atan, atanf, trig(3M) PRM
 functions and constants. math(5) PRM
 functions and libraries. intro(3) PRM
 functions. bessel: bessel(3M) PRM
 functions. /coshf, tanh, tanhf, sinh(3M) PRM
 functions. crypt: crypt(3X) PRM
 functions. /floor, ceiling, floor(3M) PRM
 functions for IEEE arithmetic. ieee_functions(3M) BSD
 functions. mbstring: mbstowcs, mbstring(3C) PRM
 functions. /sqrtf: exponential, exp(3M) PRM
 fusage: disk access profiler. fusage(1M) NUAG
 fusage: disk access profiler. fusage(1M) SARM
 fuser: identify processes using fuser(1M) SARM
 fwrite: binary input/output. fread(3S) PRM
 fwtmp, wtmpfix: manipulate fwtmp(1M) SARM
 gain access to a device. open(D2DK) DDRM
 gamma function. gamma(3M) PRM
 gamma, lgamma: log gamma gamma(3M) PRM

Permuted Index

/mdiv, mcmp, min, mout, pow, sgconvert:/ econvert, fconvert, running processes.
 number to string. ecvt, fcvt, message catalogue.
 cc command.
 /field_just: format the forms. /field_pad: format the Protocol network/ if: /tcgetpgrp, tcsetpgrp, tcgetsid: termiox: extended control. priocntlset:
 vs i-numbers. ncheck (generic): catalogue. gencat: pkgproto: random: signal. abort: cflow: cross-reference. cxref: classification and/ chrtbl: user ID. diskusg: crypt, setkey, encrypt: makekey: ctermid: pattern. lptest: i-numbers for s5/ ncheck (s5): i-numbers for ufs/ ncheck (ufs): lexical tasks. lex: /srand48, seed48, lcong48: siginfo: signal siginfo: signal fsirand: install random inode srand: simple random number srand: simple random-number /setstate: better random number generator; routines for changing system. mkfs optimal access time. dcopy type. fstyp fsdb path names vs i-numbers. ncheck statistics for a file/ ff file system. volcopy file systems. labelit disk blocks and files. df disk blocks and files. df netdir_perror, netdir_spperror: gcd, rpow, msqrt, sdiv, itom,/ mp(3X) BSD gconvert, seconvert, sfconvert, econvert(3) BSD gcore: get core images of gcore(1) URM gcvt: convert floating-point ecvt(3C) PRM gencat: generate a formatted gencat(1) URM gencc: create a front-end to the gencc(1M) SARM general appearance of forms. form_field_just(3X) CHAR general display attributes of form_field_attributes(3X) CHAR general properties of Internet if(7) NUAG general terminal interface. termiox(2) PRM general terminal interface. termiox(7) SARM generalized process scheduler priocntlset(2) PRM generate a list of path names ncheck(1M) SARM generate a formatted message gencat(1) URM generate a prototype file. pkgproto(1) SS generate a random number. random(1) XNX generate an abnormal termination abort(3C) PRM generate C flowgraph. cflow(1) PRM generate C program cxref(1) PRM generate character chrtbl(1M) SARM generate disk accounting data by diskusg(1M) SARM generate encryption. crypt(3C) PRM generate encryption key. makekey(1) URM generate file name for terminal. ctermid(3S) PRM generate lineprinter ripple lptest(1) BSD generate path names versus ncheck(1M) SARM generate pathnames versus ncheck(1M) SARM generate programs for simple lex(1) PRM generate uniformly distributed/ drand48(3C) PRM generation information. siginfo(5) PRM generation information. siginfo(5) SARM generation numbers. fsirand(1) BSD generator. rand, rand(3C) BSD generator. rand, rand(3C) PRM generator; routines for changing/ random(3) BSD generators. /random number random(3) BSD (generic): construct a file mkfs(1M) SARM (generic): copy file systems for dcopy(1M) SARM (generic): determine file system fstyp(1M) SARM (generic): file system debugger. fsdb(1M) SARM (generic): generate a list of ncheck(1M) SARM (generic): list file names and ff(1M) SARM (generic): make literal copy of volcopy(1M) SARM (generic): provide labels for labelit(1M) SARM (generic): report number of free df(1M) SARM (generic): report number of free df(1M) URM generic transport/ /uaddr2taddr, netdir(3N) NI

file.
 curs_getyx: getyx, getparyx,
 character or word from a/
 mvwgetch, ungetch:/ curs_getch:
 character or word from a/ getch,
 set current user context.
 working directory.
 date and time.
 and put in a file.
 criteria.
 which contain devices that/
 table size.
 getuid, geteuid, getgid,
 environment name.
 real user, effective/ getuid,
 frameID number.
 effective/ getuid, geteuid,
 setgrent, endgrent, fgetgrent:/
 endgrent, fgetgrent:/ getgrent,
 fgetgrent:/ getgrent, getgrgid,
 supplementary group access list/
 sethostent,/ gethostent,
 gethostent, gethostbyaddr,
 gethostbyname, sethostent,/
 of current host.
 get/set name of current host.
 currently marked menu items.
 value of interval timer.
 major device number.
 /getyx, getparyx, getbegyx,
 minor device number.
 entry. getmntent,
 file entry.
 stream.
 stream.
 setnetent,/ getnetent,
 getnetent, getnetbyaddr,
 configuration database entry.
 getnetbyname, setnetent,/ /
 authdes_getucred,
 entry corresponding to NETPATH/
 argument vector.
 options. getopt, getopts,
 command options.
 get: get a version of an SCCS get(1) PRM
 getbegyx, getmaxyx: get curses/ curs_getyx(3X) CHAR
 getc, getchar, fgetc, getw: getgetc(3S) PRM
 getch, wgetch, mvgetch, curs_getch(3X) CHAR
 getchar, fgetc, getw: getgetc(3S) PRM
 getcontext, setcontext: get andgetcontext(2) PRM
 getcwd: get pathname of currentgetcwd(3C) PRM
 getdate: convert user formatgetdate(3C) PRM
 getdents: read directory entriesgetdents(2) PRM
 getdev: lists devices based ongetdev(1M) SARM
 getdgrp: lists device groupsgetdgrp(1M) SARM
 getdtablesize: get descriptorgetdtablesize(3) BSD
 getegid: get real user,/getuid(2) PRM
 getenv: return value forgetenv(3C) PRM
 geterror: return I/O error.geterror(D3DK) DDRM
 geteuid, getgid, getegid: getgetuid(2) PRM
 getfrm: returns the currentgetfrm(1F) CHAR
 getgid, getegid: get real user,getuid(2) PRM
 getgrent, getgrgid, getgrnam,getgrent(3C) PRM
 getgrgid, getgrnam, setgrent,getgrent(3C) PRM
 getgrnam, setgrent, endgrent,getgrent(3C) PRM
 getgroups, setgroups: get or setgetgroups(2) PRM
 gethostbyaddr, gethostbyname,gethostent(3N) NI
 gethostbyname, sethostent,/gethostent(3N) NI
 gethostent, gethostbyaddr,gethostent(3N) NI
 gethostid: get unique identifiergethostid(3) BSD
 gethostname, sethostname:gethostname(3) BSD
 getitimers: return a list ofgetitimers(1F) CHAR
 getitimer, setitimer: get/setgetitimer(3C) PRM
 getlogin: get login name.getlogin(3C) PRM
 getmajor: get major or internalgetmajor(D3DK) DDRM
 getmaxyx: get curses cursor and/ curs_getyx(3X) CHAR
 getminor: get minor or internalgetminor(D3DK) DDRM
 getmntany: get mnttab filegetmntent(3C) PRM
 getmntent, getmntany: get mnttabgetmntent(3C) PRM
 getmsg: get next message off agetmsg(2) PRM
 getmsg: get next message off agetmsg(2) STRM
 getnetbyaddr, getnetbyname,getnetent(3N) NI
 getnetbyname, setnetent,/getnetent(3N) NI
 getnetconfig: get networkgetnetconfig(3N) NI
 getnetent, getnetbyaddr,getnetent(3N) NI
 getnetname, host2netname,/secure_rpc(3N) NI
 getnetpath: get /etc/netconfiggetnetpath(3N) NI
 getopt: get option letter fromgetopt(3C) PRM
 getopt: parse command options.getopt(1) URM
 getoptcv: parse commandgetopts(1) URM
 getopts, getoptcv: parsegetopts(1) URM

size.
 get curses/ curs_getyx: getyx,
 connected peer.
 getpid, getpgrp, getppid,
 process, process group,/ getpid,
 getpgid: get process, process/
 process group,/ getpid, getpgrp,
 get/set program scheduling/
 getprotoent, getprotobyname,
 getprotobyname,/ getprotoent,
 getprotobyname, setprotoent,
 retrieve public or/ publickey:

 setpwent, endpwent, fgetpwent:/
 fgetpwent:/ getpwent, getpwuid,
 endpwent, fgetpwent:/ getpwent,
 a queue.
 header.
 maximum system resource/
 resource utilization.
 stream.
 mb2_gethostid:
 readfile, longline: reads file,
 endpoint/ mb2a_getinfo:
 endpoint/ mb2s_getinfo:
 secret/ publickey: getpublickey,
 getservent, getservbyport,
 setservent,/ getservent,
 getservbyname, setservent,
 getservbyname, setservent,
 gethostname, sethostname:
 RPC domain. domainname:
 getpriority, setpriority:
 getitimer, setitimer:

 set options on sockets.
 endspent, fgetspent, lckpwdf,/
 fgetspent, lckpwdf,/ getspent,
 mvwgetstr,/ curs_getstr:
 a string.
 curs_set,/ /resetty, savetty,
 format host table from a host.
 format host table from a host.
 or set the date and time.
 or set the date and time.
 create message files for use by
 getpagesize: get system page getpagesize(3) BSD
 getparyx, getbegyx, getmaxyx: curs_getyx(3X) CHAR
 getpass: read a password. getpass(3C) PRM
 getpeername: get name of getpeername(3N) NI
 getpgid: get process, process/ getpid(2) PRM
 getpgrp, getppid, getpgid: get getpid(2) PRM
 getpid, getpgrp, getppid, getpid(2) PRM
 getppid, getpgid: get process, getpid(2) PRM
 getpriority, setpriority: getpriority(3) BSD
 getprotobyname, setprotoent,/ getprotoent(3N) NI
 getprotobyname, getprotoent(3N) NI
 getprotoent, getprotobyname, getprotoent(3N) NI
 getpublickey, getsecretkey: publickey(3N) NI
 getpw: get name from UID. getpw(3C) PRM
 getpwent, getpwuid, getpwnam, getpwent(3C) PRM
 getpwnam, setpwent, endpwent, getpwent(3C) PRM
 getpwuid, getpwnam, setpwent, getpwent(3C) PRM
 getq: get the next message from getq(D3DK) DDRM
 getrbuf: get a raw buffer getrbuf(D3DK) DDRM
 getrlimit, setrlimit: control getrlimit(2) PRM
 getrusage: get information about getrusage(3) BSD
 gets, fgets: get a string from a gets(3S) PRM
 gets host ID.. mb2_gethostid(3tai) TAI
 gets longest line. readfile(1F) CHAR
 Gets specific asynchronous mb2a_getinfo(3tai) TAI
 Gets specific synchronous mb2s_getinfo(3tai) TAI
 getsecretkey: retrieve public or publickey(3N) NI
 getservbyname, setservent,/ getservent(3N) NI
 getservbyport, getservbyname, getservent(3N) NI
 getservent, getservbyport, getservent(3N) NI
 get/set name of current host. gethostname(3) BSD
 get/set name of current secure domainname(1M) NI
 get/set program scheduling/ getpriority(3) BSD
 get/set value of interval timer. getitimer(3C) PRM
 getsid: get session ID. getsid(2) PRM
 getsockname: get socket name. getsockname(3N) NI
 getsockopt, setsockopt: get and getsockopt(3N) NI
 getspent, getspnam, setspent, getspent(3C) PRM
 getspnam, setspent, endspent, getspent(3C) PRM
 getstr, wgetstr, mvwgetstr, curs_getstr(3X) CHAR
 getsubopt: parse suboptions from getsubopt(3C) PRM
 getsyx, setsyx, ripoffline, curs_kernel(3X) CHAR
 gettable: get DoD Internet gettable(1M) NUAG
 gettable: get DoD Internet gettable(1M) SARM
 gettimeofday, settimeofday: get gettimeofday(3) BSD
 gettimeofday, settimeofday: get gettimeofday(3C) PRM
 gettxt. mkmsgs: mkmsgs(1) URM

from a message data base.

speed, and line discipline.

 getegid: get real user/
 endusershell: get legal user/
 getutline, pututline, setutent,/
 pututline, setutent,/ getut:
 setutent,/ getut: getutent,
 getut: getutent, getutid,
/setutxent, endutxent, utmpxname,
/endutxent, utmpxname, getutmp,
 getutxline, pututxline,/
 pututxline, setutxent,/ getutx:
 pututxline,/ getutx: getutxent,
 getutx: getutxent, getutxid,
 entry. /getvfile, getvfsspec,
 getvfsspec, getvfsany: get/
 getvfsany: get/ getvfile,
 vfstab/ getvfile, getvfile,
 accessibility.

a stream. getc, getchar, fgetc,
 directory pathname.
 /filter, use_env, putwin,
 getmaxyx: get/ curs_getyx:
 /bgplot, crtplot, dumbplot,
 messages. biff:
 bzero: clear memory for a
 timezone: get time zone name
 reset: resets the processor in a
 close down the system at a
 /unset: set and unset local or
 gmatch: shell
 matching.

time zone name given offset from
 date and time/ ctime, localtime,
 setjmp, longjmp: non-local
 sigsetjmp, siglongjmp: non-local
 /siglongjmp: a non-local
 and check access to a resource
 this/ /print the list of service
 pseudo-terminal/ grantpt:
 request.. /construct a buffer
 slave pseudo-terminal device.
 sag: system activity
 /t300, t300s, t4013, t450, tek:
 pattern.
 pattern.

gettxt: retrieve a text string gettxt(1) URM
 gettxt: retrieve a text string. gettxt(3C) PRM
 getty: set terminal type, modes, getty(1M) SARM
 getuid, geteuid, getgid, getuid(2) PRM
 getusershell, setusershell, getusershell(3) BSD
 getut: getutent, getutid, getut(3C) PRM
 getutent, getutid, getutline, getut(3C) PRM
 getutid, getutline, pututline, getut(3C) PRM
 getutline, pututline, setutent,/ getut(3C) PRM
 getutmp, getutmpx, updwtmp,/ getut(3C) PRM
 getutmpx, updwtmp, updwtmpx:/ getut(3C) PRM
 getutx: getutxent, getutxid, getutx(3C) PRM
 getutxent, getutxid, getutxline, getutx(3C) PRM
 getutxid, getutxline, getutx(3C) PRM
 getutxline, pututxline,/ getutx(3C) PRM
 getvfsany: get vfstab file getvfile(3C) PRM
 getvfile, getvfile, getvfile(3C) PRM
 getvfile, getvfsspec, getvfile(3C) PRM
 getvfsspec, getvfsany: get getvfile(3C) PRM
 getvol: verifies device getvol(1M) SARM
 getw: get character or word from getc(3S) PRM
 getwd: get current working getwd(3) BSD
 getwin, delay_output, flushinp:/ curs_util(3X) CHAR
 getyx, getparyx, getbegyx, curs_getyx(3X) CHAR
 gigiplot, hpplot, implot, t300,/ plot(1G) BSD
 give notice of incoming mail biff(1) BSD
 given number of bytes. bzero(D3DK) DDRM
 given offset from GMT. timezone(3C) BSD
 given slot. reset(1M) MBRMAN
 given time. shutdown: shutdown(1M) BSD
 global environment variables. set(1F) CHAR
 global pattern matching. gmatch(3G) PRM
 gmatch: shell global pattern gmatch(3G) PRM
 GMT. timezone: get timezone(3C) BSD
 gmtime, asctime, tzset: convert ctime(3C) PRM
 goto. setjmp(3C) PRM
 goto. / setjmp, longjmp, setjmp(3) BSD
 goto with signal state. sigsetjmp(3C) PRM
 governed by a semaphore. /await waitsem(2) XNX
 grades that are available on uuglist(1C) URM
 grant access to the slave grantpt(3C) STRM
 grant in response to a buffer mps_mk_bgrant(D3D) DDRM
 grantpt: grant access to the grantpt(3C) STRM
 graph. sag(1) URM
 graphics filters for various/ plot(1G) BSD
 grep: search a file for a grep(1) URM
 grep: search a file for a grep(1) XNX

killpg: send signal to a process	group. killpg(3) BSD
newgrp: log in to a new	group. newgrp(1M) SARM
newgrp: log in to a new	group. newgrp(1M) URM
/get or set supplementary	group access list IDs. getgroups(2) PRM
initialize the supplementary	group access list. initgroups: initgroups(3C) PRM
/real user, effective user, real	group, and effective group IDs. getuid(2) PRM
/getpgid: get process, process	group, and parent process IDs. getpid(2) PRM
grpck: check	group database entries. grpck(1M) BSD
system. groupdel: delete a	group definition from the groupdel(1M) SARM
groupadd: add (create) a new	group definition on the system. groupadd(1M) SARM
groupmod: modify a	group definition on the system. groupmod(1M) SARM
group:	group file. group(4) SARM
endgrent, fgetgrent: get	group file entry. /setgrent, getgrent(3C) PRM
setpgid: set process	group: group file. group(4) SARM
setpgrp: set process	group ID. setpgid(2) PRM
prompts for and validates a	group ID. setpgrp(2) PRM
prompts for and validates a	group id. /helpgid, valgid: ckgid(1) SARM
set terminal foreground process	group id. /helpgid, valgid: ckgid(1) SS
setregid: set real and effective	group id. tsetpgrp: tsetpgrp(3C) PRM
setuid, setgid: set user and	group IDs. setregid(3) BSD
user, real group, and effective	group ID. setuid(2) PRM
lists members of a device	group IDs. /real user, effective getuid(2) PRM
Remote File Sharing user and	group. listdgrp: listdgrp(1M) SARM
Remote File Sharing user and	group mapping. idload: idload(1M) NUAG
groups: print	group mapping. idload: idload(1M) SARM
groups: display a user's	group membership of user. groups(1) URM
print the user name and ID, and	group memberships. groups(1) BSD
print the user name and ID, and	group name and ID. id: id(1M) SARM
displays a list of all valid	group name and ID. id: id(1M) URM
displays a list of all valid	group names. dispgid: dispgid(1) SARM
ichown, fchown: change owner and	group names. dispgid: dispgid(1) SS
send a signal to a process or a	group of a file. chown, chown(2) PRM
send a signal to a process or a	group of processes. kill: kill(2) PRM
chgrp: change the	group of processes. /sigsendset: sigsend(2) PRM
send a signal to a process	group ownership of a file. chgrp(1) URM
putdgrp: edits device	group. signal: signal(D3D) DDRM
group definition on the system.	group table. putdgrp(1) SARM
definition from the system.	groupadd: add (create) a new groupadd(1M) SARM
definition on the system.	groupdel: delete a group groupdel(1M) SARM
memberships.	groupmod: modify a group groupmod(1M) SARM
copy: copy	groups: display a user's group groups(1) BSD
maintain, update, and regenerate	groups of files. copy(1) XNX
of user.	groups of programs. make: make(1) PRM
that/ getdgrp: lists device	groups: print group membership groups(1) URM
entries.	groups which contain devices getdgrp(1M) SARM
checkers. pwck,	grpck: check group database grpck(1M) BSD
	grpck: password/group file pwck(1M) SARM

/cbreak, nocbreak, echo, noecho,
 reboot: reboot system or
 the system shutdowns.
 ssignal, gsignal: software signals. ssignal(3C) PRM
 halfdelay, intrflush, keypad,/ curs_inopts(3X) CHAR
 halt processor. reboot(3) BSD
 halt: shutdown the driver when halt(D2D) DDRM
 halt: stop the processor. halt(1M) BSD
 stdarg: handle variable argument list. stdarg(5) PRM
 varargs: handle variable argument list. varargs(5) PRM
 IEEE exception trap
 re_exec: regular expression
 and manipulation of CLIENT
 with the creation of server
 elf_errmsg, elf_errno: error
 package. curses: CRT screen
 codes. sigfpe: signal
 isgraph, isascii: character
 wctomb: multibyte character
 nohup: run a command immune to
 hdd: MULTIBUS
 hdformat: low level
 hd:
 ln: make
 setclk: set system time from
 Programming a Channel for a
 /init_pair, init_color,
 hcreate, hdestroy: manage
 elf_hash: compute
 spell, hashmake, spellin,
 find spelling errors. spell,
 longname,/ /baudrate, erasechar,
 /baudrate, erasechar, has_ic,
 number for kernel address.
 search tables. hsearch,
 format.
 drivers.
 tables. hsearch, hcreate,
 formatter..
 files.
 remove a message block from the
 putbq: place a message at the
 freerbuf: free a raw buffer
 getrbuf: get a raw buffer
 retrieve class-dependent section
 class-dependent object file
 retrieve archive member
 archives: device
 implementation-specific/ limits:

gsignal: software signals. ssignal(3C) PRM
 halfdelay, intrflush, keypad,/ curs_inopts(3X) CHAR
 halt processor. reboot(3) BSD
 halt: shutdown the driver when halt(D2D) DDRM
 halt: stop the processor. halt(1M) BSD
 stdarg: handle variable argument list. stdarg(5) PRM
 varargs: handle variable argument list. varargs(5) PRM
 IEEE exception trap
 re_exec: regular expression
 and manipulation of CLIENT
 with the creation of server
 elf_errmsg, elf_errno: error
 package. curses: CRT screen
 codes. sigfpe: signal
 isgraph, isascii: character
 wctomb: multibyte character
 nohup: run a command immune to
 hdd: MULTIBUS
 hdformat: low level
 hd:
 ln: make
 setclk: set system time from
 Programming a Channel for a
 /init_pair, init_color,
 hcreate, hdestroy: manage
 elf_hash: compute
 spell, hashmake, spellin,
 find spelling errors. spell,
 longname,/ /baudrate, erasechar,
 /baudrate, erasechar, has_ic,
 number for kernel address.
 search tables. hsearch,
 format.
 drivers.
 tables. hsearch, hcreate,
 formatter..
 files.
 remove a message block from the
 putbq: place a message at the
 freerbuf: free a raw buffer
 getrbuf: get a raw buffer
 retrieve class-dependent section
 class-dependent object file
 retrieve archive member
 archives: device
 implementation-specific/ limits:

implementation-specific/ limits:	header file for	limits(4) SARM
constants. unistd:	header file for symbolic	unistd(4) SARM
filehdr: file	header for common object files.	filehdr(4) SARM
retrieve class-dependent program	header table. /elf32_newphdr:	elf_getphdr(3E) PRM
numbers or SCCS commands.	help: ask for help with message	help(1) PRM
SCCS commands. help: ask for	help with message numbers or	help(1) PRM
and validates/ ckdate, errdate,	helpdate, valdate: prompts for	ckdate(1) SARM
and validates/ ckdate, errdate,	helpdate, valdate: prompts for	ckdate(1) SS
validates a/ ckgid, errgid,	helpgid, valgid: prompts for and	ckgid(1) SARM
validates a/ ckgid, errgid,	helpgid, valgid: prompts for and	ckgid(1) SS
hd: display files in	hexadecimal format.	hd(1) XNX
deck/ panel show: show_panel,	hide_panel, panel_hidden: panels	panel_show(3X) CHAR
display the delta and commentary	history of an SCCS file. prt:	prt(1) BSD
/wvline: create curses borders,	horizontal and vertical lines.	cursor(3X) CHAR
sdi_init: initialize the	host adapter.	sdi_init(D3I) SCSI
/ntohs: convert values between	host and network byte order.	byteorder(3N) NI
layers: protocol used between	host and windowing terminal/	layers(5) PRM
layers: protocol used between	host and windowing terminal/	layers(5) SARM
the architecture of the current	host. arch: display	arch(1) BSD
terminal. jagent:	host control of windowing	jagent(5) PRM
terminal. jagent:	host control of windowing	jagent(5) SARM
endhostent: get network	host entry. /sethostent,	gethostent(3N) NI
format host table from a	host. /get DoD Internet	gettable(1M) NUAG
format host table from a	host. /get DoD Internet	gettable(1M) SARM
get unique identifier of current	host. gethostid:	gethostid(3) BSD
get/set name of current	host. gethostname, sethostname:	gethostname(3) BSD
identifier of the current	host. hostid: print the numeric	hostid(1) BSD
mb2_gethostid: gets	host ID..	mb2_gethostid(3tai) TAI
record/ ics_hostid: returns the	host id field of the HOST ID	ics_hostid(D3D) DDRM
/returns the host id field of the	HOST ID record in this.	ics_hostid(D3D) DDRM
processor type of the current	host. mach: display the	mach(1) BSD
hosts:	host name data base.	hosts(4) NI
hosts:	host name data base.	hosts(4) NUAG
hosts:	host name data base.	hosts(4) SARM
/a subchannel on a line to a new	host on the communications/	cciswitch(1M) MBRMAN
/switches a line to a new	host on the communications/	ttyswitch(1M) MBRMAN
change Remote File Sharing	host password. rpasswd:	rpasswd(1M) NUAG
change Remote File Sharing	host password. rpasswd:	rpasswd(1M) SARM
set system date from a remote	host. rdate:	rdate(1M) NUAG
set system date from a remote	host. rdate:	rdate(1M) SARM
of a YP map at a YP server	host. /return current version	yppoll(1M) NI
ruptime: show	host status of local machines.	ruptime(1) NUAG
ruptime: show	host status of local machines.	ruptime(1) URM
set or print name of current	host system. hostname:	hostname(1) BSD
/get DoD Internet format	host table from a host.	gettable(1M) NUAG
/get DoD Internet format	host table from a host.	gettable(1M) SARM
convert DoD Internet format	host table. htable:	htable(1M) NUAG

convert DoD Internet format
 smtp: send SMTP mail to a remote
 ccidetch: disassociates this
 YP map from a YP server to
 /authdes_getcred, getnetname,
 identifier of the current host.
 ethers: Ethernet address to
 ethers: Ethernet address to
 current host system.
 hosts.equiv, rhosts: trusted
 hosts.equiv, rhosts: trusted

ECHO_REQUEST packets to network
 ECHO_REQUEST packets to network
 hosts by system and by user.
 hosts by system and by user.
 /crtplot, dumbplot, gigiplot,
 manage hash search tables.
 format host table.
 format host table.
 convert values/ byteorder,
 values/ byteorder, htonl,
 /maintain line settings and
 tanh, asinh, acosh, atanh:
 function.
 controller disk device driver.
 controller tape device driver.
 controller disk device driver.
 peripheral controller tape/
 controller device driver.
 controller tape device driver.
 serial-port device driver.
 communications controller/
 controller device driver.
 controller.
 d258:
 351 driver.
 device driver.
 network hosts. ping: send
 network hosts. ping: send
 Protocol.
 Protocol.
 Protocol.
 utility.
 tables.

host table. htable: htable(1M) SARM
 host using Simple Mail Transfer/ smtp(1M) SARM
 host with a subchannel on a line/ ccidetch(1M) MBRMAN
 host. ypxfr: transfer ypxfr(1M) NI
 host2netname,/ secure_rpc(3N) NI
 hostid: print the numeric hostid(1) BSD
 hostname database or domain. ethers(4) NUAG
 hostname database or domain. ethers(4) SARM
 hostname: set or print name of hostname(1) BSD
 hosts by system and by user. hosts.equiv(4) NUAG
 hosts by system and by user. hosts.equiv(4) SARM
 hosts: host name data base. hosts(4) NI
 hosts: host name data base. hosts(4) NUAG
 hosts: host name data base. hosts(4) SARM
 hosts. ping: send ICMP ping(1M) NUAG
 hosts. ping: send ICMP ping(1M) SARM
 hosts.equiv, rhosts: trusted hosts.equiv(4) NUAG
 hosts.equiv, rhosts: trusted hosts.equiv(4) SARM
 hplot, implot, t300, t300s,/ plot(1G) BSD
 hsearch, hcreate, hdestroy: hsearch(3C) PRM
 htable: convert DoD Internet htable(1M) NUAG
 htable: convert DoD Internet htable(1M) SARM
 htonl, htols, ntohl, ntohs: byteorder(3N) NI
 htols, ntohl, ntohs: convert byteorder(3N) NI
 hunt sequences for TTY ports. sttydefs(1M) SARM
 hyperbolic functions. /tanh, sinh(3M) PRM
 hypot: Euclidean distance hypot(3M) PRM
 i214: iSBCR 214 peripheral i214(1M) MBRMAN
 i214tp: iSBC 214 peripheral i214tp(1M) MBRMAN
 i224a: iSBC 186/224A peripheral i224a(1M) MBRMAN
 i224atp: iSBC 186/224A i224atp(1M) MBRMAN
 i258: iSBC 386/258 peripheral i258(1M) MBRMAN
 i258tp: iSBC 386/258 peripheral i258tp(1M) MBRMAN
 i354: iSBX 354 dual channel i354(1M) MBRMAN
 i410: iSBC 186/410 serial i410(1M) MBRMAN
 i530: iSBC 186/530 Ethernet i530(1M) MBRMAN
 i546: iSBC 546 multi-port serial i546(1M) MBRMAN
 i82258 ADMA device driver. d258(1M) MBRMAN
 i8251: console terminal/iSBXT i8251(1M) MBRMAN
 iasy: asynchronous (terminal) iasy(1M) MBRMAN
 ICMP ECHO_REQUEST packets to ping(1M) NUAG
 ICMP ECHO_REQUEST packets to ping(1M) SARM
 ICMP: Internet Control Message ICMP(7) NI
 ICMP: Internet Control Message ICMP(7) NUAG
 ICMP: Internet Control Message ICMP(7) SARM
 iconv: code set conversion iconv(1) URM
 iconv: code set conversion iconv(5) SARM

Permuted Index

driver.
 certain board types in the/
 interconnect register of the/
 record in the interconnect/
 MULTIBUS/ icsrd, icswr, icsslot,
 field of the HOST ID record in/
 icsgetrec: utilities to access/
 specified number of/
 registers of the board and/
 register of the board in the.
 access MULTIBUS/ icsrd, icswr,
 utilities to access/ icsrd,
 the specified register of the/
 interconnect registers of the/
 getsid: get session
 mb2_gethostid: gets host
 setpgid: set process group
 setpgrp: set process group
 setsid: set session
 id: print the user name and
 id: print the user name and
 prompts for and validates a user
 prompts for and validates a user
 disk accounting data by user
 in/ ics_hostid: returns the host
 for and validates a group
 for and validates a group
 name and ID, and group name and
 name and ID, and group name and
 previously allocated transaction
 and group name and ID.
 and group name and ID.
 the host id field of the HOST
 semaphore set, or shared memory
 foreground process group
 kernel.
 information.
 curs_outopts: clearok, idlok,
 configuration.
 module_info: STREAMS driver
 elf_getident: retrieve file
 issue: issue
 what: print
 gethostid: get numeric
 hostid: print the unique
 get shared memory segment
 or file structure. fuser:

ics: Interconnect Space device ics(1M) MBRMAN
 ics_agent_cmp: checks for ics_agent_cmp(D3D) DDRM
 ics_find_rec: reads the ics_find_rec(D3D) DDRM
 ics_find_rec: finds a specific ics_find_rec(1M) MBRMAN
 icsgetrec: utilities to access icsrd(1M) MBRMAN
 ics_hostid: returns the host id ics_hostid(D3D) DDRM
 icsrd, icswr, icsslot, icsrd(1M) MBRMAN
 ics_rdwr: reads or writes a ics_rdwr(D3D) DDRM
 ics_read: reads interconnect ics_read(1M) MBRMAN
 ics_read: reads the interconnect ics_read(D3D) DDRM
 icsslot, icsgetrec: utilities to icsrd(1M) MBRMAN
 icswr, icsslot, icsgetrec: icsrd(1M) MBRMAN
 ics_write: writes a value into ics_write(D3D) DDRM
 ics_write: writes into ics_write(1M) MBRMAN
 ID. getsid(2) PRM
 ID.. mb2_gethostid(3tai) TAI
 ID. setpgid(2) PRM
 ID. setpgrp(2) PRM
 ID. setsid(2) PRM
 ID, and group name and ID. id(1M) SARM
 ID, and group name and ID. id(1M) URM
 ID. ckuid: ckuid(1) SARM
 ID. ckuid: ckuid(1) SS
 ID. diskusg: generate diskusg(1M) SARM
 id field of the HOST ID record ics_hostid(D3D) DDRM
 id. /helpgid, valgid: prompts ckgid(1) SARM
 id. /helpgid, valgid: prompts ckgid(1) SS
 ID. id: print the user id(1M) SARM
 ID. id: print the user id(1M) URM
 id.. mps_free_tid: frees a mps_free_tid(D3D) DDRM
 id: print the user name and ID, id(1M) SARM
 id: print the user name and ID, id(1M) URM
 ID record in this. /returns ics_hostid(D3D) DDRM
 ID. /remove a message queue, ipcrm(1) URM
 id. tcsetpgrp: set terminal tcsetpgrp(3C) PRM
 idbuild: build new UNIX System idbuild(1M) SARM
 idcheck: returns selected idcheck(1M) SARM
 idcok immedok, leaveok, / curs_outopts(3X) CHAR
 idconfig: produce a new kernel idconfig(1M) SARM
 identification and limit value/ module_info(D4DK) DDRM
 identification data. elf_getident(3E) PRM
 identification file. issue(4) SARM
 identification strings. what(1) PRM
 identifier of current host. gethostid(3) BSD
 identifier of the current host. hostid(1) BSD
 identifier. shmget: shmget(2) PRM
 identify processes using a file fuser(1M) SARM

or get device driver/
 and group mapping.
 and group mapping.
 curs_outopts: clearok,
 specifications.
 specifications of nodes.
 kernel.
 group, and parent process
 allocates transaction
 real group, and effective group
 supplementary group access list
 set real and effective group
 set real and effective user
 setgid: set user and group
 space.
 a tunable parameter.
 miscellaneous functions for
 function. ieee handler:
 floatingpoint:
 /fpgetsticky, fpsetsticky:
 copysign, scalbn: miscellaneous/
 trap handler function.
 Internet Protocol network/
 interface parameters.
 interface parameters.
 undef: resolve and remove
 bootserver: starts a MULTIBUS
 config: MULTIBUS
 /utilities to access MULTIBUS
 Receives incoming MULTIBUS
 Receives incoming MULTIBUS
 /Closes an asynchronous MULTIBUS
 /Opens a asynchronous MULTIBUS
 /Opens a synchronous MULTIBUS
 Closes a synchronous MULTIBUS
 ff (s5): display
 fimage: create, restore an
 core: core
 core: core
 crash: examine system
 pnch: file format for card
 gcore: get core
 perform requested operation
 /clearok, idlok, idcok
 nohup: run a command
 limits: header file for
 limits: header file for
 idinstall: add, delete, update, idinstall(1M) SARM
 idload: Remote File Sharing user idload(1M) NUAG
 idload: Remote File Sharing user idload(1M) SARM
 idlok, idcok immedok, leaveok,/ curs_outopts(3X) CHAR
 idmkinit: reads files containing idmkinit(1M) SARM
 idmknod: removes nodes and reads idmknod(1M) SARM
 idmkunix: build new UNIX System idmkunix(1M) SARM
 IDs. /get process, process getpid(2) PRM
 ids.. mps_get_tid: mps_get_tid(D3D) DDRM
 IDs. /real user, effective user, getuid(2) PRM
 IDs. /setgroups: get or set getgroups(2) PRM
 IDs. setregid: setregid(3) BSD
 IDs. setreuid: setreuid(3) BSD
 IDs. setuid, setuid(2) PRM
 idspace: investigates free idspace(1M) SARM
 idtune: attempts to set value of idtune(1M) SARM
 IEEE arithmetic. /scalbn: ieee_functions(3M) BSD
 IEEE exception trap handler ieee_handler(3M) BSD
 IEEE floating point definitions. floatingpoint(3) BSD
 IEEE floating-point environment/ fpgetround(3C) PRM
 ieee_functions, fp_class, isnan, ieee_functions(3M) BSD
 ieee_handler: IEEE exception ieee_handler(3M) BSD
 if: general properties of if(7) NUAG
 ifconfig: configure network ifconfig(1M) NUAG
 ifconfig: configure network ifconfig(1M) SARM
 ifdef'ed lines from C program/ undef(1) BSD
 II boot service daemon. bootserver(1M) MBRMAN
 II configuration file format. config(1M) MBRMAN
 II interconnect space. icsrd(1M) MBRMAN
 II messages in. mb2a_receive: mb2a_receive(3tai) TAI
 II messages in. mb2s_receive: mb2s_receive(3tai) TAI
 II transport endpoint.. mb2a_closeport(3tai) TAI
 II transport endpoint.. mb2a_openport(3tai) TAI
 II transport endpoint.. mb2s_openport(3tai) TAI
 II transport. mb2s_closeport: mb2s_closeport(3tai) TAI
 i-list information. ff(1M) SARM
 image archive of a filesystem. fimage(1M) SARM
 image file. core(4) PRM
 image file. core(4) SARM
 images. crash(1M) SARM
 images. pnch(4) SARM
 images of running processes. gcore(1) URM
 immediately. sdi_icmd: sdi_icmd(D3I) SCSI
 immedok, leaveok, setscreg,/ curs_outopts(3X) CHAR
 immune to hangups and quits. nohup(1) URM
 implementation-specific/ limits(4) PRM
 implementation-specific/ limits(4) SARM

Permuted Index

/dumbplot, gigiplot, hplot, implot, t300, t300s, t4013,/ plot(1G) BSD
 w: who is logged in, and what are they doing. w(1) BSD
 incoming MULTIBUS II messages in. mb2a_receive: Receives mb2a_receive(3tai) TAI
 incoming MULTIBUS II messages in. mb2s_receive: Receives mb2s_receive(3tai) TAI
 /of, or search for a text string in, message data bases. srchtxt(1) URM
 string on the processor in. /the bootstrap parameter initbp(1M) MBRMAN
 a compact list of users logged in. users: display users(1) BSD
 specified register of the board in. /writes a value into the ics_write(D3D) DDRM
 External Data Link driver for iNA961 Release 3.0.. edlina: edlina(1M) MBRMAN
 I/O port inb: read a byte from a 8-bit inb(D3D) DDRM
 incremental filesystem archive. incfile: create, restore an incfile(1M) SARM
 get a character and/ curs_inch: inch, winch, mvinch, mvwinch: curs_inch(3X) CHAR
 curs_inchstr: inchstr, inchnstr, winchstr, winchnstr,/ curs_inchstr(3X) CHAR
 winchnstr,/ curs_inchstr: inchstr, inchnstr, winchstr, curs_inchstr(3X) CHAR
 invoke recipient command for incoming mail. mail_pipe: mail_pipe(1M) SARM
 biff: give notice of incoming mail messages. biff(1) BSD
 /automatically respond to incoming mail messages. vacation(1) URM
 in. mb2a_receive: Receives incoming MULTIBUS II messages mb2a_receive(3tai) TAI
 in. mb2s_receive: Receives incoming MULTIBUS II messages mb2s_receive(3tai) TAI
 smtpd: receive incoming SMTP messages. smtpd(1M) SARM
 comsat, in.comsat: biff server. comsat(1M) NUAG
 comsat, in.comsat: biff server. comsat(1M) SARM
 ufsdump: incremental file system dump. ufsdump(1M) SARM
 ufsrestore: incremental file system restore. ufsrestore(1M) SARM
 incfile: create, restore an incremental filesystem archive. incfile(1M) SARM
 xrestore, xrestor: invoke XENIX incremental filesystem restorer. xrestore(1M) SARM
 dirent: file system independent directory entry. dirent(4) SARM
 operations. index, rindex: string index(3) BSD
 indxbib: create an inverted index to a bibliographic/ indxbib(1) BSD
 logins. last: indicate last user or terminal last(1) URM
 receipt of an orderly release indication. /acknowledge t_rcvrel(3N) NI
 receive a unit data error indication. t_rcvuderr: t_rcvuderr(3N) NI
 alarms and/or the "working" indicator. /application specific indicator(1F) CHAR
 specific alarms and/or the/ indicator: display application indicator(1F) CHAR
 syscall: indirect system call. syscall(3) BSD
 index to a bibliographic/ indxbib: create an inverted indxbib(1) BSD
 inet_makeaddr, inet_lnaof,/ inet: inet_addr, inet_network, inet(3N) NI
 inet: Internet protocol family. inet(7) NUAG
 inet: Internet protocol family. inet(7) SARM
 inet_addr, inet_network, inet(3N) NI
 inetd: Internet services daemon. inetd(1M) NI
 inetd: Internet services daemon. inetd(1M) NUAG
 inetd: Internet services daemon. inetd(1M) SARM
 database. inetd.conf: Internet servers inetd.conf(4) NUAG
 database. inetd.conf: Internet servers inetd.conf(4) SARM
 /inet_network, inet_makeaddr, inet_lnaof, inet_netof,/ inet(3N) NI
 inet: inet_addr, inet_network, inet_makeaddr, inet_lnaof,/ inet(3N) NI

/inet_makeaddr, inet_lnaof, inet_netof, inet_ntoa: Internet/ inet(3N) NI
 inet_lnaof,/ inet: inet_addr, inet_network, inet_makeaddr, inet(3N) NI
 /inet_lnaof, inet_netof, inet_ntoa: Internet address/ inet(3N) NI
 information server. fingerd, in.fingerd: remote user fingerd(1M) NUAG
 information server. fingerd, in.fingerd: remote user fingerd(1M) SARM
 terminfo descriptions. infocmp: compare or print out infocmp(1M) CHAR
 terminfo descriptions. infocmp: compare or print out infocmp(1M) SARM
 has occurred. pollwakeupp: inform a process that an event pollwakeupp(D3DK) DDRM
 dllerror: get diagnostic information. dllerror(3X) PRM
 ff (s5): display i-list information. ff(1M) SARM
 idcheck: returns selected information. idcheck(1M) SARM
 listusers: list user login information. listusers(1) URM
 nl_langinfo: language information. nl_langinfo(3C) PRM
 .ott: FACE object architecture information. ott(4) SARM
 rpcinfo: report RPC information. rpcinfo(1M) NI
 rpcinfo: report RPC information. rpcinfo(1M) NUAG
 setuname: changes machine information. setuname(1M) SARM
 siginfo: signal generation information. siginfo(5) PRM
 siginfo: signal generation information. siginfo(5) SARM
 sysfs: get file system type information. sysfs(2) PRM
 yp_update: changes yp information. yppupdate(3N) NI
 band of the queue. strqget: get information about a queue or strqget(D3DK) DDRM
 band of the/ strqset: change information about a queue or strqset(D3DK) DDRM
 remote users. finger: display information about local and finger(1) NUAG
 remote users. finger: display information about local and finger(1) URM
 utilization. getrusage: get information about resource getrusage(3) BSD
 the LP print/ lpstat: print information about the status of lpstat(1) URM
 remote machines. rusers: return information about users on rusers(3N) NI
 langinfo: language information constants. langinfo(5) PRM
 langinfo: language information constants. langinfo(5) SARM
 display mounted NFS resource information. dfmounts: dfmounts(1M) NUAG
 display mounted NFS resource information. dfmounts: dfmounts(1M) SARM
 retrieve kernel state information. drv_getparm: drv_getparm(D3DK) DDRM
 elf_nextscn: get section information. /elf_nextscn, elf_nextscn(3E) PRM
 copyright: copyright information file. copyright(4) SS
 prototype: package information file. prototype(4) SS
 mailcnfg: initialization information for mail and rmail. mailcnfg(4) SARM
 modes. tset: provide information for setting terminal tset(1) XNX
 binarsys: remote system information for the ckbinarsys/ binarsys(4) SARM
 contains terminal line settings information for ttymon. /file ttymon(4) NI
 t_rcvdis: retrieve information from disconnect. t_rcvdis(3N) NI
 specific asynchronous endpoint information.. /Gets mb2a_getinfo(3tai) TAI
 specific synchronous endpoint information.. /Gets mb2s_getinfo(3tai) TAI
 get numeric formatting information. localeconv: localeconv(3C) PRM
 list user and system login information. logins: logins(1M) SARM
 node's/ enetinfo: provides information on an ethernet enetinfo(1M) MBRMAN
 ccldinfo: line discipline information on the/ ccldinfo(1M) MBRMAN

Permuted Index

communications/ ccilinfo: line information on the ccilinfo(1M) MBRMAN
 ccisrvinfo: CCI server information on the/ ccisrvinfo(1M) MBRMAN
 ccisubinfo: subchannel information on the/ ccisubinfo(1M) MBRMAN
 usermod: modify a user's login information on the system. usermod(1M) SARM
 display software package information. pkginfo: pkginfo(1) SARM
 display software package information. pkginfo: pkginfo(1) SS
 display mounted resource information. rmntstat: rmntstat(1M) NUAG
 display mounted resource information. rmntstat: rmntstat(1M) SARM
 fingerd, in.fingerd: remote user information server. fingerd(1M) NUAG
 fingerd, in.fingerd: remote user information server. fingerd(1M) SARM
 fstatvfs: get file system information. statvfs, statvfs(2) PRM
 sysinfo: get and set system information strings. sysinfo(2) PRM
 STREAMS queue flow control information structure. qband: qband(D4DK) DDRM
 get protocol-specific service information. t_getinfo: t_getinfo(3N) NI
 and output port monitor-specific information. ttyadm: format ttyadm(1M) SARM
 server for changing YP information. ypupdated: ypupdated(1M) NI
 inittab: script for init. inittab(4) SARM
 init: initialize a device. init(D2D) DDRM
 init, telinit: process control init(1M) SARM
 initbp: initializes the initbp(1M) MBRMAN
 init_color, has_colors,/ curs_color(3X) CHAR
 initgroups: initialize the initgroups(3C) PRM
 initialization. init(1M) SARM
 initialization and manipulation/ curs_initscr(3X) CHAR
 initialization file. reinittab(1F) CHAR
 initialization information for mailcnfg(4) SARM
 initialization procedures. brc(1M) SARM
 init: initialize a device. init(D2D) DDRM
 management map. rmunit: initialize a private space rmunit(D3DK) DDRM
 terminfo database. tput: initialize a terminal or query tput(1) URM
 ownership. fixperm: correct or initialize file permissions and fixperm(1) XNX
 user. setup: initialize system for first setup(1M) SARM
 sdi_init: initialize the host adapter. sdi_init(D3I) SCSI
 group access list. initgroups: initialize the supplementary initgroups(3C) PRM
 permissions/ fixperm: correct or initialize XENIX file fixperm(1M) SARM
 parameter string on the/ initbp: initializes the bootstrap initbp(1M) MBRMAN
 socket. connect: initiate a connection on a connect(3N) NI
 a message to be sent to initiate a. /constructs mps_mk_solrply(D3D) DDRM
 /a message to be sent to initiate a solicited. mps_mk_sol(D3D) DDRM
 t_sndrel: initiate an orderly release. t_sndrel(3N) NI
 backup session. backup: initiate or control a system backup(1M) SARM
 popen, pclose: initiate pipe to/from a process. popen(3S) PRM
 filesystems, data/ restore: initiate restores of restore(1M) SARM
 curs_color: start_color, init_pair, init_color,/ curs_color(3X) CHAR
 isendwin,/ curs_initscr: initscr, newterm, endwin, curs_initscr(3X) CHAR
 random number/ random, srand, initstate, setstate: better random(3) BSD
 inittab: script for init. inittab(4) SARM

32-bit I/O port. inl: read a 32-bit word from a inl(D3D) DDRM
 fsync: synchronize a file's in-memory state with that on the/ fsync(2) PRM
 server. named, in.named: Internet domain name named(1M) NUAG
 server. named, in.named: Internet domain name named(1M) SARM
 mvinstr,/ curs_instr: instr, innstr, winstr, winnstr, curs_instr(3X) CHAR
 inode (s5): format of an s5 i-node. inode(4) SARM
 i-node. inode (bfs): format of a bfs inode(4) SARM
 fsirand: install random inode generation numbers. fsirand(1) BSD
 i-node. inode (s5): format of an s5 inode(4) SARM
 inode. inode (ufs): format of a ufs inode(4) SARM
 /number of free disk blocks and i-nodes for s5 file systems. df(1M) SARM
 between versions of a troff input file. /mark differences diffmk(1) BSD
 checknr: check nroff and troff input files; report possible/ checknr(1) BSD
 /vwscanw: convert formatted input from a curses widow. curs_scanw(3X) CHAR
 /typeahead: curses terminal input option control routines. curs_inopts(3X) CHAR
 .so requests from nroff or troff input. /resolve and eliminate soelim(1) BSD
 sscanf: convert formatted input. scanf, fscanf, scanf(3S) PRM
 ungetc: push character back onto input stream. ungetc(3S) PRM
 fread, fwrite: binary input/output. fread(3S) PRM
 poll: input/output multiplexing. poll(2) PRM
 poll: input/output multiplexing. poll(2) STRM
 stdio: standard buffered input/output package. stdio(3S) PRM
 clearerr, fileno: stream status inquiries. ferror, feof, ferror(3S) PRM
 uostat: uucp status inquiry and job control. uostat(1C) URM
 rwhod, in.rwhod: system status server. rwhod(1M) NUAG
 rwhod, in.rwhod: system status server. rwhod(1M) SARM
 mvwinsch: insert a/ curs_insch: insch, winsch, mvwinsch, curs_insch(3X) CHAR
 /deleteln, wdeleteln, insdelln, winsdelln, insertln,/ curs_deleteln(3X) CHAR
 /winsch, mvwinsch, mvwinsch: insert a character before the/ curs_insch(3X) CHAR
 insq: insert a message into a queue. insq(D3DK) DDRM
 /winsertln: delete and insert lines in a curses window. curs_deleteln(3X) CHAR
 bibliographic/ refer: expand and insert references from a refer(1) BSD
 under/ /mvwinsstr, mvwinsstr: insert string before character curs_insstr(3X) CHAR
 operations to service media insertion prompts. /with backup bkoper(1M) SARM
 and/ /insdelln, winsdelln, insertln, winsertln: delete curs_deleteln(3X) CHAR
 queue. insque, remque: insert/remove element from a insque(3C) PRM
 mvwinsstr,/ curs_instr: instr, insnr, winsstr, winsnstr, curs_insstr(3X) CHAR
 queue. insq: insert a message into a insq(D3DK) DDRM
 element from a queue. insque, remque: insert/remove insque(3C) PRM
 winsnstr, mvwinsstr,/ curs_instr: instr, insnr, winsstr, curs_insstr(3X) CHAR
 suspend process/ sigsuspend: install a signal mask and sigsuspend(2) PRM
 install: install commands. install(1M) PRM
 install: install commands. install(1M) SARM
 xinstall: install commands. xinstall(1M) XNX
 install: install files. install(1) BSD
 install: install commands. install(1M) PRM
 install: install commands. install(1M) SARM

loaders. sgib:	install: install files.	install(1) BSD
numbers. fsirand:	install MULTIBUS bootstrap	sgib(1M) MBRMAN
UNIX package. custom:	install random inode generation	fsirand(1) BSD
UNIX package. custom:	install specific portions of a	custom(1M) SARM
ypinit: build and	install specific portions of a	custom(1) XNX
pkgmk: produce an	install YP database.	ypinit(1M) NI
pkgchk: check accuracy of	installable package.	pkgmk(1) SS
pkgchk: check accuracy of	installation.	pkgchk(1M) SARM
/add a file to the software	installation.	pkgchk(1M) SS
admin:	installation database.	install(1M) SS
xinstall: XENIX	installation defaults file.	admin(4) SARM
software installation database.	installation shell script.	xinstall(1M) SARM
pwconv:	installf: add a file to the	installf(1M) SS
creatsem: create an	Installs and updates.	pwconv(1M) SARM
mvinstr, mvinnstr,/ curs_instr:	instance of a binary semaphore.	creatsem(2) XNX
program. talkd,	instr, innstr, winstr, winnstr,	curs_instr(3X) CHAR
program. talkd,	in.talkd: server for talk	talkd(1M) NUAG
abs, labs: return	in.talkd: server for talk	talkd(1M) SARM
a64l, l64a: convert between long	integer absolute value.	abs(3C) PRM
mtox, mfree: multiple precision	integer and base-64 ASCII/	a64l(3C) PRM
prompts for and validates an	integer arithmetic. /itom, xtom,	mp(3X) BSD
prompts for and validates an	integer. ckrange:	ckrange(1) SARM
sputl, sgetl: access long	integer. ckrange:	ckrange(1) SS
atol, atoi: convert string to	integer data in a/	sputl(3X) PRM
a prompt; verify and return an	integer. strtol, strtoul,	strtol(3C) PRM
a prompt; verify and return an	integer value. ckint: display	ckint(1) SARM
max: return the larger of two	integer value. ckint: display	ckint(1) SS
min: return the lesser of two	integers.	max(D3DK) DDRM
/ltol3: convert between 3-byte	integers.	min(D3DK) DDRM
between 3-byte integers and long	integers and long integers.	l3tol(3C) PRM
to service media/ bkoper:	integers. l3tol, ltol3: convert	l3tol(3C) PRM
system. mailx:	interact with backup operations	bkoper(1M) SARM
system consistency check and	interactive message processing	mailx(1) URM
nslookup: query name servers	interactive repair. /(ufs): file	fsock(1M) SARM
nslookup: query name servers	interactively.	nslookup(1M) NUAG
program. cscope:	interactively.	nslookup(1M) SARM
or writes a specified number of	interactively examine a C	cscope(1) PRM
board/ ics_find_rec: reads the	interconnect. ics_rdwr: reads	ics_rdwr(D3D) DDRM
board in/ ics_read: reads the	interconnect register of the	ics_find_rec(D3D) DDRM
board and/ ics_read: reads	interconnect register of the	ics_read(D3D) DDRM
board. ics_write: writes into	interconnect registers of the	ics_read(1M) MBRMAN
driver. ics:	interconnect registers of the	ics_write(1M) MBRMAN
utilities to access MULTIBUS II	Interconnect Space device	ics(1M) MBRMAN
/finds a specific record in the	interconnect space. /icsgetrec:	icsrd(1M) MBRMAN
console: STREAMS-based console	interconnect space of.	ics_find_rec(1M) MBRMAN
cram- CMOS RAM	interface.	console(7) STRM
	interface.	cram(7) SARM

lo: software loopback network
 lo: software loopback network
 lp: parallel port
 rtc: real time clock
 swap: swap administrative
 tty: controlling terminal
 module. timod: Transport
 cci: communications controller
 edsysadm: sysadm
 edsysadm: sysadm
 Access Command Environment
 tool. delsysadm: sysadm
 tool. delsysadm: sysadm
 ifconfig: configure network
 ifconfig: configure network
 cartridge magnetic tape streamer
 STREAMS/ tirdwr: Transport
 line asynchronous communications
 /Transport Interface read/write
 tcgetsid: general terminal
 extended general terminal
 using the. telnet: user
 using the. telnet: user
 administration. sysadm: visual
 logging and event tracing. log:
 ypprot_err: YP client
 /tgetstr, tgoto, tputs: curses
 of Internet Protocol network
 /tigetnum, tigetstr: curses
 getmajor: get major or
 getminor: get minor or
 sendmail: send mail over the
 /inet_netof, inet_ntoa:
 Protocol. ICMP:
 Protocol. ICMP:
 Protocol. ICMP:
 named, in.named:
 named, in.named:
 htable: convert DoD
 htable: convert DoD
 a host. gettable: get DoD
 a host. gettable: get DoD
 IP:
 IP:
 IP:
 inet:
 inet:
 interface. lo(7) NUAG
 interface. lo(7) SARM
 interface. lp(7) SARM
 interface. rtc(7) SARM
 interface. swap(1M) SARM
 interface. tty(7) SARM
 Interface cooperating STREAMS timod(7) STRM
 interface device driver. cci(1M) MBRMAN
 interface editing tool. edsysadm(1M) SARM
 interface editing tool. edsysadm(1M) SS
 Interface. /for the Framed face(1) URM
 interface menu or task removal delsysadm(1M) SARM
 interface menu or task removal delsysadm(1M) SS
 interface parameters. ifconfig(1M) NUAG
 interface parameters. ifconfig(1M) SARM
 interface. qt: QIC qt(7) SARM
 Interface read/write interface tirdwr(7) STRM
 interface STREAMS driver. /5 ports(7) STRM
 interface STREAMS module. tirdwr(7) STRM
 interface. /tcsetpgrp, termios(2) PRM
 interface. termiox: termiox(7) SARM
 interface to a remote system telnet(1) NUAG
 interface to a remote system telnet(1) URM
 interface to perform system sysadm(1M) SARM
 interface to STREAMS error log(7) STRM
 interface. /yperr_string, ypcnt(3N) NI
 interfaces (emulated) to the/ curs_termcap(3X) CHAR
 interfaces. /general properties if(7) NUAG
 interfaces to terminfo database. curs_terminfo(3X) CHAR
 internal major device number. getmajor(D3DK) DDRM
 internal minor device number. getminor(D3DK) DDRM
 internet. sendmail(1M) BSD
 Internet address manipulation. inet(3N) NI
 Internet Control Message ICMP(7) NI
 Internet Control Message ICMP(7) NUAG
 Internet Control Message ICMP(7) SARM
 Internet domain name server. named(1M) NUAG
 Internet domain name server. named(1M) SARM
 Internet format host table. htable(1M) NUAG
 Internet format host table. htable(1M) SARM
 Internet format host table from gettable(1M) NUAG
 Internet format host table from gettable(1M) SARM
 IP:
 IP:
 IP:
 inet:
 inet:

if: general properties of	Internet Protocol network/	if(7) NUAG
inetd.conf:	Internet servers database.	inetd.conf(4) NUAG
inetd.conf:	Internet servers database.	inetd.conf(4) SARM
services:	Internet services and aliases.	services(4) NUAG
services:	Internet services and aliases.	services(4) SARM
inetd:	Internet services daemon.	inetd(1M) NI
inetd:	Internet services daemon.	inetd(1M) NUAG
inetd:	Internet services daemon.	inetd(1M) SARM
Protocol. TCP:	Internet Transmission Control	TCP(7) NI
Protocol. TCP:	Internet Transmission Control	TCP(7) NUAG
Protocol. TCP:	Internet Transmission Control	TCP(7) SARM
UDP:	Internet User Datagram Protocol.	UDP(7) NI
UDP:	Internet User Datagram Protocol.	UDP(7) NUAG
UDP:	Internet User Datagram Protocol.	UDP(7) SARM
service. whois:	Internet user name directory	whois(1) NUAG
service. whois:	Internet user name directory	whois(1) URM
control, and restricted command	interpreter. /the standard, job	sh(1) URM
syntax. csh: shell command	interpreter with a C-like	csh(1) URM
pipe: create an	interprocess channel.	pipe(2) PRM
facilities status. ipc:	inter-process communication	ipc(1) URM
package. stdipc: ftok: standard	interprocess communication	stdipc(3C) PRM
intr: process a device	interrupt.	intr(D2D) DDRM
blocked signals and wait for	interrupt. /automatically release	sigpause(3) BSD
siginterrupt: allow signals to	interrupt system calls.	siginterrupt(3) BSD
spl: block/allow	interrupts.	spl(D3D) DDRM
sleep: suspend execution for an	interval.	sleep(1) URM
sleep: suspend execution for	interval.	sleep(3) BSD
sleep: suspend execution for	interval.	sleep(3C) PRM
busy-wait for specified	interval. drv_usecwait:	drv_usecwait(D3DK) DDRM
ualarm: schedule signal after	interval in microseconds.	ualarm(3) BSD
usleep: suspend execution for	interval in microseconds.	usleep(3) BSD
suspend execution for a short	interval. nap:	nap(2) XNX
setitimer: get/set value of	interval timer. getitimer,	getitimer(3C) PRM
server. tnamed,	in.tnamed: DARPA trivial name	tnamed(1M) NUAG
server. tnamed,	in.tnamed: DARPA trivial name	tnamed(1M) SARM
interrupt.	intr: process a device	intr(D2D) DDRM
/echo, noecho, halfdelay,	intrflush, keypad, meta,/	cursor_inopts(3X) CHAR
intro.	intro.	intro(2) XNX
maintenance commands and/	intro: introduction to	intro(1M) SARM
programming commands.	intro: introduction to	intro(1) PRM
miscellany.	intro: introduction to	intro(5) PRM
miscellany.	intro: introduction to	intro(5) SARM
and application programs.	intro: introduction to commands	intro(1) URM
formats.	intro: introduction to file	intro(4) PRM
formats.	intro: introduction to file	intro(4) SARM
and libraries.	intro: introduction to functions	intro(3) PRM
libraries.	intro: introduction to math	intro(3M) PRM

files. intro: introduction to special intro(7) SARM
 calls and error numbers. intro: introduction to system intro(2) PRM
 application programs. intro: introduction to commands and intro(1) URM
 intro: introduction to file formats. intro(4) PRM
 intro: introduction to file formats. intro(4) SARM
 libraries. intro: introduction to functions and intro(3) PRM
 commands and application/ intro: introduction to maintenance intro(1M) SARM
 intro: introduction to math libraries. intro(3M) PRM
 intro: introduction to miscellany. intro(5) PRM
 intro: introduction to miscellany. intro(5) SARM
 commands. intro: introduction to programming intro(1) PRM
 intro: introduction to special files. intro(7) SARM
 error numbers. intro: introduction to system calls and intro(2) PRM
 /(s5): generate path names versus i-numbers for s5 file systems. ncheck(1M) SARM
 /(ufs): generate pathnames versus i-numbers for ufs file systems. ncheck(1M) SARM
 a list of path names vs i-numbers. /(generic): generate ncheck(1M) SARM
 indxbib: create an inverted index to a/ indxbib(1) BSD
 idspace: investigates free space. idspace(1M) SARM
 /routines for invocation by forms. form_hook(3X) CHAR
 /routines for automatic invocation by menus. menu_hook(3X) CHAR
 fmli: invoke FMLI. fmli(1) CHAR
 fmli: invoke FMLI. fmli(1) URM
 incoming mail. mail_pipe: invoke recipient command for mail_pipe(1M) SARM
 filesystem/ xrestore, xrestor: invoke XENIX incremental xrestore(1M) SARM
 from a 16-bit I/O port. inw: read a 16 bit short word inw(D3D) DDRM
 strategy: perform block I/O. strategy(D2DK) DDRM
 /release buffer after block I/O and wakeup processes. biodone(D3DK) DDRM
 pending completion of block I/O. biowait: suspend processes biowait(D3DK) DDRM
 buf: block I/O data transfer structure. buf(D4DK) DDRM
 geterror: return I/O error. geterror(D3DK) DDRM
 select: synchronous I/O multiplexing. select(3C) NI
 inb: read a byte from a 8-bit I/O port. inb(D3D) DDRM
 outb: write a byte to an 8-bit I/O port. outb(D3D) DDRM
 a 32-bit word from a 32-bit I/O port. inl: read inl(D3D) DDRM
 16 bit short word from a 16-bit I/O port. inw: read a inw(D3D) DDRM
 a 32-bit long word to a 32-bit I/O port. outl: write outl(D3D) DDRM
 a 16-bit short word to a 16-bit I/O port. outw: write outw(D3D) DDRM
 write bytes from buffer to an I/O port. repoutsb: repoutsb(D3D) DDRM
 32-bit words from buffer to an I/O port. repoutsd: write repoutsd(D3D) DDRM
 16-bit words from buffer to an I/O port. repoutsw: write repoutsw(D3D) DDRM
 repinsb: read bytes from I/O port to buffer. repinsb(D3D) DDRM
 repinsd: read 32-bit words from I/O port to buffer. repinsd(D3D) DDRM
 repinsw: read 16-bit words from I/O port to buffer. repinsw(D3D) DDRM
 physiock: validate and issue raw I/O request. physiock(D3D) DDRM
 units. dma_pageio: break up an I/O request into manageable dma_pageio(D3DK) DDRM
 uio: scatter/gather I/O request structure. uio(D4DK) DDRM
 data storage structure for I/O using DMA. dma_buf: dma_buf(D4D) DDRM

Permuted Index

data storage structure for
 data storage structure for
 streamio: STREAMS
 device.
 sockets. sockio:
 for I/O using uio(D4DK).

 semaphore set, or shared memory/
 communication facilities/
 /islower, isupper, isalpha,
 /isxdigit, islower, isupper,
 /ispunct, isprint, isgraph,
 descriptor.
 ttyname,
 controller disk device/ i224a:
 controller tape device/ i224atp:
 communications controller/ i410:
 device driver. i530:
 tape device driver. i214tp:
 controller device driver. i258:
 controller tape device/ i258tp:
 controller. i546:
 disk device driver. i214:
 serial-port device/ i354:
 /isalpha, isalnum, isspace,
 isupper, isalpha,/ ctype:
 character buffer is encrypted.
 /initscr, newterm, endwin,
 /isctrl, ispunct, isprint,
 /touchline, utoouchwin, wtouchln,
 ctype: isdigit, isxdigit,
 state.
 ieee_functions, fp_class,
 fpclass, unordered: determine/
 unordered: determine/ isnan,
 unordered:/ isnan, isnand,
 /isspace, isctrl, ispunct,
 /isalnum, isspace, isctrl,
 /isupper, isalpha, isalnum,
 system:
 issue:
 file.
 physiock: validate and
 /isdigit, isxdigit, islower,

I/O using DMA. dma_cb: dma_cb(D4D) DDRM
 I/O using uio(D4DK). iovec: iovec(D4DK) DDRM
 ioctl commands. streamio(7) STRM
 ioctl: control a character ioctl(D2DK) DDRM
 ioctl: control device. ioctl(2) PRM
 ioctls that operate directly on sockio(7) NI
 iovec: data storage structure iovec(D4DK) DDRM
 IP: Internet Protocol. IP(7) NI
 IP: Internet Protocol. IP(7) NUAG
 IP: Internet Protocol. IP(7) SARM
 ipcrm: remove a message queue, ipcrm(1) URM
 ipc: report inter-process ipc(1) URM
 isalnum, isspace, isctrl,/ ctype(3C) PRM
 isalpha, isalnum, isspace,/ ctype(3C) PRM
 isascii: character handling. ctype(3C) PRM
 isastream: test a file isastream(3C) STRM
 isatty: find name of a terminal. ttyname(3C) PRM
 iSBC 186/224A peripheral i224a(1M) MBRMAN
 iSBC 186/224A peripheral i224atp(1M) MBRMAN
 iSBC 186/410 serial i410(1M) MBRMAN
 iSBC 186/530 Ethernet controller i530(1M) MBRMAN
 iSBC 214 peripheral controller i214tp(1M) MBRMAN
 iSBC 386/258 peripheral i258(1M) MBRMAN
 iSBC 386/258 peripheral i258tp(1M) MBRMAN
 iSBC 546 multi-port serial i546(1M) MBRMAN
 iSBCR 214 peripheral controller i214(1M) MBRMAN
 iSBX 354 dual channel i354(1M) MBRMAN
 isctrl, ispunct, isprint,/ ctype(3C) PRM
 isdigit, isxdigit, islower, ctype(3C) PRM
 isencrypt: determine whether a isencrypt(3G) PRM
 isendwin, set_term, delscreen:/ curs_initscr(3X) CHAR
 isgraph, isascii: character/ ctype(3C) PRM
 is_linetouched, is_wintouched:/ curs_touch(3X) CHAR
 islower, isupper, isalpha,/ ctype(3C) PRM
 ismpx: return windowing terminal ismpx(1) URM
 isnan, copysign, scalbn:/ ieee_functions(3M) BSD
 isnan, isnand, isnanf, finite, isnan(3C) PRM
 isnand, isnanf, finite, fpclass, isnan(3C) PRM
 isnanf, finite, fpclass, isnan(3C) PRM
 isprint, isgraph, isascii:/ ctype(3C) PRM
 ispunct, isprint, isgraph,/ ctype(3C) PRM
 isspace, isctrl, ispunct,/ ctype(3C) PRM
 issue a shell command. system(3S) PRM
 issue identification file. issue(4) SARM
 issue: issue identification issue(4) SARM
 issue raw I/O request. physiock(D3D) DDRM
 isupper, isalpha, isalnum,/ ctype(3C) PRM

/wtouchln, is_linetouched, isalpha,/ ctype: isdigit, prompt for and return a menu
 /item_visible: tell if menus /item_description: get menus
 item_value: set and get menus /set_menu_items, menu_items,
 menu_item_name: item_name, is_wintouched: curses refresh/ curs_touch(3X) CHAR
 isxdigit, islower, isupper, ctype(3C) PRM
 item. ckitem: build a menu; ckitem(1) SARM
 item. ckitem: build a menu; ckitem(1) SS
 item is visible. menu_item_visible(3X) CHAR
 item name and description. menu_item_name(3X) CHAR
 item values. /set_item_value, menu_item_value(3X) CHAR
 item_count: connect and/ menu_items(3X) CHAR
 item_description: get menus item/
 menu_item_name(3X) CHAR
 menu_index: set and get current ... menu_item_current(3X) CHAR
 item_init, set_item_term,/ menu_hook(3X) CHAR
 item_name, item_description: get ... menu_item_name(3X) CHAR
 item_opts_off,. menu_item_opts(3X) CHAR
 item_opts_on, item_opts_off,. menu_item_opts(3X) URM
 items. news(1) URM
 items. getitems: return getitems(1F) CHAR
 items. /item_userptr: associate ... menu_item_userptr(3X) CHAR
 items. /new_item, free_item: menu_item_new(3X) CHAR
 items to and from menus. menu_items(3X) CHAR
 items. /top_row, item_index: menu_item_current(3X) CHAR
 item_term, set_menu_init,/ menu_hook(3X) CHAR
 application/ /set_item_userptr, item_userptr: associate menu_item_userptr(3X) CHAR
 item values. /set_item_value, item_value: set and get menus menu_item_value(3X) CHAR
 is visible. menu_item_visible: item_visible: tell if menus item menu_item_visible(3X) CHAR
 pow, gcd, rpow, msqrt, sddiv, itom, xtom, mtox, mfree:/ /mout, mp(3X) BSD
 functions. bessel: j0, j1, jn, y0, y1, yn: Bessel bessel(3M) PRM
 functions. bessel: j0, jn, y0, y1, yn: Bessel bessel(3M) PRM
 windowing terminal. jagent: host control of jagent(5) PRM
 windowing terminal. jagent: host control of jagent(5) SARM
 functions. bessel: j0, j1, jn, y0, y1, yn: Bessel bessel(3M) PRM
 uustat: uucp status inquiry and job control. uustat(1C) URM
 jsh, rsh: shell, the standard, job control, and restricted/ sh, sh(1) URM
 lpr: send a job to the printer. lpr(1) BSD
 lprm: remove jobs from the printer queue. lprm(1) BSD
 display the queue of printer jobs. lpq: lpq(1) BSD
 times. atq: display the jobs queued to run at specified atq(1) URM
 atrm: remove jobs spooled by at or batch. atrm(1) URM
 operator. join: relational database join(1) URM
 /lrand48, nrand48, mrand48, jrand48, srand48, seed48,/ drand48(3C) PRM
 job control, and restricted/ sh, jsh, rsh: shell, the standard, sh(1) URM
 terminal. jterm: reset layer of windowing jterm(1) URM
 idbuild: build new UNIX System jwin: print size of layer. jwin(1) URM
 idmkunix: build new UNIX System kernel. idbuild(1M) SARM
 get page frame number for kernel. idmkunix(1M) SARM
 between address locations in the kernel address. hat_getkpfnum: hat_getkpfnum(D3K) DDRM
 kernel. bcopy: copy data bcopy(D3DK) DDRM

idconfig: produce a new structure. uiomove: copy
 kmem_alloc: allocate space from /allocate and clear space from free previously allocated
 drv_getparm: retrieve physical/ kvtophys: convert
 chkey: change user encryption
 chkey: change user encryption
 makekey: generate encryption
 publickey: public
 publickey: public
 retrieve public or secret
 newkey: create a new
 newkey: create a new
 decrypt and store secret
 decrypt and store secret
 keyboard: system console
 strings from curses terminal
 characters from curses terminal
 keyboard.
 /getnetname, host2netname,
 /key_decryptsession,
 /key_encryptsession,
 secret key.
 secret key.
 putwin,/ curs_util: unctrl,
 noecho, halfdelay, intrflush,
 print the value of one or more
 for storing public and private
 for storing public and private
 public and private keys.
 public and private keys.
 /key_encryptsession, key_gendes,
 prompts for and validates a
 prompts for and validates a
 apropos: locate commands by
 pages; find reference pages by
 a one-line summary about a
 killall:
 or a group of processes.
 default.
 processes.
 /erasechar, has_ic, has_il,
 group.
 mem,
 kernel free memory.
 kernel configuration. idconfig(1M) SARM
 kernel data using uio(D4DK) uiomove(D3DK) DDRM
 kernel free memory. kmem_alloc(D3DK) DDRM
 kernel free memory. kmem_zalloc(D3DK) DDRM
 kernel memory. kmem_free: kmem_free(D3DK) DDRM
 kernel state information. drv_getparm(D3DK) DDRM
 kernel virtual address to kvtophys(D3D) DDRM
 key. chkey(1) NI
 key. chkey(1) NUAG
 key. makekey(1) URM
 key database. publickey(4) NI
 key database. publickey(4) NUAG
 key. /getsecretkey: publickey(3N) NI
 key in the publickey database. newkey(1M) NI
 key in the publickey database. newkey(1M) NUAG
 key. keylogin: keylogin(1) NI
 key. keylogin: keylogin(1) NUAG
 keyboard. keyboard(7) SARM
 keyboard. /get character curs_getstr(3X) CHAR
 keyboard. /get (or push back) curs_getch(3X) CHAR
 keyboard: system console keyboard(7) SARM
 key_decryptsession,/ secure_rpc(3N) NI
 key_encryptsession, key_gendes,/ secure_rpc(3N) NI
 key_gendes, key_setsecret,/ secure_rpc(3N) NI
 keylogin: decrypt and store keylogin(1) NI
 keylogin: decrypt and store keylogin(1) NUAG
 keyname, filter, use_env, curs_util(3X) CHAR
 keypad, meta, nodelay,/ /echo, curs_inopts(3X) CHAR
 keys from a YP map. ypmatch: ypmatch(1) NI
 keys. keyserv: server keyserv(1M) NI
 keys. keyserv: server keyserv(1M) NUAG
 keyserv: server for storing keyserv(1M) NI
 keyserv: server for storing keyserv(1M) NUAG
 key_setsecret, netname2host,/ secure_rpc(3N) NI
 keyword. ckkeywd: ckkeywd(1) SARM
 keyword. ckkeywd: ckkeywd(1) SS
 keyword lookup. apropos(1) BSD
 keyword. /reference manual man(1) BSD
 keyword. whatis: display whatis(1) BSD
 kill all active processes. killall(1M) SARM
 kill: send a signal to a process kill(2) PRM
 kill: terminate a process by kill(1) URM
 killall: kill all active killall(1M) SARM
 killchar, longname, termattr,/ curs_termattr(3X) CHAR
 killpg: send signal to a process killpg(3) BSD
 kmem: core memory. mem(7) SARM
 kmem_alloc: allocate space from kmem_alloc(D3DK) DDRM

allocated kernel memory. **knmem_free**: free previously **knmem_free(D3DK)** DDRM
 space from kernel free memory. **knmem_zalloc**: allocate and clear **knmem_zalloc(D3DK)** DDRM
 command and/ **ksh**, **rksh**: KornShell, a standard/restricted **ksh(1)** URM
 standard/restricted command and/ **ksh**, **rksh**: KornShell, a **ksh(1)** URM
 address to physical address. **kvtophys**: convert kernel virtual **kvtophys(D3D)** DDRM
 3-byte integers and long/ **l3tol**, **l3tol3**: convert between **l3tol(3C)** PRM
 integer and base-64 ASCII/ **a64l**, **l64a**: convert between long **a64l(3C)** PRM
slk_attroff: curses soft label routines. /**slk_attrset**, **curlk(3X)** CHAR
 labels for file systems. **labelit** (generic): provide **labelit(1M)** SARM
s5 file systems. **labelit** (s5): provide labels for **labelit(1M)** SARM
 for ufs file systems. **labelit** (ufs): provide labels **labelit(1M)** SARM
labelit (generic): provide labels for file systems. **labelit(1M)** SARM
labelit (s5): provide labels for s5 file systems. **labelit(1M)** SARM
labelit (ufs): provide labels for ufs file systems. **labelit(1M)** SARM
 value. **abs**, **labs**: return integer absolute **abs(3C)** PRM
 constants. **langinfo**: language information **langinfo(5)** PRM
 constants. **langinfo**: language information **langinfo(5)** SARM
 command and programming **language**. /a standard/restricted **ksh(1)** URM
 pattern scanning and processing **language**. **awk**: **awk(1)** URM
 arbitrary-precision arithmetic **language**. **bc**: **bc(1)** URM
nl_types: native **language** data types. **nl_types(5)** PRM
nl_types: native **language** data types. **nl_types(5)** SARM
nl_langinfo: **language** information. **nl_langinfo(3C)** PRM
langinfo: **language** information constants. **langinfo(5)** PRM
langinfo: **language** information constants. **langinfo(5)** SARM
 pattern scanning and processing **language**. **nawk**: **nawk(1)** URM
strftime: **language** specific strings. **strftime(4)** PRM
strftime: **language** specific strings. **strftime(4)** SARM
max: return the larger of two integers. **max(D3DK)** DDRM
 terminal logins. **last**: indicate last user or **last(1)** URM
 executed, in reverse order. **lastcomm**: show the last commands **lastcomm(1)** BSD
chargefee, **ckpacct**, **dodisk**, **lastlogin**, **monacct**, **nulladm**,/ **acctsh(1M)** SARM
at, **batch**: execute commands at a later time. **at(1)** URM
jwin: print size of a layer. **jwin(1)** URM
shl: shell layer manager. **shl(1)** URM
 terminals. **layers**: layer multiplexor for windowing **layers(1)** URM
jterm: reset layer of windowing terminal. **jterm(1)** URM
 login entry to show current layer. **relogin**: rename **relogin(1M)** SARM
 login entry to show current layer. **relogin**: rename **relogin(1M)** URM
 windowing terminals. **layers**: layer multiplexor for **layers(1)** URM
host and windowing terminal/ **layers**: protocol used between **layers(5)** PRM
host and windowing terminal/ **layers**: protocol used between **layers(5)** SARM
ls, **lc**: list contents of directory. **ls(1)** XNX
 group of a file. **chown**, **lchown**, **fchown**: change owner and **chown(2)** PRM
/setspent, **endspent**, **fgetspent**, **lckpwwdf**, **ulckpwwdf**: manipulate/ **getspent(3C)** PRM
/jrand48, **srand48**, **seed48**, **lcong48**: generate uniformly/ **drand48(3C)** PRM
 editor. **ld**: link editor, dynamic link **ld(1)** BSD

files. ld: link editor for object ld(1) PRM
 ldd: list dynamic dependencies. ldd(1) PRM
 ldexp, logb, modf, modff, frexp(3C) PRM
 ldiv: compute the quotient and div(3C) PRM
 ldsysdump: load system dump from ldsysdump(1M) SARM
 ldterm: standard STREAMS ldterm(7) STRM
 leaveok, setscreg, wsetscreg,/ curs_outopts(3X) CHAR
 legal user shells. getusershell(3) BSD
 length for a solicited reply.. mps_get_reply_len(D3D) DDRM
 length of time. /execute timeout(D3DK) DDRM
 length. truncate, ftruncate: truncate(3C) PRM
 lesser of two integers. min(D3DK) DDRM
 letter from argument vector. getopt(3C) PRM
 level hard disk formatter.. hdformat(1M) MBRMAN
 levels for an application for addseverity(3C) PRM
 lex: generate programs for lex(1) PRM
 lexical tasks. lex: lex(1) PRM
 lfind: linear search and update. lsearch(3C) PRM
 lgamma: log gamma function. gamma(3M) PRM
 libraries. intro(3M) PRM
 libraries. tam(3X) CHAR
 libraries. intro: intro(3) PRM
 library. ar(1) PRM
 library. ar(1) URM
 library. elf(3E) PRM
 library. t_sync(3N) NI
 library and application/ elf_version(3E) PRM
 library and object module/ download(1M) MBRMAN
 library. /curses interfaces curs_termcap(3X) CHAR
 library. libwindows: libwindows(3X) PRM
 library. lorder: find lorder(1) PRM
 library routines for client side rpc_clnt_auth(3N) NI
 library routines for client side rpc_clnt_calls(3N) NI
 library routines for dealing/ rpc_clnt_create(3N) NI
 library routines for dealing/ rpc_svc_create(3N) NI
 library routines for external xdr(3N) NI
 library routines for external xdr_admin(3N) NI
 library routines for external/ xdr_complex(3N) NI
 library routines for external/ xdr_create(3N) NI
 library routines for external xdr_simple(3N) NI
 library routines for registering/ rpc_svc_calls(3N) NI
 library routines for remote rpc(3N) NI
 library routines for remote rpc_xdr(3N) NI
 library routines for RPC/ rpc_svc_reg(3N) NI
 library routines for RPC bind rpcbind(3N) NI
 library routines for secure/ secure_rpc(3N) NI
 library routines for server side rpc_svc_err(3N) NI

t_alloc: allocate a	library structure.	t_alloc(3N) NI
t_free: free a	library structure.	t_free(3N) NI
function library.	libwindows: windowing terminal	libwindows(3X) PRM
driver identification and	limit value structure. /STREAMS ... module_info(D4DK) DDRM	
ulimit: get and set user	limits.	ulimit(2) PRM
implementation-specific/	limits: header file for	limits(4) PRM
implementation-specific/	limits: header file for	limits(4) SARM
line: read one	line.	line(1) URM
interface STREAMS/	line asynchronous communications	ports(7) STRM
ports: 5	line connection. dial:	dial(3C) NI
establish an outgoing terminal	line discipline for unique	connld(7) STRM
stream connections. connld:	line discipline from a line on	ccunbind(1M) MBRMAN
the/	line discipline. getty: set	getty(1M) SARM
ccunbind: unbinds a	line discipline information on	ccldinfo(1M) MBRMAN
terminal type, modes, speed, and	line discipline list on the	ccldlist(1M) MBRMAN
the communications/	line discipline module. ldterm:	ldterm(7) STRM
ccldinfo:	line discipline program on the	ccifree(1M) MBRMAN
communications/	line discipline program on to	ccload(1M) MBRMAN
ccldlist:	line discipline to a line on the	ccbind(1M) MBRMAN
standard STREAMS terminal	line information on the	ccilinfo(1M) MBRMAN
communications/	line. message: puts	message(1F) CHAR
ccifree: frees a	line numbering filter.	nl(1) URM
the/	line of a file. cut:	cut(1) URM
ccload: load a	line of a file. fmlcut:	fmlcut(1F) CHAR
communication/	line on. /disassociates	ccidetach(1M) MBRMAN
ccbind: binds a	line on the communication/	ccbind(1M) MBRMAN
communications/	line on the communications/	ccunbind(1M) MBRMAN
ccilinfo:	line printer control program.	lpc(1M) BSD
its arguments on FMLI message	line: read one line.	line(1) URM
nl:	line. readfile, longline:	readfile(1F) CHAR
cut out selected fields of each	line settings and hunt sequences	sttydefs(1M) SARM
cut out selected fields of each	line settings information for/	ttydefs(4) NI
this host with a subchannel on a	line. strip: strip	strip(1) PRM
/binds a line discipline to a	line to a new host on the/	cciswitch(1M) MBRMAN
/unbinds a line discipline from a	line to a new host on the/	ttyswitch(1M) MBRMAN
lpc:	linear search and update.	lsearch(3C) PRM
reads file, gets longest	line-by-line execution count	lprof(1) PRM
for TTY/	line-feeds.	col(1) URM
sttydefs: maintain	line-printer. nroff:	nroff(1) BSD
ttydefs: file contains terminal	lineprinter ripple pattern.	lptest(1) BSD
symbol table, debugging and	lines.	fold(1) URM
/switch a subchannel on a	lines common to two sorted	comm(1) URM
ttyswitch: switches a	lines from C program source.	unifdef(1) BSD
lsearch, lfind:	lines in a curses window.	curs_deleteln(3X) CHAR
profile data. lprof: display	lines in a file.	uniq(1) URM
col: filter reverse	lines in a sorted list. /words	look(1) BSD
format documents for display or	lines of files.	head(1) URM
lptest: generate		
fold: fold long		
files. comm: select or reject		
/resolve and remove ifdef'ed		
/winsertln: delete and insert		
uniq: report repeated		
in the system dictionary or		
head: display first few		

of several files or subsequent
subsequent/ paste: merge same
refresh curses windows and
borders, horizontal and vertical
directories. link, unlink:
3.0.. edlina: External Data
ld: link editor, dynamic
editor. ld:
ld:
link editor for object files.
In:
In:

read the value of a symbolic
link:
symlink: make a symbolic
files and directories.
blocks.
slink: streams
slink: streams
and/ /new_field, dup_field,
routines. /set_fieldtype_choice,
a.out: ELF (Executable and
ln: make hard or symbolic

nlist: get entries from name
stdarg: handle variable argument
from remote systems. dfshares:
from remote systems. dfshares:
remote or local/ dfshares:
remote or local/ dfshares:
from remote systems. dfshares:
from remote systems. dfshares:
change or display an exception
ls:
ls, lc:
ldd:
for a file/ ff (generic):
dictionary or lines in a sorted
ttsrch: directory search
set supplementary group access
the supplementary group access
write the manufacturer's defect
dispgid: displays a
dispgid: displays a
dispuuid: displays a
dispuuid: displays a
nm: print name
items. getitems: return a
lines of one file. /same lines
lines of several files or
lines. /redrawwin, wredrawln:
lines. /wvline: create curses
link and unlink files and
Link driver for iNA961 Release
link editor.
link editor, dynamic link
link editor for object files.
link files.
link: link to a file.
link. readlink:
link to a file.
link to a file.
link, unlink: link and unlink
linkb: concatenate two message
linker.
linker.
link_field, free_field,: create
link_fieldtype: forms fieldtype
Linking Format) files.
links to files.
lint: a C program checker.
list.
list.
list available NFS resources
list available NFS resources
list available resources from
list available resources from
list available RFS resources
list available RFS resources
list. bkexcept:
list contents of directory.
list contents of directory.
list dynamic dependencies.
list file names and statistics
list. /find words in the system
list for ttyname.
list IDs. /setgroups: get or
list. initgroups: initialize
list.. mdl: read and optionally
list of all valid group names.
list of all valid group names.
list of all valid user names.
list of all valid user names.
list of an object file.
list of currently marked menu
paste(1) URM
paste(1) URM
curs_refresh(3X) CHAR
curs_border(3X) CHAR
link(1M) SARM
edlina(1M) MBRMAN
ld(1) BSD
ld(1) BSD
ld(1) PRM
ln(1) URM
link(2) PRM
readlink(2) PRM
link(2) PRM
symlink(2) PRM
link(1M) SARM
linkb(D3DK) DDRM
slink(1M) NUAG
slink(1M) SARM
form_field_new(3X) CHAR
form_fieldtype(3X) CHAR
a.out(4) PRM
ln(1) BSD
lint(1) PRM
nlist(3E) PRM
stdarg(5) PRM
dfshares(1M) NUAG
dfshares(1M) SARM
dfshares(1M) NUAG
dfshares(1M) SARM
dfshares(1M) NUAG
dfshares(1M) SARM
dfshares(1M) SARM
bkexcept(1M) SARM
ls(1) URM
ls(1) XNX
ldd(1) PRM
ff(1M) SARM
look(1) BSD
ttsrch(4) SARM
getgroups(2) PRM
initgroups(3C) PRM
mdl(1M) MBRMAN
dispgid(1) SARM
dispgid(1) SS
dispuuid(1) SARM
dispuuid(1) SS
nm(1) PRM
getitems(1F) CHAR

mps_free_dmabuf: frees a list of data buffer/ mps_free_dmabuf(D3D) DDRM
 /returns a pointer to a list of data buffer/ mps_get_dmabuf(D3D) DDRM
 ncheck (generic): generate a list of path names vs i-numbers. ncheck(1M) SARM
 available on/ uuglist: print the list of service grades that are uuglist(1C) URM
 addseverity: build a list of severity levels for an/ addseverity(3C) PRM
 users: display a compact list of users logged in. users(1) BSD
 ccildlist: line discipline list on the communications/ ccildlist(1M) MBRMAN
 directory. ls: list the contents of a ls(1) BSD
 information. logins: list user and system login logins(1M) SARM
 listusers: list user login information. listusers(1) URM
 handle variable argument list. varargs: varargs(5) PRM
 output of a variable argument list. /vsprintf: print formatted vprintf(3S) PRM
 device group. listdgrp: lists members of a listdgrp(1M) SARM
 t listen: listen for a connect request. t_listen(3N) NI
 socket. listen: listen for connections on a listen(3N) NI
 on a socket. listen: listen for connections listen(3N) NI
 listen: network listener daemon. listen(1M) SARM
 listener daemon. listen(1M) SARM
 get client's data passed via the listener. nlsgetcall: nlsgetcall(3N) NI
 nlsadmin: network listener service administration. nlsadmin(1M) SARM
 nlsrequest: format and send listener service request/ nlsrequest(3N) NI
 xargs: construct argument list(s) and execute command. xargs(1) URM
 devattr: lists device attributes. devattr(1M) SARM
 contain devices that/ getdgrp: lists device groups which getdgrp(1M) SARM
 getdev: lists devices based on criteria. getdev(1M) SARM
 listdgrp: lists members of a device group. listdgrp(1M) SARM
 STREAMS/ autopush: configure lists of automatically pushed autopush(1M) SARM
 STREAMS/ autopush: configure lists of automatically pushed autopush(1M) STRM
 information. listusers: list user login listusers(1) URM
 system. volcopy (ufs): make a literal copy of a ufs file volcopy(1M) SARM
 system. volcopy (s5): make a literal copy of an s5 file volcopy(1M) SARM
 volcopy (generic): make literal copy of file system. volcopy(1M) SARM
 ln: link files. ln(1) URM
 to files. ln: make hard or symbolic links ln(1) BSD
 interface. lo: software loopback network lo(7) NUAG
 interface. lo: software loopback network lo(7) SARM
 on to the/ cciload: load a line discipline program cciload(1M) MBRMAN
 diskettes. ldsysdump: load system dump from floppy ldsysdump(1M) SARM
 sgib: install MULTIBUS bootstrap loaders. sgib(1M) MBRMAN
 object module/ download: loads 8086 absolute library and download(1M) MBRMAN
 display information about local and remote users. finger: finger(1) NUAG
 display information about local and remote users. finger: finger(1) URM
 ruptime: show host status of local machines. ruptime(1) NUAG
 ruptime: show host status of local machines. ruptime(1) URM
 rusers: who's logged in on local machines. rusers(1) NI
 rusers: who's logged in on local machines. rusers(1) NUAG
 rwho: who's logged in on local machines. rwho(1) NUAG

rwho: who's logged in on
 mounting by remote/ share: make
 mounting by remote/ share: make
 for mounting by/ unshare: make
 for mounting by/ unshare: make
 set, unset: set and unset
 mounting by remote/ share: make
 mounting by remote/ share: make
 mounting by/ unshare: make
 mounting by/ unshare: make
 mounting by remote/ share: make
 mounting by remote/ share: make
 for mounting by/ unshare: make
 for mounting by/ unshare: make
 resources from remote or
 resources from remote or
 modify and query a program's
 formatting information.
 tzset: convert date and/ ctime,
 pathname or alias. which:
 lookup. apropos:
 end, etext, edata: last
 bcopy: copy data between address
 memory. lock:
 lockd: network
 process, text, or data. plock:
 memory.
 reading or writing. locking:
 mlockall, munlockall:
 memory. mlock, munlock:

 maillock: manage
 region for reading or writing.
 lockf: record
 file for syslogd system
 strlog: submit messages to the
 gamma, lgamma:
 newgrp:
 newgrp:
 logging and event tracing.
 powf, sqrt,/ exp, expf, cbrt,
 add entries to the system
 loginlog:
 setlogmask: control system
 syslogd:
 exp, expf, cbrt, log, logf,
 local machines. rwho(1) URM
 local NFS resource available for share(1M) NUAG
 local NFS resource available for share(1M) SARM
 local NFS resource unavailable unshare(1M) NUAG
 local NFS resource unavailable unshare(1M) SARM
 local or global environment/ set(1F) CHAR
 local resource available for share(1M) NUAG
 local resource available for share(1M) SARM
 local resource unavailable for unshare(1M) NUAG
 local resource unavailable for unshare(1M) SARM
 local RFS resource available for share(1M) NUAG
 local RFS resource available for share(1M) SARM
 local RFS resource unavailable unshare(1M) NUAG
 local RFS resource unavailable unshare(1M) SARM
 local systems. /list available dfshares(1M) NUAG
 local systems. /list available dfshares(1M) SARM
 locale. setlocale: setlocale(3C) PRM
 localeconv: get numeric localeconv(3C) PRM
 localtime, gmtime, asctime, ctime(3C) PRM
 locate a command; display its which(1) BSD
 locate commands by keyword apropos(1) BSD
 locations in program. end(3C) PRM
 locations in the kernel. bcopy(D3DK) DDRM
 lock a process in primary lock(2) XNX
 lock daemon. lockd(1M) NUAG
 lock into memory or unlock plock(2) PRM
 lock: lock a process in primary lock(2) XNX
 lock or unlock a file region for locking(2) XNX
 lock or unlock address space. mlockall(3C) PRM
 lock (or unlock) pages in mlock(3C) PRM
 lockd: network lock daemon. lockd(1M) NUAG
 lockf: record locking on files. lockf(3C) PRM
 lockfile for user's mailbox. maillock(3X) PRM
 locking: lock or unlock a file locking(2) XNX
 locking on files. lockf(3C) PRM
 log daemon. /configuration syslog.conf(4) BSD
 log driver. strlog(D3DK) DDRM
 log gamma function. gamma(3M) PRM
 log in to a new group. newgrp(1M) SARM
 log in to a new group. newgrp(1M) URM
 log: interface to STREAMS error log(7) STRM
 log, logf, log10, log10f, pow, exp(3M) PRM
 log. logger: logger(1) BSD
 log of failed login attempts. loginlog(4) SARM
 log. syslog, openlog, closelog, syslog(3) BSD
 log system messages. syslogd(1M) BSD
 log10, log10f, pow, powf, sqrt,/ exp(3M) PRM

/expf, cbrt, log, logf, log10, log10f, pow, powf, sqrt, sqrtf:/ exp(3M) PRM
 /powf, sqrt, sqrtf: exponential, logarithm, power, square root/ exp(3M) PRM
 scalb:/ frexp, ldexp, logb, modf, modff, nextafter, frexp(3C) PRM
 sqrt,/ exp, expf, cbrt, log, logf, log10, log10f, pow, powf, exp(3M) PRM
 doing. w: who is logged in, and what are they w(1) BSD
 rusers: who's logged in on local machines. rusers(1) NI
 rusers: who's logged in on local machines. rusers(1) NUAG
 rwho: who's logged in on local machines. rwho(1) NUAG
 rwho: who's logged in on local machines. rwho(1) URM
 display a compact list of users: logged in. users: users(1) BSD
 system log. logger: add entries to the logger(1) BSD
 strclean: STREAMS error logger cleanup program. strclean(1M) SARM
 strclean: STREAMS error logger cleanup program. strclean(1M) STRM
 strerr: STREAMS error logger daemon. strerr(1M) SARM
 strerr: STREAMS error logger daemon. strerr(1M) STRM
 log: interface to STREAMS error logging and event tracing. log(7) STRM
 size: return size of logical device. size(D2DK) DDRM
 rlogin: remote login. rlogin(1) NUAG
 rlogin: remote login. rlogin(1) URM
 loginlog: log of failed login attempts. loginlog(4) SARM
 netrc: file for ftp remote login data. netrc(4) NUAG
 netrc: file for ftp remote login data. netrc(4) SARM
 login: login default file. login(4) SARM
 layer. relogin: rename login entry to show current relogin(1M) SARM
 layer. relogin: rename login entry to show current relogin(1M) URM
 userdel: delete a user's login from the system. userdel(1M) SARM
 listusers: list user login information. listusers(1) URM
 logins: list user and system login information. logins(1M) SARM
 usermod: modify a user's login information on the system. usermod(1M) SARM
 login: login default file. login(4) SARM
 getlogin: get login name. getlogin(3C) PRM
 logname: get login name. logname(1) URM
 cuserid: get character login name of the user. cuserid(3S) PRM
 useradd: administer a new user login on the system. useradd(1M) SARM
 attributes. passwd: change login password and password passwd(1) URM
 rlogind: remote login server. rlogind(1M) NUAG
 rlogind: remote login server. rlogind(1M) SARM
 login: sign on. login(1) URM
 setting up an environment at login time. profile: profile(4) SARM
 ct: spawn login to a remote terminal. ct(1C) URM
 attempts. loginlog: log of failed login loginlog(4) SARM
 indicate last user or terminal logins. last: last(1) URM
 login information. logins: list user and system logins(1M) SARM
 longline: reads file, gets logname: get login name. logname(1) URM
 setjmp, longest line. readfile, readfile(1F) CHAR
 sigsetjmp, siglongjmp:/ setjmp, longjmp: non-local goto. setjmp(3C) PRM
 longjmp, _setjmp, _longjmp, setjmp(3) BSD

setjmp, longjmp, _setjmp, _longjmp, sigsetjmp, siglongjmp:/ setjmp(3) BSD
 longest line. readfile, longline: reads file, gets readfile(1F) CHAR
 /has_ic, has_il, killchar, longname, termatrrs, termname:/ curs_termatrrs(3X) CHAR
 transport endpoint. t_look: look at the current event on a t_look(3N) NI
 dictionary or lines in a sorted/ look: find words in the system look(1) BSD
 bibliographic database. lookbib: find references in a lookbib(1) BSD
 locate commands by keyword lookup. apropos: apropos(1) BSD
 lo: software loopback network interface. lo(7) NUAG
 lo: software loopback network interface. lo(7) SARM
 ticlts, ticots, ticotsord: loopback transport providers. ticlts(7) NI
 for an object library. lorder: find ordering relation lorder(1) PRM
 hdformat: low level hard disk formatter. hdformat(1M) MBRMAN
 nice: run a command at low priority. nice(1) URM
 /ripoffline, curs_set, napms: low-level curses routines. curs_kernel(3X) CHAR
 to an LP print service. lp, cancel: send/cancel requests lp(1) URM
 lp: parallel port interface. lp(7) SARM
 lpadmin: configure the LP print service. lpadmin(1M) SARM
 /lpshut, lpmove: start/stop the LP print service and move/ lpsched(1M) SARM
 about the status of the LP print service. /information lpstat(1) URM
 send/cancel requests to an LP print service. lp, cancel: lp(1) URM
 administer filters used with the LP print service. lpfilter: lpfilter(1M) SARM
 the administer forms used with the LP print service. lpforms: lpforms(1M) SARM
 enable, disable: enable/disable LP printers. enable(1) URM
 service. lpadmin: configure the LP print lpadmin(1M) SARM
 program. lpc: line printer control lpc(1M) BSD
 used with the LP print service. lpfilter: administer filters lpfilter(1M) SARM
 with the LP print service. lpforms: administer forms used lpforms(1M) SARM
 service and/ lpsched, lpshut, lpmove: start/stop the LP print lpsched(1M) SARM
 printer jobs. lpq: display the queue of lpq(1) BSD
 printer queue. lpr: send a job to the printer. lpr(1) BSD
 execution count profile data. lprm: remove jobs from the lprm(1) BSD
 start/stop the LP print service/ lprof: display line-by-line lprof(1) PRM
 LP print service and/ lpsched, lpshut, lpmove: lpsched(1M) SARM
 the status of the LP print/ lpsched(1M) SARM
 systems with the print service. lpstat: print information about lpstat(1) URM
 ripple pattern. lpssystem: register remote lpssystem(1M) SARM
 priorities. lptest: generate lineprinter lptest(1) BSD
 jrand48,/ drand48, erand48, lpusers: set printing queue lpusers(1M) SARM
 directory. ls, lc: list contents of ls(1) XNX
 ls: list contents of directory. ls(1) URM
 directory. ls: list the contents of a ls(1) BSD
 and update. lsearch, lfind: linear search lsearch(3C) PRM
 pointer. lseek: move read/write file lseek(2) PRM
 stat, lstat, fstat: get file status. stat(2) PRM
 stat, lstat, fstat: get file status. stat(2) XNX
 integers and long/ l3tol, ltol3: convert between 3-byte l3tol(3C) PRM

of the current host. mach: macro processor. m4(1) PRM
 u3b15, vax, 386, u370: get/ machid: display the processor type mach(1) BSD
 setuname: changes machid: pdp11, u3b, u3b2, u3b5, machid(1) URM
 sysi86: machine information. setuname(1M) SARM
 values: machine specific functions. sysi86(2) PRM
 machine-dependent values. values(5) PRM
 machine-independent fashion. sputl(3X) PRM
 machines. rusers(1) NI
 machines. rusers(1) NUAG
 machines. rwall(3N) NI
 machines. rwho(1) NUAG
 machines. rwho(1) URM
 machines. /return information rusers(3N) NI
 machines. ruptime: ruptime(1) NUAG
 machines. ruptime: ruptime(1) URM
 macro processor. m4(1) PRM
 macros. ms(7) BSD
 macros for formatting papers. me(7) BSD
 macros to format Reference man(7) BSD
 madd, msub, mult, mdiv, mcmp, mp(3X) BSD
 magnetic tape. tcopy(1) BSD
 magnetic tape control. mt(1) BSD
 magnetic tape streamer qt(7) SARM
 mail alias names. mailalias(1) URM
 mail aliases file. newaliases: newaliases(1M) BSD
 mail and rmail. mailcnfg: mailcnfg(4) SARM
 mail automatically. vacation(1) BSD
 mail for delivery by SMTP. smtpqer(1M) SARM
 mail from SMTP. fromsmtp(1M) SARM
 mail. mail_pipe: invoke mail_pipe(1M) SARM
 mail messages. biff(1) BSD
 mail messages. /automatically vacation(1) URM
 mail. notify: notify notify(1) URM
 mail or send mail to users. mail(1) URM
 mail over the internet. sendmail(1M) BSD
 mail queue. smtptsched: process smtptsched(1M) SARM
 mail, rmail: read mail or send mail(1) URM
 mail. /surrogate commands mailsurr(4) SARM
 mail to a remote host using smtp(1M) SARM
 mail to SMTP. tosmtp(1M) SARM
 mail to users. mail(1) URM
 Mail Transfer Protocol. /mail smtp(1M) SARM
 mailalias: translate mail alias mailalias(1) URM
 mailbox. maillock: maillock(3X) PRM
 mailcnfg: initialization mailcnfg(4) SARM
 maillock: manage lockfile for maillock(3X) PRM
 mail_pipe: invoke recipient mail_pipe(1M) SARM

Permuted Index

collected by sendmail.
routing and transport of mail.
processing message.
sequences for TTY/ sttydefs:
library. ar:
library. ar:
groups of programs. make:
intro: introduction to
mkpart: disk
/make device number from
getmajor: get major or internal
number. makedev,
number. getmajor: get
STREAMS driver. clone: open any
regenerate groups of programs.
manipulate user contexts.
device number.
from major and minor numbers..
key.
/free, realloc, calloc, malloc,
memalign, valloc.; memory/
malloc, free, realloc, calloc, malloc(3X) PRM
malloc, free, realloc, calloc, malloc(3X) PRM
malloc, free, realloc, calloc, malloc(3X) PRM
pages; find reference pages by/
Manual pages.
makedev, major, minor:
tsearch, tfind, tdelete, twalk:
hsearch, hcreate, hdestroy:
mailbox. maillock:
endpoint. t_optmgmt:
swapctl:
break up an I/O request into
passmgmt: password files
sigaction: detailed signal
mctl: memory
memcntl: memory
space from a private space
space back into a private space
initialize a private space
sigpause: simplified signal
sh: shell layer
records. fwtmp, wtmpfix:
elf_flagscn, elf_flagshdr:
modf, modfl, nextafter, scalb:
/overwrite, copywin: overlap and
mailstats: print statistics mailstats(1M) BSD
mailsur: surrogate commands for mailsurr(4) SARM
mailx: interactive message mailx(1) URM
maintain line settings and hunt sttydefs(1M) SARM
maintain portable archive or ar(1) PRM
maintain portable archive or ar(1) URM
maintain, update, and regenerate make(1) PRM
maintenance commands and/ intro(1M) SARM
maintenance utility. mkpart(1M) SARM
major and minor numbers.. makedevice(D3DK) DDRM
major device number. getmajor(D3DK) DDRM
major, minor: manage a device makedev(3C) PRM
major or internal major device getmajor(D3DK) DDRM
major/minor device pair on a clone(7) STRM
make: maintain, update, and make(1) PRM
makecontext, swapcontext: makecontext(3C) PRM
makedbm: make a YP dbm file. makedbm(1M) NI
makedev, major, minor: manage a makedev(3C) PRM
makedevice: make device number makedevice(D3DK) DDRM
makefsys: create a file system. makefsys(1M) SARM
makekey: generate encryption makekey(1) URM
mallinfo: memory allocator. malloc(3X) PRM
malloc, free, realloc, calloc, malloc(3C) PRM
malloc, free, realloc, calloc, malloc(3X) PRM
malloc, free, realloc, calloc, malloc(3X) PRM
man: display reference manual man(1) BSD
man: macros to format Reference man(7) BSD
manage a device number. makedev(3C) PRM
manage binary search trees. tsearch(3C) PRM
manage hash search tables. hsearch(3C) PRM
manage lockfile for user's maillock(3X) PRM
manage options for a transport t_optmgmt(3N) NI
manage swap space. swapctl(2) PRM
manageable units. dma_pageio: dma_pageio(D3DK) DDRM
management. passmgmt(1M) SARM
management. sigaction(2) PRM
management control. mctl(3) BSD
management control. memcntl(2) PRM
management map. /allocate rmalloc(D3DK) DDRM
management map. rmfree: free rmfree(D3DK) DDRM
management map. rminit: rminit(D3DK) DDRM
management. /sigignore, signal(2) PRM
manager. shl(1) URM
manipulate connect accounting fwtmp(1M) SARM
manipulate flags. /elf_flagphdr, elf_flag(3E) PRM
manipulate. frexp, ldexp, logb, frexp(3C) PRM
manipulate overlapped curses/ curs_overlay(3X) CHAR

/setpwent, endpwent, fgetpwent: manipulate password file entry. getpwent(3C) PRM
 /sigdelset, sigismember: manipulate sets of signals. sigsetops(3C) PRM
 /fgetspent, lckpwwdf, ulckpwwdf: manipulate shadow password file/ getspent(3C) PRM
 of an object file. mcs: manipulate the comment section mcs(1) PRM
 route: manually manipulate the routing tables. route(1M) NUAG
 route: manually manipulate the routing tables. route(1M) SARM
 makecontext, swapcontext: manipulate user contexts. makecontext(3C) PRM
 inet_ntoa: Internet address manipulation. /inet_netof, inet(3N) NI
 /for dealing with creation and manipulation of CLIENT handles. rpc_clnt_create(3N) NI
 /pair_content: curses color manipulation routines. curs_color(3X) CHAR
 /curses screen initialization and manipulation routines. curs_initscr(3X) CHAR
 /panel_hidden: panels deck manipulation routines. panel_show(3X) CHAR
 /bottom_panel: panels deck manipulation routines. panel_top(3X) CHAR
 wbgd: curses window background manipulation routines. /bkgd, curs_bkgd(3X) CHAR
 strsrpn, strtrns: string manipulations. str: strfind, str(3G) PRM
 create the cat files for the manual. catman: catman(1M) BSD
 man: macros to format Reference Manual pages. man(7) BSD
 pages by/ man: display reference manual pages; find reference man(1) BSD
 tables. route: manually manipulate the routing route(1M) NUAG
 tables. route: manually manipulate the routing route(1M) SARM
 /read and optionally write the manufacturer's defect list. mdl(1M) MBRMAN
 /return current version of a YP map at a YP server host. yppoll(1M) NI
 space. segmap: map device memory into user segmap(D2K) DDRM
 ypxfr: transfer YP map from a YP server to host. ypxfr(1M) NI
 return name of YP server or map master. ypwhich: ypwhich(1) NI
 ascii: map of ASCII character set. ascii(5) PRM
 ascii: map of ASCII character set. ascii(5) SARM
 mmap: mmap pages of memory. mmap(2) PRM
 structure. map: private memory map map(D4DK) DDRM
 from a private space management map. rmalloc: allocate space rmalloc(D3DK) DDRM
 into a private space management map. rmfree: free space back rmfree(D3DK) DDRM
 a private space management map. rminit: initialize rminit(D3DK) DDRM
 map: private memory map structure. map(D4DK) DDRM
 of one or more keys from a YP map. ypmatch: print the value ypmatch(1) NI
 propagation of a changed YP map. yppush: force yppush(1M) NI
 check virtual mapping for memory mapped device. mmap: mmap(D2K) DDRM
 addresses to RPC program number mapper. rpcbind: universal rpcbind(1M) NI
 addresses to RPC program number mapper. rpcbind: universal rpcbind(1M) NUAG
 device. mmap: check virtual mapping for memory mapped mmap(D2K) DDRM
 physmap: obtain virtual address mapping for physical addresses. physmap(D3DK) DDRM
 File Sharing user and group mapping. idload: Remote idload(1M) NUAG
 File Sharing user and group mapping. idload: Remote idload(1M) SARM
 set protection of memory mapping. mprotect: mprotect(2) PRM
 ethers: Ethernet address mapping operations. ethers(3N) NI
 ethers: Ethernet address mapping operations. ethers(3N) NUAG
 rmsetwant: set the map's wait flag for a wakeup. rmsetwant(D3DK) DDRM
 versions of a troff/ diffmk: mark differences between diffmk(1) BSD

<p><code>/set_menu_mark</code>, <code>menu_mark</code>: menus return a list of currently <code>sigsetmask</code>: set current signal mask. <code>umask</code>: set file-creation mode mask. <code>umask</code>: set and get file creation mask and suspend process until/ <code>sigsuspend</code>: install a signal change or examine signal Remote File Sharing name server Remote File Sharing name server return name of YP server or map unlock a pseudo-terminal set and get menus pattern which contain devices that <code>regex</code>: regular expression compile and regular expression compile and regular expression compile and <code>gmatch</code>: shell global pattern <code>math</code>: intro: introduction to constants. <code>eqn</code>, <code>neqn</code>, <code>checkeq</code>: typeset function. integers. <code>/menu_format</code>: set and get <code>getrlimit</code>, <code>setrlimit</code>: control message in asynchronous mode.. asynchronous MULTIBUS II/ asynchronous endpoint/ fragment of a request message. asynchronous MULTIBUS II/ MULTIBUS II messages in. transactionless message in/ message asynchronously.. message asynchronously.. request message.. message in synchronous mode.. synchronous MULTIBUS II/ synchronous endpoint/ fragment of a request message.. synchronous MULTIBUS II/ MULTIBUS II messages in. transactionless message in/ message in synchronous.</p>	<p>mark string routines. <code>menu_mark(3X)</code> CHAR marked menu items. <code>getitems</code>: <code>getitems(1F)</code> CHAR mask. <code>sigsetmask(3)</code> BSD mask. <code>umask(1)</code> URM mask. <code>umask(2)</code> PRM mask and suspend process until/ <code>sigsuspend(2)</code> PRM mask. <code>sigprocmask</code>: <code>sigprocmask(2)</code> PRM master file. <code>rfmaster</code>: <code>rfmaster(4)</code> NUAG master file. <code>rfmaster</code>: <code>rfmaster(4)</code> SARM master. <code>ypwhich</code>: <code>ypwhich(1)</code> NI master/slave pair. <code>unlockpt</code>: <code>unlockpt(3C)</code> STRM match buffer. <code>/menu_pattern</code>: <code>menu_pattern(3X)</code> CHAR match criteria. <code>/device groups</code> <code>getdgrp(1M)</code> SARM match patterns against a string. <code>regex(1F)</code> CHAR match routines. <code>/step, advance</code>: <code>regex(5)</code> PRM match routines. <code>/step, advance</code>: <code>regex(5)</code> SARM match routines. <code>/step, advance</code>: <code>regex(3G)</code> PRM matching. <code>gmatch(3G)</code> PRM math functions and constants. <code>math(5)</code> PRM math libraries. <code>intro(3M)</code> PRM math: math functions and <code>math(5)</code> PRM mathematics. <code>eqn(1)</code> BSD <code>matherr</code>: error-handling <code>matherr(3M)</code> PRM <code>max</code>: return the larger of two <code>max(D3DK)</code> DDRM maximum numbers of rows and/ <code>menu_format(3X)</code> CHAR maximum system resource/ <code>getrlimit(2)</code> PRM <code>mb2a_brdcst</code>: Sends a broadcast <code>mb2a_brdcst(3tai)</code> TAI <code>mb2a_closeport</code>: Closes an <code>mb2a_closeport(3tai)</code> TAI <code>mb2a_getinfo</code>: Gets specific <code>mb2a_getinfo(3tai)</code> TAI <code>mb2a_getreqfrag</code>: Receives a data <code>mb2a_getreqfrag(3tai)</code> TAI <code>mb2a_openport</code>: Opens an <code>mb2a_openport(3tai)</code> TAI <code>mb2a_receive</code>: Receives incoming <code>mb2a_receive(3tai)</code> TAI <code>mb2a_send</code>: Sends a <code>mb2a_send(3tai)</code> TAI <code>mb2a_sendcancel</code>: Sends a cancel <code>mb2a_sendcancel(3tai)</code> TAI <code>mb2a_sendreply</code>: Sends a reply <code>mb2a_sendreply(3tai)</code> TAI <code>mb2a_sendsrvp</code>: Sends a rsvp <code>mb2a_sendsrvp(3tai)</code> TAI <code>mb2_gethostid</code>: gets host ID. <code>mb2_gethostid(3tai)</code> TAI <code>mb2s_brdcst</code>: Sends a broadcast <code>mb2s_brdcst(3tai)</code> TAI <code>mb2s_closeport</code>: Closes a <code>mb2s_closeport(3tai)</code> TAI <code>mb2s_getinfo</code>: Gets specific <code>mb2s_getinfo(3tai)</code> TAI <code>mb2s_getreqfrag</code>: Receives a data <code>mb2s_getreqfrag(3tai)</code> TAI <code>mb2s_openport</code>: Opens a <code>mb2s_openport(3tai)</code> TAI <code>mb2s_receive</code>: Receives incoming <code>mb2s_receive(3tai)</code> TAI <code>mb2s_send</code>: Sends a <code>mb2s_send(3tai)</code> TAI <code>mb2s_sendcancel</code>: Sends a cancel <code>mb2s_sendcancel(3tai)</code> TAI</p>
--	--

message in synchronous mode.. message and receives a/
 multibyte character handling. character/ mbchar: mbtowc,
 string functions. mbstring:
 multibyte string functions.
 character handling. mbchar:
 mp: madd, msub, mult, mdiv,
 section of an object file.

mpow,/ mp: madd, msub, mult,
 the manufacturer's defect/
 papers.
 backup operations to service
 disk: random access bulk storage

malloc, free, realloc, calloc,
 offsetof: offset of structure
 elf_next: sequential archive
 elf_rand: random archive
 elf_getarhdr: retrieve archive
 listdgrp: lists
 groups: print group
 groups: display a user's group

memmove, memset memory/ memory:
 memset: memory/ memory: memccpy,
 memory/ memory: memccpy, memchr,
 control.

memory: memccpy, memchr, memcmp,
 memccpy, memchr, memcmp, memcpy,
 copylist: copy a file into

lock: lock a process in primary
 mem, kmem: core
 mmap: map pages of
 munmap: unmap pages of
 rmwant: wait for free
 alloca:
 calloc, memalign, valloc,;
 calloc, mallot, mallinfo:
 shmctl: shared
 spawn new process in a virtual
 bytes. bzero: clear
 queue, semaphore set, or shared
 segmap: map device
 allocate space from kernel free
 free previously allocated kernel

mb2s_sendreply: Sends a reply mb2s_sendreply(3tai) TAI
 mb2s_sendsrvp: Sends a request mb2s_sendsrvp(3tai) TAI
 mbchar: mbtowc, mblen, wctomb: mbchar(3C) PRM
 mblen, wctomb: multibyte mbchar(3C) PRM
 mbstowcs, wcstombs: multibyte mbstring(3C) PRM
 mbstring: mbstowcs, wcstombs: mbstring(3C) PRM
 mbtowc, mblen, wctomb: multibyte mbchar(3C) PRM
 mcmp, min, mout, pow, gcd, rpow,/ mp(3X) BSD
 mcs: manipulate the comment mcs(1) PRM
 mctl: memory management control. mctl(3) BSD
 mdevice: file format. mdevice(4) SARM
 mdiv, mcmp, min, mout, pow, gcd, mp(3X) BSD
 mdl: read and optionally write mdl(1M) MBRMAN
 me: macros for formatting me(7) BSD
 media insertion prompts. /with bkoper(1M) SARM
 medium. disk(7) SARM
 mem, kmem: core memory. mem(7) SARM
 memalign, valloc,; memory/ malloc(3C) PRM
 member. offsetof(3C) PRM
 member access. elf_next(3E) PRM
 member access. elf_rand(3E) PRM
 member header. elf_getarhdr(3E) PRM
 members of a device group. listdgrp(1M) SARM
 membership of user. groups(1) URM
 memberships. groups(1) BSD
 memccpy, memchr, memcmp, memcpy, memory(3C) PRM
 memchr, memcmp, memcpy, memmove, memory(3C) PRM
 memcmp, memcpy, memmove, memset: memory(3C) PRM
 memcntl: memory management memcntl(2) PRM
 memcpy, memmove, memset: memory/ memory(3C) PRM
 memmove, memset: memory/ memory: memory(3C) PRM
 memory. copylist(3G) PRM
 memory. lock(2) NXN
 memory. mem(7) SARM
 memory. mmap(2) PRM
 memory. munmap(2) PRM
 memory. rmwant(D3DK) DDRM
 memory allocator. alloca(3) BSD
 memory allocator. /realloc, malloc(3C) PRM
 memory allocator. /realloc, malloc(3X) PRM
 memory control operations. shmctl(2) PRM
 memory efficient way. vfork: vfork(2) PRM
 memory for a given number of bzero(D3DK) DDRM
 memory ID. /remove a message ipcrm(1) URM
 memory into user space. segmap(D2K) DDRM
 memory. kmem_alloc: kmem_alloc(D3DK) DDRM
 memory. kmem_free: kmem_free(D3DK) DDRM

Permuted Index

and clear space from kernel free
 mctl: memory management control. mctl(3) BSD
 memcntl: memory management control. memcntl(2) PRM
 map: private memory map structure. map(D4DK) DDRM
 mmap: check virtual mapping for memory mapped device. mmap(D2K) DDRM
 mprotect: set protection of memory mapping. mprotect(2) PRM
 memcpy, memmove, memset: memory/ memory: memccpy, memchr, memcmp, memory(3C) PRM
 lock (or unlock) pages in memory. mlock, munlock: mlock(3C) PRM
 shmop: shmat, shmdt: shared memory operations. shmop(2) PRM
 memcmp, memcpy, memmove, memset: /memchr, memory(3C) PRM
 or data. plock: lock into memory or unlock process, text, plock(2) PRM
 mincore: determine residency of memory pages. mincore(2) PRM
 display the size of a page of memory. pagesize: pagesize(1) BSD
 puts a buffer back into the free memory pool. mps_free_msgbuf: mps_free_msgbuf(D3D) DDRM
 memory segment identifier. memory segment identifier. shmget(2) PRM
 memory. useracc: verify useracc(D3DK) DDRM
 memory with physical storage. msync(3C) PRM
 memset: memory operations. memory(3C) PRM
 menu item. ckitem: build ckitem(1) SARM
 menu item. ckitem: build ckitem(1) SS
 menu items. getitems: return getitems(1F) CHAR
 menu or task removal tool. delsysadm(1M) SARM
 menu or task removal tool. delsysadm(1M) SS
 menu; prompt for and return a ckitem(1) SARM
 menu; prompt for and return a ckitem(1) SS
 a list of currently marked menu attributes: set_menu_fore, menu_attributes(3X) CHAR
 delsysadm: sysadm interface menu_back, set_menu_grey, menu_attributes(3X) CHAR
 delsysadm: sysadm interface menu_cursor: pos_menu_cursor: menu_cursor(3X) CHAR
 menu item. ckitem: build a menu_driver: command processor menu_driver(3X) CHAR
 menu item. ckitem: build a menu_fore, set_menu_back, menu_attributes(3X) CHAR
 menu_fore, set_menu_grey, /menu_fore, set_menu_back, correctly position a menus/ menu_format: set and get/ menu_format(3X) CHAR
 for the menus subsystem. menu_format: set and get maximum/ .. menu_format(3X) CHAR
 menu_attributes: set_menu_fore, menu_grey, set_menu_pad, / menu_attributes(3X) CHAR
 menu_format: set and get/ menu_hook: set_item_init, menu_hook(3X) CHAR
 menu_back, set_menu_grey, item_init, set_item_term, / menu_init, set_menu_term, / menu_hook(3X) CHAR
 /menu_term, set_menu_init, set_current item, / menu_item_current: menu_item_current(3X) CHAR
 set_current item, / menu_item_name: item_name, menu_item_name(3X) CHAR
 item_description: get menus/ menu_item_new: new_item, menu_item_new(3X) CHAR
 free_item: create and destroy/ menu_item_opts: set_item_opts, menu_item_opts(3X) CHAR
 item_opts_on, item_opts_off, menu_items: set_menu_items, menu_items(3X) CHAR
 menu_items, item_count: connect/ menu_items, item_count: connect/ menu_items(3X) CHAR
 menu_items: set_menu_items, menu_item_userptr: menu_item_userptr(3X) CHAR
 set_item_userptr, / menu_item_value: menu_item_value(3X) CHAR
 set_item_value, item_value: / menu_item_visible: menu_item_visible(3X) CHAR
 item_visible: tell if menus/ menu_mark: set_menu_mark, menu_mark(3X) CHAR
 menu_mark: menus mark string/ menu_mark: set_menu_mark, menu_mark(3X) CHAR
 menu_mark: menus mark string/

create and destroy menus.
 menu_opts_on, menu_opts_off,/
 /menu_opts_on, menu_opts_off,
 /set_menu_opts, menu_opts_on,
 menu_opts: set_menu_opts,
 /menu_grey, set_menu_pad,
 menu_pattern: set and get menus/
 menu_pattern: set_menu_pattern,
 unpost_menu: write or erase/
 package.
 correctly position a
 /set_menu_pad, menu_pad: control
 /unpost_menu: write or erase
 /item_visible: tell if
 /item_name, item_description: get
 /item_value: set and get
 and disconnect items to and from
 associate application data with
 free_item: create and destroy
 item_index: set and get current
 /set_menu_mark, menu_mark:
 free_menu: create and destroy
 associate application data with
 /menu_opts_off, menu_opts:
 menus: character based
 /menu_pattern: set and get
 for automatic invocation by
 numbers of rows and columns in
 command processor for the
 /menu_sub, scale_menu:
 window/ /menu_win, set_menu_sub,
 /menu_init, set_menu_term,
 menu_userptr: associate/
 menu_userptr: set_menu_userptr,
 menu_win, set_menu_sub,/
 menu_win: set_menu win,
 sort: sort and/or
 files. acctmerg:
 files or subsequent/ paste:
 adjmsg: trim bytes from a
 catgets: read a program
 copymsg: copy a
 dupmsg: duplicate a
 t_error: produce error
 menu_new: new_menu, free_menu: menu_new(3X) CHAR
 menu_opts: set_menu_opts, menu_opts(3X) CHAR
 menu_opts: menu option/ menu_opts(3X) CHAR
 menu_opts_off, menu_opts: menus/ menu_opts(3X) CHAR
 menu_opts_on, menu_opts_off,/ menu_opts(3X) CHAR
 menu_pad: control menu display/
 menu_attributes(3X) CHAR
 menu_pattern: set_menu_pattern, menu_pattern(3X) CHAR
 menu_pattern: set and get menus/ menu_pattern(3X) CHAR
 menu_post: post_menu, menu_post(3X) CHAR
 menus: character based menus menus(3X) CHAR
 menus cursor. /pos_menu_cursor: menu_cursor(3X) CHAR
 menus display attributes. menu_attributes(3X) CHAR
 menus from associated/ menu_post(3X) CHAR
 menus item is visible. menu_item_visible(3X) CHAR
 menu item name and description.
 menu_item_name(3X) CHAR
 menu item values. menu_item_value(3X) CHAR
 menus. /item_count: connect menu_items(3X) CHAR
 menus items. /item_userptr: menu_item_userptr(3X) CHAR
 menus items. /new_item, menu_item_new(3X) CHAR
 menus items. /top_row, menu_item_current(3X) CHAR
 menu mark string routines. menu_mark(3X) CHAR
 menus. menu_new: new_menu, menu_new(3X) CHAR
 menus. /menu_userptr: menu_userptr(3X) CHAR
 menus option routines. menu_opts(3X) CHAR
 menu package. menus(3X) CHAR
 menu pattern match buffer. menu_pattern(3X) CHAR
 menus. /routines menu_hook(3X) CHAR
 menus. /set and get maximum menu_format(3X) CHAR
 menu subsystem. menu_driver: menu_driver(3X) CHAR
 menu window and subwindow/ menu_win(3X) CHAR
 menu_sub, scale_menu: menus menu_win(3X) CHAR
 menu_term: assign/ menu_hook(3X) CHAR
 menu_userptr: set_menu_userptr, menu_userptr(3X) CHAR
 menu_userptr: associate/ menu_userptr(3X) CHAR
 menu_win: set_menu_win, menu_win(3X) CHAR
 menu_win, set_menu_sub,/ menu_win(3X) CHAR
 merge files. sort(1) URM
 merge or add total accounting acctmerg(1M) SARM
 merge same lines of several paste(1) URM
 msg: permit or deny messages. msg(1) URM
 message. adjmsg(D3DK) DDRM
 message. catgets(3C) PRM
 message. copymsg(D3DK) DDRM
 message. dupmsg(D3DK) DDRM
 message. t_error(3N) NI

mb2s_sendrsvp: Sends a request message and receives a/ mb2s_sendrsvp(3tai) TAI
 mb2a_sendcancel: Sends a cancel message asynchronously.. mb2a_sendcancel(3tai) TAI
 mb2a_sendreply: Sends a reply message asynchronously.. mb2a_sendreply(3tai) TAI
 putbq: place a message at the head of a queue. putbq(D3DK) DDRM
 allocb: allocate a message block. allocb(D3DK) DDRM
 copyb: copy a message block. copyb(D3DK) DDRM
 freeb: free a message block. freeb(D3DK) DDRM
 dupb: duplicate a message block descriptor. dupb(D3DK) DDRM
 rmvb: remove a message block from a message. rmvb(D3DK) DDRM
 message. unlnkb: remove a message block from the head of a unlnkb(D3DK) DDRM
 msgb: STREAMS message block structure. msgb(D4DK) DDRM
 buffer. esballo: allocate a message block using a shared esballo(D3DK) DDRM
 linkb: concatenate two message blocks. linkb(D3DK) DDRM
 freemsg: free all message blocks in a message. freemsg(D3DK) DDRM
 mps_get_msgbuf: allocates a message buffer.. mps_get_msgbuf(D3D) DDRM
 /copies user data from the message buffer.. mps_get_soldata(D3D) DDRM
 /copies user data from the message buffer.. mps_get_unsoldata(D3D) DDRM
 catopen, catclose: open/close a message catalogue. catopen(3C) PRM
 gencat: generate a formatted message catalogue. gencat(1) URM
 msgctl: message control operations. msgctl(2) PRM
 retrieve a text string from a message data base. gettxt: gettxt(1) URM
 or search for a text string in message data bases. /of, srchtxt(1) URM
 datab: STREAMS message data structure. datab(D4DK) DDRM
 test whether a message is a data message. datamsg: datamsg(D3DK) DDRM
 sets target for front panel message delivery. dbon: dbon(1M) MBRMAN
 mkmsgs: create message files for use by gettxt. mkmsgs(1) URM
 /that specifies a driver's message freeing routine. free_rtn(D4DK) DDRM
 free all message blocks in a message. freemsg: freemsg(D3DK) DDRM
 getq: get the next message from a queue. getq(D3DK) DDRM
 rmvq: remove a message from a queue. rmvq(D3DK) DDRM
 send, sendto, sendmsg: send a message from a socket. send(3N) NI
 recvfrom, recvmsg: receive a message from a socket. recv, recv(3N) NI
 mb2a_brdcst: Sends a broadcast message in asynchronous mode.. mb2a_brdcst(3tai) TAI
 /Sends a transactionless message in asynchronous mode.. mb2a_send(3tai) TAI
 mb2s_sendcancel: Sends a cancel message in synchronous. mb2s_sendcancel(3tai) TAI
 mb2s_brdcst: Sends a broadcast message in synchronous mode.. mb2s_brdcst(3tai) TAI
 /Sends a transactionless message in synchronous mode.. mb2s_send(3tai) TAI
 mb2s_sendreply: Sends a reply message in synchronous mode.. mb2s_sendreply(3tai) TAI
 insq: insert a message into a queue. insq(D3DK) DDRM
 datamsg: test whether a message is a data message. datamsg(D3DK) DDRM
 puts its arguments on FMLI message line. message: message(1F) CHAR
 Sends a rsvp request message.. mb2a_sendrsvp: mb2a_sendrsvp(3tai) TAI
 return the number of bytes in a message. msgdsize: msgdsize(D3DK) DDRM
 send listener service request message. nlsrequest: format and nlsrequest(3N) NI
 help: ask for help with message numbers or SCCS/ help(1) PRM
 getmsg: get next message off a stream. getmsg(2) PRM
 getmsg: get next message off a stream. getmsg(2) STRM

putq: put a message on a queue. putq(D3DK) DDRM
 putmsg: send a message on a stream. putmsg(2) PRM
 putmsg: send a message on a stream. putmsg(2) STRM
 reverse/ qreply: send a message on a stream in the qreply(D3DK) DDRM
 console. fmtmsg: display a message on stderr or system fmtmsg(1) URM
 console. fmtmsg: display a message on stderr or system fmtmsg(3C) PRM
 print: display a driver message on system console. print(D2DK) DDRM
 msgop: msgsnd, msgrcv: message operations. msgop(2) PRM
 cmn_err: display an error message or panic the system. cmn_err(D3DK) DDRM
 driver. mps: Message Passing Space device mps(1M) MBRMAN
 mailx: interactive message processing system. mailx(1) URM
 ICMP: Internet Control Message Protocol. ICMP(7) NI
 ICMP: Internet Control Message Protocol. ICMP(7) NUAG
 ICMP: Internet Control Message Protocol. ICMP(7) SARM
 concatenate bytes in a message. pullupmsg: pullupmsg(D3DK) DDRM
 FMLI message line. message: puts its arguments on message(1F) CHAR
 canput: test for room in a message queue. canput(D3DK) DDRM
 msgget: get message queue. msgget(2) PRM
 shared memory/ ipcrm: remove a message queue, semaphore set, or ipcrm(1) URM
 a data fragment of a request message. /Receives mb2a_getreqfrag(3tai) TAI
 a data fragment of a request message.. /Receives mb2s_getreqfrag(3tai) TAI
 remove a message block from a message. rmvb: rmvb(D3DK) DDRM
 strerror: get error message string. strerror(3C) PRM
 putctl: send a control message to a queue. putctl(D3DK) DDRM
 /constructs a broadcast message to be sent.. mps_mk_brdcst(D3D) DDRM
 /constructs an unsolicited message to be sent.. mps_mk_unsol(D3D) DDRM
 /constructs an unsolicited reply message to be sent.. mps_mk_unsolrply(D3D) DDRM
 a. mps_mk_solrply: constructs a message to be sent to initiate mps_mk_solrply(D3D) DDRM
 mps_mk_sol: constructs a message to be sent to initiate a/ mps_mk_sol(D3D) DDRM
 putnext: send a message to the next queue. putnext(D3DK) DDRM
 message block from the head of a message. unlinkb: remove a unlinkb(D3DK) DDRM
 putctl1: send a control message with a one-byte/ putctl1(D3DK) DDRM
 msg: permit or deny messages. msg(1) URM
 perror: print system error messages. perror(3C) PRM
 psignal, psiginfo: system signal messages. psignal(3C) PRM
 smtpd: receive incoming SMTP messages. smtpd(1M) SARM
 srv: service queued messages. srv(D2DK) DDRM
 strace: print STREAMS trace messages. strace(1M) SARM
 strace: print STREAMS trace messages. strace(1M) STRM
 syslogd: log system messages. syslogd(1M) BSD
 respond to incoming mail messages. /automatically vacation(1) URM
 give notice of incoming mail messages. biff: biff(1) BSD
 remote system can accept binary messages. /determine whether ckbinarsys(1M) SARM
 priority band. flushband: flush messages for a specified flushband(D3DK) DDRM
 mps_AMPsend_rsvp: queues request messages for transmission. mps_AMPsend_rsvp(D3D) DDRM
 flushq: remove messages from a queue. flushq(D3DK) DDRM
 queue. put: receive messages from the preceding put(D2DK) DDRM

Receives incoming MULTIBUS II messages in. `mb2a_receive`: `mb2a_receive(3tai)` TAI
 Receives incoming MULTIBUS II messages in. `mb2s_receive`: `mb2s_receive(3tai)` TAI
 `qsize`: find the number of messages on a queue. `qsize(D3DK)` DDRM
 `sys_siglist`: system signal messages. `psignal`, `psignal(3)` BSD
 queue. `smtpsched`: process messages queued in the SMTP mail `smtpsched(1M)` SARM
`mps_AMPsend`: sends unsolicited messages that are not part of. `mps_AMPsend(D3D)` DDRM
 `strlog`: submit messages to the log driver. `strlog(D3DK)` DDRM
 /`halfdelay`, `inrflush`, `keypad`, meta, `nodelay`, `notimeout`, `raw`,/ `curlnopts(3X)` CHAR
 `msqrt`, `sdiv`, `itom`, `xtom`, `mtox`, `mfree`: multiple precision/ `rpow`, `mp(3X)` BSD
 `mfsys`: file format.. `mfsys(4)` SARM
 microseconds. `drv_hztousec`: `drv_hztousec(D3DK)` DDRM
 microseconds to clock ticks. `drv_usectohz(D3DK)` DDRM
 microseconds. `ualarm`: schedule `ualarm(3)` BSD
 microseconds. `usleep`: suspend `usleep(3)` BSD
`madd`, `msub`, `mult`, `mdiv`, `mcmp`, migration: move an archive from `migration(1M)` SARM
 integers. `min`, `mout`, `pow`, `gcd`, `rpow`,/ `mp`: `mp(3X)` BSD
 memory pages. `min`: return the lesser of two `min(D3DK)` DDRM
`getminor`: get minor or internal `mincore`: determine residency of `mincore(2)` PRM
 `makedev`, `major`, minor device number. `getminor(D3DK)` DDRM
 number. `getminor`: get minor: manage a device number. `makedev(3C)` PRM
 /`getwin`, `delay_output`, `flushinp`: minor numbers.. /`make` `makedevice(D3DK)` DDRM
 /`isnan`, `copysign`, `scalbn`: minor or internal minor device `getminor(D3DK)` DDRM
 `intro`: introduction to miscellaneous curses utility/ `curlnopts(3X)` CHAR
 `intro`: introduction to miscellaneous functions for IEEE/ `ieee_functions(3M)` BSD
 directories in a path. miscellany. `intro(5)` PRM
 file system. miscellany. `intro(5)` SARM
 system. `mkdir`: make a directory. `mkdir(2)` PRM
 system. `mkdir`: make directories. `mkdir(1)` URM
 system. `mkdirp`, `rmdirp`: create, remove `mkdirp(3G)` PRM
 use by `gettext`. `mkfifo`: create a new FIFO. `mkfifo(3C)` PRM
 special or ordinary file. `mkfifo`: make FIFO special file. `mkfifo(1M)` SARM
 special or ordinary file. `mkfs` (bfs): construct a boot `mkfs(1M)` SARM
 utility. `mkfs` (generic): construct a file `mkfs(1M)` SARM
 name. `mkfs` (s5): construct an s5 file `mkfs(1M)` SARM
 to a calendar time. `mkfs` (ufs): construct a ufs file `mkfs(1M)` SARM
 pages in memory. `mkmsgs`: create message files for `mkmsgs(1)` URM
 unlock address space. `mknod`: make a directory, or a `mknod(2)` PRM
 memory mapped device. `mknod`: make a directory, or a `mknod(2)` XNX
 `mkpart`: disk maintenance `mkpart(1M)` SARM
 `mkstemp`: make a unique file `mkstemp(3)` BSD
 `mktemp`: make a unique file name. `mktemp(3C)` PRM
 `mktime`: converts a `tm` structure `mktime(3C)` PRM
 `mlock`, `munlock`: lock (or unlock) `mlock(3C)` PRM
 `mlockall`, `munlockall`: lock or `mlockall(3C)` PRM
 `mmap`: check virtual mapping for `mmap(D2K)` DDRM
 `mmap`: map pages of memory. `mmap(2)` PRM

getmntent, getmntany: get table.
 chmod: change file
 sulogin: access single-user is aligned for efficient flyby
 umask: set file-creation broadcast message in synchronous
 a reply message in synchronous
 pkt: STREAMS Packet
 chmod, fchmod: change message in asynchronous
 message in asynchronous
 message in synchronous
 getty: set terminal type, information for setting terminal
 manipulate. frexp, ldexp, logb, frexp, ldexp, logb, modf, settime: change the access and
 utime: set file access and touch: update access and
 system. groupmod: information on the/ usermod:
 locale. setlocale:
 pkt: STREAMS Packet Mode
 8086 absolute library and object
 STREAMS terminal line discipline
 to the write queue for this
 Pseudo Terminal Emulation
 Interface cooperating STREAMS
 read/write interface STREAMS
 and XENIX STREAMS compatibility
 identification and limit value/
 of automatically pushed STREAMS
 of automatically pushed STREAMS
 /ckpacct, dodisk, lastlogin,
 montbl: create
 statd: network status
 pmadm: port
 pmadm: port
 ttymon: port
 profile.
 ttyadm: format and output port
 database.
 through a text file.
 mount:
 mount (s5):
 mount (bfs):
 mnttab file entry. getmntent(3C) PRM
 mnttab: mounted file system mnttab(4) SARM
 mode. chmod(1) URM
 mode. sulogin(1M) SARM
 mode. /Confirm data dma_get_best_mode(D3D) DDRM
 mode mask. umask(1) URM
 mode.. mb2s_brdcst: Sends a mb2s_brdcst(3tai) TAI
 mode.. mb2s_sendreply: Sends mb2s_sendreply(3tai) TAI
 Mode module. pkt(7) STRM
 mode of file. chmod(2) PRM
 mode.. /Sends a broadcast mb2a_brdcst(3tai) TAI
 mode.. /Sends a transactionless mb2a_send(3tai) TAI
 mode.. /Sends a transactionless mb2s_send(3tai) TAI
 modes, speed, and line/ getty(1M) SARM
 modes. tset: provide tset(1) XNX
 modf, modff, nextafter, scalb: frexp(3C) PRM
 modff, nextafter, scalb:/ frexp(3C) PRM
 modification dates of files. settime(1) XNX
 modification times. utime(2) PRM
 modification times of a file. touch(1) URM
 modify a group definition on the groupmod(1M) SARM
 modify a user's login usermod(1M) SARM
 modify and query a program's setlocale(3C) PRM
 module. pkt(7) STRM
 module formats. download: loads download(1M) MBRMAN
 module. ldterm: standard ldterm(7) STRM
 module or driver. /get pointer WR(D3DK) DDRM
 module. ptem: STREAMS ptem(7) STRM
 module. timod: Transport timod(7) STRM
 module. /Transport Interface tirdwr(7) STRM
 module. tcompat: V7, 4BSD tcompat(7) SARM
 module_info: STREAMS driver module_info(D4DK) DDRM
 modules. /configure lists autopush(1M) SARM
 modules. /configure lists autopush(1M) STRM
 monacct, nulladm, prctmp,/ acctsh(1M) SARM
 monetary database. montbl(1M) SARM
 monitor. statd(1M) NUAG
 monitor administration. pmadm(1M) NI
 monitor administration. pmadm(1M) SARM
 monitor for terminal ports. ttymon(1M) SARM
 monitor: prepare execution monitor(3C) PRM
 monitor-specific information. ttyadm(1M) SARM
 montbl: create monetary montbl(1M) SARM
 more, page: browse or page more(1) URM
 mount a file system. mount(2) PRM
 mount an s5 file system. mount(1M) SARM
 mount bfs file systems. mount(1M) SARM

Permuted Index

systems. mount (bfs): mount bfs file mount(1M) SARM
 mount: mount a file system. mount(2) PRM
 resources. mount: mount remote NFS mount(1M) NUAG
 resources. mount: mount remote NFS mount(1M) SARM
 mount: mount remote resources. mount(1M) NUAG
 mount: mount remote resources. mount(1M) SARM
 mount NFS file systems. automount(1M) NUAG
 mount NFS file systems. automount(1M) SARM
 automount: automatically mount or unmount file systems mount(1M) SARM
 automount: automatically mount queued remote resources. rmnttry(1M) NUAG
 and remote/ mount, umount: mount queued remote resources. rmnttry(1M) SARM
 rmnttry: attempt to mount: mount remote NFS resources. mount(1M) NUAG
 rmnttry: attempt to mount: mount remote NFS resources. mount(1M) SARM
 mount: mount remote resources. mount(1M) NUAG
 mount: mount remote resources. mount(1M) SARM
 mountd: NFS mount request server. mountd(1M) NUAG
 mountd: NFS mount request server. mountd(1M) SARM
 system. mount (s5): mount an s5 file mount(1M) SARM
 setmnt: establish mount table. setmnt(1M) SARM
 mount (ufs): mount ufs file systems. mount(1M) SARM
 systems. mount (ufs): mount ufs file mount(1M) SARM
 file systems and remote/ mount, umount: mount or unmount mount(1M) SARM
 mountfsys, umountfsys: mount, unmount a file system. mountfsys(1M) SARM
 systems. mountall, umountall: mount, unmount multiple file mountall(1M) SARM
 Sharing/ rmountall, rumountall: mount, unmount Remote File rmountall(1M) NUAG
 Sharing/ rmountall, rumountall: mount, unmount Remote File rmountall(1M) SARM
 unmount multiple file systems. mountall, umountall: mount, mountall(1M) SARM
 server. mountd: NFS mount request mountd(1M) NUAG
 server. mountd: NFS mount request mountd(1M) SARM
 mnttab: mounted file system table. mnttab(4) SARM
 information. dfmounts: display mounted NFS resource dfmounts(1M) NUAG
 information. dfmounts: display mounted NFS resource dfmounts(1M) SARM
 dfmounts: display mounted resource information. dfmounts(1M) NUAG
 dfmounts: display mounted resource information. dfmounts(1M) SARM
 rmntstat: display mounted resource information. rmntstat(1M) NUAG
 rmntstat: display mounted resource information. rmntstat(1M) SARM
 information. dfmounts: display mounted RFS resource dfmounts(1M) NUAG
 information. dfmounts: display mounted RFS resource dfmounts(1M) SARM
 unmount a file system. mountfsys, umountfsys: mount, mountfsys(1M) SARM
 /local NFS resource available for mounting by remote systems. share(1M) NUAG
 /local NFS resource available for mounting by remote systems. share(1M) SARM
 /local resource unavailable for mounting by remote systems. unshare(1M) NUAG
 /NFS resource unavailable for mounting by remote systems. unshare(1M) SARM
 rmount: queue remote resource mounts. rmount(1M) NUAG
 rmount: queue remote resource mounts. rmount(1M) SARM
 mouseadmin: mouse administration. mouseadmin(1) MOUSE
 Bus, serial, and AT&T. mouse: mouse device driver supporting mouse(7) MOUSE

supporting Bus, serial, and/ administration.
 /msub, mult, mdiv, mcmp, min, mvdire: move a directory. mvdire(1M) SARM
 panel_move: move panel: move a panels window on the/ panel_move(3X) CHAR
 volumes to another. migration: move an archive from one set of migration(1M) SARM
 curs_move: move, wmove: move curses window cursor. curs_move(3X) CHAR
 mv: move files. mv(1) URM
 lseek: move read/write file pointer. lseek(2) PRM
 the LP print service and move requests. /start/stop lpsched(1M) SARM
 cursor. curs_move: move, wmove: move curses window curs_move(3X) CHAR
 /form_fields, field_count, move_field: connect fields to/ form_field(3X) CHAR
 on the virtual/ panel_move: move panel: move a panels window panel_move(3X) CHAR
 mcmp, min, mout, pow, gcd, / mp: madd, msub, mult, mdiv, mp(3X) BSD
 mpc: MPC device driver. mpc(1M) MBRMAN
 mpc: MPC device driver. mpc(1M) MBRMAN
 mprotect: set protection of mprotect(2) PRM
 mps: Message Passing Space mps(1M) MBRMAN
 mps_AMPcancel: cancels an mps_AMPcancel(D3D) DDRM
 mps_AMPreceive: receives mps_AMPreceive(D3D) DDRM
 mps_AMPreceive_frag: receives mps_AMPreceive_frag(D3D) DDRM
 mps_AMPsend: sends unsolicited ... mps_AMPsend(D3D) DDRM
 mps_AMPsend_data: sends mps_AMPsend_data(D3D) DDRM
 mps_AMPsend_reply: replies to a mps_AMPsend_reply(D3D) DDRM
 mps_AMPsend_rsvp: queues request mps_AMPsend_rsvp(D3D) DDRM
 mps_close_chan: closes a mps_close_chan(D3D) DDRM
 mpencv: Convert transport code. mpencv(1M) MBRMAN
 mps_free_dmabuf: frees a list of mps_free_dmabuf(D3D) DDRM
 mps_free_msgbuf: puts a buffer mps_free_msgbuf(D3D) DDRM
 mps_free_tid: frees a previously mps_free_tid(D3D) DDRM
 mps_get_dmabuf: returns a mps_get_dmabuf(D3D) DDRM
 mps_get_msgbuf: allocates a mps_get_msgbuf(D3D) DDRM
 mps_get_reply_len: get data mps_get_reply_len(D3D) DDRM
 mps_get_soldata: copies user mps_get_soldata(D3D) DDRM
 mps_get_tid: allocates mps_get_tid(D3D) DDRM
 mps_get_unsoldata: copies user mps_get_unsoldata(D3D) DDRM
 mps_mk_bgrant: construct a mps_mk_bgrant(D3D) DDRM
 mps_mk_brdcst: constructs a mps_mk_brdcst(D3D) DDRM
 mps_mk_breject: construct a mps_mk_breject(D3D) DDRM
 mps_mk_sol: constructs a message mps_mk_sol(D3D) DDRM
 mps_mk_solrply: constructs a mps_mk_solrply(D3D) DDRM

Permuted Index

unsolicited message to be/ mps_mk_unsol: constructs an mps_mk_unsol(D3D) DDRM
 unsolicited reply message to be/ mps_mk_unsolrply: constructs a
 mps_mk_unsolrply(D3D) DDRM
 mps_msg_getmsgtyp,/ mps_msg: mps_msg_getsrcmid, mps_msg(D3D) DDRM
 /mps_msg_getmsgtyp, mps_msg_getbrlen,/ mps_msg(D3D) DDRM
 /mps_msg_getreqid, mps_msg_getlsnid,/ mps_msg(D3D) DDRM
 mps_msg: mps_msg_getsrcmid, mps_msg_getmsgtyp,/ mps_msg(D3D) DDRM
 /mps_msg_getbrlen, mps_msg_getreqid,/ mps_msg(D3D) DDRM
 mps_msg_getmsgtyp,/ mps_msg: mps_msg_getsrcmid, mps_msg(D3D) DDRM
 /mps_msg_getlsnid, mps_msg_getsrcpid,/ mps_msg(D3D) DDRM
 /mps_msg_getsrcpid, mps_msg_gettrnsid,/ mps_msg(D3D) DDRM
 /mps_msg_gettrnsid, mps_msg_getudp,/ mps_msg(D3D) DDRM
 /mps_msg_getudp, mps_msg_iscancel, mps_msg_iseot,/ mps_msg(D3D) DDRM
 /mps_msg_iseot, mps_msg_iserror, mps_msg_iscompletion,/ mps_msg(D3D) DDRM
 /mps_msg_iscancel, mps_msg_iseot, mps_msg_iserror,/ mps_msg(D3D) DDRM
 /mps_msg_iscancel, mps_msg_iseot, mps_msg_isreq, /mps_msg_iserror, mps_msg(D3D) DDRM
 mps_msg_iscompletion, mps_open_chan: opens a channel.
 mps_open_chan(D3D) DDRM
 /erand48, lrand48, nrand48, mrand48, jrand48, srand48,/ drand48(3C) PRM
 structure. ms: text formatting macros. ms(7) BSD
 operations. msgb: STREAMS message block msgb(D4DK) DDRM
 bytes in a message. msgctl: message control msgctl(2) PRM
 operations. msgdsz: return the number of msgdsz(D3DK) DDRM
 msgop: mpsgsnd, msgrcv: message msgop(2) PRM
 operations. msgrcv: message operations. msgop(2) PRM
 /mcmp, min, mout, pow, gcd, rpow, mpsgsnd, msgrcv: message msgop(2) PRM
 mout, pow, gcd,/ mp: madd, msqrt, sddiv, itom, xtom, mtom, mp(3X) BSD
 physical storage. msub, mult, mdiv, mcmp, min, mp(3X) BSD
 /rpow, msqrt, sddiv, itom, xtom, msync: synchronize memory with msync(3C) PRM
 pow, gcd,/ mp: madd, msub, mt: magnetic tape control. mt(1) BSD
 sgib: install mtom, mfree: multiple precision/ mp(3X) BSD
 rtc: mtune: file format. mtune(4) SARM
 drivers. fddd: MULTIBUS bootstrap loaders. sgib(1M) MBRMAN
 drivers. hddd: MULTIBUS clock driver. rtc(1M) MBRMAN
 bootserver: starts a MULTIBUS flexible disk device fddd(1M) MBRMAN
 format. config: MULTIBUS hard disk device hddd(1M) MBRMAN
 /icsgetrec: utilities to access MULTIBUS II boot service daemon. bootserver(1M) MBRMAN
 mb2a_receive: Receives incoming MULTIBUS II configuration file config(1M) MBRMAN
 mb2s_receive: Receives incoming MULTIBUS II interconnect space. icsrd(1M) MBRMAN
 /Closes a synchronous MULTIBUS II messages in. mb2a_receive(3tai) TAI
 /Closes an asynchronous MULTIBUS II messages in. mb2s_receive(3tai) TAI
 /Opens an asynchronous MULTIBUS II transport. mb2s_closeport(3tai) TAI
 MULTIBUS II transport endpoint. mb2a_closeport(3tai) TAI
 MULTIBUS II transport endpoint. mb2a_openport(3tai) TAI

/Opens a synchronous ramd: MULTIBUS II transport endpoint.. mb2s_openport(3tai) TAI
 tdd: MULTIBUS RAM disk driver. ramd(1M) MBRMAN
 mbchar: mbtowc, mblen, wctomb: MULTIBUS tape device drivers. tdd(1M) MBRMAN
 mbstring: mbstowcs, wcstombs: multibyte character handling. mbchar(3C) PRM
 /umountall: mount, unmount multibyte string functions. mbstring(3C) PRM
 /sdiv, itom, xtom, mtox, mfree: multiple file systems. mountall(1M) SARM
 unshareall: share, unshare multiple precision integer/ mp(3X) BSD
 unshareall: share, unshare multiple resources. shareall, shareall(1M) NUAG
 cpout: copy out file archives to multiple resources. shareall, shareall(1M) SARM
 used by xt driver. xtproto: multiple volumes. cpout(1M) MBRMAN
 used by xt driver. xtproto: multiplexed channels protocol xtproto(5) PRM
 windowing. xt: STREAMS-based multiplexed channels protocol xtproto(5) SARM
 windowing. xt: STREAMS-based multiplexed tty driver for AT&T xt(7) SARM
 poll: input/output multiplexed tty driver for AT&T xt(7) STRM
 poll: input/output multiplexing. poll(2) PRM
 select: synchronous I/O multiplexing. poll(2) STRM
 terminals. layers: layer multiplexing. select(3C) NI
 i546: iSBC 546 multiplexor for windowing layers(1) URM
 rc2: run commands performed for multi-port serial controller. i546(1M) MBRMAN
 in memory. mlock, multi-user environment. rc2(1M) SARM
 address space. mlockall, munlock: lock (or unlock) pages mlock(3C) PRM
 mlockall: lock or unlock mlockall(3C) PRM
 munmap: unmap pages of memory. munmap(2) PRM
 mv: move files. mv(1) URM
 mvaddch, mvwaddch, echochar,/ curs_addch(3X) CHAR
 mvaddchnstr, mvwaddchnstr,/ curs_addchstr(3X) CHAR
 mvaddchstr, mvaddchnstr,/ curs_addchstr(3X) CHAR
 mvaddnstr, mvwaddnstr,/ /addnstr, curs_addnstr(3X) CHAR
 mvaddstr, mvaddnstr, mvwaddstr,/ curs_addstr(3X) CHAR
 mvcur, tigetflag, tigetnum,/ curs_terminfo(3X) CHAR
 mvdelch, mvwdelch: delete/ curs_delch(3X) CHAR
 mvderwin, dupwin, wsyncup,/ curs_window(3X) CHAR
 mvdir: move a directory. mvdir(1M) SARM
 mvgetch, mvwgetch, ungetch: get curs_getch(3X) CHAR
 mvgetstr, mvwgetstr, wgetstr:/ curs_getstr(3X) CHAR
 mvinch, mvwinch: get a character curs_inch(3X) CHAR
 mvinchnstr, mvwinchnstr,/ curs_inchstr(3X) CHAR
 mvinchstr, mvinchnstr,/ /inchstr, curs_inchstr(3X) CHAR
 mvinnstr, mvwinstr, mvwinstr: curs_instr(3X) CHAR
 mvinsch, mvwinsch: insert a/ curs_insch(3X) CHAR
 mvinsnstr, mvwinsnstr,/ /insnstr, curs_insstr(3X) CHAR
 mvinsstr, mvinsnstr, mvwinsnstr,/ curs_insstr(3X) CHAR
 mvinstr, mvinnstr, mvwinstr,/ curs_instr(3X) CHAR
 mvprintw, mvwprintw, vwprintw:/ curs_printw(3X) CHAR
 mvscanw, mvwscanw, vwscanw:/ curs_scanw(3X) CHAR
 add a/ /addch, waddch, mvaddch, mvwaddchstr: add string of/ curs_addchstr(3X) CHAR

Permuted Index

string/ /mvaddchstr, mvaddchnstr, mvwaddchstr, mvwaddchnstr: add curs_addchstr(3X) CHAR
 /mvaddstr, mvaddnstr, mvwaddstr, mvwaddnstr: add a string of/ curs_addstr(3X) CHAR
 /waddnstr, mvaddstr, mvaddnstr: add a/ curs_addstr(3X) CHAR
 cursor/ /delch, wdelch, mvdelch, mvwdelch: delete character under curs_delch(3X) CHAR
 back/ /getch, wgetch, mvgetch, mvwgetch, ungetch: get (or push curs_getch(3X) CHAR
 /getstr, wgetstr, mvgetstr, mvwgetstr, wgetstr: get/ curs_getstr(3X) CHAR
 curs_window: newwin, delwin, mvwin, subwin, derwin, mvderwin,/ .. curs_window(3X) CHAR
 curs_inch: inch, winch, mvinch, mvwinch: get a character and its/ curs_inch(3X) CHAR
 /mvinchstr, mvwinchstr, mvwinchstr: get a string of/ curs_inchstr(3X) CHAR
 string/ /mvinchstr, mvinchnstr, mvwinchstr, mvwinchnstr: get a curs_inchstr(3X) CHAR
 /mvinstr, mvinnstr, mvwinstr, mvwininstr: get a string of/ curs_instr(3X) CHAR
 before/ /insch, winsch, mvinsch, mvwinsch: insert a character curs_insch(3X) CHAR
 /mvinsstr, mvinsnstr, mvwinsstr, mvwinsnstr: insert string before/ curs_insstr(3X) CHAR
 /winsstr, mvinsstr, mvinsnstr, mvwinsstr, mvwinsnstr: insert/ curs_insstr(3X) CHAR
 /winnstr, mvinstr, mvinnstr, mvwinstr, mvwininstr: get a/ curs_instr(3X) CHAR
 /printw, wprintw, mvprintw, mvwprintw: print/ curs_printw(3X) CHAR
 /scanw, wscanw, mvscanw, mvwscanw: convert/ curs_scanw(3X) CHAR
 devnm: device name. devnm(1M) SARM
 getlogin: get login name. getlogin(3C) PRM
 getsockname: get socket name. getsockname(3N) NI
 logname: get login name. logname(1) URM
 mkstemp: make a unique file name. mkstemp(3) BSD
 mktemp: make a unique file name. mktemp(3C) PRM
 pwd: working directory name. pwd(1) URM
 realpath: returns the real name. realpath(3C) PRM
 /item_description: get menu name and description. menu_item_name(3X) CHAR
 ID. id: print the user name and ID, and group name and ID. id(1M) SARM
 ID. id: print the user name and ID, and group name and ID. id(1M) URM
 the user name and ID, and group name and ID. id: print id(1M) SARM
 the user name and ID, and group name and ID. id: print id(1M) URM
 the last element of a path name. basename: return basename(3G) PRM
 hosts: host name data base. hosts(4) NI
 hosts: host name data base. hosts(4) NUAG
 hosts: host name data base. hosts(4) SARM
 networks: network name data base. networks(4) NUAG
 networks: network name data base. networks(4) SARM
 protocols: protocol name data base. protocols(4) NUAG
 protocols: protocol name data base. protocols(4) SARM
 whois: Internet user name directory service. whois(1) NUAG
 whois: Internet user name directory service. whois(1) URM
 tmpnam, tempnam: create a name for a temporary file. tmpnam(3S) PRM
 ctermid: generate file name for terminal. ctermid(3S) PRM
 descriptor. fdetach: detach name from a STREAMS-based file fdetach(1M) STRM
 descriptor. fdetach: detach name from a STREAMS-based file fdetach(3C) STRM
 getpw: get name from UID. getpw(3C) PRM
 return value for environment name. getenv: getenv(3C) PRM
 timezone: get time zone name given offset from GMT. timezone(3C) BSD

nlist: get entries from
 nm: print
 rename: change the
 /report the parent directory
 ttyname, isatty: find
 sdi_name: get
 getpeername: get
 /sethostname: get/set
 hostname: set or print
 domain. domainname: get/set
 uname: print
 uname: get
 pseudo-terminal/ ptsname: get
 tty: get the
 cuserid: get character login
 nlsprovider: get
 ypwhich: return
 directory name of a file path
 named, in.named: Internet domain
 named, in.named: Internet domain
 tnamed, in.tnamed: DARPA trivial
 tnamed, in.tnamed: DARPA trivial
 rfmaster: Remote File Sharing
 rfmaster: Remote File Sharing
 nsquery: Remote File Sharing
 nsquery: Remote File Sharing
 /configuration file for
 /configuration file for
 nslookup: query
 nslookup: query
 bind: bind a
 search for named file in
 pathfind: search for
 name server.
 name server.
 mailalias: translate mail alias
 system. ff (generic): list file
 file/ ff (ufs): list file
 deliver portions of path
 a list of all valid group
 a list of all valid group
 a list of all valid user
 a list of all valid user
 File Sharing domain and network
 File Sharing domain and network
 term: conventional
 term: conventional

 name list. nlist(3E) PRM
 name list of an object file. nm(1) PRM
 name of a file. rename(2) PRM
 name of a file path name. dirname(3G) PRM
 name of a terminal. ttyname(3C) PRM
 name of addressed controller. sdi_name(D3I) SCSI
 name of connected peer. getpeername(3N) NI
 name of current host. gethostname(3) BSD
 name of current host system. hostname(1) BSD
 name of current secure RPC domainname(1M) NI
 name of current UNIX system. uname(1) URM
 name of current UNIX system. uname(2) PRM
 name of the slave ptsname(3C) STRM
 name of the terminal. tty(1) URM
 name of the user. cuserid(3S) PRM
 name of transport provider. nlsprovider(3N) NI
 name of YP server or map master. ypwhich(1) NI
 name. /report the parent dirname(3G) PRM
 name server. named(1M) NUAG
 name server. named(1M) SARM
 name server. tnamed(1M) NUAG
 name server. tnamed(1M) SARM
 name server master file. rfmaster(4) NUAG
 name server master file. rfmaster(4) SARM
 name server query. nsquery(1M) NUAG
 name server query. nsquery(1M) SARM
 name server routines. resolv.conf(4) NUAG
 name server routines. resolv.conf(4) SARM
 name servers interactively. nslookup(1M) NUAG
 name servers interactively. nslookup(1M) SARM
 name to a socket. bind(3N) NI
 named directories. pathfind: pathfind(3G) PRM
 named file in named directories. pathfind(3G) PRM
 named, in.named: Internet domain named(1M) NUAG
 named, in.named: Internet domain named(1M) SARM
 names. mailalias(1) URM
 names and statistics for a file ff(1M) SARM
 names and statistics for a ufs ff(1M) SARM
 names. basename, dirname: basename(1) URM
 names. dispgid: displays dispgid(1) SARM
 names. dispgid: displays dispgid(1) SS
 names. dispuid: displays dispuid(1) SARM
 names. dispuid: displays dispuid(1) SS
 names. dname: print Remote dname(1M) NUAG
 names. dname: print Remote dname(1M) SARM
 names for terminals. term(5) CHAR
 names for terminals. term(5) SARM

file/ ncheck (s5): generate path
 generate a list of path
 /generic transport
 short interval.
 setsyx, ripoffline, curs_set,
 nl_types:
 nl_types:
 processing language.
 access to a resource/ waitsem,
 list of path names vs/
 versus i-numbers for s5 file/
 versus i-numbers for ufs file/
 dbm_delete, dbm_error,/
 mathematics. eqn,
 database.
 database.
 taddr2uaddr,/ /netdir_getbyaddr,
 netdir_getbyname,
 netdir_getbyaddr, netdir_free,/
 /netdir_getbyaddr, netdir_free,
 /taddr2uaddr, uaddr2taddr,
 /uaddr2taddr, netdir_perror,
 /key_gendes, key_setsecret,
 /key_setsecret, netname2host,
 /entry corresponding to
 data.
 data.
 rwall: write to all users over a
 rwall: write to all users over a
 convert values between host and
 entry. getnetconfig: get
 netconfig:
 netconfig:
 setnetent, endnetent: get
 /sethostent, endhostent: get
 ICMP ECHO_REQUEST packets to
 ICMP ECHO_REQUEST packets to
 lo: software loopback
 lo: software loopback
 ifconfig: configure
 ifconfig: configure
 properties of Internet Protocol
 listen:
 names versus i-numbers for s5 ncheck(1M) SARM
 names vs i-numbers. /(generic): ncheck(1M) SARM
 name-to-address translation. netdir(3N) NI
 nap: suspend execution for a nap(2) XNX
 napms: low-level curses/ /getsyx, curs_kernel(3X) CHAR
 native language data types. nl_types(5) PRM
 native language data types. nl_types(5) SARM
 nawk: pattern scanning and nawk(1) URM
 nbwaitsem: await and check waitsem(2) XNX
 ncheck (generic): generate a ncheck(1M) SARM
 ncheck (s5): generate path names ncheck(1M) SARM
 ncheck (ufs): generate pathnames ncheck(1M) SARM
 ndbm: dbm_clearerr, dbm_close, ndbm(3) BSD
 neqn, checkeq: typeset eqn(1) BSD
 netconfig: network configuration netconfig(4) NI
 netconfig: network configuration netconfig(4) SARM
 netdir_free, netdir_mergeaddr, netdir(3N) NI
 netdir_getbyaddr, netdir_free,/
 netdir_getbyname, netdir(3N) NI
 netdir_getbyaddr, netdir_free,/
 netdir_mergeaddr, taddr2uaddr,/ netdir(3N) NI
 netdir_perror, netdir_sperror:/ netdir(3N) NI
 netdir_sperror: generic/ netdir(3N) NI
 netname2host, netname2user,/ secure_rpc(3N) NI
 netname2user, user2netname:/ secure_rpc(3N) NI
 NETPATH component. getnetpath(3N) NI
 netrc: file for ftp remote login netrc(4) NUAG
 netrc: file for ftp remote login netrc(4) SARM
 netstat: show network status. netstat(1M) NUAG
 netstat: show network status. netstat(1M) SARM
 network. rwall(1M) NI
 network. rwall(1M) NUAG
 network byte order. /ntohs: byteorder(3N) NI
 network configuration database getnetconfig(3N) NI
 network configuration database. netconfig(4) NI
 network configuration database. netconfig(4) SARM
 network entry. /getnetbyname, getnetent(3N) NI
 network host entry. gethostent(3N) NI
 network hosts. ping: send ping(1M) NUAG
 network hosts. ping: send ping(1M) SARM
 network interface. lo(7) NUAG
 network interface. lo(7) SARM
 network interface parameters. ifconfig(1M) NUAG
 network interface parameters. ifconfig(1M) SARM
 network interfaces. if: general if(7) NUAG
 network listener daemon. listen(1M) SARM

administration. nlsadmin: network listener service nlsadmin(1M) SARM
 lockd: network lock daemon. lockd(1M) NUAG
 networks: network name data base. networks(4) NUAG
 networks: network name data base. networks(4) SARM
 Remote File Sharing domain and network names. dname: print dname(1M) NUAG
 Remote File Sharing domain and network names. dname: print dname(1M) SARM
 routed: network routing daemon. routed(1M) NUAG
 routed: network routing daemon. routed(1M) SARM
 system supporting for packet network routing. routing: routing(4) NUAG
 system supporting for packet network routing. routing: routing(4) SARM
 rpc.rwalld: network rwall server. rwalld(1M) NI
 rpc.rwalld: network rwall server. rwalld(1M) NUAG
 data in order to check the network. spray: scatter spray(3N) NI
 netstat: show network status. netstat(1M) NUAG
 netstat: show network status. netstat(1M) SARM
 statd: network status monitor. statd(1M) NUAG
 rpc.rusersd: network username server. rusersd(1M) NI
 rpc.rusersd: network username server. rusersd(1M) NUAG
 base. networks: network name data networks(4) NUAG
 base. networks: network name data networks(4) SARM
 base for the mail aliases file. newaliases: rebuild the data newaliases(1M) BSD
 link_field,/ form_field_new: new_field, dup_field, form_field_new(3X) CHAR
 form_fieldtype: new_fieldtype, free_fieldtype,/ form_fieldtype(3X) CHAR
 text file. newform: change the format of a newform(1) URM
 destroy forms. form_new: new_form, free_form: create and form_new(3X) CHAR
 system. newfs: construct a new file newfs(1M) BSD
 newgrp: log in to a new group. newgrp(1M) SARM
 newgrp: log in to a new group. newgrp(1M) URM
 destroy menus/ menu_item_new: new_item, free_item: create and menu_item_new(3X) CHAR
 publickey database. newkey: create a new key in the newkey(1M) NI
 publickey database. newkey: create a new key in the newkey(1M) NUAG
 destroy menus. menu_new: new_menu, free_menu: create and menu_new(3X) CHAR
 pnoutrefresh,/ curs_pad: newpad, subpad, prefresh, curs_pad(3X) CHAR
 form_new_page: set_new_page: new_page: forms pagination. form_new_page(3X) CHAR
 destroy panels. panel_new: new_panel, del_panel: create and panel_new(3X) CHAR
 news: print news items. news(1) URM
 news: print news items. news(1) URM
 curs_initscr: initscr, newterm, endwin, isendwin,/ curs_initscr(3X) CHAR
 derwin, mvderwin,/ curs_window: newwin, delwin, mvwin, subwin, curs_window(3X) CHAR
 bgets: read stream up to next delimiter. bgets(3G) PRM
 getq: get the next message from a queue. getq(D3DK) DDRM
 getmsg: get next message off a stream. getmsg(2) PRM
 getmsg: get next message off a stream. getmsg(2) STRM
 putnext: send a message to the next queue. putnext(D3DK) DDRM
 SAMESTR: test if next queue is same type. SAMESTR(D3DK) DDRM

Permuted Index

frexp, ldexp, logb, modf, modff,
 /fetch, store, delete, firstkey,
 /fetch, store, delete, firstkey,
 biod:
 biod:
 nfsd:
 nfsd:
 automount: automatically mount
 automount: automatically mount
 mountd:
 mountd:
 mounting by/ share: make local
 mounting by/ share: make local
 dfmounts: display mounted
 dfmounts: display mounted
 mounting by/ unshare: make local
 mounting by/ unshare: make local
 mount: mount remote
 mount: mount remote
 dfshares: list available
 dfshares: list available

 ftw,
 time-sharing process.
 process.
 priority.

 /setscreg, wsetscreg, scrollok,
 list.
 table.
 information.
 service administration.
 passed via the listener.
 transport provider.
 listener service request/
 types.
 types.
 file.
 curs_inopts: cbreak,
 /intrflush, keypad, meta,
 of nodes. idmknod: removes
 /information on an ethernet
 and reads specifications of
 keypad,/ /cbreak, nocbreak, echo,
 being scheduled.
 hangups and quits.

 nextafter, scalb: manipulate. frexp(3C) PRM
 nextkey: data base subroutines. dbm(3X) BSD
 nextkey: database subroutines. dbm(3) NI
 NFS daemon. biod(1M) NUAG
 NFS daemon. biod(1M) SARM
 NFS daemon. nfsd(1M) NUAG
 NFS daemon. nfsd(1M) SARM
 NFS file systems. automount(1M) NUAG
 NFS file systems. automount(1M) SARM
 NFS mount request server. mountd(1M) NUAG
 NFS mount request server. mountd(1M) SARM
 NFS resource available for share(1M) NUAG
 NFS resource available for share(1M) SARM
 NFS resource information. dfmounts(1M) NUAG
 NFS resource information. dfmounts(1M) SARM
 NFS resource unavailable for unshare(1M) NUAG
 NFS resource unavailable for unshare(1M) SARM
 NFS resources. mount(1M) NUAG
 NFS resources. mount(1M) SARM
 NFS resources from remote/ dfshares(1M) NUAG
 NFS resources from remote/ dfshares(1M) SARM
 nfsd: NFS daemon. nfsd(1M) NUAG
 nfsd: NFS daemon. nfsd(1M) SARM
 nftw: walk a file tree. ftw(3C) PRM
 nice: change priority of a nice(2) PRM
 nice: change priority of a nice(3C) BSD
 nice: run a command at low nice(1) URM
 nl: line numbering filter. nl(1) URM
 nl, nonl: curses terminal output/ curs_outopts(3X) CHAR
 nlist: get entries from name nlist(3E) PRM
 nlist: get entries from symbol nlist(3) BSD
 nl_langinfo: language nl_langinfo(3C) PRM
 nlsadmin: network listener nlsadmin(1M) SARM
 nlsgetcall: get client's data nlsgetcall(3N) NI
 nlsprovider: get name of nlsprovider(3N) NI
 nlsrequest: format and send nlsrequest(3N) NI
 nl_types: native language data nl_types(5) PRM
 nl_types: native language data nl_types(5) SARM
 nm: print name list of an object nm(1) PRM
 nocbreak, echo, noecho,/ curs_inopts(3X) CHAR
 nodelay, notimeout, raw, noraw,/ curs_inopts(3X) CHAR
 nodes and reads specifications idmknod(1M) SARM
 node's ethernet controller. enetinfo(1M) MBRMAN
 nodes. idmknod: removes nodes idmknod(1M) SARM
 noecho, halfdelay, intrflush, curs_inopts(3X) CHAR
 noenable: prevent a queue from noenable(D3DK) DDRM
 nohup: run a command immune to nohup(1) URM

/wsetscreg, scrollok, nl, setjmp, longjmp:
 _longjmp, sigsetjmp, siglongjmp:
 state. sigsetjmp, siglongjmp: a
 chpoll: poll entry point for a
 /nodelay, notimeout, raw, noraw,
 /meta, nodelay, notimeout, raw,
 rfuadmin: Remote File Sharing
 rfuadmin: Remote File Sharing
 arrival of new mail.
 new mail. notify:
 /keypad, meta, nodelay,
 drand48, erand48, lrand48,
 report possible/ checknr: check
 display or line-printer.
 tbl: format tables for
 and eliminate .so requests from
 constructs. deroff: remove
 constructs. deroff: remove
 interactively.
 interactively.
 name server query.
 name server query.
 byteorder, htonl, htons,
 byteorder, htonl, htons, ntohl,
 null: the

 /dodisk, lastlogin, monacct,
 random: generate a random
 number to pass-through device
 rpc: rpc program
 obtain the prime factors of a
 hat_getkpfnum: get page frame
 determine type of floating-point
 makedevice: make device
 rand, srand: simple random
 /setstate: better random
 returns the current frameID
 major or internal major device
 minor or internal minor device
 major, minor: manage a device
 addresses to RPC program
 addresses to RPC program
 bzero: clear memory for a given
 msgdsiz: return the
 execution for a specified
 directory or/ du: display the

 nonl: curses terminal output/ curs_outopts(3X) CHAR
 non-local goto. setjmp(3C) PRM
 non-local goto. /_setjmp, setjmp(3) BSD
 non-local goto with signal sigsetjmp(3C) PRM
 non-STREAMS character driver. chpoll(D2DK) DDRM
 noqiflush, qiflush, timeout,/ curs_inopts(3X) CHAR
 noraw, noqiflush, qiflush,/ curs_inopts(3X) CHAR
 notification shell script. rfuadmin(1M) NUAG
 notification shell script. rfuadmin(1M) SARM
 notify: notify user of the notify(1) URM
 notify user of the arrival of notify(1) URM
 notimeout, raw, noraw,/ curs_inopts(3X) CHAR
 nrand48, mrand48, jrand48,/ drand48(3C) PRM
 nroff and troff input files; checknr(1) BSD
 nroff: format documents for nroff(1) BSD
 nroff or troff. tbl(1) BSD
 nroff or troff input. /resolve soelim(1) BSD
 nroff, troff, tbl and eqn deroff(1) BSD
 nroff/troff, tbl, and eqn deroff(1) URM
 nslookup: query name servers nslookup(1M) NUAG
 nslookup: query name servers nslookup(1M) SARM
 nsquery: Remote File Sharing nsquery(1M) NUAG
 nsquery: Remote File Sharing nsquery(1M) SARM
 ntohl, ntohs: convert values/ byteorder(3N) NI
 ntohs: convert values between/ byteorder(3N) NI
 null file. null(7) SARM
 null: the null file. null(7) SARM
 nulladm, prctmp, prdaily,/ acctsh(1M) SARM
 number. random(1) XNX
 number. /convert device sdi_getdev(D3I) SCSI
 number data base. rpc(4) NI
 number. factor: factor(1) URM
 number for kernel address. hat_getkpfnum(D3DK) DDRM
 number. /fpclass, unordered: isnan(3C) PRM
 number from major and minor/ makedevice(D3DK) DDRM
 number generator. rand(3C) BSD
 number generator; routines for/ random(3) BSD
 number. getfrm: getfrm(1F) CHAR
 number. getmajor: get getmajor(D3DK) DDRM
 number. getminor: get getminor(D3DK) DDRM
 number. makedev, makedev(3C) PRM
 number mapper. /universal rpcbind(1M) NI
 number mapper. /universal rpcbind(1M) NUAG
 number of bytes. bzero(D3DK) DDRM
 number of bytes in a message. msgdsiz(D3DK) DDRM
 number of clock ticks. /process delay(D3DK) DDRM
 number of disk blocks used per du(1M) BSD

files. df (generic): report
files. df (generic): report
/reads or writes a specified
 qsize: find the
page structure to page frame
string to double-precision
page_numtopp: convert page frame
sdi_getdev: convert device
gcvt: convert floating-point
 nl: line
install random inode generation
distributed pseudo-random
to system calls and error
number from major and minor
/menu_format: set and get maximum
help: ask for help with message
 localeconv: get
current host. hostid: print the
dlclose: close a shared
dlopen: open a shared
 .ott: FACE
file descriptor to an
 dis:
address of a symbol in shared
DMD terminal. wtnit:
dump: dump selected parts of an
 elf_end: finish using an
 nm: print name list of an
 elf:
get the base offset for an
/retrieve class-dependent
the comment section of an
find printable strings in an
 cof2elf: COFF to ELF
return the size of an
filehdr: file header for common
 ld: link editor for
print section sizes in bytes of
find ordering relation for an
/loads 8086 absolute library and
number. factor:
for physical/ physmap:
a process that an event has
od:
record in the interconnect space
in,/ srchtxt: display contents
number of free disk blocks and df(1M) SARM
number of free disk blocks and df(1M) URM
number of interconnect. ics_rdwr(D3D) DDRM
number of messages on a queue. qsize(D3DK) DDRM
number. page_pptonum: convert page_pptonum(D3DK) DDRM
number. strtod, atof;: convert strtod(3C) PRM
number to page structure. page_numtopp(D3DK) DDRM
number to pass-through device/ sdi_getdev(D3I) SCSI
number to string. ecvt, fcvt, ecvt(3C) PRM
numbering filter. nl(1) URM
numbers. fsrand: fsrand(1) BSD
numbers. /generate uniformly drand48(3C) PRM
numbers. intro: introduction intro(2) PRM
numbers.. /make device makedevice(D3DK) DDRM
numbers of rows and columns in/ menu_format(3X) CHAR
numbers or SCCS commands. help(1) PRM
numeric formatting information. localeconv(3C) PRM
numeric identifier of the hostid(1) BSD
object. dlclose(3X) PRM
object. dlopen(3X) PRM
object architecture information. ott(4) SARM
object. /attach a STREAMS-based fattach(3C) STRM
object code disassembler. dis(1) PRM
object. dlsym: get the dlsym(3X) PRM
object downloader for the 5620 wtnit(1M) SARM
object file. dump(1) PRM
object file. elf_end(3E) PRM
object file. nm(1) PRM
object file access library. elf(3E) PRM
object file. elf_getbase: elf_getbase(3E) PRM
object file header. elf_getehdr(3E) PRM
object file. mcs: manipulate mcs(1) PRM
object file or binary. strings: strings(1) URM
object file translation. cof2elf(1) PRM
object file type. /elf32_fsize: elf_fsize(3E) PRM
object files. filehdr(4) SARM
object files. ld(1) PRM
object files. size: size(1) PRM
object library. lorder: lorder(1) PRM
object module formats. download(1M) MBRMAN
obtain the prime factors of a factor(1) URM
obtain virtual address mapping physmap(D3DK) DDRM
occurred. pollwakeup: inform pollwakeup(D3DK) DDRM
octal dump. od(1) URM
od: octal dump. od(1) URM
of. /finds a specific ics_find_rec(1M) MBRMAN
of, or search for a text string srchtxt(1) URM

messages that are not part
 turn file system quotas on and
 /tell if forms field has
 elf_getbase: get the base
 get time zone name given
 offsetof:
 member.
 login: sign
 host with a subchannel on a line
 remote system with debugging
 new file or rewrite an existing
 /send a control message with a
 keyword. whatis: display a
 mps_AMPcancel: cancels an
 ungetc: push character back
 opensem:
 dlopen:
 fopen, freopen, fdopen:
 fopen, freopen, fdopen:
 on a STREAMS driver. clone:
 command. p2open, p2close:
 dup2: duplicate an
 dup: duplicate an
 open:
 writing.
 catopen, catclose:
 seekdir, rewinddir, / directory:
 closes a previously
 control system log. syslog,
 mps_open_chan:
 transport/ mb2s_openport:
 II transport/ mb2a_openport:
 sockio: ioctl's that
 reboot: restart the
 prf:
 commands performed to stop the
 performed to stop and reboot the
 sdi_icmd: perform requested
 ethers: Ethernet address mapping
 ethers: Ethernet address mapping
 index, rindex: string
 msgctl: message control
 msgop: msgsnd, msgrcv: message
 semctl: semaphore control
 semop: semaphore

of. /sends unsolicited mps_AMPsend(D3D) DDRM
 off. quotaon, quotaoff: quotaon(1M) SARM
 off-screen data ahead or behind. form_data(3X) CHAR
 offset for an object file. elf_getbase(3E) PRM
 offset from GMT. timezone: timezone(3C) BSD
 offset of structure member. offsetof(3C) PRM
 offsetof: offset of structure offsetof(3C) PRM
 on. login(1) URM
 on. /disassociates this ccidetach(1M) MBRMAN
 on. Utry: try to contact Utry(1M) SARM
 one. creat: create a creat(2) PRM
 one-byte parameter to a queue. putctl1(D3DK) DDRM
 one-line summary about a whatis(1) BSD
 ongoing rsvp transaction. mps_AMPcancel(D3D) DDRM
 onto input stream. ungetc(3S) PRM
 open a semaphore. opensem(2) XNX
 open a shared object. dlopen(3X) PRM
 open a stream. fopen(3S) BSD
 open a stream. fopen(3S) PRM
 open any major/minor device pair clone(7) STRM
 open, close pipes to and from a p2open(3G) PRM
 open file descriptor. dup2(3C) PRM
 open file descriptor. dup(2) PRM
 open for reading or writing. open(2) PRM
 open: gain access to a device. open(D2DK) DDRM
 open: open for reading or open(2) PRM
 open/close a message catalogue. catopen(3C) PRM
 opendir, readdir, telldir, directory(3C) PRM
 opened channel. mps_close_chan: mps_close_chan(D3D) DDRM
 openlog, closelog, setlogmask: syslog(3) BSD
 opens a channel. mps_open_chan(D3D) DDRM
 Opens a synchronous MULTIBUS II ... mb2s_openport(3tai) TAI
 Opens an asynchronous MULTIBUS ... mb2a_openport(3tai) TAI
 opensem: open a semaphore. opensem(2) XNX
 operate directly on sockets. sockio(7) NI
 operating system. reboot(1M) BSD
 operating system profiler. prf(7) SARM
 operating system. rc0: run rc0(1M) SARM
 operating system. /run commands rc6(1M) SARM
 operation immediately. sdi_icmd(D3I) SCSI
 operations. ethers(3N) NI
 operations. ethers(3N) NUAG
 operations. index(3) BSD
 operations. msgctl(2) PRM
 operations. msgop(2) PRM
 operations. semctl(2) PRM
 operations. semop(2) PRM

Permuted Index

shmctl: shared memory control
 bzero, ffs: bit and byte string
 report on completed backup
 display the status of backup
 memcopy, memmove, memset: memory
 shmat, shmdt: shared memory
 strcasemcpy, strncasemcpy: string
 strcspn, strtok, strstr: string
 rewinddir, closedir: directory
 bkoper: interact with backup
 join: relational database
 (generic): copy file systems for
 curses: CRT screen handling and
 /typeahead: curses terminal input
 /nl, nonl: curses terminal output
 vector. getopt: get
 /field_opts: forms field
 form_opts_off, form_opts: forms
 menu_opts_off, menu_opts: menus
 manufacturer's/ mdl: read and
 fcntl: file control
 fcntl: file control
 getopt: parse command
 stty: set the
 stty: set the
 endpoint. t_optmgmt: manage
 getoptcv: parse command
 setsockopt: get and set
 /mvgetch, mvwgetch, ungetch: get
 mlock, munlock: lock
 commands executed, in reverse
 between host and network byte
 spray: scatter data in
 library. lorder: find
 t_sndrel: initiate an
 /acknowledge receipt of an
 a directory, or a special or
 a directory, or a special or
 filesystem: file system
 restore: restore file to
 device/ ots: System V/386
 partner queue.
 V/386 OSI Transport Service
 Service (ots) device driver.
 information.
 cpio: copy file archives in and
 I/O port.
 operations. shmctl(2) PRM
 operations. /bcopy, bcmp, bstring(3) BSD
 operations. bkhistory: bkhistory(1M) SARM
 operations. bkstatus: bkstatus(1M) SARM
 operations. /memchr, memcmp, memory(3C) PRM
 operations. shmop: shmop(2) PRM
 operations. string: string(3) BSD
 operations. /strpbrk, strspn, string(3C) PRM
 operations. /telldir, seekdir, directory(3C) PRM
 operations to service media/ bkoper(1M) SARM
 operator. join(1) URM
 optimal access time. dcopy dcopy(1M) SARM
 optimization package. curses(3X) CHAR
 option control routines. curs_inopts(3X) CHAR
 option control routines. curs_outopts(3X) CHAR
 option letter from argument getopt(3C) PRM
 option routines. form_field_opts(3X) CHAR
 option routines. /form_opts_on, form_opts(3X) CHAR
 option routines. /menu_opts_on, menu_opts(3X) CHAR
 optionally write the mdl(1M) MBRMAN
 options. fcntl(5) PRM
 options. fcntl(5) SARM
 options. getopt(1) URM
 options for a terminal. stty(1) BSD
 options for a terminal. stty(1) URM
 options for a transport t_optmgmt(3N) NI
 options. getopts, getopts(1) URM
 options on sockets. getsockopt, getsockopt(3N) NI
 (or push back) characters from/ curs_getch(3X) CHAR
 (or unlock) pages in memory. mlock(3C) PRM
 order. lastcomm: show the last lastcomm(1) BSD
 order. /ntohs: convert values byteorder(3N) NI
 order to check the network. spray(3N) NI
 ordering relation for an object lorder(1) PRM
 orderly release. t_sndrel(3N) NI
 orderly release indication. t_rcvrel(3N) NI
 ordinary file. mknod: make mknod(2) PRM
 ordinary file. mknod: make mknod(2) XNX
 organization. filesystem(7) SARM
 original directory. restore(1) XNX
 OSI Transport Service (ots) ots(1M) MBRMAN
 OTHERQ: get pointer to queue's OTHERQ(D3DK) DDRM
 (ots) device driver. /System ots(1M) MBRMAN
 ots: System V/386 OSI Transport ots(1M) MBRMAN
 .ott: FACE object architecture ott(4) SARM
 out. cpio(1) URM
 outb: write a byte to an 8-bit outb(D3D) DDRM

connection. dial: establish an outgoing terminal line dial(3C) NI
 to a 32-bit I/O port. outl: write a 32-bit long word outl(D3D) DDRM
 echo: put string on virtual output. echo(1F) CHAR
 printf: print formatted output. printf(1) URM
 seconvert, sfconvert, sgconvert: output conversion. /gconvert, econvert(3) BSD
 vfprintf, vsprintf: formatted output conversion. /vprintf, printf(3S) BSD
 /vwprintw: print formatted output in curses windows. curs_printw(3X) CHAR
 list. /vsprintf: print formatted output of a variable argument vprintf(3S) PRM
 /nl, nonl: curses terminal output option control routines. curs_outopts(3X) CHAR
 information. ttyadm: format and output port monitor-specific ttyadm(1M) SARM
 sprintf: print formatted output. printf, fprintf, printf(3S) PRM
 to a 16-bit I/O port. outw: write a 16-bit short word outw(D3D) DDRM
 /overlay, overwrite, copywin: overlap and manipulate/ curs_overlay(3X) CHAR
 /copywin: overlap and manipulate overlapped curses windows. curs_overlay(3X) CHAR
 overlay and/ curs_overlay: overlay, overwrite, copywin: curs_overlay(3X) CHAR
 curs_overlay: overlap and/ curs_overlay(3X) CHAR
 chown: change file owner. chown(1) BSD
 chown: change file owner. chown(1) URM
 chown, lchown, fchown: change owner and group of a file. chown(2) PRM
 quot: summarize file system ownership. quot(1M) SARM
 initialize file permissions and ownership. fixperm: correct or fixperm(1) XNX
 chgrp: change the group ownership of a file. chgrp(1) URM
 XENIX file permissions and ownership. /or initialize fixperm(1M) SARM
 and from a command. p2open, p2close: open, close pipes to p2open(3G) PRM
 pipes to and from a command. p2open, p2close: open, close p2open(3G) PRM
 expand files. pack, pcat, unpack: compress and pack(1) URM
 forms: character based forms package. forms(3X) CHAR
 menus: character based menus package. menus(3X) CHAR
 panels: character based panels package. panels(3X) CHAR
 pkgmk: produce an installable package. pkgmk(1) SS
 pkginfo: package characteristics file. pkginfo(4) SS
 file. pkgmap: package contents description pkgmap(4) SS
 screen handling and optimization package. curses: CRT curses(3X) CHAR
 specific portions of a UNIX package. custom: install custom(1M) SARM
 specific portions of a UNIX package. custom: install custom(1) XNX
 pkgtrans: translate package format. pkgtrans(1) SARM
 pkgtrans: translate package format. pkgtrans(1) SS
 pkgrm: removes a package from the system. pkgrm(1M) SARM
 pkgrm: removes a package from the system. pkgrm(1M) SS
 pkginfo: display software package information. pkginfo(1) SARM
 pkginfo: display software package information. pkginfo(1) SS
 prototype: package information file. prototype(4) SS
 pktparam: displays package parameter values. pktparam(1) SARM
 pktparam: displays package parameter values. pktparam(1) SS
 sadc: system activity report package. sar: sa1, sa2, sar(1M) SARM
 standard buffered input/output package. stdio: stdio(3S) PRM
 interprocess communication package. stdipc: ftok: standard stdipc(3C) PRM

pkgadd: transfer software
 pkgadd: transfer software
 distributed file system
 distributed file system
 pckt: STREAMS
 routing: system supporting for
 routing: system supporting for
 xtt: extract and print xt driver
 spray: spray
 spray: spray
 ping: send ICMP ECHO_REQUEST
 ping: send ICMP ECHO_REQUEST
 create and display curses
 field_index: set forms current
 text file. more,
 /convert page structure to
 address. hat_getkpfnum: get
 page_numtopp: convert
 pagesize: display the size of a
 getpagesize: get system
 convert page frame number to
 number. page_pptonum: convert
 more, page: browse or
 number to page structure.

 structure to page frame number.
 manual pages; find reference
 man: display reference manual
 mlock, munlock: lock (or unlock)
 to format Reference Manual
 determine residency of memory
 mmap: map
 munmap: unmap
 convert size in bytes to size in
 convert size in bytes to size in
 ptob: convert size in
 page of memory.
 set_new_page, new_page: forms
 socketpair: create a
 /open any major/minor device
 a pseudo-terminal master/slave
 /can_change_color, color_content,
 the current window of a panels
 dbon: sets target for front
 application data with a panels
 panel_below: panels deck/
 deck traversal/ panel_above:

 package to the system. pkgadd(1M) SARM
 package to the system. pkgadd(1M) SS
 packages. /file that registers fstypes(4) NUAG
 packages. /file that registers fstypes(4) SARM
 Packet Mode module. pckt(7) STRM
 packet network routing. routing(4) NUAG
 packet network routing. routing(4) SARM
 packet traces. xtt(1M) SARM
 packets. spray(1M) NI
 packets. spray(1M) NUAG
 packets to network hosts. ping(1M) NUAG
 packets to network hosts. ping(1M) SARM
 pads. /pnoutrefresh, pechochar: curs_pad(3X) CHAR
 page and field. /current_field, form_page(3X) CHAR
 page: browse or page through a more(1) URM
 page frame number. page_pptonum(D3DK) DDRM
 page frame number for kernel hat_getkpfnum(D3K) DDRM
 page frame number to page/ page_numtopp(D3DK) DDRM
 page of memory. pagesize(1) BSD
 page size. getpagesize(3) BSD
 page structure. page_numtopp: ... page_numtopp(D3DK) DDRM
 page structure to page frame page_pptonum(D3DK) DDRM
 page through a text file. more(1) URM
 page_numtopp: convert page frame
 page_numtopp(D3DK) DDRM
 page_pptonum: convert page page_pptonum(D3DK) DDRM
 pages by keyword. /reference man(1) BSD
 pages; find reference pages by/ man(1) BSD
 pages in memory. mlock(3C) PRM
 pages. man: macros man(7) BSD
 pages. mincore: mincore(2) PRM
 pages of memory. mmap(2) PRM
 pages of memory. munmap(2) PRM
 pages (round down). btop: btop(D3DK) DDRM
 pages (round up). btopr: btopr(D3DK) DDRM
 pages to size in bytes. ptob(D3DK) DDRM
 pagesize: display the size of a pagesize(1) BSD
 pagination. form_new_page: form_new_page(3X) CHAR
 pair of connected sockets. socketpair(3N) NI
 pair on a STREAMS driver. clone(7) STRM
 pair. unlockpt: unlock unlockpt(3C) STRM
 pair_content: curses color/ curs_color(3X) CHAR
 panel. /get or set panel_window(3X) CHAR
 panel message delivery. dbon(1M) MBRMAN
 panel. /panel_userptr: associate panel_userptr(3X) CHAR
 panel above: panel_above, panel_above(3X) CHAR
 panel_above, panel_below: panels panel_above(3X) CHAR

panel_above: panel_above,
 /show_panel, hide_panel,
 panels window on the virtual/
 del_panel: create and destroy/
 package.
 /hide_panel, panel_hidden:
 /top_panel, bottom_panel:
 /panel_above, panel_below:
 panels: character based
 application data with a
 or set the current window of a
 del_panel: create and destroy
 panel_update: update panels:
 panel_move: move_panel: move a
 hide_panel, panel_hidden:/
 bottom_panel: panels deck/
 panels virtual screen refresh/
 set_panel_userptr,/
 application/ /set_panel_userptr,
 replace_panel: get or set the/
 or set the/ panel_window:
 display an error message or
 me: macros for formatting
 lp:
 to set value of a tunable
 bootparamd: boot
 bootparamd: boot
 bps: Bootstrap
 initializes the bootstrap
 rt_dptbl: real-time dispatcher
 time-sharing dispatcher
 control message with a one-byte
 pkgparam: displays package
 pkgparam: displays package
 configure network interface
 configure network interface
 path name. dirname: report the
 get process, process group, and
 getopt:
 getopts, getoptcv:
 getsubopt:
 /wclrtoeol: clear all or
 tail: deliver the last
 connection. shutdown: shut down
 to a received request that is
 panel_below: panels deck/ panel_above(3X) CHAR
 panel_hidden: panels deck/ panel_show(3X) CHAR
 panel_move: move_panel: move a panel_move(3X) CHAR
 panel_new: new_panel, panel_new(3X) CHAR
 panels: character based panels panels(3X) CHAR
 panels deck manipulation/ panel_show(3X) CHAR
 panels deck manipulation/ panel_top(3X) CHAR
 panels deck traversal/ panel_above(3X) CHAR
 panels package. panels(3X) CHAR
 panels panel. /associate panel_userptr(3X) CHAR
 panels panel. /get panel_window(3X) CHAR
 panels. panel_new: new_panel, panel_new(3X) CHAR
 panels virtual screen refresh/ panel_update(3X) CHAR
 panels window on the virtual/ panel_move(3X) CHAR
 panel_show: show_panel, panel_show(3X) CHAR
 panel_top: top_panel, panel_top(3X) CHAR
 panel_update: update_panels: panel_update(3X) CHAR
 panel_userptr: panel_userptr(3X) CHAR
 panel_userptr: associate panel_userptr(3X) CHAR
 panel_window: panel_window, panel_window(3X) CHAR
 panel_window, replace_panel: get panel_window(3X) CHAR
 panic the system. cmn_err: cmn_err(D3DK) DDRM
 papers. me(7) BSD
 parallel port interface. lp(7) SARM
 parameter. idtune: attempts idtune(1M) SARM
 parameter server. bootparamd(1M) NI
 parameter server. bootparamd(1M) NUAG
 Parameter String driver. bps(1M) MBRMAN
 parameter string on the/ initbp: initbp(1M) MBRMAN
 parameter table. rt_dptbl(4) SARM
 parameter table. ts_dptbl: ts_dptbl(4) SARM
 parameter to a queue. /send a putctl(D3DK) DDRM
 parameter values. pkgparam(1) SARM
 parameter values. pkgparam(1) SS
 parameters. ifconfig: ifconfig(1M) NUAG
 parameters. ifconfig: ifconfig(1M) SARM
 parent directory name of a file dirname(3G) PRM
 parent process IDs. /getpgid: getpid(2) PRM
 parse command options. getopt(1) URM
 parse command options. getopts(1) URM
 parse suboptions from a string. getsubopt(3C) PRM
 part of a curses window. curs_clear(3X) CHAR
 part of a file. tail(1) URM
 part of a full-duplex shutdown(3N) NI
 part of a. /replies mps_AMPsend_reply(D3D) DDRM

sends solicited data that is not
 messages that are not
 ypset: point ypbind at a
 restores of filesystems, data
 OTHERQ: get pointer to queue's
 dump: dump selected
 nlsgetcall: get client's data
 mps: Message
 management.
 /convert device number to
 and password attributes.

getpass: read a
 functions. crypt:
 passwd: change login
 change login password and
 pwck: check
 passwd:
 shadow: shadow
 putpwent: write
 putspent: write shadow
 ulckpwwd: manipulate shadow
 endpwent, fgetpwent: manipulate
 passwd:
 change Remote File Sharing host
 change Remote File Sharing host
 pwck, grpck:
 several files or subsequent/
 create, remove directories in a
 return the last element of a
 parent directory name of a file
 dirname: deliver portions of
 s5 file/ ncheck (s5): generate
 (generic): generate a list of

pathname variables. fpathconf,
 for filename.
 in named directories.
 a prompt; verify and return a
 a prompt; verify and return a
 get current working directory
 directory. getcwd: get
 locate a command; display its
 pathconf: get configurable
 basename: display portions of
 ufs file/ ncheck (ufs): generate

part of any. mps_AMPsend_data:
 mps_AMPsend_data(D3D) DDRM
 part of. /sends unsolicited mps_AMPsend_data(D3D) DDRM
 particular server. ypset(1M) NI
 partitions, or disks. /initiate restore(1M) SARM
 partner queue. OTHERQ(D3DK) DDRM
 parts of an object file. dump(1) PRM
 passed via the listener. nlsgetcall(3N) NI
 Passing Space device driver. mps(1M) MBRMAN
 passwd: password files passwd(1M) SARM
 pass-through device number. sdi_getdev(D3I) SCSI
 passwd: change login password passwd(1) URM
 passwd: password file. passwd(4) SARM
 password. getpass(3C) PRM
 password and file encryption crypt(3X) PRM
 password and password/ passwd(1) URM
 password attributes. passwd: passwd(1) URM
 password database entries. pwck(1M) BSD
 password file. passwd(4) SARM
 password file. shadow(4) SARM
 password file entry. putpwent(3C) PRM
 password file entry. putspent(3C) PRM
 password file entry. /lckpwwd, getspent(3C) PRM
 password file entry. /setpwent, getpwent(3C) PRM
 password files management. passwd(1M) SARM
 password. rpasswd: rpasswd(1M) NUAG
 password. rpasswd: rpasswd(1M) SARM
 password/group file checkers. pwck(1M) SARM
 paste: merge same lines of paste(1) URM
 path. mkdirp, rmdirp: mkdirp(3G) PRM
 path name. basename: basename(3G) PRM
 path name. dirname: report the dirname(3G) PRM
 path names. basename, basename(1) URM
 path names versus i-numbers for ncheck(1M) SARM
 path names vs i-numbers. ncheck ncheck(1M) SARM
 pathalias: alias file for FACE. pathalias(4) SARM
 pathconf: get configurable fpathconf(2) PRM
 pathconv: search FMLI criteria pathconv(1F) CHAR
 pathfind: search for named file pathfind(3G) PRM
 pathname. ckpath: display ckpath(1) SARM
 pathname. ckpath: display ckpath(1) SS
 pathname. getwd: getwd(3) BSD
 pathname of current working getcwd(3C) PRM
 pathname or alias. which: which(1) BSD
 pathname variables. fpathconf, fpathconf(2) PRM
 pathnames. basename(1) BSD
 pathnames versus i-numbers for ncheck(1M) SARM

fmlgrep: search a file for a
grep: search a file for a
grep: search a file for a
generate lineprinter ripple
/menu_pattern: set and get menus
gmatch: shell global
language. **awk**:
language. **nawk**:
egrep: search a file for a
egrep: search a file for a
regex: match
signal.
expand files. **pack**,
module.
process. **popen**,
vax, 386, u370: **get/ machid**:
/subpad, prefetch, pnoutrefresh,
get name of connected
unbufcall: cancel a
biowait: suspend processes
sleep: suspend process activity
signals that are blocked and
the number of disk blocks used
backup:
strategy:
immediately. **sdi_icmd**:
sysadm: visual interface to
environment. **rc2**: run commands
operating/ **rc6**: run commands
system. **rc0**: run commands
driver. **i258**: ISBC 386/258
device driver. **i214**: ISBCR 214
device/ **i224a**: ISBC 186/224A
device driver. **i214tp**: ISBC 214
device/ **i224atp**: ISBC 186/224A
device/ **i258tp**: ISBC 386/258
/correct or initialize XENIX file
/correct or initialize file
check the uucp directories and
ckperms: set and/or verify
mesg:
format. **acct**:
acctcms: command summary from
messages.
pg: file
CRTs.
vtop: convert virtual to
pattern. **fmlgrep(1F)** CHAR
pattern. **grep(1)** URM
pattern. **grep(1)** XNX
pattern. **lptest**: **lptest(1)** BSD
pattern match buffer. **menu_pattern(3X)** CHAR
pattern matching. **gmatch(3G)** PRM
pattern scanning and processing **awk(1)** URM
pattern scanning and processing **nawk(1)** URM
pattern using full regular/ **egrep(1)** URM
pattern using full regular/ **egrep(1)** XNX
patterns against a string. **regex(1F)** CHAR
pause: suspend process until **pause(2)** PRM
pcat, unpack: compress and **pack(1)** URM
pckt: STREAMS Packet Mode **pckt(7)** STRM
pclose: initiate pipe to/from a **popen(3S)** PRM
pdp11, u3b, u3b2, u3b5, u3b15, **machid(1)** URM
pechochar: create and display/ **curs_pad(3X)** CHAR
peer. **getpeername**: **getpeername(3N)** NI
pending/ **unbufcall(D3DK)** DDRM
pending completion of block I/O. **biowait(D3DK)** DDRM
pending execution of an event. **sleep(D3DK)** DDRM
pending, sigpending: examine **sigpending(2)** PRM
per directory or file. **/display** **du(1M)** BSD
perform backup functions. **backup(1)** XNX
perform block I/O. **strategy(D2DK)** DDRM
perform requested operation **sdi_icmd(D3I)** SCSI
perform system administration. **sysadm(1M)** SARM
performed for multi-user **rc2(1M)** SARM
performed to stop and reboot the **rc6(1M)** SARM
performed to stop the operating **rc0(1M)** SARM
peripheral controller device **i258(1M)** MBRMAN
peripheral controller disk **i214(1M)** MBRMAN
peripheral controller disk **i224a(1M)** MBRMAN
peripheral controller tape **i214tp(1M)** MBRMAN
peripheral controller tape **i224atp(1M)** MBRMAN
peripheral controller tape **i258tp(1M)** MBRMAN
permissions and ownership. **fixperm(1M)** SARM
permissions and ownership. **fixperm(1)** XNX
permissions file. **uucheck**: **uucheck(1M)** SARM
permissions on files. **ckperms(1M)** MBRMAN
permit or deny messages. **mesg(1)** URM
per-process accounting file **acct(4)** SARM
per-process accounting records. **acctcms(1M)** SARM
per-perror: print system error **perror(3C)** PRM
perusal filter for CRTs. **pg(1)** URM
pg: file perusal filter for **pg(1)** URM
physical address. **vtop(D3D)** DDRM

kernel virtual address to physical address. /convert kvtophys(D3D) DDRM
 virtual address mapping for physical addresses. /obtain physmap(D3DK) DDRM
 msync: synchronize memory with physical storage. msync(3C) PRM
 in-memory state with that on the physical. /synchronize a file's fsync(2) PRM
 I/O request. physiock: validate and issue raw physiock(D3D) DDRM
 mapping for physical addresses. physmap: obtain virtual address physmap(D3DK) DDRM
 split: split a file into pieces. split(1) URM
 packets to network hosts. ping: send ICMP ECHO_REQUEST ping(1M) NUAG
 packets to network hosts. ping: send ICMP ECHO_REQUEST ping(1M) SARM
 channel. pipe: create an interprocess pipe(2) PRM
 tee: pipe fitting. tee(1) URM
 popen, pclose: initiate pipe to/from a process. popen(3S) PRM
 p2open, p2close: open, close pipes to and from a command. p2open(3G) PRM
 package to the system. pkgadd: transfer software pkgadd(1M) SARM
 package to the system. pkgadd: transfer software pkgadd(1M) SS
 request script. pkgask: stores answers to a pkgask(1M) SARM
 request script. pkgask: stores answers to a pkgask(1M) SS
 installation. pkgchk: check accuracy of pkgchk(1M) SARM
 installation. pkgchk: check accuracy of pkgchk(1M) SS
 package information. pkginfo: display software pkginfo(1) SARM
 package information. pkginfo: display software pkginfo(1) SS
 file. pkginfo: package characteristics pkginfo(4) SS
 description file. pkgmap: package contents pkgmap(4) SS
 package. pkgmk: produce an installable pkgmk(1) SS
 parameter values. pkgparam: displays package pkgparam(1) SARM
 parameter values. pkgparam: displays package pkgparam(1) SS
 file. pkgproto: generate a prototype pkgproto(1) SS
 the system. pkgrm: removes a package from pkgrm(1M) SARM
 the system. pkgrm: removes a package from pkgrm(1M) SS
 format. pkgtrans: translate package pkgtrans(1) SARM
 format. pkgtrans: translate package pkgtrans(1) SS
 queue. putbq: place a message at the head of a putbq(D3DK) DDRM
 unlock process, text, or data. plock: lock into memory or plock(2) PRM
 dumbplot, gigiplot, hpplot, / plot, aedplot, bgplot, crtplot, plot(1G) BSD
 graphics filters for various plotters. /t4013, t450, tek: plot(1G) BSD
 administration. pmadm: port monitor pmadm(1M) NI
 administration. pmadm: port monitor pmadm(1M) SARM
 images. pnch: file format for card pnch(4) SARM
 and/ /newpad, subpad, prefresh, pnoutrefresh, pechochar: create curs_pad(3X) CHAR
 floatingpoint: IEEE floating point definitions. floatingpoint(3) BSD
 character/ chpoll: poll entry point for a non-STREAMS chpoll(D2DK) DDRM
 server. ypset: point ypbind at a particular ypset(1M) NI
 elf_strptr: make a string pointer. elf_strptr(3E) PRM
 lseek: move read/write file pointer. lseek(2) PRM
 rewind, ftell: reposition a file pointer in a stream. fseek, fseek(3S) PRM
 fgetpos: reposition a file pointer in a stream. fsetpos, fsetpos(3C) PRM
 mps_get_dmabuf: returns a pointer to a list of data buffer/ mps_get_dmabuf(D3D) DDRM

queue. OTHERQ: get
 current queue. backq: get
 RD: get
 this module or driver. WR: get
 non-STREAMS character/ chpoll:

 that an event has occurred.
 buffer back into the free memory
 to/from a process.
 asy: asynchronous serial
 console: console
 read a byte from a 8-bit I/O
 a 32-bit word from a 32-bit I/O
 lp: parallel
 short word from a 16-bit I/O
 pmadm:
 pmadm:
 tymon:
 ttyadm: format and output
 write a byte to an 8-bit I/O
 32-bit long word to a 32-bit I/O
 short word to a 16-bit I/O
 bytes from buffer to an I/O
 words from buffer to an I/O
 words from buffer to an I/O
 repinsb: read bytes from I/O
 read 32-bit words from I/O
 read 16-bit words from I/O
 ar: maintain
 ar: maintain
 custom: install specific
 custom: install specific
 basename, dirname: deliver
 basename: display
 communications interface/
 and hunt sequences for TTY
 port monitor for terminal
 window cursor. form_cursor:
 /pos_menu_cursor: correctly
 form_cursor: pos_form_cursor:
 position a menus/ menu_cursor:
 and troff input files; report
 banner: make
 erase forms from/ form_post:
 erase menus from/ menu_post:
 /mult, mdiv, mcmp, min, mout,
 pointer to queue's partner OTHERQ(D3DK) DDRM
 pointer to the queue behind the backq(D3DK) DDRM
 pointer to the read queue. RD(D3DK) DDRM
 pointer to the write queue for WR(D3DK) DDRM
 poll entry point for a chpoll(D2DK) DDRM
 poll: input/output multiplexing. poll(2) PRM
 poll: input/output multiplexing. poll(2) STRM
 pollwakeup: inform a process pollwakeup(D3DK) DDRM
 pool. mps_free_msgbuf: puts a .. mps_free_msgbuf(D3D) DDRM
 popen, pclose: initiate pipe popen(3S) PRM
 port. asy(7) SARM
 port device driver. console(1M) MBRMAN
 port. inb: inb(D3D) DDRM
 port. inl: read inl(D3D) DDRM
 port interface. lp(7) SARM
 port. inw: read a 16 bit inw(D3D) DDRM
 port monitor administration. pmadm(1M) NI
 port monitor administration. pmadm(1M) SARM
 port monitor for terminal ports. ttymon(1M) SARM
 port monitor-specific/ ttyadm(1M) SARM
 port. outb: outb(D3D) DDRM
 port. outl: write a outl(D3D) DDRM
 port. outw: write a 16-bit outw(D3D) DDRM
 port. repoutsb: write repoutsb(D3D) DDRM
 port. repoutsd: write 32-bit repoutsd(D3D) DDRM
 port. repoutsw: write 16-bit repoutsw(D3D) DDRM
 port to buffer. repinsb(D3D) DDRM
 port to buffer. repinsd: repinsd(D3D) DDRM
 port to buffer. repinsw: repinsw(D3D) DDRM
 portable archive or library. ar(1) PRM
 portable archive or library. ar(1) URM
 portions of a UNIX package. custom(1M) SARM
 portions of a UNIX package. custom(1) XNX
 portions of path names. basename(1) URM
 portions of pathnames. basename(1) BSD
 ports: 5 line asynchronous ports(7) STRM
 ports. /maintain line settings sttydefs(1M) SARM
 ports. ttymon: ttymon(1M) SARM
 pos_form_cursor: position forms form_cursor(3X) CHAR
 position a menus cursor. menu_cursor(3X) CHAR
 position forms window cursor. form_cursor(3X) CHAR
 pos_menu_cursor: correctly menu_cursor(3X) CHAR
 possible errors. /check nroff checknr(1) BSD
 posters. banner(1) URM
 post_form, unpost_form: write or form_post(3X) CHAR
 post_menu, unpost_menu: write or menu_post(3X) CHAR
 pow, gcd, rpow, msqrt, sdiv,/ mp(3X) BSD

Permuted Index

cbrt, log, logf, log10, log10f, /sqrtf: exponential, logarithm, /log, logf, log10, log10f, pow, pow, powf, sqrt, sqrtf: / expf, exp(3M) PRM
 power, square root functions. exp(3M) PRM
 powf, sqrt, sqrtf: exponential,/ exp(3M) PRM
 pr: print files. pr(1) URM
 prctmp, prdaily, prtacct,/ acctsh(1M) SARM
 prdaily, prtacct, runacct,/ acctsh(1M) SARM
 preceding queue. put(D2DK) DDRM
 precision integer arithmetic. mp(3X) BSD
 .pref, .variables: environ(4) SARM
 prefetch, pnoutrefresh,/ curs_pad(3X) CHAR
 prepare execution profile. monitor(3C) PRM
 prevent a queue from being noenable(D3DK) DDRM
 previous get of an SCCS file. unget(1) PRM
 previous timeout(D3DK) function untimeout(D3DK) DDRM
 previously allocated kernel kmem_free(D3DK) DDRM
 previously allocated SCSI block. sdi_freeblk(D3I) SCSI
 previously allocated transaction mps_free_tid(D3D) DDRM
 previously opened channel. mps_close_chan(D3D) DDRM
 prf: operating system profiler. prf(7) SARM
 prfdc, prfsnap, prfpr: UNIX/ profiler(1M) SARM
 prfld, prfstat, prfdc, prfsnap, profiler(1M) SARM
 prfpr: UNIX system profiler. profiler(1M) SARM
 prfsnap, prfpr: UNIX system/ profiler(1M) SARM
 prfstat, prfdc, prfsnap, prfpr: profiler(1M) SARM
 primary memory. lock(2) XNX
 prime factors of a number. factor(1) URM
 primitive system data types. types(5) PRM
 primitives. /panel_below: panel_above(3X) CHAR
 print a bibliographic database. roffb(1) BSD
 print an SCCS file. prs(1) PRM
 print and set the date. date(1) URM
 print calendar. cal(1) URM
 print checksum and block count sum(1) URM
 print current SCCS file editing sact(1) PRM
 print: display a driver message print(D2DK) DDRM
 print file differences sdiff(1) URM
 print files. cat(1) URM
 print files. pr(1) URM
 print formatted output. printf(1) URM
 print formatted output. printf(3S) PRM
 print formatted output in curses/ curs_printw(3X) CHAR
 print formatted output of a/ vprintf(3S) PRM
 print group membership of user. groups(1) URM
 print identification strings. what(1) PRM
 print information about the lpstat(1) URM
 print name list of an object nm(1) PRM
 print name of current host hostname(1) BSD

system. uname: print name of current UNIX uname(1) URM
 news: print news items. news(1) URM
 infocmp: compare or print out terminfo descriptions. infocmp(1M) CHAR
 infocmp: compare or print out terminfo descriptions. infocmp(1M) SARM
 file(s). acctcom: search and print process accounting acctcom(1) URM
 and network names. dname: print Remote File Sharing domain dname(1M) NUAG
 and network names. dname: print Remote File Sharing domain dname(1M) SARM
 accept, reject: accept or reject print requests. accept(1M) SARM
 object files. size: print section sizes in bytes of size(1) PRM
 lpadmin: configure the LP print service. lpadmin(1M) SARM
 filters used with the LP print service. /administer lpfilter(1M) SARM
 forms used with the LP print service. /administer lpforms(1M) SARM
 /lpmove: start/stop the LP print service and move requests. lpsched(1M) SARM
 about the status of the LP print service. /information lpstat(1) URM
 send/cancel requests to an LP print service. lp, cancel: lp(1) URM
 register remote systems with the print service. lpsystem: lpsystem(1M) SARM
 jwin: print size of layer. jwin(1) URM
 sendmail. mailstats: print statistics collected by mailstats(1M) BSD
 strace: print STREAMS trace messages. strace(1M) SARM
 strace: print STREAMS trace messages. strace(1M) STRM
 yes: print string repeatedly. yes(1) XNX
 perror: print system error messages. perror(3C) PRM
 that are available on/ uuglist: print the list of service grades uuglist(1C) URM
 the current host. hostid: print the numeric identifier of hostid(1) BSD
 group name and ID. id: print the user name and ID, and id(1M) SARM
 group name and ID. id: print the user name and ID, and id(1M) URM
 keys from a YP map. ypmatch: print the value of one or more ypmatch(1) NI
 device. prtvtoc: print the VTOC of a block prtvtoc(1M) SARM
 ypcat: print values in a YP data base. ypcat(1) NI
 xtt: extract and print xt driver packet traces. xtt(1M) SARM
 xts: extract and print xt driver statistics. xts(1M) SARM
 file or binary. strings: find printable strings in an object strings(1) URM
 variables currently set. printenv: display environment printenv(1) BSD
 lpr: send a job to the printer. lpr(1) BSD
 lpc: line printer control program. lpc(1M) BSD
 lpq: display the queue of printer jobs. lpq(1) BSD
 lprm: remove jobs from the printer queue. lprm(1) BSD
 disable: enable/disable LP printers. enable, enable(1) URM
 vprintf, fprintf, sprintf, print, print(3S) BSD
 vprintf, fprintf, sprintf: print print(3S) PRM
 formatted output. printf: print formatted output. printf(1) URM
 printing queue priorities. lpusers(1M) SARM
 lpusers: set printw, wprintw, mvprintw, curs_printw(3X) CHAR
 mvwprintw, curs_printw: control priocntl: process scheduler priocntl(1) URM
 control. priocntl: process scheduler priocntl(2) PRM
 control. priocntlset: generalized process priocntlset(2) PRM
 scheduler control. priorities. lpusers(1M) SARM
 lpusers: set printing queue

Permuted Index

nice: run a command at low
for flow control in specified
flush messages for a specified
 nice: change
 process. nice: change
 renice: alter
get/set program scheduling
server for storing public and
server for storing private and
 map:
rmalloc: allocate space from a
rmfree: free space back into a
 rminit: initialize a
 drv_priv: determine driver

/routines for client side remote
routines for server side remote
rpc: library routines for remote
 routines for secure remote
XDR library routines for remote
bcheckrc: system initialization
 /startup, turnacct: shell
qinit: STREAMS queue processing
 exit, _exit: terminate
 fork: create a new
 nice: change priority of a
 psignal: send signal to a
 wait: await completion of
 intr:
 acct: enable or disable
 acctprc, acctprc1, acctprc2:
 acctcom: search and print
execution of an/ sleep: suspend
 alarm: set a
 times: get
 kill: terminate a
codestroy: communicate with a
 init, telinit:
timex: time a command; report
 wakeup: resume suspended
specified number/ delay: delay
 /proc:
 killpg: send signal to a
 signal: send a signal to a
/getppid, getpgid: get process,
 setpgid: set
 setpgrp: set
priority. nice(1) URM
priority band. bcanput: test bcanput(D3DK) DDRM
priority band. flushband: flushband(D3DK) DDRM
priority of a process. nice(3C) BSD
priority of a time-sharing nice(2) PRM
priority of running processes. renice(1M) BSD
priority. /setpriority: getpriority(3) BSD
private keys. keysevr: keysevr(1M) NI
private keys. keysevr: keysevr(1M) NUAG
private memory map structure. map(D4DK) DDRM
private space management map. rmalloc(D3DK) DDRM
private space management map. rmfree(D3DK) DDRM
private space management map. rminit(D3DK) DDRM
privilege. drv_priv(D3DK) DDRM
/proc: process file system. proc(4) SARM
procedure call authentication. rpc_clnt_auth(3N) NI
procedure call errors. /library rpc_svc_err(3N) NI
procedure calls. rpc(3N) NI
procedure calls. /library secure_rpc(3N) NI
procedure calls. /xdr_replmsg: rpc_xdr(3N) NI
procedures. brc, brc(1M) SARM
procedures for accounting. acctsh(1M) SARM
procedures structure. qinit(D4DK) DDRM
process. exit(2) PRM
process. fork(2) PRM
process. nice(3C) BSD
process. psignal(D3D) DDRM
process. wait(1) URM
process a device interrupt. intr(D2D) DDRM
process accounting. acct(2) PRM
process accounting. acctprc(1M) SARM
process accounting file(s). acctcom(1) URM
process activity pending sleep(D3DK) DDRM
process alarm clock. alarm(2) PRM
process and child process times. times(2) PRM
process by default. kill(1) URM
process. /cocheck, coreceive, coproc(1F) CHAR
process control initialization. init(1M) SARM
process data and system/ timex(1) URM
process execution. wakeup(D3DK) DDRM
process execution for a delay(D3DK) DDRM
process file system. proc(4) SARM
process group. killpg(3) BSD
process group. signal(D3D) DDRM
process group, and parent/ getpid(2) PRM
process group ID. setpgid(2) PRM
process group ID. setpgrp(2) PRM

set terminal foreground process group, and parent efficient way. vfork: spawn new lock: lock a SMTP mail queue. smtpsched: priority of a time-sharing kill: send a signal to a /sigsendset: send a signal to a pclose: initiate pipe to/from a /getpgrp, getppid, getpgid: get Remote File Sharing daemon Remote File Sharing daemon administration. dispadmin: priocntl: priocntl: priocntlset: generalized ps: report lock into memory or unlock occurred. pollwakeup: inform a times: get process and child times: get waitid: wait for child waitpid: wait for child wait: wait for child /WIFSIGNALED, WIFEXITED: wait for ptrace: pause: suspend a signal mask and suspend sigsem: signal a killall: kill all active get core images of running to a process or a group of block I/O. biowait: suspend display the status of current after block I/O and wakeup alter priority of running to a process or a group of structure. fuser: identify ypbind: YP server and binder awk: pattern scanning and nawk: pattern scanning and qinit: STREAMS queue mailx: interactive message halt: stop the m4: macro reboot: reboot system or halt subsystem. form_driver: command process group id. tcsetpgrp: tcsetpgrp(3C) PRM process IDs. /get process, getpid(2) PRM process in a virtual memory vfork(2) PRM process in primary memory. lock(2) XNX process messages queued in the smtpsched(1M) SARM process. nice: change nice(2) PRM process or a group of processes. kill(2) PRM process or a group of processes. sigsend(2) PRM process. popen, popen(3S) PRM process, process group, and/ getpid(2) PRM process. rfudaemon: rfudaemon(1M) NUAG process. rfudaemon: rfudaemon(1M) SARM process scheduler dispadmin(1M) SARM process scheduler control. priocntl(1) URM process scheduler control. priocntl(2) PRM process scheduler control. priocntlset(2) PRM process status. ps(1) URM process, text, or data. plock: plock(2) PRM process that an event has pollwakeup(D3DK) DDRM process times. times(2) PRM process times. times(3C) BSD process to change state. waitid(2) PRM process to change state. waitpid(2) PRM process to stop or terminate. wait(2) PRM process to terminate or stop. wait(3) BSD process trace. ptrace(2) PRM process until signal. pause(2) PRM process until signal. /install sigsuspend(2) PRM process waiting on a semaphore. sigsem(2) XNX processes. killall(1M) SARM processes. gcore: gcore(1) URM processes. kill: send a signal kill(2) PRM processes pending completion of biowait(D3DK) DDRM processes. ps: ps(1) BSD processes. /release buffer biodone(D3DK) DDRM processes. renice: renice(1M) BSD processes. /send a signal sigsend(2) PRM processes using a file or file fuser(1M) SARM processes. ypserv, ypserv(1M) NI processing language. awk(1) URM processing language. nawk(1) URM processing procedures structure. qinit(D4DK) DDRM processing system. mailx(1) URM processor. halt(1M) BSD processor. m4(1) PRM processor. reboot(3) BSD processor for the forms form_driver(3X) CHAR

Permuted Index

subsystem. menu_driver: command
 reset: resets the
 parameter string on the
 host. mach: display the
 /u3b5, u3b15, vax, 386, u370: get
 configuration. idconfig:
 pkgmk:
 t_error:
 processor for the menus menu_driver(3X) CHAR
 processor in a given slot. reset(1M) MBRMAN
 processor in. /the bootstrap initbp(1M) MBRMAN
 processor type of the current mach(1) BSD
 processor type truth value. machid(1) URM
 produce a new kernel idconfig(1M) SARM
 produce an installable package. pkgmk(1) SS
 produce error message. t_error(3N) NI
 prof: display profile data. prof(1) PRM
 prof: profile within a function. prof(5) PRM
 profil: execution time profile. profil(2) PRM
 profile. monitor(3C) PRM
 profile. profil(2) PRM
 profile data. prof(1) PRM
 profile data. lprof: display lprof(1) PRM
 profile: setting up an profile(4) SARM
 profile within a function. prof(5) PRM
 profiler. fusage(1M) NUAG
 profiler. fusage(1M) SARM
 profiler. prf(7) SARM
 profiler: prfld, prfstat, prfdc, profiler(1M) SARM
 profiler. /prfstat, prfdc, profiler(1M) SARM
 program. boot(1M) SARM
 program. ftp(1) NUAG
 program. ftp(1) URM
 program. lpc(1M) BSD
 program. raise(3C) PRM
 program. talkd(1M) NUAG
 program. talkd(1M) SARM
 program. tftp(1) NUAG
 program. tftp(1) URM
 program. units(1) URM
 program assertion. assert(3X) PRM
 program beautifier. cb(1) PRM
 program checker. lint(1) PRM
 program. copyout: copy copyout(D3DK) DDRM
 program cross-reference. cxref(1) PRM
 program. cscope: cscope(1) PRM
 program debugger. ctrace(1) PRM
 program. end, end(3C) PRM
 program for the uucp system. uucico(1M) SARM
 program header table. elf_getphdr(3E) PRM
 program message. catgets(3C) PRM
 program number data base. rpc(4) NI
 program number mapper. rpcbind: rpcbind(1M) NI
 program number mapper. rpcbind: rpcbind(1M) NUAG
 program on the communications/ ccifree(1M) MBRMAN

monitor: prepare execution
 profil: execution time
 prof: display
 line-by-line execution count
 environment at login time.
 prof:
 fusage: disk access
 fusage: disk access
 prf: operating system
 prfsnap, prfpr: UNIX system/
 prfsnap, prfpr: UNIX system
 boot: UNIX system boot
 ftp: file transfer
 ftp: file transfer
 lpc: line printer control
 raise: send signal to
 talkd, in.talkd: server for talk
 talkd, in.talkd: server for talk
 tftp: trivial file transfer
 tftp: trivial file transfer
 units: conversion
 assert: verify
 cb: C
 lint: a C
 data from a driver to a user
 cxref: generate C
 interactively examine a C
 ctrace: C
 etext, edata: last locations in
 uucico: file transport
 /retrieve class-dependent
 catgets: read a
 rpc: rpc
 universal addresses to RPC
 universal addresses to RPC
 ccifree: frees a line discipline

ccload: load a line discipline
 /setpriority: get/set
 and remove ifdef'ed lines from C
 STREAMS error logger cleanup
 STREAMS error logger cleanup
 atexit: add
 copyin: copy data from a user
 for the uucp file transport
 hardware request. dma_prog:
 intro: introduction to
 standard/restricted command and
 tasks. lex: generate
 to commands and application
 setlocale: modify and query a
 update, and regenerate groups of
 commands and application
 item. ckitem: build a menu;
 item. ckitem: build a menu;
 pathname. ckpath: display a
 pathname. ckpath: display a
 string answer. ckstr: display a
 string answer. ckstr: display a
 of day. cktime: display a
 of day. cktime: display a
 integer value. ckint: display a
 integer value. ckint: display a
 yes/no. ckyorn:
 yes/no. ckyorn:
 /errdate, helpdate, valdate:
 /errdate, helpdate, valdate:
 ckgid, errgid, helpgid, valgid:
 ckgid, errgid, helpgid, valgid:
 keyword. ckkeywd:
 keyword. ckkeywd:
 ID. ckuid:
 ID. ckuid:
 integer. ckrange:
 integer. ckrange:
 to service media insertion
 yppush: force
 network interfaces. if: general
 mprotect: set
 ARP: Address Resolution
 ARP: Address Resolution
 ICMP: Internet Control Message
 ICMP: Internet Control Message
 ICMP: Internet Control Message
 program on to the communications/
 program scheduling priority. ccload(1M) MBRMAN
 program source. /resolve getpriority(3) BSD
 program. strclean: undef(1) BSD
 program. strclean: strclean(1M) SARM
 program. strclean: strclean(1M) STRM
 program termination routine. atexit(3C) PRM
 program to a driver buffer. copyin(D3DK) DDRM
 program. uusched: the scheduler uusched(1M) SARM
 Programming a Channel for a dma_prog(D3D) DDRM
 programming commands. intro(1) PRM
 programming language. /a ksh(1) URM
 programs for simple lexical lex(1) PRM
 programs. intro: introduction intro(1) URM
 program's locale. setlocale(3C) PRM
 programs. make: maintain, make(1) PRM
 programs. /to maintenance intro(1M) SARM
 prompt for and return a menu ckitem(1) SARM
 prompt for and return a menu ckitem(1) SS
 prompt; verify and return a ckpath(1) SARM
 prompt; verify and return a ckpath(1) SS
 prompt; verify and return a ckstr(1) SARM
 prompt; verify and return a ckstr(1) SS
 prompt; verify and return a time cktime(1) SARM
 prompt; verify and return a time cktime(1) SS
 prompt; verify and return an ckint(1) SARM
 prompt; verify and return an ckint(1) SS
 prompts for and validates ckyorn(1) SARM
 prompts for and validates ckyorn(1) SS
 prompts for and validates a/ ckdate(1) SARM
 prompts for and validates a/ ckdate(1) SS
 prompts for and validates a/ ckgid(1) SARM
 prompts for and validates a/ ckgid(1) SS
 prompts for and validates a ckkeywd(1) SARM
 prompts for and validates a ckkeywd(1) SS
 prompts for and validates a user ckuid(1) SARM
 prompts for and validates a user ckuid(1) SS
 prompts for and validates an ckrange(1) SARM
 prompts for and validates an ckrange(1) SS
 prompts. /with backup operations bkoper(1M) SARM
 propagation of a changed YP map. yppush(1M) NI
 properties of Internet Protocol if(7) NUAG
 protection of memory mapping. mprotect(2) PRM
 Protocol. ARP(7) NUAG
 Protocol. ARP(7) SARM
 Protocol. ICMP(7) NI
 Protocol. ICMP(7) NUAG
 Protocol. ICMP(7) SARM

Permuted Index

IP: Internet Protocol.	IP(7) NI
IP: Internet Protocol.	IP(7) NUAG
IP: Internet Protocol.	IP(7) SARM
UDP: Internet User Datagram Protocol.	UDP(7) NI
UDP: Internet User Datagram Protocol.	UDP(7) NUAG
UDP: Internet User Datagram Protocol.	UDP(7) SARM
rpcgen: an RPC protocol compiler.	rpcgen(1) NI
rci: debug console/rci protocol driver.	rci(1M) MBRMAN
setprotoent, endprotoent: get protocol entry. /getprotobyname,	getprotoent(3N) NI
inet: Internet protocol family.	inet(7) NUAG
inet: Internet protocol family.	inet(7) SARM
protocols: protocol name data base.	protocols(4) NUAG
protocols: protocol name data base.	protocols(4) SARM
/general properties of Internet Protocol network interfaces.	if(7) NUAG
ftpd: file transfer protocol server.	ftpd(1M) NUAG
ftpd: file transfer protocol server.	ftpd(1M) SARM
telnetd: DARPA TELNET protocol server.	telnetd(1M) NUAG
telnetd: DARPA TELNET protocol server.	telnetd(1M) SARM
DARPA Reverse Address Resolution Protocol server. rarpd:	rarpd(1M) NUAG
DARPA Reverse Address Resolution Protocol server. rarpd:	rarpd(1M) SARM
DARPA Trivial File Transfer Protocol server. tftpd:	tftpd(1M) NUAG
DARPA Trivial File Transfer Protocol server. tftpd:	tftpd(1M) SARM
host using Simple Mail Transfer Protocol. /SMTP mail to a remote	smtp(1M) SARM
Internet Transmission Control Protocol. TCP:	TCP(7) NI
Internet Transmission Control Protocol. TCP:	TCP(7) NUAG
Internet Transmission Control Protocol. TCP:	TCP(7) SARM
trpt: transliterate protocol trace.	trpt(1M) NUAG
trpt: transliterate protocol trace.	trpt(1M) SARM
windowing terminal/ layers: protocol used between host and	layers(5) PRM
windowing terminal/ layers: protocol used between host and	layers(5) SARM
xtproto: multiplexed channels protocol used by xt driver.	xtproto(5) SARM
xtproto: multiplexed channels protocol used by xt driver.	xtproto(5) PRM
base. protocols: protocol name data	protocols(4) NUAG
base. protocols: protocol name data	protocols(4) SARM
information. t_getinfo: get protocol-specific service	t_getinfo(3N) NI
pkgproto: generate a prototype file.	pkgproto(1) SS
prototype: package information	prototype(4) SS
provide information for setting	tset(1) XNX
provide labels for file systems.	labelit(1M) SARM
provide labels for s5 file	labelit(1M) SARM
provide labels for ufs file	labelit(1M) SARM
provide truth values.	true(1) URM
provider. nlsprovider:	nlsprovider(3N) NI
providers. ticlts, ticots,	ticlts(7) NI
provides information on an	enetinfo(1M) MBRMAN
prs: print an SCCS file.	prs(1) PRM
prt: display the delta and	prt(1) BSD
commentary history of an SCCS/	

/nulladm, prctmp, prdaily,
 block device.
 current processes.
 module. ptem: STREAMS
 sxt:
 sxt:
 generate uniformly distributed
 /grant access to the slave
 ptsname: get name of the slave
 pair. unlockpt: unlock a
 messages. psignal,
 messages.
 process.
 signal messages.
 Emulation module.
 size in bytes.
 pseudo-terminal device.
 keyserv: server for storing
 keyserv: server for storing
 publickey:
 publickey:
 /getsecretkey: retrieve
 copy. uuto, uupick:
 newkey: create a new key in the
 newkey: create a new key in the
 getsecretkey: retrieve public/
 a message.
 /mvwgetch, ungetch: get (or
 stream. ungetc:
 /configure lists of automatically
 /configure lists of automatically
 putq:
 puts, fputs:
 putc, putchar, fputc, putw:
 read directory entries and
 preceding queue.
 echo:
 head of a queue.
 character or word on a stream.
 character or word on a/ putc,
 to a queue.
 with a one-byte parameter to a/
 prtacct, runacct, shutacct./ acctsh(1M) SARM
 prvtoc: print the VTOC of a prvtoc(1M) SARM
 ps: display the status of ps(1) BSD
 ps: report process status. ps(1) URM
 Pseudo Terminal Emulation ptem(7) STRM
 pseudo-device driver. sxt(7) SARM
 pseudo-device driver. sxt(7) STRM
 pseudo-random numbers. /lcong48: drand48(3C) PRM
 pseudo-terminal device. grantpt(3C) STRM
 pseudo-terminal device. ptsname(3C) STRM
 pseudo-terminal master/slave unlockpt(3C) STRM
 psiginfo: system signal psignal(3C) PRM
 psignal, psiginfo: system signal psignal(3C) PRM
 psignal: send signal to a psignal(D3D) DDRM
 psignal, sys_siglist: system psignal(3) BSD
 ptem: STREAMS Pseudo Terminal ptem(7) STRM
 ptob: convert size in pages to ptob(D3DK) DDRM
 ptrace: process trace. ptrace(2) PRM
 ptsname: get name of the slave ptsname(3C) STRM
 public and private keys. keyserv(1M) NI
 public and private keys. keyserv(1M) NUAG
 public key database. publickey(4) NI
 public key database. publickey(4) NUAG
 public or secret key. publickey(3N) NI
 public UNIX-to-UNIX system file uuto(1C) URM
 publickey database. newkey(1M) NI
 publickey database. newkey(1M) NUAG
 publickey: getpublickey, publickey(3N) NI
 publickey: public key database. publickey(4) NI
 publickey: public key database. publickey(4) NUAG
 pullupmsg: concatenate bytes in pullupmsg(D3DK) DDRM
 push back) characters from/ curs_getch(3X) CHAR
 push character back onto input ungetc(3S) PRM
 pushed STREAMS modules. autopush(1M) SARM
 pushed STREAMS modules. autopush(1M) STRM
 put a message on a queue. putq(D3DK) DDRM
 put a string on a stream. puts(3S) PRM
 put character or word on a/ putc(3S) PRM
 put in a file. getdents: getdents(2) PRM
 put: receive messages from the put(D2DK) DDRM
 put string on virtual output. echo(1F) CHAR
 putbq: place a message at the putbq(D3DK) DDRM
 putc, putchar, fputc, putw: put putc(3S) PRM
 putchar, fputc, putw: put putc(3S) PRM
 putctl: send a control message putctl(D3DK) DDRM
 putctl1: send a control message putctl1(D3DK) DDRM
 putdev: edits device table. putdev(1) SARM

table. putdgrp: edits device group putdgrp(1) SARM
 environment. putenv: change or add value to putenv(3C) PRM
 stream. putmsg: send a message on a putmsg(2) PRM
 stream. putmsg: send a message on a putmsg(2) STRM
 next queue. putnext: send a message to the putnext(D3DK) DDRM
 /restartterm, tparm, tputs, putp, vidputs, vidattr, mvcur, / curs_terminfo(3X) CHAR
 entry. putpwent: write password file putpwent(3C) PRM
 putq: put a message on a queue. putq(D3DK) DDRM
 puts a buffer back into the free ... mps_free_msgbuf(D3D) DDRM
 memory pool. mps_free_msgbuf: puts, fputs: put a string on a puts(3S) PRM
 stream. puts its arguments on FMLI message(1F) CHAR
 message line. message: puts spent: write shadow password putspent(3C) PRM
 file entry. pututline, setutent, endutent, / getut(3C) PRM
 /getutent, getutid, getutline, pututxline, setutxent, / getutx: getutx(3C) PRM
 getutent, getutxid, getutline, putw: put character or word on a putc(3S) PRM
 stream. putc, putchar, fputc, putwin, getwin, delay_output, / curs_util(3X) CHAR
 /keyname, filter, use_env, pwck: check password database pwck(1M) BSD
 entries. pwck, grpck: password/group file pwck(1M) SARM
 checkers. pwconv: Installs and updates. pwconv(1M) SARM
 control information structure. pwd: working directory name. pwd(1) URM
 qband: STREAMS queue flow qband(D4DK) DDRM
 qenable: enable a queue. qenable(D3DK) DDRM
 streamer interface. qt: QIC cartridge magnetic tape qt(7) SARM
 /raw, noraw, noqiflush, qiflush, timeout, wtimeout, / curs_inopts(3X) CHAR
 procedures structure. qinit: STREAMS queue processing qinit(D4DK) DDRM
 stream in the reverse/ qreply: send a message on a qreply(D3DK) DDRM
 messages on a queue. qsize: find the number of qsize(D3DK) DDRM
 streamer interface. qsort: quicker sort. qsort(3C) PRM
 setlocale: modify and qt: QIC cartridge magnetic tape qt(7) SARM
 interactively. nslookup: query a program's locale. setlocale(3C) PRM
 interactively. nslookup: query name servers nslookup(1M) NUAG
 Remote File Sharing name server query name servers nslookup(1M) SARM
 Remote File Sharing name server query. nsquery: nsquery(1M) NUAG
 termname: curses environment query. nsquery: nsquery(1M) SARM
 strchg, strconf: change or query routines. /termatrs, curs_termatrs(3X) CHAR
 strchg, strconf: change or query stream configuration. strchg(1) STRM
 tput: initialize a terminal/ query stream configuration. strchg(1) URM
 flushq: remove messages from a query terminfo database. tput(1) URM
 insq: insert a message into a queue. flushq(D3DK) DDRM
 msgget: get message queue. insq(D3DK) DDRM
 putq: put a message on a queue. msgget(2) PRM
 qenable: enable a queue. putq(D3DK) DDRM
 RD: get pointer to the read queue. qenable(D3DK) DDRM
 rmvq: remove a message from a queue. RD(D3DK) DDRM
 to the queue behind the current queue. rmvq(D3DK) DDRM
 backq: get pointer queue. backq: get pointer backq(D3DK) DDRM
 queue behind the current queue. backq(D3DK) DDRM

test for room in a message
 about a queue or band of the
 structure. **qband**: STREAMS
 enableok: reschedule a
WR: get pointer to the write
 noenable: prevent a
 get the next message from a
 insert/remove element from a
 SAMESTR: test if next
 remove jobs from the printer
 smtpqer:
 lpq: display the
strqget: get information about a
 /change information about a
 get pointer to queue's partner
 lpusers: set printing
 structure. **qinit**: STREAMS
 messages from the preceding
 place a message at the head of a
 send a control message to a
 send a message to the next
 find the number of messages on a
 rmount:
 rmount:
 memory/ **ipcrm**: remove a message
 with a one-byte parameter to a
 messages queued in the SMTP mail

 about a queue or band of the
 queue: STREAMS
smtpsched: process messages
 srv: service
 rumount: cancel
 rumount: cancel
 rmntry: attempt to mount
 rmntry: attempt to mount
 times. **atq**: display the jobs
 OTHERQ: get pointer to
 transmission. **mps_AMPsend_rsvp**:

 qsort:
 a command immune to hangups and
 ownership.
 quota: display a user's disk
 quotacheck: file system
 quota and usage.
 consistency checker.

queue. **canput**: **canput(D3DK)** DDRM
 queue. /change information **strqset(D3DK)** DDRM
 queue flow control information **qband(D4DK)** DDRM
 queue for service. **enableok(D3DK)** DDRM
 queue for this module or driver. **WR(D3DK)** DDRM
 queue from being scheduled. **noenable(D3DK)** DDRM
 queue. **getq**: **getq(D3DK)** DDRM
 queue. **insque**, **remque**: **insque(3C)** PRM
 queue is same type. **SAMESTR(D3DK)** DDRM
 queue. **lprm**: **lprm(1)** BSD
 queue mail for delivery by SMTP. **smtpqer(1M)** SARM
 queue of printer jobs. **lpq(1)** BSD
 queue or band of the queue. **strqget(D3DK)** DDRM
 queue or band of the queue. **strqset(D3DK)** DDRM
 queue. **OTHERQ**: **OTHERQ(D3DK)** DDRM
 queue priorities. **lpusers(1M)** SARM
 queue processing procedures **qinit(D4DK)** DDRM
 queue. **put**: receive **put(D2DK)** DDRM
 queue. **putbq**: **putbq(D3DK)** DDRM
 queue. **putctl**: **putctl(D3DK)** DDRM
 queue. **putnext**: **putnext(D3DK)** DDRM
 queue. **qsize**: **qsize(D3DK)** DDRM
 queue remote resource mounts. **rmount(1M)** NUAG
 queue remote resource mounts. **rmount(1M)** SARM
 queue, semaphore set, or shared **ipcrm(1)** URM
 queue. /send a control message **putctl1(D3DK)** DDRM
 queue. **smtpsched**: process **smtpsched(1M)** SARM
 queue: STREAMS queue structure. **queue(D4DK)** DDRM
 queue. **strqget**: get information **strqget(D3DK)** DDRM
 queue structure. **queue(D4DK)** DDRM
 queued in the SMTP mail queue. **smtpsched(1M)** SARM
 queued messages. **srv(D2DK)** DDRM
 queued remote resource request. **rumount(1M)** NUAG
 queued remote resource request. **rumount(1M)** SARM
 queued remote resources. **rmntry(1M)** NUAG
 queued remote resources. **rmntry(1M)** SARM
 queued to run at specified **atq(1)** URM
 queue's partner queue. **OTHERQ(D3DK)** DDRM
 queues request messages for
 **mps_AMPsend_rsvp(D3D)** DDRM
 quicker sort. **qsort(3C)** PRM
 quits. **nohup**: run **nohup(1)** URM
 quot: summarize file system **quot(1M)** SARM
 quota and usage. **quota(1M)** SARM
 quota consistency checker. **quotacheck(1M)** SARM
 quota: display a user's disk **quota(1M)** SARM
 quotacheck: file system quota **quotacheck(1M)** SARM

quotas on and off. quotaon, system quotas on and off. edquota: edit user repquota: summarize quotaoff: turn file system div, ldiv: compute the

ramd: MULTIBUS cram- CMOS

random-number generator. number generator. medium. disk: elf_rand: number. fsirand: install random: generate a rand, srand: simple /initstate, setstate: better setstate: better random number/ rand, srand: simple Resolution Protocol server. Resolution Protocol server. freerbuf: free a getrbuf: get a physiock: validate and issue /meta, nodelay, notimeout, stop the operating system. multi-user environment. stop and reboot the operating/ driver.

queue. remote host. remote host. data to be read. to buffer. repinsw: to buffer. repinsd: 16-bit I/O port. inw: 32-bit I/O port. inl: port. inb: getpass: catgets: manufacturer's defect/ mdl: buffer. repinsb: read: quotaoff: turn file system quotaon(1M) SARM quotaon, quotaoff: turn file quotaon(1M) SARM quotas. edquota(1M) SARM quotas for a file system. repquota(1M) SARM quotas on and off. quotaon, quotaon(1M) SARM quotient and remainder. div(3C) PRM raise: send signal to program. raise(3C) PRM RAM disk driver. ramd(1M) MBRMAN RAM interface. cram(7) SARM ramd: MULTIBUS RAM disk driver. ramd(1M) MBRMAN rand, srand: simple rand(3C) PRM rand, srand: simple random rand(3C) BSD random access bulk storage disk(7) SARM random archive member access. elf_rand(3E) PRM random: generate a random random(1) XNX random inode generation numbers. fsirand(1) BSD random number. random(1) XNX random number generator. rand(3C) BSD random number generator;/ random(3) BSD random, srand, initstate, random(3) BSD random-number generator. rand(3C) PRM rarpd: DARPA Reverse Address rarpd(1M) NUAG rarpd: DARPA Reverse Address rarpd(1M) SARM raw buffer header. freerbuf(D3DK) DDRM raw buffer header. getrbuf(D3DK) DDRM raw I/O request. physiock(D3D) DDRM raw, noraw, noqiflush, qiflush,/ curs_inopts(3X) CHAR rc0: run commands performed to rc0(1M) SARM rc2: run commands performed for rc2(1M) SARM rc6: run commands performed to rc6(1M) SARM rci: debug console/rci protocol rci(1M) MBRMAN rcp: remote file copy. rcp(1) NUAG rcp: remote file copy. rcp(1) URM RD: get pointer to the read RD(D3DK) DDRM rdate: set system date from a rdate(1M) NUAG rdate: set system date from a rdate(1M) SARM rdchk: check to see if there is rdchk(2) XNX read 16-bit words from I/O port repinsw(D3D) DDRM read 32-bit words from I/O port repinsd(D3D) DDRM read a 16 bit short word from a inw(D3D) DDRM read a 32-bit word from a inl(D3D) DDRM read a byte from a 8-bit I/O inb(D3D) DDRM read a password. getpass(3C) PRM read a program message. catgets(3C) PRM read and optionally write the mdl(1M) MBRMAN read bytes from I/O port to repinsb(D3D) DDRM read data from a device. read(D2DK) DDRM

in a file. <code>getdents:</code>	<code>read directory entries and put</code>	<code>getdents(2)</code> PRM
<code>read:</code>	<code>read from file.</code>	<code>read(2)</code> PRM
<code>mail, rmail:</code>	<code>read mail or send mail to users.</code>	<code>mail(1)</code> URM
<code>line:</code>	<code>read one line.</code>	<code>line(1)</code> URM
RD: get pointer to the to see if there is data to be	<code>read queue.</code>	<code>RD(D3DK)</code> DDRM
	<code>read. rdchk: check</code>	<code>rdchk(2)</code> XNX
	<code>read: read data from a device.</code>	<code>read(D2DK)</code> DDRM
	<code>read: read from file.</code>	<code>read(2)</code> PRM
<code>delimiter. bgets:</code>	<code>read stream up to next</code>	<code>bgets(3G)</code> PRM
<code>link. readlink:</code>	<code>read the value of a symbolic</code>	<code>readlink(2)</code> PRM
<code>/scr_restore, scr_init, scr_set:</code>	<code>read (write) a curses screen/</code>	<code>scr_dump(3X)</code> CHAR
<code>rewinddir,/ directory: opendir,</code>	<code>readdir, telldir, seekdir,</code>	<code>directory(3C)</code> PRM
<code>gets longest line.</code>	<code>readfile, longline: reads file,</code>	<code>readfile(1F)</code> CHAR
<code>open: open for</code>	<code>reading or writing.</code>	<code>open(2)</code> PRM
<code>lock or unlock a file region for</code>	<code>reading or writing. locking:</code>	<code>locking(2)</code> XNX
<code>symbolic link.</code>	<code>readlink: read the value of a</code>	<code>readlink(2)</code> PRM
<code>readfile, longline:</code>	<code>reads file, gets longest line.</code>	<code>readfile(1F)</code> CHAR
<code>specifications. idmkinit:</code>	<code>reads files containing</code>	<code>idmkinit(1M)</code> SARM
the board and returns/ <code>ics_read:</code>	<code>reads interconnect registers of</code>	<code>ics_read(1M)</code> MBRMAN
<code>number of/ ics_rdwr:</code>	<code>reads or writes a specified</code>	<code>ics_rdwr(D3D)</code> DDRM
<code>idmnod: removes nodes and</code>	<code>reads specifications of nodes.</code>	<code>idmnod(1M)</code> SARM
<code>of the board in/ ics_find_rec:</code>	<code>reads the interconnect register</code>	<code>ics_find_rec(D3D)</code> DDRM
<code>of the board in the. ics_read:</code>	<code>reads the interconnect register</code>	<code>ics_read(D3D)</code> DDRM
<code>lseek: move</code>	<code>read/write file pointer.</code>	<code>lseek(2)</code> PRM
<code>tirdwr: Transport Interface</code>	<code>read/write interface STREAMS/</code>	<code>tirdwr(7)</code> STRM
<code>setregid: set</code>	<code>real and effective group IDs.</code>	<code>setregid(3)</code> BSD
<code>setreuid: set</code>	<code>real and effective user IDs.</code>	<code>setreuid(3)</code> BSD
<code>realpath: returns the</code>	<code>real file name.</code>	<code>realpath(3C)</code> PRM
<code>/get real user, effective user,</code>	<code>real group, and effective group/</code>	<code>getuid(2)</code> PRM
<code>rtc:</code>	<code>real time clock interface.</code>	<code>rtc(7)</code> SARM
<code>/geteuid, getgid, getegid: get</code>	<code>real user, effective user, real/</code>	<code>getuid(2)</code> PRM
<code>mallinfo: memory/ malloc, free,</code>	<code>realloc, calloc, mallopt,</code>	<code>malloc(3X)</code> PRM
<code>valloc; memory/ malloc, free,</code>	<code>realloc, calloc, memalign,</code>	<code>malloc(3C)</code> PRM
<code>name.</code>	<code>realpath: returns the real file</code>	<code>realpath(3C)</code> PRM
<code>table. rt_dptbl:</code>	<code>real-time dispatcher parameter</code>	<code>rt_dptbl(4)</code> SARM
<code>processor.</code>	<code>reboot: reboot system or halt</code>	<code>reboot(3)</code> BSD
<code>system.</code>	<code>reboot: restart the operating</code>	<code>reboot(1M)</code> BSD
<code>reboot:</code>	<code>reboot system or halt processor.</code>	<code>reboot(3)</code> BSD
/commands performed to stop and checking/ <code>fastboot, fasthalt:</code>	<code>reboot the operating system.</code>	<code>rc6(1M)</code> SARM
<code>mail aliases file. newaliases:</code>	<code>reboot/halt the system without</code>	<code>fastboot(1M)</code> BSD
<code>ypmake:</code>	<code>rebuild the data base for the</code>	<code>newaliases(1M)</code> BSD
register of the board/ <code>ics_find</code>	<code>rebuild YP database.</code>	<code>ypmake(1M)</code> NI
<code>t_rcvrel: acknowledge</code>	<code>_rec: reads the interconnect</code>	<code>ics_find_rec(D3D)</code> DDRM
<code>t_rcvudata:</code>	<code>receipt of an orderly release/</code>	<code>t_rcvrel(3N)</code> NI
<code>recv, recvfrom, recvmsg:</code>	<code>receive a data unit.</code>	<code>t_rcvudata(3N)</code> NI
<code>indication. t_rcvuderr:</code>	<code>receive a message from a socket.</code>	<code>recv(3N)</code> NI
	<code>receive a unit data error</code>	<code>t_rcvuderr(3N)</code> NI

Permuted Index

sent over a connection. `t_rcv`: receive data or expedited data `t_rcv(3N)` NI
`smtpd`: receive incoming SMTP messages. `smtpd(1M)` SARM
preceding queue. `put`: receive messages from the `put(D2DK)` DDRM
`fromsmtp`: receive RFC822 mail from SMTP. `fromsmtp(1M)` SARM
connect request. `t_rcvconnect`: receive the confirmation from a `t_rcvconnect(3N)` NI
`mps_AMPsend_reply`: replies to a received request that is part of/
..... `mps_AMPsend_reply(D3D)` DDRM
request/ `mb2a_getreqfrag`: Receives a data fragment of a `mb2a_getreqfrag(3tai)` TAI
request/ `mb2s_getreqfrag`: Receives a data fragment of a `mb2s_getreqfrag(3tai)` TAI
/Sends a request message and receives a response.. `mb2s_sendrspv(3tai)` TAI
messages in. `mb2a_receive`: Receives incoming MULTIBUS II `mb2a_receive(3tai)` TAI
messages in. `mb2s_receive`: Receives incoming MULTIBUS II `mb2s_receive(3tai)` TAI
fragments/ `mps_AMPreceive_frag`: receives solicited data in `mps_AMPreceive_frag(D3D)` DDRM
corresponds to/ `mps_AMPreceive`: receives solicited data that `mps_AMPreceive(D3D)` DDRM
mail. `mail_pipe`: invoke recipient command for incoming `mail_pipe(1M)` SARM
expression handler. `regex`, `re_comp`, `re_exec`: regular `regex(3)` BSD
floating-point value to decimal record. /convert `floating_to_decimal(3)` BSD
`ics_find_rec`: finds a specific record in the interconnect space/ `ics_find_rec(1M)` MBRMAN
the host id field of the HOST ID record in this. /returns `ics_hostid(D3D)` DDRM
`lockf`: record locking on files. `lockf(3C)` PRM
/convert decimal record to floating-point value. `decimal_to_floating(3)` BSD
from per-process accounting records. /command summary `acctcms(1M)` SARM
manipulate connect accounting records. `fwtmp`, `wtmpfix`: `fwtmp(1M)` SARM
a message from a socket. `recv`, `recvfrom`, `recvmsg`: receive `recv(3N)` NI
message from a socket. `recv`, `recvfrom`, `recvmsg`: receive a `recv(3N)` NI
a socket. `recv`, `recvfrom`, `recvmsg`: receive a message from `recv(3N)` NI
`ed`, `red`: text editor. `ed(1)` URM
`setcolor`: redefine or create a color. `setcolor(1F)` CHAR
`curses/ /wnoutrefresh`, `doupdate`, `redrawwin`, `wredrawln`: refresh `curl_refresh(3X)` CHAR
handler. `regex`, `re_comp`, `re_exec`: regular expression `regex(3)` BSD
references from a bibliographic/ `refer`: expand and insert `refer(1)` BSD
`man`: macros to format Reference Manual pages. `man(7)` BSD
reference pages by/ `man`: display reference manual pages; find `man(1)` BSD
/reference manual pages; find reference pages by keyword. `man(1)` BSD
`refer`: expand and insert references from a bibliographic/ `refer(1)` BSD
database. `lookbib`: find references in a bibliographic `lookbib(1)` BSD
/is_wintouched: `curses` refresh control routines. `curl_touch(3X)` CHAR
/doupdate, `redrawwin`, `wredrawln`: refresh `curses` windows and/ `curl_refresh(3X)` CHAR
panels virtual screen refresh routine. /update panels: `panel_update(3X)` CHAR
`doupdate`,/ `curl_refresh`: refresh, `wrefresh`, `wnoutrefresh`, `curl_refresh(3X)` CHAR
execute regular expression. `regcmp`, `regex`: compile and `regcmp(3G)` PRM
`regcmp`: regular expression `regcmp(1)` PRM
make: maintain, update, and regenerate groups of programs. `make(1)` PRM
regular expression. `regcmp`, `regex`: compile and execute `regcmp(3G)` PRM
string. `regex`: match patterns against a `regex(1F)` CHAR
expression handler. `regex`, `re_comp`, `re_exec`: regular `regex(3)` BSD
regular expression compile and/ `regexp`: compile, step, advance: `regexp(5)` PRM

regular expression compile and/
regular expression compile and/
locking: lock or unlock a file
display the contents of a backup
/a value into the specified
/_rec: reads the interconnect
ics_read: reads the interconnect
print service. lpsystem:
/library routines for
system/ fstypes: file that
system/ fstypes: file that
/writes into interconnect
ics_read: reads interconnect
regcmp:
regular expression compile, step, advance: regexp(5) SARM
regular expression compile, step, advance: regexpr(3G) PRM
region for reading or writing. locking(2) XNX
register. bkreg: change or bkreg(1M) SARM
register of the board in. ics_write(D3D) DDRM
register of the board in the. ics_find_rec(D3D) DDRM
register of the board in the. ics_read(D3D) DDRM
register remote systems with the lpsystem(1M) SARM
registering servers. rpc_svc_calls(3N) NI
registers distributed file fstypes(4) NUAG
registers distributed file fstypes(4) SARM
registers of the board. ics_write(1M) MBRMAN
registers of the board and/ ics_read(1M) MBRMAN
regular expression compile. regcmp(1) PRM
regular expression compile and/ regexp(5) PRM
regular expression compile and/ regexpr(5) SARM
regular expression compile and/ regexpr(3G) PRM
regular expression handler. regex(3) BSD
regular expression. regcmp, regcmp(3G) PRM
regular expressions. /search egrep(1) URM
regular expressions. /search egrep(1) XNX
reinit: runs an initialization reinit(1F) CHAR
reject: accept or reject print accept(1M) SARM
reject in response to a buffer. mps_mk_breject(D3D) DDRM
reject lines common to two comm(1) URM
reject print requests. accept(1M) SARM
relation for an object library. lorder(1) PRM
relational database operator. join(1) URM
release. t_sndrel(3N) NI
Release 3.0.. edlina: External edlina(1M) MBRMAN
release a previously allocated sdi_freeblk(D3I) SCSI
release blocked signals and wait sigpause(3) BSD
release buffer after block I/O biodone(D3DK) DDRM
release devices from exclusive devfree(1M) SARM
release indication. /acknowledge t_rcvrel(3N) NI
release the channel. dma_stop(D3D) DDRM
relinquish access to a device. close(D2DK) DDRM
reloads the ethernet controller. enetload(1M) MBRMAN
relogin: rename login entry to relogin(1M) SARM
relogin: rename login entry to relogin(1M) URM
remainder, absolute value/ floor(3M) PRM
remainder. div, div(3C) PRM
remainder: floor, ceiling,/ floor(3M) PRM
reminder service. calendar(1) URM
remote command. rexec(3N) NI
remote command. rexec(3N) NUAG
remote command requests. uuxqt(1M) SARM

rexecd:	remote execution server.	rexecd(1M) NUAG
rexecd:	remote execution server.	rexecd(1M) SARM
rcp:	remote file copy.	rcp(1) NUAG
rcp:	remote file copy.	rcp(1) URM
rfstart:	Remote File Sharing.	rfstart(1M) NUAG
rfstart:	Remote File Sharing.	rfstart(1M) SARM
process. rfudaemon:	Remote File Sharing daemon	rfudaemon(1M) NUAG
process. rfudaemon:	Remote File Sharing daemon	rfudaemon(1M) SARM
administration. rfadmin:	Remote File Sharing domain	rfadmin(1M) NUAG
administration. rfadmin:	Remote File Sharing domain	rfadmin(1M) SARM
network names. dname: print	Remote File Sharing domain and	dname(1M) NUAG
network names. dname: print	Remote File Sharing domain and	dname(1M) SARM
rfstop: stop the	Remote File Sharing environment.	rfstop(1M) NUAG
rfstop: stop the	Remote File Sharing environment.	rfstop(1M) SARM
password. rfpasswd: change	Remote File Sharing host	rfpasswd(1M) NUAG
password. rfpasswd: change	Remote File Sharing host	rfpasswd(1M) SARM
query. nsquery:	Remote File Sharing name server	nsquery(1M) NUAG
query. nsquery:	Remote File Sharing name server	nsquery(1M) SARM
master file. rfmaster:	Remote File Sharing name server	rfmaster(4) NUAG
master file. rfmaster:	Remote File Sharing name server	rfmaster(4) SARM
shell script. rfuadmin:	Remote File Sharing notification	rfuadmin(1M) NUAG
shell script. rfuadmin:	Remote File Sharing notification	rfuadmin(1M) SARM
/rumountall: mount, unmount	Remote File Sharing resources.	rmountall(1M) NUAG
/rumountall: mount, unmount	Remote File Sharing resources.	rmountall(1M) SARM
group mapping. idload:	Remote File Sharing user and	idload(1M) NUAG
group mapping. idload:	Remote File Sharing user and	idload(1M) SARM
rdate: set system date from a	remote host.	rdate(1M) NUAG
rdate: set system date from a	remote host.	rdate(1M) SARM
smtp: send SMTP mail to a	remote host using Simple Mail/	smtp(1M) SARM
rlogin:	remote login.	rlogin(1) NUAG
rlogin:	remote login.	rlogin(1) URM
netrc: file for ftp	remote login data.	netrc(4) NUAG
netrc: file for ftp	remote login data.	netrc(4) SARM
rlogind:	remote login server.	rlogind(1M) NUAG
rlogind:	remote login server.	rlogind(1M) SARM
rwall: write to specified	remote machines.	rwall(3N) NI
information about users on	remote machines. rusers: return	rusers(3N) NI
mount: mount	remote NFS resources.	mount(1M) NUAG
mount: mount	remote NFS resources.	mount(1M) SARM
/list available resources from	remote or local systems.	dfshares(1M) NUAG
/list available resources from	remote or local systems.	dfshares(1M) SARM
/library routines for client side	remote procedure call/	rpc_clnt_auth(3N) NI
/library routines for server side	remote procedure call errors.	rpc_svc_err(3N) NI
rpc: library routines for	remote procedure calls.	rpc(3N) NI
/XDR library routines for	remote procedure calls.	rpc_xdr(3N) NI
/library routines for secure	remote procedure calls.	secure_rpc(3N) NI
rmount: queue	remote resource mounts.	rmount(1M) NUAG

rmount: queue
 rumount: cancel queued
 rmount: cancel queued
 mount: mount
 mount: mount
 rmnttry: attempt to mount queued
 rmnttry: attempt to mount queued
 or unmount file systems and
 rsh:
 rsh:
 rshd:
 rshd:
 ckbinarsys: determine whether
 the ckbinarsys/ binarsys:
 telnet: user interface to a
 telnet: user interface to a
 Utry: try to contact
 available NFS resources from
 available NFS resources from
 available for mounting by
 available for mounting by
 unavailable for mounting by
 unavailable for mounting by
 available for mounting by
 available for mounting by
 unavailable for mounting by
 unavailable for mounting by
 available for mounting by
 available for mounting by
 unavailable for mounting by
 unavailable for mounting by
 service. lpsystem: register
 ct: spawn login to a
 fingerd, in.fingerd:
 fingerd, in.fingerd:
 information about local and
 information about local and
 sysadm interface menu or task
 sysadm interface menu or task
 structure. uwritec:
 file. rmdel:
 rmdir:
 database. removef:
 message. rmvb:
 head of a message. unlnkb:
 rmvq:
 semaphore set, or shared/ ipcrm:
 remote resource mounts. rmount(1M) SARM
 remote resource request. rumount(1M) NUAG
 remote resource request. rumount(1M) SARM
 remote resources. mount(1M) NUAG
 remote resources. mount(1M) SARM
 remote resources. rmnttry(1M) NUAG
 remote resources. rmnttry(1M) SARM
 remote resources. /umount: mount mount(1M) SARM
 remote shell. rsh(1) NUAG
 remote shell. rsh(1) URM
 remote shell server. rshd(1M) NUAG
 remote shell server. rshd(1M) SARM
 remote system can accept binary/ ckbinarsys(1M) SARM
 remote system information for binarsys(4) SARM
 remote system using the. telnet(1) NUAG
 remote system using the. telnet(1) URM
 remote system with debugging on. Utry(1M) SARM
 remote systems. dfshares: list dfshares(1M) NUAG
 remote systems. dfshares: list dfshares(1M) SARM
 remote systems. /local resource share(1M) NUAG
 remote systems. /local resource share(1M) SARM
 remote systems. /local resource unshare(1M) NUAG
 remote systems. /local resource unshare(1M) SARM
 remote systems. /NFS resource share(1M) NUAG
 remote systems. /NFS resource share(1M) SARM
 remote systems. /NFS resource unshare(1M) NUAG
 remote systems. /NFS resource unshare(1M) SARM
 remote systems. /RFS resource share(1M) NUAG
 remote systems. /RFS resource share(1M) SARM
 remote systems. /RFS resource unshare(1M) NUAG
 remote systems. /RFS resource unshare(1M) SARM
 remote systems with the print lpsystem(1M) SARM
 remote terminal. ct(1C) URM
 remote user information server. fingerd(1M) NUAG
 remote user information server. fingerd(1M) SARM
 remote users. finger: display finger(1) NUAG
 remote users. finger: display finger(1) URM
 removal tool. delsysadm: delsysadm(1M) SARM
 removal tool. delsysadm: delsysadm(1M) SS
 remove a character from a uio uwritec(D3DK) DDRM
 remove a delta from an SCCS rmdel(1) PRM
 remove a directory. rmdir(2) PRM
 remove a file from software removef(1M) SS
 remove a message block from a rmvb(D3DK) DDRM
 remove a message block from the unlnkb(D3DK) DDRM
 remove a message from a queue. rmvq(D3DK) DDRM
 remove a message queue, ipcrm(1) URM

makedirs, rmdir: create, remove directories in a path. makedirs(3C) PRM
 unlink: remove directory entry. unlink(2) PRM
 remove: remove file. remove(3C) PRM
 rm, rmdir: remove files or directories. rm(1) URM
 program/ _unifdef: resolve and remove ifdef'ed lines from C unifdef(1) BSD
 queue. lprm: remove jobs from the printer lprm(1) BSD
 batch. atrm: remove jobs spooled by at or atrm(1) URM
 flushq: remove messages from a queue. flushq(D3DK) DDRM
 constructs. deroff: remove nroff, troff, tbl and eqn deroff(1) BSD
 constructs. deroff: remove nroff/troff, tbl, and eqn deroff(1) URM
 remove: remove file. remove(3C) PRM
 software database. removef: remove a file from removef(1M) SS
 system. pkgrm: removes a package from the pkgrm(1M) SARM
 system. pkgrm: removes a package from the pkgrm(1M) SS
 specifications of/ idmknod: removes nodes and reads idmknod(1M) SARM
 from a queue. insque, remque: insert/remove element insque(3C) PRM
 file. rename: change the name of a rename(2) PRM
 current layer. relogin: rename login entry to show relogin(1M) SARM
 current layer. relogin: rename login entry to show relogin(1M) URM
 running processes. renice: alter priority of renice(1M) BSD
 fsck (bfs): check and repair bfs file systems. fsck(1M) SARM
 check and interactive repair. /file system consistency fsck(1M) SARM
 fsck: check and repair file systems. fsck(1M) SARM
 fsck (s5): check and repair s5 file systems. fsck(1M) SARM
 xfck: check and repair XENIX filesystems. xfck(1M) SARM
 uniq: report repeated lines in a file. uniq(1) URM
 yes: print string repeatedly. yes(1) XNX
 port to buffer. repinsb: read bytes from I/O repinsb(D3D) DDRM
 I/O port to buffer. repinsd: read 32-bit words from repinsd(D3D) DDRM
 I/O port to buffer. repinsw: read 16-bit words from repinsw(D3D) DDRM
 panel_window: panel_window, replace_panel: get or set the/ panel_window(3X) CHAR
 that is part/ mps_AMPsend_reply: replies to a received request
 mps_AMPsend_reply(D3D) DDRM
 mb2a_sendreply: Sends a reply message asynchronously.. mb2a_sendreply(3tai) TAI
 mode.. mb2s_sendreply: Sends a reply message in synchronous mb2s_sendreply(3tai) TAI
 /constructs a unsolicited reply message to be sent.. mps_mk_unsolrply(D3D) DDRM
 get data length for a solicited reply.. mps_get_reply_len: mps_get_reply_len(D3D) DDRM
 vacation: reply to mail automatically. vacation(1) BSD
 clock: report CPU time used. clock(3C) PRM
 systems. df: report free disk space on file df(1) BSD
 file systems. df (ufs): report free disk space on ufs df(1M) SARM
 communication facilities/ ipc: report inter-process ipc(1) URM
 blocks and files. df (generic): report number of free disk df(1M) SARM
 blocks and files. df (generic): report number of free disk df(1M) URM
 operations. bkhistory: report on completed backup bkhistory(1M) SARM
 sa1, sa2, sadc: system activity report package. sar: sar(1M) SARM
 nroff and troff input files; report possible errors. /check checknr(1) BSD

timex: time a command; report process data and system/ timex(1) URM
 ps: report process status. ps(1) URM
 uniq: report repeated lines in a file. uniq(1) URM
 rpcinfo: report RPC information. rpcinfo(1M) NI
 rpcinfo: report RPC information. rpcinfo(1M) NUAG
 of a file path name. dirname: report the parent directory name dirname(3G) PRM
 sar: system activity reporter. sar(1) URM
 stream. fseek, rewind, ftell: reposition a file pointer in a fseek(3S) PRM
 stream. fsetpos, fgetpos: reposition a file pointer in a fsetpos(3C) PRM
 buffer to an I/O port. repoutsb: write bytes from repoutsb(D3D) DDRM
 from buffer to an I/O port. repoutsd: write 32-bit words repoutsd(D3D) DDRM
 from buffer to an I/O port. repoutsw: write 16-bit words repoutsw(D3D) DDRM
 file system. repquota: summarize quotas for a repquota(1M) SARM
 binary file, or decode its ASCII representation. /encode a uuencode(1C) URM
 routines for external data representation. /library xdr_admin(3N) NI
 routines for external data representation. /library xdr_complex(3N) NI
 routines for external data representation. /library xdr_simple(3N) NI
 /routines for external data representation stream creation. xdr_create(3N) NI
 routines for external data representation. xdr: library xdr(3N) NI
 t_accept: accept a connect request. t_accept(3N) NI
 t_listen: listen for a connect request. t_listen(3N) NI
 call-when-buffer-available request. /cancel a pending unbufcall(D3DK) DDRM
 grant in response to a buffer request.. /construct a buffer mps_mk_bgrant(D3D) DDRM
 a Channel for a hardware request. dma_prog: Programming dma_prog(D3D) DDRM
 Setting a channel for software request. dma_swsetup: dma_swsetup(D3D) DDRM
 dma_pageio: break up an I/O request into manageable units. dma_pageio(D3DK) DDRM
 /Receives a data fragment of a request message. mb2a_getreqfrag(3tai) TAI
 mb2a_sendrspv: Sends a rspv request message.. mb2a_sendrspv(3tai) TAI
 /Receives a data fragment of a request message.. mb2s_getreqfrag(3tai) TAI
 mb2s_sendrspv: Sends a request message and receives a/ mb2s_sendrspv(3tai) TAI
 format and send listener service request message. nlsrequest: nlsrequest(3N) NI
 mps_AMPsend_rspv: queues request messages for/ mps_AMPsend_rspv(D3D) DDRM
 validate and issue raw I/O request. physiock: physiock(D3D) DDRM
 cancel queued remote resource request. rumount: rumount(1M) NUAG
 request. rumount: rumount(1M) SARM
 pkgask: stores answers to a request script. pkgask(1M) SARM
 pkgask: stores answers to a request script. pkgask(1M) SS
 mountd: NFS mount request server. mountd(1M) NUAG
 mountd: NFS mount request server. mountd(1M) SARM
 uio: scatter/gather I/O request structure. uio(D4DK) DDRM
 /replies to a received request that is part of a. mps_AMPsend_reply(D3D) DDRM
 the confirmation from a connect request. t_rcvconnect: receive t_rcvconnect(3N) NI
 send user-initiated disconnect request. t_snddis: t_snddis(3N) NI
 sdi_icmd: perform requested operation immediately. sdi_icmd(D3I) SCSI
 uuxqt: execute remote command requests. uuxqt(1M) SARM
 reject: accept or reject print requests. accept, accept(1M) SARM
 /resolve and eliminate .so requests from nroff or troff/ soelim(1) BSD

the LP print service and move requests. /lpmove: start/stop Ipsched(1M) SARM
 channel. dma_disable: Disable requests on a DMA Controller dma_disable(D3D) DDRM
 channel. dma_enable: Enable requests on a DMA Controller dma_enable(D3D) DDRM
 lp, cancel: send/cancel requests to an LP print service. lp(1) URM
 space: disk space requirement file. space(4) SS
 enableok: reschedule a queue for service. enableok(D3DK) DDRM
 use. devreserv: reserves devices for exclusive devreserv(1M) SARM
 terminal characteristics. tset, reset: establish or restore tset(1) BSD
 terminal. jterm: reset layer of windowing jterm(1) URM
 field to its default values. reset: reset the current form reset(1F) CHAR
 given slot. reset: resets the processor in a reset(1M) MBRMAN
 its default values. reset: reset the current form field to reset(1F) CHAR
 def_prog_mode, def_shell_mode, reset_prog_mode,/ curs kernel: curs_kernel(3X) CHAR
 controller. enetload: resets and reloads the ethernet enetload(1M) MBRMAN
 slot. reset: resets the processor in a given reset(1M) MBRMAN
 /def_shell_mode, reset_prog_mode, reset_shell_mode, resetty,/ curs_kernel(3X) CHAR
 setsyx,/ /reset_shell_mode, resetty, savetty, getsyx, curs_kernel(3X) CHAR
 mincore: determine residency of memory pages. mincore(2) PRM
 resolver, res_mkquery, res_send, res_init, dn_comp, dn_expand:/ resolver(3N) NI
 resolver, res_mkquery, res_send, res_init, dn_comp, dn_expand:/ resolver, res_send, res_init, resolver(3N) NUAG
 dn_comp, dn_expand:/ resolver, res_mkquery, res_send, res_init, resolver(3N) NI
 arp: address resolution display and control. arp(1M) NUAG
 arp: address resolution display and control. arp(1M) SARM
 ARP: Address Resolution Protocol. ARP(7) NUAG
 ARP: Address Resolution Protocol. ARP(7) SARM
 rarpd: DARPA Reverse Address Resolution Protocol server. rarpd(1M) NUAG
 rarpd: DARPA Reverse Address Resolution Protocol server. rarpd(1M) SARM
 for name server routines. resolv.conf: configuration file resolv.conf(4) NUAG
 for name server routines. resolv.conf: configuration file resolv.conf(4) SARM
 requests from nroff or/ soelim: resolve and eliminate .so soelim(1) BSD
 lines from C program/ unifdef: resolve and remove ifdef'ed unifdef(1) BSD
 res_init, dn_comp, dn_expand:/ resolver, res_mkquery, res_send, resolver(3N) NI
 res_init, dn_comp, dn_expand:/ resolver, res_mkquery, res_send, resolver(3N) NUAG
 res_init, dn_comp, dn_expand:/ resolver routines. /res_send, resolver(3N) NI
 res_init, dn_comp, dn_expand:/ resolver routines. /res_send, resolver(3N) NUAG
 by remote/ share: make local resource available for mounting share(1M) NUAG
 by remote/ share: make local resource available for mounting share(1M) SARM
 /control maximum system resource consumption. getrlimit(2) PRM
 forced unmount of an advertised resource. fumount: fumount(1M) NUAG
 forced unmount of an advertised resource. fumount: fumount(1M) SARM
 /await and check access to a resource governed by a/ waitsem(2) XNX
 dfmounts: display mounted resource information. dfmounts(1M) NUAG
 dfmounts: display mounted resource information. dfmounts(1M) SARM
 rmntstat: display mounted resource information. rmntstat(1M) NUAG
 rmntstat: display mounted resource information. rmntstat(1M) SARM
 rmount: queue remote resource mounts. rmount(1M) NUAG

rmount: queue remote
 rumount: cancel queued remote
 rmount: cancel queued remote
 mounting by/ unshare: make local
 mounting by/ unshare: make local
 getrusage: get information about
 mount: mount remote NFS
 mount: mount remote NFS
 containing commands for sharing
 containing commands for sharing
 dfshares: list available
 dfshares: list available
 dfshares: list available RFS
 dfshares: list available RFS
 attempt to mount queued remote
 attempt to mount queued remote
 unmount Remote File Sharing
 unmount Remote File Sharing
 unmount file systems and remote
 share, unshare multiple
 share, unshare multiple
 vacation: automatically
 a request message and receives a
 /construct a buffer reject in
 /construct a buffer grant in
 resolver, res_mkquery,
 resolver, res_mkquery,
 reboot:
 /set_curterm, del_curterm,
 filesystem. fimage: create,
 filesystem/ incfile: create,
 directory. restore:
 archive. fdp: create, or
 archive. ffile: create, or
 filesystems, data partitions,
 original directory.
 tset, reset: establish or
 incremental file system
 XENIX incremental filesystem
 partitions,/ restore: initiate
 /the standard, job control, and
 execution. wakeup:
 gettxt:
 message data base. gettxt:
 elf_getarhdr:
 elf_getarsym:
 /elf32_getehdr, elf32_newehdr:
 resource mounts. rmount(1M) SARM
 resource request. rumount(1M) NUAG
 resource request. rumount(1M) SARM
 resource unavailable for unshare(1M) NUAG
 resource unavailable for unshare(1M) SARM
 resource utilization. getrusage(3) BSD
 resources. mount(1M) NUAG
 resources. mount(1M) SARM
 resources. dfstab: file dfstab(4) NUAG
 resources. dfstab: file dfstab(4) SARM
 resources from remote or local/ dfshares(1M) NUAG
 resources from remote or local/ dfshares(1M) SARM
 resources from remote systems. dfshares(1M) NUAG
 resources from remote systems. dfshares(1M) SARM
 resources. rmntry: rmntry(1M) NUAG
 resources. rmntry: rmntry(1M) SARM
 resources. /rumountall: mount, rmountall(1M) NUAG
 resources. /rumountall: mount, rmountall(1M) SARM
 resources. /umount: mount or mount(1M) SARM
 resources. /unshareall: shareall(1M) NUAG
 resources. /unshareall: shareall(1M) SARM
 respond to incoming mail/ vacation(1) URM
 response.. mb2s_sendrsvp: Sends mb2s_sendrsvp(3tai) TAI
 response to a buffer. mps_mk_breject(D3D) DDRM
 response to a buffer request.. mps_mk_bgrant(D3D) DDRM
 res_send, res_init, dn_comp,/ resolver(3N) NI
 res_send, res_init, dn_comp,/ resolver(3N) NUAG
 restart the operating system. reboot(1M) BSD
 restartterm, tparm, tputs, putp,/ curs_terminfo(3X) CHAR
 restore an image archive of a fimage(1M) SARM
 restore an incremental incfile(1M) SARM
 restore file to original restore(1) XNX
 restore from, a full file system fdp(1M) SARM
 restore from, a full file system ffile(1M) SARM
 restore: initiate restores of restore(1M) SARM
 restore: restore file to restore(1) XNX
 restore terminal/ tset(1) BSD
 restore. ufsrestore: ufsrestore(1M) SARM
 restorer. /xrestor: invoke xrestore(1M) SARM
 restores of filesystems, data restore(1M) SARM
 restricted command interpreter. sh(1) URM
 resume suspended process wakeup(D3DK) DDRM
 retrieve a text string. gettxt(3C) PRM
 retrieve a text string from a gettxt(1) URM
 retrieve archive member header. elf_getarhdr(3E) PRM
 retrieve archive symbol table. elf_getarsym(3E) PRM
 retrieve class-dependent object/ elf_getehdr(3E) PRM

/elf32_getphdr, elf32_newphdr: retrieve class-dependent program/ elf_getphdr(3E) PRM
 elf_getshdr: elf32_getshdr: retrieve class-dependent section/ elf_getshdr(3E) PRM
 data. elf_getident: retrieve file identification elf_getident(3E) PRM
 disconnect. t_rcvdis: retrieve information from t_rcvdis(3N) NI
 information. drv_getparm: retrieve kernel state drv_getparm(D3DK) DDRM
 /getpublickey, getsecretkey: retrieve public or secret key. publickey(3N) NI
 contents. elf_rawfile: retrieve uninterpreted file elf_rawfile(3E) PRM
 marked menu items. getitems: return a list of currently getitems(1F) CHAR
 build a menu; prompt for and return a menu item. ckitem: ckitem(1) SARM
 build a menu; prompt for and return a menu item. ckitem: ckitem(1) SS
 display a prompt; verify and return a pathname. ckpath: ckpath(1) SARM
 display a prompt; verify and return a pathname. ckpath: ckpath(1) SS
 display a prompt; verify and return a string answer. ckstr: ckstr(1) SARM
 display a prompt; verify and return a string answer. ckstr: ckstr(1) SS
 display a prompt; verify and return a time of day. cktime: cktime(1) SARM
 display a prompt; verify and return a time of day. cktime: cktime(1) SS
 display a prompt; verify and return an integer value. ckint: ckint(1) SARM
 display a prompt; verify and return an integer value. ckint: ckint(1) SS
 and brelse: return buffer to the bfreelist. brelse(D3DK) DDRM
 map at a YP server/ yppoll: return current version of a YP yppoll(1M) NI
 on remote machines. rusers: return information about users rusers(3N) NI
 abs, labs: return integer absolute value. abs(3C) PRM
 geterror: return I/O error. geterror(D3DK) DDRM
 master. ypwhich: return name of YP server or map ypwhich(1) NI
 size: return size of logical device. size(D2DK) DDRM
 command. rexec: return stream to a remote rexec(3N) NI
 command. rexec: return stream to a remote rexec(3N) NUAG
 integers. max: return the larger of two max(D3DK) DDRM
 path name. basename: return the last element of a basename(3G) PRM
 integers. min: return the lesser of two min(D3DK) DDRM
 message. msgdsize: return the number of bytes in a msgdsize(D3DK) DDRM
 file/ elf_fsize: elf32_fsize: return the size of an object elf_fsize(3E) PRM
 name. getenv: return value for environment getenv(3C) PRM
 ismpx: return windowing terminal state. ismpx(1) URM
 stat: data returned by stat system call. stat(4) XNX
 stat: data returned by stat system call. stat(5) PRM
 stat: data returned by stat system call. stat(5) SARM
 data buffer/ mps_get_dmabuf: returns a pointer to a list of mps_get_dmabuf(D3D) DDRM
 idcheck: returns selected information. idcheck(1M) SARM
 number. getfrm: returns the current frameID getfrm(1F) CHAR
 HOST ID record in/ ics_hostid: returns the host id field of the ics_hostid(D3D) DDRM
 registers of the board and returns the. /reads interconnect ics_read(1M) MBRMAN
 realpath: returns the real file name. realpath(3C) PRM
 Protocol server. rarpd: DARPA Reverse Address Resolution rarpd(1M) NUAG
 Protocol server. rarpd: DARPA Reverse Address Resolution rarpd(1M) SARM
 a message on a stream in the reverse direction. qreply: send qreply(D3DK) DDRM
 col: filter reverse line-feeds. col(1) URM

the last commands executed, in pointer in a stream. `fseek`, `/readdir`, `telldir`, `seekdir`, `creat`: create a new file or command.

domain administration.

domain administration.

`fromsmtp`: receive name server master file.

name server master file.

Sharing host password.

Sharing host password.

mounting by/ `share`: make local mounting by/ `share`: make local `dfmounts`: display mounted `dfmounts`: display mounted

mounting by/ `unshare`: make local mounting by/ `unshare`: make local `dfshares`: list available `dfshares`: list available

Sharing.

Sharing.

Sharing environment.

Sharing environment.

notification shell script.

notification shell script.

daemon process.

daemon process.

and by user. `hosts.equiv`, and by user. `hosts.equiv`, index, `/fmod`, `fmodf`, `fabs`, `fabsf`, `/savetty`, `getsyx`, `setsyx`, `lptest`: generate lineprinter standard/restricted/ `ksh`,

directories.

information for mail and users. `mail`, private space management map. `SCCS` file.

reverse order. `lastcomm`: show `lastcomm(1)` BSD

rewind, `ftell`: reposition a file `fseek(3S)` PRM

`rewinddir`, `closedir`: directory/ `directory(3C)` PRM

rewrite an existing one. `creat(2)` PRM

`rexec`: return stream to a remote `rexec(3N)` NI

`rexec`: return stream to a remote `rexec(3N)` NUAG

`rexc`: remote execution server. `rexc(1M)` NUAG

`rexc`: remote execution server. `rexc(1M)` SARM

`rfadmin`: Remote File Sharing `rfadmin(1M)` NUAG

`rfadmin`: Remote File Sharing `rfadmin(1M)` SARM

RFC822 mail from SMTP. `fromsmtp(1M)` SARM

`rfmaster`: Remote File Sharing `rfmaster(4)` NUAG

`rfmaster`: Remote File Sharing `rfmaster(4)` SARM

`rfpasswd`: change Remote File `rfpasswd(1M)` NUAG

`rfpasswd`: change Remote File `rfpasswd(1M)` SARM

RFS resource available for `share(1M)` NUAG

RFS resource available for `share(1M)` SARM

RFS resource information. `dfmounts(1M)` NUAG

RFS resource information. `dfmounts(1M)` SARM

RFS resource unavailable for `unshare(1M)` NUAG

RFS resource unavailable for `unshare(1M)` SARM

RFS resources from remote/ `dfshares(1M)` NUAG

RFS resources from remote/ `dfshares(1M)` SARM

`rfstart`: start Remote File `rfstart(1M)` NUAG

`rfstart`: start Remote File `rfstart(1M)` SARM

`rfstop`: stop the Remote File `rfstop(1M)` NUAG

`rfstop`: stop the Remote File `rfstop(1M)` SARM

`rfuadmin`: Remote File Sharing `rfuadmin(1M)` NUAG

`rfuadmin`: Remote File Sharing `rfuadmin(1M)` SARM

`rfudaemon`: Remote File Sharing `rfudaemon(1M)` NUAG

`rfudaemon`: Remote File Sharing `rfudaemon(1M)` SARM

`.rhosts`: trusted hosts by system `hosts.equiv(4)` NUAG

`.rhosts`: trusted hosts by system `hosts.equiv(4)` SARM

`rint`: string operations. `index(3)` BSD

`rint`, remainder: floor, ceiling,/ `floor(3M)` PRM

`ripoffline`, `curs_set`, `napms`:/ `curs_kernel(3X)` CHAR

ripple pattern. `lptest(1)` BSD

`rksh`: KornShell, a `ksh(1)` URM

`rlogin`: remote login. `rlogin(1)` NUAG

`rlogin`: remote login. `rlogin(1)` URM

`rlogind`: remote login server. `rlogind(1M)` NUAG

`rlogind`: remote login server. `rlogind(1M)` SARM

`rm`, `rmdir`: remove files or `rm(1)` URM

`rmail`. `mailcnfg`: initialization `mailcnfg(4)` SARM

`rmail`: read mail or send mail to `mail(1)` URM

`rmalloc`: allocate space from a `rmalloc(D3DK)` DDRM

`rmdel`: remove a delta from an `rmdel(1)` PRM

directories. rm,
 directories in a path. mkdirp,
 private space management map.
 space management map.
 resource information.
 resource information.
 remote resources.
 remote resources.
 mounts.
 unmount Remote File Sharing/
 unmount Remote File Sharing/
 flag for a wakeup.
 from a message.
 queue.

 bibliographic database.
 canput: test for
 chroot: change
 chroot: change
 logarithm, power, square
 size in bytes to size in pages
 size in bytes to size in pages
 routing tables.
 routing tables.

 atexit: add program termination
 a driver's message freeing
 panels virtual screen refresh
 expression compile and match
 expression compile and match
 expression compile and match
 curses bell and screen flash
 window background manipulation
 panels deck manipulation
 file for name server
 file for name server
 and window attribute control
 initialization and manipulation
 field_opts: forms field option
 miscellaneous curses utility
 /assign application-specific
 better random number generator;
 /rpc_broadcast, rpc_call: library
 /authsys_create_default: library

 rmdir: remove a directory. rmdir(2) PRM
 rmdir: remove files or rm(1) URM
 mkdirp: create, remove mkdirp(3G) PRM
 rmfree: free space back into a rmfree(D3DK) DDRM
 rminit: initialize a private rminit(D3DK) DDRM
 rmntstat: display mounted rmntstat(1M) NUAG
 rmntstat: display mounted rmntstat(1M) SARM
 rmntry: attempt to mount queued rmntry(1M) NUAG
 rmntry: attempt to mount queued rmntry(1M) SARM
 rmount: queue remote resource rmount(1M) NUAG
 rmount: queue remote resource rmount(1M) SARM
 rmountall, rumountall: mount, rmountall(1M) NUAG
 rmountall, rumountall: mount, rmountall(1M) SARM
 rmsetwant: set the map's wait rmsetwant(D3DK) DDRM
 rmvb: remove a message block rmvb(D3DK) DDRM
 rmvq: remove a message from a rmvq(D3DK) DDRM
 rmwant: wait for free memory. rmwant(D3DK) DDRM
 roffbib: format and print a roffbib(1) BSD
 room in a message queue. canput(D3DK) DDRM
 root directory. chroot(2) PRM
 root directory for a command. chroot(1M) SARM
 root functions. /exponential, exp(3M) PRM
 (round down). btop: convert btop(D3DK) DDRM
 (round up). btopr: convert btopr(D3DK) DDRM
 route: manually manipulate the route(1M) NUAG
 route: manually manipulate the route(1M) SARM
 routed: network routing daemon. routed(1M) NUAG
 routed: network routing daemon. routed(1M) SARM
 routine. atexit(3C) PRM
 routine. /that specifies free_rtn(D4DK) DDRM
 routine. /update_panels: panel_update(3X) CHAR
 routines. /advance: regular regexp(5) PRM
 routines. /advance: regular regexp(5) SARM
 routines. /advance: regular regexp(3G) PRM
 routines. /beep, flash: curs_beep(3X) CHAR
 routines. /bkgd, wbkgd: curses curs_bkgd(3X) CHAR
 routines. /bottom_panel: panel_top(3X) CHAR
 routines. /configuration resolv.conf(4) NUAG
 routines. /configuration resolv.conf(4) SARM
 routines. /curses character curs_attr(3X) CHAR
 routines. /curses screen curs_initscr(3X) CHAR
 routines. /field_opts_off, form_field_opts(3X) CHAR
 routines. /flushinp: curs_util(3X) CHAR
 routines for automatic/ menu_hook(3X) CHAR
 routines for changing/ /setstate: random(3) BSD
 routines for client side calls. rpc_clnt_calls(3N) NI
 routines for client side remote/ rpc_clnt_auth(3N) NI

/clnt_vc_create: library routines for dealing with/ rpc_clnt_create(3N) NI
 creation/ /svc_vc_create: library routines for dealing with the rpc_svc_create(3N) NI
 representation. xdr: library routines for external data xdr(3N) NI
 /xdrrec_eof, xdr_setpos: library routines for external data/ xdr_admin(3N) NI
 /xdr_wrapstring: library routines for external data/ xdr_complex(3N) NI
 /xdrstdio_create: library routines for external data/ xdr_create(3N) NI
 /xdr_u_short, xdr_void: library routines for external data/ xdr_simple(3N) NI
 /assign application-specific routines for invocation by/ form_hook(3X) CHAR
 /xprt_unregister: library routines for registering/ rpc_svc_calls(3N) NI
 calls. rpc: library routines for remote procedure rpc(3N) NI
 /xdr_replymsg: XDR library routines for remote procedure/ rpc_xdr(3N) NI
 /rpcb_set, rpcb_unset: library routines for RPC bind service. rpcbind(3N) NI
 /svc_run, svc_sendreply: library routines for RPC servers. rpc_svc_reg(3N) NI
 procedure/ /user2netname: library routines for secure remote secure_rpc(3N) NI
 /svcerr_weakauth: library routines for server side remote/ rpc_svc_err(3N) NI
 form_opts: forms option routines. /form_opts_off, form_opts(3X) CHAR
 curses refresh control routines. /is_wintouched: curs_touch(3X) CHAR
 menu_opts: menus option routines. /menu_opts_off, menu_opts(3X) CHAR
 terminal output option control routines. /nl, nonl: curses curs_outopts(3X) CHAR
 curses color manipulation routines. /pair_content: curs_color(3X) CHAR
 panels deck manipulation routines. /panel_hidden: panel_show(3X) CHAR
 dn_comp, dn_expand: resolver routines. /res_send, res_init, resolver(3N) NI
 dn_comp, dn_expand: resolver routines. /res_send, res_init, resolver(3N) NUAG
 napms: low-level curses routines. /ripoffline, curs_set, curs_kernel(3X) CHAR
 window and subwindow association routines. /scale_form: forms form_win(3X) CHAR
 window and subwindow association routines. /scale_menu: menus menu_win(3X) CHAR
 link_fieldtype: forms fieldtype routines. /set_fieldtype_choice, form_fieldtype(3X) CHAR
 menu_mark: menus mark string routines. /set_menu_mark, menu_mark(3X) CHAR
 silk_attroff: curses soft label routines. /silk_attrset, curs_silk(3X) CHAR
 curses environment query routines. /termattrs, termname: curs_termattrs(3X) CHAR
 terminal input option control routines. /typeahead: curses curs_inopts(3X) CHAR
 mailsurr: surrogate commands for routing and transport of mail. mailsurr(4) SARM
 routed: network routing daemon. routed(1M) NUAG
 routed: network routing daemon. routed(1M) SARM
 supporting for packet network routing. routing: system routing(4) NUAG
 supporting for packet network routing. routing: system routing(4) SARM
 packet network routing. routing: system supporting for routing(4) NUAG
 packet network routing. routing: system supporting for routing(4) SARM
 route: manually manipulate the routing tables. route(1M) NUAG
 route: manually manipulate the routing tables. route(1M) SARM
 /set and get maximum numbers of rows and columns in menus. menu_format(3X) CHAR
 rpcb_unset: library routines for RPC bind service. /rpcb_set, rpcbind(3N) NI
 get/set name of current secure RPC domain. domainname: domainname(1M) NI
 rpcinfo: report RPC information. rpcinfo(1M) NI
 rpcinfo: report RPC information. rpcinfo(1M) NUAG
 procedure calls. rpc: library routines for remote rpc(3N) NI
 rpc: rpc program number data base. rpc(4) NI

rpcbind: universal addresses to
 rpcbind: universal addresses to
 rpcgen: an
 base.
 library routines for
 rpcbind: rpcb_getmaps,
 rpcb_gettime,/ rpcbind:
 /rpcb_getmaps, rpcb_getaddr,
 rpcb_getaddr, rpcb_gettime,
 RPC program number mapper.
 RPC program number mapper.
 /rpcb_getaddr, rpcb_gettime,
 /clnt_sperrno, clnt_sperror,
 /rpcb_gettime, rpcb_rmtcall,
 RPC/ /rpcb_rmtcall, rpcb_set,
 /clnt_sperror, rpc_broadcast,
 authnone_create,
 clnt_freeres, clnt_geterr,
 clnt_create, clnt_destroy,
 compiler.

 xprt_register,/ rpc_svc calls:
 server.
 server.
 server.
 server.

 svc_reg, svc_unreg,
 svc_destroy, svc_dg_create,
 svcerr_decode, svcerr_noproc,
 svc_getargs, svc_getreqset,
 xdr_authsys_parms, xdr_callhdr,
 /mdiv, mcmp, min, mout, pow, gcd,

 control, and/ sh, jsh,

 mb2a_sendrsvp: Sends a
 /cancels an ongoing

 parameter table.
 resource request.
 resource request.

 RPC program number mapper. rpcbind(1M) NI
 RPC program number mapper. rpcbind(1M) NUAG
 RPC protocol compiler. rpcgen(1) NI
 rpc: rpc program number data rpc(4) NI
 RPC servers. /svc_sendreply: rpc_svc_reg(3N) NI
 rpcb_getaddr, rpcb_gettime,/ rpcbind(3N) NI
 rpcb_getmaps, rpcb_getaddr, rpcbind(3N) NI
 rpcb_gettime, rpcb_rmtcall,/ rpcbind(3N) NI
 rpcbind: rpcb_getmaps, rpcbind(3N) NI
 rpcbind: universal addresses to rpcbind(1M) NI
 rpcbind: universal addresses to rpcbind(1M) NUAG
 rpcb_rmtcall, rpcb_set,/ rpcbind(3N) NI
 rpc_broadcast, rpc_call: library/ rpc_clnt_calls(3N) NI
 rpcb_set, rpcb_unset: library/ rpcbind(3N) NI
 rpcb_unset: library routines for rpcbind(3N) NI
 rpc_call: library routines for/ rpc_clnt_calls(3N) NI
 rpc_clnt_auth: auth_destroy, rpc_clnt_auth(3N) NI
 rpc_clnt_calls: clnt_call, rpc_clnt_calls(3N) NI
 rpc_clnt_create: clnt_control, rpc_clnt_create(3N) NI
 rpcgen: an RPC protocol rpcgen(1) NI
 rpcinfo: report RPC information. rpcinfo(1M) NI
 rpcinfo: report RPC information. rpcinfo(1M) NUAG
 rpc_reg, svc_reg, svc_unreg, rpc_svc_calls(3N) NI
 rpc.rusersd: network username rusersd(1M) NI
 rpc.rusersd: network username rusersd(1M) NUAG
 rpc.rwall: network rwall rwall(1M) NI
 rpc.rwall: network rwall rwall(1M) NUAG
 rpc.sprayd: spray server. sprayd(1M) NI
 rpc.sprayd: spray server. sprayd(1M) NUAG
 rpc_svc calls: rpc_reg rpc_svc_calls(3N) NI
 rpc_svc_create: svc_create, rpc_svc_create(3N) NI
 rpc_svc_err: svcerr_auth, rpc_svc_err(3N) NI
 rpc_svc_reg: svc_freeargs, rpc_svc_reg(3N) NI
 rpc_xdr: xdr_accepted_reply, rpc_xdr(3N) NI
 rpow, msqrt, sdiv, itom, xtom,/ mp(3X) BSD
 rsh: remote shell. rsh(1) NUAG
 rsh: remote shell. rsh(1) URM
 rsh: shell, the standard, job sh(1) URM
 rshd: remote shell server. rshd(1M) NUAG
 rshd: remote shell server. rshd(1M) SARM
 rsvp request message.. mb2a_sendrsvp(3tai) TAI
 rsvp transaction. mps_AMPcancel(D3D) DDRM
 rtc: MULTIBUS clock driver. rtc(1M) MBRMAN
 rtc: real time clock interface. rtc(7) SARM
 rt_dptbl: real-time dispatcher rt_dptbl(4) SARM
 rumount: cancel queued remote rumount(1M) NUAG
 rumount: cancel queued remote rumount(1M) SARM

Remote File Sharing/ rmountall, rmountall(1M) NUAG
 Remote File Sharing/ rmountall, rmountall(1M) SARM
 nice: run a command at low priority. nice(1) URM
 and quits. nohup: run a command immune to hangups nohup(1) URM
 shell: run a command using shell. shell(1F) CHAR
 run: run an executable. run(1F) CHAR
 atq: display the jobs queued to run at specified times. atq(1) URM
 multi-user environment. rc2: run commands performed for rc2(1M) SARM
 the operating system. rc0: run commands performed to stop rc0(1M) SARM
 and reboot the operating/ rc6: run commands performed to stop rc6(1M) SARM
 runacct: run daily accounting. runacct(1M) SARM
 run: run an executable. run(1F) CHAR
 runacct: run daily accounting. runacct(1M) SARM
 runacct, shutacct, startup,/ acctsh(1M) SARM
 /prctmp, prdaily, prtacct, running processes. gcore(1) URM
 gcore: get core images of running processes. renice(1M) BSD
 renice: alter priority of runs an initialization file. reinit(1F) CHAR
 reinit: runs an initialization file. reinit(1F) CHAR
 local machines. ruptime: show host status of ruptime(1) NUAG
 local machines. ruptime: show host status of ruptime(1) URM
 users on remote machines. rusers: return information about rusers(3N) NI
 machines. rusers: who's logged in on local rusers(1) NI
 machines. rusers: who's logged in on local rusers(1) NUAG
 rpc.rwalld: network rwall server. rwall(1M) NI
 rpc.rwalld: network rwall server. rwall(1M) NUAG
 network. rwall: write to all users over a rwall(1M) NI
 network. rwall: write to all users over a rwall(1M) NUAG
 machines. rwall: write to specified remote rwall(3N) NI
 machines. rwho: who's logged in on local rwho(1) NUAG
 machines. rwho: who's logged in on local rwho(1) URM
 server. rwhod, in.rwhod: system status rwhod(1M) NUAG
 server. rwhod, in.rwhod: system status rwhod(1M) SARM
 systems. fsck (s5): check and repair s5 file fsck(1M) SARM
 system. mkfs (s5): construct an s5 file mkfs(1M) SARM
 optimal access time. dcopy (s5): copy s5 file systems for dcopy(1M) SARM
 dir (s5): format of s5 directories. dir(4) SARM
 information. ff (s5): display i-list ff(1M) SARM
 mkfs (s5): construct an s5 file system. mkfs(1M) SARM
 mount (s5): mount an s5 file system. mount(1M) SARM
 fsdb (s5): s5 file system debugger. fsdb(1M) SARM
 (s5): make a literal copy of an s5 file system. volcopy volcopy(1M) SARM
 fs (s5): format of s5 file system volume. fs(4) SARM
 fsck (s5): check and repair s5 file systems. fsck(1M) SARM
 labelit (s5): provide labels for s5 file systems. labelit(1M) SARM
 access time. dcopy (s5): copy s5 file systems for optimal dcopy(1M) SARM
 free disk blocks and i-nodes for s5 file systems. /number of df(1M) SARM
 path names versus i-numbers for s5 file systems. /(s5): generate ncheck(1M) SARM
 inode (s5): format of an s5 i-node. inode(4) SARM

dir	(s5): format of s5 directories.	dir(4) SARM
volume. fs	(s5): format of s5 file system	fs(4) SARM
i-numbers for s5 file/ ncheck	(s5): generate path names versus	ncheck(1M) SARM
inode (s5): format of an	s5 i-node.	inode(4) SARM
s5 file system. volcopy	(s5): make a literal copy of an	volcopy(1M) SARM
mount	(s5): mount an s5 file system.	mount(1M) SARM
systems. labelit	(s5): provide labels for s5 file	labelit(1M) SARM
blocks and i-nodes for s5/ df	(s5): report number of free disk	df(1M) SARM
fsdb	(s5): s5 file system debugger.	fsdb(1M) SARM
System Administration.	SA: devices administered by	sa(7) SARM
report package. sar:	sa1, sa2, sadc: system activity	sar(1M) SARM
report package. sar: sa1,	sa2, sadc: system activity	sar(1M) SARM
	sac: service access controller.	sac(1M) NI
	sac: service access controller.	sac(1M) SARM
controller administration.	sacadm: service access	sacadm(1M) NI
controller administration.	sacadm: service access	sacadm(1M) SARM
editing activity.	sact: print current SCCS file	sact(1) PRM
Driver.	sad: STREAMS Administrative	sad(7) STRM
package. sar: sa1, sa2,	sadc: system activity report	sar(1M) SARM
	sag: system activity graph.	sag(1) URM
same type.	SAMESTR: test if next queue is	SAMESTR(D3DK) DDRM
activity report package.	sar: sa1, sa2, sadc: system	sar(1M) SARM
	sar: system activity reporter.	sar(1) URM
/reset_shell_mode, resetty,	savetty, getsyx, setsyx,/	cursor_kernel(3X) CHAR
allocation. brk,	sbrk: change data segment space	brk(2) PRM
logb, modf, modff, nextafter,	scalb: manipulate. /ldexp,	frexp(3C) PRM
for/ /fp_class, isnan, copysign,	scalbn: miscellaneous functions	ieee_functions(3M) BSD
/set_form_sub, form_sub,	scale_form: forms window and/	form_win(3X) CHAR
/set_menu_sub, menu_sub,	scale_menu: menus window and/	menu_win(3X) CHAR
scandir, alphasort:	scan a directory.	scandir(3) BSD
directory.	scandir, alphasort: scan a	scandir(3) BSD
formatted input.	scanf, fscanf, sscanf: convert	scanf(3S) PRM
bfs: big file	scanner.	bfs(1) URM
language. awk: pattern	scanning and processing	awk(1) URM
language. nawk: pattern	scanning and processing	nawk(1) URM
mvwscanw, vwscanw:/ curs_scanw:	scanw, wscanw, mvscanw,	cursor_scanw(3X) CHAR
the network. spray:	scatter data in order to check	spray(3N) NI
structure. uio:	scatter/gather I/O request	uio(D4DK) DDRM
sdi_translate: translate	scb virtual addresses.	sdi_translate(D3I) SCSI
for help with message numbers or	SCCS commands. help: ask	help(1) PRM
change the delta comment of an	SCCS delta. cdc:	cdc(1) PRM
comb: combine	SCCS deltas.	comb(1) PRM
get: get a version of an	SCCS file.	get(1) PRM
prs: print an	SCCS file.	prs(1) PRM
rmdel: remove a delta from a	SCCS file.	rmdel(1) PRM
scsfile: format of	SCCS file.	scsfile(4) PRM
scsfile: format of	SCCS file.	scsfile(4) SARM

unget: undo a previous get of an
 val: validate an
 make a delta (change) to an
 and commentary history of an
 sact: print current
 compare two versions of an
 admin: create and administer
 Code Control System (SCCS).
 the Source Code Control System
 of an SCCS file.

 check file system backup
 in microseconds. ualarm:
 prevent a queue from being
 dispadmin: process
 priosctnl: process
 priosctnl: process
 priosctnlset: generalized process
 transport program. uusched: the
 /setpriority: get/set program
 scr_set: read/ curs_scr_dump:
 clear: clear the terminal
 /beep, flash: curses bell and
 /scr_set: read (write) a curses
 package. curses: CRT
 /set_term, delscreen: curses
 a panels window on the virtual
 /update_panels: panels virtual
 editor based on ex. vi:
 a curses/ /scr_dump, scr_restore,
 asynchronous terminal controller
 execute a configuration
 inittab:
 terminal session.
 stores answers to a request
 stores answers to a request
 File Sharing notification shell
 File Sharing notification shell
 XENIX installation shell
 /scroll, srcl, wscr:
 curses window. curs_scroll:
 /leaveok, setscreg, wssetscreg,
 read/ curs_scr_dump: scr_dump,
 /scr_dump, scr_restore, scr_init,
 driver. sdi_getblk: allocate a
 release a previously allocated
 SCCS file.

 SCCS file.

 SCCS file.

 SCCS file. delta:

 SCCS file. /display the delta

 SCCS file editing activity.

 SCCS file. sccsdiff:

 SCCS files.

 scs: front end for the Source
 (SCCS). scs: front end for

 scsdiff: compare two versions

 scsfile: format of SCCS file.

 scsfile: format of SCCS file.

 schedule. ckbupscd:

 schedule signal after interval

 scheduled. noenable:

 scheduler administration.

 scheduler control.

 scheduler control.

 scheduler control.

 scheduler for the uucp file

 scheduling priority.

 scr_dump, scr_restore, scr_init,

 screen.

 screen flash routines.

 screen from (to) a file.

 screen handling and optimization

 screen initialization and/

 screen. /move_panel: move

 screen refresh routine.

 screen-oriented (visual) display

 scr_init, scr_set: read (write)

 script device driver. atcs:

 script. doconfig:

 script for init.

 script: make typescript of a

 script. pkgask:

 script. pkgask:

 script. rfuadmin: Remote

 script. rfuadmin: Remote

 script. xinstall:

 scroll a curses window.

 scroll, srcl, wscr: scroll a

 scrollok, nl, nonl: curses/

 scr_restore, scr_init, scr_set:

 scr_set: read (write) a curses/

 SCSI block for the target

 SCSI block. sdi_freeblk:

 unget(1) PRM
 val(1) PRM
 delta(1) PRM
 prt(1) BSD
 sact(1) PRM
 scsdiff(1) PRM
 admin(1) PRM
 scs(1) BSD
 scs(1) BSD
 scsdiff(1) PRM
 scsfile(4) PRM
 scsfile(4) SARM
 ckbupscd(1M) SARM
 ualarm(3) BSD
 noenable(D3DK) DDRM
 dispadmin(1M) SARM
 priosctnl(1) URM
 priosctnl(2) PRM
 priosctnlset(2) PRM
 uusched(1M) SARM
 getpriority(3) BSD
 curs_scr_dump(3X) CHAR
 clear(1) URM
 curs_beep(3X) CHAR
 curs_scr_dump(3X) CHAR
 curses(3X) CHAR
 curs_initscr(3X) CHAR
 panel_move(3X) CHAR
 panel_update(3X) CHAR
 vi(1) URM
 curs_scr_dump(3X) CHAR
 atcs(1M) MBRMAN
 doconfig(3N) NI
 inittab(4) SARM
 script(1) URM
 pkgask(1M) SARM
 pkgask(1M) SS
 rfuadmin(1M) NUAG
 rfuadmin(1M) SARM
 xinstall(1M) SARM
 curs_scroll(3X) CHAR
 curs_scroll(3X) CHAR
 curs_outopts(3X) CHAR
 curs_scr_dump(3X) CHAR
 curs_scr_dump(3X) CHAR
 sdi_getblk(D3I) SCSI
 sdi_freeblk(D3I) SCSI

sdi_send: send SCSI command to the controller.	sdi_send(D3I) SCSI
sdb: symbolic debugger.	sdb(1) PRM
sdenter, sdleave: synchronize access to a shared data/	sdenter(2) XNX
sdevice: file format.	sdevice(4) SARM
sdfree: attach and detach a shared data segment. sdget,	sdget(2) XNX
a shared data segment. sdget, sdfree: attach and detach	sdget(2) XNX
access. sdgetv: synchronize shared data	sdgetv(2) XNX
side-by-side. sdiff: print file differences	sdiff(1) URM
previously allocated SCSI/ sdi_freeblk: release a	sdi_freeblk(D3I) SCSI
block for the target driver. sdi_getblk: allocate a SCSI	sdi_getblk(D3I) SCSI
number to pass-through device/ sdi_getdev: convert device	sdi_getdev(D3I) SCSI
operation immediately. sdi_icmd: perform requested	sdi_icmd(D3I) SCSI
adapter. sdi_init: initialize the host	sdi_init(D3I) SCSI
controller. sdi_name: get name of addressed	sdi_name(D3I) SCSI
the controller. sdi_send: send SCSI command to	sdi_send(D3I) SCSI
virtual addresses. sdi_translate: translate scb	sdi_translate(D3I) SCSI
/mout, pow, gcd, rpow, msqrt, sdiv, itom, xtom, mtox, mfree: /	mp(3X) BSD
shared data segment. sdenter, sdleave: synchronize access to a	sdenter(2) XNX
string. fgrep: search a file for a character	fgrep(1) URM
string. fgrep: search a file for a character	fgrep(1) XNX
using full regular/ egrep: search a file for a pattern	egrep(1) URM
using full regular/ egrep: search a file for a pattern	egrep(1) XNX
fmlgrep: search a file for a pattern.	fmlgrep(1F) CHAR
grep: search a file for a pattern.	grep(1) URM
grep: search a file for a pattern.	grep(1) XNX
bsearch: binary search a sorted table.	bsearch(3C) PRM
accounting file(s). acctcom: search and print process	acctcom(1) URM
lsearch, lfind: linear search and update.	lsearch(3C) PRM
filename. pathconv: search FMLI criteria for	pathconv(1F) CHAR
srchtxt: display contents of, or search for a text string in,/	srchtxt(1) URM
directories. pathfind: search for named file in named	pathfind(3G) PRM
tysrch: directory search list for ttyname.	tysrch(4) SARM
hcreate, hdestroy: manage hash search tables. hsearch,	hsearch(3C) PRM
tdelete, twalk: manage binary search trees. tsearch, tfind,	tsearch(3C) PRM
econvert, fconvert, gconvert, seconvert, sfconvert, sgconvert:/	econvert(3) BSD
keylogin: decrypt and store secret key.	keylogin(1) NI
keylogin: decrypt and store secret key.	keylogin(1) NUAG
getsecretkey: retrieve public or secret key. /getpublickey,	publickey(3N) NI
elf_newdata, elf_rawdata: get section data. elf_getdata,	elf_getdata(3E) PRM
retrieve class-dependent section header. /elf32_getshdr:	elf_getshdr(3E) PRM
/elf_newscn, elf_nextscn: get section information.	elf_getscn(3E) PRM
mcs: manipulate the comment section of an object file.	mcs(1) PRM
files. size: print section sizes in bytes of object	size(1) PRM
/library routines for secure remote procedure calls.	secure_rpc(3N) NI
get/set name of current secure RPC domain. domainname:	domainname(1M) NI
authdes_getucred, getnetname,/ secure_rpc: authdes_seccreate,	secure_rpc(3N) NI
sed: stream editor.	sed(1) URM

/mrand48, jrand48, srand48, seed48, lcong48: generate/ drand48(3C) PRM
 /opendir, readdir, telldir, seekdir, rewinddir, closedir:/ directory(3C) PRM
 user space. segmap: map device memory into segmap(D2K) DDRM
 shmget: get shared memory segment identifier. shmget(2) PRM
 attach and detach a shared data segment. sdget, sdfree: sdget(2) XNX
 access to a shared data segment. /sdlleave: synchronize sdenter(2) XNX
 brk, sbrk: change data segment space allocation. brk(2) PRM
 two sorted files. comm: select or reject lines common to comm(1) URM
 multiplexing. select: synchronous I/O select(3C) NI
 a file. cut: cut out selected fields of each line of cut(1) URM
 a file. fmlcut: cut out selected fields of each line of fmlcut(1F) CHAR
 idcheck: returns selected information. idcheck(1M) SARM
 file. dump: dump selected parts of an object dump(1) PRM
 opensem: open a semaphore. opensem(2) XNX
 to a resource governed by a semaphore. /and check access waitsem(2) XNX
 semctl: semaphore control operations. semctl(2) PRM
 create an instance of a binary semaphore. creatsem: creatsem(2) XNX
 semop: semaphore operations. semop(2) PRM
 ipcrm: remove a message queue, semaphore set, or shared memory/ ipcrm(1) URM
 signal a process waiting on a semaphore. sigsem: sigsem(2) XNX
 semget: get set of semaphores. semget(2) PRM
 operations. semctl: semaphore control semctl(2) PRM
 semget: get set of semaphores. semget(2) PRM
 semop: semaphore operations. semop(2) PRM
 queue. putctl: send a control message to a putctl(D3DK) DDRM
 one-byte parameter to/ putctl1: send a control message with a putctl1(D3DK) DDRM
 t_sndudata: send a data unit. t_sndudata(3N) NI
 lpr: send a job to the printer. lpr(1) BSD
 send, sendto, sendmsg: send a message from a socket. send(3N) NI
 putmsg: send a message on a stream. putmsg(2) PRM
 putmsg: send a message on a stream. putmsg(2) STRM
 the reverse direction. qreply: send a message on a stream in qreply(D3DK) DDRM
 queue. putnext: send a message to the next putnext(D3DK) DDRM
 group. signal: send a signal to a process signal(D3D) DDRM
 group of processes. kill: send a signal to a process or a kill(2) PRM
 group of/ sigsend, sigsendset: send a signal to a process or a sigsend(2) PRM
 a connection. t_snd: send data or expedited data over t_snd(3N) NI
 to network hosts. ping: send ICMP ECHO_REQUEST packets ping(1M) NUAG
 to network hosts. ping: send ICMP ECHO_REQUEST packets ping(1M) SARM
 message. nlsrequest: format and send listener service request nlsrequest(3N) NI
 sendmail: send mail over the internet. sendmail(1M) BSD
 tosmtp: send mail to SMTP. tosmtp(1M) SARM
 mail, rmail: read mail or send mail to users. mail(1) URM
 controller. sdi_send: send SCSI command to the sdi_send(D3I) SCSI
 message from a socket. send, sendto, sendmsg: send a send(3N) NI
 psignal: send signal to a process. psignal(D3D) DDRM
 killpg: send signal to a process group. killpg(3) BSD

raise: send signal to program. raise(3C) PRM
 using Simple Mail/ smtp: send SMTP mail to a remote host smtp(1M) SARM
 request. t_snddis: send user-initiated disconnect t_snddis(3N) NI
 print service. lp, cancel: send/cancel requests to an LP lp(1) URM
 addresses and aliases for sendmail. /addresses, forward: aliases(4) BSD
 print statistics collected by sendmail. mailstats: mailstats(1M) BSD
 internet. sendmail: send mail over the sendmail(1M) BSD
 socket. send, sendto, sendmsg: send a message from a send(3N) NI
 asynchronous/ mb2a_brdcst: Sends a broadcast message in mb2a_brdcst(3tai) TAI
 synchronous mode.. mb2s_brdcst: Sends a broadcast message in mb2s_brdcst(3tai) TAI
 mb2a_sendcancel: Sends a cancel message/ mb2a_sendcancel(3tai) TAI
 synchronous. mb2s_sendcancel: Sends a cancel message in mb2s_sendcancel(3tai) TAI
 mb2a_sendreply: Sends a reply message/ mb2a_sendreply(3tai) TAI
 synchronous/ mb2s_sendreply: Sends a reply message in mb2s_sendreply(3tai) TAI
 receives a/ mb2s_sendsrvp: Sends a request message and mb2s_sendsrvp(3tai) TAI
 mb2a_sendsrvp: Sends a rsvp request message.. mb2a_sendsrvp(3tai) TAI
 in asynchronous/ mb2a_send: Sends a transactionless message mb2a_send(3tai) TAI
 in synchronous/ mb2s_send: Sends a transactionless message mb2s_send(3tai) TAI
 part of any. mps_AMPsend_data: sends solicited data that is not
 mps_AMPsend_data(D3D) DDRM
 are not part of. mps_AMPsend: sends unsolicited messages that mps_AMPsend(D3D) DDRM
 from a socket. send, sendto, sendmsg: send a message send(3N) NI
 unsolicited reply message to be sent.. /constructs a mps_mk_unsolrply(D3D) DDRM
 a broadcast message to be sent.. /constructs mps_mk_brdcst(D3D) DDRM
 an unsolicited message to be sent.. mps_mk_unsol: constructs ... mps_mk_unsol(D3D) DDRM
 receive data or expedited data sent over a connection. t_rcv: t_rcv(3N) NI
 /constructs a message to be sent to initiate a. mps_mk_solrply(D3D) DDRM
 /constructs a message to be sent to initiate a solicited. mps_mk_sol(D3D) DDRM
 /maintain line settings and hunt sequences for TTY ports. sttydefs(1M) SARM
 access. elf_next: sequential archive member elf_next(3E) PRM
 device driver supporting Bus, serial, and AT&T. mouse: mouse mouse(7) MOUSE
 device/ i410: iSBC 186/410 serial communications controller i410(1M) MBRMAN
 i546: iSBC 546 multi-port serial controller. i546(1M) MBRMAN
 asy: asynchronous serial port. asy(7) SARM
 i354: iSBX 354 dual channel serial-port device driver. i354(1M) MBRMAN
 bootparamd: boot parameter server. bootparamd(1M) NI
 bootparamd: boot parameter server. bootparamd(1M) NUAG
 comsat, in.comsat: biff server. comsat(1M) NUAG
 comsat, in.comsat: biff server. comsat(1M) SARM
 ftpd: file transfer protocol server. ftpd(1M) NUAG
 ftpd: file transfer protocol server. ftpd(1M) SARM
 mountd: NFS mount request server. mountd(1M) NUAG
 mountd: NFS mount request server. mountd(1M) SARM
 rexecd: remote execution server. rexecd(1M) NUAG
 rexecd: remote execution server. rexecd(1M) SARM
 rlogind: remote login server. rlogind(1M) NUAG
 rlogind: remote login server. rlogind(1M) SARM

rshd: remote shell	server.	rshd(1M) NUAG
rshd: remote shell	server.	rshd(1M) SARM
rpc.rusersd: network username	server.	rusersd(1M) NI
rpc.rusersd: network username	server.	rusersd(1M) NUAG
rpc.rwalld: network rwall	server.	rwalld(1M) NI
rpc.rwalld: network rwall	server.	rwalld(1M) NUAG
rwhod, in.rwhod: system status	server.	rwhod(1M) NUAG
rwhod, in.rwhod: system status	server.	rwhod(1M) SARM
rpc.sprayd: spray	server.	sprayd(1M) NI
rpc.sprayd: spray	server.	sprayd(1M) NUAG
telnetd: DARPA TELNET protocol	server.	telnetd(1M) NUAG
telnetd: DARPA TELNET protocol	server.	telnetd(1M) SARM
ypserv, ypbind: YP	server and binder processes.	ypserv(1M) NI
remote user information	server. fingerd, in.fingerd:	fingerd(1M) NUAG
remote user information	server. fingerd, in.fingerd:	fingerd(1M) SARM
information. yppupdated:	server for changing YP	yppupdated(1M) NI
private keys. keyserv:	server for storing public and	keyserv(1M) NI
private keys. keyserv:	server for storing public and	keyserv(1M) NUAG
talkd, in.talkd:	server for talk program.	talkd(1M) NUAG
talkd, in.talkd:	server for talk program.	talkd(1M) SARM
for dealing with the creation of	server handles. /routines	rpc_svc_create(3N) NI
version of a YP map at a YP	server host. /return current	yppoll(1M) NI
communications/ ccisrvinf: CCI	server information on the	ccisrvinf(1M) MBRMAN
Remote File Sharing name	server master file. rfmaster:	rfmaster(4) NUAG
Remote File Sharing name	server master file. rfmaster:	rfmaster(4) SARM
in.named: Internet domain name	server. named,	named(1M) NUAG
in.named: Internet domain name	server. named,	named(1M) SARM
ypwhich: return name of YP	server or map master.	ypwhich(1) NI
Remote File Sharing name	server query. nsquery:	nsquery(1M) NUAG
Remote File Sharing name	server query. nsquery:	nsquery(1M) SARM
Address Resolution Protocol	server. rarpd: DARPA Reverse	rarpd(1M) NUAG
Address Resolution Protocol	server. rarpd: DARPA Reverse	rarpd(1M) SARM
configuration file for name	server routines. resolv.conf:	resolv.conf(4) NUAG
configuration file for name	server routines. resolv.conf:	resolv.conf(4) SARM
call/ /library routines for	server side remote procedure	rpc_svc_err(3N) NI
Trivial File Transfer Protocol	server. tftpd: DARPA	tftpd(1M) NUAG
Trivial File Transfer Protocol	server. tftpd: DARPA	tftpd(1M) SARM
in.tnamed: DARPA trivial name	server. tnamed,	tnamed(1M) NUAG
in.tnamed: DARPA trivial name	server. tnamed,	tnamed(1M) SARM
ypxfr: transfer YP map from a YP	server to host.	ypxfr(1M) NI
point ypbind at a particular	server. ypset:	ypset(1M) NI
inetd.conf: Internet	servers database.	inetd.conf(4) NUAG
inetd.conf: Internet	servers database.	inetd.conf(4) SARM
nslookup: query name	servers interactively.	nslookup(1M) NUAG
nslookup: query name	servers interactively.	nslookup(1M) SARM
library routines for RPC	servers. /svc_sendreply:	rpc_svc_reg(3N) NI
library routines for registering	servers. /xprt_unregister:	rpc_svc_calls(3N) NI

Permuted Index

calendar: reminder service. calendar(1) URM
 enableok: reschedule a queue for service. enableok(D3DK) DDRM
 lpadmin: configure the LP print service. lpadmin(1M) SARM
 sac: service access controller. sac(1M) NI
 sac: service access controller. sac(1M) SARM
 administration. sacadm: service access controller sacadm(1M) NI
 administration. sacadm: service access controller sacadm(1M) SARM
 nlsadmin: network listener service administration. nlsadmin(1M) SARM
 /lpmove: start/stop the LP print service and move requests. lpsched(1M) SARM
 requests to an LP print service. /cancel: send/cancel lp(1) URM
 starts a MULTIBUS II boot service daemon. bootserver: bootserver(1M) MBRMAN
 setservent, endservent: get service entry. /getservbyname, getservent(3N) NI
 uuglist: print the list of service grades that are/ uuglist(1C) URM
 t_getinfo: get protocol-specific service information. t_getinfo(3N) NI
 filters used with the LP print service. lpfiler: administer lpfiler(1M) SARM
 forms used with the LP print service. lpforms: administer lpforms(1M) SARM
 remote systems with the print service. lpsystem: register lpsystem(1M) SARM
 /with backup operations to service media insertion prompts. bkoper(1M) SARM
 ots: System V/386 OSI Transport Service (ots) device driver. ots(1M) MBRMAN
 about the status of the LP print service. /print information lpstat(1) URM
 srv: service queued messages. srv(D2DK) DDRM
 /format and send listener service request message. nlsrequest(3N) NI
 library routines for RPC bind service. /rpcb_set, rpcb_unset: rpcbind(3N) NI
 Internet user name directory service. whois: whois(1) NUAG
 Internet user name directory service. whois: whois(1) URM
 services: Internet services and aliases. services(4) NUAG
 services: Internet services and aliases. services(4) SARM
 inetd: Internet services daemon. inetd(1M) NI
 inetd: Internet services daemon. inetd(1M) NUAG
 inetd: Internet services daemon. inetd(1M) SARM
 aliases. services: Internet services and services(4) NUAG
 aliases. services: Internet services and services(4) SARM
 or control a system backup session. backup: initiate backup(1M) SARM
 getsid: get session ID. getsid(2) PRM
 setsid: set session ID. setsid(2) PRM
 make typescript of a terminal session. script: script(1) URM
 ascii: map of ASCII character set. ascii(5) PRM
 ascii: map of ASCII character set. ascii(5) SARM
 length. truncate, ftruncate: set a file to a specified truncate(3C) PRM
 alarm: set a process alarm clock. alarm(2) PRM
 /top_row, item_index: set and get current menus items. menu_item_current(3X) CHAR
 umask: set and get file creation mask. umask(2) PRM
 /field_status, set_max_field: set and get forms field/ form_field_buffer(3X) CHAR
 /set_menu_format, menu_format: set and get maximum numbers of/ menu_format(3X) CHAR
 /set_item_value, item_value: set and get menus item values. menu_item_value(3X) CHAR
 /set_menu_pattern, menu_pattern: set and get menus pattern match/ menu_pattern(3X) CHAR

environment/ set, unset: set and unset local or global set(1F) CHAR
 context. sigstack: set and/or get signal stack sigstack(3) BSD
 files. ckperms: set and/or verify permissions on ckperms(1M) MBRMAN
 ffs: find first set bit ffs(3C) PRM
 iconv: code set conversion tables. iconv(5) SARM
 iconv: code set conversion utility. iconv(1) URM
 sigsetmask: set current signal mask. sigsetmask(3) BSD
 getcontext, setcontext: get and set current user context. getcontext(2) PRM
 timezone: set default system time zone. timezone(4) PRM
 timezone: set default system time zone. timezone(4) SARM
 execution. env: set environment for command env(1) URM
 times. utime: set file access and modification utime(2) PRM
 utimes: set file times. utimes(3) BSD
 umask: set file-creation mode mask. umask(1) URM
 elf fill: set fill byte. elf_fill(3E) PRM
 /current_field, field_index: set forms current page and/ form_page(3X) CHAR
 semget: set of semaphores. semget(2) PRM
 /move an archive from one set of volumes to another. migration(1M) SARM
 getsockopt, setsockopt: get and set options on sockets. getsockopt(3N) NI
 stack context. sigaltstack: set or get signal alternate sigaltstack(2) PRM
 host system. hostname: set or print name of current hostname(1) BSD
 /a message queue, semaphore set, or shared memory ID. ipcrm(1) URM
 environment variables currently set. printenv: display printenv(1) BSD
 lpusers: set printing queue priorities. lpusers(1M) SARM
 setpgid: set process group ID. setpgid(2) PRM
 setpgrp: set process group ID. setpgrp(2) PRM
 mapping. mprotect: set protection of memory mprotect(2) PRM
 IDs. setregid: set real and effective group setregid(3) BSD
 setreuid: set real and effective user IDs. setreuid(3) BSD
 setsid: set session ID. setsid(2) PRM
 getgroups, setgroups: get or set supplementary group access/ getgroups(2) PRM
 host. rdate: set system date from a remote rdate(1M) NUAG
 host. rdate: set system date from a remote rdate(1M) SARM
 sysinfo: get and set system information strings. sysinfo(2) PRM
 clock. setclk: set system time from hardware setclk(1M) SARM
 tabs: set tabs on a terminal. tabs(1) URM
 group id. tcsetpgrp: set terminal foreground process tcsetpgrp(3C) PRM
 and line discipline. getty: set terminal type, modes, speed, getty(1M) SARM
 panels/ /replace_panel: get or set the current window of a panel_window(3X) CHAR
 date: print and set the date. date(1) URM
 /settimeofday: get or set the date and time. gettimeofday(3) BSD
 /settimeofday: get or set the date and time. gettimeofday(3C) PRM
 wakeup. rmsetwant: set the map's wait flag for a rmsetwant(D3DK) DDRM
 stty: set the options for a terminal. stty(1) BSD
 stty: set the options for a terminal. stty(1) URM
 stime: set time. stime(2) PRM
 or global environment/ set, unset: set and unset local set(1F) CHAR

/f4diskadd/f1: disk set up utility. diskadd(1M) SARM
 disksetup: disk set up utility. disksetup(1M) SARM
 setuid, setgid: set user and group IDs. setuid(2) PRM
 ulimit: get and set user limits. ulimit(2) PRM
 parameter. idtune: attempts to set value of a tunable idtune(1M) SARM
 setvbuf: assign buffering to a/ setbuf, setbuffer, setlinebuf, setbuf(3S) BSD
 buffering to a stream. setbuf, setvbuf: assign setbuf(3S) PRM
 buffering to a stream. setbuffer, setlinebuf: assign setbuffer(3S) BSD
 assign buffering to a/ setbuf, setbuffer, setlinebuf, setvbuf: setbuf(3S) BSD
 hardware clock. setcl: set system time from setcl(1M) SARM
 color. setcolor: redefine or create a setcolor(1F) CHAR
 user context. getcontext: setcontext: get and set current getcontext(2) PRM
 set_form_page, form_page: set_current_field,/ form_page: form_page(3X) CHAR
 set_top_row,/ menu_item_current: set_current_item, current_item, menu_item_current(3X) CHAR
 /setupterm, setterm, set_curterm, del_curterm,/ curs_terminfo(3X) CHAR
 /set_field_fore, field_fore, set_field_back, field_back,/ form_field_attributes(3X) CHAR
 form field buffer: set_field_buffer, field_buffer,/ form_field_buffer(3X) CHAR
 form_field_attributes: set_field_fore, field_fore,/ form_field_attributes(3X) CHAR
 /set_form_term, form_term, set_field_init, field_init,/ form_hook(3X) CHAR
 format the/ form_field_just: set_field_just, field_just: form_field_just(3X) CHAR
 form_field_opts: set_field_opts, field_opts_on,/ form_field_opts(3X) CHAR
 the/ /set_field_back, field_back, set_field_pad, field_pad: format form_field_attributes(3X) CHAR
 form_field_attributes(3X) CHAR
 /set_field_buffer, field_buffer, set_field_status, field_status,/ form_field_buffer(3X) CHAR
 /set_field_init, field_init, set_field_term, field_term:/ form_hook(3X) CHAR
 form_field_validation: set_field_type, field_type,/ form_field_validation(3X) CHAR
 /new_fieldtype, free_fieldtype, set_fieldtype_arg,/ form_fieldtype(3X) CHAR
 /set_fieldtype_arg, set_fieldtype_choice,/ form_fieldtype(3X) CHAR
 form_field_userptr: set_field_userptr,/ form_field_userptr(3X) CHAR
 field_count,/ form_field: set_form_fields, form_fields, form_field(3X) CHAR
 set_form_term,/ form_hook: set_form_init, form_init, form_hook(3X) CHAR
 form_opts_off,/ form_opts: set_form_opts, form_opts_on, form_opts(3X) CHAR
 set_current_field,/ form_page: set_form_page, form_page, form_page(3X) CHAR
 /set_form_win, form_win, set_form_sub, form_sub,/ form_win(3X) CHAR
 /set_form_init, form_init, set_form_term, form_term,/ form_hook(3X) CHAR
 associate/ form_userptr: set_form_userptr, form_userptr: form_userptr(3X) CHAR
 set_form_sub,/ form_win: set_form_win, form_win, form_win(3X) CHAR
 setuid, setgid: set user and group IDs. setuid(2) PRM
 getgrent, getgrgid, getgrnam: setgrent, endgrent, fgetgrent:/ getgrent(3C) PRM
 supplementary group/ getgroups: setgroups: get or set getgroups(2) PRM
 /gethostbyaddr, gethostbyname, sethostent, endhostent: get/ gethostent(3N) NI
 current host. gethostname: sethostname: get/set name of gethostname(3) BSD
 set_item_term,/ menu_hook: set_item_init, item_init, menu_hook(3X) CHAR
 item_opts_off, menu_item_opts: set_item_opts, item_opts_on, menu_item_opts(3X) CHAR
 /set_item_init, item_init, set_item_term, item_term,/ menu_hook(3X) CHAR
 associate/ menu_item_userptr: set_item_userptr, item_userptr: menu_item_userptr(3X) CHAR
 and get menus/ menu_item_value: set_item_value, item_value: set menu_item_value(3X) CHAR

interval timer. `getitimer`,
 `_longjmp`, `sigsetjmp`,/
`siglongjmp`:/ `setjmp`, `longjmp`,
 encryption. `crypt`,
 a stream. `setbuffer`,
buffering to/ `setbuf`, `setbuffer`,
 program's locale.
 `syslog`, `openlog`, `closelog`,
 /`set_field_status`, `field_status`,
 /`set_menu_fore`, `menu_fore`,
`set_menu_back`,/ `menu_attributes`:
 set and get/ `menu_format`:
 /`set_menu_back`, `menu_back`,
 /`set_item_term`, `item_term`,
item_count: connect/ `menu_items`:
 mark string/ `menu_mark`:
 `menu_opts_off`,/ `menu_opts`:
menus/ `set_menu_grey`, `menu_grey`,
set and get menus/ `menu_pattern`:
 /`set_menu_win`, `menu_win`,
 /`set_menu_init`, `menu_init`,
 associate/ `menu_userptr`:
 `set_menu_sub`,/ `menu_win`:

 /`getnetbyaddr`, `getnetbyname`,
 pagination. `form_new_page`:
`panel_userptr`:/ `panel_userptr`:

 scheduling/ `getpriority`,
protocol entry. `getprotobyname`,
`getpwent`, `getpwuid`, `getpwnam`,
 group IDs.
 user IDs.
system resource/ `getrlimit`,
 `sigismember`: manipulate
 message delivery. `dbon`:
/`idlok`, `idcok` `immedok`, `leaveok`,
/`getservbyport`, `getservbyname`,

 on sockets. `getsockopt`,
`lckpwdf`,/ `getspent`, `getspnam`,
 `random`, `random`, `initstate`,
 /`resetty`, `save tty`, `getsyx`,
/`newterm`, `endwin`, `isendwin`,
 `curst_terminfo`: `setupterm`,

`setitimer`: get/set value of `getitimer`(3C) PRM
`setjmp`, `longjmp`: non-local goto. `setjmp`(3C) PRM
`setjmp`, `longjmp`, `_setjmp`, `setjmp`(3) BSD
`_setjmp`, `_longjmp`, `sigsetjmp`, `setjmp`(3) BSD
`setkey`, `encrypt`: generate `crypt`(3C) PRM
`setlinebuf`: assign buffering to `setbuffer`(3S) BSD
`setlinebuf`, `setvbuf`: assign `setbuf`(3S) BSD
`setlocale`: modify and query a `setlocale`(3C) PRM
`setlogmask`: control system log. `syslog`(3) BSD
`set_max_field`: set and get forms/ ... `form_field_buffer`(3X) CHAR
`set_menu_back`, `menu_back`,/ `menu_attributes`(3X) CHAR
`set_menu_fore`, `menu_fore`, `menu_attributes`(3X) CHAR
`set_menu_format`, `menu_format`: `menu_format`(3X) CHAR
`set_menu_grey`, `menu_grey`,/ `menu_attributes`(3X) CHAR
`set_menu_init`, `menu_init`,/ `menu_hook`(3X) CHAR
`set_menu_items`, `menu_items`, `menu_items`(3X) CHAR
`set_menu_mark`, `menu_mark`: menus `menu_mark`(3X) CHAR
`set_menu_opts`, `menu_opts_on`, `menu_opts`(3X) CHAR
`set_menu_pad`, `menu_pad`: control ... `menu_attributes`(3X) CHAR
`set_menu_pattern`, `menu_pattern`: `menu_pattern`(3X) CHAR
`set_menu_sub`, `menu_sub`,/ `menu_win`(3X) CHAR
`set_menu_term`, `menu_term`: assign/ `menu_hook`(3X) CHAR
`set_menu_userptr`, `menu_userptr`: `menu_userptr`(3X) CHAR
`set_menu_win`, `menu_win`, `menu_win`(3X) CHAR
`setmnt`: establish mount table. `setmnt`(1M) SARM
`setnetent`, `endnetent`: get/ `getnetent`(3N) NI
`set_new_page`, `new_page`: forms `form_new_page`(3X) CHAR
`set_panel_userptr`, `panel_userptr`(3X) CHAR
`setpgid`: set process group ID. `setpgid`(2) PRM
`setpgrp`: set process group ID. `setpgrp`(2) PRM
`setpriority`: get/set program `getpriority`(3) BSD
`setprotoent`, `endprotoent`: get `getprotoent`(3N) NI
`setpwent`, `endpwent`, `fgetpwent`:/ `getpwent`(3C) PRM
`setregid`: set real and effective `setregid`(3) BSD
`setreuid`: set real and effective `setreuid`(3) BSD
`setrlimit`: control maximum `getrlimit`(2) PRM
sets of signals. `sigdelset`, `sigsetops`(3C) PRM
sets target for front panel `dbon`(1M) MBRMAN
`setscreg`, `wsetscreg`, `scrollok`,/ `curs_outopts`(3X) CHAR
`setservent`, `endservent`: get/ `getservent`(3N) NI
`setsid`: set session ID. `setsid`(2) PRM
`setsockopt`: get and set options `getsockopt`(3N) NI
`setspent`, `endspent`, `fgetspent`, `getspent`(3C) PRM
`setstate`: better random number/ `random`(3) BSD
`setsyx`, `ripoffline`, `curs_set`,/ `curs_kernel`(3X) CHAR
`set_term`, `delscreen`: curses/ `curs_initscr`(3X) CHAR
`setterm`, `set_curterm`,/ `curs_terminfo`(3X) CHAR

modification dates of files. `settime`: change the access and `settime(1)` XNX
 date and time. `gettimeofday`, `settimeofday`: get or set the `gettimeofday(3)` BSD
 date and time. `gettimeofday`, `settimeofday`: get or set the `gettimeofday(3C)` PRM
 request. `dma_swsetup`: Setting a channel for software `dma_swsetup(D3D)` DDRM
`tset`: provide information for setting terminal modes. `tset(1)` XNX
 login time. `profile`: setting up an environment at `profile(4)` SARM
 TTY/ `sttydefs`: maintain line settings and hunt sequences for `sttydefs(1M)` SARM
 /file contains terminal line settings information for `ttymon`. `tttydefs(4)` NI
 /set_current_item, current_item, menu_item_current(3X) CHAR
 group IDs. `setuid`, `setgid`: set user and `setuid(2)` PRM
 information. `setuname`: changes machine `setuname(1M)` SARM
 first user. `setup`: initialize system for `setup(1M)` SARM
`del_curterm`,/ `cursor_terminfo`: `setupterm`, `setterm`, `set_curterm`, `cursor_terminfo(3X)` CHAR
 legal user/ `getusershell`, `setusershell`, `endusershell`: get `getusershell(3)` BSD
 /`getutid`, `getutline`, `pututline`, `setutent`, `endutent`, `utmpname`:/ `getut(3C)` PRM
 /`getutxline`, `pututxline`, `setutxent`, `endutxent`, `utmpxname`:/ `getutx(3C)` PRM
`setvbuf`, `setbuffer`, `setlinebuf`, `setvbuf`: assign buffering to a/ `setvbuf(3S)` BSD
 stream. `setvbuf`, `setvbuf`: assign buffering to a `setvbuf(3S)` PRM
 paste: merge same lines of several files or subsequent/ `paste(1)` URM
 addseverity: build a list of severity levels for an/ `addseverity(3C)` PRM
 /`fconvert`, `gconvert`, `seconvert`, `sfconvert`: output/ `econvert(3)` BSD
 /`gconvert`, `seconvert`, `sfconvert`, `sfsys`: file format. `sfsys(4)` SARM
 in a machine-independent/ `sputl`, `sgconvert`: output conversion. `econvert(3)` BSD
 loaders. `sgctl`: access long integer data `sputl(3X)` PRM
 standard, job control, and/ `sgib`: install MULTIBUS bootstrap `sgib(1M)` MBRMAN
 shadow: `sh`, `jsh`, `rsh`: shell, the `sh(1)` URM
 /`lckpwwdf`, `ulckpwwdf`: manipulate shadow password file. `shadow(4)` SARM
`putspent`: write shadow password file entry. `getspent(3C)` PRM
 shadow password file entry. `putspent(3C)` PRM
 shadow: shadow password file. `shadow(4)` SARM
 available for mounting by/ `share`: make local NFS resource `share(1M)` NUAG
 available for mounting by/ `share`: make local NFS resource `share(1M)` SARM
 available for mounting by/ `share`: make local resource `share(1M)` NUAG
 available for mounting by/ `share`: make local resource `share(1M)` SARM
 available for mounting by/ `share`: make local RFS resource `share(1M)` NUAG
 available for mounting by/ `share`: make local RFS resource `share(1M)` SARM
`shareall`, `unshareall`: `share`, `unshare multiple`/ `shareall(1M)` NUAG
`shareall`, `unshareall`: `share`, `unshare multiple`/ `shareall(1M)` SARM
`unshare multiple resources`. `shareall`, `unshareall`: `share`, `shareall(1M)` NUAG
`unshare multiple resources`. `shareall`, `unshareall`: `share`, `shareall(1M)` SARM
 allocate a message block using a shared buffer. `esballoc`: `esballoc(D3DK)` DDRM
`sdgetv`: synchronize shared data access. `sdgetv(2)` XNX
`sdlleave`: synchronize access to a shared data segment. `sdenter`, `sdenter(2)` XNX
`sdfree`: attach and detach a shared data segment. `sdget`, `sdget(2)` XNX
`sharetab`: shared file system table. `sharetab(4)` NUAG
`sharetab`: shared file system table. `sharetab(4)` SARM
 operations. `shmctl`: shared memory control `shmctl(2)` PRM

message queue, semaphore set, or
 shmop: shmat, shmdt:
 identifier. shmget: get
 dlclose: close a
 dlopen: open a
 get the address of a symbol in
 table.
 table.
 rfstart: start Remote File
 rfstart: start Remote File
 rfudaemon: Remote File
 rfudaemon: Remote File
 rfadmin: Remote File
 rfadmin: Remote File
 names. dname: print Remote File
 names. dname: print Remote File
 rfstop: stop the Remote File
 rfstop: stop the Remote File
 rfpaswd: change Remote File
 rfpaswd: change Remote File
 rfmaster: Remote File
 rfmaster: Remote File
 nsquery: Remote File
 nsquery: Remote File
 script. rfadmin: Remote File
 script. rfadmin: Remote File
 file containing commands for
 file containing commands for
 mount, unmount Remote File
 mount, unmount Remote File
 idload: Remote File
 idload: Remote File
 rsh: remote
 rsh: remote
 shell: run a command using
 system: issue a
 C-like syntax. csh:
 gmatch:
 shl:
 /shutacct, startup, turnacct:
 shell.
 xinstall: XENIX installation
 Remote File Sharing notification
 Remote File Sharing notification
 rshd: remote
 rshd: remote
 control, and/ sh, jsh, rsh:
 shared memory ID. /remove a ipcrm(1) URM
 shared memory operations. shmop(2) PRM
 shared memory segment shmget(2) PRM
 shared object. dlclose(3X) PRM
 shared object. dlopen(3X) PRM
 shared object. dlsym: dlsym(3X) PRM
 sharetab: shared file system sharetab(4) NUAG
 sharetab: shared file system sharetab(4) SARM
 Sharing. rfstart(1M) NUAG
 Sharing. rfstart(1M) SARM
 Sharing daemon process. rfudaemon(1M) NUAG
 Sharing daemon process. rfudaemon(1M) SARM
 Sharing domain administration. rfadmin(1M) NUAG
 Sharing domain administration. rfadmin(1M) SARM
 Sharing domain and network dname(1M) NUAG
 Sharing domain and network dname(1M) SARM
 Sharing environment. rfstop(1M) NUAG
 Sharing environment. rfstop(1M) SARM
 Sharing host password. rfpaswd(1M) NUAG
 Sharing host password. rfpaswd(1M) SARM
 Sharing name server master file. rfmaster(4) NUAG
 Sharing name server master file. rfmaster(4) SARM
 Sharing name server query. nsquery(1M) NUAG
 Sharing name server query. nsquery(1M) SARM
 Sharing notification shell rfadmin(1M) NUAG
 Sharing notification shell rfadmin(1M) SARM
 sharing resources. dfstab: dfstab(4) NUAG
 sharing resources. dfstab: dfstab(4) SARM
 Sharing resources. /rumountall: rmountall(1M) NUAG
 Sharing resources. /rumountall: rmountall(1M) SARM
 Sharing user and group mapping. idload(1M) NUAG
 Sharing user and group mapping. idload(1M) SARM
 shell. rsh(1) NUAG
 shell. rsh(1) URM
 shell. shell(1F) CHAR
 shell command. system(3S) PRM
 shell command interpreter with a csh(1) URM
 shell global pattern matching. gmatch(3G) PRM
 shell layer manager. shl(1) URM
 shell procedures for accounting. acctsh(1M) SARM
 shell: run a command using shell(1F) CHAR
 shell script. xinstall(1M) SARM
 shell script. rfadmin: rfadmin(1M) NUAG
 shell script. rfadmin: rfadmin(1M) SARM
 shell server. rshd(1M) NUAG
 shell server. rshd(1M) SARM
 shell, the standard, job sh(1) URM

endusershell: get legal user
 operations. shmop:
 operations.
 shmop: shmat,
 segment identifier.
 memory operations.
 nap: suspend execution for a
 port. inw: read a 16 bit
 outw: write a 16-bit
 relogin: rename login entry to
 relogin: rename login entry to
 machines. ruptime:
 been up. uptime:
 netstat:
 netstat:
 in reverse order. lastcomm:
 panel_hidden:/ panel_show:
 connection. shutdown:
 state. shutdown:
 /prdaily, prtacct, runacct,
 at a given time.
 full-duplex connection.
 change system state.
 system shutdown. halt:
 the driver when the system
 library routines for client
 /library routines for client
 /library routines for server
 sdiff: print file differences
 management.
 sigemptyset, sigfillset,
 alternate stack context.
 signals.
 /sigfillset, sigaddset,
 sigaddset, sigdelset,/
 sigdelset,/ sigemptyset,
 signal handling for specific
 specific SIGFPE codes.
 sigpause:/ signal, sigset,
 /sigset, sighold, sigrelse,
 information.
 information.
 interrupt system calls.
 signals. /sigaddset, sigdelset,
 with signal state. sigsetjmp,
 shells. /setusershell, getusershell(3) BSD
 shl: shell layer manager. shl(1) URM
 shmat, shmdt: shared memory shmop(2) PRM
 shmctl: shared memory control shmctl(2) PRM
 shmdt: shared memory operations. shmop(2) PRM
 shmget: get shared memory shmget(2) PRM
 shmop: shmat, shmdt: shared shmop(2) PRM
 short interval. nap(2) XNX
 short word from a 16-bit I/O inw(D3D) DDRM
 short word to a 16-bit I/O port. outw(D3D) DDRM
 show current layer. relogin(1M) SARM
 show current layer. relogin(1M) URM
 show host status of local ruptime(1) NUAG
 show host status of local ruptime(1) URM
 show how long the system has uptime(1) BSD
 show network status. netstat(1M) NUAG
 show network status. netstat(1M) SARM
 show the last commands executed, lastcomm(1) BSD
 show_panel, hide_panel, panel_show(3X) CHAR
 shut down part of a full-duplex shutdown(3N) NI
 shut down system, change system shutdown(1M) SARM
 shutacct, startup, turnacct:/ acctsh(1M) SARM
 shutdown: close down the system shutdown(1M) BSD
 shutdown: shut down part of a shutdown(3N) NI
 shutdown: shut down system, shutdown(1M) SARM
 shutdown the driver when the halt(D2D) DDRM
 shutdown. halt: shutdown halt(D2D) DDRM
 side calls. /rpc_call: rpc_cnt_calls(3N) NI
 side remote procedure call/ rpc_cnt_auth(3N) NI
 side remote procedure call/ rpc_svc_err(3N) NI
 side-by-side. sdiff(1) URM
 sigaction: detailed signal sigaction(2) PRM
 sigaddset, sigdelset,/ sigsetops(3C) PRM
 sigaltstack: set or get signal sigaltstack(2) PRM
 sigblock, sigmask: block sigblock(3) BSD
 sigdelset, sigismember:/ sigsetops(3C) PRM
 sigemptyset, sigfillset, sigsetops(3C) PRM
 sigfillset, sigaddset, sigsetops(3C) PRM
 SIGFPE codes. sigfpe: sigfpe(3) BSD
 sigfpe: signal handling for sigfpe(3) BSD
 sighold, sigrelse, sigignore, signal(2) PRM
 sigignore, sigpause: simplified/ signal(2) PRM
 siginfo: signal generation siginfo(5) PRM
 siginfo: signal generation siginfo(5) SARM
 siginterrupt: allow signals to siginterrupt(3) BSD
 sigismember: manipulate sets of sigsetops(3C) PRM
 siglongjmp: a non-local goto sigsetjmp(3C) PRM

/_setjmp, _longjmp, sigsetjmp, siglongjmp: non-local goto. setjmp(3) BSD
 sigblock, sigmask: block signals. sigblock(3) BSD
 login: signal on. login(1) URM
 pause: suspend process until signal. pause(2) PRM
 semaphore. sigsem: signal a process waiting on a sigsem(2) XNX
 generate an abnormal termination signal. abort: abort(3C) PRM
 microseconds. ualarm: schedule signal after interval in ualarm(3) BSD
 sigaltstack: set or get signal alternate stack context. sigaltstack(2) PRM
 signal: base signals. signal(5) PRM
 signal: base signals. signal(5) SARM
 signal facilities. signal(3) BSD
 signal facilities. sigvec(3) BSD
 sigvec: software signal generation information. siginfo(5) PRM
 siginfo: signal generation information. siginfo(5) SARM
 siginfo: signal handling for specific sigfpe(3) BSD
 SIGFPE codes. sigfpe: signal. /install a signal sigsuspend(2) PRM
 mask and suspend process until signal management. sigaction(2) PRM
 sigaction: detailed signal management. /sigrelse, signal(2) PRM
 sigignore, sigpause: simplified signal mask. sigprocmask(2) PRM
 sigprocmask: change or examine signal mask. sigsetmask(3) BSD
 sigsetmask: set current signal mask and suspend process sigsuspend(2) PRM
 until/ sigsuspend: install a signal(3) BSD
 psignal, sys_siglist: system signal messages. psignal(3C) PRM
 psignal, psiginfo: system signal messages. psignal(3C) PRM
 process group. signal: send a signal to a signal(D3D) DDRM
 sigrelse, sigignore, sigpause:/ signal, sigset, sighold, signal(2) PRM
 signal facilities. signal: simplified software signal(3) BSD
 sigstack: set and/or get signal stack context. sigstack(3) BSD
 a non-local goto with signal state. /siglongjmp: sigsetjmp(3C) PRM
 psignal: send signal to a process. psignal(D3D) DDRM
 killpg: send signal to a process group. killpg(3) BSD
 signal: send a signal to a process group. signal(D3D) DDRM
 of processes. kill: send a signal to a process or a group kill(2) PRM
 of/ sigsend, sigsendset: send a signal to a process or a group sigsend(2) PRM
 raise: send signal to program. raise(3C) PRM
 sigblock, sigmask: block signals. sigblock(3) BSD
 signal: base signals. signal(5) PRM
 signal: base signals. signal(5) SARM
 signals. signal(5) SARM
 signals. ssignal(3C) PRM
 ssignal, gsignal: software signals. truss(1) URM
 truss: trace system calls and signals and wait for interrupt. sigpause(3) BSD
 /automatically release blocked signals. /sigaddset, sigdelset, sigsetops(3C) PRM
 sigismember: manipulate sets of signals that are blocked and sigpending(2) PRM
 pending. sigpending: examine signals to interrupt system siginterrupt(3) BSD
 calls. siginterrupt: allow sigpause: automatically release sigpause(3) BSD
 blocked signals and wait for/ sigpause: simplified signal/ signal(2) PRM
 /sighold, sigrelse, sigignore, sigpending: examine signals that sigpending(2) PRM
 are blocked and pending. sigprocmask: change or examine sigprocmask(2) PRM
 signal mask.

signal, sigset, sighold, on a semaphore.	sigrelse, sigignore, sigpause:/	signal(2) PRM
signal to a process or a group/ process or a group of/ sigsend, sigignore, sigpause:/ signal, non-local goto with signal/ /longjmp, _setjmp, _longjmp, mask.	sigsem: signal a process waiting	sigsem(2) XNX
stack context.	sigsend, sigsendset: send a	sigsend(2) PRM
mask and suspend process until/ facilities.	sigsendset: send a signal to a	sigsend(2) PRM
lex: generate programs for	sigset, sighold, sigrelse,	signal(2) PRM
/SMTP mail to a remote host using	sigsetjmp, siglongjmp: a	sigsetjmp(3C) PRM
rand, srand:	sigsetjmp, siglongjmp: non-local/	setjmp(3) BSD
rand, srand:	sigsetmask: set current signal	sigsetmask(3) BSD
fmt:	sigstack: set and/or get signal	sigstack(3) BSD
/sigrelse, sigignore, sigpause: facilities. signal:	sigsuspend: install a signal	sigsuspend(2) PRM
asin, asinf, acos, acosf,/ trig: asin, asinf, acos,/ trig: sin, floating_to_decimal:/	sigvec: software signal	sigvec(3) BSD
sulogin: access	simple lexical tasks.	lex(1) PRM
tanhf, asinh, acosh, atanh:/ asinh, acosh, atanh:/ sinh, getpagesize: get system page get descriptor table	Simple Mail Transfer Protocol.	smtp(1M) SARM
ptob: convert size in pages to (round down). btop: convert (round up). btopr: convert btop: convert size in bytes to btopr: convert size in bytes to ptob: convert	simple random number generator.	rand(3C) BSD
chsize: change the	simple random-number generator.	rand(3C) PRM
pagesize: display the	simple text formatters.	fmt(1) URM
/elf32_fsize: return the	simplified signal management.	signal(2) PRM
jwin: print	simplified software signal	signal(3) BSD
size: return	sin, sinf, cos, cosf, tan, tanf,	trig(3M) PRM
bytes of object files.	sinf, cos, cosf, tan, tanf,	trig(3M) PRM
device.	single_to_decimal,/	floating_to_decimal(3) BSD
size: print section	single-user mode.	sulogin(1M) SARM
grantpt: grant access to the	sinh, sinhf, cosh, coshf, tanh,	sinh(3M) PRM
ptsname: get name of the	sinhf, cosh, coshf, tanh, tanhf,	sinh(3M) PRM
interval.	size.	getpagesize(3) BSD
interval.	size. getdtablesize:	getdtablesize(3) BSD
interval.	size in bytes.	ptob(D3DK) DDRM
pending execution of an event.	size in bytes to size in pages	btop(D3DK) DDRM
	size in bytes to size in pages	btopr(D3DK) DDRM
	size in pages (round down).	btop(D3DK) DDRM
	size in pages (round up).	btopr(D3DK) DDRM
	size in pages to size in bytes.	ptob(D3DK) DDRM
	size of a file.	chsize(2) XNX
	size of a page of memory.	pagesize(1) BSD
	size of an object file type.	elf_fsize(3E) PRM
	size of layer.	jwin(1) URM
	size of logical device.	size(D2DK) DDRM
	size: print section sizes in	size(1) PRM
	size: return size of logical	size(D2DK) DDRM
	sizes in bytes of object files.	size(1) PRM
	slave pseudo-terminal device.	grantpt(3C) STRM
	slave pseudo-terminal device.	ptsname(3C) STRM
	sleep: suspend execution for	sleep(3) BSD
	sleep: suspend execution for	sleep(3C) PRM
	sleep: suspend execution for an	sleep(1) URM
	sleep: suspend process activity	sleep(D3DK) DDRM
	slink: streams linker.	slink(1M) NUAG

/slk_attron, slk_attrset,
 /slk_restore, slk_touch,
 soft/ /slk_touch, slk_attron,
 /slk_noutrefresh, slk_label,
 slk_noutrefresh,/ curs_slk:
 slk_refresh, slk_noutrefresh,
 /slk_init, slk_set, slk_refresh,
 curs_slk: slk_init, slk_set,
 /slk_label, slk_clear,
 curs_slk: slk_init,
 /slk_clear, slk_restore,
 board types in the designated
 current user. ttyslot: find the
 resets the processor in a given
 tosmtp: send mail to
 receive RFC822 mail from
 process messages queued in the
 Simple Mail Transfer/ smtp: send
 smtpd: receive incoming
 host using Simple Mail Transfer/
 queue mail for delivery by
 messages.
 by SMTP.
 queued in the SMTP mail queue.
 soelim: resolve and eliminate
 accept: accept a connection on a
 bind: bind a name to a
 initiate a connection on a
 communication.
 listen for connections on a
 getsockname: get
 receive a message from a
 sendmsg: send a message from a
 connected sockets.
 get and set options on
 create a pair of connected
 ioctls that operate directly on
 directly on sockets.
 .so requests from nroff or/
 /slk_attrset, slk_attroff: curses
 removef: remove a file from
 depend:
 installf: add a file to the
 interface. lo:
 interface. lo:
 pkginfo: display
 slink: streams linker. slink(1M) SARM
 slk_attroff: curses soft label/ curs_slk(3X) CHAR
 slk_attron, slk_attrset,/ curs_slk(3X) CHAR
 slk_attrset, slk_attroff: curses curs_slk(3X) CHAR
 slk_clear, slk_restore,/ curs_slk(3X) CHAR
 slk_init, slk_set, slk_refresh, curs_slk(3X) CHAR
 slk_label, slk_clear,/ /slk_set, curs_slk(3X) CHAR
 slk_noutrefresh, slk_label,/ curs_slk(3X) CHAR
 slk_refresh, slk_noutrefresh,/ curs_slk(3X) CHAR
 slk_restore, slk_touch,/ curs_slk(3X) CHAR
 slk_set, slk_refresh,/ curs_slk(3X) CHAR
 slk_touch, slk_attron,/ curs_slk(3X) CHAR
 slot. /checks for certain ics_agent_cmp(D3D) DDRM
 slot in the utmp file of the ttyslot(3C) PRM
 slot. reset: reset(1M) MBRMAN
 SMTP. tosmtp(1M) SARM
 SMTP. fromsmtp: fromsmtp(1M) SARM
 SMTP mail queue. smtpsched: smtpsched(1M) SARM
 SMTP mail to a remote host using smtp(1M) SARM
 SMTP messages. smtpd(1M) SARM
 smtp: send SMTP mail to a remote smtp(1M) SARM
 SMTP. smtpqer: smtpqer(1M) SARM
 smtpd: receive incoming SMTP smtpd(1M) SARM
 smtpqer: queue mail for delivery smtpqer(1M) SARM
 smtpsched: process messages smtpsched(1M) SARM
 .so requests from nroff or troff/ soelim(1) BSD
 socket. accept(3N) NI
 socket. bind(3N) NI
 socket. connect: connect(3N) NI
 socket: create an endpoint for socket(3N) NI
 socket. listen: listen(3N) NI
 socket name. getsockname(3N) NI
 socket. /recvfrom, recvmsg: recv(3N) NI
 socket. send, sendto, send(3N) NI
 socketpair: create a pair of socketpair(3N) NI
 sockets. /setsockopt: getsockopt(3N) NI
 sockets. socketpair: socketpair(3N) NI
 sockets. sockio: sockio(7) NI
 sockio: ioctls that operate sockio(7) NI
 soelim: resolve and eliminate soelim(1) BSD
 soft label routines. curs_slk(3X) CHAR
 software database. removef(1M) SS
 software dependencies files. depend(4) SS
 software installation database. installf(1M) SS
 software loopback network lo(7) NUAG
 software loopback network lo(7) SARM
 software package information. pkginfo(1) SARM

pkginfo: display software package information. pkginfo(1) SS
 pkgadd: transfer software package to the system. pkgadd(1M) SARM
 pkgadd: transfer software package to the system. pkgadd(1M) SS
 Setting a channel for software request. dma_swsetup: dma_swsetup(D3D) DDRM
 signal: simplified software signal facilities. signal(3) BSD
 sigvec: software signal facilities. sigvec(3) BSD
 ssignal, gsignal: software signals. ssignal(3C) PRM
 dma_swstart: to start a Software-programmed Transfer. dma_swstart(D3D) DDRM
 message to be sent to initiate a solicited. /constructs a mps_mk_sol(D3D) DDRM
 mps_AMPreceive_frag: receives solicited data in fragments/
 mps_AMPreceive_frag(D3D) DDRM
 to an. mps_AMPreceive: receives solicited data that corresponds ... mps_AMPreceive(D3D) DDRM
 of any. mps_AMPsend_data: sends solicited data that is not part
 mps_AMPsend_data(D3D) DDRM
 /get data length for a solicited reply.. mps_get_reply_len(D3D) DDRM
 qsort: quicker sort. qsort(3C) PRM
 tsort: topological sort. tsort(1) PRM
 sortbib: sort a bibliographic database. sortbib(1) BSD
 sort: sort and/or merge files. sort(1) URM
 sort: sort and/or merge files. sort(1) URM
 sortbib: sort a bibliographic sortbib(1) BSD
 database. comm(1) URM
 or reject lines common to two sorted files. comm: select look(1) BSD
 system dictionary or lines in a sorted list. /find words in the bsearch(3C) PRM
 bsearch: binary search a source. /resolve and remove undef(1) BSD
 ifdef'ed lines from C program space. idspace(1M) SARM
 idspace: investigates free space. swapctl(2) PRM
 swapctl: manage swap space allocation. brk(2) PRM
 brk, sbrk: change data segment space back into a private space rmfree(D3DK) DDRM
 management map. rmfree: free space. bp_mapin: bp_mapin(D3DK) DDRM
 allocate virtual address space. bp_mapout: bp_mapout(D3DK) DDRM
 deallocate virtual address Space device driver. ics(1M) MBRMAN
 ics: Interconnect Space device driver. mps(1M) MBRMAN
 mps: Message Passing space: disk space requirement space(4) SS
 file. rmalloc(D3DK) DDRM
 management/ rmalloc: allocate space from a private space kmem_alloc(D3DK) DDRM
 kmem_alloc: allocate space from kernel free memory. kmem_zalloc(D3DK) DDRM
 kmem_zalloc: allocate and clear space from kernel free memory. icsrd(1M) MBRMAN
 access MULTIBUS II interconnect space. /icsgetrec: utilities to rminit(D3DK) DDRM
 rminit: initialize a private space management map. rmalloc: rmalloc(D3DK) DDRM
 allocate space from a private space management map. rmfree: rmfree(D3DK) DDRM
 free space back into a private space. mlockall, munlockall: mlockall(3C) PRM
 lock or unlock address space of. /finds a specific ics_find_rec(1M) MBRMAN
 record in the interconnect space on file systems. df(1) BSD
 df: report free disk space on ufs file systems. df(1M) SARM
 df (ufs): report free disk space requirement file. space(4) SS
 space: disk space. segmap: segmap(D2K) DDRM
 map device memory into user

terminal. ct:	spawn login to a remote	ct(1C) URM
memory efficient way. vfork:	spawn new process in a virtual	vfork(2) PRM
for eqn. eqnchar:	special character definitions	eqnchar(7) BSD
mkfifo: make FIFO	special file.	mkfifo(1M) SARM
mknod: make a	special file.	mknod(1M) SARM
intro: introduction to	special files.	intro(7) SARM
mknod: make a directory, or a	special or ordinary file.	mknod(2) PRM
mknod: make a directory, or a	special or ordinary file.	mknod(2) XNX
indicator: display application	specific alarms and/or the/	indicator(1F) CHAR
mb2a_getinfo: Gets	specific asynchronous endpoint/	mb2a_getinfo(3tai) TAI
sysi86: machine	specific functions.	sysi86(2) PRM
package. custom: install	specific portions of a UNIX	custom(1M) SARM
package. custom: install	specific portions of a UNIX	custom(1) XNX
ics_find_rec: finds a	specific record in the/	ics_find_rec(1M) MBRMAN
sigfpe: signal handling for	specific SIGFPE codes.	sigfpe(3) BSD
strftime: language	specific strings.	strftime(4) PRM
strftime: language	specific strings.	strftime(4) SARM
mb2s_getinfo: Gets	specific synchronous endpoint/	mb2s_getinfo(3tai) TAI
fspec: format	specification in text files.	fspec(4) SARM
idmkinit: reads files containing	specifications.	idmkinit(1M) SARM
idmknod: removes nodes and reads	specifications of nodes.	idmknod(1M) SARM
drv_usecwait: busy-wait for	specified interval.	drv_usecwait(D3DK) DDRM
/execute a function after a	specified length of time.	timeout(D3DK) DDRM
ftruncate: set a file to a	specified length. truncate,	truncate(3C) PRM
ics_rdwr: reads or writes a	specified number of /	ics_rdwr(D3D) DDRM
/delay process execution for a	specified number of clock ticks.	delay(D3DK) DDRM
/test for flow control in	specified priority band.	bcanput(D3DK) DDRM
flushband: flush messages for a	specified priority band.	flushband(D3DK) DDRM
in. /writes a value into the	specified register of the board	ics_write(D3D) DDRM
rwall: write to	specified remote machines.	rwall(3N) NI
the jobs queued to run at	specified times. atq: display	atq(1) URM
free_rtn: structure that	specifies a driver's message/	free_rtn(D4DK) DDRM
getty: set terminal type, modes,	speed, and line discipline.	getty(1M) SARM
hashcheck: find spelling/	spell, hashmake, spellin,	spell(1) URM
spelling/ spell, hashmake,	spellin, hashcheck: find	spell(1) URM
spellin, hashcheck: find	spelling errors. /hashmake,	spell(1) URM
	spl: block/allow interrupts.	spl(D3D) DDRM
	split.	csplit(1) URM
csplit: context	split a file into pieces.	split(1) URM
split:	split buffer into fields.	bufsplit(3G) PRM
bufsplit:	split: split a file into pieces.	split(1) URM
	spool directory clean-up.	uucleanup(1M) SARM
uucleanup: uucp	spooled by at or batch.	atrm(1) URM
atrm: remove jobs	spray packets.	spray(1M) NI
spray:	spray packets.	spray(1M) NUAG
spray:	spray: scatter data in order to	spray(3N) NI
check the network.	spray server.	sprayd(1M) NI
rpc.sprayd:		

rpc.sprayd: spray server. sprayd(1M) NUAG
 spray: spray packets. spray(1M) NI
 spray: spray packets. spray(1M) NUAG
 printf, fprintf,
 vsprintf:/ printf, fprintf,
 integer data in a/
 logf, log10, log10f, pow, powf,
 /log10, log10f, pow, powf, sqrt,
 exponential, logarithm, power,
 generator. rand,
 generator. rand,
 /nrand48, mrand48, jrand48,
 better random number/ random,
 search for a text string in/
 window. curs_scroll: scroll,

 scanf, fscanf,
 signals.
 sigstack: set and/or get signal
 set or get signal alternate
 package. stdio:
 communication/ stdipc: ftok:
 sh, jsh, rsh: shell, the
 discipline module. ldterm:
 ksh, rksh: KornShell, a
 /wattronn, attrset, wattrset,
 /wattrset, standend, wstandend,
 Transfer. dma_swstart: to
 start:
 rfstart:
 rfstart:

 init_color,/ curs_color:
 service daemon. bootserver:
 and/ lpsched, lpshut, lpmove:
 /prtacct, runacct, shutacct,
 system call.
 system call.
 system call.
 status.
 status.
 stat: data returned by
 stat: data returned by
 stat: data returned by

 ismpx: return windowing terminal
 t_getstate: get the current

sprayd(1M) NUAG
 spray(1M) NI
 spray(1M) NUAG
 printf(3S) PRM
 printf(3S) BSD
 sputl(3X) PRM
 exp(3M) PRM
 exp(3M) PRM
 exp(3M) PRM
 rand(3C) BSD
 rand(3C) PRM
 drand48(3C) PRM
 random(3) BSD
 srctxt(1) URM
 curs_scroll(3X) CHAR
 srv(D2DK) DDRM
 scanf(3S) PRM
 ssignal(3C) PRM
 sigstack(3) BSD
 sigaltstack(2) PRM
 stdio(3S) PRM
 stdipc(3C) PRM
 sh(1) URM
 ldterm(7) STRM
 ksh(1) URM
 curs_attr(3X) CHAR
 curs_attr(3X) CHAR
 dma_swstart(D3D) DDRM
 start(D2D) DDRM
 rfstart(1M) NUAG
 rfstart(1M) SARM
 start(D2D) DDRM
 curs_color(3X) CHAR
 bootserver(1M) MBRMAN
 lpsched(1M) SARM
 acctsh(1M) SARM
 stat(4) XNX
 stat(5) PRM
 stat(5) SARM
 stat(2) PRM
 stat(2) XNX
 stat(4) XNX
 stat(5) PRM
 stat(5) SARM
 statd(1M) NUAG
 ismpx(1) URM
 t_getstate(3N) NI

shut down system, change system state. shutdown: shutdown(1M) SARM
a non-local goto with signal state. sigsetjmp, siglongjmp: sigsetjmp(3C) PRM
wait for child process to change state. waitid: waitid(2) PRM
wait for child process to change state. waitpid: waitpid(2) PRM
ustat: get file system statistics. ustat(2) PRM
xts: extract and print xt driver statistics. xts(1M) SARM
sendmail. mailstats: print statistics collected by mailstats(1M) BSD
/(generic): list file names and statistics for a file system. ff(1M) SARM
ff (ufs): list file names and statistics for a ufs file/ ff(1M) SARM
netstat: show network status. netstat(1M) NUAG
netstat: show network status. netstat(1M) SARM
ps: report process status. ps(1) URM
stat, lstat, fstat: get file status. stat(2) PRM
stat, lstat, fstat: get file status. stat(2) XNX
wstat: wait status. wstat(5) PRM
feof, clearerr, fileno: stream status inquiries. ferror, ferror(3S) PRM
uustat: uucp status inquiry and job control. uustat(1C) URM
statd: network status monitor. statd(1M) NUAG
bkstatus: display the status of backup operations. bkstatus(1M) SARM
ps: display the status of current processes. ps(1) BSD
runtime: show host status of local machines. runtime(1) NUAG
runtime: show host status of local machines. runtime(1) URM
/print information about the status of the LP print service. lpstat(1) URM
communication facilities status. /report inter-process ipc(1) URM
rwhod, in.rwhod: system status server. rwhod(1M) NUAG
rwhod, in.rwhod: system status server. rwhod(1M) SARM
system information. statvfs, fstatvfs: get file statvfs(2) PRM
list. stdarg: handle variable argument stdarg(5) PRM
fmtmsg: display a message on stderr or system console. fmtmsg(1) URM
fmtmsg: display a message on stderr or system console. fmtmsg(3C) PRM
input/output package. stdio: standard buffered stdio(3S) PRM
interprocess communication/ stdipc: ftok: standard stdipc(3C) PRM
expression/ regexp: compile step, advance: regular regexp(5) PRM
expression/ regexp: compile step, advance: regular regexp(5) SARM
expression/ regexpr: compile step, advance: regular regexpr(3G) PRM
rc6: run commands performed stime: set time. stime(2) PRM
performed to stop and reboot the operating/ rc6(1M) SARM
the channel. dma_stop: stop DMA activity and release dma_stop(D3D) DDRM
wait: wait for child process to stop or terminate. wait(2) PRM
rc0: run commands performed stop the operating system. rc0(1M) SARM
halt: stop the processor. halt(1M) BSD
environment. rfstop: stop the Remote File Sharing rfstop(1M) NUAG
environment. rfstop: stop the Remote File Sharing rfstop(1M) SARM
wait for process to terminate or stop. /WIFSIGNALED, WIFEXITED: wait(3) BSD
Descriptor. dma_free_buf: Free storage for a DMA Buffer dma_free_buf(D3D) DDRM
dma_get_buf: Allocate storage for a DMA Buffer/ dma_get_buf(D3D) DDRM

dma_free_cb: Free	storage for a DMA Command Block.
dma_get_cb: Allocate dma_free_cb(D3D) DDRM
disk: random access bulk	storage for a DMA Command Block. ... dma_get_cb(D3D) DDRM
synchronize memory with physical	storage medium. disk(7) SARM
DMA. dma_buf: data	storage. msync: msync(3C) PRM
DMA. dma_cb: data	storage structure for I/O using dma_buf(D4D) DDRM
uio(D4DK). iovec: data	storage structure for I/O using dma_cb(D4D) DDRM
dbm, dbminit, dbmclose, fetch,	storage structure for I/O using iovec(D4DK) DDRM
dbm: dbminit, dbmclose, fetch,	store, delete, firstkey,/ dbm(3) NI
keylogin: decrypt and	store, delete, firstkey,/ dbm(3X) BSD
keylogin: decrypt and	store secret key. keylogin(1) NI
script. pkgask:	store secret key. keylogin(1) NUAG
script. pkgask:	stores answers to a request pkgask(1M) SARM
keyserv: server for	stores answers to a request pkgask(1M) SS
keyserv: server for	storing public and private keys. keyserv(1M) NI
string manipulations.	storing public and private keys. keyserv(1M) NUAG
messages.	str: strfind, strrspn, strtrns: str(3G) PRM
messages.	strace: print STREAMS trace strace(1M) SARM
compressing. strccpy: streadd,	strace: print STREAMS trace strace(1M) STRM
operations. string:	strategy: perform block I/O. strategy(D2DK) DDRM
strncmp, strcpy,/ string:	strcadd, strcpy: copy strings, strccpy(3G) PRM
strcpy: copy strings,/	strcasecmp, strncasecmp: string string(3) BSD
File for STREAMS TCP/IP.	strcat, strdup, strncat, strcmp, string(3C) PRM
File for STREAMS TCP/IP.	strccpy: streadd, strcadd, strccpy(3G) PRM
stream configuration.	strcf: STREAMS Configuration strcf(4) NUAG
stream configuration.	strcf: STREAMS Configuration strcf(4) SARM
/strcpy, strncpy, strlen,	strchg, strconf: change or query strchg(1) STRM
cleanup program.	strchg, strconf: change or query strchg(1) URM
cleanup program.	strchr, strchr, strpbrk,/ string(3C) PRM
string: strcat, strdup, strncat,	strclean: STREAMS error logger strclean(1M) SARM
configuration. strchg,	strclean: STREAMS error logger strclean(1M) STRM
configuration. strchg,	strcmp, strncmp, strcpy,/ string(3C) PRM
/strncat, strcmp, strncmp,	strcoll: string collation. strcoll(3C) PRM
/strchr, strpbrk, strspn,	strconf: change or query stream strchg(1) STRM
strncmp,/ string: strcat,	strconf: change or query stream strchg(1) URM
strings, compressing. strccpy:	strcpy, strncpy, strlen, strchr,/ string(3C) PRM
fclose, fflush: close or flush a	strcspn, strtok, strstr: string/ string(3C) PRM
fopen, freopen, fdopen: open a	strdup, strncat, strcmp, string(3C) PRM
fopen, freopen, fdopen: open a	streadd, strcadd, strcpy: copy strccpy(3G) PRM
getmsg: get next message off a	stream. fclose(3S) PRM
getmsg: get next message off a	stream. fopen(3S) BSD
gets, fgets: get a string from a	stream. fopen(3S) PRM
putmsg: send a message on a	stream. getmsg(2) PRM
putmsg: send a message on a	stream. getmsg(2) STRM
	stream. gets(3S) PRM
	stream. putmsg(2) PRM
	stream. putmsg(2) STRM

puts, fputs: put a string on a stream. puts(3S) PRM
 strchg, strconf: change or query stream configuration. strchg(1) STRM
 strchg, strconf: change or query stream configuration. strchg(1) URM
 line discipline for unique stream connections. connld: connld(7) STRM
 for external data representation stream creation. /routines xdr_create(3N) NI
 sed: stream editor. sed(1) URM
 reposition a file pointer in a stream. fseek, rewind, ftell: fseek(3S) PRM
 reposition a file pointer in a stream. fsetpos, fgetpos: fsetpos(3C) PRM
 a get character or word from a stream. /getchar, fgetc, getw: getc(3S) PRM
 qreply: send a message on a stream in the reverse direction. qreply(D3DK) DDRM
 putw: put character or word on a stream. putc, putchar, fputc, putc(3S) PRM
 setvbuf: assign buffering to a stream. setbuf, setbuf(3S) PRM
 setvbuf: assign buffering to a stream. /setbuffer, setlinebuf, setbuf(3S) BSD
 assign buffering to a stream. setbuffer, setlinebuf: setbuffer(3S) BSD
 ferror, feof, clearerr, fileno: stream status inquiries. ferror(3S) PRM
 rexec: return stream to a remote command. rexec(3N) NI
 rexec: return stream to a remote command. rexec(3N) NUAG
 push character back onto input stream. ungetc: ungetc(3S) PRM
 bgets: read stream up to next delimiter. bgets(3G) PRM
 qt: QIC cartridge magnetic tape streamer interface. qt(7) SARM
 commands. streamio: STREAMS ioctl streamio(7) STRM
 sad: STREAMS Administrative Driver. sad(7) STRM
 ttcompat: V7, 4BSD and XENIX STREAMS compatibility module. ttcompat(7) SARM
 STREAMS TCP/IP. strcf: STREAMS Configuration File for strcf(4) NUAG
 STREAMS TCP/IP. strcf: STREAMS Configuration File for strcf(4) SARM
 communications interface STREAMS driver. /asynchronous ports(7) STRM
 any major/minor device pair on a STREAMS driver. clone: open clone(7) STRM
 and limit value/ module_info: STREAMS driver identification module_info(D4DK) DDRM
 structure. streamtab: STREAMS entity declaration streamtab(D4DK) DDRM
 program. strclean: STREAMS error logger cleanup strclean(1M) SARM
 program. strclean: STREAMS error logger cleanup strclean(1M) STRM
 strerr: STREAMS error logger daemon. strerr(1M) SARM
 strerr: STREAMS error logger daemon. strerr(1M) STRM
 tracing. log: interface to STREAMS error logging and event log(7) STRM
 streamio: STREAMS ioctl commands. streamio(7) STRM
 slink: streams linker. slink(1M) NUAG
 slink: streams linker. slink(1M) SARM
 msgb: STREAMS message block structure. msgb(D4DK) DDRM
 datab: STREAMS message data structure. datab(D4DK) DDRM
 Transport Interface cooperating STREAMS module. timod: timod(7) STRM
 Interface read/write interface STREAMS module. /Transport tirdwr(7) STRM
 lists of automatically pushed STREAMS modules. /configure autopush(1M) SARM
 lists of automatically pushed STREAMS modules. /configure autopush(1M) STRM
 pckt: STREAMS Packet Mode module. pckt(7) STRM
 Emulation module. ptem: STREAMS Pseudo Terminal ptem(7) STRM
 information structure. qband: STREAMS queue flow control qband(D4DK) DDRM
 procedures structure. qinit: STREAMS queue processing qinit(D4DK) DDRM

queue: STREAMS queue structure. queue(D4DK) DDRM
 STREAMS Configuration File for STREAMS TCP/IP. strcf: strcf(4) NUAG
 STREAMS Configuration File for module. ldterm: standard
 strace: print STREAMS trace messages. strace(1M) SARM
 strace: print STREAMS trace messages. strace(1M) STRM
 console: STREAMS-based console interface. console(7) STRM
 fdetach: detach a name from a STREAMS-based file descriptor. fdetach(1M) STRM
 fdetach: detach a name from a STREAMS-based file descriptor. fdetach(3C) STRM
 an object. fattach: attach a STREAMS-based file descriptor to fattach(3C) STRM
 driver for AT&T windowing. xt: STREAMS-based multiplexed tty xt(7) SARM
 driver for AT&T windowing. xt: STREAMS-based multiplexed tty xt(7) STRM
 declaration structure. streamtab: STREAMS entity streamtab(D4DK) DDRM
 strccpy: streadd, strcadd, strccpy: copy strings,/ strccpy(3G) PRM
 daemon. strerr: STREAMS error logger strerr(1M) SARM
 daemon. strerr: STREAMS error logger strerr(1M) STRM
 string. strerror: get error message strerror(3C) PRM
 string manipulations. str: strfind, strrspn, strtrns: str(3G) PRM
 convert date and time to/ strftime, cftime, ascftime: strftime(3C) PRM
 strings. strftime: language specific strftime(4) PRM
 strings. strftime: language specific strftime(4) SARM
 gettxt: retrieve a text string. gettxt(3C) PRM
 regex: match patterns against a string. regex(1F) CHAR
 strncasecmp: string operations. string: strcasecmp, string(3) BSD
 strerror: get error message string. strerror(3C) PRM
 a prompt; verify and return a string answer. ckstr: display ckstr(1) SARM
 a prompt; verify and return a string answer. ckstr: display ckstr(1) SS
 /mvwinsstr, mvwinsnstr: insert string before character under/ curs_insstr(3X) CHAR
 convert date and time to string. /cftime, ascftime: strftime(3C) PRM
 strcoll: string collation. strcoll(3C) PRM
 bps: Bootstrap Parameter String driver. bps(1M) MBRMAN
 convert floating-point number to string. ecvt, fcvt, gcvt: ecvt(3C) PRM
 search a file for a character string. fgrep: fgrep(1) URM
 search a file for a character string. fgrep: fgrep(1) XNX
 gettxt: retrieve a text string from a message data base. gettxt(1) URM
 gets, fgets: get a string from a stream. gets(3S) PRM
 mbstowcs, wcstombs: multibyte string functions. mbstring: mbstring(3C) PRM
 parse suboptions from a string. getsubopt: getsubopt(3C) PRM
 tzset: convert date and time to string. /gmtime, ascftime, ctime(3C) PRM
 /of, or search for a text string in, message data bases. srchtxt(1) URM
 long integer and base-64 ASCII string. /l64a: convert between a64l(3C) PRM
 str: strfind, strrspn, strtrns: string manipulations. str(3G) PRM
 /mvwaddchstr, mvwaddchnstr: add string of characters (and/ curs_addchstr(3X) CHAR
 /mvwinchstr, mvwinchnstr: get a string of characters (and/ curs_inchstr(3X) CHAR
 /mvwinstr, mvwinstr: get a string of characters from a/ curs_instr(3X) CHAR
 /mvwaddstr, mvwaddnstr: add a string of characters to a curses/ curs_addstr(3X) CHAR
 puts, fputs: put a string on a stream. puts(3S) PRM

/the bootstrap parameter
 echo: put
 index, rindex:
 bcmp, bzero, ffs: bit and byte
 strcasemp, strncasemp:
 strspn, strcspn, strtok, strstr:
 elf_strptr: make a
 yes: print
 menu_mark: menus mark
 strcmp, strncmp, strcpy,/
 number. strtod, atof,: convert
 strtoul, atol, atoi: convert
 strxfrm:
 strftime: language specific
 strftime: language specific
 what: print identification
 streadd, strcadd, strecpy: copy
 in an object file or binary.
 /wgetnstr: get character
 exstr: extract
 binary. strings: find printable
 get and set system information
 debugging and line.
 and line. strip:
 /strncmp, strcpy, strncpy,
 log driver.
 string: strcasemp,
 strcpy,/
 string: strcat, strdup,
 /strcat, strdup, strcat, strcmp,
 /strcmp, strncmp, strcpy,
 /strlen, strchr, strrchr,
 queue or band of the queue.
 about a queue or band of the/
 /strcpy, strncpy, strlen, strchr,
 manipulations. str: strfind,
 /strchr, strrchr, strpbrk,
 /strspn, strcspn, strtok,
 double-precision number.
 strpbrk, strspn, strcspn,
 convert string to integer.
 string to integer. strtol,
 str: strfind, strspn,
 buf: block I/O data transfer
 cred: access credential
 datab: STREAMS message data
 map: private memory map
 msgb: STREAMS message block

 string on the processor in. initbp(1M) MBRMAN
 string on virtual output. echo(1F) CHAR
 string operations. index(3) BSD
 string operations. /bcopy, bstring(3) BSD
 string operations. string: string(3) BSD
 string operations. /strpbrk, string(3C) PRM
 string pointer. elf_strptr(3E) PRM
 string repeatedly. yes(1) NXN
 string routines. /set_menu_mark, menu_mark(3X) CHAR
 string: strcat, strdup, strcat, string(3C) PRM
 string to double-precision strtod(3C) PRM
 string to integer. strtol, strtol(3C) PRM
 string transformation. strxfrm(3C) PRM
 strings. strftime(4) PRM
 strings. strftime(4) SARM
 strings. what(1) PRM
 strings, compressing. strcpy: strcpy(3G) PRM
 strings: find printable strings strings(1) URM
 strings from curses terminal/ curs_getstr(3X) CHAR
 strings from source files. exstr(1) URM
 strings in an object file or strings(1) URM
 strings. sysinfo: sysinfo(2) PRM
 strip: strip symbol table, strip(1) PRM
 strip symbol table, debugging strip(1) PRM
 strlen, strchr, strrchr,/ string(3C) PRM
 strlog: submit messages to the strlog(D3DK) DDRM
 strncasemp: string operations. string(3) BSD
 strncmp, strcmp, strncmp, string(3C) PRM
 strncmp, strcpy, strncpy,/ string(3C) PRM
 strncpy, strlen, strchr,/ string(3C) PRM
 strpbrk, strspn, strcspn,/ string(3C) PRM
 strqget: get information about a strqget(D3DK) DDRM
 strqset: change information strqset(D3DK) DDRM
 strrchr, strpbrk, strspn,/ string(3C) PRM
 strspn, strtrns: string str(3G) PRM
 strspn, strcspn, strtok, strstr:/ string(3C) PRM
 strstr: string operations. string(3C) PRM
 strtod, atof,: convert string to strtod(3C) PRM
 strtok, strstr: string/ /strchr, string(3C) PRM
 strtol, strtoul, atol, atoi: strtol(3C) PRM
 strtoul, atol, atoi: convert strtol(3C) PRM
 strtrns: string manipulations. str(3G) PRM
 structure. buf(D4DK) DDRM
 structure. cred(D4DK) DDRM
 structure. datab(D4DK) DDRM
 structure. map(D4DK) DDRM
 structure. msgb(D4DK) DDRM

queue: STREAMS queue
 t_alloc: allocate a library
 t_free: free a library
 uio: scatter/gather I/O request
 ureadc: add character to a uio
 page frame number to page
 uio(D4DK). iovec: data storage
 dma_buf: data storage
 dma_cb: data storage
 processes using a file or file
 offsetof: offset of
 queue flow control information
 queue processing procedures
 identification and limit value
 STREAMS entity declaration
 driver's message/ free_rtn:
 mktime: converts a tm
 page_pptonum: convert page
 structure. queue(D4DK) DDRM
 structure. t_alloc(3N) NI
 structure. t_free(3N) NI
 structure. uio(D4DK) DDRM
 structure. ureadc(D3DK) DDRM
 structure. /convert page_numtopp(D3DK) DDRM
 structure for I/O using iovec(D4DK) DDRM
 structure for I/O using DMA. dma_buf(D4D) DDRM
 structure for I/O using DMA. dma_cb(D4D) DDRM
 structure. fuser: identify fuser(1M) SARM
 structure member. offsetof(3C) PRM
 structure. qband: STREAMS qband(D4DK) DDRM
 structure. qinit: STREAMS qinit(D4DK) DDRM
 structure. /STREAMS driver module_info(D4DK) DDRM
 structure. streamtab: streamtab(D4DK) DDRM
 structure that specifies a free_rtn(D4DK) DDRM
 structure to a calendar time. mktime(3C) PRM
 structure to page frame number.
 page_pptonum(D3DK) DDRM
 structure. uiomove: uiomove(D3DK) DDRM
 structure. uwritel: uwritel(D3DK) DDRM
 structure. ypfiles: ypfiles(4) NI
 strxfm: string transformation. strxfm(3C) PRM
 stty: set the options for a stty(1) BSD
 stty: set the options for a stty(1) URM
 sttydefs: maintain line settings sttydefs(1M) SARM
 stune: file format.. stune(4) SARM
 su. su(4) SARM
 su: become super-user or another su(1M) SARM
 su: become super-user or another su(1M) URM
 subchannel information on the ccisubinfo(1M) MBRMAN
 subchannel on a line on. ccidetach(1M) MBRMAN
 subchannel on a line to a new cciswitch(1M) MBRMAN
 subchannel on the/ cciattach: cciattach(1M) MBRMAN
 submit messages to the log strlog(D3DK) DDRM
 suboptions from a string. getsubopt(3C) PRM
 subpad, prefetch, pnoutrefresh, curs_pad(3X) CHAR
 subroutines. /dbm_nextkey, ndbm(3) BSD
 subroutines. /store, delete, dbm(3) NI
 subroutines. /store, delete, dbm(3X) BSD
 subsequent lines of one file. paste(1) URM
 subsystem. form_driver: form_driver(3X) CHAR
 subsystem. menu_driver: menu_driver(3X) CHAR
 subwin, derwin, mvderwin, curs_window(3X) CHAR
 subwindow association routines. form_win(3X) CHAR
 subwindow association routines. menu_win(3X) CHAR
 subwindows. /unpost_form: write form_post(3X) CHAR

copy kernel data using uio(D4DK)
 remove a character from a uio
 the YP database and directory

terminal.
 terminal.
 and hunt sequences for TTY/

user.
 user.

communications/ ccisubinfo:
 /disassociates this host with a
 host on the/ cciswitch: switch a
 Creates a connection to a
 driver. strlog:
 getsubopt: parse
 pechochar:/ curs_pad: newpad,
 dbm_open, dbm_store: data base
 firstkey, nextkey: database
 firstkey, nextkey: data base
 /same lines of several files or
 command processor for the forms
 command processor for the menus
 dupwin,/ /newwin, delwin, mvwin,
 /scale_form: forms window and
 /scale_menu: menus window and
 or erase forms from associated

or erase menus from associated subwindows. /unpost_menu: write menu_post(3X) CHAR
 mode. sulin: access single-user sulin(1M) SARM
 file. sum: calculate a checksum for a sum(1) BSD
 count of a file. sum: print checksum and block sum(1) URM
 du: summarize disk usage. du(1M) SARM
 du: summarize disk usage. du(1M) URM
 quot: summarize file system ownership. quot(1M) SARM
 system. repquota: summarize quotas for a file repquota(1M) SARM
 whatis: display a one-line summary about a keyword. whatis(1) BSD
 accounting/ acctcms: command summary from per-process acctcms(1M) SARM
 sync: update the super block. sync(1M) SARM
 sync: update the super block. sync(1M) URM
 sync: update super block. sync(2) PRM
 su: become super-user or another user. su(1M) SARM
 su: become super-user or another user. su(1M) URM
 getgroups, setgroups: get or set supplementary group access list/ getgroups(2) PRM
 initgroups: initialize the supplementary group access list. initgroups(3C) PRM
 mouse: mouse device driver supporting Bus, serial, and/ mouse(7) MOUSE
 routing. routing: system supporting for packet network routing(4) NUAG
 routing. routing: system supporting for packet network routing(4) SARM
 and transport of/ mailsurr: surrogate commands for routing mailsurr(4) SARM
 interval. nap: suspend execution for a short nap(2) XNX
 interval. sleep: suspend execution for an sleep(1) URM
 sleep: suspend execution for interval. sleep(3) BSD
 sleep: suspend execution for interval. sleep(3C) PRM
 in microseconds. usleep: suspend execution for interval usleep(3) BSD
 execution of an event. sleep: suspend process activity pending sleep(D3DK) DDRM
 pause: suspend process until signal. pause(2) PRM
 /install a signal mask and suspend process until signal. sigsuspend(2) PRM
 completion of block/ biowait: suspend processes pending biowait(D3DK) DDRM
 wakeup: resume suspended process execution. wakeup(D3DK) DDRM
 svc_dg_create,/ rpc_svc_create: svc_create, svc_destroy, rpc_svc_create(3N) NI
 rpc_svc_create: svc_create, svc_destroy, svc_dg_create,/ rpc_svc_create(3N) NI
 /svc_create, svc_destroy, svc_dg_create, svc_fd_create,/ rpc_svc_create(3N) NI
 svcerr_noproc,/ rpc_svc_err: svcerr_auth, svcerr_decode, rpc_svc_err(3N) NI
 rpc_svc_err: svcerr_auth, svcerr_decode, svcerr_noproc,/ rpc_svc_err(3N) NI
 /svcerr_auth, svcerr_decode, svcerr_noproc, svcerr_noprog,/ rpc_svc_err(3N) NI
 /svcerr_decode, svcerr_noproc, svcerr_progrvers,/ rpc_svc_err(3N) NI
 svcerr_noproc, svcerr_noprog, svcerr_progrvers,/ /svcerr_decode, rpc_svc_err(3N) NI
 /svcerr_noproc, svcerr_progrvers, svcerr_systemerr,/ rpc_svc_err(3N) NI
 routines for/ /svcerr_systemerr, svcerr_weakauth: library rpc_svc_err(3N) NI
 /svc_destroy, svc_dg_create, svc_fd_create, svc_raw_create,/ rpc_svc_create(3N) NI
 svc_freeargs, svc_getargs, rpc_svc_reg(3N) NI
 svc_getreqset,/ rpc_svc_reg: svc_getargs, svc_getreqset,/ rpc_svc_reg(3N) NI
 /svc_freeargs, svc_getargs, /svc_getreqset, svc_getrpcaller,/ rpc_svc_reg(3N) NI
 /svc_getargs, svc_getreqset, /svc_getrpcaller, svc_run,/ rpc_svc_reg(3N) NI
 /svc_dg_create, svc_fd_create, svc_raw_create, svc_tli_create,/ rpc_svc_create(3N) NI

Permuted Index

rpc_svc_calls: rpc_reg, svc_unreg, / rpc_svc_calls(3N) NI
 /svc_getreaset, svc_getrpcaller, svc_run, svc_sendreply: library / rpc_svc_reg(3N) NI
 for / /svc_getrpcaller, svc_run, svc_sendreply: library routines rpc_svc_reg(3N) NI
 /svc_fd_create, svc_raw_create, svc_tli_create, svc_tp_create, / rpc_svc_create(3N) NI
 /svc_raw_create, svc_tli_create, svc_tp_create, svc_vc_create: / rpc_svc_create(3N) NI
 /rpc_reg, svc_reg, svc_unreg, xpvt_register, / rpc_svc_calls(3N) NI
 /svc_tli_create, svc_tp_create, svc_vc_create: library routines / rpc_svc_create(3N) NI
 swab: swap bytes. swab(3C) PRM
 swap administrative interface. swab(1M) SARM
 swap bytes. swab(3C) PRM
 swap space. swapctl(2) PRM
 swap: swap administrative swap(1M) SARM
 swapcontext: manipulate user makecontext(3C) PRM
 swapctl: manage swap space. swapctl(2) PRM
 switch a subchannel on a line to cciswitch(1M) MBRMAN
 switches a line to a new host on ttyswitch(1M) MBRMAN
 sxt: pseudo-device driver. sxt(7) SARM
 sxt: pseudo-device driver. sxt(7) STRM
 symbol in shared object. dlsym(3X) PRM
 symbol table. elf_getarsym(3E) PRM
 symbol table. nlist(3) BSD
 symbol table, debugging and strip(1) PRM
 symbolic constants. unistd(4) SARM
 symbolic debugger. sdb(1) PRM
 symbolic link. readlink(2) PRM
 symbolic link to a file. symlink(2) PRM
 symbolic links to files. ln(1) BSD
 symlink: make a symbolic link to symlink(2) PRM
 sync: update super block. sync(2) PRM
 sync: update the super block. sync(1M) SARM
 sync: update the super block. sync(1M) URM
 synchronization of the system / adjtime(2) PRM
 synchronize a co-process with vsig(1F) CHAR
 synchronize a file's in-memory fsync(2) PRM
 synchronize access to a shared sdenter(2) XNX
 synchronize memory with physical msync(3C) PRM
 synchronize shared data access. sdgetv(2) XNX
 synchronize transport library. t_sync(3N) NI
 synchronous endpoint / mb2s_getinfo(3tai) TAI
 synchronous I/O multiplexing. select(3C) NI
 synchronous. mb2s_sendcancel: mb2s_sendcancel(3tai) TAI
 synchronous mode.. mb2s_sendreply(3tai) TAI
 synchronous mode.. mb2s_brdcst: mb2s_brdcst(3tai) TAI
 synchronous mode.. /Sends mb2s_send(3tai) TAI
 synchronous MULTIBUS II / mb2s_closeport(3tai) TAI
 synchronous MULTIBUS II / mb2s_openport(3tai) TAI
 /mvderwin, dupwin, wsyncup, syncok, wcursyncup, wsyncdown: / curs_window(3X) CHAR

interpreter with a C-like
 edsysadm:
 edsysadm:
 removal tool. delsysadm:
 removal tool. delsysadm:
 perform system administration.

 variables.
 information.
 functions.
 information strings.
 setlogmask: control system log.
 for syslogd system log daemon.

 /configuration file for
 messages. psignal,
 checkfsys: check a file
 cu: call another UNIX
 makefsys: create a file
 mkfs (s5): construct an s5 file
 mount (s5): mount an s5 file
 mount: mount a file
 newfs: construct a new file
 /proc: process file
 reboot: restart the operating
 tuneufs: tune up an existing file
 umount: unmount a file
 uname: get name of current UNIX
 who: who is on the
 sag:
 sar: sa1, sa2, sadc:
 sar:
 command; report process data and
 SA: devices administered by
 visual interface to perform
 /.rhosts: trusted hosts by
 /.rhosts: trusted hosts by
 or restore from, a full file
 or restore from, a full file
 shutdown: close down the
 ckbupscd: check file
 backup: initiate or control a
 fsba: file
 boot: UNIX
 stat: data returned by stat
 stat: data returned by stat
 stat: data returned by stat

 syntax. csh: shell command csh(1) URM
 sysadm interface editing tool. edsysadm(1M) SARM
 sysadm interface editing tool. edsysadm(1M) SS
 sysadm interface menu or task delsysadm(1M) SARM
 sysadm interface menu or task delsysadm(1M) SS
 sysadm: visual interface to sysadm(1M) SARM
 syscall: indirect system call. syscall(3) BSD
 sysconf: get configurable system sysconf(3C) PRM
 sysfs: get file system type sysfs(2) PRM
 sysi86: machine specific sysi86(2) PRM
 sysinfo: get and set system sysinfo(2) PRM
 syslog, openlog, closelog, syslog(3) BSD
 syslog.conf: configuration file syslog.conf(4) BSD
 syslogd: log system messages. syslogd(1M) BSD
 syslogd system log daemon. syslog.conf(4) BSD
 sys siglist: system signal psignal(3) BSD
 system. checkfsys(1M) SARM
 system. cu(1C) URM
 system. makefsys(1M) SARM
 system. mkfs(1M) SARM
 system. mount(1M) SARM
 system. mount(2) PRM
 system. newfs(1M) BSD
 system. proc(4) SARM
 system. reboot(1M) BSD
 system. tuneufs(1M) SARM
 system. umount(2) PRM
 system. uname(2) PRM
 system. who(1) URM
 system activity graph. sag(1) URM
 system activity report package. sar(1M) SARM
 system activity reporter. sar(1) URM
 system activity. timex: time a timex(1) URM
 System Administration. sa(7) SARM
 system administration. sysadm: sysadm(1M) SARM
 system and by user. hosts.equiv(4) NUAG
 system and by user. hosts.equiv(4) SARM
 system archive. fdp: create, fdp(1M) SARM
 system archive. ffile: create, ffile(1M) SARM
 system at a given time. shutdown(1M) BSD
 system backup schedule. ckbupscd(1M) SARM
 system backup session. backup(1M) SARM
 system block analyzer. fsba(1M) SARM
 system boot program. boot(1M) SARM
 system call. stat(4) XNX
 system call. stat(5) PRM
 system call. stat(5) SARM

syscall: indirect
 intro: introduction to
 truss: trace
 allow signals to interrupt
 /determine whether remote
 shutdown: shut down
 to allow synchronization of the
 an error message or panic the
 uux: UNIX-to-UNIX
 interactive/ fsck (ufs): file
 display:
 display a message on stderr or
 display a message on stderr or
 keyboard:
 display a driver message on
 uulog, uuname: UNIX-to-UNIX
 types: primitive
 rdate: set
 rdate: set
 fsdb (s5): s5 file
 vfstab: table of file
 sorted/ look: find words in the
 ufsdump: incremental file
 diskettes. ldsysdump: load
 perror: print
 names and statistics for a file
 uupick: public UNIX-to-UNIX
 setup: initialize
 a new group definition on the
 a group definition from the
 modify a group definition on the
 uptime: show how long the
 or print name of current host
 crash: examine
 entry. dirent: file
 statvfs, fstatvfs: get file
 ckbinarsys/ binarsys: remote
 sysinfo: get and set
 procedures. brc, bcheckrc:
 idbuild: build new UNIX
 idmkunix: build new UNIX
 that are available on this UNIX
 logger: add entries to the
 configuration file for syslogd
 closelog, setlogmask: control
 logins: list user and
 system call. syscall(3) BSD
 system calls and error numbers. intro(2) PRM
 system calls and signals. truss(1) URM
 system calls. siginterrupt: siginterrupt(3) BSD
 system can accept binary/ ckbinarsys(1M) SARM
 system, change system state. shutdown(1M) SARM
 system clock. /correct the time adjtime(2) PRM
 system. cmn_err: display cmn_err(D3DK) DDRM
 system command execution. uux(1C) URM
 system consistency check and fsck(1M) SARM
 system console display. display(7) SARM
 system console. ffmtmsg: ffmtmsg(1) URM
 system console. ffmtmsg: ffmtmsg(3C) PRM
 system console keyboard. keyboard(7) SARM
 system console. print: print(D2DK) DDRM
 system copy. uucp, uucp(1C) URM
 system data types. types(5) PRM
 system date from a remote host. rdate(1M) NUAG
 system date from a remote host. rdate(1M) SARM
 system debugger. fsdb(1M) SARM
 system defaults. vfstab(4) SARM
 system dictionary or lines in a look(1) BSD
 system dump. ufsdump(1M) SARM
 system dump from floppy ldsysdump(1M) SARM
 system error messages. perror(3C) PRM
 system. ff (generic): list file ff(1M) SARM
 system file copy. uuto, uuto(1C) URM
 system for first user. setup(1M) SARM
 system. groupadd: add (create) groupadd(1M) SARM
 system. groupdel: delete groupdel(1M) SARM
 system. groupmod: groupmod(1M) SARM
 system has been up. uptime(1) BSD
 system. hostname: set hostname(1) BSD
 system images. crash(1M) SARM
 system independent directory dirent(4) SARM
 system information. statvfs(2) PRM
 system information for the binarsys(4) SARM
 system information strings. sysinfo(2) PRM
 system initialization brc(1M) SARM
 system: issue a shell command. system(3S) PRM
 System kernel. idbuild(1M) SARM
 System kernel. idmkunix(1M) SARM
 system. /list of service grades uuglist(1C) URM
 system log. logger(1) BSD
 system log daemon. syslog.conf: syslog.conf(4) BSD
 system log. syslog, openlog, syslog(3) BSD
 system login information. logins(1M) SARM

interactive message processing
 syslogd: log
 (bfs): construct a boot file
 mount, unmount a file
 reboot: reboot
 filesystem: file
 quot: summarize file
 that registers distributed file
 that registers distributed file
 getpagesize: get
 transfer software package to the
 transfer software package to the
 removes a package from the
 removes a package from the
 prf: operating
 prfdc, prfsnap, prfpr: UNIX
 checker. quotacheck: file
 quotaon, quotaoff: turn file
 performed to stop the operating
 summarize quotas for a file
 /setrlimit: control maximum
 ufsrestore: incremental file
 to stop and reboot the operating
 end for the Source Code Control
 shutdown the driver when the
 psignal, sys_siglist:
 psignal, psiginfo:
 shut down system, change
 ustat: get file
 rwhod, in.rwhod:
 rwhod, in.rwhod:
 network routing. routing:
 network routing. routing:
 mnttab: mounted file
 sharetab: shared file
 sharetab: shared file
 setclk: set
 timezone: set default
 timezone: set default
 fstyp (generic): determine file
 sysfs: get file
 and statistics for a ufs file
 print name of current UNIX
 a new user login on the
 delete a user's login from the
 user's login information on the
 user interface to a remote
 system. mailx: mailx(1) URM
 system messages. syslogd(1M) BSD
 system. mkfs mkfs(1M) SARM
 system. mountsys, umountsys: mountsys(1M) SARM
 system or halt processor. reboot(3) BSD
 system organization. filesystem(7) SARM
 system ownership. quot(1M) SARM
 system packages. fstypes: file fstypes(4) NUAG
 system packages. fstypes: file fstypes(4) SARM
 system page size. getpagesize(3) BSD
 system. pkgadd: pkgadd(1M) SARM
 system. pkgadd: pkgadd(1M) SS
 system. pkgrm: pkgrm(1M) SARM
 system. pkgrm: pkgrm(1M) SS
 system profiler. prf(7) SARM
 system profiler. /prfstat, profiler(1M) SARM
 system quota consistency quotacheck(1M) SARM
 system quotas on and off. quotaon(1M) SARM
 system. rc0: run commands rc0(1M) SARM
 system. repquota: repquota(1M) SARM
 system resource consumption. getrlimit(2) PRM
 system restore. ufsrestore(1M) SARM
 system. /run commands performed rc6(1M) SARM
 System (SCCS). sccs: front sccs(1) BSD
 system shutdown. halt: halt(D2D) DDRM
 system signal messages. psignal(3) BSD
 system signal messages. psignal(3C) PRM
 system state. shutdown: shutdown(1M) SARM
 system statistics. ustat(2) PRM
 system status server. rwhod(1M) NUAG
 system status server. rwhod(1M) SARM
 system supporting for packet routing(4) NUAG
 system supporting for packet routing(4) SARM
 system table. mnttab(4) SARM
 system table. sharetab(4) NUAG
 system table. sharetab(4) SARM
 system time from hardware clock. setclk(1M) SARM
 system time zone. timezone(4) PRM
 system time zone. timezone(4) SARM
 system type. fstyp(1M) SARM
 system type information. sysfs(2) PRM
 system. /(ufs): list file names fff(1M) SARM
 system. uname: uname(1) URM
 system. useradd: administer useradd(1M) SARM
 system. userdel: userdel(1M) SARM
 system. usermod: modify a usermod(1M) SARM
 system using the. telnet: telnet(1) NUAG

Permuted Index

user interface to a remote transport program for the uucp Service (ots) device/ ots: sysconf: get configurable make literal copy of file a literal copy of an s5 file a literal copy of a ufs file fs (bfs): format of the bfs file Uutry: try to contact remote /fasthalt: reboot/halt the fsck: check and repair file mount (ufs): mount ufs file /umount: mount or unmount file automatically mount NFS file automatically mount NFS file report free disk space on file free disk space on ufs file dcopy (generic): copy file (s5): check and repair s5 file versus i-numbers for s5 file versus i-numbers for ufs file provide labels for file (s5): provide labels for s5 file provide labels for ufs file RFS resources from remote RFS resources from remote available for mounting by remote available for mounting by remote available for mounting by remote available for mounting by remote available for mounting by remote mount, unmount multiple file blocks and i-nodes for s5 file for mounting by remote for mounting by remote lpsystem: register remote /gigiplot, hpplot, implot, /gigiplot, hpplot, implot, t300, /hpplot, implot, t300, t300s, /implot, t300, t300s, t4013, bsearch: binary search a sorted mnttab: mounted file system nlist: get entries from symbol putdev: edits device putdgrp: edits device group setmnt: establish mount system using the. telnet: telnet(1) URM system. uucico: file uucico(1M) SARM System V/386 OSI Transport ots(1M) MBRMAN system variables. sysconf(3C) PRM system. volcopy (generic): volcopy(1M) SARM system. volcopy (s5): make volcopy(1M) SARM system. volcopy (ufs): make volcopy(1M) SARM system volume. fs(4) SARM system with debugging on. Uutry(1M) SARM system without checking the/ fastboot(1M) BSD systems. fsck(1M) SARM systems. mount(1M) SARM systems and remote resources. mount(1M) SARM systems. automount: automount(1M) NUAG systems. automount: automount(1M) SARM systems. df: df(1) BSD systems. df (ufs): report df(1M) SARM systems for optimal access time. dcopy(1M) SARM systems. fsck fsck(1M) SARM systems. /generate path names ncheck(1M) SARM systems. /generate pathnames ncheck(1M) SARM systems. labelit (generic): labelit(1M) SARM systems. labelit labelit(1M) SARM systems. labelit (ufs): labelit(1M) SARM systems. /list available dfshares(1M) NUAG systems. /list available dfshares(1M) SARM systems. /local NFS resource share(1M) NUAG systems. /local NFS resource share(1M) SARM systems. /local RFS resource share(1M) NUAG systems. /local RFS resource share(1M) SARM systems. /make local resource share(1M) NUAG systems. /make local resource share(1M) SARM systems. mountall, umountall: mountall(1M) SARM systems. /number of free disk df(1M) SARM systems. /resource unavailable unshare(1M) NUAG systems. /resource unavailable unshare(1M) SARM systems with the print service. lpsystem(1M) SARM t300, t300s, t4013, t450, tek:/ plot(1G) BSD t300s, t4013, t450, tek:/ plot(1G) BSD t4013, t450, tek: graphics/ plot(1G) BSD t450, tek: graphics filters for/ plot(1G) BSD table. bsearch(3C) PRM table. mnttab(4) SARM table. nlist(3) BSD table. putdev(1) SARM table. putdgrp(1) SARM table. setmnt(1M) SARM

sharetab: shared file system
 sharetab: shared file system
 strip: strip symbol
 class-dependent program header
 retrieve archive symbol
 get DoD Internet format host
 get DoD Internet format host
 convert DoD Internet format host
 convert DoD Internet format host
 vfstab:
 real-time dispatcher parameter
 getdtablesize: get descriptor
 dispatcher parameter
 iconv: code set conversion
 tbl: format
 classification and conversion
 hdestroy: manage hash search
 manually manipulate the routing
 manually manipulate the routing
 tabs: set
 request.
 /netdir_free, netdir_mergeaddr,
 ctags: create a
 file.
 talkd, in.talkd: server for
 talkd, in.talkd: server for
 talk:
 talk:
 program.
 program.
 structure.
 tam:
 trig: sin, sinf, cos, cosf,
 trig: sin, sinf, cos, cosf, tan,
 sinh, sinhf, cosh, coshf,
 sinh, sinhf, cosh, coshf, tanh,
 tcopy: copy a magnetic
 mt: magnetic
 iSBC 214 peripheral controller
 186/224A peripheral controller
 386/258 peripheral controller
 tdd: MULTIBUS
 tar:
 table. sharetab(4) NUAG
 table. sharetab(4) SARM
 table, debugging and line. strip(1) PRM
 table. /elf32_newphdr: retrieve elf_getphdr(3E) PRM
 table. elf_getarsym: elf_getarsym(3E) PRM
 table from a host. gettable: gettable(1M) NUAG
 table from a host. gettable: gettable(1M) SARM
 table. htable: htable(1M) NUAG
 table. htable: htable(1M) SARM
 table of file system defaults. vfstab(4) SARM
 table. rt_dptbl: rt_dptbl(4) SARM
 table size. getdtablesize(3) BSD
 table. ts_dptbl: time-sharing ts_dptbl(4) SARM
 tables. iconv(5) SARM
 tables for nroff or troff. tbl(1) BSD
 tables. /generate character chrtbl(1M) SARM
 tables. hsearch, hcreate, hsearch(3C) PRM
 tables. route: route(1M) NUAG
 tables. route: route(1M) SARM
 tabs on a terminal. tabs(1) URM
 tabs: set tabs on a terminal. tabs(1) URM
 t_accept: accept a connect t_accept(3N) NI
 taddr2uaddr, uaddr2taddr,/ netdir(3N) NI
 tags file for use with vi. ctags(1) URM
 tail: deliver the last part of a tail(1) URM
 talk program. talkd(1M) NUAG
 talk program. talkd(1M) SARM
 talk: talk to another user. talk(1) NUAG
 talk: talk to another user. talk(1) URM
 talk: talk to another user. talk(1) NUAG
 talk: talk to another user. talk(1) URM
 talkd, in.talkd: server for talk talkd(1M) NUAG
 talkd, in.talkd: server for talk talkd(1M) SARM
 t_alloc: allocate a library t_alloc(3N) NI
 tam: TAM transition libraries. tam(3X) CHAR
 TAM transition libraries. tam(3X) CHAR
 tan, tanf, asin, asinf, acos,/ trig(3M) PRM
 tanf, asin, asinf, acos, acosf,/ trig(3M) PRM
 tanh, tanhf, asinh, acosh,/ sinh(3M) PRM
 tanhf, asinh, acosh, atanh:/ sinh(3M) PRM
 tape. tcopy(1) BSD
 tape control. mt(1) BSD
 tape device driver. i214tp: i214tp(1M) MBRMAN
 tape device driver. /iSBC i224atp(1M) MBRMAN
 tape device driver. /iSBC i258tp(1M) MBRMAN
 tape device drivers. tdd(1M) MBRMAN
 tape file archiver. tar(1) URM

qt: QIC cartridge magnetic
 allocate a SCSI block for the
 delivery. dbon: sets
 sysadm interface menu or
 sysadm interface menu or
 programs for simple lexical
 transport endpoint.
 deroff: remove nroff, troff,
 deroff: remove nroff/troff,
 troff.
 /tcsetattr, tcsendbreak,
 tcsendbreak, tcdrain, tcfly,
 /tcsetattr, tcsendbreak, tcdrain,
 tcsendbreak, tcdrain,/ termios:
 /cfsetispeed, cfsetospeed,
 /tcgetpgrp, tcsetpgrp,
 endpoint.
 connection with another/
 Control Protocol.
 Control Protocol.
 Control Protocol.
 Configuration File for STREAMS
 Configuration File for STREAMS
 termios: tcgetattr, tcsetattr,
 tcfly,/ termios: tcgetattr,
 foreground process group id.
 /cfsetospeed, tcgetpgrp,
 drivers.
 search trees. tsearch, tfind,
 /t300, t300s, t4013, t450,
 initialization. init,
 /data_ahed, data_behind:
 /item_visible:
 directory: opendir, readdir,
 telnetd: DARPA
 telnetd: DARPA
 remote system using the.
 remote system using the.
 server.
 server.
 temporary file. tmpnam,
 tmpfile: create a
 tmpnam: create a name for a
 terminals.
 tape streamer interface. qt(7) SARM
 tar: tape file archiver. tar(1) URM
 target driver. sdi_getblk: sdi_getblk(D3I) SCSI
 target for front panel message dbon(1M) MBRMAN
 task removal tool. delsysadm: delsysadm(1M) SARM
 task removal tool. delsysadm: delsysadm(1M) SS
 tasks. lex: generate lex(1) PRM
 t_bind: bind an address to a t_bind(3N) NI
 tbl and eqn constructs. deroff(1) BSD
 tbl, and eqn constructs. deroff(1) URM
 tbl: format tables for nroff or tbl(1) BSD
 tcdrain, tcfly, tcfly,/ termios(2) PRM
 tcfly, cfgetospeed,/ /tcsetattr, termios(2) PRM
 tcfly, tcfly, cfgetospeed,/ termios(2) PRM
 tcgetattr, tcsetattr, termios(2) PRM
 tcgetpgrp, tcsetpgrp, tcgetsid:/ termios(2) PRM
 tcgetsid: general terminal/ termios(2) PRM
 t_close: close a transport t_close(3N) NI
 t_connect: establish a t_connect(3N) NI
 tcopy: copy a magnetic tape. tcopy(1) BSD
 TCP: Internet Transmission TCP(7) NI
 TCP: Internet Transmission TCP(7) NUAG
 TCP: Internet Transmission TCP(7) SARM
 TCP/IP. strcf: STREAMS strcf(4) NUAG
 TCP/IP. strcf: STREAMS strcf(4) SARM
 tcsendbreak, tcdrain, tcfly,/ termios(2) PRM
 tcsetattr, tcsendbreak, tcdrain, termios(2) PRM
 tcsetpgrp: set terminal tcsetpgrp(3C) PRM
 tcsetpgrp, tcgetsid: general/ termios(2) PRM
 tdd: MULTIBUS tape device tdd(1M) MBRMAN
 tdelete, twalk: manage binary tsearch(3C) PRM
 tee: pipe fitting. tee(1) URM
 tek: graphics filters for/ plot(1G) BSD
 telinit: process control init(1M) SARM
 tell if forms field has/ form_data(3X) CHAR
 tell if menus item is visible. menu_item_visible(3X) CHAR
 telldir, seekdir, rewinddir,/ directory(3C) PRM
 TELNET protocol server. telnetd(1M) NUAG
 TELNET protocol server. telnetd(1M) SARM
 telnet: user interface to a telnet(1) NUAG
 telnet: user interface to a telnet(1) URM
 telnetd: DARPA TELNET protocol telnetd(1M) NUAG
 telnetd: DARPA TELNET protocol telnetd(1M) SARM
 tmpnam: create a name for a tmpnam(3S) PRM
 temporary file. tmpfile(3S) PRM
 temporary file. tmpnam, tmpnam(3S) PRM
 term: conventional names for term(5) CHAR

terminals.
 term: format of compiled
 term: format of compiled
 file.
 file.
 /has_il, killchar, longname,
 terminfo/ captainfo: convert a
 terminfo/ captainfo: convert a
 interfaces (emulated) to the
 ct: spawn login to a remote
 ctermid: generate file name for
 jterm: reset layer of windowing
 stty: set the options for a
 stty: set the options for a
 tabs: set tabs on a
 tty: get the name of the
 ttyname, isatty: find name of a
 terminfo:
 terminfo:
 reset: establish or restore
 device/ atcs: asynchronous
 iasy: asynchronous
 ptem: STREAMS Pseudo
 group id. tcsetpgrp: set
 libwindows: windowing
 /wtimeout, typeahead: curses
 termiox: extended general
 tty: controlling
 tcsetpgrp, tcgetsid: general
 host control of windowing
 host control of windowing
 character strings from curses
 back) characters from curses
 dial: establish an outgoing
 ldterm: standard STREAMS
 ttydefs: file contains
 last: indicate last user or
 provide information for setting
 database. tput: initialize a
 /scrollok, nl, nonl: curses
 ttymon: port monitor for
 clear: clear the
 script: make typescript of a
 ismpx: return windowing
 line discipline. getty: set
 used between host and windowing
 used between host and windowing
 term: conventional names for term(5) SARM
 term file. term(4) CHAR
 term file. term(4) SARM
 term: format of compiled term term(4) CHAR
 term: format of compiled term term(4) SARM
 termattrs, termname: curses/ curs_termattrs(3X) CHAR
 termcap description into a captainfo(1M) CHAR
 termcap description into a captainfo(1M) SARM
 termcap library. /tputs: curses curs_termcap(3X) CHAR
 terminal. ct(1C) URM
 terminal. ctermid(3S) PRM
 terminal. jterm(1) URM
 terminal. stty(1) BSD
 terminal. stty(1) URM
 terminal. tabs(1) URM
 terminal. tty(1) URM
 terminal. ttyname(3C) PRM
 terminal capability data base. terminfo(4) CHAR
 terminal capability data base. terminfo(4) SARM
 terminal characteristics. tset, tset(1) BSD
 terminal controller script atcs(1M) MBRMAN
 (terminal) device driver. lasy(1M) MBRMAN
 Terminal Emulation module. ptem(7) STRM
 terminal foreground process tcsetpgrp(3C) PRM
 terminal function library. libwindows(3X) PRM
 terminal input option control/ curs_inopts(3X) CHAR
 terminal interface. termiox(7) SARM
 terminal interface. tty(7) SARM
 terminal interface. /tcsetpgrp, termios(2) PRM
 terminal. jagent: jagent(5) PRM
 terminal. jagent: jagent(5) SARM
 terminal keyboard. /get curs_getstr(3X) CHAR
 terminal keyboard. /get (or push curs_getch(3X) CHAR
 terminal line connection. dial(3C) NI
 terminal line discipline module. ldterm(7) STRM
 terminal line settings/ ttydefs(4) NI
 terminal logins. last(1) URM
 terminal modes. tset: tset(1) XNX
 terminal or query terminfo tput(1) URM
 terminal output option control/ curs_outopts(3X) CHAR
 terminal ports. ttymon(1M) SARM
 terminal screen. clear(1) URM
 terminal session. script(1) URM
 terminal state. ismpx(1) URM
 terminal type, modes, speed, and getty(1M) SARM
 terminal under. /protocol layers(5) PRM
 terminal under. /protocol layers(5) SARM

downloader for the 5620 DMD
 i8251: console
 term: conventional names for
 term: conventional names for
 layer multiplexor for windowing
 kill:
 WIFEXITED: wait for process to
 exit, _exit:
 for child process to stop or
 atexit: add program
 abort: generate an abnormal
 tic:
 tic:
 tigetstr: curses interfaces to
 initialize a terminal or query
 a termcap description into a
 a termcap description into a
 infocmp: compare or print out
 infocmp: compare or print out
 data base.
 data base.
 tcsendbreak, tcdrain, tclflush,/
 terminal interface.
 /killchar, longname, termattrs,
 isastream:
 command.
 command.
 command.
 specified priority/ bcanput:
 queue. canput:
 SAMESTR:
 message. datamsg:
 buffer.
 ed, red:
 ex:
 casual users). edit:
 newform: change the format of a
 page: browse or page through a
 fspec: format specification in
 fmt: simple
 ms:
 into memory or unlock process,
 gettext: retrieve a
 base. gettext: retrieve a
 /contents of, or search for a
 binary search trees. tsearch,
 terminal. wtinit: object wtinit(1M) SARM
 terminal/iSBXT 351 driver. i8251(1M) MBRMAN
 terminals. term(5) CHAR
 terminals. term(5) SARM
 terminals. layers: layers(1) URM
 terminate a process by default. kill(1) URM
 terminate or stop. /WIFSIGNALED, wait(3) BSD
 terminate process. exit(2) PRM
 terminate. wait: wait wait(2) PRM
 termination routine. atexit(3C) PRM
 termination signal. abort(3C) PRM
 terminfo compiler. tic(1M) CHAR
 terminfo compiler. tic(1M) SARM
 terminfo database. /tigetnum, curs_terminfo(3X) CHAR
 terminfo database. tput: tput(1) URM
 terminfo description. /convert captinfo(1M) CHAR
 terminfo description. /convert captinfo(1M) SARM
 terminfo descriptions. infocmp(1M) CHAR
 terminfo descriptions. infocmp(1M) SARM
 terminfo: terminal capability terminfo(4) CHAR
 terminfo: terminal capability terminfo(4) SARM
 termios: tcgetattr, tcsetattr, termios(2) PRM
 termiox: extended general termiox(7) SARM
 termname: curses environment/ curs_termattrs(3X) CHAR
 t_error: produce error message. t_error(3N) NI
 test a file descriptor. isastream(3C) STRM
 test: condition evaluation test(1) BSD
 test: condition evaluation test(1F) CHAR
 test: condition evaluation test(1) URM
 test for flow control in bcanput(D3DK) DDRM
 test for room in a message canput(D3DK) DDRM
 test if next queue is same type. SAMESTR(D3DK) DDRM
 test whether a message is a data datamsg(D3DK) DDRM
 testb: check for an available testb(D3DK) DDRM
 text editor. ed(1) URM
 text editor. ex(1) URM
 text editor (variant of ex for edit(1) URM
 text file. newform(1) URM
 text file. more, more(1) URM
 text files. fspec(4) SARM
 text formatters. fmt(1) URM
 text formatting macros. ms(7) BSD
 text, or data. plock: lock plock(2) PRM
 text string. gettext(3C) PRM
 text string from a message data gettext(1) URM
 text string in, message data/ srchtxt(1) URM
 tfind, tdelete, twalk: manage tsearch(3C) PRM

structure.
 program.
 program.
 Transfer Protocol server.
 Transfer Protocol server.
 tgetstr, tgoto, / curs_ termcap:
 tgoto, / curs_ termcap: tgetent,
 service information.
 curses / /tgetent, tgetflag,
 state.
 /tgetent, tgetflag, tgetnum,
 /tgetflag, tgetnum, tgetstr,
 of the board and returns
 register of the board in
 register of the board in
 to a remote system using
 to a remote system using
 field of the HOST ID record in

t_free: free a library t_free(3N) NI
 tftp: trivial file transfer tftp(1) NUAG
 tftp: trivial file transfer tftp(1) URM
 tftpd: DARPA Trivial File tftpd(1M) NUAG
 tftpd: DARPA Trivial File tftpd(1M) SARM
 tgetent, tgetflag, tgetnum, curs_ termcap(3X) CHAR
 tgetflag, tgetnum, tgetstr, curs_ termcap(3X) CHAR
 t_getinfo: get protocol-specific t_getinfo(3N) NI
 tgetnum, tgetstr, tgoto, tputs: curs_ termcap(3X) CHAR
 t_getstate: get the current t_getstate(3N) NI
 tgetstr, tgoto, tputs: curses / curs_ termcap(3X) CHAR
 tgoto, tputs: curses interfaces / curs_ termcap(3X) CHAR
 the. /interconnect registers ics_read(1M) MBRMAN
 the. /reads the interconnect ics_find_rec(D3D) DDRM
 the. /reads the interconnect ics_read(D3D) DDRM
 the. telnet: user interface telnet(1) NUAG
 the. telnet: user interface telnet(1) URM
 this. /returns the host id ics_hostid(D3D) DDRM
 tic: terminfo compiler. tic(1M) CHAR
 tic: terminfo compiler. tic(1M) SARM
 ticks. /delay process execution delay(D3DK) DDRM
 ticks. drv_ usectohz: drv_ usectohz(D3DK) DDRM
 ticks to microseconds. drv_ hztousec(D3DK) DDRM
 ticlts, ticots, ticotsord: ticlts(7) NI
 ticots, ticotsord: loopback tidlts(7) NI
 ticotsord: loopback transport ticlts(7) NI
 tgetflag, tgetnum, tgetstr: / curs_ terminfo(3X) CHAR
 tgetnum, tgetstr: curses / curs_ terminfo(3X) CHAR
 tgetstr: curses interfaces to / curs_ terminfo(3X) CHAR
 time. ftime(3C) BSD
 time. stime(2) PRM
 time. time(2) PRM
 time. at, batch: at(1) URM
 time. dcopy (generic): copy dcopy(1M) SARM
 time. dcopy (s5): copy s5 dcopy(1M) SARM
 time. /execute a function timeout(D3DK) DDRM
 time: get time. time(2) PRM
 time. getdate: getdate(3C) PRM
 time. mktime: converts mktime(3C) PRM
 time. profile: setting profile(4) SARM
 time. /settimeofday: gettimeofday(3) BSD
 time. /settimeofday: gettimeofday(3C) PRM
 time. shutdown: shutdown(1M) BSD
 time: time a command. time(1) URM
 timeout: execute a function timeout(D3DK) DDRM
 timeout, wtimeout, typeahead: / curs_ inopts(3X) CHAR
 timeout(D3DK) function call. untimeout(D3DK) DDRM

for a specified number of clock
 convert microseconds to clock
 drv_ hztousec: convert clock
 loopback transport providers.
 transport providers. ticlts,
 providers. ticlts, ticots,
 /putp, vidputs, vidattr, mvcur,
 /vidattr, mvcur, tgetflag,
 /mvcur, tgetflag, tgetnum,
 ftime: get date and
 stime: set
 time: get
 execute commands at a later
 file systems for optimal access
 file systems for optimal access
 after a specified length of

convert user format date and
 a tm structure to a calendar
 up an environment at login
 get or set the date and
 get or set the date and
 close down the system at a given

after a specified length of /
 /raw, noraw, noqiflush, qiflush,
 untimeout: cancel previous

Permuted Index

get/set value of interval timer. `getitimer, setitimer`: `getitimer(3C)` PRM
 times: get process times. `times(3C)` BSD
 utimes: set file times. `utimes(3)` BSD
 jobs queued to run at specified times. `atq`: display the `atq(1)` URM
 difference between two calendar times. `difftime`: computes the `difftime(3C)` PRM
 process times. times: get process and child `times(2)` PRM
 times: get process times. `times(3C)` BSD
 update access and modification times of a file. `touch`: `touch(1)` URM
 get process and child process times. `times`: `times(2)` PRM
 set file access and modification times. `utime`: `utime(2)` PRM
 parameter table. `ts_dptbl`: time-sharing dispatcher `ts_dptbl(4)` SARM
 nice: change priority of a time-sharing process. `nice(2)` PRM
 process data and system/ `timex`: time a command; report `timex(1)` URM
 given offset from GMT. `timezone`: get time zone name `timezone(3C)` BSD
 time zone. `timezone`: set default system `timezone(4)` PRM
 time zone. `timezone`: set default system `timezone(4)` SARM
 cooperating STREAMS module. `timod`: Transport Interface `timod(7)` STRM
 read/write interface STREAMS/ `tirdwr`: Transport Interface `tirdwr(7)` STRM
 request. `t_listen`: listen for a connect `t_listen(3N)` NI
 event on a transport endpoint. `t_look`: look at the current `t_look(3N)` NI
 mktime: converts a `tm` structure to a calendar time. `mktime(3C)` PRM
 file. `tmpfile`: create a temporary `tmpfile(3S)` PRM
 for a temporary file. `tmpnam, tmpnam`: create a name `tmpnam(3S)` PRM
 name server. `tnamed, in.tnamed`: DARPA trivial `tnamed(1M)` NUAG
 name server. `tnamed, in.tnamed`: DARPA trivial `tnamed(1M)` SARM
 (write) a curses screen from (to) a file. `/scr_set`: read `curs_scr_dump(3X)` CHAR
 /`tolower, _toupper, tolower` `toascii`: translate characters. `conv(3C)` PRM
 , `popen, pclose`: initiate pipe to/from a process. `popen(3S)` PRM
 /`toupper, tolower, _toupper` `_tolower, toascii`: translate/ `conv(3C)` PRM
 , `toascii`:/ `conv`: `toupper`, `tolower, _toupper, _tolower`, `conv(3C)` PRM
 interface menu or task removal tool. `delsysadm`: `sysadm` `delsysadm(1M)` SARM
 interface menu or task removal tool. `delsysadm`: `sysadm` `delsysadm(1M)` SS
 sysadm interface editing tool. `edsysadm`: `edsysadm(1M)` SARM
 sysadm interface editing tool. `edsysadm`: `edsysadm(1M)` SS
 endpoint. `t_open`: establish a transport `t_open(3N)` NI
 tsort: topological sort. `tsort(1)` PRM
 deck manipulation/ `panel_top`: top_panel, bottom_panel: panels `panel_top(3X)` CHAR
 /`current_item, set_top_row`, `top_row, item_index`: set and get/
 `menu_item_current(3X)` CHAR
 transport endpoint. `t_optmgmt`: manage options for a `t_optmgmt(3N)` NI
 acctmrg: merge or add `tosmtp`: send mail to SMTP. `tosmtp(1M)` SARM
 modification times of a file. total accounting files. `acctmrg(1M)` SARM
 curs_touch: touchwin, touch: update access and `touch(1)` URM
 , `wtouchln`,/ `curs_touch` `touchline, untouchwin, wtouchln`,/ `curs_touch(3X)` CHAR
 , `conv`: `toupper, tolower`, `touchwin, touchline, untouchwin`, `curs_touch(3X)` CHAR
 , `_tolower, toascii`:/ `conv`: `toupper, _tolower, toascii`:/ `conv(3C)` PRM
 , `_tolower, toascii`:/ `conv`: `toupper, tolower, _toupper`, `conv(3C)` PRM

/del_curterm, restartterm, query terminfo database.
 /tgetnum, tgetstr, tgoto, /del_curterm, restartterm, tparm,
 tparm, tputs, putp, vidputs,/ curs_terminfo(3X) CHAR
 tput: initialize a terminal or tput(1) URM
 tputs: curses interfaces/ curs_termcap(3X) CHAR
 tputs, putp, vidputs, vidattr,/ curs_terminfo(3X) CHAR
 tr: translate characters. tr(1) BSD
 tr: translate characters. tr(1) URM
 trace. ptrace(2) PRM
 trace. trpt(1M) NUAG
 trace. trpt(1M) SARM
 trace messages. strace(1M) SARM
 trace messages. strace(1M) STRM
 trace system calls and signals. truss(1) URM
 traces. xtt: extract xtt(1M) SARM
 tracing. log: interface to log(7) STRM
 transaction id.. mps_free_tid: mps_free_tid(D3D) DDRM
 transaction ids.. mps_get_tid(D3D) DDRM
 transaction. mps_AMPcancel: mps_AMPcancel(D3D) DDRM
 transactionless message in mb2a_send(3tai) TAI
 transactionless message in mb2s_send(3tai) TAI
 Transfer. dma_swstart: dma_swstart(D3D) DDRM
 transfer program. ftp(1) NUAG
 transfer program. ftp(1) URM
 transfer program. tftp(1) NUAG
 transfer program. tftp(1) URM
 transfer protocol server. ftpd(1M) NUAG
 transfer protocol server. ftpd(1M) SARM
 Transfer Protocol server. tftpd(1M) NUAG
 Transfer Protocol server. tftpd(1M) SARM
 Transfer Protocol. /SMTP mail to smtp(1M) SARM
 transfer software package to the pkgadd(1M) SARM
 transfer software package to the pkgadd(1M) SS
 transfer structure. buf(D4DK) DDRM
 transfer YP map from a YP server ypxfr(1M) NI
 transformation. strxfm(3C) PRM
 transition libraries. tam(3X) CHAR
 translate characters. tr(1) BSD
 translate characters. tr(1) URM
 translate characters. /tolower, conv(3C) PRM
 translate mail alias names. mailalias(1) URM
 translate package format. pkgtrans(1) SARM
 translate package format. pkgtrans(1) SS
 translate scb virtual addresses. sdi_translate(D3I) SCSI
 translation. cof2elf(1) PRM
 translation. /elf32_xlatetom: elf_xlate(3E) PRM
 translation. /generic netdir(3N) NI
 transliterate protocol trace. trpt(1M) NUAG
 transliterate protocol trace. trpt(1M) SARM

Permuted Index

TCP: Internet Transmission Control Protocol. TCP(7) NI
 TCP: Internet Transmission Control Protocol. TCP(7) NUAG
 TCP: Internet Transmission Control Protocol. TCP(7) SARM
 queues request messages for transmission. mps_AMPsend_rsvp:
 mps_AMPsend_rsvp(D3D) DDRM
 mpscnv: Convert transport code. mpscnv(1M) MBRMAN
 /Opens a synchronous MULTIBUS II transport endpoint.. mb2s_openport(3tai) TAI
 t_bind: bind an address to a transport endpoint. t_bind(3N) NI
 t_close: close a transport endpoint. t_close(3N) NI
 t_open: establish a transport endpoint. t_open(3N) NI
 t_optmgmt: manage options for a transport endpoint. t_optmgmt(3N) NI
 t_unbind: disable a transport endpoint. t_unbind(3N) NI
 an asynchronous MULTIBUS II transport endpoint.. /Closes mb2a_closeport(3tai) TAI
 an asynchronous MULTIBUS II transport endpoint.. /Opens mb2a_openport(3tai) TAI
 look at the current event on a transport endpoint. t_look: t_look(3N) NI
 STREAMS module. timod: Transport Interface cooperating timod(7) STRM
 interface STREAMS/ tirdwr: Transport Interface read/write tirdwr(7) STRM
 t_sync: synchronize transport library. t_sync(3N) NI
 Closes a synchronous MULTIBUS II transport. mb2s_closeport: mb2s_closeport(3tai) TAI
 /netdir_spperror: generic transport name-to-address/ netdir(3N) NI
 commands for routing and transport of mail. /surrogate mailsurr(4) SARM
 system. uucico: file transport program for the uucp uucico(1M) SARM
 the scheduler for the uucp file transport program. uusched: uusched(1M) SARM
 nlsprovider: get name of transport provider. nlsprovider(3N) NI
 ticots, ticotsord: loopback transport providers. ticlts, ticlts(7) NI
 driver. ots: System V/386 OSI Transport Service (ots) device ots(1M) MBRMAN
 a connection with another transport user. /establish t_connect(3N) NI
 ieec_handler: IEEE exception trap handler function. ieec_handler(3M) BSD
 /panel_below: panels deck traversal primitives. panel_above(3X) CHAR
 data sent over a connection. t_rcv: receive data or expedited t_rcv(3N) NI
 confirmation from a connect/ t_rcvconnect: receive the t_rcvconnect(3N) NI
 from disconnect. t_rcvdis: retrieve information t_rcvdis(3N) NI
 an orderly release indication. t_rcvrel: acknowledge receipt of t_rcvrel(3N) NI
 t_rcvdata: receive a data unit. t_rcvdata(3N) NI
 t_rcvuderr: receive a unit data t_rcvuderr(3N) NI
 error indication. tree. ftw(3C) PRM
 ftw, nftw: walk a file trees. tsearch, tfind, tdelete, tsearch(3C) PRM
 twalk: manage binary search trig: sin, sinf, cos, cosf, tan, trig(3M) PRM
 tanf, asin, asinf, acos, acosf, trigonometric functions. /acosf, trig(3M) PRM
 atan, atanf, atan2, atan2f: trim bytes from a message. adjmsg(D3DK) DDRM
 adjmsg: trivial file transfer program. tftp(1) NUAG
 tftp: trivial file transfer program. tftp(1) URM
 tftp: Trivial File Transfer Protocol tftpd(1M) NUAG
 server. tftpd: DARPA Trivial File Transfer Protocol tftpd(1M) SARM
 server. tftpd: DARPA Trivial File Transfer Protocol tftpd(1M) SARM
 tnamed, in.tnamed: DARPA trivial name server. tnamed(1M) NUAG
 tnamed, in.tnamed: DARPA trivial name server. tnamed(1M) SARM
 tbl: format tables for nroff or troff. tbl(1) BSD

.so requests from nroff or between versions of a checknr: check nroff and deroff: remove nroff documents. trace. trace. values. to a specified length. signals. user. hosts.equiv, .rhosts: user. hosts.equiv, .rhosts: 386, u370: get processor type true, false: provide with debugging on. Uutry: dispatcher parameter table. manage binary search trees. setting terminal modes. restore terminal/ data over a connection. disconnect request. release. library. STREAMS compatibility module. interface. xt: STREAMS-based multiplexed xt: STREAMS-based multiplexed terminal. settings and hunt sequences for monitor-specific information. line settings information for/ line settings information for terminal ports. terminal. directory search list for utmp file of the current user. for ttyname. new host on the communications/ attempts to set value of a endpoint. tunefs: system. off. quotaon, quotaoff: /runacct, shutacct, startup, trees. tsearch, tfind, tdelete, troff input. /and eliminate soelim(1) BSD troff input file. /differences diffmk(1) BSD troff input files; report/ checknr(1) BSD troff, tbl and eqn constructs. deroff(1) BSD troff: typeset or format troff(1) BSD trpt: transliterate protocol trpt(1M) NUAG trpt: transliterate protocol trpt(1M) SARM true, false: provide truth true(1) URM truncate, ftruncate: set a file truncate(3C) PRM truss: trace system calls and truss(1) URM trusted hosts by system and by hosts.equiv(4) NUAG trusted hosts by system and by hosts.equiv(4) SARM truth value. /u3b5, u3b15, vax, machid(1) URM truth values. true(1) URM try to contact remote system Uutry(1M) SARM ts_dptbl: time-sharing ts_dptbl(4) SARM tsearch, tfind, tdelete, twalk: tsearch(3C) PRM tset: provide information for tset(1) XNX tset, reset: establish or tset(1) BSD t_snd: send data or expedited t_snd(3N) NI t_snddis: send user-initiated t_snddis(3N) NI t_sndrel: initiate an orderly t_sndrel(3N) NI t_sndudata: send a data unit. t_sndudata(3N) NI tsort: topological sort. tsort(1) PRM t_sync: synchronize transport t_sync(3N) NI tcompat: V7, 4BSD and XENIX tcompat(7) SARM tty: controlling terminal tty(7) SARM tty driver for AT&T windowing. xt(7) SARM tty driver for AT&T windowing. xt(7) STRM tty: get the name of the tty(1) URM TTY ports. /maintain line sttydefs(1M) SARM ttyadm: format and output port ttyadm(1M) SARM ttydefs: file contains terminal ttydefs(4) NI ttymon. /file contains terminal ttydefs(4) NI ttymon: port monitor for ttymon(1M) SARM ttyname, isatty: find name of a ttyname(3C) PRM ttyname. ttysrch: ttysrch(4) SARM ttyslot: find the slot in the ttyslot(3C) PRM ttysrch: directory search list ttysrch(4) SARM ttyswitch: switches a line to a ttyswitch(1M) MBRMAN tunable parameter. idtune: idtune(1M) SARM t_unbind: disable a transport t_unbind(3N) NI tune up an existing file system. tunefs(1M) SARM tunefs: tune up an existing file tunefs(1M) SARM turn file system quotas on and quotaon(1M) SARM turnacct: shell procedures for/ acctsh(1M) SARM twalk: manage binary search tsearch(3C) PRM

elf_kind: determine file type. elf_kind(3E) PRM
 file: determine file type. file(1) URM
 the size of an object file type. /elf32_fsize: return elf_fsize(3E) PRM
 (generic): determine file system type. fstyp fstyp(1M) SARM
 sysfs: get file system type information. sysfs(2) PRM
 discipline. getty: set terminal type, modes, speed, and line getty(1M) SARM
 /fpclass, unordered: determine type of floating-point number. isnan(3C) PRM
 mach: display the processor type of the current host. mach(1) BSD
 test if next queue is same type. SAMESTR: SAMESTR(D3DK) DDRM
 vax, 386, u370: get processor type truth value. /u3b5, u3b15, machid(1) URM
 field_arg: forms field data type validation. /field_type, form_field_validation(3X) CHAR
 /qiflush, timeout, wtimeout, typeahead: curses terminal input/ curs_inopts(3X) CHAR
 nl_types: native language data types. nl_types(5) PRM
 nl_types: native language data types. nl_types(5) SARM
 types: primitive system data types. types(5) PRM
 /checks for certain board types in the designated slot. ics_agent_cmp(D3D) DDRM
 session. script: make types: primitive system data types(5) PRM
 eqn, neqn, checkeq: typescript of a terminal script(1) URM
 troff: typeset mathematics. eqn(1) BSD
 /localtime, gmtime, asctime, typeset or format documents. troff(1) BSD
 /u3b2, u3b5, u3b15, vax, 386, tzset: convert date and time to/ ctime(3C) PRM
 386, u370: get processor type truth/ machid(1) URM
 machid: pdp11, u3b, u3b2, u3b5, u3b, u3b2, u3b5, u3b15, vax, machid(1) URM
 u370: get/ machid: pdp11, u3b, u3b15, vax, 386, u370: get/ machid(1) URM
 machid: pdp11, u3b, u3b2, u3b2, u3b5, u3b15, vax, 386, machid(1) URM
 u3b5, u3b15, vax, 386, u370: get/ machid(1) URM
 /netdir_mergeaddr, taddr2uaddr, uaddr2taddr, netdir_perror,/ netdir(3N) NI
 interval in microseconds. uadmin: administrative control. uadmin(1M) SARM
 Protocol. uadmin: administrative control. uadmin(2) PRM
 Protocol. ualarm: schedule signal after ualarm(3) BSD
 Protocol. ucontext: user context. ucontext(5) PRM
 system. mkfs UDP: Internet User Datagram UDP(7) NI
 dir (ufs): format of UDP: Internet User Datagram UDP(7) NUAG
 mkfs (ufs): construct a UDP: Internet User Datagram UDP(7) SARM
 check and interactive/ fsck (ufs): construct a ufs file mkfs(1M) SARM
 fsdb (ufs): ufs directories. dir(4) SARM
 file names and statistics for ufs file system. mkfs(1M) SARM
 (ufs): make a literal copy of a ufs file system consistency fsck(1M) SARM
 fs (ufs): format of ufs file system debugger. fsdb(1M) SARM
 mount (ufs): mount ufs file system. ff (ufs): list ff(1M) SARM
 (ufs): report free disk space on ufs file system. volcopy volcopy(1M) SARM
 pathnames versus i-numbers for ufs file system volume. fs(4) SARM
 (ufs): provide labels for ufs file systems. mount(1M) SARM
 inode ufs file systems. df df(1M) SARM
 ufs file systems. /generate ncheck(1M) SARM
 ufs file systems. labelit labelit(1M) SARM
 (ufs): format of a ufs inode. inode(4) SARM

directories. dir (ufs): format of ufs dir(4) SARM
 volume. fs (ufs): format of ufs file system fs(4) SARM
 i-numbers for ufs file/ ncheck (ufs): generate pathnames versus ncheck(1M) SARM
 inode (ufs): format of a inode(4) SARM
 statistics for a ufs file/ ff (ufs): list file names and ff(1M) SARM
 ufs file system. volcopy (ufs): make a literal copy of a volcopy(1M) SARM
 mount (ufs): mount ufs file systems. mount(1M) SARM
 file systems. labelit (ufs): provide labels for ufs labelit(1M) SARM
 ufs file systems. df (ufs): report free disk space on df(1M) SARM
 fsdb (ufs): ufs file system debugger. fsdb(1M) SARM
 dump. ufsdump: incremental file system ufsdump(1M) SARM
 system restore. ufsrestore: incremental file ufsrestore(1M) SARM
 getpw: get name from UID getpw(3C) PRM
 structure. uio: scatter/gather I/O request uio(D4DK) DDRM
 ureadc: add character to a uio structure. ureadc(D3DK) DDRM
 remove a character from a uio structure. uwritec uwritec(D3DK) DDRM
 storage structure for I/O using uio(D4DK). iovec: data iovec(D4DK) DDRM
 uiomove: copy kernel data using uio(D4DK) structure. uiomove(D3DK) DDRM
 uio(D4DK) structure. uiomove(D3DK) DDRM
 uiomove: copy kernel data using uiomove(D3DK) DDRM
 ul: underline. ul(1) BSD
 /endspent, fgetspent, lckpwwdf, ulckpwwdf: manipulate shadow/ getspent(3C) PRM
 mask. ulimit: get and set user limits. ulimit(2) PRM
 mask. umask: set and get file creation umask(2) PRM
 systems and remote/ mount, umask: set file-creation mode umask(1) URM
 mount. umount: mount or unmount file mount(1M) SARM
 umount: unmount a file system. umount(2) PRM
 multiple file/ mountall, umountall: mount, unmount mountall(1M) SARM
 file system. mountfsys, umountfsys: mount, unmount a mountfsys(1M) SARM
 system. unname: get name of current UNIX unname(2) PRM
 UNIX system. unname: print name of current unname(1) URM
 unshare: make local NFS resource unavailable for mounting by/ unshare(1M) NUAG
 unshare: make local NFS resource unavailable for mounting by/ unshare(1M) SARM
 line on the/ cciunbind: unbinds a line discipline from a cciunbind(1M) MBRMAN
 call-when-buffer-available/ unbufcall: cancel a pending unbufcall(D3DK) DDRM
 expand or display/ compress, uncompress, zcat: compress, compress(1) URM
 use_env, putwin,/ curs_util: unctrl, keyname, filter, curs_util(3X) CHAR
 host and windowing terminal under. /protocol used between layers(5) PRM
 host and windowing terminal under. /protocol used between layers(5) SARM
 ul: underline. ul(1) BSD
 file. unget: undo a previous get of an SCCS unget(1) PRM
 SCCS file. unget: undo a previous get of an unget(1) PRM
 input stream. ungetc: push character back onto ungetc(3S) PRM
 /wgetch, mvwgetch, mvwgetch, ungetch: get (or push back)/ curs_getch(3X) CHAR
 ifdef'ed lines from C program/ unifdef: resolve and remove unifdef(1) BSD
 seed48, lcong48: generate uniformly distributed/ /srand48, drand48(3C) PRM
 elf_rawfile: retrieve uninterpreted file contents. elf_rawfile(3E) PRM
 file. uniq: report repeated lines in a uniq(1) URM

Permuted Index

mkstemp: make a	unique file name.	mkstemp(3) BSD
mktemp: make a	unique file name.	mktemp(3C) PRM
host. gethostid: get	unique identifier of current	gethostid(3) BSD
connld: line discipline for	unique stream connections.	connld(7) STRM
constants.	unistd: header file for symbolic	unistd(4) SARM
t_rcvudata: receive a data	unit.	t_rcvudata(3N) NI
t_sndudata: send a data	unit.	t_sndudata(3N) NI
t_rcvuderr: receive a	unit data error indication.	t_rcvuderr(3N) NI
	units: conversion program.	units(1) URM
an I/O request into manageable	units. dma_pageio: break up	dma_pageio(D3DK) DDRM
program number mapper. rpcbind:	universal addresses to RPC	rpcbind(1M) NI
program number mapper. rpcbind:	universal addresses to RPC	rpcbind(1M) NUAG
install specific portions of a	UNIX package. custom:	custom(1M) SARM
install specific portions of a	UNIX package. custom:	custom(1) XNX
cu: call another	UNIX system.	cu(1C) URM
uname: print name of current	UNIX system.	uname(1) URM
uname: get name of current	UNIX system.	uname(2) PRM
boot:	UNIX system boot program.	boot(1M) SARM
idbuild: build new	UNIX System kernel.	idbuild(1M) SARM
idmkunix: build new	UNIX System kernel.	idmkunix(1M) SARM
that are available on this	UNIX system. /of service grades	uuglist(1C) URM
prfstat, prfdc, prfsnap, prfpr:	UNIX system profiler. /prfld,	profiler(1M) SARM
execution. uux:	UNIX-to-UNIX system command	uux(1C) URM
uucp, uulog, uname:	UNIX-to-UNIX system copy.	uucp(1C) URM
uuto, uupick: public	UNIX-to-UNIX system file copy.	uuto(1C) URM
link, unlink: link and	unlink files and directories.	link(1M) SARM
and directories. link,	unlink: link and unlink files	link(1M) SARM
	unlink: remove directory entry.	unlink(2) PRM
from the head of a message.	unlinkb: remove a message block	unlinkb(D3DK) DDRM
or writing. locking: lock or	unlock a file region for reading	locking(2) XNX
master/slave pair. unlockpt:	unlock a pseudo-terminal	unlockpt(3C) STRM
mlockall, munlockall: lock or	unlock address space.	mlockall(3C) PRM
mlock, munlock: lock (or	unlock) pages in memory.	mlock(3C) PRM
plock: lock into memory or	unlock process, text, or data.	plock(2) PRM
pseudo-terminal master/slave/	unlockpt: unlock a	unlockpt(3C) STRM
munmap:	unmap pages of memory.	munmap(2) PRM
mountfsys, umountfsys: mount,	unmount a file system.	mountfsys(1M) SARM
umount:	unmount a file system.	umount(2) PRM
mount, umount: mount or	unmount file systems and remote/	mount(1M) SARM
mountall, umountall: mount,	unmount multiple file systems.	mountall(1M) SARM
resource. fumount: forced	unmount of an advertised	fumount(1M) NUAG
resource. fumount: forced	unmount of an advertised	fumount(1M) SARM
rmountall, rumountall: mount,	unmount Remote File Sharing/	rmountall(1M) NUAG
rmountall, rumountall: mount,	unmount Remote File Sharing/	rmountall(1M) SARM
/isnand, isnanf, finite, fpclass,	unordered: determine type of/	isnan(3C) PRM
files. pack, pcat,	unpack: compress and expand	pack(1) URM
forms/ form_post: post_form,	unpost_form: write or erase	form_post(3X) CHAR

clock: report CPU time
 /unctrl, keyname, filter
 su: become super-user or another
 su: become super-user or another
 talk: talk to another
 talk: talk to another
 write: write to another
 setuid, setgid: set
 idload: Remote File Sharing
 idload: Remote File Sharing
 information. logins: list
 ucontext:
 setcontext: get and set current
 swapcontext: manipulate
 crontab:
 get character login name of the
 mps_get_soldata: copies
 mps_get_unsoldata: copies
 UDP: Internet
 UDP: Internet
 UDP: Internet
 /getgid, getegid: get real
 chkey: change
 chkey: change
 environ:
 environ:
 environ:
 with another transport
 getdate: convert
 print group membership of
 useracc: verify whether
 trusted hosts by system and by
 trusted hosts by system and by
 prompts for and validates a
 prompts for and validates a
 generate disk accounting data by
 setreuid: set real and effective
 fingerd, in.fingerd: remote
 fingerd, in.fingerd: remote
 system using the. telnet:
 system using the. telnet:
 ulimit: get and set
 listusers: list
 useradd: administer a new
 and ID. id: print the
 and ID. id: print the
 whois: Internet
 used. clock(3C) PRM
 user_env, putwin, getwin,/ curs_util(3X) CHAR
 user. su(1M) SARM
 user. su(1M) URM
 user. talk(1) NUAG
 user. talk(1) URM
 user. write(1) URM
 user and group IDs. setuid(2) PRM
 user and group mapping. idload(1M) NUAG
 user and group mapping. idload(1M) SARM
 user and system login logins(1M) SARM
 user context. ucontext(5) PRM
 user context. getcontext, getcontext(2) PRM
 user contexts. makecontext, makecontext(3C) PRM
 user crontab file. crontab(1) URM
 user. cuserid: cuserid(3S) PRM
 user data from the message/ mps_get_soldata(D3D) DDRM
 user data from the message/ mps_get_unsoldata(D3D) DDRM
 User Datagram Protocol. UDP(7) NI
 User Datagram Protocol. UDP(7) NUAG
 User Datagram Protocol. UDP(7) SARM
 user, effective user, real/ getuid(2) PRM
 user encryption key. chkey(1) NI
 user encryption key. chkey(1) NUAG
 user environment. environ(5) NI
 user environment. environ(5) PRM
 user environment. environ(5) SARM
 user. /establish a connection t_connect(3N) NI
 user format date and time. getdate(3C) PRM
 user. groups: groups(1) URM
 user has access to memory. useracc(D3DK) DDRM
 user. hosts.equiv, rhosts: hosts.equiv(4) NUAG
 user. hosts.equiv, rhosts: hosts.equiv(4) SARM
 user ID. ckuid: ckuid(1) SARM
 user ID. ckuid: ckuid(1) SS
 user ID. diskusg: diskusg(1M) SARM
 user IDs. setreuid(3) BSD
 user information server. fingerd(1M) NUAG
 user information server. fingerd(1M) SARM
 user interface to a remote telnet(1) NUAG
 user interface to a remote telnet(1) URM
 user limits. ulimit(2) PRM
 user login information. listusers(1) URM
 user login on the system. useradd(1M) SARM
 user name and ID, and group name id(1M) SARM
 user name and ID, and group name id(1M) URM
 user name directory service. whois(1) NUAG

whois: Internet user name directory service. whois(1) URM
 displays a list of all valid user names. dispuid: dispuid(1) SARM
 displays a list of all valid user names. dispuid: dispuid(1) SS
 notify: notify user of the arrival of new mail. notify(1) URM
 last: indicate last user or terminal logins. last(1) URM
 copy data from a driver to a user program. copyout: copyout(D3DK) DDRM
 copyin: copy data from a user program to a driver buffer. copyin(D3DK) DDRM
 edquota: edit user quotas. edquota(1M) SARM
 group/ /get real user, effective user, real group, and effective getuid(2) PRM
 initialize system for first user. setup: setup(1M) SARM
 endusershell: get legal user shells. /setusershell, getusershell(3) BSD
 segmap: map device memory into user space. segmap(D2K) DDRM
 in the utmp file of the current user. ttyslot: find the slot ttyslot(3C) PRM
 for/ /netname2host, netname2user, user2netname: library routines secure_rpc(3N) NI
 access to memory. useracc: verify whether user has useracc(D3DK) DDRM
 login on the system. useradd: administer a new user useradd(1M) SARM
 from the system. userdel: delete a user's login userdel(1M) SARM
 request. t_snddis: send user-initiated disconnect t_snddis(3N) NI
 information on the system. usermod: modify a user's login usermod(1M) SARM
 rpc.rusersd: network username server. rusersd(1M) NI
 rpc.rusersd: network username server. rusersd(1M) NUAG
 display the effective current username. whoami: whoami(1) BSD
 .environ, .pref, .variables: user-preference variable files/ environ(4) SARM
 wall: write to all users. wall(1M) SARM
 quota: display a user's disk quota and usage. quota(1M) SARM
 users logged in. users: display a compact list of users(1) BSD
 about local and remote users. /display information finger(1) NUAG
 about local and remote users. /display information finger(1) URM
 editor (variant of ex for casual users). edit: text edit(1) URM
 groups: display a user's group memberships. groups(1) BSD
 users: display a compact list of users logged in. users(1) BSD
 userdel: delete a user's login from the system. userdel(1M) SARM
 system. usermod: modify a user's login information on the usermod(1M) SARM
 rmail: read mail or send mail to users. mail, mail(1) URM
 maillock: manage lockfile for user's mailbox. maillock(3X) PRM
 rusers: return information about users on remote machines. rusers(3N) NI
 rwall: write to all users over a network. rwall(1M) NI
 rwall: write to all users over a network. rwall(1M) NUAG
 fuser: identify processes using a file or file structure. fuser(1M) SARM
 /allocate a message block using a shared buffer. esballoc(D3DK) DDRM
 elf_end: finish using an object file. elf_end(3E) PRM
 data storage structure for I/O using DMA. dma_buf: dma_buf(D4D) DDRM
 data storage structure for I/O using DMA. dma_cb: dma_cb(D4D) DDRM
 /search a file for a pattern using full regular expressions. egrep(1) URM
 /search a file for a pattern using full regular expressions. egrep(1) XNX
 shell: run a command using shell. shell(1F) CHAR
 /send SMTP mail to a remote host using Simple Mail Transfer/ smtp(1M) SARM

interface to a remote system
 interface to a remote system
 data storage structure for I/O
 uiomove: copy kernel data
 interval in microseconds.
 statistics.
 /icswr, icsslot, icsgetrec:
 /f4diskadd/fl: disk set up
 disksetup: disk set up
 edvtoc: VTOC editing
 iconv: code set conversion
 mkpart: disk maintenance
 flushinp: miscellaneous curses
 get information about resource
 modification times.
 utmp, wtmp:
 utmp, wtmp:
 endutent, utmpname: access
 tyslot: find the slot in the
 formats.
 formats.
 /pututline, setutent, endutent,
 utmpx, wtmpx:
 utmpx, wtmpx:
 updwtmp, updwtmpx: access
 entry formats.
 entry formats.
 updwtmp,/ /setutxent, endutxent,
 directories and permissions/
 for the uucp system.
 clean-up.
 file. uuccheck: check the
 uusched: the scheduler for the
 uucleanup:
 control. uustat:
 file transport program for the
 UNIX-to-UNIX system copy.
 or decode its ASCII/ uuencode,
 binary file, or decode its/
 service grades that are/
 system copy. uucp,
 copy. uucp, uulog,
 system file copy. uuto,
 uucp file transport program.
 job control.
 UNIX-to-UNIX system file copy.

using the. telnet: user telnet(1) NUAG
 using the. telnet: user telnet(1) URM
 using uio(D4DK). iovec: iovec(D4DK) DDRM
 using uio(D4DK) structure. uiomove(D3DK) DDRM
 usleep: suspend execution for usleep(3) BSD
 ustat: get file system ustat(2) PRM
 utilities to access MULTIBUS II/ icsrd(1M) MBRMAN
 utility. diskadd(1M) SARM
 utility. disksetup(1M) SARM
 utility. edvtoc(1M) SARM
 utility. iconv(1) URM
 utility. mkpart(1M) SARM
 utility routines. /delay_output, curs_util(3X) CHAR
 utilization. getrusage: getrusage(3) BSD
 utime: set file access and utime(2) PRM
 utimes: set file times. utimes(3) BSD
 utmp and wtmp entry formats. utmp(4) PRM
 utmp and wtmp entry formats. utmp(4) SARM
 utmp file entry. /setutent, getut(3C) PRM
 utmp file of the current user. ttyslot(3C) PRM
 utmp, wtmp: utmp and wtmp entry utmp(4) PRM
 utmp, wtmp: utmp and wtmp entry utmp(4) SARM
 utmpname: access utmp file/ getut(3C) PRM
 utmpx and wtmpx entry formats. utmpx(4) PRM
 utmpx and wtmpx entry formats. utmpx(4) SARM
 utmpx file entry. /getutmpx, getutx(3C) PRM
 utmpx, wtmpx: utmpx and wtmpx utmpx(4) PRM
 utmpx, wtmpx: utmpx and wtmpx utmpx(4) SARM
 utmpxname, getutmp, getutmpx, getutx(3C) PRM
 uuccheck: check the uucp uuccheck(1M) SARM
 uuico: file transport program uuico(1M) SARM
 uucleanup: uucp spool directory uucleanup(1M) SARM
 uucp directories and permissions uuccheck(1M) SARM
 uucp file transport program. uusched(1M) SARM
 uucp spool directory clean-up. uucleanup(1M) SARM
 uucp status inquiry and job uustat(1C) URM
 uucp system. uuico: uuico(1M) SARM
 uucp, uulog, uuname: uucp(1C) URM
 uuencode: encode a binary file, uuencode(1C) URM
 uuencode, uuencode: encode a uuencode(1C) URM
 uuglist: print the list of uuglist(1C) URM
 uulog, uuname: UNIX-to-UNIX uucp(1C) URM
 uuname: UNIX-to-UNIX system uucp(1C) URM
 uupick: public UNIX-to-UNIX uuto(1C) URM
 uusched: the scheduler for the uusched(1M) SARM
 uustat: uucp status inquiry and uustat(1C) URM
 uuto, uupick: public uuto(1C) URM

system with debugging on. execution. requests. a uio structure. (ots) device/ ots: System compatibility module. ttcompat: to incoming mail messages. automatically.

ckdate, errdate, helpdate, ckdate, errdate, helpdate, ckgid, errgid, helpgid, ckgid, errgid, helpgid, dispgid: displays a list of all dispgid: displays a list of all dispuid: displays a list of all dispuid: displays a list of all val: request. physiock: valdate: prompts for and valdate: prompts for and helpgid, valgid: prompts for and helpgid, valgid: prompts for and ckkeywd: prompts for and ckkeywd: prompts for and ckuid: prompts for and ckuid: prompts for and ckrange: prompts for and ckrange: prompts for and ckyorn: prompts for and ckyorn: prompts for and field_arg: forms field data type /free, realloc, calloc, memalign, elf_hash: compute hash labs: return integer absolute verify and return an integer verify and return an integer decimal record to floating-point getenv: return ceiling, remainder, absolute register of/ ics_write: writes a readlink: read the idtune: attempts to set getitimer, setitimer: get/set YP map. ypmatch: print the driver identification and limit /convert floating-point

Uutry: try to contact remote Uutry(1M) SARM
 uux: UNIX-to-UNIX system command uux(1C) URM
 uuxqt: execute remote command uuxqt(1M) SARM
 uwritec: remove a character from uwritec(D3DK) DDRM
 V/386 OSI Transport Service ots(1M) MBRMAN
 V7, 4BSD and XENIX STREAMS ttcompat(7) SARM
 vacation: automatically respond vacation(1) URM
 vacation: reply to mail vacation(1) BSD
 val: validate an SCCS file. val(1) PRM
 valdate: prompts for and/ ckdate(1) SARM
 valdate: prompts for and/ ckdate(1) SS
 valgid: prompts for and/ ckgid(1) SARM
 valgid: prompts for and/ ckgid(1) SS
 valid group names. dispgid(1) SARM
 valid group names. dispgid(1) SS
 valid user names. dispuid(1) SARM
 valid user names. dispuid(1) SS
 validate an SCCS file. val(1) PRM
 validate and issue raw I/O physiock(D3D) DDRM
 validates a date. /helpdate, ckdate(1) SARM
 validates a date. /helpdate, ckdate(1) SS
 validates a group id. /errgid, ckgid(1) SARM
 validates a group id. /errgid, ckgid(1) SS
 validates a keyword. ckkeywd(1) SARM
 validates a keyword. ckkeywd(1) SS
 validates a user ID. ckuid(1) SARM
 validates a user ID. ckuid(1) SS
 validates an integer. ckrange(1) SARM
 validates an integer. ckrange(1) SS
 validates yes/no. ckyorn(1) SARM
 validates yes/no. ckyorn(1) SS
 validation. /field_type, form_field_validation(3X) CHAR
 valloc,: memory allocator. malloc(3C) PRM
 value. elf_hash(3E) PRM
 value. abs, abs(3C) PRM
 value. ckint: display a prompt; ckint(1) SARM
 value. ckint: display a prompt; ckint(1) SS
 value. /convert decimal_to_floating(3) BSD
 value for environment name. getenv(3C) PRM
 value functions. /floor, floor(3M) PRM
 value into the specified ics_write(D3D) DDRM
 value of a symbolic link. readlink(2) PRM
 value of a tunable parameter. idtune(1M) SARM
 value of interval timer. getitimer(3C) PRM
 value of one or more keys from a ypmatch(1) NI
 value structure. /STREAMS module_info(D4DK) DDRM
 value to decimal record. floating_to_decimal(3) BSD

putenv: change or add value to environment. putenv(3C) PRM
 u370: get processor type true value. /u3b5, u3b15, vax, 386, machid(1) URM
 true, false: provide truth values. true(1) URM
 values: machine-dependent values. values(5) PRM
 /htons, ntohl, ntohs: convert values between host and network/ byteorder(3N) NI
 ypcat: print values in a YP data base. ypcat(1) NI
 set and get menus item values. /item value: menu_item_value(3X) CHAR
 values: machine-dependent values(5) PRM
 displays package parameter values. pkgparam: pkgparam(1) SARM
 displays package parameter values. pkgparam: pkgparam(1) SS
 form field to its default argument list. /reset the current reset(1F) CHAR
 stdarg: handle varargs: handle variable varargs(5) PRM
 varargs: handle variable argument list. stdarg(5) PRM
 variable argument list. varargs(5) PRM
 /print formatted output of a variable argument list. vprintf(3S) PRM
 /.variables: user-preference variable files for AT&T FACE. environ(4) SARM
 variables. sysconf(3C) PRM
 sysconf: get configurable system variables currently set. printenv(1) BSD
 printenv: display environment variables. fpathconf, pathconf: fpathconf(2) PRM
 get configurable pathname variables. /unset: set and unset set(1F) CHAR
 local or global environment .variables: user-preference environ(4) SARM
 variable files/ .environ, .pref (variant of ex for casual edit(1) URM
 users). edit: text editor various plotters. /t300s, t4013, plot(1G) BSD
 t450, tek: graphics filters for vax, 386, u370: get processor/ machid(1) URM
 /pdp11, u3b, u3b2, u3b5, u3b15, vc: version control. vc(1) PRM
 get option letter from argument vector. getopt: getopt(3C) PRM
 getvol: verifies device accessibility. getvol(1M) SARM
 ckpath: display a prompt; verify and return a pathname. ckpath(1) SARM
 ckpath: display a prompt; verify and return a pathname. ckpath(1) SS
 ckstr: display a prompt; verify and return a string/ ckstr(1) SARM
 ckstr: display a prompt; verify and return a string/ ckstr(1) SS
 cktime: display a prompt; verify and return a time of day. cktime(1) SARM
 cktime: display a prompt; verify and return a time of day. cktime(1) SS
 value. ckint: display a prompt; verify and return an integer ckint(1) SARM
 value. ckint: display a prompt; verify and return an integer ckint(1) SS
 ckperms: set and/or verify permissions on files. ckperms(1M) MBRMAN
 assert: verify program assertion. assert(3X) PRM
 to memory. useracc: verify whether user has access useracc(D3DK) DDRM
 vc: version control. vc(1) PRM
 server/ yppoll: return current version of a YP map at a YP yppoll(1M) NI
 get: get a version of an SCCS file. get(1) PRM
 ELF library and application versions. /coordinate elf_version(3E) PRM
 compver: compatible versions file. compver(4) SS
 diffmk: mark differences between versions of a troff input file. diffmk(1) BSD
 scsdiff: compare two versions of an SCCS file. scsdiff(1) PRM
 ncheck (s5): generate path names versus i-numbers for s5 file/ ncheck(1M) SARM
 ncheck (ufs): generate pathnames versus i-numbers for ufs file/ ncheck(1M) SARM

curses borders, horizontal and virtual memory efficient way.
 /fprintf, sprintf, vprintf, formatted output of a/ vprintf, getvfsspec, getvfssany: get defaults.
 create a tags file for use with display editor based on ex. get client's data passed /tparm, tputs, putp, vidputs, /restartterm, tparm, tputs, putp, physical/ physmap: obtain bp_mapin: allocate bp_mapout: deallocate kvtophys: convert kernel sdi_translate: translate scb mapped device. mmap: check vfork: spawn new process in a echo: put string on move a panels window on the /update_panels: panels vtop: convert tell if menus item is ex. vi: screen-oriented system administration. sysadm: copy of file system. copy of an s5 file system. copy of a ufs file system. format of the bfs file system (ufs): format of ufs file system out file archives to multiple move an archive from one set of printf, fprintf, sprintf, print formatted output of a/ generate a list of path names with the controlling FMLI/ /sprintf, vprintf, fprintf, of a/ vprintf, fprintf, edvtoc: prtvtoc: print the physical address. /wprintw, mvprintw, mvwprintw, from/ /wscanw, mvscanw, mvwscanw, are they doing. echochar,/ curs addch: addch, /addchstr, addchnstr, waddchstr, /addchstr, addchnstr, vertical lines. /wvline: create curs_border(3X) CHAR vfork: spawn new process in a vfork(2) PRM vfprintf, vsprintf: formatted/ printf(3S) BSD vprintf, vsprintf: print vprintf(3S) PRM vfstab file entry. /getvfssfile, getvfssent(3C) PRM vfstab: table of file system vfstab(4) SARM vi. ctags: ctags(1) URM vi: screen-oriented (visual) vi(1) URM via the listener. nlsgetcall: nlsgetcall(3N) NI vidattr, mvcur, tigetflag,/ curs_terminfo(3X) CHAR vidputs, vidattr, mvcur,/ curs_terminfo(3X) CHAR virtual address mapping for physmap(D3DK) DDRM virtual address space. bp_mapin(D3DK) DDRM virtual address space. bp_mapout(D3DK) DDRM virtual address to physical/ kvtophys(D3D) DDRM virtual addresses. sdi_translate(D3I) SCSI virtual mapping for memory mmap(D2K) DDRM virtual memory efficient way. vfork(2) PRM virtual output. echo(1F) CHAR virtual screen. /move_panel: panel_move(3X) CHAR virtual screen refresh routine. panel_update(3X) CHAR virtual to physical address. vtop(D3D) DDRM visible. /item_visible: menu_item_visible(3X) CHAR (visual) display editor based on vi(1) URM visual interface to perform sysadm(1M) SARM volcopy (generic): make literal volcopy(1M) SARM volcopy (s5): make a literal volcopy(1M) SARM volcopy (ufs): make a literal volcopy(1M) SARM volume. fs (bfs): fs(4) SARM volume. fs fs(4) SARM volumes. cpout: copy cpout(1M) MBRMAN volumes to another. migration: migration(1M) SARM vprintf, fprintf, vsprintf:/ printf(3S) BSD vprintf, fprintf, vsprintf: vprintf(3S) PRM vs l-numbers. ncheck (generic): ncheck(1M) SARM vsig: synchronize a co-process vsig(1F) CHAR vsprintf: formatted output/ printf(3S) BSD vsprintf: print formatted output vprintf(3S) PRM VTOC editing utility. edvtoc(1M) SARM VTOC of a block device. prtvtoc(1M) SARM vtop: convert virtual to vtop(D3D) DDRM vwprintw: print formatted output/ curs_printw(3X) CHAR vwscanw: convert formatted input curs_scanw(3X) CHAR w: who is logged in, and what w(1) BSD waddch, mvaddch, mvwaddch, curs_addch(3X) CHAR waddchnstr, mvaddchstr,/ curs_addchstr(3X) CHAR waddchstr, waddchnstr,/ curs_addchstr(3X) CHAR

/addstr, addnstr, waddstr, waddnstr, mvaddstr, mvaddnstr,/ curs_addstr(3X) CHAR
 curs_addstr: addstr, addnstr, waddstr, waddnstr, mvaddstr,/ curs_addstr(3X) CHAR
 process. wait(1) URM
 rmsetwant: set the map's wait flag for a wakeup. rmsetwant(D3DK) DDRM
 state. waitid: wait for child process to change waitid(2) PRM
 state. waitpid: wait for child process to change waitpid(2) PRM
 or terminate. wait: wait for child process to stop wait(2) PRM
 rmwant: wait for free memory. rmwant(D3DK) DDRM
 release blocked signals and wait for interrupt. /automatically sigpause(3) BSD
 stop. /WIFSIGNALED, WIFEXITED: wait for process to terminate or wait(3) BSD
 wstat: wait status. wstat(5) PRM
 stop or terminate. wait: wait for child process to wait(2) PRM
 WIFSIGNALED, WIFEXITED: wait/ wait, wait3, WIFSTOPPED, wait(3) BSD
 WIFEXITED: wait for/ wait, wait3, WIFSTOPPED, WIFSIGNALED, wait(3) BSD
 to change state. waitid: wait for child process waitid(2) PRM
 sigsem: signal a process waiting on a semaphore. sigsem(2) XNX
 to change state. waitpid: wait for child process waitpid(2) PRM
 check access to a resource/ waitsem, nbwaitsem: await and waitsem(2) XNX
 buffer after block I/O and wakeup processes. /release biodone(D3DK) DDRM
 execution. wakeup: resume suspended process wakeup(D3DK) DDRM
 set the map's wait flag for a wakeup. rmsetwant: rmsetwant(D3DK) DDRM
 ftw, nftw: walk a file tree. ftw(3C) PRM
 wall: write to all users. wall(1M) SARM
 wattroff, attron, wattron, curs_attr(3X) CHAR
 wattron, attrset, wattrset,/ curs_attr(3X) CHAR
 wattrset, standend, wstandend,/ curs_attr(3X) CHAR
 way. vfork: spawn new process vfork(2) PRM
 /bkgdset, wbgkdset, bkgd, wbgkdset, bkgd, wbgkd: curses window background/ curs_bkgd(3X) CHAR
 window/ curs_bkgd: bkgdset, wborder, box, whline, wvline: curs_bkgd(3X) CHAR
 create/ curs_border: border, wc: word count. wc(1) URM
 wclear, clrtoeol, wclrtoeol, curs_clear(3X) CHAR
 wclrtoeol, clrtoeol, wclrtoeol:/ curs_clear(3X) CHAR
 wclrtoeol: clear all or part of/ curs_clear(3X) CHAR
 functions. mbstring: multibyte string mbstring(3C) PRM
 mbchar: multibyte character/ mbchar(3C) PRM
 mblink: multibyte character/ mblink(3C) PRM
 wcursor: create curs_window(3X) CHAR
 curses/ /dupwin, wsyncup, syncok, wd: Western Digital 8003 Adapter wd(7) SARM
 Board. delete/ curs_delch: delch, wdelch, mvdelch: curs_delch(3X) CHAR
 curs_deleteln: deleteln, wdeleteln, insdelln,/ curs_deleteln(3X) CHAR
 /mvaddch, mvwaddch, echochar, wechochar: add a character (with/ curs_addch(3X) CHAR
 wclrtoeol,/ curs_clear: erase, werase, clear, wclear, clrtoeol, curs_clear(3X) CHAR
 Board. wd: Western Digital 8003 Adapter wd(7) SARM
 ungetch:/ curs_getch: getch, wgetch, mvgetch, mvwgetch, curs_getch(3X) CHAR
 /wgetstr, mvgetstr, mvwgetstr, wgetnstr: get character strings/ curs_getstr(3X) CHAR
 wgetstr, mvgetstr, mvwgetstr, curs_getstr(3X) CHAR
 whodo: who is doing what. whodo(1M) SARM

strings.
summary about a keyword.
solicited data in fragments
encrypted. isencrypt: determine
message. datams: test
binary/ ckbinarsys: determine
memory. useracc: verify
its pathname or alias.
borders,/ /border, wborder, box,

current username.

directory service.
directory service.
machines. rusers:
machines. rusers:
machines. rwho:
machines. rwho:
formatted input from a curses
/wait3, WIFSTOPPED, WIFSIGNALED,
wait, wait3, WIFSTOPPED,
WIFEXITED: wait/ wait, wait3,
character and/ curs_inch: inch,
/inchstr, inchstr, winchstr,
mvinchnstr,/ /inchstr, inchstr,
(and attributes) from a curses
/(with attributes) to a curses
string of characters to a curses
/form_sub, scale_form: forms
/menu_sub, scale_menu: menus
/wstandout: curses character and
/wbkgdset, bkgd, wbkgd: curses
under the cursor in a curses
clear all or part of a curses
getmaxyx: get curses cursor and
move, wmove: move curses
pos_form_cursor: position forms
srcl, wsrl: scroll a curses
under cursor in a curses
and its attributes from a curses
of characters from a curses
/get or set the current
/move_panel: move a panels
under the cursor in a curses
(and attributes) to a curses
and insert lines in a curses
jagent: host control of

what: print identification what(1) PRM
whatis: display a one-line whatis(1) BSD
when. /receives mps_AMPreceive_frag(D3D) DDRM
whether a character buffer is isencrypt(3G) PRM
whether a message is a data datams(D3DK) DDRM
whether remote system can accept ckbinarsys(1M) SARM
whether user has access to useracc(D3DK) DDRM
which: locate a command; display which(1) BSD
whline, wvline: create curses curs_border(3X) CHAR
who: who is on the system. who(1) URM
whoami: display the effective whoami(1) BSD
whodo: who is doing what. whodo(1M) SARM
whois: Internet user name whois(1) NUAG
whois: Internet user name whois(1) URM
who's logged in on local rusers(1) NI
who's logged in on local rusers(1) NUAG
who's logged in on local rwho(1) NUAG
who's logged in on local rwho(1) URM
widow. /vwscanw: convert curs_scanw(3X) CHAR
WIFEXITED: wait for process to/ wait(3) BSD
WIFSIGNALED, WIFEXITED: wait for/ wait(3) BSD
WIFSTOPPED, WIFSIGNALED, wait(3) BSD
winch, mvinch, mvwinch: get a curs_inch(3X) CHAR
winchnstr, mvinchnstr,/ curs_inchstr(3X) CHAR
winchstr, winchnstr, mvinchnstr, curs_inchstr(3X) CHAR
window. /a string of characters curs_inchstr(3X) CHAR
window and advance cursor. curs_addch(3X) CHAR
window and advance cursor. /a curs_addstr(3X) CHAR
window and subwindow association/ form_win(3X) CHAR
window and subwindow association/ menu_win(3X) CHAR
window attribute control/ curs_attr(3X) CHAR
window background manipulation/ curs_bkgd(3X) CHAR
window. /before the character curs_insch(3X) CHAR
window. /clrtoeol, wclrtoeol: curs_clear(3X) CHAR
window coordinates. /getbegyx, curs_getyx(3X) CHAR
window cursor. curs_move: curs_move(3X) CHAR
window cursor. form_cursor: form_cursor(3X) CHAR
window. curs_scroll: scroll, curs_scroll(3X) CHAR
window. /delete character curs_delch(3X) CHAR
window. /get a character curs_inch(3X) CHAR
window. /mvwinstr: get a string curs_instr(3X) CHAR
window of a panels panel. panel_window(3X) CHAR
window on the virtual screen. panel_move(3X) CHAR
window. /string before character curs_insstr(3X) CHAR
window. /string of characters curs_addchstr(3X) CHAR
window. /winsertln: delete curs_deleteln(3X) CHAR
windowing terminal. jagent(5) PRM

jagent: host control of	jagent(5) SARM
jterm: reset layer of	jterm(1) URM
library. libwindows:	libwindows(3X) PRM
ismpx: return	ismpx(1) URM
/protocol used between host and	layers(5) PRM
/protocol used between host and	layers(5) SARM
layers: layer multiplexor for	layers(1) URM
multiplexed tty driver for AT&T	xt: STREAMS-based xt(7) SARM
multiplexed tty driver for AT&T	xt: STREAMS-based xt(7) STRM
wredrawln: refresh curses	./redrawwin, curs_refresh(3X) CHAR
and manipulate overlapped curses	./copywin: overlap curs_overlay(3X) CHAR
print formatted output in curses	./mvwprintw, vwprintw: curs_printw(3X) CHAR
wsyncdown: create curses	./syncok, wcursyncup, curs_window(3X) CHAR
./instr, innstr, winstr,	winnstr, mvinnstr, mvinnstr,/ curs_instr(3X) CHAR
insert a/ curs_insch: insch,	winsch, mvinsch, mvwinsch: curs_insch(3X) CHAR
/deleteln, wdeleteln, insdelln,	winsdelln, insertln,/ curs_deleteln(3X) CHAR
/insdelln, winsdelln, insertln,	winsertln: delete and insert/ curs_deleteln(3X) CHAR
./insstr, insnstr, winsstr,	winsnstr, mvinsstr, mvinsnstr,/ curs_insnstr(3X) CHAR
curs_instr: insstr, insnstr,	winsstr, winsnstr, mvinsstr,/ curs_insnstr(3X) CHAR
curs_instr: instr, innstr,	winstr, winsnstr, mvinstr,/ curs_instr(3X) CHAR
/wechochar: add a character	(with attributes) to a curses/ curs_addch(3X) CHAR
prof: profile	within a function. prof(5) PRM
/fasthalt: reboot/halt the system	without checking the disks. fastboot(1M) BSD
cursor. curs_move: move,	wmove: move curses window curs_move(3X) CHAR
redrawwin,/ /refresh, wrefresh,	wnoutrefresh, douupdate, curs_refresh(3X) CHAR
inw: read a 16 bit short	word from a 16-bit I/O port. inw(D3D) DDRM
inl: read a 32-bit	word from a 32-bit I/O port. inl(D3D) DDRM
wc:	word count. wc(1) URM
fgetc, getw: get character or	word from a stream. /getchar,getc(3S) PRM
fputc, putw: put character or	word on a stream. /putchar,putc(3S) PRM
outw: write a 16-bit short	word to a 16-bit I/O port. outw(D3D) DDRM
outl: write a 32-bit long	word to a 32-bit I/O port. outl(D3D) DDRM
repinsd: read 32-bit	words from I/O port to buffer. repinsd(D3D) DDRM
port. repoutsd: write 32-bit	words from buffer to an I/O repoutsd(D3D) DDRM
port. repoutsw: write 16-bit	words from buffer to an I/O repoutsw(D3D) DDRM
repinsw: read 16-bit	words from I/O port to buffer. repinsw(D3D) DDRM
or lines in a sorted/ look: find	words in the system dictionary look(1) BSD
cd: change	working directory. cd(1) URM
chdir, fchdir: change	working directory. chdir(2) PRM
getcwd: get pathname of current	working directory. getcwd(3C) PRM
pwd:	working directory name. pwd(1) URM
getwd: get current	working directory pathname. getwd(3) BSD
/specific alarms and/or the	"working" indicator. indicator(1F) CHAR
vwprintw:/ curs_printw: printw,	wprintw, mvprintw, mvwprintw, curs_printw(3X) CHAR
queue for this module or/	WR: get pointer to the write WR(D3DK) DDRM
windows/ /douupdate, redrawwin,	wredrawln: refresh curses curs_refresh(3X) CHAR
curs_refresh: refresh,	wrefresh, wnoutrefresh,/ curs_refresh(3X) CHAR

to an I/O port. repoutsw: write 16-bit words from buffer repoutsw(D3D) DDRM
to an I/O port. repoutsd: write 32-bit words from buffer repoutsd(D3D) DDRM
16-bit I/O port. outw: write a 16-bit short word to a outw(D3D) DDRM
32-bit I/O port. outl: write a 32-bit long word to a outl(D3D) DDRM
port. outb: write a byte to an 8-bit I/O outb(D3D) DDRM
(to) a/ /scr_init, scr_set: read (write) a curses screen from curs_scr_dump(3X) CHAR
I/O port. repoutsb: write bytes from buffer to an repoutsb(D3D) DDRM
write: write data to a device. write(D2DK) DDRM
write, writev: write on a file. write(2) PRM
/post_form, unpost_form: write or erase forms from/ form_post(3X) CHAR
/post_menu, unpost_menu: write or erase menus from/ menu_post(3X) CHAR
putpwent: write password file entry. putpwent(3C) PRM
driver. WR: get pointer to the write queue for this module or WR(D3DK) DDRM
entry. putspent: write shadow password file putspent(3C) PRM
list.. mdl: read and optionally write the manufacturer's defect mdl(1M) MBRMAN
wall: write to all users. wall(1M) SARM
network. rwall: write to all users over a rwall(1M) NI
network. rwall: write to all users over a rwall(1M) NUAG
write: write to another user. write(1) URM
machines. rwall: write to specified remote rwall(3N) NI
write: write data to a device. write(D2DK) DDRM
write: write to another user. write(1) URM
write, writev: write on a file. write(2) PRM
writes a specified number of/ ics_rdwr(D3D) DDRM
ics_rdwr: reads or writes a value into the ics_write(D3D) DDRM
specified register/ ics_write: writes into interconnect ics_write(1M) MBRMAN
registers of the/ ics_write: writev: write on a file. write(2) PRM
write: write on a file. write(2) PRM
open: open for reading or writing. open(2) PRM
a file region for reading or writing. /lock or unlock locking(2) XNX
vwscanw:/ curs_scanw: scanw, wscanw, mvscanw, mvwscanw, curs_scanw(3X) CHAR
curs_scroll: scroll, srcl, wscl: scroll a curses window. curs_scroll(3X) CHAR
/immedok, leaveok, setscreg, wsetscrrg, scrollok, nl, nonl:/ curs_outopts(3X) CHAR
/attrset, wattrset, standend, wstandend, standout, wstandout:/ curs_attr(3X) CHAR
/standend, wstandend, standout, wstandout: curses character and/ curs_attr(3X) CHAR
wstat: wait status. wstat(5) PRM
/wsyncup, syncok, wcursyncup, wsynccount: create curses/ curs_window(3X) CHAR
/derwin, mvderwin, dupwin, wsynccount, syncok, wcursyncup,/ curs_window(3X) CHAR
/nowiflush, qiflush, timeout, wtimeout, typeahead: curses/ curs_inopts(3X) CHAR
the 5620 DMD terminal. wtinit: object downloader for wtinit(1M) SARM
utmp, wtmp: utmp and wtmp entry formats. utmp(4) PRM
utmp, wtmp: utmp and wtmp entry formats. utmp(4) SARM
formats. utmp, wtmp: utmp and wtmp entry utmp(4) PRM
formats. utmp, wtmp: utmp and wtmp entry utmp(4) SARM
accounting records. fwtmp, wtmpfix: manipulate connect fwtmp(1M) SARM
utmpx, wtmpx: utmpx and wtmpx entry formats. utmpx(4) PRM
utmpx, wtmpx: utmpx and wtmpx entry formats. utmpx(4) SARM
formats. utmpx, wtmpx: utmpx and wtmpx entry utmpx(4) PRM

formats. utmpx, wtmpx: utmpx and wtmpx entry utmpx(4) SARM
 /touchwin, touchline, untouchwin, wtouchln, is_linetouched,/ curs_touch(3X) CHAR
 /border, wborder, box, whline, wvline: create curses borders,/ curs_border(3X) CHAR
 list(s) and execute command. xargs: construct argument xargs(1) URM
 external data representation. xdr: library routines for xdr(3N) NI
 procedure calls. /xdr_replymsg: XDR library routines for remote rpc_xdr(3N) NI
 xdr_authsys_parms,/ rpc_xdr: xdr_accepted_reply, rpc_xdr(3N) NI
 xdr_inline, xdrrec_eof,/ xdr_admin: xdr_getpos, xdr_admin(3N) NI
 xdr_opaque,/ xdr_complex: xdr_array, xdr_bytes, xdr_complex(3N) NI
 rpc_xdr: xdr_accepted_reply, xdr_authsys_parms, xdr_callhdr,/ rpc_xdr(3N) NI
 xdr_enum,/ xdr_simple: xdr_bool, xdr_char, xdr_double, xdr_simple(3N) NI
 xdr_complex: xdr_array, xdr_bytes, xdr_opaque,/ xdr_complex(3N) NI
 /xdr_authsys_parms, xdr_callhdr, xdr_callmsg,/ rpc_xdr(3N) NI
 /xdr_authsys_parms, xdr_callhdr, xdr_callmsg, xdr_opaque_auth,/ rpc_xdr(3N) NI
 xdr_simple: xdr_bool, xdr_char, xdr_double, xdr_enum,/ xdr_simple(3N) NI
 xdr_bytes, xdr_opaque,/ xdr_complex: xdr_array, xdr_complex(3N) NI
 xdrmem_create, xdrrec_create,/ xdr_create: xdr_destroy, xdr_create(3N) NI
 xdrrec_create,/ xdr_create: xdr_destroy, xdrmem_create, xdr_create(3N) NI
 xdr_free,/ /xdr bool, xdr char, xdr_double, xdr_enum, xdr_float, xdr_simple(3N) NI
 /xdr_bool, xdr_char, xdr_double, xdr_enum, xdr_float, xdr_free, xdr_int,/ xdr_simple(3N) NI
 /xdr_double, xdr_enum, xdr_float, xdr_free, xdr_int, xdr_long,/ xdr_simple(3N) NI
 xdrrec_eof,/ xdr_admin: xdr_getpos, xdr_inline, xdr_admin(3N) NI
 xdr_admin: xdr_getpos, xdr_inline, xdrrec_eof,/ xdr_admin(3N) NI
 /xdr_enum, xdr float, xdr free, xdr_int, xdr long, xdr short,/ xdr_simple(3N) NI
 /xdr_float, xdr_free, xdr_int, xdr_long, xdr_short, xdr_u_char,/ xdr_simple(3N) NI
 xdr_create: xdr_destroy, xdrmem_create, xdrrec_create,/ xdr_create(3N) NI
 /xdr_array, xdr_bytes, xdr_opaque, xdr_pointer,/ xdr_complex(3N) NI
 /xdr_callhdr, xdr_callmsg, xdr_opaque_auth,/ rpc_xdr(3N) NI
 /xdr_bytes, xdr_opaque, xdr_pointer, xdr_reference,/ xdr_complex(3N) NI
 /xdr_destroy, xdrmem_create, xdrrec_create, xdrstdio_create:/ xdr_create(3N) NI
 /xdr_getpos, xdr_inline, xdrrec_eof, xdr_setpos: library/ xdr_admin(3N) NI
 /xdr_opaque, xdr_pointer, xdr_reference, xdr_string,/ xdr_complex(3N) NI
 /xdr_callmsg, xdr_opaque_auth, xdr_rejected_reply,/ rpc_xdr(3N) NI
 routines/ /xdr_rejected_reply, xdr_repliedmsg: XDR library rpc_xdr(3N) NI
 /xdr_inline, xdrrec_eof, xdr_setpos: library routines for/ xdr_admin(3N) NI
 /xdr_free, xdr int, xdr long, xdr_short, xdr_u_char,/ xdr_simple(3N) NI
 xdr_char, xdr_double, xdr_enum,/ xdr_simple: xdr_bool, xdr_simple(3N) NI
 /xdrmem_create, xdrrec_create, xdrstdio_create: library/ xdr_create(3N) NI
 /xdr_pointer, xdr_reference, xdr_string, xdr_union,/ xdr_complex(3N) NI
 /xdr_int, xdr_long, xdr_short, xdr_u_char, xdr_u_long,/ xdr_simple(3N) NI
 /xdr_long, xdr_short, xdr_u_char, xdr_u_long, xdr_u_short,/ xdr_simple(3N) NI
 /xdr_reference, xdr_string, xdr_union, xdr_vector,/ xdr_complex(3N) NI
 /xdr_u_char, xdr_u long, xdr_u short, xdr_u short, xdr_void: library/ xdr_simple(3N) NI
 library/ /xdr_string, xdr_union, xdr_vector, xdr_wrapstring: xdr_complex(3N) NI
 /xdr_u long, xdr_u short, xdr_void: library routines for/ xdr_simple(3N) NI
 for/ /xdr_union, xdr_vector, xdr_wrapstring: library routines xdr_complex(3N) NI

fixperm: correct or initialize	XENIX file permissions and/	fixperm(1M) SARM
xfscck: check and repair	XENIX filesystems.	xfscck(1M) SARM
xrestore, xrestor: invoke	XENIX incremental filesystem/	xrestore(1M) SARM
xinstall:	XENIX installation shell script.	xinstall(1M) SARM
module. ttcompat: V7, 4BSD and	XENIX STREAMS compatibility	ttcompat(7) SARM
filesystems.	xfscck: check and repair XENIX	xfscck(1M) SARM
	xinstall: install commands.	xinstall(1M) XNX
shell script.	xinstall: XENIX installation	xinstall(1M) SARM
/rpc_reg, svc_reg, svc_unreg,	xprt_register, xprt_unregister:/	rpc_svc_calls(3N) NI
/svc_unreg, xprt_register,	xprt_unregister: library/	rpc_svc_calls(3N) NI
incremental/ xrestore,	xrestor: invoke XENIX	xrestore(1M) SARM
incremental filesystem/	xrestore, xrestor: invoke XENIX	xrestore(1M) SARM
xtt: extract and print	xt driver packet traces.	xtt(1M) SARM
xts: extract and print	xt driver statistics.	xts(1M) SARM
channels protocol used by	xt driver. xtproto: multiplexed	xtproto(5) PRM
channels protocol used by	xt driver. xtproto: multiplexed	xtproto(5) SARM
tty driver for AT&T windowing.	xt: STREAMS-based multiplexed	xt(7) SARM
tty driver for AT&T windowing.	xt: STREAMS-based multiplexed	xt(7) STRM
/gcd, rpow, msqrt, sdiv, itom,	xtom, mtom, mfree: multiple/	mp(3X) BSD
protocol used by xt driver.	xtproto: multiplexed channels	xtproto(5) PRM
protocol used by xt driver.	xtproto: multiplexed channels	xtproto(5) SARM
statistics.	xts: extract and print xt driver	xts(1M) SARM
packet traces.	xtt: extract and print xt driver	xtt(1M) SARM
bessel: j0, j1, jn,	y0, y1, yn: Bessel functions.	bessel(3M) PRM
bessel: j0, j1, jn, y0,	y1, yn: Bessel functions.	bessel(3M) PRM
compiler-compiler.	yacc: yet another	yacc(1) PRM
	yes: print string repeatedly.	yes(1) XNX
prompts for and validates	yes/no. ckyorn:	ckyorn(1) SARM
prompts for and validates	yes/no. ckyorn:	ckyorn(1) SS
yacc:	yet another compiler-compiler.	yacc(1) PRM
bessel: j0, j1, jn, y0, y1,	yn: Bessel functions.	bessel(3M) PRM
yperr_string, ypprot_err:	YP client interface. /yp_master,	ypcint(3N) NI
ypcat: print values in a	YP data base.	ypcat(1) NI
ypinit: build and install	YP database.	ypinit(1M) NI
ypmake: rebuild	YP database.	ypmake(1M) NI
structure. ypfiles: the	YP database and directory	ypfiles(4) NI
makedbm: make a	YP dbm file.	makedbm(1M) NI
yp_update: changes	yp information.	ypupdate(3N) NI
ypupdated: server for changing	yp information.	ypupdated(1M) NI
/return current version of a	YP map at a YP server host.	yppoll(1M) NI
ypxfr: transfer	YP map from a YP server to host.	ypxfr(1M) NI
value of one or more keys from a	YP map. ypmatch: print the	ypmatch(1) NI
force propagation of a changed	YP map. yppush:	yppush(1M) NI
ypserv, ypbind:	YP server and binder processes.	ypserv(1M) NI
current version of a YP map at a	YP server host. yppoll: return	yppoll(1M) NI
ypwhich: return name of	YP server or map master.	ypwhich(1) NI
ypxfr: transfer YP map from a	YP server to host.	ypxfr(1M) NI

<p> updaters: configuration file for /yp_match, yp_first, yp_next, ypset: point processes. ypserv, ypclnt, yp_get_default_domain, base. yp_bind, yp_unbind, yp_match,/ /yp_all, yp_order, yp_master, directory structure. /yp_bind, yp_unbind, yp_match, yp_unbind, yp_match, / ypclnt, database. /yp_next, yp_all, yp_order, or more keys from a YP map. yp_all, / /yp_bind, yp_unbind, /yp_unbind, yp_match, yp_first, yp_first, yp_next, yp_all, of a YP map at a YP server/ /yp_master, yperr_string, changed YP map. binder processes. particular server. /yp_get_default_domain, yp_bind, information. YP information. server or map master. server to host. display/ compress, uncompress, zero: source of zic: time zdump: time timezone: get time set default system time set default system time </p>	<p> YP updating. updaters(4) NI yp_all, yp_order, yp_master, / ypclnt(3N) NI ypbind at a particular server. ypset(1M) NI ypbind: YP server and binder ypserv(1M) NI yp_bind, yp_unbind, yp_match, / ypclnt(3N) NI ypcat: print values in a YP data ypcat(1) NI ypclnt, yp_get_default_domain, ypclnt(3N) NI yperr_string, ypprot_err: YP / ypclnt(3N) NI ypfiles: the YP database and ypfiles(4) NI yp_first, yp_next, yp_all, / ypclnt(3N) NI yp_get_default_domain, yp_bind, ypclnt(3N) NI ypinit: build and install YP ypinit(1M) NI ypmake: rebuild YP database. ypmake(1M) NI yp_master, yperr_string, / ypclnt(3N) NI ypmatch: print the value of one ypmatch(1) NI yp_match, yp_first, yp_next, ypclnt(3N) NI yp_next, yp_all, yp_order, / ypclnt(3N) NI yp_order, yp_master, / /yp_match, ypclnt(3N) NI yppoll: return current version yppoll(1M) NI ypprot_err: YP client interface. ypclnt(3N) NI yppush: force propagation of a yppush(1M) NI ypserv, ypbind: YP server and ypserv(1M) NI ypset: point ypbind at a ypset(1M) NI yp_unbind, yp_match, yp_first, / ypclnt(3N) NI yp_update: changes yp ypupdate(3N) NI y pupdated: server for changing y pupdated(1M) NI ypwhich: return name of YP ypwhich(1) NI ypxfr: transfer YP map from a YP ypxfr(1M) NI zcat: compress, expand or compress(1) URM zdump: time zone dumper. zdump(1M) SARM zero: source of zeroes. zero(7) SARM zeroes. zero(7) SARM zic: time zone compiler. zic(1M) SARM zone compiler. zic(1M) SARM zone dumper. zdump(1M) SARM zone name given offset from GMT. timezone(3C) BSD zone. timezone: timezone(4) PRM zone. timezone: timezone(4) SARM </p>
--	--

3. MASTER SUBJECT INDEX

3. MASTER SUBJECT INDEX

3 Master Subject Index

Using the Subject Index	3-1
Book Acronyms Used in the Master Index	3-2

Master Subject Index	3-3
-----------------------------	------------

Using the Subject Index

Entries in the index point to guide material and/or manual pages. Manual pages are concise, comprehensive reference information that was written with the expectation that readers are somewhat familiar with the material. Guide information is more explanatory and procedural.

The example below illustrates an entry for the term `find(1)`. The word `find` is printed in constant width typeface because it is a literal; that is, it is either typed in by the user or printed out by the system. The index includes both literal and non-literal entries.

EXAMPLE `find(1)` SAG 5: 40, 8: 4, 6; URM

Whenever you see a number in parentheses following an entry, such as (1) in the example above, it indicates that there is a manual page for that term. In these cases, at least one of the references is given without chapter and page numbers, pointing to the book where the manual page can be found. In the example, the URM contains the `find(1)` manual page. Manual pages are arranged alphabetically in numbered sections and therefore no chapter and page numbers apply to them. Simply go to the referenced book and look for the term alphabetically in the appropriate section.

There is one book of manual pages that is an exception to this rule, the *Device Driver Interface/Driver-Kernel Interface Reference Manual*. For that book, manual pages have been assigned page numbers and complete references are given, such as those for guide material as described below.

References to guide material appear in the index as follows:

index entry BOOK ACRONYM CHAPTER: PAGE-PAGE

In the above example, SAG 5: 40, 8: 4, 6 indicates that there is guide information on `find` in the *System Administrator's Guide*, chapter 5, page 40 and chapter 8, pages 4 and 6.

If you see a letter where a chapter number is usually given, it means that information is located in an appendix with that alphabetic name.

Book Acronyms Used in the Master Index

BSD	BSD Compatibility Guide
CGP	Programmer's Guide: ANSI C and Programming Support Tools
CHAR	Programmer's Guide: Character User Interface
CTG	ANSI C Transition Guide
DDRM	Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual
ISDG	Integrated Software Development Guide
MG	Migration Guide
MOUSE	Mouse Driver Administrator's Guide
NI	Programmer's Guide: Networking Interfaces
NUAG	Network User's and Administrator's Guide
PRM	Programmer's Reference Manual
PSX	Programmer's Guide: POSIX Conformance
SAG	System Administrator's Guide
SARM	System Administrator's Reference Manual
SCSI	Programmer's Guide: SCSI Driver Interface
SS	Programmer's Guide: System Services and Application Packag- ing Tools
STRM	Programmer's Guide: STREAMS
UG	User's Guide
URM	User's Reference Manual
XNX	XENIX Compatibility Guide

Master Subject Index

- \ (backslash) CHAR 2: 6
- . (see current directory)
- .. (see parent directory)
- + addition operator CGP 3: 37
- & address operator CGP 3: 34
- &= assignment operator CGP 3: 42
- *= assignment operator CGP 3: 42
- += assignment operator CGP 3: 42
- = assignment operator CGP 3: 42
- /= assignment operator CGP 3: 42
- <<= assignment operator CGP 3: 42
CGP 3: 42
- = assignment operator CGP 3: 42
- >>= assignment operator CGP 3: 42
- ^= assignment operator CGP 3: 42
- |= assignment operator CGP 3: 42
- & (background symbol) CHAR A: 4
- ` (backquote) CHAR 2: 6
- & bitwise AND operator CGP 3: 39
- ^ bitwise exclusive OR operator CGP
3: 40
- | bitwise inclusive OR operator CGP
3: 40
- , comma operator CGP 3: 42
- && (conditional execution) CHAR 2: 8
- || (conditional execution) CHAR 2: 8
- ? : conditional operator CGP 3: 41
- decrement operator CGP 3: 35
- / division operator CGP 3: 36
- ... ellipsis notation CGP 3: 28
- == equality operator CGP 3: 39
- >= greater or equal operator CGP
3: 39
- > greater than operator CGP 3: 39
- << (here document) CHAR 4: 21
- ++ increment operator CGP 3: 35
- * indirection operator CGP 3: 34
- != inequality operator CGP 3: 39
- << left shift operator CGP 3: 38
- <= less or equal operator CGP 3: 39
- < less than operator CGP 3: 39
- && logical AND operator CGP 3: 40
- ! logical negation operator CGP 3: 35
- || logical OR operator CGP 3: 41
- * multiplication operator CGP 3: 36
- negation operator CGP 3: 34
- ~ one's complement operator CGP
3: 35
- | (pipe) CHAR 2: 8
- # (pound sign) CHAR 2: 2
- # preprocessing operator CGP 3: 12
- ## preprocessing operator CGP 3: 13
- < (redirect input) CHAR 2: 8
- > (redirect output) CHAR 2: 8
- >> right shift operator CGP 3: 38
- ; (semicolon) CHAR 2: 8
- ! (shell escape) CHAR 1: 24-25; UG
5: 90
- ' (single quote) CHAR 2: 6
- . structure member operator CGP
3: 42
- > structure pointer operator CGP
3: 42
- subtraction operator CGP 3: 37-38
- + unary plus operator CGP 3: 35
- 9-track tape SAG 15: 16
defined in device database SAG
15: 32

A

- a64l MG A: 13
- a64l(3C) PRM
- abort MG A: 13

- abort(3C) PRM
- abortive release NI 2: 13, 36
- abs MG A: 13
- abs(3C) PRM
- absolute pathname (see pathname)
- accept(1M) SARM
- accept(3N) NI 3: 7, 34
- access MG A: 13
- access permission SAG 12: 21
- access(2) PRM
- account log (BNU) SAG 7: 69
- accounting
 - billing users SAG 2: 18-19
 - command summary SAG 2: 36
 - daily process SAG 2: 7-9
 - daily reports SAG 2: 20-27
 - files SAG 2: 32-35
 - fixing corrupted files SAG 2: 15-16
 - last login report SAG 2: 28-29
 - monthly reports SAG 2: 18-28
 - overview SAG 2: 1-2
 - programs SAG 2: 4
- acct MG A: 13
- acct(1M) SARM
- acct(2) PRM
- acct(4) SARM
- acctcms(1M) SARM
- acctcom MG A: 5
- acctcom(1) URM
- acctcon(1M) SARM
- acctmerg(1M) SARM
- accton MG A: 5
- acctprc(1M) SARM
- acctsh(1M) SARM
- acos MG A: 13
- action descriptor CHAR 2: 19, 22, 24, 28-29, 3: 9, 4: 14, 17
- action installation parameter SAG 14: 14
- active_border descriptor CHAR 2: 28, 4: 7
- active_title_bar descriptor CHAR 2: 28, 4: 7
- active_title_text descriptor CHAR 2: 28, 4: 7
- adb MG A: 12
- addbib(1) BSD; BSD/XNX 1: 16
- add-books ISDG A: 2; SS A: 2
- addch MG A: 14
- addch(3X) CHAR 7: 3
- address
 - binding NI 3: 48
 - wildcard NI 3: 48
- address space of a process SS 7: 1, 15
- address style translation (see mail(1))
- addscr ISDG A: 11; SS A: 11
- addseverity(3C) PRM
- addstr MG A: 14
- addstr(3X) CHAR 6: 4, 7: 5
- adjmsg(D3DK) DDRM
- adjtime(2) PRM
- adm login SAG 12: 19, 16: 19
- admin MG A: 12
- admin(1) CGP 10: 2-3, 12, 26-29, 39-40; PRM
- admin(4) SAG 14: 3; SARM
- administrative logins SAG 12: 18
- advisory locking SS 3: 2
- alarm MG A: 14
- alarm(2) PRM
- alias file (FMLI) CHAR 4: 20-21
 - defining pathname aliases CHAR 4: 20
 - defining search paths CHAR 4: 20
 - overview CHAR 1: 10
- alias lists, mail(1) SAG F: 5

- aliases NUAG 23: 1
- aliases(4) BSD
- alloca(3) BSD; BSD/XNX 1: 27
- Allocating SCSI Block structures
SCSI 4: 2
- allocb(D3DK) DDRM
- alphasort(3) BSD/XNX 1: 28
- alternate character set CHAR 2: 47-49
- altslks descriptor CHAR 2: 18, 20, 23,
3: 4, 34, 57
- anon, option to NFS share com-
mand NUAG 14: 6
- ANSI C CGP 3: 1 (see also C
language)
- ANSI Specification document SCSI
1: 10
- a.out(4) CGP 2: 4-6; PRM
- application gateway (see gateway)
- application layer, TCP/IP NUAG
1: 8-9
- application level files (FMLI)
lists of descriptors CHAR 2: 25-30
overview CHAR 1: 9-10
- application programming SS 2: 1-29
- applications software installation (see
package)
- apropos(1) BSD
- ar MG A: 12
- ar(1) CGP 2: 14; PRM; URM
- ar(4) PRM; SARM
- arch(1) BSD; BSD/XNX 1: 22
- ARCHITECTURE parameter SAG 8: 71
- archive commands SS 6: 11-12
- archive libraries CGP 2: 2-3, 12-32
creating CGP 2: 14
implementation CGP 2: 2-3, 21
linking with CGP 2: 12-19, 31-32
maintaining CGP 9: 14-16
- archives(4) SARM
- argc and argv CGP 1: 5-6, 2: 52-55
- ARG_MAX parameter SAG 8: 68
- ARGn variable CHAR 2: 11
- arithmetic, awk(1) UG 10: 21
- arithmetic conversions CGP 3: 2, 32-33
- arithmetic types CGP 3: 21
- ARP (Address Resolution Protocol)
NUAG 1: 3, 6
- arp(1M) NUAG; SARM
- ARP(7) NUAG; SARM
- ARPANET NUAG 1: 3
- array
declaration CGP 3: 28
initialization CGP 3: 45-46
- as(1) CGP 2: 6; PRM
- ASCII (American Standard Code for
Information Interchange) UG GL: 1
- ascii(5) PRM; SARM
- asctime MG A: 14
- asin MG A: 14
- asktime MG A: 5
- asm CGP 3: 6, A: 1-9
macros CGP A: 4-7
usage example CGP A: 2-3
usage guidelines CGP A: 8-9
- assembler CGP 2: 2, 6
- assembly language escapes (see asm)
- #assert CGP 3: 17-18
- assert MG A: 14
- assert(3X) PRM
- assert.h header file BSD/XNX 1: 35
- assign MG A: 27
- asx MG A: 12
- asy(7) SARM
- asynchronous input/output, in pol-
ling STRM 6: 6
- asynchronous mode NI 2: 28, 51, 63,
4: 14-15

- asynchronous protocol Stream, example STRM 4: 4–11
- at MG A: 5
- at(1) UG 9: 28–30; URM
- atan MG A: 14
- atan2 MG A: 14
- atexit(3C) PRM
- atof MG A: 14
- atoi MG A: 14
- atol MG A: 14
- atq(1) URM
- atrm(1) URM
- attroff(3X) CHAR 7: 24
- attron(3X) CHAR 7: 24
- attrset(3X) CHAR 7: 24
- authentication NI 3: 50, 7: 46, 9: 3, 6, 14
 - AUTH_DES NI 7: 51–53, 9: 15–18, 20
 - AUTH_NONE NI 7: 47–48, 9: 14
 - AUTH_SHORT NI 9: 15
 - AUTH_SYS NI 7: 49, 9: 14
 - clock synchronization NI 9: 17
 - DES NUAG 18: 3
 - nicknames NI 9: 17
 - RPC NUAG 18: 2
 - UNIX NUAG 18: 2
- auto CGP 3: 26
- autoadvance descriptor CHAR 2: 21, 3: 38
- autobaud SAG 13: 32–33, 47
- autoconfiguration, changed for Release 4.0 MG 5: 11
- automount(1M) NUAG 19: 1; SARM
- automounter
 - debugging NUAG 19: 18–20
 - hierarchical mounts NUAG 19: 8
 - invoking NUAG 19: 14
 - multiple mounts NUAG 19: 7–8
 - special mount point NUAG 19: 2
- automounter maps NUAG 19: 3
 - modifying NUAG 19: 17
 - using environment variables NUAG 19: 13
 - using substitution NUAG 19: 11–12
 - writing a direct map NUAG 19: 5–13
 - writing a master map NUAG 19: 4–5
 - writing an indirect map NUAG 19: 6–7
- autopush(1M) SAG 13: 35; SARM; STRM, E: 6
- awk MG A: 5
 - awk(1) SS 1: 5–6; URM
 - arithmetic UG 10: 21
 - arrays UG 10: 33
 - built-in arithmetic functions UG 10: 21, 60
 - built-in string functions UG 10: 24, 59
 - built-in variables UG 10: 20, 61
 - command line arguments UG 10: 47
 - control flow UG 10: 31, 58
 - cooperation with the shell UG 10: 49
 - error messages UG 10: 11
 - field variable UG 10: 28
 - fields UG 10: 4
 - input UG 10: 43–48, 59
 - input from files and pipes UG 10: 43
 - multi-line record UG 10: 44
 - operators UG 10: 60
 - output UG 10: 38–42, 59
 - output to files and pipes UG 10: 40–42
 - patterns UG 10: 7, 12, 18–19, 58
 - regular expressions UG 10: 14–17, 61
 - relational expressions UG 10: 13
 - sample applications UG 10: 52–57
 - strings and string functions UG 10: 24

summary UG 10: 58–64
 type coercion UG 10: 29
 user-defined functions UG 10: 36

B

back-enable of a queue STRM 5: 29
 background execution UG 9: 10, GL: 1
 background job, in job control STRM
 6: 9
 background process NUAG 24: 3, 6
 backq(D3DK) DDRM
 backquoted expression (FMLI) CHAR
 2: 7–8, 4: 4, A: 8
 statement operators CHAR 2: 8
 backslash (\) CGP 3: 8; CHAR 2: 6; UG
 2: 8, 9: 11
 backup MG A: 5
 backup commands
 quick reference SAG 3: 58–62
 task and command summary SAG
 3: 1–5
 backup exception list SAG 3: 20–27
 convert from earlier backups SAG
 3: 24–27
 create a customized list SAG 3: 21
 customize SAG 3: 20
 ignore during backup SAG 3: 20
 modify a list SAG 3: 21
 backup frame CHAR 1: 22
 backup history log SAG 3: 53–57
 backup jobs
 controlling jobs SAG 3: 51
 core file system SAG 3: 29–30
 demand SAG 3: 33, 45–46
 determine media and time require-
 ments SAG 3: 12
 label checking override SAG 3: 48
 limited SAG 3: 45–46
 log SAG 3: 53–57
 monitor and control SAG 3: 47–52
 operator assistance SAG 3: 41–44,
 47–49
 overview SAG 3: 6–7
 planning SAG 3: 11–14
 preview schedules SAG 3: 44
 rotation period SAG 3: 32
 specify interactive, automatic, or
 background SAG 3: 41–44
 status SAG 3: 49–51
 suspend, resume, and cancel SAG
 3: 51
 backup menu, sysadm SAG 3: 1–5
 backup methods
 common options to SAG 3: 18–19
 full data partition SAG 3: 27
 full disk SAG 3: 27
 full file SAG 3: 19
 full image SAG 3: 27
 incremental SAG 3: 19–27
 migration SAG 3: 28–29
 operations changed for Release 4.0
 MG 5: 30
 selection of SAG 3: 7–10
 backup tables SAG 3: 15–40
 add or change entries SAG 3: 36–38
 bkexcept.tab SAG 3: 20–27
 bkreg.tab SAG 3: 15–40
 create tables of contents SAG
 3: 34–36
 customize SAG 3: 16–17
 defaults SAG 3: 17
 define originating objects SAG 3: 30
 dependencies and priorities SAG
 3: 33

- display SAG 3: 15
- history (bkhis.tab) SAG 3: 53–57
- specify destination devices SAG 3: 31
- specify methods SAG 3: 17–30
- specify rotation period SAG 3: 32
- status (bkstatus.tab) SAG 3: 49–51
- validate tables SAG 3: 38–40
- backup(1) XNX
- backup(1M) SARM
- bad blocks SAG 4: 30–42, 15: 15
 - and dd command SAG 15: 17
 - detecting SAG 4: 32, 34
 - fixing SAG 4: 41
 - handling SAG 4: 33
 - recovery SAG 4: 33
 - repairing SAG 4: 30
 - special cases SAG 4: 40
- bancol descriptor CHAR 2: 26, 4: 5
- banner MG A: 5
- banner descriptor CHAR 2: 26, 4: 5
- banner line (FMLI) CHAR 1: 3
 - descriptor definitions CHAR 4: 5
 - list of descriptors CHAR 2: 26
- banner(1) UG 9: 13, 21; URM
- banner_text descriptor CHAR 2: 28, 4: 7, 9
- base address CGP 13: 42
- base system device drivers ISDG 3: 56
- basedir installation parameter SAG 14: 13
- basename MG A: 6
- basename(1) BSD; BSD/XNX 1: 17; URM
- basename(3G) PRM
- Basic Networking Utilities package (see BNU)
- basic types CGP 3: 21
- batch MG A: 6
- batch RPC NI 7: 42, 9: 9
- batch(1) UG 9: 26–28
- baud rate SAG 13: 32–33; UG 2: 3, 11, GL: 2
- bc MG A: 6
- bc(1) SS 1: 7; URM
- bcanput(D3DK) DDRM
- bcheckrc SAG 6: 16
- bcmp(3) BSD/XNX 1: 28
- bcopy(3) BSD/XNX 1: 28
- bcopy(D3DK) DDRM
- bdevsw (block device switch table) SCSI 1: 6
- bdiff MG A: 6
- bdiff(1) URM
- beep(3X) CHAR 7: 37
- begcol descriptor CHAR 2: 18, 20, 23, 3: 4, 34, 57
- begrow descriptor CHAR 2: 18, 20, 23, 3: 4, 34, 57, A: 12
- Berkeley Software Distribution (see BSD)
- bessel(3M) PRM
- bfs MG A: 6
- bfs (boot file system) SAG 5: 19–22, 6: 9
- bfs(1) URM
- B_GETDEV SCSI 2: 6, 10
- B_GETDEV code example SCSI 1: 7
- bgets(3G) PRM
- B_GETTYPE SCSI 2: 4
- B_GETTYPE code example SCSI 1: 7
- bidirectional transfer, example STRM 7: 24–29
- biff(1) BSD
- bin directory UG 9: 40
- bin login SAG 12: 19, 16: 19
- binarsys(4) SARM

bind(3N) NI 3: 52
 binding, YP NI 11: 5-6
 binding local names NI 3: 4
 biod(1M) NUAG 17: 2, 4, 10; SARM
 biodone(D3DK) DDRM
 biowait(D3DK) DDRM
 bit-fields CGP 3: 22, 5: 4; CTG46
 promotion of CTG12
 bkexcept(1M) SAG 3: 20-27; SARM
 bkexcept.tab SAG 3: 20-27
 bkhistory(1M) SARM
 bkhist.tab SAG 3: 53-57
 bkoper(1M) SAG 3: 47-49; SARM
 bkreg(1M) SAG 3: 15-40; SARM
 bkreg.tab SAG 3: 15-40
 bkstatus(1M) SAG 3: 49-51; SARM
 bkstatus.tab SAG 3: 49-51
 blinking attribute CHAR 2: 47
 block device SAG 15: 7-8
 add a SAG 15: 14
 partition SAG 15: 10
 path of node for SAG 15: 33
 block files SAG 15: 7
 block I/O DDRM 3: 4-7
 BNU administrative support files
 SAG 7: 62-65
 checkpoint (P.) SAG 7: 64
 data (D.) SAG 7: 63
 execute (X.) SAG 7: 64
 lock (LCK.) SAG 7: 62
 lock (LK.) SAG 7: 62
 temporary data (TM.) SAG 7: 62
 work (C.) SAG 7: 63
 BNU (Basic Networking Utilities)
 SAG 7: 15-75
 daemons SAG 7: 19-20
 debugging SAG 7: 29-31
 error messages SAG E: 47-51
 list basic information SAG 7: 29
 maintenance (automatic) SAG
 7: 25-27
 maintenance (manual) SAG 7: 27
 poll remote machines SAG 7: 56
 queue jobs for remote machines
 SAG 7: 59-61
 security SAG 7: 48-56, 61, 70
 setup SAG 7: 22-25
 unknown calling machines SAG
 7: 61
 with STARLAN or STREAMS-based
 providers SAG 7: 23
 BNU database support files SAG
 7: 31-61
 Config file SAG 7: 33
 Devconfig file SAG 7: 23, 56
 Devices file SAG 7: 33-39
 Dialcodes file SAG 7: 47
 Dialers file SAG 7: 39-41
 Grades file SAG 7: 59-61
 Limits file SAG 7: 24, 58
 Permissions file SAG 7: 22, 48-56
 Poll file SAG 7: 56
 remote.unknown file SAG 7: 61
 Sysfiles file SAG 7: 23, 57
 Systems file SAG 7: 42-47
 BNU logs SAG 7: 65-75
 account SAG 7: 69
 command SAG 7: 65
 errors SAG 7: 66
 Foreign SAG 7: 74
 perflog (performance) SAG
 7: 72-74
 security SAG 7: 70
 system history SAG 7: 66
 xferstats (transfer) SAG 7: 67
 bold attribute CHAR 2: 47

- boot SAG 6: 17
 - disk partitions SAG 6: 5–12, 15: 10,
A: 3
 - error messages SAG E: 36–44
 - file system changed for Release 4.0
MG 5: 11
 - default parameters SAG 6: 27
 - procedure SAG 6: 5
- boot program SAG 6: 7
- boot(1M) SARM
- boot(4) SARM
- /boot, directory SAG 6: 12
- bootable disk SAG 6: 44–48
- bootable operating system,
configuring SAG 8: 52–62
- booting and setuid problems, NFS
NUAG 18: 9
- bootparamd(1M) NI; NUAG
- Bourne shell (see shell and sh(1))
- box MG A: 14
- bp_mapin(D3DK) DDRM
- bp_mapout(D3DK) DDRM
- brc(1M) SAG 6: 16; SARM
- break statement CGP 3: 51
- breelse(D3DK) DDRM
- bridge NUAG 4: 2
- brk MG A: 14
- brk(2) PRM; SS 7: 16
- brkctl MG A: 14
- broadcast RPC NI 6: 25, 7: 40, 9: 9
 - synopsis NI 7: 41
- broadcasting NI 3: 51
- browser (see cscope(1))
- b_scb SCSI 4: 2
- BSD Compatibility Package BSD/XNX
1: 1
 - accessing BSD/XNX 1: 2
 - commands BSD/XNX 1: 7, 14–25
 - contents BSD/XNX 1: 7–13
 - file operations unified for Release
4.0 MG 5: 8
 - files SAG B: 27
 - header files BSD/XNX 1: 34–36
 - installing BSD/XNX 1: 2
 - languages BSD/XNX 1: 19
 - libraries BSD/XNX 1: 8–11, 26–31
 - macro packages BSD/XNX 1: 15
 - mail BSD/XNX 1: 21
 - NFS commands BSD/XNX 1: 22
 - signals BSD/XNX 1: 33
 - system calls BSD/XNX 1: 32–33
- bsearch MG A: 14
- bsearch(3C) PRM
- b_sfb SCSI 4: 2
- bstring(3) BSD
- btop(D3DK) DDRM
- btopr(D3DK) DDRM
- bufcall(D3DK) DDRM
- buf(D4DK) DDRM
- buffer pool ISDG 3: 3, 33
- buffer space ISDG 3: 32
- buffers, for raw I/O DDRM 3: 7
- BUFHWM parameter SAG 8: 72
- bufsplit(3G) PRM
- built-in utilities (FMLI) CHAR 2: 39–41,
C: 1
- built-in variables (FMLI) CHAR
2: 11–12
- bus error CGP 6: 1
- Bus reset SCSI 1: 9
- bus_type structure SCSI 1: 8
- button descriptor CHAR 2: 19, 22, 24,
28, 4: 14
- byte swapping NI 3: 27
- byteorder(3N) NI
- bzero(3) BSD/XNX 1: 28

bzero(D3DK) DDRM

C

C language CGP 1: 4-5, 3: 1-52; SS 1: 4-8
 changed for Release 4.0 MG 5: 44
 comments CGP 3: 10
 compilation modes and dependencies CGP 3: 1-2, 6-7, 20-21, 31, 38
 constants CGP 3: 6-9
 conversions CGP 3: 2, 31-33
 declarations CGP 3: 21-29
 definitions CGP 3: 29-30
 escape sequences CGP 3: 8
 expressions CGP 3: 33-46
 identifiers CGP 3: 5
 initialization CGP 3: 44-46
 keywords CGP 3: 6
 operators CGP 3: 12-13, 34-43
 phases of translation CGP 3: 3-4
 portability CGP 3: 52
 preprocessing CGP 3: 11-20
 scope CGP 3: 24-25
 statements CGP 3: 47-51
 storage duration CGP 3: 25-26
 string literals CGP 3: 9
 tokens CGP 3: 5-12
 types CGP 3: 21-24, 27-30

C library
 linking with CGP 2: 11-14
 partial contents CGP 2: 38-44
 standard I/O CGP 2: 50-55

C shell (see shell and csh(1))
 cabs MG A: 29
 cal MG A: 6
 cal(1) URM

calculator programs SS 1: 7
 calendar MG A: 6
 calendar(1) URM
 call terminal (see ct)
 call UNIX computer (see cu)
 calloc MG A: 14
 callout table ISDG 3: 6
 cancel MG A: 6
 cancel command
 FACE UG C: 14
 FMLI CHAR 2: 31
 can_change_colors(3X) CHAR 7: 30
 canput(D3DK) DDRM
 capinfo MG A: 6
 captainfo(1M) CHAR 13: 19; SARM
 cartridge tape
 copy files to SAG 15: 19
 defined in device database SAG 15: 32
 device names SAG A: 2
 display information about SAG 15: 17
 duplicate a SCSI SAG 15: 20-22
 erase a SAG 15: 23
 format a SAG 15: 16-17
 partition SAG 15: 11-12, 15
 partitions SAG A: 9
 reformat a SAG 15: 23
 SCSI (see SCSI cartridge tape)
 cartridge tape drive, remove SAG 15: 25-30
 case sensitivity (FMLI) CHAR 2: 3
 casts, FMLI CHAR 2: 3-5
 cat MG A: 6
 cat(1) UG 3: 28-32, 9: 16; URM
 catgets(3C) PRM
 catman(1M) BSD; BSD/XNX 1: 14
 catopen(3C) PRM

Caution about the pass-through interface SCSI 2: 9
cb MG A: 12
cb(1) PRM
cbreak(3X) CHAR 7: 42
cc MG A: 12
cc(1) BSD; BSD/XNX 1: 19; CGP 2: 1-32; PRM
 compilation modes and dependencies CGP 3: 1-2, 6-7, 20-21, 31, 38
 creating shared objects CGP 2: 15-16, 23, 28-29
 debugging option CGP 2: 9, 6: 3
 header search option CGP 2: 8-9
 library linking option CGP 2: 12-20, 31-32
 library search option CGP 2: 17-20, 32
 optimizing option CGP 2: 10
 profiling options CGP 2: 9-10, 7: 2, 4
 program naming option CGP 2: 4
 static linking options CGP 2: 13-14, 17-19, 31-32
cd MG A: 6
cd(1) UG 3: 24-25; URM
CDB SCSI 4: 15
CDB (Command Descriptor Block) SCSI 4: 4
CDB use SCSI 2: 10
cdc MG A: 12
cdc(1) CGP 10: 12, 32-33; PRM
cdevsw (character device switch table) SCSI 1: 6
CDPATH environment variable SAG 17: 27
ceil MG A: 14
cflow MG A: 12
cflow(1) CGP 1: 12; PRM

char CGP 3: 21, 31
character classification routines CGP 2: 42
character constants CGP 3: 7-9
character conversion routines CGP 2: 43
character device SAG 15: 7-8
 add a SAG 15: 15
 path of node for SAG 15: 33
character sequences for terminal attributes, table of CHAR 2: 47
character sets SAG 9: 28-31
chdir MG A: 14
chdir(2) PRM
checkeq(1) BSD/XNX 1: 15
checkfsys(1M) SARM
 login SAG 12: 18
checknr(1) BSD; BSD/XNX 1: 15
checkworld command (FMLI) CHAR 2: 32, A: 9
chgrp MG A: 6
chgrp(1) URM
child directory UG 3: 2
chkey(1) NI; NUAG
chmod MG A: 6, 14
chmod(1) SS 3: 17; UG 3: 28, 49-56, 9: 39; URM
chmod(2) PRM
choicemsg descriptor CHAR 2: 21, 3: 38, A: 5
choices command (FMLI) CHAR 1: 9, 2: 32
Choices menu CHAR 1: 9, 3: 41
chown MG A: 6, 14
chown(1) BSD; BSD/XNX 1: 17; UG 3: 56-57; URM
chown(2) PRM
chpoll(D2DK) DDRM

chroot MG A: 6, 14
 chroot(1M) SARM
 chroot(2) PRM
 chrtbl(1M) SARM
 chsize MG A: 14
 chsize(2) XNX
 circuit_n transport type NI 5: 9
 circuit_v transport type NI 5: 9
 ckbinarsys(1M) SARM
 ckbupscd(1M) SARM
 ckdate(1) SARM; SS
 ckgid(1) SARM; SS
 ckint(1) SARM; SS
 ckitem(1) SARM; SS
 ckkeywd(1) SARM; SS
 ckpath(1) SARM; SS
 ckrange(1) SARM; SS
 ckstr(1) SARM; SS
 cktime(1) SARM; SS
 ckuid(1) SARM; SS
 ckyorn(1) SARM; SS
 class, scheduler (see scheduler class)
 class action script ISDG 2: 18, 23-30;
 SS 8: 19, 25-31
 classes
 assigning objects to ISDG 2: 35-36,
 44; SS 8: 42, 50
 installation of ISDG 2: 25; SS 8: 26
 removal of ISDG 2: 26; SS 8: 28
 system ISDG 2: 27; SS 8: 29
 the awk class ISDG 2: 28; SS 8: 30
 the build class ISDG 2: 29; SS 8: 31
 the sed class ISDG 2: 27; SS 8: 29
 cleanup command
 FACE UG C: 14
 FMLI CHAR 2: 31
 clear MG A: 6, 14
 clear(1) URM
 clear(3X) CHAR 7: 10
 clearerr MG A: 14
 clearok MG A: 14
 clients NI 2: 11, 16, 24-26, 38-39, 4: 2,
 5: 2; NUAG 7: 3
 handle used by rpcgen(1) NI 6: 10
 RPC NI 5: 12
 client/server model NI 3: 29
 clnt_create function NI 7: 2
 clnt_dg_create function NI 7: 4
 clnt_stat type (in RPC program-
 ming) NI 7: 10
 clnt_tli_create function NI 7: 3
 clnt_tp_create function NI 7: 3
 clnt_vc_create function NI 7: 4
 clocal MG A: 14
 clock MG A: 14
 clock(3C) PRM
 clone(7) STRM
 cloning (STREAMS) STRM 9: 18
 close MG A: 14
 close command (FMLI) CHAR 2: 31
 close descriptor CHAR 2: 18, 20, 23,
 3: 5, 35, 58, A: 18
 close(2) NI 2: 48-50; PRM
 close(D2DK) DDRM
 closedir MG A: 14
 clrbuf(D3DK) DDRM
 clri MG A: 6
 clrtobot MG A: 14
 clrtobot(3X) CHAR 7: 11
 clrtoeol MG A: 14
 clrtoeol(3X) CHAR 7: 11
 CMA ISDG 3: 37
 cmchk MG A: 27
 cmd-menu command (FMLI) CHAR
 2: 32
 cmn_err(D3DK) DDRM

- cmp MG A: 6
- cmp(1) URM
- cmpress(1M) SAG 8: 6
- cocheck(1F) CHAR 2: 41, C: 1
- cocreate(1F) CHAR 2: 41, C: 1
- codestroy(1F) CHAR 2: 41, C: 1
- cof2elf(1) CGP 1: 12; PRM
- col descriptor CHAR 2: 21
- col(1) URM
- colltbl(1M) SARM
- color_content(3X) CHAR 7: 31
- COLOR_PAIR CHAR 7: 28
- colors
 - A_COLOR (ETI) CHAR 7: 30
 - attribute (ETI) CHAR 7: 28
 - changing definitions (ETI) CHAR 7: 29
 - COLOR_PAIR (ETI) CHAR 7: 28
 - descriptor definitions (FMLI) CHAR 4: 6-8
 - (ETI) CHAR 7: 25-36
 - examples of (FMLI) CHAR 4: 8-9
 - list of descriptors (FMLI) CHAR 2: 27
 - other macros and routines (ETI) CHAR 7: 30
 - PAIR_NUMBER (ETI) CHAR 7: 30
 - portability (ETI) CHAR 7: 29
 - redefining defaults (ETI) CHAR 7: 29
 - table (ETI) CHAR 7: 26
 - table of defaults (ETI) CHAR 7: 26
- columns descriptor CHAR 2: 18, 21, 23, 26, 3: 5, 42, 61, 4: 4
- comb MG A: 12
- comb(1) CGP 10: 12, 34-35; PRM
- comm MG A: 6
- comm(1) URM
- command execution (FMLI) CHAR 1: 22-25
- command interpreter (see shell)
- command line
 - CTRL-fc CHAR 1: 4, 14, 21, 23
 - CTRL-j CHAR 1: 4, 14, 21, 23
 - CTRL-z CHAR 1: 24
 - execute commands from CHAR 2: 36
 - FACE UG 5: 5
 - FMLI CHAR 1: 4, 23-24, A: 12
- command log (BNU) SAG 7: 65
- Command Menu
 - FACE UG C: 3
 - FMLI CHAR 1: 10, 22-23, 2: 32, A: 15
 - FMLI, modifying CHAR 4: 17
 - FMLI table of defaults CHAR 2: 36
- command prompt UG 2: 6
- command substitution UG 9: 26
- commands UG 1: 1, 5-9
 - background execution UG 9: 10
 - changed for Release 4.0 MG 5: 3
 - executing UG 9: 26-36
 - flow of control UG 1: 8
 - how to execute UG 1: 6-9
 - run at a later time UG 9: 26-29
 - sequential execution UG 9: 11
 - summary UG B: 1-5
 - syntax UG 1: 6-9
- commands file (FMLI) CHAR 4: 17-19, 21
 - descriptor definitions CHAR 4: 17
 - examples of CHAR 4: 18-19
 - list of descriptors CHAR 2: 29-30
 - overview CHAR 1: 10
- commands (FMLI) CHAR 2: 31-38
 - syntax and use CHAR 2: 31-36
 - user access to CHAR 2: 36-38
- comments CGP 2: 5, 3: 10
 - FMLI CHAR 2: 2
- communication

- C run-time routines NI 3: 27
- interprocess (IPC) SS 4: 1-87
- remote machine (see remote machine communication)
- communication tutorial UG 12: 1-26
- compiler CGP 2: 2-10 (see also C language; cc(1))
- compiler construction (see yacc(1))
- compiler diagnostics CGP 4: 1-133
 - error defined CGP 4: 2
 - fatal error defined CGP 4: 2
 - list of CGP 4: 3-129
 - operator names in CGP 4: 2, 130-131
 - warning defined CGP 4: 2
- compress(1) URM
- compver(4) ISDG 2: 13; SS 8: 15
- comsat(1M) NUAG; SARM
- concatenate file UG 3: 29-32
- conditional compilation CGP 3: 15-16
- conditional statements (FMLI) CHAR 2: 42-43
 - && CHAR 2: 8
 - || CHAR 2: 8
- config MG A: 6
- _config NI 13: 21, 24, 26
- Config file (BNU) SAG 7: 33
- _config file (SAF) SAG 13: 8, 21, 23
- configuration
 - display SAG 6: 39
 - error messages SAG E: 36-44
 - SAF (see SAF, configuration scripts)
 - system SAG 8: 52-62
- configure MG A: 6
- configuring unix SAG 8: 52-62
- conflgs(1M) SARM
- conflict installation parameter SAG 14: 13
- connect requests NI 2: 11-13, 17, 20-31, 34-35, 51-65, 69
 - multiple NI 2: 81
- connect(3N) NI
- connection errors NI 3: 7
- connection establishment NI 2: 10-12, 23-31
 - using sockets NI 3: 6
- connection release NI 2: 13, 36-39
 - table of routines NI 2: 13
- connectionless sockets NI 3: 13
- connectionless-mode NI 2: 7, 14, 40-47, 4: 9-13
 - example of transaction server NI 2: 77-79
 - servers NI 3: 36
 - state table NI 2: 67
- connection-mode NI 2: 7-13, 16-39, 47, 4: 2-9
 - client side NI 4: 4-5
 - example of client NI 2: 71-72
 - example of server NI 2: 73-76, 81-87
 - server side NI 4: 6-9
 - state table NI 2: 68
- connection-oriented transports NI 7: 61
- connld(7) STRM 11: 12
- console STRM G: 11
 - setup SAG 16: 6
- console(7) STRM
- const CGP 2: 25, 3: 22, 27; CTG20-22, 45
- const cast CHAR 2: 4
- constant expressions CGP 3: 43
- constants CGP 3: 6-9
 - promotion of integral CTG13
- continue statement CGP 3: 51
- control character UG 2: 9, GL: 3

control sequences (FACE) UG 5: 9
controller
 interface requirements ISDG 3: 36
 memory addresses ISDG 3: 37
controlling terminal STRM 6: 13
contty STRM G: 10
conv(3C) PRM
conversions CGP 3: 2, 31-33
convert(1) PRM
converting drivers (XENIX to UNIX)
 ISDG 3: 72-76
coproc(1F) CHAR, C: 1
co-processing CHAR 3: 71, A: 16
copy MG A: 6
 directories using rcp NUAG 22: 4
 symbolic links SS 6: 8-9
copy command (FACE) UG 5: 46, 55,
 C: 4
copy files UG 3: 40-42
 disk to disk SAG 15: 19
 disk to floppy SAG 15: 19
 disk to tape SAG 15: 19
 floppy to floppy SAG 15: 20
 using ftp NUAG 25: 1, 3
 using rcp NUAG 22: 2
 using tftp NUAG 25: 6
copy(1) XNX
copyb(D3DK) DDRM
copyin(D3DK) DDRM
Copying SCSI Block structures SCSI
 4: 2
copylist(3G) PRM
copymsg(D3DK) DDRM
copyout(D3DK) DDRM
copyreq structure STRM A: 10
copyresp structure STRM A: 11
copyright messages, write ISDG 2: 39;
 SS 8: 44

copyright(4) ISDG 2: 16; SS 8: 16
copysign function BSD/XNX 1: 29
core file system, backup SAG 3: 29-30
core(4) CGP 6: 1-6; PRM; SARM
coreceive(1F) CHAR 2: 41, C: 1
CORLIM parameter SAG 8: 75
cos MG A: 14
cosend(1F) CHAR 2: 41, C: 1
cosh MG A: 14
cp MG A: 6
cp(1) UG 3: 28, 40-42; URM
cpio MG A: 6
cpio(1) SAG 8: 4; URM
 restore service SAG 11: 5
cpp MG A: 12
cram(7) SARM
crash(1M) SAG 5: 29; SARM
 STREAMS debugging STRM D: 2-6
CRC (Cyclic Redundancy Check)
 SAG 4: 34
creat MG A: 14
creat(2) PRM
create command (FACE) UG 5: 42,
 44, C: 5
creatsem MG A: 15
creatsem(2) XNX
cred(D4DK) DDRM
credentials, in RPC authentication
 NUAG 18: 2
cref MG A: 12
crmode MG A: 15
cron MG A: 6
cron(1M) SAG 8: 8; SARM
cron(4) SARM
crontab MG A: 6
crontab(1) URM
crypt(1) SAG 12: 3; URM
crypt(3C) PRM

crypt(3X) PRM
cscope(1) CGP 1: 9, 8: 1-27; PRM
 command line CGP 8: 3, 15-17
 environment setup CGP 8: 2, 26-27
 environment variable CGP 8: 17-18
 usage examples CGP 8: 2-14, 18-25
csh MG A: 6
csh(1) URM
 changed for Release 4.0 MG 5: 6
 .cshrc NUAG 23: 1
csplit MG A: 6
csplit(1) URM
ct(1C) UG 12: 16-19; URM
ctags MG A: 12
ctags(1) URM
ctermid MG A: 15
ctermid(3S) PRM
ctime MG A: 15
ctime(3C) PRM
ctrace(1) CGP 1: 12; PRM
ctype(3C) PRM
cu MG A: 6
cu(1C) UG 12: 19-24; URM
current directory UG 3: 6-7, 10, GL: 3
current frame, definition CHAR 1: 5
current_field(3X) CHAR 11: 81
current_item(3X) CHAR 10: 60
curs_addch(3X) CHAR
curs_addchstr(3X) CHAR
curs_addstr(3X) CHAR
curs_attr(3X) CHAR
curs_beep(3X) CHAR
curs_bkgd(3X) CHAR
curs_border(3X) CHAR
curs_clear(3X) CHAR
curs_color(3X) CHAR
curs_delch(3X) CHAR
curs_deleteln(3X) CHAR
curses library CHAR 13: 1
curses(3X) CHAR 2: 47, 4: 6; SS 1: 8
curses.h CHAR 13: 4
curs_getch(3X) CHAR
curs_getstr(3X) CHAR
curs_getyx(3X) CHAR
curs_inch(3X) CHAR
curs_inchstr(3X) CHAR
curs_initscr(3X) CHAR
curs_inopts(3X) CHAR
curs_insch(3X) CHAR
curs_insstr(3X) CHAR
curs_instr(3X) CHAR
curs_kernel(3X) CHAR
curs_move(3X) CHAR
cursor CHAR 1: 16; UG GL: 3
curs_outopts(3X) CHAR
curs_overlay(3X) CHAR
curs_pad(3X) CHAR
curs_printw(3X) CHAR
curs_refresh(3X) CHAR
curs_scanw(3X) CHAR
curs_scr_dump(3X) CHAR
curs_scroll(3X) CHAR
curs_slk(3X) CHAR
curs_termattrs(3X) CHAR
curs_termcap(3X) CHAR
curs_terminfo(3X) CHAR
curs_touch(3X) CHAR
curs_util(3X) CHAR
curs_window(3X) CHAR
cuserid MG A: 15
cuserid(3S) PRM
custom MG A: 6
custom(1) XNX
custom(1M) SARM
cut(1) UG 9: 21-23; URM
cxref MG A: 12

cxref(1) CGP 1: 12; PRM
Cyclic Redundancy Check (CRC)
 SAG 4: 34
cylinder (disk) SAG 4: 38

D

daemon login SAG 12: 19, 16: 19
daemon mode, in STREAMS-based
 console STRM G: 14
daemons (network) SAG 7: 19-20
 biod NUAG 17: 2, 4, 10
 fingerd NUAG 1: 9, 3: 7
 ftpd NUAG 1: 8
 mountd NUAG 17: 2-4, 7
 named NUAG 1: 9
 nfsd NUAG 17: 2, 9
 remote NUAG 23: 1
 rlogind NUAG 1: 9
 routed NUAG 6: 10
 rwho NUAG 7: 10
 rwhod NUAG 3: 7
 telnetd NUAG 1: 8
DARPA protocols (see TCP/IP)
data block (STREAMS)
 linkage STRM 5: 6
 structure STRM 5: 5
data encryption SAG 12: 3
data flow, in xt driver STRM G: 26
data link layer, TCP/IP NUAG 1: 6
data loss SAG 4: 42
data movement between kernel and
 device ISDG 3: 9
data representation CGP 13: 3
 external (see XDR)
data segment CGP 2: 2, 22-27 (see
 also object files)
data transfer NI 2: 12, 31-36, 43-45, 3: 9
 during connection NI 2: 24
 in byte stream mode NI 2: 32
 message interface NI 2: 32
 routines ISDG 3: 4
 system and user ISDG 3: 4
 table of routines NI 2: 12
data types (see C language, types)
data validation tools
 characteristics SS 10: 3
 error messages SS 10: 5
 formatting SS 10: 5
 help messages SS 10: 4
 list of shell commands SS 10: 6-7
 list of visual tools SS 10: 9
 prompts SS 10: 3
 purpose SS 10: 1
 types SS 10: 2
 when to use SS 10: 1
data_ahed(3X) CHAR 11: 49
datab(D4DK) DDRM
data_behind(3X) CHAR 11: 49
datagram NUAG 1: 7
 errors NI 2: 45-46
 Internet domain NI 3: 15-16
 socket NI 3: 2
datagram_n transport type NI 5: 9
datagram_v transport type NI 5: 9
datamsq(D3DK) DDRM
date MG A: 6
 system SAG 16: 12, 15
date(1) SAG 16: 15; UG 9: 23; URM
dbm MG A: 15
dbm(3) NI
dbm(3X) BSD
dbm_clearerr(3) BSD/XNX 1: 27
dbmclose(3) BSD/XNX 1: 26
dbm_close(3) BSD/XNX 1: 27
dbm_delete(3) BSD/XNX 1: 27

- dbm_error(3) BSD/XNX 1: 27
 dbm_fetch(3) BSD/XNX 1: 27
 dbm_firstkey(3) BSD/XNX 1: 27
 dbm.h header file BSD/XNX 1: 34
 dbminit MG A: 15
 dbminit(3) BSD/XNX 1: 26
 dbm_nextkey(3) BSD/XNX 1: 27
 dbm_open(3) BSD/XNX 1: 27
 dbm_store(3) BSD/XNX 1: 27
 dc MG A: 6
 dc(1) SS 1: 7; URM
 dcopy(1M) SARM
 dd MG A: 6
 dd(1M) SARM; URM
 DDI/DKI (Device Driver
 Interface/Driver–Kernel Inter-
 face) DDRM 1: 1–5
 data structures DDRM 4: 1–2
 driver entry point routines DDRM
 2: 1–3
 error codes DDRM A: 1–2
 kernel functions DDRM 3: 1–9
 migration from Release 3.2 to
 Release 4.0 MG 5: 22
 migration from Release 3.2 to
 Release 4.0 Version 1.0 DDRM
 B: 1–5
 deadlock (file and record locking) SS
 3: 16, 18
 deassign MG A: 27
 debugger (see sdb(1))
 debugging with raw RPC NI 7: 36
 debugging with rpcgen(1) NI 6: 29
 decimal_to_double(3) BSD/XNX 1: 27
 decimal_to_extended(3) BSD/XNX
 1: 27
 decimal_to_floating(3) BSD
 decimal_to_single(3) BSD/XNX 1: 27
 declarations CGP 3: 21–29
 default title
 description SS 9: 14
 example SS 9: 14
 format SS 9: 14
 #define CGP 2: 2, 33, 3: 13–14
 with rpcgen(1) NI 6: 24
 definition, function CGP 3: 29–30
 defopen MG A: 29
 defread MG A: 29
 delay Function ISDG 3: 24
 delay(D3DK) DDRM
 delch MG A: 15
 delete MG A: 15
 delete command (FACE) UG 5: 54,
 C: 6
 delete(3) BSD/XNX 1: 26
 deleteln MG A: 15
 delsysadm(1M) SAG D: 5–6; SARM; SS
 9: 1–3
 delta MG A: 12
 delta(1) CGP 10: 4, 12, 23–26; PRM
 delwin MG A: 15
 demand backups SAG 3: 33, 45–46
 depend(4) ISDG 2: 13–14; SS 8: 16–17
 deroff(1) BSD; BSD/XNX 1: 15; URM
 DES authentication NUAG 18: 3
 description descriptor CHAR 2: 19,
 3: 9
 example of CHAR 3: 22
 descriptors
 default values CHAR 1: 12
 definitions for form frames CHAR
 3: 32–45
 definitions for menu frames CHAR
 3: 2–11
 definitions for text frames CHAR
 3: 56–61

- evaluation order CHAR A: 17
- lists of CHAR 2: 15-30
- statement syntax CHAR 2: 15
- types of CHAR 2: 15
- deserialize NI 5: 2, 16
- desk calculator programs SS 1: 7
- /dev
 - directory SAG 15: 4, 6
 - null UG 9: 71
 - restructured for Release 4.0 MG 5: 19
 - zero SS 7: 8
- /dev directory ISDG 3: 13
- devattr(1M) SAG 15: 38-40; SARM
- Devconfig file (BNU) SAG 7: 56
 - setup SAG 7: 23
- devflag ISDG 3: 17
- devfree(1M) SAG 15: 48; SARM
- device SCSI 4: 15
 - add a SAG 15: 14-15, 35-36
 - alias SAG 15: 31-32
 - block SAG 15: 7-8
 - character SAG 15: 7-8
 - display information SAG 15: 17-18
 - grouping SAG 15: 42-46
 - identification files SAG 15: 6-8
 - ls -l listing SAG 15: 6-7
 - remove SAG 15: 25-30
 - reservations SAG 15: 47-48
 - types SAG 15: 4-5
- device database SAG 15: 31-41
 - alias attribute SAG 15: 31-32
 - attributes SAG 15: 31-35
 - create an entry SAG 15: 35-36
 - list device attributes SAG 15: 38-40
 - list devices SAG 15: 36-38
 - modify entry SAG 15: 40-41
 - remove entry SAG 15: 41

- device #defines ISDG 3: 44
- device driver
 - activities ISDG 3: 3
 - critical sections ISDG 3: 7
 - definition ISDG 3: 1
 - entry points ISDG 3: 19
 - generic ISDG 3: 2
 - interrupt section ISDG 3: 6
 - object module ISDG 3: 49
 - synchronous section ISDG 3: 6
- Device Driver Interface (see DDI/DKI)
- device drivers
 - base system installed ISDG 3: 56
 - block type ISDG 3: 12
 - character type ISDG 3: 12
 - compiling ISDG 3: 61
 - development procedures ISDG 3: 62-64
 - DMA allocation routines ISDG 3: 11
 - interfacing changes for Release 4.0 MG 5: 22
 - number allowed ISDG 3: 40
 - packaging ISDG 3: 61
 - pseudo type ISDG 3: 12
 - writing code for ISDG 3: 61
- device files SAG 15: 6-8
 - block SAG 15: 7-8
 - character SAG 15: 7-8
 - major and minor numbers SAG 15: 6-7
- device groups SAG 15: 42-46
 - create SAG 15: 42
 - list SAG 15: 42-45
 - modify SAG 15: 45-46
 - remove SAG 15: 46
- device management menu SAG 15: 1-3

- device names NI 2: 70, 4: 5; SAG A: 2-4
 - diagnostics SAG 4: 11
- device nodes ISDG 3: 12
- device numbers STRM 9: 5-6
- device partitions SAG 15: 9-12
 - floppy diskette SAG 15: 10-11
 - hard disk SAG 15: 11
 - tape SAG 15: 11-12
- device (storage)
 - management tips SAG 15: 13
 - verify usability of media SAG 15: 55
 - WORM (write-once-read-many) SAG 15: 5
- Devices file (BNU) SAG 7: 33-39
 - and uucp command UG 12: 6
- device.tab (see device database)
- devinfo(1M) SAG 15: 18
- devnm MG A: 6
- devnm(1M) SARM
- devreserv(1M) SAG 15: 47-48; SARM
- df MG A: 7
- df(1) BSD; BSD/XNX 1: 18
- df(1M) SAG 5: 39, 8: 9; SARM; URM
- dfile ISDG 3: 14
- dfmounts(1M) NUAG 9: 22, 15: 22; SARM
 - used to display mounted NFS resources NUAG 16: 4
- DFS Administration NUAG 7: 6
 - command syntax NUAG 7: 8
 - commands NUAG 7: 7
 - files NUAG 7: 7
 - installation prerequisites NUAG 8: 2
 - menus NUAG, C: 1-5
 - operating states (run levels) NUAG 8: 4
- dfshares(1M) NUAG 9: 20, 15: 20; SARM
 - used to display available NFS resources NUAG 16: 2
- dfstab(4) NUAG 9: 4, 15: 4, 18: 8; SARM
 - used as input to shareall(1M) NUAG 9: 6, 15: 6
- dgmon(8) SAG 4: 7, 13, 6: 8
 - error messages SAG E: 30-31
- dgn diagnostics command SAG 4: 11, 20
- diagnostics
 - automounter problems NUAG 19: 18-20
 - bad blocks SAG 4: 30, 33-34
 - continuous SAG 4: 11
 - default SAG 4: 4, 21
 - demand SAG 4: 5, 22
 - device name SAG 4: 11
 - disk errors SAG 4: 4
 - firmware state SAG 4: 7
 - fixing hung NFS programs NUAG 17: 9
 - hardware SAG 4: 4
 - interactive SAG 4: 5, 23
 - monitor program SAG 4: 7, 13
 - NFS client problems NUAG 17: 6
 - NFS mounting problems NUAG 17: 6-8
 - NFS server problems NUAG 17: 5-6
 - phases SAG 4: 4-5, 13
 - recovering data SAG 4: 42
 - repetitive SAG 4: 11
 - sysadm(1M) SAG 4: 2
 - system state 1 errors SAG 4: 36
 - system trouble SAG 4: 24
 - unconditional SAG 4: 11
- dial MG A: 15
- dial terminal (see ct)

- dial UNIX computer (see cu)
- dial(3C) NI
- Dialcodes file (BNU) SAG 7: 47
- Dialers file (BNU) SAG 7: 39-41
- dial-up password SAG 12: 9-13
- diff MG A: 7
- diff(1) UG 3: 57-59; URM
- diff3 MG A: 7
- diff3(1) URM
- diffmk(1) BSD; BSD/XNX 1: 15
- difftime(3C) PRM
- dir(4) SARM
- dircmp MG A: 7
- dircmp(1) URM
- direct memory access ISDG 3: 9
- directory SAG 5: 3; UG 1: 9-14, 3: 2
 - bin UG 9: 40
 - change UG 3: 24-25
 - compress SAG 8: 5-7
 - create UG 3: 15-16
 - current UG 3: 6-7
 - /etc SAG B: 11-14
 - home UG 3: 4-6
 - list contents of UG 3: 17-23
 - naming rules UG 3: 14
 - remove UG 3: 26-27
 - root SAG B: 7-10; UG 1: 11-13
 - size of SAG 8: 5
 - tree structure UG 3: 2
 - UNIX System directory changed for Release 4.0 MG 5: 14-20
 - /usr SAG B: 25-27
 - /var SAG B: 30-33
- directory(3C) PRM
- dirent(4) SARM
- dirname MG A: 7
- dirname(3G) PRM
- dis(1) CGP 1: 12
- disable MG A: 7
- disabling FMLI commands CHAR 4: 17
- disconnect requests NI 2: 26-31, 33, 35-38, 49-51, 57-60, 63, 69
- discriminated union (XDR) NI 8: 15
- disk
 - address SAG 4: 38
 - bootable SAG 6: 44-48
 - device names SAG A: 2-4
 - error diagnostics SAG 4: 4
 - formatting SAG 5: 23
 - hard (see hard disk)
 - partitions SAG 5: 23, A: 5-8
 - space swapping, changed for Release 4.0 MG 5: 23
- disk(7) SARM
- diskadd(1M) SARM
- diskcmp MG A: 7
- diskcp MG A: 7
- diskette
 - device names SAG A: 2
 - floppy (see floppy diskette)
 - partitions SAG A: 10
- disksetup(1M) SARM
- diskusg(1M) SARM
- dispadm(1M) SAG 10: 16-19; SARM
- dispatcher (see scheduler)
- dispgid(1) SARM; SS
- display command (FACE) UG 5: 59, C: 6
- display(7) SARM
- DISPLAYH variable CHAR 2: 11
- DISPLAYW variable CHAR 2: 11
- dispuid(1) SARM; SS
- Distributed File System Administration (see DFS Administration)
- distributed file systems

- displaying information about NUAG
 - 9: 19–24, 15: 19–24
 - setting a default type NUAG 8: 3
 - types of NUAG 7: 1
- div(3C) PRM
- divvy MG A: 7
- DKI (Driver–Kernel Interface) (see DDI/DKI)
- dlclose(3X) PRM
- dLError(3X) PRM
- dlopen(3X) PRM
- dlsym(3X) PRM
- DMA ISDG 3: 9, 36
- DMA allocation routines ISDG 3: 11
- DMA controller ISDG 3: 38
- dma_buf(D4D) DDRM
- dma_cb(D4D) DDRM
- dma_disable(D3D) DDRM
- dma_enable(D3D) DDRM
- dma_free_buf(D3D) DDRM
- dma_free_cb(D3D) DDRM
- dma_get_best_mode(D3D) DDRM
- dma_get_buf(D3D) DDRM
- dma_get_cb(D3D) DDRM
- dma_pageio(D3DK) DDRM
- dma_prog(D3D) DDRM
- dma_stop(D3D) DDRM
- dma_swsetup(D3D) DDRM
- dma_swstart(D3D) DDRM
- DMD
 - 5620 CHAR A: 1
 - 630 CHAR A: 1; SAG C: 14
- dmesg MG A: 27
- dname(1M) NUAG; SARM
- doconfig() NI 13: 6, 21–23, 25
- doconfig(3N) NI; SAG 13: 7, 19–21
- documentation, related ISDG 1: 5
- documenting driver installation ISDG
 - 3: 71
- domain NUAG 10: 3, 5
 - add/delete members NUAG 12: 6–7
 - establishing NUAG 2: 5
 - local administrative NUAG 2: 8
 - multiple name service NUAG 12: 11
 - name server NUAG 12: 62–65
 - primary NUAG 10: 5–6
 - primary name server NUAG
 - 12: 62–65
 - registering NUAG 2: 9
 - registration form guidelines NUAG
 - B: 1–4
 - root-level NUAG 2: 7
 - secondary NUAG 10: 6
 - secondary name server NUAG 12: 63
 - second-level NUAG 2: 7
 - selecting a name NUAG 2: 8
 - set name NUAG 12: 3
 - share table NUAG 12: 39–40
 - sharing resources between NUAG
 - 12: 9–10
 - top-level NUAG 2: 7
 - YP NI 11: 3, 7–8
- domain name service
 - administrative zones NUAG 5: 1–4
 - configuring a name server NUAG
 - 5: 13–37
 - overview NUAG 5: 1
 - setting up the resolver NUAG 5: 9–12
- Domain Name Service (DNS) NUAG
 - 1: 9
- domain names NUAG 18: 7
 - YP NI 11: 3
- domainname(1M) NI
- done command (FMLI) CHAR 2: 32
- done descriptor CHAR 2: 18, 20, 23,
 - 3: 5, 35, 58
- dos MG A: 27

- doscat MG A: 27
- doscp MG A: 27
- dosdir MG A: 27
- dosformat MG A: 28
- dosld MG A: 12
- dosls MG A: 28
- dosmkdir MG A: 28
- dosrm MG A: 28
- dosrmdir MG A: 28
- dot UG 3: 10
- dot dot UG 3: 10
- double CGP 3: 21, 32
- double_to_decimal(3) BSD/XNX 1: 28
- doupdate(3X) CHAR 8: 3
- do-while statement CGP 3: 49-50
- downstream, definition STRM 2: 3
- dparam MG A: 7
- drand48 MG A: 15
- drand48(3C) PRM
- driver
 - block DDRM 3: 4-7
 - classification STRM 9: 1
 - configuration STRM 9: 2
 - device (see device drivers)
 - device numbers STRM 9: 5-6
 - entry points DDRM 2: 1-3; STRM 9: 3
 - functions DDRM 3: 1-7
 - interface to STREAMS STRM 7: 37-41
 - MULTIBUS II DDRM 3: 7-9
 - overview STRM 9: 1-6
 - porting DDRM 1: 2
 - STREAMS DDRM 3: 2-3; STRM 2: 14, 9: 6-8
 - STREAMS-based console STRM G: 10-14
 - STREAMS-based ports STRM G: 2-9
 - STREAMS-based sxt STRM 12: 3
 - STREAMS-based xt STRM G: 15-34
 - structures DDRM 4: 1-2
 - writing a driver STRM 9: 3-5
- Driver portability SCSI 1: 3
- driver routine
 - kenter ISDG 3: 19
 - kexit ISDG 3: 20
 - poll ISDG 3: 19
- driver software package ISDG 3: 48
- Driver-Kernel Interface (see DDI/DKI)
- driver.o file ISDG 3: 49
- drv_getparm(D3DK) DDRM
- drv_hztousec(D3DK) DDRM
- drv_priv(D3DK) DDRM
- drv_usectohz(D3DK) DDRM
- drv_usecwait(D3DK) DDRM
- DSP ISDG 3: 1, 35, 48
- DSP module
 - driver.o ISDG 3: 49
 - init ISDG 3: 52
 - master ISDG 3: 49
 - node ISDG 3: 51
 - postinstall ISDG 3: 53
 - preremove ISDG 3: 54
 - rc ISDG 3: 52
 - shutdown ISDG 3: 52
 - space.c ISDG 3: 50
 - system ISDG 3: 50
- dtype MG A: 28
- du MG A: 7
- du(1M) BSD; BSD/XNX 1: 18; SAG 5: 41, 8: 9; SARM; URM
- dump MG A: 7
- dump (system) SAG 4: 27
- dump(1) CGP 1: 12; PRM
- dump(4) SARM
- dumpdir MG A: 7

dup MG A: 15
 dup2 MG A: 15
 dup(2) PRM
 dup2(3C) PRM
 dupb(D3DK) DDRM
 dup_field(3X) CHAR 11: 9
 dupmsg(D3DK) DDRM
 dynamic frame (FMLI)
 example of CHAR 3: 28
 generation CHAR 2: 7
 dynamic linking CGP 2: 2-3, 10-32
 implementation CGP 2: 2-3, 21-23,
 13: 30-38, 50-69
 dynamic_field_info(3X) CHAR
 11: 16

E

EBUSY error code SCSI 2: 6, 11
 echo MG A: 7, 15
 echo(1) BSD; BSD/XNX 1: 14; UG 9: 4-5,
 64; URM
 echo(1F) CHAR 2: 39, , C: 1
 echo(3X) CHAR 7: 41
 econvert(3) BSD; BSD/XNX 1: 28
 ecvt MG A: 15
 ecvt(3C) PRM
 ed MG A: 7
 ed(1) UG 6: 1; URM
 add text UG 6: 29-35
 change text UG 6: 42-48
 command format UG 6: 13
 command mode UG 6: 4
 delete text UG 6: 6, 38-39
 delimiters UG 6: 48
 display non-printing characters UG
 6: 76-77
 display text UG 6: 5, 26-28
 global search UG 6: 22-23
 global substitute UG 6: 46-48
 help command UG 6: 73-76
 input mode UG 6: 4
 line addressing UG 6: 14-24
 move around in file UG 6: 7-8
 move text UG 6: 63-71
 patterns UG 6: 19-21, 24, 51-60
 print current filename UG 6: 77-79
 quick reference UG D: 1-4
 quit UG 6: 9-10
 recover from hangup UG 6: 80
 regular expressions UG 6: 51-60
 relative addressing UG 6: 17-19
 save edited file UG 6: 8-9
 search for text UG 6: 19-22, 24, 51-60
 shell escape UG 6: 79-80
 special characters UG 6: 51-60
 substitute text UG 6: 42-48
 undo previous command UG
 6: 39-41
 edata MG A: 15
 edit descriptor CHAR 2: 23, 3: 58
 edit(1) URM
 edquota(1M) SARM
 edsysadm(1M) SAG D: 4; SARM; SS
 9: 1-2
 EDT SCSI 4: 15
 EDT (Equipped Device Table) SAG
 4: 5, 11, 6: 39
 error messages SAG E: 32-33
 edvtoc(1M) SARM
 EFAULT error code SCSI 2: 4, 11
 EFT Data Types ISDG 4: 30
 egrep MG A: 7
 egrep(1) URM; XNX
 EINVAL error code SCSI 2: 11

Master Subject Index

ELF CGP 13: 1-69 (see also object files)
elf(3E) PRM
elf_begin(3E) PRM
elf_cntl(3E) PRM
elf_end(3E) PRM
elf_error(3E) PRM
elf_fill(3E) PRM
elf_flag(3E) PRM
elf_fsize(3E) PRM
elf_getarhdr(3E) PRM
elf_getarsym(3E) PRM
elf_getbase(3E) PRM
elf_getdata(3E) PRM
elf_getehdr(3E) PRM
elf_getident(3E) PRM
elf_getphdr(3E) PRM
elf_getscn(3E) PRM
elf_getshdr(3E) PRM
elf_hash(3E) PRM
elf_kind(3E) PRM
elf_next(3E) PRM
elf_rand(3E) PRM
elf_rawfile(3E) PRM
elf_strptr(3E) PRM
elf_update(3E) PRM
elf_version(3E) PRM
elf_xlate(3E) PRM
#elif CGP 3: 15-16
ellipsis notation CGP 3: 28; CTG3, 7, 46
#else CGP 3: 15-16
else statement CGP 3: 48
emergency recovery procedure ISDG 3: 65-67
enable MG A: 7
enable(1) UG 8: 12-13, 25, 31; URM
enableok(D3DK) DDRM
encoded transmission, in xt driver STRM G: 33
encryption keys NUAG 18: 3
end MG A: 15
end(3C) PRM
endgrent MG A: 15
#endif CGP 3: 15-16
endpwent MG A: 15
endusershell(3) BSD/XNX 1: 27
endutent MG A: 15
endwin MG A: 15
endwin(3X) CHAR 6: 3
ENOMEM error code SCSI 2: 11
enumeration (enum) CGP 3: 24
env MG A: 7
env(1) URM
environ(4) SARM
environ(5) NI; PRM; SARM
environment, login UG 4: 8-9, 9: 89-94, GL: 4
environment variables SAG 17: 26-28
envp CGP 1: 6
eqn(1) BSD; BSD/XNX 1: 15
eqnchar(7) BSD; BSD/XNX 1: 15
equipped device table (see EDT)
erand48 MG A: 15
erase MG A: 15
erase(3X) CHAR 7: 10
erf MG A: 15
erf(3M) PRM
erfc MG A: 15
errno MG A: 15
#error CGP 3: 19
error handling SS 2: 2-3
error messages
NOTICE SAG E: 2-5
PANIC SAG E: 12-19
WARNING SAG E: 6-11
Error Reporting ISDG 3: 31
errors log (BNU) SAG 7: 66

- esballoc(D3DK) DDRM
- esbcall(D3DK) DDRM
- escape character, telnet(1) NUAG 24: 7
- escape sequences CGP 3: 8
- escape special character UG 2: 8
- /etc
 - default/su SAG 12: 17
 - dialup SAG 12: 9
 - directories SAG B: 11-14
 - d_passwd SAG 12: 9-13
 - files SAG B: 15-24
 - master.d/kernel BSD/XNX 1: 1
 - master.d/xnamfs BSD/XNX 1: 3
 - passwd SAG 16: 15
- /etc/conf directory ISDG 3: 42
- etext MG A: 15
- ethers(3N) NI; NUAG
- ethers(4) NUAG; SARM
- ETI SS 1: 9-10
 - basic programming CHAR 6: 1-12
 - colors (see colors)
 - compile program CHAR 6: 6
 - components CHAR 5: 8-10
 - connection with terminfo CHAR 5: 7
 - forms (see forms)
 - header files CHAR 6: 2-3
 - input options CHAR 7: 38-42
 - input options settings CHAR 7: 38
 - input routines CHAR 7: 14-21
 - libraries CHAR 5: 5-6
 - lines and columns CHAR 6: 8
 - low-level interface (curses) CHAR 8: 11-13
 - menus (see menus)
 - output attributes CHAR 7: 22-36
 - output routines CHAR 7: 2-13
 - panels (see panels)
 - program examples (see ETI examples)
 - routines for drawing lines and other graphics CHAR 12: 2-3
 - routines for soft labels CHAR 12: 4-5
 - run program CHAR 6: 7
 - windows CHAR 6: 9
 - windows (see windows)
 - working with more than one terminal CHAR 12: 6-7
- ETI examples CHAR E: 1-19
 - colors program CHAR E: 19
 - editor program CHAR E: 1
 - highlight program CHAR E: 8
 - scatter program CHAR E: 10
 - show program CHAR E: 12
 - two program CHAR E: 14
 - window program CHAR E: 17
- EUC handling in ldterm(7) STRM 12: 8
- event handling NI 2: 26-27
- event-driven TLI software, example of NI 2: 52-60
- events, definition ISDG 3: 7
- ex MG A: 7
- ex(1) URM
- exec(2) PRM; SS 2: 8-10
- execl MG A: 15
- execle MG A: 15
- execlp MG A: 15
- execseg MG A: 15
- executable files CGP 2: 4-6, 13: 1
 - interrupting in FMLI CHAR 2: 44
 - use of (FACE) UG C: 24
- executing commands on another machine NUAG 23: 1
- executing new kernel ISDG 3: 58

execv MG A: 15
execve MG A: 16
execvp MG A: 16
exit MG A: 16
exit command
 FACE UG 5: 36, C: 23
 FMLI CHAR 1: 14, 2: 33, A: 10
exit(2) CGP 2: 51–52, 7: 20; PRM
EXLUN value SCSI 4: 12
exp MG A: 16
exp(3M) PRM
expedited data NI 2: 35; STRM 5: 3,
 7: 35
export shell command UG 9: 82
expr MG A: 7
expr(1) URM
expressions CGP 3: 33–46
 grouping and evaluation in
 CTG36–39
exstr(1) URM
Extended logical unit SCSI 4: 12
extended STREAMS buffers STRM
 5: 60–61
 allocation STRM 5: 60
 freeing STRM 5: 61
 in different hardware STRM
 G: 35–38
extended_to_decimal(3) BSD/XNX
 1: 28
extern CGP 3: 25–26
External Data Representation (see
 XDR)
external device number STRM 9: 5–6

F

fabs MG A: 16

FACE
 (see files and file folders (FACE)
 also)
 (see frames (FACE) also)
 administration UG 5: 82–90
 bin directory UG 5: 84
 cancel commands UG C: 3
 changed for Release 4.0 MG 5: 38
 choices menus UG 5: 24
 cleaning up screen UG C: 14
 command line UG 5: 5
 Command Menu UG 5: 30, C: 3
 customizing your FACE Office UG
 5: 28
 default editor UG 5: 69
 default \$HOME UG 5: 41
 defined UG 5: 1
 executables, use of UG C: 24
 exiting from UG C: 23
 filecabinets, access other users' UG
 5: 62
 global programs administration UG
 5: 86–90
 help UG 5: 32–35, C: 23
 logging in UG 5: 2–3
 logging out UG 5: 36
 mail UG 5: 89, C: 19–20
 message line UG 5: 5
 navigation UG 5: 12, 20, 25–27, 35,
 C: 14
 Office Functions UG 5: 68
 ott(4) files UG 5: 84
 pathnames in UG 5: 41
 pref directory UG 5: 84
 Preferences Menu UG 5: 63, C: 11
 Printer Operations UG 5: 72–74,
 C: 13
 .profile created for users UG 5: 84

Programs UG 5: 77, C: 18-19
 Programs Administration UG
 5: 78-81, C: 17
 refreshing the screen UG C: 23
 return from the UNIX System UG
 C: 25
 running shell scripts UG 5: 94-95,
 C: 24
 screen deterioration UG 5: 2
 screen structure UG 5: 4-6
 screen-labeled function keys UG 5: 5
 special characters UG 5: 40, 49
 spell checker UG 5: 89, C: 21
 status of UG 5: 4
 title line UG 5: 4
 tmp directory UG 5: 84
 UNIX System access UG 5: 90-92
 use of executables UG 5: 93-94
 use of other applications UG 5: 77
 wastebasket UG 5: 55, 84
 work area UG 5: 5
 face(1) URM
 facesuspend (FACE) UG 5: 96
 factor MG A: 7
 factor(1) URM
 false MG A: 7
 fastboot(1M) BSD; BSD/XNX 1: 22
 fasthalt(1M) BSD/XNX 1: 22
 fattach(3C) STRM 11: 7
 fclose MG A: 16
 fclose(3S) CGP 2: 52; PRM
 fcntl MG A: 16
 fcntl(2) PRM; SS 3: 6, 8, 13-15
 fcntl(5) PRM; SARM
 fcol descriptor CHAR 3: 39
 fconvert(3) BSD/XNX 1: 28
 fcvt MG A: 16
 fd(4) SARM
 fd(7) SARM
 FD_CLR macro NI 3: 18
 fdetach(1M) STRM
 fdetach(3C) STRM 11: 8
 fdisk MG A: 7
 fdisk backup method SAG 3: 27
 fdopen MG A: 16
 fdp backup method SAG 3: 27
 fdp(1M) SARM
 FD_SET macro NI 3: 18
 feof MG A: 16
 ferror MG A: 16
 ferror(3S) PRM
 fetch MG A: 16
 fetch(3) BSD/XNX 1: 26
 ff(1M) SARM
 ffile backup method SAG 3: 19
 ffile(1M) SARM
 fflush MG A: 16
 ffs(3C) PRM
 fgetc MG A: 16
 fgets MG A: 16
 fgrep MG A: 7
 fgrep(1) URM; XNX
 field (FMLI)
 scrollable CHAR A: 13
 validation CHAR A: 13
 values CHAR A: 14
 field item help message
 description SS 9: 14
 example SS 9: 15
 format SS 9: 15
 field_arg(3X) CHAR 11: 20
 field_back(3X) CHAR 11: 29
 field_buffer(3X) CHAR 11: 31
 field_count(3X) CHAR 11: 48
 field_fore(3X) CHAR 11: 29
 field_index(3X) CHAR 11: 81

- field_info(3X) CHAR 11: 13
- field_init(3X) CHAR 11: 77
- field_just(3X) CHAR 11: 27
- fieldmsg descriptor CHAR 2: 21, 3: 38
- field_opts(3X) CHAR 11: 38
- field_opts_off(3X) CHAR 11: 41
- field_opts_on(3X) CHAR 11: 41
- field_pad(3X) CHAR 11: 29
- fields, awk(1) UG 10: 4
- field_status(3X) CHAR 11: 33
- field_term(3X) CHAR 11: 77
- field_type(3X) CHAR 11: 20
- field_userptr(3X) CHAR 11: 35
- FIFO (STREAMS) STRM 11: 1
 - basic operations STRM 11: 1-6
 - flush STRM 11: 6
 - queue scheduling STRM 4: 2
- file MG A: 7
- file and record locking SS 2: 19-20, 3: 1-19
- file descriptor passing STRM 11: 9
- file mode (see permissions, file)
- file operations
 - BSD/XENIX unification with Release 4.0 MG 5: 8
 - changed for Release 4.0 MG 5: 7-8
 - memory mapping changed for Release 4.0 MG 5: 7
 - number open simultaneously changed for Release 4.0 MG 5: 7
 - POSIX conformance and Release 4.0 MG 5: 8
- FILE structure CGP 2: 51-52
- file system SAG 5: 3; UG 1: 2, 9-14
 - bfs SAG 5: 19-22, 6: 9
 - boot (see boot file system)
 - boot and autoconfig changed for Release 4.0 MG 5: 11
 - changed for Release 4.0 MG 5: 9-13
 - compress SAG 8: 6
 - consistency (see fsck(1M))
 - copy SAG 15: 19-22
 - create SAG 5: 30-36
 - free space in SAG 8: 9 (see also df(1M))
 - linking changed for Release 4.0 MG 5: 12
 - maintenance SAG 5: 39-43
 - mount SAG 5: 37-38
 - organization SAG 5: 3-5
 - organizing UG 3: 15-27
 - repair (see fsck(1M))
 - root (see root file system)
 - s5 SAG 5: 6-11
 - status system calls changed for Release 4.0 MG 5: 12
 - storage device SAG 5: 23-26
 - structure UG 3: 2-3, A: 1-4
 - types changed for Release 4.0 MG 5: 10
 - types defined NUAG 7: 3
 - ufs SAG 5: 12-18
 - unmount SAG 6: 15
 - used space in SAG 8: 9 (see also du(1M))
 - VFS changed for Release 4.0 MG 5: 9
 - virtual (see virtual file systems)
- file transfer
 - aborting NUAG 25: 5
 - getting a listing of remote files NUAG 25: 3
 - preparing the connection NUAG 25: 1-2
 - quitting NUAG 25: 5
 - using ftp NUAG 25: 3

- file(1) URM
- filecabinet (FACE), accessing other users' UG 5: 62
- filehdr(4) SARM
- fileno MG A: 16
- files
 - advanced commands UG 3: 56–64
 - basic commands UG 3: 28–56
 - change ownership UG 3: 56–57
 - compare contents UG 3: 58–59
 - concatenate UG 3: 29–32
 - copy to disk SAG 15: 19
 - copy to floppy SAG 15: 19–20
 - copy to tape SAG 15: 19
 - create in ed(1) UG 6: 3–4
 - create in vi(1) UG 7: 7
 - creation mask SAG 17: 28
 - directory UG 1: 11
 - display contents UG 3: 29–39
 - /etc SAG B: 15–24
 - linking changed for Release 4.0 MG 5: 12
 - lock SS 2: 19–20
 - locking (see locking)
 - make a copy UG 3: 40–42
 - mapped (see mapped files)
 - memory-mapped (see mapped files)
 - merge UG 3: 62–64
 - move UG 3: 43–45
 - naming rules UG 3: 14
 - ordinary UG 1: 11, 3: 2
 - ownership SS 6: 3, 12
 - page through UG 3: 32–37
 - permissions SAG 12: 22–24, 17: 6; UG 3: 49–56
 - print (see printing)
 - protecting UG 3: 49–56
 - remove UG 3: 45–46
 - rename UG 3: 43–45
 - renamed in Release 4.0 SAG B: 2–6
 - retrieve from public directory (see uupick)
 - search for a pattern UG 3: 60–61
 - security SAG 12: 21–24
 - size of UG 3: 47–49
 - sort UG 3: 62–64
 - special UG 1: 11–14, 3: 2, GL: 12
 - status system calls changed for Release 4.0 MG 5: 12
 - transfer (see send files)
 - /usr SAG B: 28–29
 - /var SAG B: 34–37
- files and file folders (FACE) UG 5: 38–62
 - changing display order UG 5: 50
 - copying UG 5: 46, C: 4
 - creating UG 5: 42, C: 4
 - deleting UG 5: 53, C: 5
 - description guidelines UG 5: 49
 - displaying files UG 5: 59, C: 6
 - editing files UG C: 12
 - file description UG 5: 51
 - file identifiers UG 5: 39, 51
 - file permissions UG 5: 61
 - files defined UG 5: 38
 - finding UG 5: 56, C: 7
 - folders defined UG 5: 38
 - folders nested UG 5: 43
 - moving UG C: 8
 - naming guidelines UG 5: 39
 - organizing UG 5: 50, C: 8
 - permissions UG 5: 60–62, 67, C: 10–11
 - printing files UG 5: 74
 - redescribing UG 5: 48, C: 9
 - renaming UG 5: 47, C: 10

- returning to suspended files UG 5: 97-98, C: 25
- security for files UG 5: 60-62
- suspending files UG 5: 96-97, C: 25
- undeleting UG 5: 55, C: 12
- filesystem(7) SARM
- filledt program SAG 6: 8
- fimage backup method SAG 3: 27
- fimage(1M) SARM
- find MG A: 7
- find command (FACE) UG 5: 56, C: 7
- find(1) SAG 5: 40, 8: 4, 6; URM
- finger MG A: 7
- finger(1) NUAG 26: 1; URM
- fingerd(1M) NUAG; SARM
- firmware
 - error messages SAG E: 34-35
 - password recovery SAG 12: 20, 16: 8
- firmware mode SAG 6: 9-12, 24, 30
 - returning from SAG 6: 43
- firmware state, diagnostic monitor program SAG 4: 7
- firstkey MG A: 16
- firstkey(3) BSD/XNX 1: 26
- fixhdr MG A: 28
- fixperm(1) XNX
- fixperm(1M) SARM
- flash(3X) CHAR 7: 37
- float CGP 3: 21, 32
- floating point constants CGP 3: 7
- floating point operations CGP 14: 1-22
 - binary-decimal conversion CGP 14: 13-15, 22
 - compares CGP 14: 21
 - data representation CGP 14: 2-7
 - data types and formats CGP 14: 2
 - denormalized numbers CGP 14: 4
 - double-extended CGP 14: 19
 - double-precision CGP 14: 3
 - exception handling CGP 14: 8-12
 - exception handling (3B2) CGP 14: 10
 - exception handling (6386) CGP 14: 12
 - exceptions CGP 14: 8
 - floating point to integer conversion CGP 14: 20
 - IEEE requirements CGP 14: 20
 - infinities CGP 14: 6
 - infinities I/O CGP 14: 21
 - maximum and minimum values CGP 14: 4
 - NaNs CGP 14: 6
 - NaNs I/O CGP 14: 21
 - normalized numbers CGP 14: 3
 - rounding CGP 14: 7
 - single-precision CGP 14: 2, 16-18
 - single-precision functions CGP 14: 18
 - special-case values CGP 14: 5
 - square root CGP 14: 20
 - sticky bits CGP 14: 8
 - trap bits CGP 14: 8
 - unordered condition CGP 14: 21
- floating types
 - conversion CGP 3: 32
 - declaration CGP 3: 21
- floatingpoint(3) BSD
- floatingpoint.h header file BSD/XNX 1: 34
- floating_to_decimal(3) BSD
- floor MG A: 16
- floor(3M) PRM
- floppy diskette
 - copy floppy to floppy SAG 15: 20
 - copy from disk SAG 15: 19
 - display information about SAG 15: 17-18

- erase SAG 15: 23
 format SAG 15: 15-17
 partition SAG 15: 10-11
 floppy diskette drive, remove SAG
 15: 25-30
 floppy key SAG 6: 33, 16: 8-11
 flow control STRM 5: 28-32
 definition STRM 2: 7
 in driver STRM 9: 16
 in line discipline module STRM 8: 11
 in module STRM 8: 10-12
 in xt driver STRM G: 32
 routines STRM 5: 29-32
 fltboot(1M) SAG 6: 27
 flush handling
 description STRM 7: 31-35
 flags STRM 7: 31, B: 16
 in driver STRM 9: 12
 in line discipline STRM 7: 31
 in pipes and FIFOs STRM 11: 6
 priority band data STRM 7: 35
 priority band data example STRM
 7: 36
 read-side example STRM 7: 34
 write-side example STRM 7: 33
 flushband(D3DK) DDRM
 flushq(D3DK) DDRM
 fmlcut(1F) CHAR 2: 39, , A: 17, C: 1
 fmlexpr(1F) CHAR 2: 39, , C: 1
 fmlgrep(1F) CHAR 2: 39, , C: 1
 FMLI SS 1: 8-9; UG 5: 84
 changed with Release 4.0 MG 5: 38
 commands CHAR 2: 31-38
 comments CHAR 2: 2
 disabling commands CHAR 4: 18
 file type casts CHAR 2: 2, 34
 filename conventions CHAR 2: 2,
 4: 21
 filename expansion CHAR A: 12
 overview CHAR 1: 2-14
 redirection of input and output
 CHAR 2: 8, 4: 21, A: 10
 re-evaluation of descriptors CHAR
 2: 4-5
 referencing variables CHAR 2: 12-14
 screen layout CHAR 1: 3-5
 special characters CHAR 2: 5-6
 syntax CHAR 2: 2-9
 use on asynchronous terminal
 CHAR 1: 11
 using an application CHAR 1: 15-25
 fml(1) CHAR 1: 2, 13, 3: 37, , C: 1; URM
 command syntax CHAR 4: 21
 fmod MG A: 16
 fmt(1) URM
 fmtmsg(1) URM
 fmtmsg(3C) PRM
 Fn variable CHAR 2: 11
 fold(1) URM
 fopen MG A: 16
 fopen(3) BSD/XNX 1: 30
 fopen(3S) BSD; CGP 2: 51-52; PRM
 for statement CGP 3: 50
 foreground execution UG GL: 5
 foreground job, in job control STRM
 6: 9
 Foreign log (BNU) SAG 7: 74
 fork MG A: 16
 fork(2) PRM; SS 2: 8-10
 Form and Menu Language Interface
 (see FMLI)
 form descriptor CHAR 2: 20, 3: 35
 format MG A: 7
 formatting storage media SAG
 15: 15-17
 Form_Choice variable CHAR 2: 11,
 A: 14

- form_cursor(3X) CHAR
- form_data(3X) CHAR
- form_driver(3X) CHAR 11: 59
- form_field(3X) CHAR
- form_field_attributes(3X) CHAR
- form_field_buffer(3X) CHAR
- form_field_info(3X) CHAR
- form_field_just(3X) CHAR
- form_field_new(3X) CHAR
- form_field_opts(3X) CHAR
- form_fields(3X) CHAR 11: 47
- form_fieldtype(3X) CHAR
- form_field_userptr(3X) CHAR
- form_field_validation(3X) CHAR
- form_hook(3X) CHAR
- form_init(3X) CHAR 11: 77
- form-letter generation, awk(1) UG
10: 56
- form_new(3X) CHAR
- form_new_page(3X) CHAR
- form_opts(3X) CHAR 11: 88
- form_opts_off(3X) CHAR 11: 90
- form_opts_on(3X) CHAR 11: 90
- form_page(3X) CHAR 11: 83
- form_post(3X) CHAR
- forms (ETI) CHAR 11: 1–105
 - application-defined commands
CHAR 11: 69
 - associate windows and subwindows
with a form CHAR 11: 52
 - build field type CHAR 11: 92
 - call form driver CHAR 11: 70
 - change and fetch fields CHAR 11: 47
 - change default attributes CHAR
11: 50
 - change default field attributes
CHAR 11: 19
 - change form page CHAR 11: 83–84
 - choice requests CHAR 11: 69
 - compile and link form programs
CHAR 11: 2
 - count number of fields CHAR 11: 48
 - create and free CHAR 11: 43–46
 - create and free fields CHAR 11: 9–12
 - create field type CHAR 11: 93
 - dimensions CHAR 11: 51
 - display CHAR 11: 51–58
 - driver processing CHAR 11: 59–85
 - dynamically growable fields CHAR
11: 14
 - field buffers CHAR 11: 31
 - field editing requests CHAR 11: 66
 - field size CHAR 11: 13
 - field status CHAR 11: 33
 - field types argument support CHAR
11: 97
 - field user pointer CHAR 11: 35
 - field validation requests CHAR
11: 68
 - form options CHAR 11: 88–91
 - form requests CHAR 11: 63–69
 - form user pointer CHAR 11: 86–87
 - free programmer-defined field
types CHAR 11: 96
 - helpful field features CHAR 11: 31–37
 - initialization routines CHAR
11: 76–80
 - inter-field navigation requests
CHAR 11: 63
 - intra-field navigation requests
CHAR 11: 65
 - justify data in field CHAR 11: 27
 - location information CHAR 11: 13
 - manipulate current field CHAR
11: 81–82
 - manipulate field attributes CHAR
11: 13–28

- manipulate field options CHAR
 - 11: 38-42
- manipulate form attributes CHAR
 - 11: 47-50
- move a field CHAR 11: 18
- next and previous choice functions
 - support CHAR 11: 102
- page navigation requests CHAR
 - 11: 63
- position form cursor CHAR 11: 84-85
- posting CHAR 11: 56
- presence of offscreen data CHAR
 - 11: 49
- programmer-defined field types
 - CHAR 11: 92-105
- sample program CHAR 11: 4
- scale CHAR 11: 51
- scrolling requests CHAR 11: 67
- set and fetch field user pointer
 - CHAR 11: 35
- set and read field buffers CHAR
 - 11: 31
- set and read field status CHAR
 - 11: 33
- set field background CHAR 11: 29-30
- set field foreground CHAR 11: 29-30
- set field type CHAR 11: 19
- set pad character CHAR 11: 29-30
- support programmer-defined field
 - types CHAR 11: 97-105
- termination routines CHAR 11: 76-80
- terminology summary CHAR 11: 3
- unposting CHAR 11: 56
- validation functions CHAR 11: 93
- virtual key mapping CHAR 11: 60
- what a form program does CHAR
 - 11: 4
- write form programs CHAR 11: 3-8
- forms (FACE) UG 5: 18
 - default values UG 5: 19
 - editing UG 5: 20
 - fields in UG 5: 18
 - function keys UG 5: 19
 - navigation UG 5: 20
- forms (FMLI)
 - Choices menu CHAR 1: 9, 3: 41
 - definition CHAR 1: 8-9
 - descriptor descriptions CHAR
 - 3: 32-45
 - examples of CHAR 3: 46-55
 - graphic characters in CHAR 2: 47
 - lists of descriptors CHAR 2: 20-23
 - multi-page CHAR 1: 8, 3: 41, A: 12
 - navigation in CHAR B: 1-6
 - order of descriptors CHAR 3: 32
 - positioning CHAR 3: 34
- forms (printer) SAG 9: 34-35, 45-47,
 - 63-71
- forms(3X) CHAR
- form_sub(3X) CHAR 11: 53
- form_term(3X) CHAR 11: 77
- form_userptr(3X) CHAR 11: 86
- form_win(3X) CHAR 11: 53
- forwarder NUAG 4: 3
- Foundation Set ISDG 4: 2
- fpathconf(2) PRM
- fp_class function BSD/XNX 1: 29
- fpgetround(3C) PRM
- fp.h header file BSD/XNX 1: 34
- fprintf MG A: 16
- fputc MG A: 16
- fputs MG A: 16
- fragments (ufs) SAG 5: 26
- frame definition file
 - descriptors CHAR 2: 16-25
 - overview CHAR 1: 7-9

- Framed Access Command Environment (see FACE)
- framemsg descriptor CHAR 2: 18, 20, 23, 3: 5, 35, 58
- frames (FACE)
 - active UG 5: 6
 - canceling UG C: 14
 - closing UG C: 14
 - defined UG 5: 5
 - display of UG 5: 5
 - help for UG 5: 32–34, C: 15
 - ID number UG 5: 6, 25
 - inactive UG 5: 6
 - moving UG 5: 28, C: 15
 - navigation UG 5: 12, 20, 25–27, 35, C: 14
 - reshaping UG 5: 29, C: 15
 - updating UG C: 15
- frames (FMLI)
 - definition of CHAR 1: 5–6
 - dynamically generated CHAR 2: 7
 - ID number CHAR 1: 7
 - overlap CHAR A: 11
- frame-to-frame navigation CHAR 1: 21–22
- fread MG A: 16
- fread(3S) PRM
- free MG A: 16
- freeb(D3DK) DDRM
- free_field(3X) CHAR 11: 9
- free_fieldtype(3X) CHAR 11: 96
- free_form(3X) CHAR 11: 43
- free_item(3X) CHAR 10: 9
- freemsg(D3DK) DDRM
- freerbuf(D3DK) DDRM
- free_rtn(D4DK) DDRM
- freopen MG A: 16
- frexp MG A: 16
- frexp(3C) PRM
- frm-mgmt command
 - FACE UG 5: 27–29, 97, C: 16
 - FMLI CHAR 2: 33, A: 18
- fromsmtp(1M) SARM
- frow descriptor CHAR 2: 21, 3: 39
- fs(4) SARM
- fsba(1M) SARM
- fscanf MG A: 16
- fsck MG A: 7
- fsck(1M) SAG 5: 38, 44–118; SARM
 - bfs file system SAG 5: 118
 - s5 file system SAG 5: 46–77
 - ufs file system SAG 5: 77–118
- fsdb(1M) SARM
- fseek MG A: 16
- fseek(3S) PRM
- fsetpos(3C) PRM
- FSFLUSHR parameter SAG 8: 72
- fsrand(1) BSD; BSD/XNX 1: 22
- fspec(4) SARM
- fstat MG A: 16
- fstyp(1M) SAG 5: 30; SARM
- FSTypes (file system types) SAG 5: 3
- fstypes(4) NUAG; SARM
 - changing defaults NUAG 8: 3
- fsync(2) PRM; SS 7: 3
- ftell MG A: 16
- ftime MG A: 16
- ftime(2) XNX
- ftime(3) BSD/XNX 1: 28
- ftime(3C) BSD
- ftok MG A: 16
- ftp(1) NUAG 1: 8, 25: 1; URM
- ftpd(1M) NUAG; SARM
- ftw MG A: 17
- ftw(3C) PRM
- full duplex UG 2: 2, 8, GL: 6

- full pathname (see pathname)
 fumount(1M) NUAG; SARM
 function calls
 data input and output ISDG 3: 9
 syntax ISDG 3: 9–11
 function declaration CGP 3: 28–29
 function definition CGP 3: 29–30
 function keys
 FMLI (see screen-labeled function keys)
 problems SAG C: 14
 sysadm(1M) alternatives to SAG C: 26
 use with sysadm(1M) SAG C: 10
 function keys (FACE) UG 5: 5
 alternatives to UG 5: 9
 in forms UG 5: 19
 in menus UG 5: 11
 in text frames UG 5: 35
 function naming conventions ISDG 3: 22
 function prototypes CGP 2: 36, 3: 25, 28–30; CTG2–6
 lint(1) checks for CGP 5: 3, 10
 Function summary SCSI 3: 1
 functions with varying argument lists CTG7–9
 fusage(1M) NUAG; SARM
 fuser(1M) SARM
 fwrite MG A: 17
 fwtmp(1M) SARM
 fxlist MG A: 17
- G**
- gamma MG A: 17
 gamma(3M) PRM
 gateway NUAG 4: 2–3
 gcd(3) BSD/XNX 1: 26
 gconvert(3) BSD/XNX 1: 28
 gcore(1) URM
 gcvt MG A: 17
 gencat(1) URM
 gencc(1M) SARM
 general system data structures ISDG 3: 16
 generic pointer CGP 3: 28
 get MG A: 12
 get(1) CGP 10: 3–6, 12–23; PRM
 getc MG A: 17
 getc(3S) CGP 2: 52; PRM
 getch MG A: 17
 getch(3X) CHAR 7: 15
 getchar MG A: 17
 getcontext(2) PRM
 getcwd MG A: 17
 getcwd(3C) PRM
 getdate(3C) PRM
 getdents(2) PRM
 getdev(1M) SAG 15: 36–38; SARM
 getdgrp(1M) SAG 15: 42–45; SARM
 getdtablesize(3) BSD; BSD/XNX 1: 32
 getegid MG A: 17
 getenv MG A: 17
 getenv(3C) PRM
 geterror(D3DK) DDRM
 geteuid MG A: 17
 getfrm(1F) CHAR 2: 40, , C: 1
 getgid MG A: 17
 getgrent MG A: 17
 getgrent(3C) PRM
 getgrgid MG A: 17
 getgrnam MG A: 17
 getgroups(2) PRM
 gethostbyaddr function NI 3: 24

gethostbyname function NI 3: 24
gethostent(3N) NI
gethostid(3) BSD; BSD/XNX 1: 32
gethostname(3) BSD; BSD/XNX 1: 32
gethz MG A: 17
getititems(1F) CHAR 2: 39,, C: 1
getitimer(3C) PRM
getlogin MG A: 17
getlogin(3C) PRM
getmajor(D3DK) DDRM
getminor(D3DK) DDRM
getmntent(3C) PRM
getmsg(2) PRM; STRM 5: 10
getnetconfig(3N) NI
getnetent(3N) NI
getnetpath(3N) NI
getopt MG A: 7, 17
getopt(1) URM
getopt(3C) CGP 2: 53, 55; PRM
getopts(1) URM
getpagesize(3) BSD
getpass MG A: 17
getpass(3C) PRM
getpeername(3N) NI
getpgrp MG A: 17
getpid MG A: 17
getpid(2) PRM
getpmsg function STRM 5: 12
getppid MG A: 17
getpriority(3) BSD
getprotoent(3N) NI
getpw MG A: 17
getpw(3C) PRM
getpwent MG A: 17
getpwent(3C) PRM
getpwnam MG A: 17
getpwuid MG A: 17
getq(D3DK) DDRM
getrbuf(D3DK) DDRM
getrlimit(2) PRM
getrusage(3) BSD; BSD/XNX 1: 32
gets MG A: 12, 17
gets(3S) PRM
getservent(3N) NI
getsid(2) PRM
getsockname(3N) NI
getsockopt(3N) NI
getspent(3C) PRM
getstr MG A: 17
getstr(3X) CHAR 7: 18
getsubopt(3C) PRM
gettable(1M) NUAG; SARM
gettimeofday(3) BSD; BSD/XNX 1: 32
gettimeofday(3C) PRM
gettmode MG A: 17
gettxt(1) URM
gettxt(3C) PRM
getty(1M) SAG 13: 1, 31; SARM
getuid MG A: 17
getuid(2) PRM
getusershell(3) BSD; BSD/XNX 1: 27
getut(3C) PRM
getutent MG A: 17
getutid MG A: 17
getutline MG A: 17
getutx(3C) PRM
GETVER SCSI 2: 4
getvfsent(3C) PRM
getvol(1M) SARM
getw MG A: 18
getwd(3) BSD; BSD/XNX 1: 29
getyx MG A: 18
global programs (FACE), administration of UG 5: 86-90
global symbols CGP 2: 29, 13: 26-27
global variables CHAR 2: 10

glyphs, table of CHAR 2: 48
 gmatch(3G) PRM
 gmtime MG A: 18
 goto command
 FACE UG C: 14
 FMLI CHAR 2: 33
 goto statement CGP 3: 50
 GPGSLO parameter SAG 8: 73
 Grades file (BNU) SAG 7: 59–61
 grantpt(3C) STRM 12: 25
 with pseudo-tty driver STRM 12: 22
 graphic characters CHAR 2: 47
 graphic toolkits, changed for Release
 4.0 MG 5: 39
 graphical user interfaces, changed for
 Release 4.0 MG 5: 39
 grep MG A: 7
 grep(1) UG 3: 57, 60–61, 9: 10–12, 27;
 URM; XNX
 group
 adding a SAG 17: 15
 ID SAG 17: 7; UG 3: 57
 identification bits SAG 12: 25–28
 list information SAG 17: 20
 permissions UG 3: 50
 renaming a SAG 17: 15
 Group 10 CDB SCSI 4: 11
 Group 6 CDB SCSI 4: 11
 group(4) SARM
 groupadd(1M) SARM
 groupdel(1M) SARM
 groupmod(1M) SARM
 groups(1) BSD; BSD/XNX 1: 25; UG
 3: 57; URM
 grpcheck MG A: 7
 grpck(1M) BSD; BSD/XNX 1: 22
 gsignal MG A: 18

H

HA SCSI 4: 15
 half bright attribute CHAR 2: 47, 3: 9
 half duplex UG 2: 2, GL: 6
 halt(1M) BSD; BSD/XNX 1: 23
 halt(D2D) DDRM
 haltsys MG A: 7
 hard disk
 display information about SAG
 15: 17–18
 format a SAG 15: 15–17
 partition SAG 15: 11
 SCSI (see SCSI hard disk)
 hard disk drive, remove SAG
 15: 25–30
 hard link SS 6: 1, 7
 hard mounts NUAG 17: 5
 hardware emulation module STRM
 12: 13–14
 HAS_COLORS variable CHAR 2: 11
 has_colors(3X) CHAR 7: 30
 hat_getkpfnum(D3K) DDRM
 HA_VER SCSI 2: 4
 HCPULIM parameter SAG 8: 74
 hcreate MG A: 18
 hd MG A: 7
 hd(1) XNX
 hd(7) SARM
 HDATLIM parameter SAG 8: 75
 hdeadd(1M) SAG 4: 37
 hdefix(1M) SAG 4: 37
 hdelogger(1M) SAG 4: 40
 hdestroy MG A: 18
 hdr MG A: 12
 head MG A: 8
 head(1) URM
 header descriptor CHAR 2: 23, 3: 58

header files CGP 2: 33-36
 how to include CGP 2: 8-9, 3: 14-15
 lint(1)ing CGP 5: 8-9
 standard place CGP 2: 8-9
help MG A: 12
help command
 FACE UG 5: 32-35, C: 3, 15, 23
 FMLI CHAR 1: 24, 2: 33, 3: 6
 FMLI, example of CHAR 3: 67-71
help descriptor CHAR 2: 18, 20, 23, 29,
 3: 6, 35, 59, 4: 17
help messages
 setting up in a FACE object SS
 9: 17
 title hierarchy SS 9: 16
 writing SS 9: 12
help(1) CGP 10: 6, 12, 31; PRM
hexadecimal escape CGP 3: 8
HFNOLIM parameter SAG 8: 75
HFSZLIM parameter SAG 8: 74
hierarchical mounts NUAG 19: 8
highlight_bar descriptor CHAR
 2: 28, 4: 7, A: 11
highlight_bar_text descriptor
 CHAR 2: 28, 4: 8, A: 11
highlighting CHAR 3: 9
 automatic mechanisms CHAR 1: 18
 terminal attribute CHAR 2: 47
/home, disk partitions SAG A: 3
home directory UG 1: 11, 3: 4-6
HOME environment variable SAG
 17: 26; UG 4: 8, 5: 41
hooks ISDG 3: 16
Host adapter SCSI 1: 3, 7
Host adapter release number SCSI
 2: 4, 4: 15
Host adapter version number SCSI
 2: 4, 4: 15

host computer SCSI 4: 15
host names NI 3: 24
host number NUAG 2: 3
hostid(1) BSD; BSD/XNX 1: 23
hostname(1) BSD; BSD/XNX 1: 23
hosts(4) NI; NUAG 2: 15-16; SARM
hosts.equiv(4) NUAG 3: 3; SARM
 use of + in NUAG 3: 3
HRTIME parameter SAG 8: 78
HRVTIME parameter SAG 8: 78
hsearch MG A: 18
hsearch(3C) PRM
HSTKLIM parameter SAG 8: 75
htable(1M) NUAG; SARM
hunt sequence, set up SAG 13: 3
HVMMLIM parameter SAG 8: 75
HW_PROVIDER parameter SAG 8: 71
HW_SERIAL parameter SAG 8: 71
hypot MG A: 18
hypot(3M) PRM

I

ICMP (Internet Control Message Pro-
 tocol) NUAG 1: 3, 7, 6: 1
ICMP(7) NI; NUAG; SARM
icon UG 5: 12, 14
iconv(1) URM
iconv(5) SARM
ics_agent_cmp(D3D) DDRM
ics_find_rec(D3D) DDRM
ics_hostid(D3D) DDRM
ics_rdwr(D3D) DDRM
ics_read(D3D) DDRM
ics_write(D3D) DDRM
ID ISDG 3: 1, 21
id MG A: 8

ID

- implementation ISDG 3: 34
- interactions ISDG 3: 40
- modifications ISDG 3: 41
- overview ISDG 3: 35
- ID directories
 - bin ISDG 3: 42
 - cf.d ISDG 3: 42
 - init.d ISDG 3: 44
 - mfsys.d ISDG 3: 44
 - node.d ISDG 3: 43
 - pack.d ISDG 3: 43
 - rc.d ISDG 3: 43
 - sd.d ISDG 3: 43
 - sdevice.d ISDG 3: 43
 - sfsys.d ISDG 3: 44
- id(1M) SARM; UG 3: 57; URM
- Idbuild command ISDG 3: 46
- idbuild(1M) SARM
- Idcheck command ISDG 3: 46
- idcheck(1M) SARM
- idconfig(1M) SARM
- #ident CGP 3: 18
- identifiers CGP 3: 5
- idepend installation parameter SAG 14: 15
- Idinstall command ISDG 3: 46
- idinstall(1M) SARM
- idload(1M) NUAG; SARM
- idmkinit(1M) SARM
- idmknod(1M) SARM
- idmkunix(1M) SARM
- idspace(1M) SARM
- idtune(1M) SARM
- ieee_funcs(3) BSD/XNX 1: 29
- ieee_functions(3M) BSD
- ieee_handler(3) BSD/XNX 1: 29
- ieee_handler(3M) BSD
- #if CGP 2: 2, 3: 15-16
- if statement CGP 3: 47
- if(7) NUAG
- ifconfig(1M) NUAG; SARM
- #ifdef CGP 3: 15-16
- ifiles CGP B: 1
- #ifndef CGP 3: 15-16
- if-then-else statement (FMLI) CHAR 2: 42-43, A: 16
- i-list SAG 5: 9
- imacct MG A: 28
- imprint MG A: 28
- inactive descriptor CHAR 2: 19, 21, 3: 9, 39, A: 13
- inactive_border descriptor CHAR 2: 28, 4: 8
- inactive_title_bar descriptor CHAR 2: 28, 4: 8
- inactive_title_text descriptor CHAR 2: 28, 4: 8
- inb(D3D) DDRM
- incfile backup method SAG 3: 19-27
- incfile(1M) SARM
- inch MG A: 18
- #include CGP 2: 2, 8-9, 33-35, 3: 14-15
- include files ISDG 3: 14, 16
- incoming events NI 2: 64-65
- incomplete types CGP 3: 28, 30; CTG40-43
- index(3) BSD; BSD/XNX 1: 29
- indicator(1F) CHAR 2: 40, 4: 6, , C: 1
- indxbib(1) BSD; BSD/XNX 1: 16
- inet(3N) NI
- inet(7) NUAG; SARM
- inetd(1M) NI 3: 30, 57, 6: 12, 7: 55-56; NUAG 1: 9; SARM
- port monitor changed for Release 4.0 MG 5: 34

- inetd.conf(4) NUAG; SARM
- inet_ntoa function NI 3: 24
- infocmp(1M) CHAR 13: 19; SARM
- init descriptor CHAR 2: 18, 20, 23,
3: 6, 36, 59, A: 17
- init file ISDG 3: 52
- init state (see system state)
- init(1M) SAG 13: 6; SARM
 - scheduler properties SS 5: 29
- INITCLASS parameter SAG 8: 77
- init_color(3X) CHAR 7: 35
- init.d directory SAG 6: 25
- init(D2D) DDRM
- initdefault SAG 6: 11, 43
- initgroups(3C) PRM
- initialization CGP 3: 44-46; SAG 6: 17
- initialization file (FMLI) CHAR 4: 2-16,
21, A: 11
 - graphic characters in CHAR 2: 47
 - lists of descriptors CHAR 2: 25-29
 - order of descriptors CHAR 4: 2
 - overview CHAR 1: 10
- init_pair(3X) CHAR 7: 33
- initscr MG A: 18
- initscr(3X) CHAR 6: 3, 8
- initstate(3) BSD/XNX 1: 27
- inittab(4) SAG 6: 11, 43, 13: 6, 35, 42;
SARM
- inl(D3D) DDRM
- inode SAG 5: 9, 8: 5; SS 6: 1
 - maximum number SAG 5: 26
- inode(4) SARM
- input redirection UG 9: 14-26
- input/output CGP 2: 38-39, 50-55
 - changed for Release 4.0 MG 5: 21
- input/output polling STRM 6: 1-7
- insch MG A: 18
- insertln MG A: 18
- insq(D3DK) DDRM
- insque(3C) PRM
- install(1) BSD; BSD/XNX 1: 23
- install(1M) PRM; SARM
- installable driver ISDG 3: 21
 - implementation ISDG 3: 34
 - overview ISDG 3: 35
- installation
 - admin file SAG 14: 3, 9, 12-17
 - attribute information SAG 14: 4
 - check SAG 14: 27-28
 - content information SAG 14: 4
 - default parameters SAG 14: 9
 - display information about SAG
14: 29-33
 - errors SAG 14: 23
 - from a remote machine SAG
14: 25-26
 - full SAG 14: 4, 22
 - interactive SAG 14: 7, 9, 20-21
 - non-interactive SAG 14: 7, 9, 18-19,
21-22
 - parameters ISDG 2: 20-21; SAG 14: 3,
12-16; SS 8: 21-22
 - partial SAG 14: 4, 22
 - response file SAG 14: 4
 - software database SAG 14: 4, 8
 - suggestions for SAG 14: 9-11
 - summary of commands SAG 14: 37
 - tools ISDG 2: 10; SS 8: 9
 - troubleshooting SAG 14: 22
- installation scripts (package) ISDG
2: 18-31; SS 8: 19-33
 - class action script ISDG 2: 23-30; SS
8: 25-31
 - exit codes ISDG 2: 21; SS 8: 22
 - parameters ISDG 2: 20; SS 8: 21
 - procedure script ISDG 2: 30-31; SS
8: 32-33

- processing ISDG 2: 19; SS 8: 20
- request script ISDG 2: 22–23; SS 8: 23–25
- installf(1M) ISDG 2: 30; SS 8: 32
- installing new kernel ISDG 3: 58
- instance installation parameter SAG 14: 14
- int CGP 3: 21, 31–33
- integral constants CGP 3: 6–7
 - promotion of CTG13
- integral types
 - conversion CGP 3: 31–33
 - declaration CGP 3: 21
 - initialization CGP 3: 44
- interface modifications
 - naming SS 9: 9
 - naming requirements SS 9: 10
 - planning SS 9: 6
 - planning the location of SS 9: 6
 - planning the structure SS 9: 8
 - writing SS 9: 11
- internal device number STRM 9: 5–6
- internationalization CTG24–27, 32–35
 - changed for Release 4.0 MG 5: 41
- Internet NUAG 1: 3
 - Domain Stream Connection NI 3: 10
 - support for protocols NI 12: 13
- Internet Protocol (see IP)
- internetwork NUAG 4: 4
- interrupt
 - handler ISDG 3: 20
 - priority level ISDG 3: 27
 - processing ISDG 3: 7
 - vector number ISDG 3: 36
- interrupt descriptor CHAR 4: 9, 15, 17
 - description CHAR 3: 6, 9, 36, 59
 - example of CHAR 3: 20
- table entry CHAR 2: 18–20, 22–24, 27–29
- Interrupt routine SCSI 1: 7
- interrupt signal handling (FMLI) CHAR 2: 44–46, A: 18
- interrupt-driven socket I/O NI 3: 44
- interrupts
 - in console driver STRM G: 12
 - in ports driver STRM G: 8
- intr(D2D) DDRM
- intro(1) PRM; URM
- intro(1M) SARM
- intro(2) PRM; XNX
- intro(3) PRM
- intro(3M) PRM
- intro(4) PRM; SARM
- intro(5) PRM; SARM
- intro(7) SARM
- introductory frame (FMLI)
 - descriptor definitions CHAR 4: 3–4
 - example of CHAR 4: 4
 - list of descriptors CHAR 2: 26
- i-number SAG 5: 9
- invalidmsg descriptor CHAR 2: 21, 3: 39
- inw(D3D) DDRM
- I/O addresses ISDG 3: 37
- I/O multiplexing NI 3: 17
- I/O requests, asynchronous
 - notification NI 3: 45
- IOA ISDG 3: 37
- iocblk structure STRM A: 9
 - with M_IOCTL STRM B: 3
- ioctl ISDG 3: 67; MG A: 18
- ioctl routine B_GETDEV and B_GETTYPE code example SCSI 1: 7
- ioctl1(2) PRM

- console driver write-side STRM G: 13
- general processing STRM 7: 10-12
- handled by ports driver STRM G: 6
- handled by ptem(7) STRM 12: 17
- hardware emulation module STRM 12: 13
- I_ATMARK STRM 5: 19
- I_CANPUT STRM 5: 18
- I_CKBAND STRM 5: 18
- I_GETBAND STRM 5: 18
- I_LINK STRM 10: 6, B: 3
- I_LIST STRM 7: 29
- I_PLINK STRM 10: 32, B: 3
- I_POP STRM 3: 10
- I_PUNLINK STRM 10: 32, B: 3
- I_PUSH STRM 3: 10
- I_RECVFD STRM 11: 9, B: 6
- I_SENDFD STRM 11: 9, B: 6
- I_SETSIG events STRM 6: 6
- I_STR STRM 3: 14, B: 3
- I_STR processing STRM 7: 12-14
- I_UNLINK STRM 10: 11, B: 3
- supported by ldterm(7) STRM 12: 7
- supported by master driver STRM 12: 24
- termio(7) handled by xt driver STRM G: 30
- transparent STRM 7: 14-29
- xt driver windowing STRM G: 30
- ioctl(2) system call SCSI 2: 1
- ioctl(D2) routine SCSI 1: 6
- ioctl(D2DK) DDRM
- iovec(D4DK) DDRM
- IP (Internet Protocol) NUAG 1: 7, 2: 1
 - address creating NUAG 2: 4
 - address representation NUAG 2: 3
 - header NUAG 1: 7
 - localhost address NUAG 2: 15
 - Number Registration Form NUAG A: 1-6
 - IP(7) NI; NUAG; SARM
 - ipbs MG A: 28
 - IPC (interprocess communication) SS 2: 20-22, 4: 1-87
 - XENIX compatibility BSD/XNX 1: 6
 - IPC (sockets)
 - basics NI 3: 2
 - Internet domain datagrams NI 3: 15
 - Internet Domain Stream Connection NI 3: 12
 - multiplexing NI 3: 17
 - select(3C) NI 3: 20
 - socket naming NI 3: 2
 - UNIX domain NI 3: 2
 - ipcrm MG A: 8
 - ipcrm(1) URM
 - ipcs MG A: 8
 - ipcs(1) URM
 - IPL ISDG 3: 27
 - ipr MG A: 28
 - iprint MG A: 28
 - ips MG A: 28
 - isalnum MG A: 18
 - isalpha MG A: 18
 - isascii MG A: 18
 - isastream(3C) STRM 11: 9
 - isatty MG A: 18
 - ISCB_TYPE SCSI 4: 3
 - iscntrl MG A: 18
 - isdigit MG A: 18
 - isencrypt(3G) PRM
 - isgraph MG A: 18
 - islower MG A: 18
 - ismpx(1) URM
 - isnan function BSD/XNX 1: 29

isnan(3C) PRM
 isprint MG A: 18
 ispunct MG A: 18
 isspace MG A: 18
 issue(4) SARM
 isupper MG A: 18
 isxdigit MG A: 18
 item help file SS 9: 12
 item_count(3X) CHAR 10: 24
 item_description(3X) CHAR 10: 13
 item_index(3X) CHAR 10: 60
 item_init(3X) CHAR 10: 56
 itemmsg descriptor CHAR 2: 19, 3: 10
 example of CHAR 3: 24
 item_name(3X) CHAR 10: 13
 item_opts(3X) CHAR 10: 14
 item_opts_off(3X) CHAR 10: 15
 item_opts_on(3X) CHAR 10: 15
 item_term(3X) CHAR 10: 56
 item_userptr(3X) CHAR 10: 18
 item_value(3X) CHAR 10: 11
 item_visible(3X) CHAR 10: 16
 itom(3) BSD/XNX 1: 26
 IVN ISDG 3: 36

J

j0 MG A: 18
 j1 MG A: 18
 jagent(5) PRM; SARM
 jn MG A: 18
 job control CHAR 1: 24, A: 10, 19; STRM
 6: 9–12
 changed for Release 4.0 MG 5: 27
 terminology STRM 6: 9–10
 job number NUAG 24: 3, 6
 join MG A: 8

join(1) URM
 jrand48 MG A: 18
 jterm(1) URM
 jwin(1) URM

K

kenter driver routine ISDG 3: 19
 kernel UG 1: 1, 4
 kernel preemption point SS 5: 33
 kernel print information, retrieving
 ISDG 3: 67–68
 kernel print statements ISDG 3: 67
 kernel profiling SAG 8: 12–15
 kernel timers ISDG 3: 6
 kernel tunable parameters SAG
 8: 70–71
 kernel-mode scheduler parameter
 table SAG 10: 13–14
 kexit driver routine ISDG 3: 20
 keyboard layout UG 2: 3
 keyboard(7) SARM
 keylogin(1) NI; NUAG
 keys (see encryption keys)
 keyserf(1M) NI; NUAG 18: 8
 keystrokes, alternate CHAR 1: 5, 17,
 B: 1–6; UG 5: 9
 keywords CGP 3: 6
 kill MG A: 8, 18
 kill(1) SAG 8: 8; UG 9: 34–35; URM
 kill(2) PRM
 killall(1M) SARM
 killpg(3) BSD; BSD/XNX 1: 32
 kmem_alloc(D3DK) DDRM
 kmem_free(D3DK) DDRM
 kmem_zalloc(D3DK) DDRM
 Korn shell (see shell and ksh(1))

ksh(1) URM
 changed for Release 4.0 MG 5: 6
kvtophys(D3D) DDRM

L

l MG A: 8
l3tol MG A: 18
l3tol(3C) PRM
l64a MG A: 18
labelit(1M) SAG 5: 35; SARM
LANG Environment Variable ISDG
 F: 2
langinfo(5) PRM; SARM
languages SS 1: 4-8 (see also C
 language)
last(1) URM
lastcomm(1) BSD; BSD/XNX 1: 24
lastlogin system log SAG 12: 17
latencies, software SS 5: 33-34
layers(1) CHAR A: 19; UG G: 5-9; URM
 startup file UG G: 6
layers(5) PRM; SARM
lazy symbol binding CGP 13: 66
lc MG A: 8
lcong48 MG A: 18
ld MG A: 12
ld(1) BSD; BSD/XNX 1: 19; CGP 2: 6;
 PRM
LD_BIND_NOW CGP 2: 21, 13: 53
ldd(1) CGP 2: 21; PRM
ldexp MG A: 19
LD_LIBRARY_PATH CGP 2: 18-20
LD_RUN_PATH CGP 2: 19-20
ldsysdump(1M) SARM
ldterm(7) SAG 13: 35; STRM 12: 3
ldterm_mod structure STRM 12: 4

leaveok MG A: 19
lex MG A: 12
lex(1) CGP 1: 10, 11: 1-27; PRM; SS 1: 6
 command line CGP 11: 2-3
 definitions CGP 11: 17-20, 25
 disambiguating rules CGP 11: 13
 how to write source CGP 11: 5-21
 library CGP 11: 2-3, 24
 operators CGP 11: 6-9
 quick reference CGP 11: 26-27
 routines CGP 11: 10, 14-16
 source format CGP 11: 5, 26-27
 start conditions CGP 11: 19-20
 use with yacc(1) CGP 11: 17, 22-24,
 12: 1-4, 9-11, 31-32
 user routines CGP 11: 14-15, 20-21
 yylex() CGP 11: 2-3, 22
lexical analyzer (see lex(1))
lfind MG A: 19
liber ISDG A: 1; SS A: 1
libraries CGP 2: 2-3, 10-55
 archive CGP 2: 2-3, 12-32
 BSD Compatibility Package
 BSD/XNX 1: 26-31
 creating CGP 2: 14-16, 23-29
 libc CGP 2: 11-14, 38-44, 50-55
 libdl CGP 2: 12, 14, 21
 libelf CGP 13: 1
 libgen CGP 2: 48-50
 libm CGP 2: 13-14, 44-47
 linking with CGP 2: 10-32
 lint(1) CGP 5: 9-10
 maintaining CGP 9: 14-16
 naming conventions CGP 2: 31
 shared object CGP 2: 2-3, 11-32,
 13: 30-38, 50-69
 sockets NI 3: 23
 standard place CGP 2: 12-14

- libwindows(3X) PRM
- lifetime descriptor CHAR 2: 18, 20, 23, 3: 6, 36, 59, A: 17
- LIFO, module add/remove STRM 3: 13
- Limits file (BNU) SAG 7: 58
 - setup SAG 7: 24
- limits(4) PRM; SARM
- #line CGP 3: 17
- line MG A: 8
- line discipline module
 - close STRM 12: 5
 - description STRM 12: 3-12
 - in job control STRM 6: 11
 - in pseudo-tty subsystem STRM 12: 15
 - ioctl(2) STRM 12: 7
 - open STRM 12: 5
- line editor (see ed(1))
- line(1) URM
- line-drawing glyphs, table of CHAR 2: 48
- lininfo descriptor CHAR 2: 19, 21, 3: 10, 39
- LININFO variable CHAR 2: 11
- link MG A: 19
- link count SS 6: 1
- link editing CGP 2: 2-3, 6-7, 10-32, 13: 24-38, 50-69
 - dynamic CGP 2: 2-3, 10-32, 13: 30-38, 50-69
 - library linking options CGP 2: 12-20, 31-32
 - multiply defined symbols CGP 2: 29-30, 13: 26-27
 - quick reference CGP 2: 31-32
 - static CGP 2: 2-3, 10-32
 - undefined symbols CGP 2: 10
- link(1M) SARM
- link(2) PRM
- linkb(D3DK) DDRM
- linkblk structure STRM A: 12
- link_field(3X) CHAR 11: 9
- link_fieldtype(3X) CHAR 11: 92
- links
 - create ISDG 2: 42; SS 8: 47
 - symbolic (see symbolic links)
- lint MG A: 12
- lint(1) CGP 1: 8-9, 5: 1-50; PRM
 - command line CGP 5: 8-10
 - consistency checks CGP 5: 3
 - filters CGP 5: 10-11
 - libraries CGP 5: 9-10
 - message formats CGP 5: 2
 - messages CGP 5: 17-50
 - options and directives CGP 5: 1, 11-16
 - portability checks CGP 5: 4-6
 - suspicious constructs CGP 5: 6-7
- listdgrp(1M) SAG 15: 45; SARM
- listen NI 13: 1
- listen(1M) NI 6: 12, 7: 55, 57; SAG 13: 3, 11, 17, 55-65; SARM
 - add port monitor SAG 13: 60, 65
 - add service SAG 13: 61, 65
 - administrative command SAG 13: 57-58
 - and Service Access Facility SAG 13: 57-58
 - configuration files SAG 13: 63
 - disable service SAG 13: 62
 - disable services SAG 13: 61, 65
 - dynamic addressing SAG 13: 56
 - enable services SAG 13: 61, 65
 - log file SAG 13: 63
 - passing connections to standing servers SAG 13: 56

port monitor status SAG 13: 58
 private addresses SAG 13: 55
 remove port monitor SAG 13: 60, 65
 remove service SAG 13: 61, 65
 RPC-based services SAG 13: 56
 service status SAG 13: 59–60
 socket-based services SAG 13: 56
 status SAG 13: 65
 listen(3N) NI 3: 7
 listusers(1) URM
 ln MG A: 8
 ln(1) BSD; BSD/XNX 1: 17; URM
 lo(7) NUAG; SARM
 LOADPFK variable CHAR 2: 12
 local names NI 3: 4
 local symbols CGP 13: 26–27
 local transport interface management
 NI 2: 8–10, 16–23, 40–42
 locale CTG32, 34
 changed functions CTG33
 new functions CTG34
 localeconv(3C) PRM
 localhost address NUAG 2: 13
 localtime MG A: 19
 lock MG A: 19
 lock manager, NFS NUAG 20: 1–6
 lock(2) XNX
 lockd(1M) NUAG
 lockd(3N) NUAG 20: 2
 lockf MG A: 19
 lockf(3C) PRM; SS 3: 6, 8–9, 11, 14–16
 locking MG A: 19; SS 3: 1, 5
 advisory SS 3: 2, 18
 file and record SS 3: 1–19
 mandatory SS 3: 3, 17–18
 permissions SS 3: 4
 read SS 3: 2, 4, 9
 record SS 3: 6, 9
 write SS 3: 2, 4, 9
 locking(2) XNX
 log MG A: 19
 log driver tunable parameters SAG
 8: 79
 log10 MG A: 19
 log(7) STRM
 logger(1) BSD; BSD/XNX 1: 23
 logging in UG 2: 11–17
 problems UG 2: 15–17
 logging off UG 2: 18
 logical blocks SAG 5: 26
 choosing size SAG 5: 31
 login
 directory UG 3: 4–6
 ID sharing UG 5: 2
 name SAG 17: 5; UG 2: 10
 login service SAG 13: 31, 33
 login(1) URM
 login(4) SARM
 loginlog system log SAG 12: 16–17
 loginlog(4) SARM
 logins SAG 17: 5
 adding SAG 16: 15
 administrative SAG 12: 18, 16: 16
 assigning SAG 17: 5
 dial-up password SAG 12: 9–13
 displaying information SAG 12: 15
 locking SAG 12: 13
 system SAG 12: 19, 16: 18–19
 logins(1M) SAG 12: 15; SARM
 logname MG A: 8, 19
 LOGNAME environment variable SAG
 17: 27; UG 4: 8
 logname(1) URM
 logs
 backup history SAG 3: 53–57
 bad block SAG 4: 34

- cron SAG B: 36
- disk report SAG 4: 40
- lastlogin SAG 12: 17
- login SAG B: 35
- loginlog SAG 12: 16
- mail SAG F: 6
- SMTP (see mail(1))
- spelling SAG B: 34
- su SAG B: 35
- sulog SAG 12: 3, 17
- system SAG 1: 13
- system logins SAG B: 34
- uucp (see BNU logs)
- long double CGP 3: 21, 32
- long int CGP 3: 21, 32-33
- longjmp MG A: 19
- longjmp function BSD/XNX 1: 33
- longjmp() function ISDG 3: 28
- longline(1F) CHAR 2: 40, C: 1
- longname MG A: 19
- look(1) BSD; BSD/XNX 1: 14
- lookbib(1) BSD; BSD/XNX 1: 16
- loopback transport NI 5: 8
- lorder MG A: 12
- lorder(1) CGP 1: 12; PRM
- losing data SAG 4: 42
- lower multiplexor STRM 2: 17
- lower Stream STRM 2: 15
- lp MG A: 8
- lp login SAG 16: 19
- LP print service
 - administration overview SAG 9: 1-8
 - administration summary SAG 9: 135-136
 - banner SAG 9: 40-41
 - character sets SAG 9: 28-31
 - configure SAG 9: 12-50
 - configuring SAG 9: 6-7
 - control access to SAG 9: 39-40
 - customizing SAG 9: 120-134
 - default destination SAG 9: 45
 - disable printer SAG 9: 10, 51-53
 - distributed SAG 9: 6
 - enable printer SAG 9: 10, 51-53
 - fault detection SAG 9: 35-39
 - files SAG 9: 100-109
 - filters SAG 9: 4, 72-88, 133-134
 - forms SAG 9: 4
 - install SAG 9: 9-11
 - interface program SAG 9: 21-22, 127-133
 - load management SAG 9: 89-97
 - network configuration SAG 9: 7
 - PostScript printers SAG 9: 110-119
 - print style SAG 9: 42-43
 - print wheels SAG 9: 28-33, 45-47
 - printable file types SAG 9: 23-25
 - printer classes SAG 9: 43-44, 47
 - printer configuration SAG 9: 49-50
 - printer descriptions SAG 9: 41
 - printer ports SAG 9: 25-28
 - printer types SAG 9: 22-23
 - queue priorities SAG 9: 93-97
 - remote SAG 9: 20-21
 - server configuration SAG 9: 6
 - shell interface SAG 9: 1-2
 - start SAG 9: 98-99
 - stop SAG 9: 98-99
 - sysadm(1M) interface SAG 9: 1-2
 - terminfo(4) SAG 9: 124-127
 - troubleshooting SAG 9: 54-62
 - using forms with SAG 9: 34-35, 45-47, 63-71
- lp(1) UG 8: 1-3, 7, 10-11, 14-19, 25-27;
URM
 - default values for UG 8: 2

login SAG 12: 19
lp(7) SARM
lpadmin MG A: 8
lpadmin(1M) SARM
lpc(1M) BSD; BSD/XNX 1: 20
lpfilter(1M) SARM
lpforms(1M) SARM
lpinit MG A: 8
lpmove MG A: 8
lpq(1) BSD; BSD/XNX 1: 20
lpr MG A: 8
lpr(1) BSD; BSD/XNX 1: 21
lprm(1) BSD; BSD/XNX 1: 21
lprof(1) CGP 1: 9, 2: 9-10, 7: 1-31;
PRM
cc(1) profiling option CGP 7: 4
command line CGP 7: 10-19
default output CGP 7: 12-13
environment variable CGP 7: 5-9
highlighting unexecuted lines CGP
7: 14-15
merged output CGP 7: 7, 17-19
summary output CGP 7: 16
usage examples CGP 7: 22-31
use with prof(1) CGP 7: 1, 22-28
lpsched MG A: 8
lpsched(1M) SARM
lpshut MG A: 8
lpstat MG A: 8
lpstat(1) UG 8: 7-10, 25, 28-29; URM
lpssystem(1M) SARM
lptest(1) BSD; BSD/XNX 1: 21
lpusers(1M) SARM
lrand48 MG A: 19
ls MG A: 8
ls(1) BSD; BSD/XNX 1: 17; SS 3: 17; UG
3: 17-23; URM; XNX
lsearch MG A: 19

lsearch(3C) PRM
lseek MG A: 19
lseek(2) PRM; SS 3: 8
ltol3 MG A: 19
lvalues CGP 3: 33

M

m4 MG A: 12
m4(1) CGP 15: 1-14; PRM; SS 1: 7
argument handling CGP 15: 7-9
arithmetic capabilities CGP 15: 9-10
command line CGP 15: 2-3
conditional preprocessing CGP
15: 12
defining macros CGP 15: 4-7
file manipulation CGP 15: 10-11
quoting CGP 15: 5-7
string handling CGP 15: 12-14
mach(1) BSD; BSD/XNX 1: 23
machid(1) URM
machine management SAG 6: 1-5
command summary SAG 6: 49
sysadm(1M) interface SAG 6: 1-3
macro expansion CGP 3: 13-14; CTG17
macro packages, BSD Compatibility
Package BSD/XNX 1: 15
madd(3) BSD/XNX 1: 26
mail MG A: 8
mail (FACE) UG 5: 89, C: 19-20
mail installation parameter SAG
14: 13
mail(1) SS 3: 4; UG 11: 3-14; URM
address style translation SAG F: 1
administration SAG F: 1-10
alias lists SAG F: 1, 5
bang addressing SAG F: 1

- command summary UG 11: 14
- controlling access SAG F: 7
- delete message UG 11: 13
- Deny command SAG F: 7
- domain addressing SAG F: 1-3
- files SAG F: 1, 3
- gateway machine SAG F: 1, 3
- incoming UG 11: 11-14
- LAN SAG F: 1, 3-5
- logging SAG F: 6
- machine cluster SAG F: 1, 3
- NFS SAG F: 1, 3-5
- quit UG 11: 13
- reading UG 11: 11-14
- RFS SAG F: 1, 3-5
- routing SAG F: 6
- save message UG 11: 13
- sending UG 11: 3-11
- sending to remote systems UG 11: 7
- smart host SAG F: 1
- SMTP (Simple Mail Transfer Protocol) SAG F: 1, 8-10
- undeliverable UG 11: 4
- mailalias(1) SAG F: 1; URM
- MAILCHECK variable CHAR 2: 12, A: 9
- mailcnfg(4) SAG F: 1; SARM
- maillock(3X) PRM
- mail_pipe(1M) SARM
- .mailrc UG 11: 38-41
- mailstats(1M) BSD; BSD/XNX 1: 21
- mailsurr(4) SAG F: 1; SARM
- mailx(1) UG 11: 15-37, C: 19-21; URM
 - adding your signature UG 11: 26
 - changing message header UG 11: 25
 - deleting mail UG 11: 34
 - edit message UG 11: 20-21
 - end message UG 11: 20
 - incoming UG 11: 30-37
 - incorporating existing text UG 11: 22
 - incorporating message from mailbox UG 11: 24
 - msglist argument UG 11: 30
 - options UG 11: 18, 20
 - other mail files UG 11: 33
 - quit UG 11: 28, 36
 - read file UG 11: 23
 - reading UG 11: 30-37
 - reading mail UG 11: 31-32
 - record of messages UG 11: 26-28
 - replying to mail UG 11: 35-36
 - saving mail UG 11: 35
 - sending UG 11: 19-29
- main function CGP 1: 5, 3: 29
- major device number STRM 9: 5
- major numbers ISDG 3: 12
- make MG A: 12
- make(1) CGP 1: 9-10, 9: 1-29; PRM
 - command line CGP 9: 20-21
 - environment variables CGP 9: 21-22
 - how to write source CGP 9: 2-11
 - macros CGP 9: 4-11, 13, 15-16
 - maintaining libraries CGP 9: 14-16
 - makefile convention CGP 9: 1
 - sample output CGP 9: 5-6
 - source format CGP 9: 8
 - suffix transformation rules CGP 9: 11-14, 24-29
 - usage example CGP 9: 4-6
 - use with SCCS CGP 9: 16-18
- makecontext(3C) PRM
- makedbm(1M) NI 11: 24-27
- makedev(3C) PRM
- makedevice(D3DK) DDRM
- makefsys(1M) SARM
 - login SAG 12: 18
- makekey(1) URM

- mallinfo MG A: 19
- malloc MG A: 19
- malloc(3C) PRM
- malloc(3X) PRM
- mallopt MG A: 19
- man(1) BSD; BSD/XNX 1: 14
- man(7) BSD; BSD/XNX 1: 15
- mandatory locking SS 3: 3, 17
- map IDs
 - add passwd and group files NUAG 12: 22
 - complex NUAG 12: 12–23
 - example rules files NUAG 12: 56–58
 - general NUAG 10: 10–11
 - gid.rules NUAG 12: 21–22
 - how it works NUAG 12: 50–51
 - idload(1M) NUAG 12: 13–16
 - idload(1M) and rules files NUAG 12: 52–56
 - list current map NUAG 12: 60–61
 - multi groups NUAG 12: 51–52
 - remote names NUAG 12: 58–59
 - remote users NUAG 12: 50–61
 - run idload NUAG 12: 22–23
 - uid.rules NUAG 12: 17–21
 - when not to NUAG 12: 12
 - when to NUAG 12: 12–13
- map(D4DK) DDRM
- mapfiles CGP B: 1–18
 - defaults CGP B: 11
 - error messages CGP B: 17
 - example CGP B: 9
 - map structure CGP B: 13
 - mapping directives CGP B: 6
 - segment declarations CGP B: 4
 - size-symbol declarations CGP B: 8
 - structure CGP B: 3
 - syntax CGP B: 3
 - usage CGP B: 2
- mapped files SS 7: 4–9
 - changed for Release 4.0 MG 5: 23
 - private SS 7: 5
 - shared SS 7: 5
- maps, for the automounter NUAG 19: 3
- maps (YP) NI 11: 2, 6, 8–16, 23–30
 - adding to makefile NI 11: 29–30
 - changing server NI 11: 32–34
 - creating NI 11: 24–25
 - makefiles NI 11: 12–15
 - propagating NI 11: 27–29
 - updating NI 11: 23–27
- mark command (FMLI) CHAR 2: 34
- mask, file creation SAG 17: 28
- masm MG A: 12
- master driver
 - in pseudo-tty subsystem STRM 12: 15
 - open STRM 12: 22
- master file ISDG 3: 14, 41
- master file ISDG 3: 49
- master files SAG 8: 52
- master server (YP) NI 11: 3–6
 - map changing NI 11: 32–34
 - set with ypinit NI 11: 15–16
 - start daemons NI 11: 17
- math library
 - linking with CGP 2: 13–14
 - partial contents CGP 2: 44–47
- math(5) PRM
- matherr MG A: 19
- matherr(3M) PRM
- MAXCLSYSPRI parameter SAG 8: 77
- max(D3DK) DDRM
- MAX_INPUT STRM G: 9
- MAXPMEM parameter SAG 8: 73

- MAXUP** parameter SAG 8: 69
mbchar(3C) PRM
mboot program SAG 6: 6
M_BREAK STRM B: 2
mbstring(3C) PRM
mcmp(3) BSD/XNX 1: 26
M_COPYIN STRM B: 14
 transparent ioctl example STRM
 7: 18–21
M_COPYOUT STRM B: 15
 transparent ioctl example STRM
 7: 22–24
 with **M_IOCTL** STRM B: 6
mcs(1) CGP 1: 12; PRM
M_CTL STRM B: 2
 with line discipline module STRM
 12: 3
mctl(3) BSD
M_DATA STRM 2: 10, B: 2
 xt driver write-side processing
 STRM G: 29
M_DELAY STRM B: 2
mdevice file ISDG 3: 14, 41, 49
mdevice(4) SARM
mdiv(3) BSD/XNX 1: 26
me(7) BSD; BSD/XNX 1: 15
mem(7) SARM
memccpy MG A: 19
memchr MG A: 19
memcmp MG A: 19
memcntl(2) PRM; SS 7: 10–11
memcpy MG A: 19
memory
 fault CGP 6: 1
 shared (see shared memory)
memory management SS 2: 25–26,
 7: 1–16
 address spaces SS 7: 1
 address-space layout SS 7: 15
 changed for Release 4.0 MG 5: 23
 coherence SS 7: 2
 concepts SS 7: 1
 heterogeneity SS 7: 2
 mapping SS 7: 1
 memcntl(2) SS 7: 10–11
 mincore(2) SS 7: 10
 mlock(3C) SS 7: 11
 mlockall(3C) SS 7: 12
 mmap(2) SS 7: 4–9
 MMU changed for Release 4.0 MG
 5: 24
 mprotect(2) SS 7: 13–14
 msync(3C) SS 7: 12–13
 munlock(3C) SS 7: 11
 munlockall(3C) SS 7: 12
 munmap(2) SS 7: 9
 networking SS 7: 2
 pagesize SS 7: 13
 system calls SS 7: 4
 virtual memory SS 7: 1
memory mapped I/O ISDG 3: 9
memory(3C) PRM
memory-mapped files (see mapped
 files)
memset MG A: 19
menu descriptor CHAR 2: 18, 3: 7
menu_attributes(3X) CHAR
menu_back(3X) CHAR 10: 38
menu_cursor(3X) CHAR
menu_driver(3X) CHAR 10: 44
menu_fore(3X) CHAR 10: 38
menu_format(3X) CHAR 10: 27
menu_grey(3X) CHAR 10: 38
menu_hook(3X) CHAR
menu_init(3X) CHAR 10: 56
menu_item_current(3X) CHAR

menu_item_name(3X) CHAR
menu_item_new(3X) CHAR
menu_item_opts(3X) CHAR
menu_items(3X) CHAR 10: 23
menu_item_userptr(3X) CHAR
menu_item_value(3X) CHAR
menu_item_visible(3X) CHAR
menu_mark(3X) CHAR 10: 31
menu_new(3X) CHAR
menuonly descriptor CHAR 2: 21, 3: 39
menu_opts(3X) CHAR 10: 70
menu_opts_off(3X) CHAR 10: 72
menu_opts_on(3X) CHAR 10: 72
menu_pad(3X) CHAR 10: 38
menu_pattern(3X) CHAR 10: 65
menu_post(3X) CHAR
menus (ETI) CHAR 10: 1-72
 application-defined commands
 CHAR 10: 50
 call menu driver CHAR 10: 51
 change and fetch pattern buffer
 CHAR 10: 64
 change default item attributes
 CHAR 10: 16
 change default menu attributes
 CHAR 10: 25
 check item visibility CHAR 10: 16
 compile and link programs CHAR
 10: 2
 count menu items CHAR 10: 24
 create and free CHAR 10: 21-22
 create and free items CHAR 10: 8-10
 definition of kinds CHAR 10: 11
 dimensions CHAR 10: 26-33
 directional item navigation
 requests CHAR 10: 47
 display CHAR 10: 26-43
 driver processing CHAR 10: 44-66
 ETI menu requests CHAR 10: 47
 fetch and change current item
 CHAR 10: 59
 fetch and change display attributes
 CHAR 10: 37
 fetch and change menu items CHAR
 10: 23
 fetch and change top row CHAR
 10: 62
 fetch names and descriptions CHAR
 10: 13
 format CHAR 10: 27
 item and menu initialization CHAR
 10: 56
 item and menu termination CHAR
 10: 56
 item navigation requests CHAR
 10: 47
 item select value CHAR 10: 11
 key virtualization correspondence
 CHAR 10: 45
 manipulate item attributes CHAR
 10: 13-17
 manipulate menu attributes CHAR
 10: 23-25
 manipulate menu user pointer
 CHAR 10: 67-69
 mark string CHAR 10: 31
 multi-valued CHAR 10: 11
 multi-valued menu selection
 request CHAR 10: 48
 pattern buffer requests CHAR 10: 49
 position cursor CHAR 10: 63
 post and unpost CHAR 10: 40
 sample program CHAR 10: 4
 scrolling requests CHAR 10: 48
 set and fetch menu options CHAR
 10: 70-72

- set item options CHAR 10: 13
- set item user pointer CHAR 10: 18
- single-valued CHAR 10: 11
- terminology summary CHAR 10: 3
- what a menu program does CHAR 10: 4
- windows and subwindows CHAR 10: 34
- write programs using CHAR 10: 3-7
- menus (FACE)
 - choices UG 5: 24
 - function keys UG 5: 11
 - navigation UG 5: 12
 - scrollable UG 5: 14
 - selecting items UG 5: 16
- menus (FMLI)
 - creating a dynamic CHAR 3: 28
 - definition CHAR 1: 7-8
 - descriptor definitions CHAR 3: 2-11
 - examples of CHAR 3: 12-31
 - lists of descriptors CHAR 2: 17-20
 - marking items CHAR 1: 8
 - multi-column CHAR 3: 5
 - multi-select CHAR 1: 8, 3: 5, 7, 9, 11
 - navigation in CHAR 1: 18-19
 - order of descriptors CHAR 3: 2
 - passing parameters CHAR 3: 28
 - positioning CHAR 3: 4
 - scrollable CHAR 1: 7
 - selecting an item CHAR 1: 19-20
 - single-select CHAR 1: 8, 3: 9
- menus (sysadm)
 - changing entries SS 9: 23
 - creating entries SS 9: 22
 - definition form SS 9: 25
 - deleting entries SS 9: 33
 - information file SS 9: 4
 - item help message description SS 9: 13
 - item help message example SS 9: 13
 - item help message format SS 9: 13
 - locating entries SS 9: 9
 - NFS interface NUAG D: 1-5
 - testing changes SS 9: 25
- menus(3X) CHAR
 - menu_sub(3X) CHAR 10: 34
 - menu_term(3X) CHAR 10: 56
 - menu_userptr(3X) CHAR 10: 67
 - menu_win(3X) CHAR 10: 34
- M_ERROR STRM B: 15
- mesg MG A: 8
- mesg(1) SAG 17: 32; URM
- message block (STREAMS) STRM 2: 4
 - linkage STRM 5: 6
 - structure STRM 5: 4
- message boundaries NI 2: 35, 4: 3
- message (IPC) SS 4: 3-31
 - blocking SS 4: 3
 - control (msgctl) SS 4: 14-15
 - get (msgget) SS 4: 8-11
 - identifier (msgid) SS 4: 4-7
 - msgctl example program SS 4: 15-20
 - msgget example program SS 4: 11-14
 - msgop example program SS 4: 23-31
 - operations (msgop) SS 4: 21-23
 - permission codes SS 4: 9-10
 - queue data structure SS 4: 4-7
 - receive SS 4: 22-23
 - send SS 4: 21-22
 - usage SS 4: 4-7
- message line
 - FACE UG 5: 5
 - FMLI CHAR 1: 4
 - FMLI, duration of display CHAR 3: 5, 4: 11

- message of the day SAG 1: 13, 6: 4
- message processing routines
 - (STREAMS) STRM 4: 1-3
 - design guidelines STRM 7: 44-46
- message queue (STREAMS), priority STRM 5: 15-19
- message (STREAMS) STRM 2: 9
 - allocation STRM 5: 54
 - control information STRM 2: 10, 5: 47
 - definition STRM 2: 3
 - freeing STRM 5: 54
 - handled by `pckt(7)` STRM 12: 21
 - handled by `psem(7)` STRM 12: 18
 - high priority STRM 5: 2, B: 14-21
 - `ldterm(7)` read-side STRM 12: 5
 - `ldterm(7)` write-side STRM 12: 7
 - linkage STRM 5: 6
 - `M_DATA` STRM 2: 10
 - `M_PCPROTO` STRM 2: 10
 - `M_PROTO` STRM 2: 10
 - ordinary STRM 5: 1, B: 2-13
 - processing STRM 5: 26
 - recovering from allocation failure STRM 5: 57
 - sending/receiving STRM 5: 8
 - service interface STRM 5: 34-47
 - structures STRM 5: 4-6, A: 6-8
 - types STRM 2: 10, 5: 1
- message tunable parameters SAG 8: 81-82
- `message(1F)` CHAR 2: 40, , C: 1
- metacharacters UG GL: 7
 - `sh(1)` UG 9: 4-9
- `M_FLUSH` STRM B: 16
 - flags STRM B: 16
 - in module example STRM 8: 8
 - packet mode STRM 12: 21
 - xt driver write-side processing STRM G: 29
- `mfree(3)` BSD/XNX 1: 26
- `mfsys(4)` SARM
- `M_HANGUP` STRM B: 17
- `migration(1M)` SARM
- `min(3)` BSD/XNX 1: 26
- `MINARMEM` parameter SAG 8: 73
- `MINASMEM` parameter SAG 8: 73
- `mincore(2)` PRM; SS 7: 10
- `min(D3DK)` DDRM
- `mini_system` SAG 6: 8
- minor device number STRM 9: 5
- minor numbers ISDG 3: 12
- `M_IOCACK` STRM B: 17
 - with `M_COPYOUT` STRM B: 15
 - with `M_IOCTL` STRM B: 4
- `M_IOCDATA` STRM B: 18
 - xt driver write-side processing STRM G: 29
- `M_IOCNAK` STRM B: 19
 - with `M_COPYOUT` STRM B: 15
 - with `M_IOCTL` STRM B: 4
- `M_IOCTL` STRM B: 3-6
 - transparent STRM B: 4
 - with `M_COPYOUT` STRM B: 15
 - xt driver write-side processing STRM G: 29
- `mkdev` MG A: 8
- `mkdir` MG A: 8
- `mkdir(1)` UG 3: 15-16; URM
- `mkdir(2)` PRM
- `mkdirp(3G)` PRM
- `mkfifo(1M)` SARM
- `mkfifo(3C)` PRM
- `mkfs` MG A: 8
- `mkfs(1M)` SAG 5: 30-36, 6: 9; SARM
 - with the `s5-FSType` SAG 5: 31

- with the ufs-FSType SAG 5: 34
- mkinitab MG A: 8
- mkmsgs(1) URM
- mknod MG A: 8, 19
- mknod(1M) SARM
 - define device files SAG 15: 14–15
- mknod(2) PRM
- mknod(2) SCSI 2: 6
- mknod(2) XNX
- mkpart(1M) SARM
- mkstemp(3) BSD; BSD/XNX 1: 29
- mkstr MG A: 12
- mktemp MG A: 19
- mktemp(3C) PRM
- mktemp(3C) SCSI 2: 6
- mktime(3C) PRM
- mkuser MG A: 8
- mlock(3C) PRM; SS 7: 11
- mlockall(3C) PRM; SS 7: 12
- mmap(2) PRM; SS 7: 4–9
- mmap(D2K) DDRM
- mnttab(4) SARM
- mode (file) SAG 12: 21 (see also permissions)
- modem UG GL: 8
 - interrupts SAG 8: 51
- modf MG A: 19
- modifying group attributes SAG 17: 19
- module_info(D4DK) DDRM
- monitor MG A: 19
- monitor(3C) PRM
- monitoring NUAG 12: 66–78
 - client caching NUAG 12: 71–74
 - CPU time NUAG 12: 70–71
 - remote disk space NUAG 12: 78
 - remote system calls NUAG 12: 66–68
 - resource usage NUAG 12: 76–78
- rfs operations NUAG 12: 68–69
 - server processes NUAG 12: 74–76
- montbl(1M) SARM
- more MG A: 8
- more(1) URM
- MORECTL STRM 5: 47
- MOREDATA STRM 5: 47
- motd (message-of-the-day) file SAG 1: 13, 6: 4
- mount MG A: 8, 19
 - mount point SAG 5: 37
 - definition NUAG 7: 3
 - special NUAG 19: 2
- mount(1M) NUAG; SAG 5: 37–38; SARM
 - debugging NFS problems NUAG 17: 6–8
 - display mounted remote resources NUAG 9: 19, 15: 19
 - mount remote resources NUAG 9: 10, 15: 10
- mount(2) PRM
- mountall(1M) SARM
 - mount a set of remote resources NUAG 9: 16, 15: 16
- mountd(1M) NUAG 17: 2–4, 7; SARM
- mountfsys(1M) SARM
 - login SAG 12: 18
- mouse CHAR A: 10
- mouse(7) MOUSE
- mouseadmin(1) MOUSE
- mout(3) BSD/XNX 1: 26
- move MG A: 19
- move command (FACE) UG 5: 46, 55, C: 8
- move(3X) CHAR 6: 4, 7: 8
- move_field(3X) CHAR 11: 18
- mp(3X) BSD
- M_PASSFP STRM B: 6

Master Subject Index

M_PCPROTO STRM 2: 10, B: 19
M_PCRSE STRM B: 20
M_PCSIG STRM B: 20
mp.h header file BSD/XNX 1: 34
mprotect(2) PRM; SS 7: 13-14
M_PROTO STRM 2: 10, B: 7-8
mps_AMPcancel(D3D) DDRM
mps_AMPreceive(D3D) DDRM
mps_AMPreceive_frag(D3D) DDRM
mps_AMPsend(D3D) DDRM
mps_AMPsend_data(D3D) DDRM
mps_AMPsend_reply(D3D) DDRM
mps_AMPsend_rsvp(D3D) DDRM
mps_close_chan(D3D) DDRM
mps_free_dmabuf(D3D) DDRM
mps_free_msgbuf(D3D) DDRM
mps_free_tid(D3D) DDRM
mps_get_dmabuf(D3D) DDRM
mps_get_msgbuf(D3D) DDRM
mps_get_reply_len(D3D) DDRM
mps_get_soldata(D3D) DDRM
mps_get_tid(D3D) DDRM
mps_get_unsoldata(D3D) DDRM
mps_mk_bgrant(D3D) DDRM
mps_mk_brdcst(D3D) DDRM
mps_mk_breject(D3D) DDRM
mps_mk_sol(D3D) DDRM
mps_mk_solrply(D3D) DDRM
mps_mk_unsol(D3D) DDRM
mps_mk_unsolrply(D3D) DDRM
mps_msg(D3D) DDRM
mps_open_chan(D3D) DDRM
mrand48 MG A: 19
M_READ STRM B: 20
M_RSE STRM B: 8
ms(7) BSD; BSD/XNX 1: 15
M_SETOPTS STRM B: 8-13
 SO_FLAG STRM B: 9-13
 SO_READOPT options STRM 5: 13
 SO_WROFF value STRM 5: 14
 with ldterm(7) STRM 12: 5
msgb(D4DK) DDRM
msgctl MG A: 19
msgctl(2) PRM; SS 4: 14-20
 example program SS 4: 15-20
 usage SS 4: 14-15
msgdsize(D3DK) DDRM
msgget MG A: 20
msgget(2) PRM; SS 4: 8-14
 example program SS 4: 11-14
 usage SS 4: 8-11
msglist (see mailx(1))
MSGMAP parameter SAG 8: 81
MSGMAX parameter SAG 8: 81
MSGMNB parameter SAG 8: 81
MSGMNI parameter SAG 8: 81
MSG_OOB NI 3: 41
msgop(2) PRM; SS 4: 21-31
 example program SS 4: 23-31
 usage SS 4: 21-23
MSG_PEEK NI 3: 41
msgrcv MG A: 20
msgrcv(2) SS 4: 22-23
MSGSEG parameter SAG 8: 82
msgsnd MG A: 20
msgsnd(2) SS 4: 21-22
MSGSSZ parameter SAG 8: 82
MSGTQL parameter SAG 8: 82
M_SIG STRM B: 13
 in signaling STRM 6: 7
msqrt(3) BSD/XNX 1: 26
M_START STRM B: 20
M_STARTI STRM B: 21
M_STOP STRM B: 20
M_STOPI STRM B: 21
msub(3) BSD/XNX 1: 26

- msync(3C) PRM; SS 7: 12-13
 - mt(1) BSD; BSD/XNX 1: 23
 - mtape MG A: 28
 - mtx(3) BSD/XNX 1: 26
 - mtune file ISDG 3: 41
 - mtune(4) SARM
 - mult(3) BSD/XNX 1: 26
 - MULTIBUS II functions DDRM 3: 7-9
 - multibyte characters CGP 3: 9; CTG24-27
 - multi-host functionality SCSI 4: 16
 - multiplexing NI 3: 17
 - in xt driver STRM G: 32
 - STREAMS STRM 2: 15-19
 - multiplexor
 - building STRM 10: 2-10
 - controlling Stream STRM 10: 8
 - data routing STRM 10: 12
 - declarations STRM 10: 20
 - design guidelines STRM 10: 37
 - dismantling STRM 10: 11
 - driver STRM 10: 19-31
 - example STRM 10: 16-18
 - lower STRM 10: 1
 - lower connection STRM 10: 13-15
 - lower disconnection STRM 10: 15
 - lower read put procedure STRM 10: 28-31
 - lower write service procedure STRM 10: 28
 - persistent links STRM 10: 32-36
 - upper STRM 10: 1
 - upper write put procedure STRM 10: 23-26
 - upper write service procedure STRM 10: 27
 - multiplexor ID
 - in multiplexor building STRM 10: 6
 - in multiplexor dismantling STRM 10: 11
 - multiply defined symbols CGP 2: 29-30, 13: 26-27
 - multiselect descriptor CHAR 2: 18, 3: 7
 - multi-user state SAG 6: 15, 22, 16: 5
 - munlock(3C) SS 7: 11
 - munlockall(3C) SS 7: 12
 - munmap(2) PRM; SS 7: 9
 - mv MG A: 8
 - mv(1) UG 3: 28, 43-45; URM
 - mvaddch MG A: 20
 - mvaddstr MG A: 20
 - mvcur MG A: 20
 - mvdelch MG A: 20
 - mmdir MG A: 8
 - mmdir(1M) SARM
 - mvgetch MG A: 20
 - mvgetstr MG A: 20
 - mvinch MG A: 20
 - mvinsch MG A: 20
 - mvwaddch MG A: 20
 - mvwaddstr MG A: 20
 - mvwdelch MG A: 20
 - mvwgetch MG A: 20
 - mvwgetstr MG A: 20
 - mvwin MG A: 20
 - mvwinch MG A: 20
 - mvwinsch MG A: 20
- N
- name binding NI 3: 4
 - name descriptor CHAR 2: 19, 21-22, 24, 29-30, 3: 10, 40, 4: 15, 18
 - named(1M) NUAG 5: 36

- named keys (FACE)
 - (see function keys (FACE) also)
 - alternatives to UG 5: 9
 - list of UG 5: 10
- named keys (FMLI) CHAR 1: 17, B: 1-6
 - problems CHAR A: 8
- named pipe (see FIFO)
- named Stream
 - description STRM 11: 7-9
 - fattach(3C) STRM 11: 7
 - fdetach(3C) STRM 11: 8
 - file descriptor passing STRM 11: 9
 - isastream(3C) STRM 11: 9
 - remote STRM 11: 10
- named(1M) NUAG 5: 1; SARM
- namefiles mail file SAG F: 1
- names
 - host NI 3: 24
 - network NI 3: 24
 - protocol NI 3: 25
- names mail file SAG F: 1
- name-to-address mapping NI 4: 9,
12: 13-21; SAG 7: 11-14
 - changed for Release 4.0 MG 5: 37
 - routines NI 12: 16-21
- name-to-address translation NI 5: 8, 10
- naming conventions ISDG 3: 17
- nap MG A: 20
- nap(2) XNX
- NAUTOPUSH parameter SAG 8: 79
- NAUTOUP parameter SAG 8: 72
- navigation
 - FACE UG 5: 12, 20, 25-27, C: 14
 - FMLI CHAR 1: 17-22, B: 1-6
- naawk(1) URM
- NBPW parameter SAG 8: 82
- NBUF parameter SAG 8: 71
- nbwaitsem MG A: 20
- NCALL parameter SAG 8: 69
- ncheck MG A: 8
- ncheck(1M) SAG 12: 26-28; SARM
- NCLIST parameter SAG 8: 69
- ncol descriptor CHAR 2: 21, 3: 40
- ndbm(3) BSD
- ndbm.h header file BSD/XNX 1: 34
- NDQUOT parameter SAG 8: 80
- neqm(1) BSD/XNX 1: 15
- net number NUAG 2: 3
- netconfig(4) NI 5: 7-9, 12: 2-10, 12;
SAG 7: 5-12; SARM
- netdir(3N) NI
- netdir_free function NI 12: 17
- netdir_getbyaddr function NI 5: 10,
12: 17
- netdir_getbyname function NI 5: 10,
12: 17
- netdir_options function NI 12: 18
- netdir_perror function NI 12: 19
- netdir_serror function NI 12: 19
- NETPATH environment variable NI
5: 7-9, 7: 3, 11, 17, 20, 22, 12: 2, 4-7,
12; SAG 7: 9
- netpath transport type NI 5: 9
- netrc(4) NUAG; SARM
- netstat(1M) NUAG 6: 3-7; SARM
- nettype, and rpcgen(1) NI 6: 10
- netutil MG A: 8
- network
 - classifications NUAG 2: 2
 - configuration NI 3: 51
 - configuration database file SAG
7: 5-12
 - configuration file NI 12: 2-10, 12
 - daemons in.named NUAG 5: 1
 - daemons in.named NUAG 5: 1 (see
also daemons (network))

- definition NUAG 1: 3
- files NUAG 2: 14
- hardware NUAG 4: 2
- identifier NI 5: 8
- masks NUAG 4: 14
- names NI 3: 24
- protocol NUAG 1: 3
- selection NI 5: 6
- type NI 5: 9
- type flag NI 5: 8
- network addressing NI 12: 13-21; SAG 7: 11-14
 - string address providers NI 12: 14; SAG 7: 13
 - TCP/IP NI 12: 14
 - TCP/IP SAG 7: 12
- network administration (see BNU)
- Network File System (see NFS)
- network listener NUAG 10: 7
 - set up NUAG 12: 2-3
- network lock manager NUAG 20: 1-6
- network problems, diagnosing NUAG 6: 8
- network security NUAG 3: 1
 - effect of administrative files NUAG 3: 2
 - NFS NUAG 18: 1-10
 - .rhosts NUAG 3: 2
 - Secure NFS NUAG 18: 7
- network selection NI 4: 5, 12: 1-15; SAG 7: 3-14
 - changed for Release 4.0 MG 5: 36
 - code examples NI 12: 10-12
 - default search path SAG 7: 9-10
 - directory lookup libraries SAG 7: 8-9
 - netconfig(4) SAG 7: 5-12
 - NETPATH environment variable SAG 7: 9
 - protocol family identifiers SAG 7: 6-8
 - protocol name identifiers SAG 7: 8
 - sysadm(1M) menu SAG 7: 3
- network services
 - shell command list SAG 7: 2
 - sysadm(1M) menu SAG 7: 2
- networking
 - basic (see BNU)
 - changed for Release 4.0 MG 5: 32-37
- networking utilities package (see BNU)
- networks(4) NUAG 2: 16; SARM
- new kernel
 - executing ISDG 3: 58
 - installing ISDG 3: 58
 - reconfiguring ISDG 3: 58
- newaliases(1M) BSD; BSD/XNX 1: 21
- newboot(1M) SAG 6: 8
- new_field(3X) CHAR 11: 9
- new_fieldtype(3X) CHAR 11: 94
- newform MG A: 9
- newform(1) URM
- new_form(3X) CHAR 11: 43
- newfs(1M) BSD
- newgrp MG A: 9
- newgrp(1M) SARM; URM
- new_item(3X) CHAR 10: 8
- newkey(1M) NI; NUAG YP NI 11: 9
- new_menu(3X) CHAR 10: 21
- new_page(3X) CHAR 11: 44
- news MG A: 9
- news(1) SAG 6: 4, 17: 31; UG 5: 91; URM
- newwin MG A: 20
- newwin(3X) CHAR 8: 8
- next_choice(3X) CHAR 11: 102
- next_frm command (FMLI) CHAR 2: 34, A: 12

nextkey MG A: 20
nextkey(3) BSD/XNX 1: 26
nextpage command (FMLI) CHAR
2: 34, A: 12
NFS
booting and setuid problems NUAG
18: 9
BSD Compatibility Package com-
mands BSD/XNX 1: 22
changed for Release 4.0 MG 5: 35
error messages NUAG 17: 7-8
menu interface NUAG D: 1-5
mounting process NUAG 17: 2-3
overview of administration NUAG
13: 6
security NUAG 18: 1-10
starting NUAG 14: 3
NFS lock manager NUAG 20: 1-6
NFS resources
defined NUAG 13: 3
granting root access NUAG 14: 5-6
hard mounted NUAG 17: 5
mount automatically NUAG 14: 7
soft mounted NUAG 17: 5
unsharing NUAG 9: 5, 8, 15: 5, 8
unsharing, definition NUAG 7: 4
nfsd(1M) NUAG 17: 2, 9; SARM
NGROUPS_MAX parameter SAG 8: 76
NHBUF parameter SAG 8: 71
nice MG A: 9, 20
nice(1) SS 5: 28; URM
nice(2) PRM; SS 5: 28
nice(3) BSD/XNX 1: 30
nice(3C) BSD
NINODE parameter SAG 8: 80
nl MG A: 9, 20
nl(1) URM
nlist MG A: 20
nlist(3) BSD; BSD/XNX 1: 30
nlist(3E) PRM
nl_langinfo(3C) PRM
NLOG parameter SAG 8: 79
nlsadmin(1M) SAG 13: 11, 17, 57-58,
61-62, 65; SARM
nlsgetcall(3N) NI
nlsprovider(3N) NI
nlsrequest(3N) NI
nl_types(5) PRM; SARM
nm MG A: 12
nm(1) CGP 1: 12, 2: 27; PRM
nobang descriptor CHAR 2: 27, 4: 10
nocbreak(3X) CHAR 7: 42
nocrmode MG A: 20
node file ISDG 3: 51
node name NUAG 12: 1; SAG 16: 20-21
NODE parameter SAG 8: 70
noecho MG A: 20
noecho descriptor CHAR 2: 21, 3: 40
noecho(3X) CHAR 7: 41
noenable(D3DK) DDRM
nohup MG A: 9
nohup(1) UG 9: 35-36; URM
non-blocking sockets NI 3: 43
nonl MG A: 20
nop command (FMLI) CHAR 2: 34
noraw MG A: 20
notational conventions ISDG 1: 4
NOTICE error messages SAG E: 2-5
notify(1) UG 11: 14; URM
NPBUF parameter SAG 8: 72
NPROC parameter SAG 8: 69
NR variable CHAR 2: 12
nrand48 MG A: 20
nroff(1) BSD; BSD/XNX 1: 15
nrow descriptor CHAR 2: 21, 3: 40
nslookup(1M) NUAG; SARM

nsquery(1M) NUAG; SARM
 NSTRPHASH parameter SAG 8: 79
 NSTRPUSH parameter SAG 8: 76; STRM
 3: 10, E: 5
 null(7) SARM
 nuucp login SAG 12: 19, 16: 19

O

object files CGP 1: 2, 13: 1-69
 base address CGP 13: 42
 data representation CGP 13: 3
 ELF header CGP 13: 4-11
 global offset table CGP 13: 60-61
 hash table CGP 13: 67-68
 lazy symbol binding CGP 13: 66
 libelf CGP 13: 1
 note section CGP 13: 45-47
 procedure linkage table CGP
 13: 62-66
 program header CGP 13: 39-47
 program interpreter CGP 13: 51-53
 program linking CGP 13: 4-38
 program loading CGP 13: 47-69
 relocation CGP 13: 30-38, 60
 section alignment CGP 13: 15
 section attributes CGP 13: 18-23
 section header CGP 13: 12-23
 section names CGP 13: 22-23
 section types CGP 13: 15-23
 segment contents CGP 13: 44-45
 segment permissions CGP 13: 43
 segment types CGP 13: 40-42
 string table CGP 13: 23-24
 symbol table CGP 13: 24-29
 tools for manipulating CGP 1: 12,
 13: 1

octal escape CGP 3: 8
 od MG A: 9
 od(1) URM
 Office Functions (FACE) UG 5: 68
 offsetof(3C) PRM
 oldipr MG A: 28
 O_NDELAY
 close a Stream STRM 3: 11
 with M_SETOPTS STRM B: 11
 oninterrupt descriptor CHAR 4: 10,
 15, 18
 description CHAR 3: 7, 10, 37, 60
 example of CHAR 3: 20
 table entry CHAR 2: 18-19, 21-22, 24,
 27, 29-30
 O_NONBLOCK
 close a Stream STRM 3: 11
 with M_SETOPTS STRM B: 11
 open MG A: 20
 open command (FMLI) CHAR 2: 34,
 A: 12
 OPEN LOOK SS 1: 11-12
 changed for Release 4.0 MG 5: 40
 open(2) PRM
 open(2) system call SCSI 2: 4
 open(D2DK) DDRM
 opendir MG A: 20
 opensem MG A: 20
 opensem(2) XNX
 operating mode (see system state)
 operating system release number
 SAG 6: 40
 operators (C language) CGP 3: 34-43
 additive CGP 3: 37-38
 assignment CGP 3: 42
 associativity and precedence CGP
 3: 43
 bitwise CGP 3: 38-40

cast CGP 3: 36
comma CGP 3: 42
conditional CGP 3: 41
equality CGP 3: 39
logical CGP 3: 40-41
multiplicative CGP 3: 36-37
preprocessing CGP 3: 12-13
relational CGP 3: 39
structure CGP 3: 42
unary CGP 3: 34-36
optimizer CGP 2: 10
 use with `sdb(1)` CGP 6: 3
option negotiation NI 2: 40
orderly release NI 2: 13, 33, 37, 70-71
ordinary file UG 3: 2
organize command (FACE) UG 5: 50,
 C: 9
OSI (Open Systems Interconnection)
 NI 2: 1-4
 Reference Model NI 2: 2-4
other, permissions UG 3: 50
OTHERQ(D3DK) DDRM
ott(4) SARM
 files (FACE) UG 5: 84
out of band data NI 3: 41
outb(D3D) DDRM
outgoing events NI 2: 62-63
out1(D3D) DDRM
output redirection UG 9: 14-26
outw(D3D) DDRM
overlay MG A: 21
overwrite MG A: 21

P

p2open(3G) PRM
pack MG A: 9

pack(1) URM
package
 access in scripts ISDG 2: 35; SS 8: 38
 administration SS 9: 11
 assign abbreviation ISDG 2: 33; SS
 8: 36
 basic steps to ISDG 2: 32-33; SS
 8: 34-36
 contents ISDG 2: 2; SS 8: 2-4
 copyright message ISDG 2: 39; SS
 8: 44
 create ISDG 2: 46-48; SS 8: 54, 56
 create datastream formats ISDG
 2: 48; SS 8: 56
 define dependencies ISDG 2: 39; SS
 8: 43
 description file SS 9: 4
 distribute over multiple volumes
 ISDG 2: 46; SS 8: 53
 identifier ISDG 2: 33; SAG 14: 4-5; SS
 8: 37
 information files ISDG 2: 11-17; SS
 8: 10-18
 installation ISDG 2: 10; SS 8: 9
 installation changed for Release 4.0
 MG 5: 30
 installation scripts ISDG 2: 18-31; SS
 8: 19-33
 instance ISDG 2: 33; SAG 14: 4-6; SS
 8: 37
 life cycle ISDG 2: 3; SS 8: 5
 location ISDG 2: 36; SS 8: 40
 modification file SS 9: 3
 objects SS 8: 3
 quick reference SS 8: 58-60
 relocatable SAG 14: 5
 relocatable objects ISDG 2: 36-38;
 SAG 14: 6-7; SS 8: 40-41

- remove SAG 14: 36
- space requirements ISDG 2: 40; SS 8: 45
- store SAG 14: 34
- tools ISDG 2: 4-9; SS 8: 6-8
- packet NUAG 1: 6, 4: 2
- packet mode
 - description STRM 12: 21
 - messages STRM 12: 21
- pads CHAR 6: 12
- page descriptor CHAR 2: 21, 3: 40
- page through file UG 3: 32-37
- page_numtopp(D3DK) DDRM
- page_pptonum(D3DK) DDRM
- pagesize(1) BSD; BSD/XNX 1: 23
- PAGES_UNLOCK parameter SAG 8: 73
- paging CGP 2: 23, 26-28, 13: 47-50
- paging tunable parameters SAG 8: 72-73
- pair_content(3X) CHAR 7: 31
- PAIR_NUMBER CHAR 7: 30
- panel_above(3X) CHAR
- panel_move(3X) CHAR
- panel_new(3X) CHAR
- panels CHAR 9: 1-18
 - change CHAR 9: 5
 - check if hidden CHAR 9: 11
 - compiling and linking programs CHAR 9: 2
 - create CHAR 9: 3
 - delete CHAR 9: 18
 - elementary operations CHAR 9: 4-6
 - fetch above or below CHAR 9: 13-14
 - fetch pointers CHAR 9: 4
 - hide CHAR 9: 10
 - make invisible CHAR 9: 10-12
 - move CHAR 9: 6
 - move to top or bottom CHAR 9: 7
 - reinstate CHAR 9: 11
 - setting and fetching pointer CHAR 9: 15-18
 - update CHAR 9: 8-9
- panels(3X) CHAR
- panel_show(3X) CHAR
- panel_top(3X) CHAR
- panel_update(3X) CHAR
- panel_userptr(3X) CHAR
- panel_window(3X) CHAR
- panic SAG 4: 37-40, E: 12-19
- panic() call ISDG 3: 65
- PANIC error messages SAG E: 12-19
- panic() function ISDG 3: 68
- parameter tuning (see tunable parameters)
- parent directory UG 3: 2, 10
- parity UG 2: 3, GL: 9
- parser (see yacc(1))
- parsing SS 1: 7
- partial installation parameter SAG 14: 14
- partitions
 - boot SAG 6: 5-12
 - cartridge tape SAG A: 9
 - default SAG 6: 5
 - disk SAG A: 5-8
 - diskette SAG A: 10
 - stand SAG 6: 5-12
- passmgmt(1M) SARM
- Pass-through interface SCSI 4: 11
- pass-through interface SCSI 4: 16
- Pass-through interface caution SCSI 2: 9
- passwd MGA: 9
- passwd(1) SAG 16: 2, 10, 16-19, 22; URM
- passwd(4) SARM
- password SAG 12: 3, 17: 5; UG 2: 12, GL: 9

- administration SAG 12: 7-13
- administrative SAG 16: 16
- aging SAG 12: 6
- changing SAG 12: 6, 16: 16, 17: 21
- dial-up SAG 12: 9-13
- displaying information SAG 12: 6
- expiration SAG 12: 6-8
- forgotten SAG 12: 20, 17: 21
- locking SAG 12: 8
- rules for UG 2: 13
- setup SAG 12: 7-13
- status SAG 12: 7
- system SAG 16: 18-19
- paste(1) URM
- path
 - physical SS 6: 4
 - virtual SS 6: 4
- PATH environment variable SAG 8: 7, 17: 27; UG 4: 9, 9: 92
 - XENIX compatibility BSD/XNX 1: 1
- pathalias(4) SARM
- pathconv(1F) CHAR 2: 40, , C: 1
- pathfind(3G) PRM
- pathname UG 3: 7-14
 - absolute UG GL: 6
 - FACE UG 5: 40-42
 - full UG 3: 7-9, GL: 6
 - mapping installation ISDG 2: 42; SS 8: 48
 - relative UG 3: 10-14, GL: 11
 - rename with pkgproto ISDG 2: 43; SS 8: 49
- patterns
 - awk(1) UG 10: 7, 12, 18-19
 - ed(1) UG 6: 19-21, 51-60
 - vi(1) UG 7: 43-47
- pause MG A: 21
- pause(2) PRM

- pcat MG A: 9
- PCATCH bit ISDG 3: 28
- pcinfo data structure SS 5: 15
- pkt(7) STRM 12: 21
- pclose MG A: 21
- pcparms data structure SS 5: 19
- PDI SCSI 4: 16
- dis(1) PRM
- perflog (performance) log (BNU)
 - SAG 7: 72-74
- performance SAG 8: 1-51
 - command summary SAG 8: 85-87
 - file system SAG 8: 3-7
 - improving SAG 8: 3-8
 - kernel SAG 8: 12-15
 - monitoring SAG 8: 9-11
 - scheduler effect on SS 5: 30-34
- performance log (BNU) SAG 7: 72-74
- peripheral interrupt controllers ISDG 3: 36
- permanentmsg descriptor CHAR 2: 27, 4: 11, A: 18
- permissions SAG 12: 21; UG 3: 49-56, 4: 11, GL: 9
 - change existing UG 3: 52-54
 - default UG 3: 50
 - display UG 3: 50-52
 - file SAG 12: 22-24; SS 6: 3, 8, 12; UG 5: 60-62
 - group UG 3: 50
 - impact on directories UG 3: 54-55
 - IPC messages SS 4: 9-10
 - octal UG 3: 55
 - other UG 3: 50
 - read UG 3: 51, 53-56
 - semaphores SS 4: 39-40
 - shared memory SS 4: 68-69
 - user UG 3: 50

- write UG 3: 51, 53–56
 Permissions file (BNU) SAG 7: 48–56
 and uucp command UG 12: 5
 setup SAG 7: 22
 perror MG A: 21
 perror(3C) PRM
 persistent link STRM 10: 32–36
 personal programs (FACE) UG C: 17
 modifying UG C: 18
 removing UG C: 18
 pg MG A: 9
 pg(1) UG 3: 28, 32–37, 5: 91–92; URM
 physical layer, TCP/IP NUAG 1: 6
 physiock(D3D) DDRM
 physmap(D3DK) DDRM
 PIC ISDG 3: 36
 ping(1M) NI 5: 2; NUAG 6: 1; SARM
 pipe MG A: 21
 pipe(2) PRM
 PIPE_BUF STRM 11: 5
 pipemod STREAMS module STRM
 11: 6
 pipes CHAR 2: 8; UG 5: 92, 9: 20, GL: 10
 FMLI input CHAR A: 19
 named CHAR 2: 41
 STREAMS (see STREAMS-based
 pipe)
 pkgadd command ISDG 3: 39
 pkgadd(1M) ISDG 2: 25; SAG 14: 20, 22,
 34; SARM; SS 8: 26
 pkgask(1M) SARM; SS
 pkgchk(1M) SAG 14: 27; SARM; SS
 pkginfo(1) ISDG 2: 21; SAG 14: 29–33;
 SARM; SS 8: 22
 pkginfo(4) ISDG 2: 11–12, 21; SS
 8: 10–11, 22
 creating ISDG 2: 40; SS 8: 46
 pkgmap(4) ISDG 2: 12; SS 8: 18
 pkgmk(1) ISDG 2: 4–5, 46–48; SS 8: 6–7,
 54–55
 locating package contents ISDG
 2: 47; SS 8: 54
 pkgparam(1) ISDG 2: 21; SAG 14: 33;
 SARM; SS 8: 22
 pkgparam(4) ISDG 2: 21; SS 8: 22
 pkgproto(1) ISDG 2: 5, 43–46; SS 8: 6,
 8, 49–53
 assign objects to classes ISDG 2: 44;
 SS 8: 50
 create links ISDG 2: 45; SS 8: 52
 rename pathnames ISDG 2: 45; SS
 8: 51
 pkgrm command ISDG 3: 39
 pkgrm(1M) ISDG 2: 26; SAG 14: 36;
 SARM; SS 8: 28
 pkgtrans(1) ISDG 2: 5, 48–49; SARM;
 SS 8: 6–7, 56–57
 plock MG A: 21
 plock(2) PRM
 plot(1) BSD/XNX 1: 14
 plot(1G) BSD
 pmadm NI 13: 2, 16, 24
 pmadm(1M) NI; SAG 13: 2, 6, 10, 22,
 24–30, 35, 37, 40, 53, 59, 61–62, 65;
 SARM
 pmmmsg, SAF port monitor message
 structure NI 13: 9, 32
 _pmtab NI 13: 2, 6, 13–15
 pnch(4) SARM
 pointer
 declaration CGP 3: 27–28
 initialization CGP 3: 44
 poll driver routine ISDG 3: 19
 Poll file (BNU) SAG 7: 56
 and uucp command UG 12: 6
 poll(2) PRM; STRM 6: 1

- pollfd structure STRM 6: 3
- polling
 - error events STRM 6: 5
 - events STRM 6: 1
 - example STRM 6: 3-6
- pollwakeuper(D3DK) DDRM
- popen MG A: 21
- popen(3S) PRM
- port, enabling and disabling NI 13: 6
- port allocation NI 3: 50
- port monitor NI 7: 55, 13: 1-37
 - activity monitoring NI 13: 5
 - add SAG 13: 1, 16-18, 23
 - administrative command (see pmadm(1M))
 - changing service environment NI 13: 6
 - creating utmp entries NI 13: 6
 - disable SAG 13: 1, 18-19, 23
 - enable SAG 13: 1, 18-19, 23
 - enabling and disabling NI 13: 5
 - functions NI 13: 4-7
 - inetd(1M) NI 6: 12
 - listen(1M) NI 6: 12
 - new feature with Release 4.0 MG 5: 25
 - _pid file NI 13: 6
 - port management NI 13: 4
 - private files NI 13: 7
 - remove SAG 13: 1, 19, 23
 - start SAG 13: 1, 6, 18-19, 23
 - status SAG 13: 14-16
 - stop SAG 13: 1, 18-19, 23
 - terminating NI 13: 7
 - ttymon(1M) SAG 13: 31-45
- portability CGP 3: 52; NI 3: 40
 - lint(1) checks for CGP 5: 4-6
 - system calls CGP 2: 44
- porting
 - drivers changed for Release 4.0 MG 5: 22
 - hardware addressing changed for Release 4.0 MG 5: 24
 - memory management changed for Release 4.0 MG 5: 24
- ports(7) STRM
- pos_form_cursor(3X) CHAR 11: 84
- position-independent code CGP 2: 23, 13: 50-58, 60-66
- POSIX conformance, system calls changed for Release 4.0 MG 5: 8
- pos_menu_cursor(3X) CHAR 10: 63
- post_form(3X) CHAR 11: 56
- postinstall file ISDG 3: 53
- postinstall script ISDG 3: 71
- post_menu(3X) CHAR 10: 40
- PostScript SAG 9: 110-119
- pow MG A: 21
- pow(3) BSD/XNX 1: 26
- power failure SAG 6: 25
- power up SAG 16: 7
- powerdown SAG 6: 25, 34
 - from multi-user SAG 6: 34-35
 - from single-user SAG 6: 36
- powerdown(1M), login SAG 12: 18
- powerup SAG 6: 15
- pr MG A: 9
- pr(1) UG 3: 28; URM
- #pragma CGP 2: 30, 3: 18-19
- preemption latency SS 5: 33-34
- preemption point, kernel SS 5: 33
- preprocessing CGP 3: 11-20; CTG15-19
 - directives CGP 2: 2, 8-9, 3: 12-20
 - output CGP 2: 5-6
 - predefined names CGP 3: 20
 - stringizing CTG18

token pasting CTG19
 tokens CGP 3: 11-12
 preprocessor, m4 SS 1: 7
 preremove file ISDG 3: 54
 preremove script ISDG 3: 71
 prev_choice(3X) CHAR 11: 102
 prev_frm command (FMLI) CHAR
 2: 35, A: 12
 prevpage command (FMLI) CHAR
 2: 35, A: 12
 prf profiler commands SAG 8: 12-15
 prf(7) SARM
 PRFMAX parameter SAG 8: 81
 primary expressions CGP 3: 33-34
 print command (FACE) UG 5: 72-74,
 C: 13
 print wheels SAG 9: 28-33, 45-47
 print(D2DK) DDRM
 printenv(1) BSD; BSD/XNX 1: 25
 printer
 add SAG 9: 48
 as a login terminal SAG 9: 16
 check status of UG 8: 8-29
 classes SAG 9: 43-44, 47
 configuration SAG 9: 49-50
 control access to SAG 9: 39-40
 default destination SAG 9: 45
 descriptions SAG 9: 41
 direct connection SAG 9: 16
 disable SAG 9: 10, 51-53; UG 8: 12-13,
 25, 32
 enable SAG 9: 10, 51-53; UG 8: 12-13,
 25, 31
 faults SAG 9: 35-39
 moving requests SAG 9: 90-92
 non-direct connection SAG 9: 18-20
 port characteristics SAG 9: 123-124
 ports SAG 9: 25-28

PostScript SAG 9: 110-119
 remote SAG 9: 20-21
 specify name of SAG 9: 13
 troubleshooting SAG 9: 54-62
 types SAG 9: 22-23
 use of remote UG 8: 6
 printf MG A: 21
 printf(1) URM
 printf(3) BSD/XNX 1: 30
 printf(3S) BSD; PRM
 printing
 banner page with UG 8: 14
 cancel (in progress) UG 8: 5
 cancel request for UG 8: 11-12, 25,
 27, 30
 change request for UG 8: 10-11, 27
 change specifications for UG 8: 5
 character sets for UG 8: 3, 9-10, 14,
 18, 26, 29
 check status of UG 8: 5, 7-29
 content type for UG 8: 14-15, 26
 continuous (between files) UG 8: 17
 custom specifications for UG 8: 2-3
 default printer for UG 8: 28
 default specifications for UG 8: 2
 filters for UG 8: 3, 15, 19, 26
 forms for UG 8: 9, 14, 18, 26, 28
 notification of UG 8: 7, 27
 number of copies UG 8: 26
 number of copies for UG 8: 14
 output style SAG 9: 42-43
 page size for UG 8: 14, 16-17, 26
 pages for UG 8: 26
 pitch settings for UG 8: 14, 16-17, 26
 print wheels for UG 8: 3, 9-10, 14, 18,
 26, 29
 prioritize requests for UG 8: 5-6, 27
 request UG 8: 25

- skip banner page when UG 8: 17
- special modes for UG 8: 14, 19, 26
- specify printer for UG 8: 5-6, 14-15, 26
- printw MG A: 21
- printw(3X) CHAR 7: 6
- priocntl(1) SS 5: 9-13; URM
- priocntl(2) PRM; SS 5: 13-25
- priocntlset(2) PRM; SS 5: 25-27
- priority (see process priority)
- priority band data STRM 5: 3, 7: 35
 - flow control STRM 5: 30
 - flush handling example STRM 7: 36
- ioctl(2) STRM 5: 18
- routines STRM 5: 16
- service procedure STRM 5: 27
- proc(4) SARM
- procedure scripts ISDG 2: 18, 30-31; SS 8: 19, 32-33
 - postinstall ISDG 2: 30; SS 8: 32
 - postremove ISDG 2: 30; SS 8: 32
 - preinstall ISDG 2: 30; SS 8: 32
 - preremove ISDG 2: 30; SS 8: 32
- process UG GL: 10
 - spawning SS 2: 5-10
- process address space SS 7: 1, 15
- process group NI 3: 45
- process priority SAG 10: 3-4; SS 5: 3-4, 6-8
 - global SAG 10: 4, 7; SS 5: 4
 - of a sleeping process SAG 10: 13
 - real-time SAG 10: 10; SS 5: 6
 - setting and retrieving SS 5: 9-27
 - system SS 5: 7
 - time-sharing SAG 10: 11-13; SS 5: 7
- process scheduler (see scheduler)
- process state transition SS 5: 31
- processor priority levels ISDG 3: 25
- procset data structure SS 5: 25
- proct1 MG A: 21
- prof MG A: 12
- prof(1) CGP 1: 9, 2: 9-10, 7: 1-3; PRM
 - cc(1) profiling option CGP 7: 2
 - command line CGP 7: 2-3
 - environment variable CGP 7: 2
 - sample output CGP 7: 22, 27
 - use with lprof(1) CGP 7: 22-28
- prof(5) PRM
- profil MG A: 21
- profil(2) PRM
- profile(4) NUAG 23: 1; SARM; UG 4: 9
 - security SAG 12: 3
 - system SAG 17: 22
 - user SAG 17: 25; UG 4: 9, 5: 84, 9: 89-90
- profiler(1M) SAG 8: 12; SARM
- profilers (see lprof(1); prof(1))
- profiling, kernel SAG 8: 12-15
- programming, application SS 2: 1-29
- promotion CTG10-14
 - bit-fields CTG12
 - default arguments CTG3
 - integral constants CTG13
 - unsigned preserving CTG10
 - value preserving CTG10
- protocol
 - independence NI 2: 69-70, 4: 9
 - names NI 3: 25
 - negotiation of options NI 2: 24
 - selecting specific NI 3: 47
- protocols(4) NUAG 2: 18; SARM
- prototype(4) ISDG 2: 5-9, 41-46; SS 8: 11-15, 47-52, 9: 4
 - command lines ISDG 2: 8, 43; SS 8: 14, 49
 - creating manually ISDG 2: 41-43; SS 8: 47-49

creating with pkgproto ISDG
 2: 43-46; SS 8: 49-54
 description lines ISDG 2: 6; SS 8: 12
 prototypes (see function prototypes)
 prs MG A: 12
 prs(1) CGP 10: 12, 29-31; PRM
 prt(1) BSD; BSD/XNX 1: 19
 prtconf(1M) SAG 6: 39
 prtvtoc(1M) SARM
 ps MG A: 9
 ps(1) BSD; BSD/XNX 1: 24; SAG 8: 8; UG
 9: 32-34; URM
 PS1 environment variable SAG 17: 26
 pseudo-tty emulation module STRM
 12: 17-20
 pseudo-tty subsystem STRM 12: 15
 description STRM 12: 15-26
 drivers STRM 12: 22-25
 ldterm(7) STRM 12: 15
 messages STRM 12: 18
 packet mode STRM 12: 21
 remote mode STRM 12: 20
 psignal(3) BSD; BSD/XNX 1: 33
 psignal(3C) PRM
 psignal(D3D) DDRM
 pstat MG A: 9
 ptem structure STRM 12: 19
 ptem(7) STRM 12: 17, 19
 ptm (see master driver)
 ptob(D3DK) DDRM
 ptrace MG A: 21
 ptrace(2) PRM
 pts (see slave driver)
 ptsname(3C) STRM 12: 26
 with pseudo-tty driver STRM 12: 22
 public directory
 retrieve files (uupick) UG 12: 12-15
 uucppublic UG 12: 12-15

 public key cryptography NUAG 18: 3
 publickey map NI 11: 9
 publickey(3N) NI
 publickey(4) NI; NUAG
 pullupmsg(D3DK) DDRM
 pump(1M), error messages SAG
 E: 45-46
 putbq(D3DK) DDRM
 PUTBUFSZ parameter SAG 8: 70
 putc MG A: 21
 putc(3S) PRM
 putchar MG A: 21
 putct11(D3DK) DDRM
 putct1(D3DK) DDRM
 put(D2DK) DDRM
 putdev(1) SARM
 putdev(1M) SAG 15: 35, 40-41
 putdgrp(1) SARM
 putdgrp(1M) SAG 15: 42, 45-46
 putenv MG A: 21
 putenv(3C) PRM
 putmsg(2) PRM; STRM 5: 9
 putnext(D3DK) DDRM
 putpmsg function STRM 5: 11
 putpwent MG A: 21
 putpwent(3C) PRM
 putq(D3DK) DDRM
 puts MG A: 21
 puts(3S) PRM
 putspent(3C) PRM
 pututline MG A: 21
 putw MG A: 21
 pwadmin MG A: 9
 pwcheck MG A: 9
 pwck(1M) BSD; BSD/XNX 1: 23; SARM
 pwconv(1M) SARM
 pwd MG A: 9
 pwd(1) UG 3: 6-7; URM

Q

qband(D4DK) DDRM
 qenable(D3DK) DDRM
 qinit(D4DK) DDRM
 qreply(D3DK) DDRM
 qsize(D3DK) DDRM
 qsort MG A: 21
 qsort(3C) PRM
 qt(7) SARM
 queue(D4DK) DDRM
 quot MG A: 9
 quot(1M) SARM
 quota(1M) SARM
 quotacheck(1M) SARM
 quotaon(1M) SARM
 quoting mechanisms (FMLI) CHAR
 2: 6

R

raise(3C) PRM
 rand MG A: 21
 rand(3) BSD/XNX 1: 30
 rand(3C) BSD; PRM
 random MG A: 9
 random(1) XNX
 random(3) BSD; BSD/XNX 1: 27
 ranlib MG A: 12
 RARP (Reverse Address Resolution
 Protocol) NUAG 1: 3, 6
 rarpd(1M) NUAG; SARM
 ratfor MG A: 12
 raw MG A: 21
 raw I/O DDRM 3: 7
 raw sockets NI 3: 3
 rc file ISDG 3: 52
 rc initialization scripts SAG 6: 16,
 24-25

rc0(1M) SARM
 rc2(1M) SARM
 rc6(1M) SARM
 rcp MG A: 9
 rcp(1) NUAG 22: 2; URM
 rdate(1M) NUAG; SARM
 rdchk MG A: 21
 rdchk(2) XNX
 RD(D3DK) DDRM
 rdepend installation parameter SAG
 14: 15
 read MG A: 21
 read lock SS 3: 2, 4, 9, 13
 read(2) NI 2: 47, 49; PRM
 read-ahead UG 2: 8, GL: 11
 read(D2DK) DDRM
 readdir MG A: 21
 readfile(1F) CHAR 2: 40, 4: 4, , A: 16,
 C: 1
 readlink(2) PRM
 read-side
 console driver service procedure
 STRM G: 13
 console processing STRM G: 12
 definition STRM 2: 3
 ldterm(7) messages STRM 12: 5
 ldterm(7) processing STRM 12: 5
 ports driver service procedure
 STRM G: 9
 put procedure STRM 8: 1
 xt driver STRM G: 28
 read/write interface NI 2: 47-50
 example of client program NI
 2: 79-81
 realloc MG A: 21
 realpath(3C) PRM
 real-time
 changed for Release 4.0 MG 5: 28

- process priority SAG 10: 10
 scheduler class SAG 10: 5; SS 5: 5
 scheduler parameter table SAG
 10: 5, 9–10; SS 5: 5
 scheduler policy SAG 10: 9–10
 reboot MG A: 9; SAG 6: 17, 37
 reboot system state SAG 6: 24
 reboot(1) BSD/XNX 1: 23
 reboot(1M) BSD
 reboot(3) BSD; BSD/XNX 1: 32
 re_comp(3) BSD/XNX 1: 27
 reconfiguring new kernel ISDG 3: 58
 records, locking (see locking)
 recv(3N) NI
 red MG A: 9
 redescribe command (FACE) UG
 5: 48, C: 9
 redirect input or output UG 9: 14–26
 re_exec(3) BSD/XNX 1: 27
 refer(1) BSD; BSD/XNX 1: 15
 refresh MG A: 21
 refresh command
 FACE UG C: 23
 FMLI CHAR 2: 35, A: 18
 refresh(3X) CHAR 6: 3, 9
 regcmp MG A: 12, 21
 regcmp(1) PRM
 regcmp(3G) PRM
 regex MG A: 21
 regex(1F) CHAR 2: 40, , A: 14–15, C: 1
 example of CHAR 3: 30
 regex(3) BSD
 regexp MG A: 21
 regexp(5) PRM; SARM
 regexp.h header file BSD/XNX 1: 35
 regexpr(3G) PRM
 register CGP 3: 26
 regular expressions CGP 11: 5–9; UG
 3: 60
 awk(1) UG 10: 14–17
 ed(1) UG 6: 51–60
 regular link SS 6: 1
 reinit(1F) CHAR 2: 40, , C: 1
 reinstallation SAG 14: 23
 reject MG A: 9
 REL parameter SAG 8: 70
 related documentation ISDG 1: 5
 relative pathname (see pathname)
 relay NUAG 4: 3
 release command (FMLI) CHAR
 2: 35
 Release number SCSI 2: 4, 4: 15
 relocatable files CGP 2: 11, 13: 1 (see
 also object files)
 relocation CGP 13: 30–38
 relogin(1M) SARM; URM
 CGP 3: 36
 remote MG A: 9
 remote file copy NUAG 22: 2
 copying directories from local to
 remote NUAG 22: 4
 copying directories from remote to
 local NUAG 22: 4
 copying files from local to remote
 NUAG 22: 3
 copying files from remote to local
 NUAG 22: 1
 error messages NUAG 22: 5
 Remote File Sharing ISDG 3: 57
 remote login NUAG 24: 1, 5, 25: 7
 aborting a connection NUAG 24: 2, 7
 non-existent machine NUAG 24: 5
 problems NUAG 24: 5
 suspending a connection NUAG
 24: 3, 6
 to different operating system NUAG
 24: 5

- remote machine communication
 - administration (see BNU)
 - call up terminal (see ct)
 - connect to remote (see cu)
 - execute commands on remote (see uux)
 - remote execution NUAG 23: 1
 - send files UG 12: 2-15
 - uucp(1C) UG 12: 2-7
 - uupick(1C) UG 12: 12-15
 - uustat(1C) UG 12: 10-12
 - uuto(1C) UG 12: 8-10
- Remote Procedure Call (see RPC)
- remote procedure naming conventions NI 6: 8
- remote program NI 5: 2
- remote resources
 - automatic mounting of NUAG 9: 13, 15: 13
 - automatic sharing of NUAG 9: 4, 15: 4
 - explicit mounting of NUAG 9: 10, 15: 10
 - explicit sharing of NUAG 9: 1, 15: 1
 - mounting a set of NUAG 9: 16, 15: 16
 - unmounting NUAG 9: 15, 15: 15
- remote.unknown SAG 7: 61
- remove(3C) PRM
- removef(1M) ISDG 2: 30; SS 8: 32
- rename a file UG 3: 43-45
- rename command (FACE) UG 5: 47, C: 10
- rename(2) PRM
- renice(1M) BSD; BSD/XNX 1: 24
- repeater NUAG 4: 2
- repinsb(D3D) DDRM
- repinsd(D3D) DDRM
- repinsw(D3D) DDRM
- report generation, awk(1) UG 10: 52
- repout.sb(D3D) DDRM
- repout.sd(D3D) DDRM
- repout.sw(D3D) DDRM
- repquota(1M) SARM
- reread descriptor CHAR 2: 18, 21, 24, 3: 8, 37, 60, A: 16, 18
 - example of CHAR 3: 18
- reserved names CTG28-31
 - for expansion CTG30
 - for implementation use CTG29
 - guidelines for choosing CTG31
- reset command BSD/XNX 1: 14
- reset command (FMLI) CHAR 2: 35, A: 15
- reset(1F) CHAR, C: 1
- resolv.conf(4) NUAG; SARM
- resolver, definition NUAG 5: 1
- resolver(3N) NI; NUAG
- resolver.conf(4) NUAG 5: 9
- restor MG A: 9
- restore MG A: 9
- restore jobs
 - canceling SAG 11: 16
 - checking the status SAG 11: 10-12
 - observing the progress SAG 11: 10
 - removing SAG 11: 16
 - servicing pending jobs SAG 11: 14-16
- restore operations SAG 11: 4
 - assigning an operator SAG 11: 6
 - changed for Release 4.0 MG 5: 30
 - checking the status SAG 11: 10-12
 - directories and files SAG 11: 7
 - disk objects SAG 11: 8
- restore service
 - core file systems SAG 11: 9
 - job ID SAG 11: 3

- overview SAG 11: 3-6
- pre-Release 4.0 archives SAG 11: 5
- rsnotify(1M) SAG 11: 6
- shell interface SAG 11: 1-2
- summary of commands SAG 11: 25-27
- sysadm(1M) interface SAG 11: 1-2
- tables SAG 11: 11-14
- terminology SAG 11: 3
- restore(1) XNX
- restore(1M) SARM
- restty MG A: 21
- RET variable CHAR 2: 12
- retrieving kernel information ISDG 3: 67
- return statement CGP 3: 51
- reverse video attribute CHAR 2: 47
- rewind MG A: 21
- rewinddir MG A: 22
- rexec(3N) NI; NUAG
- rexcld(1M) NUAG; SARM
- rfadmin(1M) NUAG; SARM
- rfmaster(4) NUAG; SARM
 - create NUAG 12: 4-6
- rfpasswd(1M) NUAG; SARM
- RFS ISDG 3: 57
 - active NUAG 12: 29
 - auto start init NUAG 12: 32-35
 - enter init 3 NUAG 12: 33
 - initial start NUAG 12: 29-30
 - mode scripts NUAG 12: 35
 - password NUAG 12: 30-32
 - starting NUAG 12: 29-35
 - stopping NUAG 12: 36
 - system state SAG 6: 23
- RFS resources
 - auto remote mount NUAG 12: 43
 - auto share NUAG 12: 37-38
 - identifier NUAG 10: 3
 - local mount table NUAG 12: 45-46
 - local share table NUAG 12: 38-39
 - local sharing NUAG 12: 37-42
 - local unmount NUAG 12: 48
 - mount guidelines NUAG 12: 43
 - mount rules NUAG 12: 44-45
 - remote disconnect NUAG 12: 46-48
 - remote mounting NUAG 12: 42-48
 - shared in use NUAG 12: 40
 - sharing NUAG 10: 3, 12: 48-49; SAG 9: 9-10
 - unmount remote NUAG 12: 41-42
 - unshare NUAG 12: 41
- rfstart(1M) NUAG; SARM
- rfstop(1M) NUAG; SARM
- rfuadmin(1M) NUAG; SARM
- rfudaemon(1M) NUAG; SARM
- .rhosts NUAG 3: 4
- rindex(3) BSD/XNX 1: 29
- rlogin(1) NUAG 24: 1; URM
- rlogind(1M) NUAG; SARM
- rm MG A: 9
- rm(1) UG 3: 28, 45-46; URM
- rmalloc(D3DK) DDRM
- rmdel MG A: 12
- rmdel(1) CGP 10: 12, 31-32; PRM
- rmdir MG A: 9
- rmdir(1) UG 3: 26-27
- rmdir(2) PRM
- rmenu descriptor CHAR 2: 3, 21, 3: 40-41, A: 5, 15
- rmfree(D3DK) DDRM
- rminit(D3DK) DDRM
- rmntstat(1M) NUAG; SARM
- rmnttry(1M) NUAG; SARM
- rmount(1M) NUAG; SARM
- rmountall(1M) NUAG; SARM

- rmsetwant(D3DK) DDRM
- rmuser MG A: 9
- rmvb(D3DK) DDRM
- rmvq(D3DK) DDRM
- rmwant(D3DK) DDRM
- roffbib(1) BSD; BSD/XNX 1: 16
- root
 - directories SAG 5: 4, B: 7-10; UG 1: 11-13, GL: 11
 - disk partition SAG A: 3
 - file system changed for Release 4.0 MG 5: 15
 - login SAG 12: 19
 - password recovery SAG 12: 20
- ROOTFSTYPE parameter SAG 8: 70
- rounding behavior CGP 3: 32
- route(1M) NUAG; SARM
- routed(1M) NUAG; SARM
- router NUAG 4: 2
 - configuring NUAG 4: 4
- routing(4) NUAG; SARM
- rows descriptor CHAR 2: 18, 22, 24, 26, 3: 8, 42, 61, 4: 4, A: 14
- RPC authentication NUAG 18: 2
- RPC Language Reference NI 6: 30
- RPC lower levels NI 5: 6, 13, 7: 1, 17
 - bottom level NI 5: 1, 7: 31
 - client-side NI 5: 13, 7: 18, 22, 25, 31, 47
 - expert level NI 5: 2, 7: 25
 - intermediate Level NI 5: 2, 7
 - intermediate level NI 7: 22
 - server-side NI 5: 14, 7: 20, 24, 28, 32, 48, 53
 - top Level NI 5: 3
 - top level NI 7: 18
- RPC Programming Guide NI 7: 1
- RPC (Remote Procedure Call)
 - administration NI 9: 8, 10: 1-13
 - AUTH_DES NI 7: 51
 - authentication NI 7: 46, 51
 - batching NI 7: 42
 - bottom level NI 7: 4
 - broadcast NI 7: 40
 - broadcast synopsis NI 7: 41
 - built-in routines NI 7: 12
 - callback procedures NI 7: 64
 - changed for Release 4.0 MG 5: 35
 - characteristics of RPC NI 5: 6
 - client-side structure NI 7: 33
 - data passing NI 7: 12
 - DES NI 7: 51
 - expert level NI 7: 3
 - guarantees NI 7: 48
 - intermediate level NI 7: 3
 - language NI 5: 2, 9: 23-25
 - library based network services NI 7: 6
 - package NI 5: 2, 7: 1
 - program number assignment NI 9: 7
 - protocol NI 5: 2
 - raw NI 7: 36
 - record marking NI 9: 22
 - select(3C) NI 7: 39
 - server-side structure NI 7: 33
 - simplified interface NI 5: 3, 7: 2, 6
 - top level NI 7: 2
 - versions NI 5: 6, 7: 59
- rpc(3N) NI
- rpc(4) NI
- RPC-based services SAG 13: 56
- rpcb_getaddr function NI 7: 3
- rpcbind(1M) NI 5: 2, 4, 6, 8, 10-13; NUAG
 - address registration NI 5: 11
 - operation NI 9: 28
 - protocol NI 9: 26

- RPCBPROC_UADDR2TADDR NI 9: 30
 RPCPROC_CALLIT NI 9: 29
 RPCPROC_DUMP NI 9: 29
 RPCPROC_GETADDR NI 9: 29
 RPCPROC_GETTIME NI 9: 30
 RPCPROC_NULL NI 9: 28
 RPCPROC_SET NI 9: 28
 RPCPROC_TADDR2UADDR NI 9: 30
 RPCPROC_UNSET NI 9: 29
 rpcbind(3N) NI
 rpc_broadcast function NI 7: 2
 rpcb_set function NI 7: 3
 rpcb_unset function NI 7: 3
 rpc_call function NI 7: 2, 9
 rpc_clnt_auth(3N) NI
 rpc_clnt_calls(3N) NI
 rpc_clnt_create(3N) NI
 rpcgen(1) NI 5: 1-2, 13, 6: 1-3, 11-12, 21, 25-26
 broadcast RPC NI 6: 25
 client authentication NI 6: 23
 command line define statements NI 6: 21, 24
 constants NI 6: 31
 C-preprocessing NI 6: 19
 debugging NI 6: 29
 declarations NI 6: 32
 definitions NI 6: 30
 dispatch index output NI 6: 11
 dispatch tables NI 6: 27
 enumerations NI 6: 31
 generating XDR routines NI 6: 12, 27
 local procedures NI 6: 3
 -n option NI 6: 22
 network types NI 6: 21
 null procedures NI 6: 6
 options NI 6: 1
 output NI 6: 1, 14
 output file names NI 6: 11
 overview NI 6: 1
 port monitor support NI 6: 26
 preprocessing directives NI 6: 19-20
 procedure naming conventions NI 6: 8
 programs NI 6: 35
 remote procedures NI 6: 3
 RPC Language NI 6: 30
 server wait before exit NI 6: 26
 service routine initializer NI 6: 28
 special cases NI 6: 37
 static results NI 6: 8
 structures NI 6: 34
 -T option NI 6: 11, 27
 timeout changes NI 6: 22
 transport-specific servers NI 6: 22
 tutorial NI 6: 3
 typedef NI 6: 32
 unions NI 6: 13, 34
 xdr_functions NI 6: 14
 rpcinfo(1M) NI; NUAG
 rpc_reg function NI 7: 2, 11
 rpc_svc_calls(3N) NI
 rpc_svc_create(3N) NI
 rpc_svc_err(3N) NI
 rpc_svc_reg(3N) NI
 RPC/XDR library NI 6: 14
 rpc_xdr(3N) NI
 rpow(3) BSD/XNX 1: 26
 rsh MG A: 9
 rsh(1) NUAG 23: 1; URM
 rshd(1M) NUAG; SARM
 rsnotify(1M) SAG 11: 6
 rsoper(1M) SAG 11: 14-17
 rsstatus(1M) SAG 11: 11-14
 rsstatus.tab restore table SAG 11: 11-14

RSTCHOWN parameter SAG 8: 75
rtc(7) SARM
rt_dptbl(4) SAG 10: 9-10, 17; SARM
rumount(1M) NUAG; SARM
run command
 FACE UG 5: 94, C: 24
 FMLI CHAR A: 1; SAG C: 15
run level (see system state)
run(1F) CHAR 2: 41, , C: 1
runacct(1M) SARM
 error messages SAG 2: 12-13
 files produced SAG 2: 13-14
 reentrant states SAG 2: 10-12
 restarting SAG 2: 17
 run accounting procedure SAG
 2: 10-13
runbig MG A: 9
runlevel installation parameter SAG
 14: 13
ruptime(1) NI 3: 36; NUAG; URM
rusers(1) NI; NUAG
rusers(3N) NI
rusersd(1M) NI; NUAG
rwall(1M) NI; NUAG
rwall(3N) NI
rwalld(1M) NI; NUAG
rwho(1) NUAG; URM
rwhod(1M) NUAG; SARM

S

s5-FSType SAG 5: 6-11
 boot block SAG 5: 8
 disk block address SAG 5: 9
 free blocks SAG 5: 11
 indirect addressing SAG 5: 11
 inodes SAG 5: 9
 storage blocks SAG 5: 11
 super block SAG 5: 8
sa(7) SARM
SAC (Service Access Controller) SAG
 13: 5-7
 administrative command (see
 sacadm(1M))
 administrative files SAG 13: 5, 8-10,
 13, 22
 functions SAG 13: 6-7
 port monitor SAG 13: 31-45
sac(1M) NI; SAG 6: 16, 13: 7; SARM
sacadm NI 13: 2-3, 15, 23-24
sacadm(1M) NI; SAG 13: 1, 5-6, 8,
 13-16, 19-22, 35-36, 39, 58-60, 65;
 SARM
sacmsg, SAF sac message structure
 NI 13: 8, 33
sact MG A: 13
sact(1) CGP 10: 12, 31; PRM
_sactab NI 13: 2-3, 12-13
SAD (see STREAMS Administrative
 Driver)
sad(7) STRM
sadc sampling command SAG 8: 17
sadb(1M) SAG 8: 43
sa_exlun SCSI 4: 12
SAF (Service Access Facility) NI
 13: 1-37; SAG 13: 1-13
 changed for Release 4.0 MG 5: 37
 configuration language NI 13: 22-23
 configuration script, per-port moni-
 tor NI 13: 21, 24
 configuration script, per-port moni-
 tor, sample NI 13: 26
 configuration script, per-service NI
 13: 6-7, 21, 24
 configuration script, per-service,
 sample NI 13: 26

- configuration script, per-system NI 13: 2, 21, 23
- configuration script, per-system, sample NI 13: 26
- configuration scripts NI 13: 2, 21; SAG 13: 5-8, 18-21, 23-24, 28-30
- configuration scripts, printing, installing, replacing NI 13: 23-24
- configuration scripts, sample NI 13: 26-27
- directory structure NI 13: 35-37
- header file `sac.h` NI 13: 32-34
- interpreting configuration scripts NI 13: 25
- logic diagram NI 13: 35
- managing services under SAG 13: 2, 24-30
- message classes NI 13: 10-11
- message formats NI 13: 8-10
- new service invocation NI 13: 17
- port monitor, sample code NI 13: 27-31
- port monitor administrative command `pmadm` NI 13: 2, 16
- port monitor administrative file `_pmtab` NI 13: 2, 6, 13-15
- port monitor messages NI 13: 9-10
- port monitor requirements NI 13: 18-20
- port monitor requirements, environment NI 13: 18
- port monitor requirements, files NI 13: 18-19
- "port monitor-specific" administrative command NI 13: 16-17
- SAC administrative command `sacadm` NI 13: 2-3, 15
- SAC administrative file `_sactab` NI 13: 2-3, 12-13
- `sac` messages NI 13: 8-9
- Service Access Controller, environment variables NI 13: 8
- Service Access Controller (`sac`) NI 13: 2-3, 8-11
- standing service invocation NI 13: 17
- `sa_fill` SCSI 4: 12
- `sag(1)` SAG 8: 39; URM
- `sa_lun` SCSI 4: 12
- `sa_major` SCSI 4: 12
- `SAMESTR(D3DK)` DDRM
- `sa_minor` SCSI 4: 12
- `sar(1)` SAG 8: 16-40; URM
- `sar(1M)` SARM
- `sassign` file ISDG 3: 41
- `savetty` MG A: 22
- `SAVEXP` parameter SAG 8: 78
- SB (SCSI Block) SCSI 4: 2
- `sb_b` SCSI 4: 2
- `sb(D4I)` SCSI 4: 16
- `sb(D4I)` SCSI Block structure SCSI 1: 7
- `sbrk` MG A: 22
- `sbrk` (see `brk(2)`)
- `sb_type` SCSI 4: 2
- `sb_type` use with SCSI Control Block SCSI 4: 4
- scalar types CGP 3: 21
- `scalbn` function BSD/XNX 1: 29
- `scale_form(3X)` CHAR 11: 51
- `scale_menu(3X)` CHAR 10: 33
- `scandir(3)` BSD; BSD/XNX 1: 28
- `scanf` MG A: 22
- `scanf(3S)` PRM
- scanning, in `xt` driver STRM G: 33
- `scanw` MG A: 22
- `scanw(3X)` CHAR 7: 20
- SCB (SCSI Control Block) SCSI 4: 4

- scb(D4I) SCSI 4: 16
- SCB_TYPE SCSI 4: 3
- sc_cmdpt SCSI 4: 6
- sc_cmdpt use SCSI 4: 4
- sc_cmdsz SCSI 4: 8
- sc_cmdsz use SCSI 4: 4
- sc_comp_code SCSI 4: 5
- sc_comp_code error condition usage SCSI 4: 4
- SCCS CGP 1: 10, 10: 1–40
 - auditing files CGP 10: 39–40
 - changing comments CGP 10: 32–33
 - changing file parameters CGP 10: 26, 28–29
 - commands CGP 10: 10–36
 - creating files CGP 10: 2–3, 26–29
 - file format CGP 10: 38–39
 - file protection CGP 10: 37–38
 - ID keywords CGP 10: 14
 - marking differences CGP 10: 26, 34
 - printing files CGP 10: 29–31
 - removing versions CGP 10: 31–32
 - retrieving files CGP 10: 3–6, 13–23
 - updating files CGP 10: 4, 23–26
 - usage example CGP 10: 2–6
 - use with make(1) CGP 9: 16–18
 - version numbering CGP 10: 7–9
- sccs(1) BSD; BSD/XNX 1: 19
- sccsdiff MG A: 13
- sccsdiff(1) CGP 10: 12, 34; PRM
- sccsfile(4) PRM; SARM
- sc_datapt SCSI 4: 6
- sc_datapt use SCSI 4: 4
- sc_datasz SCSI 4: 8
- sc_datasz use SCSI 4: 4
- sc_dev SCSI 4: 7
- sc_fill SCSI 4: 7
- scheduler SAG 10: 1; SS 2: 22–23, 5: 1
 - changed for Release 4.0 MG 5: 28
 - configuration SAG 10: 6–15
 - effect on performance SS 5: 30–34
 - parameter tables SAG 10: 9–14
 - real-time policy SAG 10: 5, 9–10; SS 5: 5
 - system policy SAG 10: 5; SS 5: 5
 - time-sharing policy SAG 10: 4, 10–13; SS 5: 4
 - tunable parameters SAG 10: 8–9
- scheduler class SAG 10: 3–5; SS 5: 3–5
 - default SAG 10: 7
 - installing SAG 10: 15
 - real-time SAG 10: 5; SS 5: 5
 - removing SAG 10: 14–15
 - system SAG 10: 5; SS 5: 5
 - time-sharing SAG 10: 4; SS 5: 4
- scheduler data structures
 - pcinfo SS 5: 15
 - pcparms SS 5: 19
 - procset SS 5: 25
- sc_int SCSI 4: 6
- sc_int member of scb(D4I) SCSI 1: 7
- sc_link SCSI 4: 7
- sc_link use SCSI 4: 4
- scm(D4I) SCSI 4: 16
- sc_mode SCSI 4: 7
- scope CGP 3: 24–25
- SCORLIM parameter SAG 8: 75
- SCPULIM parameter SAG 8: 74
- screen descriptor CHAR 2: 28, 4: 8
- screen editor (see vi)
- screen-labeled function keys CHAR 1: 4, 24, A: 12
 - (see function keys (FACE) also)
 - default assignments CHAR 1: 16, 2: 37, 4: 13
 - defining color of CHAR 4: 8

- descriptor definitions CHAR 4: 13–15
- display alternate set CHAR 3: 4, 34, 57
- examples of CHAR 4: 16
- FACE UG 5: 5
- how to disable CHAR 4: 15
- layout of CHAR 4: 11
- list of descriptors CHAR 2: 28
- problems CHAR A: 1
- sc_resid SCSI 4: 8
- script(1) URM
- scripts (package)
 - class action ISDG 2: 18, 23–30; SS 8: 19, 25–31
 - procedure ISDG 2: 18, 30–31; SS 8: 19, 32–33
 - processing ISDG 2: 19; SS 8: 20
 - request ISDG 2: 18, 22–23; SS 8: 19, 23–25
- scroll MG A: 22
- scroll box (FMLI) CHAR 1: 6, 8–9, A: 11
- scroll descriptor CHAR 2: 22, 3: 42
- scroll symbol
 - FACE UG 5: 14
 - FMLI CHAR 1: 6, 8–9
- scrollok MG A: 22
- scs(D4I) SCSI 4: 16
- SCSI SCSI 4: 16
- SCSI Block SCSI 4: 2
- SCSI bus SCSI 4: 16
- SCSI cartridge tape
 - SAG 15: 20 (see also cartridge tape)
 - duplicate a SAG 15: 20, 22
- SCSI Control Block SCSI 4: 4
- SCSI device, definition SAG 15: 5
- SCSI device address structure SCSI 4: 12
- SCSI Function Block SCSI 4: 13
- SCSI modes SCSI 4: 15
- SCSI software architecture figure
 - SCSI 1: 2
- SCSI target driver routines SCSI 1: 6
- scsi_ad(D4I) SCSI 4: 16
- sc_status SCSI 4: 7
- sc_time SCSI 4: 7
- sc_wd SCSI 4: 6
- SDATLIM parameter SAG 8: 74
- sdb MG A: 13
- sdb(1) CGP 1: 8–9, 2: 9, 6: 1–22; CHAR 6: 7; PRM
 - cc(1) debugging option CGP 6: 3
 - command line CGP 6: 3–4
 - displaying files CGP 6: 10–11
 - displaying machine language CGP 6: 16–17
 - examining variables CGP 6: 6–10
 - printing stack trace CGP 6: 1, 5
 - setting breakpoints CGP 6: 1, 12–16
 - setting watchpoints CGP 6: 15
 - starting live process CGP 6: 12–16
 - usage example CGP 6: 18–22
 - use with optimizer CGP 6: 3
 - use with yacc(1) CGP 12: 32
- sddate MG A: 9
- sdenter MG A: 22
- sdenter(2) XNX
- sdevice file ISDG 3: 14, 41, 50
- sdevice(4) SARM
- sdfree MG A: 22
- sdget MG A: 22
- sdget(2) XNX
- sdgetv MG A: 22
- sdgetv(2) XNX
- SDI SCSI 4: 16
- SDI defined SCSI 1: 1
- SDI failure SCSI 4: 2

- SDI input/output controls SCSI 1: 3
- SDI Interrupt routine SCSI 1: 6
- SDI_ASW SCSI 4: 10
- SDI_CKSTAT SCSI 4: 4, 10
- SDI_ERROR SCSI 4: 5
- sdiff MG A: 9
- sdiff(1) URM
- sdi_freeblk(D3I) SCSI 4: 16
- sdi_freeblk(D3I) SCSI
- sdi_getblk usage with sb(D4I) SCSI 4: 2
- sdi_getblk(D3I) SCSI 4: 16
- sdi_getblk(D3I) SCSI
- sdi_getdev(D3I) SCSI 4: 17
- sdi_getdev(D3I) SCSI
- SDI_HAERR SCSI 2: 11, 4: 10
- sdi_icmd(D3I) SCSI 4: 17
- sdi_icmd(D3I) SCSI
- sdi_icmd(D3I) use with SCSI Control Block SCSI 4: 4
- sdi_init(D3I) SCSI 4: 17
- sdi_init(D3I) SCSI
- SDI_MESS SCSI 4: 5
- sdi_name(D3I) SCSI 4: 17
- sdi_name(D3I) SCSI
- SDI_NOALLOC SCSI 4: 10
- SDI_NOSELE SCSI 4: 10
- SDI_ONEIC SCSI 4: 10
- SDI_OOS SCSI 4: 10
- SDI_PROGRES SCSI 2: 6, 11, 4: 10
- SDI_QFLUSH SCSI 4: 10
- SDI_RESET SCSI 1: 9, 4: 10
- SDI_RETRY SCSI 4: 5
- SDI_SCBERR SCSI 4: 9
- sdi_send(D3I) SCSI 4: 17
- sdi_send(D3I) SCSI
- sdi_send(D3I) use with SCSI Control Block SCSI 4: 4
- SDI_SFBERR SCSI 4: 10
- SDI_SUSPEND SCSI 4: 5
- SDI_TIME SCSI 4: 10
- sdi_translate(D3I) SCSI 4: 17
- sdi_translate(D3I) SCSI
- sdi_translate(D3I) use with SCSI Control Block SCSI 4: 4
- SDI_UNUSED SCSI 4: 10
- sdiv(3) BSD/XNX 1: 26
- sdleave MG A: 22
- sdwaitv MG A: 22
- seconvert(3) BSD/XNX 1: 28
- secret key NUAG 18: 3
- sector (disk) SAG 4: 38
- secure, option to NFS share NUAG 14: 5
- Secure NFS NUAG 18: 1
 - administering NUAG 18: 7-8
 - YP NI 11: 8
- Secure RPC NUAG 18: 1-6
 - YP NI 11: 8
- secure_rpc(3N) NI
- security NUAG 10: 8-11; SAG 12: 1-30
 - command summary SAG 12: 29-30
 - file SAG 12: 21-24
 - guidelines SAG 12: 3-4
 - login and password SAG 12: 5-17
 - map IDs NUAG 10: 10-11
 - NFS (see network security)
 - restrict resources NUAG 10: 9-10, 12: 38
 - special logins SAG 12: 18-19
 - verify remote computer NUAG 10: 9, 12: 7-9
- security command (FACE) UG 5: 60, C: 11
- security log (BNU) SAG 7: 70
- sed MG A: 9

- sed(1) URM
 seed48 MG A: 22
 seekdir MG A: 22
 segmap(D2K) DDRM
 select(3C) NI 3: 17, 19, 47, 7: 40
 connection NI 3: 20
 on the server side NI 7: 39
 selected descriptor CHAR 2: 19, 3: 11
 SELECTED variable CHAR 2: 12
 SEMAEM parameter SAG 8: 83
 semaphore SS 4: 32-62
 control (semctl) SS 4: 44-46
 get (semget) SS 4: 37-40
 identifier (semid) SS 4: 34-37
 operations (semop) SS 4: 56-57
 permission codes SS 4: 39-40
 semctl example program SS
 4: 46-56
 semget example program SS
 4: 41-44
 semop example program SS
 4: 57-62
 set data structure SS 4: 34-37
 tunable parameters SAG 8: 82-83
 usage SS 4: 34-37
 semctl MG A: 22
 semctl(2) PRM; SS 4: 44-56
 example program SS 4: 46-56
 summary SS 4: 36
 usage SS 4: 44-46
 semget MG A: 22
 semget(2) PRM; SS 4: 37-44
 example program SS 4: 41-44
 usage SS 4: 37-40
 SEMMAP parameter SAG 8: 82
 SEMNI parameter SAG 8: 82
 SEMNS parameter SAG 8: 82
 SEMNU parameter SAG 8: 82
 SEMMSL parameter SAG 8: 83
 semop MG A: 22
 semop(2) PRM; SS 4: 56-62
 example program SS 4: 57-62
 usage SS 4: 56-57
 SEMOPM parameter SAG 8: 83
 SEMUME parameter SAG 8: 83
 SEMVMX parameter SAG 8: 83
 send files
 to a local user UG 12: 2-10
 to a remote machine UG 12: 2-10
 to local from remote machine UG
 12: 2-7
 uucp(1C) UG 12: 2-7
 uupick(1C) UG 12: 12-15
 uustat(1C) UG 12: 10-12
 uuto(1C) UG 12: 8-10
 send mail (see mail(1), mailx(1))
 send(3N) NI
 sendmail(1M) BSD; BSD/XNX 1: 22
 sendto function NI 3: 55
 serialize NI 5: 3, 15
 servers NI 2: 11, 16, 20, 24, 27-31, 34-38,
 5: 3; NUAG 7: 4
 connectionless-mode NI 3: 36
 RPC NI 5: 12
 servers (YP) NI 11: 3-6, 15-21
 Service Access Controller (see SAC)
 Service Access Controller (SAC) NI
 13: 2
 Service Access Facility (see SAF)
 service interface STRM 5: 35-37
 definition STRM 5: 34
 library example STRM 5: 38-47
 rules STRM 5: 47
 service primitive STRM 5: 37
 in service procedure STRM 5: 39
 service provider STRM 5: 37

accessing STRM 5: 40
closing STRM 5: 43
receiving data STRM 5: 45
sending data STRM 5: 44
services(4) NI 12: 14; NUAG 2: 19;
SARM
set(1F) CHAR 2: 10, 41, , A: 3, C: 1
setbuf MG A: 22
setbuf(3S) BSD; PRM
setbuffer(3) BSD/XNX 1: 30
setbuffer(3S) BSD
setclk(1M) SARM
setcolor MG A: 10
setcolor(1F) CHAR 2: 41, 4: 9, , C: 1
example of CHAR 4: 8
set_current_field(3X) CHAR 11: 81
set_current_item(3X) CHAR 10: 60
set_field_back(3X) CHAR 11: 29
set_field_buffer(3X) CHAR 11: 31
set_field_fore(3X) CHAR 11: 29
set_field_init(3X) CHAR 11: 77
set_field_just(3X) CHAR 11: 27
set_field_opts(3X) CHAR 11: 38
set_field_pad(3X) CHAR 11: 29
set_field_status(3X) CHAR 11: 33
set_field_term(3X) CHAR 11: 77
set_field_type(3X) CHAR 11: 20-24,
26-27
set_fieldtype_arg(3X) CHAR 11: 98
set_fieldtype_choice(3X) CHAR
11: 102
set_field_userptr(3X) CHAR 11: 35
set_form_fields(3X) CHAR 11: 47
set_form_init(3X) CHAR 11: 77
set_form_opts(3X) CHAR 11: 88
set_form_page(3X) CHAR 11: 83
set_form_sub(3X) CHAR 11: 53
set_form_term(3X) CHAR 11: 78
set_form_userptr(3X) CHAR 11: 86
set_form_win(3X) CHAR 11: 53
setgid MG A: 22
setgrent MG A: 22
set-group identification SAG 12: 25-28
set_item_init(3X) CHAR 10: 56
set_item_opts(3X) CHAR 10: 14
set_item_term(3X) CHAR 10: 56
set_item_userptr(3X) CHAR 10: 18
set_item_value(3X) CHAR 10: 11
setjmp MG A: 22
setjmp(3) BSD; BSD/XNX 1: 33
setjmp(3C) PRM
setjmp.h header file BSD/XNX 1: 35
setkey MG A: 28
setlocale(3C) CTG32, 34; PRM
set_max_field(3X) CHAR 11: 15
set_menu_back(3X) CHAR 10: 38
set_menu_fore(3X) CHAR 10: 38
set_menu_format(3X) CHAR 10: 27
set_menu_grey(3X) CHAR 10: 38
set_menu_init(3X) CHAR 10: 56
set_menu_items(3X) CHAR 10: 23
set_menu_mark(3X) CHAR 10: 31
set_menu_opts(3X) CHAR 10: 70
set_menu_pad(3X) CHAR 10: 38
set_menu_pattern(3X) CHAR 10: 65
set_menu_sub(3X) CHAR 10: 34
set_menu_term(3X) CHAR 10: 56
set_menu_userptr(3X) CHAR 10: 67
set_menu_win(3X) CHAR 10: 34
setmnt MG A: 10
setmnt(1M) SARM
set_new_page(3X) CHAR 11: 44
setpgid(2) PRM
setpgrp MG A: 22
setpgrp(2) PRM
setprocset macro SS 5: 26

- setpwent MG A: 22
 setregid(3) BSD; BSD/XNX 1: 32
 setreuid(3) BSD; BSD/XNX 1: 32
 setsid(2) PRM
 setsockopt function NI 3: 51, 55
 setstate(3) BSD/XNX 1: 27
 setterm MG A: 22
 settime MG A: 10
 settime(1) XNX
 settimeofday function BSD/XNX 1: 32
 setting processor priority levels ISDG
 3: 25
 set_top_row(3X) CHAR 10: 62
 setuid MG A: 22
 setuid installation parameter SAG
 14: 14
 setuid(2) PRM
 setuname(1M) SAG 16: 20; SARM
 setup (see system setup)
 setup(1M) SARM
 login SAG 12: 18
 set-user identification SAG 12: 25-28
 setusershell(3) BSD/XNX 1: 27
 setutent MG A: 22
 setvbuf MG A: 22
 SFB (SCSI Function Block) SCSI 4: 13
 SFB_ABORTM SCSI 4: 13
 sfb(D4I) SCSI 4: 17
 SFB_FLUSHHR SCSI 4: 13
 SFB_NOPF SCSI 4: 13
 SFB_RESETM SCSI 4: 13
 SFB_RESUME SCSI 4: 13
 SFB_SUSPEND SCSI 4: 14
 SFB_TYPE SCSI 4: 3
 sf_comp_code SCSI 4: 13
 sfconvert(3) BSD/XNX 1: 28
 sf_dev SCSI 4: 13
 sf_func SCSI 4: 13
 sf_int SCSI 4: 13
 SFNOLIM parameter SAG 8: 75
 sfsys(4) SARM
 SFSZLIM parameter SAG 8: 74
 sf_wd SCSI 4: 13
 sgconvert(3) BSD/XNX 1: 28
 sgetl MG A: 22
 sh MG A: 10
 sh(1) UG 9: 1; URM
 append to a file UG 9: 16
 break command UG 9: 83-84
 case command UG 9: 79-83
 changed for Release 4.0 MG 5: 6
 command language UG 9: 2-37
 command substitution UG 9: 55-56
 comments UG 9: 59
 continue command UG 9: 84
 debugging UG 9: 84-88
 environment UG 9: 89-94
 exit command UG 9: 64
 export command UG 9: 82
 for command UG 9: 65
 "here document" UG 9: 59-63
 if command UG 9: 71-76
 in-line input UG 9: 59-63
 input redirection UG 9: 14-26
 kill process UG 9: 34-35
 loops UG 9: 65-71
 metacharacters UG 9: 4-9
 output redirection UG 9: 14-26
 pipes UG 9: 20
 positional parameters UG 9: 42-45,
 56-58
 pre-defined parameters UG 9: 46-49
 process control UG 9: 26-36
 process status UG 9: 32-34
 programming UG 9: 38-88
 quick reference UG F: 1-7

- quotes UG 9: 12–14
- read command UG 9: 51
- restart process UG 9: 35
- return codes UG 9: 64
- special characters UG 9: 4–14
- test command UG 9: 76–79
- user-defined variables UG 9: 49–50
- variables UG 9: 42–58, 92–94
- while command UG 9: 68
- shadow password SAG 12: 6
- shadow(4) SARM
- share(1M) NUAG 9: 1, 15: 1; SARM
 - display shared resources NUAG 9: 19, 15: 19
 - in dfstab file NUAG 14: 4
 - used to display shared NFS resources NUAG 16: 3
- shareall(1M) NUAG 9: 6, 15: 6; SARM
- shared memory SS 4: 63–87
 - changed for Release 4.0 MG 5: 23
 - control (shmctl) SS 4: 74–75
 - data structure SS 4: 64–67
 - get (shmget) SS 4: 67–70
 - identifier (shmid) SS 4: 64–67
 - operations (shmop) SS 4: 81–83
 - permissions SS 4: 68–69
 - shmctl example program SS 4: 75–81
 - shmget example program SS 4: 70–73
 - shmop example program SS 4: 83–87
 - tunable parameters SAG 8: 83–84
 - usage SS 4: 64–67
- shared objects CGP 2: 2–3, 11–32
 - creating CGP 2: 15–16, 23
 - guidelines for building CGP 2: 24–29
 - implementation CGP 2: 2–3, 21–23, 13: 30–38, 50–69
 - linking with CGP 2: 11–20, 31–32
- shared/sharing resources
 - as a set NUAG 9: 6, 15: 6
 - automatically NUAG 9: 4, 15: 4
 - browsing NUAG 9: 20, 15: 20
 - definition NUAG 7: 4
 - displaying a list of NUAG 9: 19–20, 15: 19–20
 - displaying resources mounted by remote systems NUAG 9: 22, 15: 22
 - explicitly NUAG 9: 1, 15: 1
 - setting access rights NUAG 9: 2, 15: 2
- sharetab(4) NUAG 7: 7; SARM
- sharing DMA channels ISDG 3: 21
- sharing interrupts ISDG 3: 21
- shell UG 1: 1, 5, 4: 8–11 (see also sh(1) or csh(1) or ksh(1))
- shell, changed for Release 4.0 MG 5: 6
- shell scripts UG 1: 5, 4: 10, 9: 38–88
 - creating in FACE UG 5: 94
 - running in FACE UG C: 24
- shell(1F) CHAR 2: 41, , A: 4, C: 1
- shl MG A: 10
- shl(1) URM
- SHLBMAX parameter SAG 8: 74
- shmat MG A: 22
- shmat(2) SS 4: 81–83
- shmctl MG A: 22
- shmctl(2) PRM; SS 4: 74–81
 - example program SS 4: 75–81
 - usage SS 4: 74–75
- shmdt MG A: 22
- shmdt(2) SS 4: 81–83
- shmget MG A: 22
- shmget(2) PRM; SS 4: 67–73
 - example program SS 4: 70–73
 - usage SS 4: 67–70
- SHMMAX parameter SAG 8: 83

SHMMIN parameter SAG 8: 83
SHMMNI parameter SAG 8: 83
shmop(2) PRM; SS 4: 81-87
 example program SS 4: 83-87
 usage SS 4: 81-83
SHMSEG parameter SAG 8: 83
short int CGP 3: 21, 31
show descriptor CHAR 2: 19, 22, 3: 11,
 43, A: 13
 example of CHAR 3: 26
show-path command (FACE) UG C: 6
shut down policy SAG 1: 13
shutdn MG A: 22
shutdown MG A: 10
shutdown file ISDG 3: 52
shutdown(1) BSD/XNX 1: 23
shutdown(1M) BSD; SAG 4: 39, 6: 21,
 25, 30, 35-37; SARM
shutdown(3N) NI
shV MG A: 28
sigaction(2) BSD/XNX 1: 33; PRM
SIGALRM alarm CHAR 2: 12
sigaltstack(2) PRM
sigblock(3) BSD; BSD/XNX 1: 33
SIGCHLD NI 3: 47
sigfpe(3) BSD
siginfo(5) PRM; SARM
SIGINT signal CHAR 2: 44
siginterrupt function BSD/XNX 1: 33
siginterrupt(3) BSD
signal MG A: 23
signal(2) PRM; STRM 6: 1
signal(3) BSD; BSD/XNX 1: 33
signal(5) PRM; SARM
signal(D3D) DDRM
signal.h header file BSD/XNX 1: 35
signals SS 2: 12-17
 and process groups NI 3: 45

BSD Compatibility Package
 BSD/XNX 1: 33
 changed for Release 4.0 MG 5: 28
 code blocking SS 2: 16
 extended STRM 6: 8
 handlers SS 2: 14-16
 in job control management STRM
 6: 11
 in STREAMS STRM 6: 7
 sending SS 2: 16
 stacks SS 2: 17
 types SS 2: 13-14
signed CGP 3: 2, 21, 31-33
sigpause(3) BSD; BSD/XNX 1: 33
sigpending(2) PRM
sigprocmask(2) PRM
sigsem MG A: 23
sigsem(2) XNX
sigsend(2) PRM
sigsetjmp(3C) PRM
sigsetmask(3) BSD; BSD/XNX 1: 33
sigsetops(3C) PRM
sigstack(3) BSD
sigsuspend(2) PRM
sigvec(3) BSD; BSD/XNX 1: 33
Simple Mail Transfer Protocol (see
 mail(1))
sin MG A: 23
single_to_decimal(3) BSD/XNX 1: 28
single-user state SAG 6: 15, 21, 16: 5
sinh MG A: 23
sinh(3M) PRM
size MG A: 13
size(1) CGP 1: 12, 2: 22; PRM
size(D2DK) DDRM
sizeof operator CGP 3: 36
slave driver
 in pseudo-tty subsystem STRM
 12: 15

open STRM 12: 22
 slave server (YP) NI 11: 3-6, 31-32
 set with ypinit NI 11: 18-21
 start daemons NI 11: 21
 sleep MG A: 10, 23
 sleep() call ISDG 3: 5
 sleep function ISDG 3: 23
 sleep priorities ISDG 3: 28
 sleep(1) URM
 sleep(3) BSD; BSD/XNX 1: 30
 sleep(3C) PRM
 sleep(D3DK) DDRM
 slink(1M) NUAG; SARM
 slk_bar descriptor CHAR 2: 28, 4: 8,
 A: 11
 slk_clear(3X) CHAR 12: 4
 slk_init(3X) CHAR 12: 4
 slk_layout descriptor CHAR 2: 27,
 4: 11
 slk_noutrefresh(3X) CHAR 12: 4
 slk_refresh(3X) CHAR 12: 4
 slk_restore(3X) CHAR 12: 4
 SLKs (see screen-labeled function
 keys)
 slk_set(3X) CHAR 12: 4
 slk_text descriptor CHAR 2: 28, 4: 8,
 A: 11
 sm_cont SCSI 4: 11
 sm_len SCSI 4: 11
 sm_lun SCSI 4: 11
 sm_op SCSI 4: 11
 sm_pad0 SCSI 4: 11
 sm_res1 SCSI 4: 11
 sm_res2 SCSI 4: 11
 SMTP (see mail(1))
 smtp(1M) SARM
 smtpd(1M) SARM
 smtpqer(1M) SARM

smtpsched(1M) SAG F: 9; SARM
 socket NI 3: 1, 4: 1
 changed for Release 4.0 MG 5: 33
 closing NI 3: 10
 connectionless NI 3: 13
 conversion to TLI NI 4: 1-23
 creation NI 3: 3
 datagram NI 3: 2, 14
 differences between BSD and UNIX
 System V NI 4: 1
 failure NI 3: 4
 flags NI 3: 9
 non-blocking NI 3: 43
 options NI 3: 55
 raw NI 3: 3
 stream NI 3: 2
 table of differences in implementa-
 tions NI 4: 18-23
 table of TLI equivalents NI 4: 16
 types NI 3: 2
 socket(3N) NI
 socket-based datagrams NI 4: 9-11
 socket-based services SAG 13: 56
 socketpair(3N) NI
 sockio(7) NI
 soelim(1) BSD; BSD/XNX 1: 16
 SO_FLAG, in M_SETOPTS STRM B: 9-13
 soft mounts NUAG 17: 5
 software applications packaging (see
 package)
 software latencies SS 5: 33-34
 software management
 shell interface SAG 14: 1
 summary of commands SAG 14: 37
 sysadm(1M) interface SAG 14: 1
 terminology SAG 14: 3-5
 sort MG A: 10
 sort(1) UG 3: 57, 62-64, 9: 19; URM

- sortbib(1) BSD
 Source Code Control System (see SCCS)
 space installation parameter SAG 14: 15
 space(4) ISDG 2: 15–16; SS 8: 17–18
 space.c file ISDG 3: 41, 50
 SPAU (System Performance Analysis Utilities) SAG 8: 9–11
 special characters UG GL: 12
 FMLI CHAR 2: 5–6
 sh(1) UG 9: 4–14
 special file UG 1: 11, 3: 2, GL: 12
 definition ISDG 3: 2
 special files ISDG 3: 12, 51
 spell checker (FACE) UG 5: 89, C: 21
 spell(1) UG 9: 17, 19; URM
 spl ISDG 3: 8
 spl(D3D) DDRM
 spline MG A: 13
 split MG A: 10
 split(1) URM
 splN() functions ISDG 3: 25
 spray(1M) NI; NUAG
 spray(3N) NI
 sprayd(1M) NI; NUAG
 sprintf MG A: 23
 SPTMAP parameter SAG 8: 73
 sputl MG A: 23
 sputl(3X) PRM
 sqrt MG A: 23
 srand MG A: 23
 srand48 MG A: 23
 srandom(3) BSD/XNX 1: 27
 srchtxt(1) URM
 SRPC_DOMAIN parameter SAG 8: 70
 srv(D2DK) DDRM
 ss_addr SCSI 4: 11
 sscanf MG A: 23
 ss_cont SCSI 4: 11
 ssignal MG A: 23
 ssignal(3C) PRM
 ss_len SCSI 4: 11
 ss_lun SCSI 4: 11
 ss_op SCSI 4: 11
 SSTKLIM parameter SAG 8: 75
 stackuse MG A: 13
 /stand
 disk partition SAG 6: 5–12, A: 3
 system SAG 8: 52
 unix SAG 8: 52
 standard error UG GL: 13
 standard input UG 9: 14, GL: 13
 standard output UG 9: 14, GL: 13
 standend MG A: 23
 standend(3X) CHAR 7: 25
 standout MG A: 23
 standout(3X) CHAR 7: 25
 STARLAN NI 5: 4, 8
 support for protocols NI 12: 13
 start_color(3X) CHAR 7: 32
 start(D2D) DDRM
 start-of-day counts SAG 8: 3
 startup information SAG 1: 3
 stat MG A: 23
 stat(2) PRM; XNX
 stat(4) XNX
 stat(5) PRM; SARM
 statd(1M) NUAG
 statd(3N) NUAG 20: 2, 5
 state transitions NI 2: 14–15, 61–68
 process SS 5: 31
 transport interface NI 2: 65–68
 statement operators (FMLI) CHAR 2: 8
 statements CGP 3: 47–51
 static CGP 3: 25–26

static linking CGP 2: 2–3, 10–32
 implementation CGP 2: 2–3, 21
statvfs(2) PRM
stdarg(5) PRM
stdio MG A: 23
stdio(3S) PRM
stdio.h header file BSD/XNX 1: 35;
 CGP 2: 33, 51
stdipc(3C) PRM
sticky bit SAG 12: 23
stime MG A: 23
stime(2) PRM
storage device (see device)
storage duration CGP 3: 25–26
store MG A: 23
store(3) BSD/XNX 1: 26
str(3G) PRM
strace(1M) SARM; STRM
straddr.so file NI 12: 14–15
strapush structure STRM E: 8
strategy(D2DK) DDRM
strbuf structure STRM 5: 10
strcascmp(3) BSD/XNX 1: 28
strcat MG A: 23
strccpy(3G) PRM
strcf(4) NUAG; SARM
strchg(1) STRM 7: 29; URM
strchr MG A: 23
strclean(1M) SARM; STRM
strcmp MG A: 23
strcmp function CGP 2: 35–37
strcoll(3C) PRM
strconf command STRM 7: 29
strcpy MG A: 23
strcspn MG A: 23
STRCTLSZ parameter SAG 8: 76; STRM
 E: 6
strdup MG A: 23

Stream
 controlling terminal STRM 6: 12
 definition STRM 2: 1
 hung-up STRM 6: 12
Stream construction STRM 3: 3–11
 add/remove modules STRM 3: 10
 close a Stream STRM 3: 11
 define module/driver STRM 3: 5
 example STRM 3: 11–16
 open a Stream STRM 3: 5
 queue structures STRM 3: 3
Stream head
 definition STRM 2: 1
 processing control STRM 5: 12
stream sockets NI 3: 2
streamio(7) STRM
STREAMS
 3B2 configuration STRM G: 1
 asynchronous feature NI 2: 51
 basic operations STRM 2: 5–8
 benefits STRM 2: 20–24
 changed for Release 4.0 MG 5: 21
 components STRM 2: 9–14
 configuration STRM E: 1–9
 definition STRM 2: 1
 header files STRM 7: 48
 input/output mechanism NI 2: 4, 48
 listen(1M) support SAG 13: 56
 manual pages STRM F: 1–2
 master.d STRM E: 3
 multiplexing STRM 2: 15–19
 system call interface NI 2: 52
 system calls STRM 2: 5, 3: 1
 tunable parameters STRM E: 5–6
STREAMS Administrative Driver
 STRM E: 7–9
STREAMS data structures STRM
 A: 1–13

- design STRM 7: 47
- dynamic allocation STRM 7: 47
- STREAMS debugging STRM D: 1–19
 - crash(1M) STRM D: 2–6
 - dump module STRM D: 6–17
 - error and trace logging STRM D: 17–19
- STREAMS driver STRM 2: 14, 9: 6–8
 - accessible functions STRM 7: 49
 - cloning STRM 9: 18
 - close routine design STRM 7: 43
 - declarations STRM 7: 2
 - definition STRM 2: 1
 - design guidelines STRM 7: 42–50, 9: 30
 - environment STRM 7: 1
 - flow control STRM 9: 16
 - flush handling STRM 9: 12
 - interface STRM 7: 40
 - ioctl(2) STRM 7: 9–30
 - loop-around STRM 9: 20–29
 - open routine design STRM 7: 43
 - printer driver example STRM 9: 9–16
 - pseudo-tty STRM 12: 22–25
 - pseudo-tty subsystem master STRM 12: 15
 - pseudo-tty subsystem slave STRM 12: 15
- STREAMS entry points DDRM 2: 1–3
- STREAMS functions DDRM 3: 2–3
- STREAMS message queues STRM 2: 10
 - priority STRM 2: 11–12
- STREAMS module STRM 2: 12–14, 8: 1–9
 - accessible functions STRM 7: 49
 - autopush facility STRM E: 6, 9
 - close routine design STRM 7: 43
 - connld(7) STRM 11: 12
 - control information STRM 2: 1
 - declarations STRM 7: 2
 - definition STRM 2: 1
 - design guidelines STRM 7: 42–50, 8: 13
 - environment STRM 7: 1
 - filter STRM 8: 5
 - flow control STRM 8: 10–12
 - ioctl(2) STRM 7: 9–30
 - line discipline STRM 12: 3
 - null module example STRM 7: 6
 - open routine design STRM 7: 43
 - ptem(7) STRM 12: 17
 - read-side put procedure STRM 8: 1
 - routines STRM 8: 1–5
 - service interface example STRM 5: 47–53
 - service procedure STRM 8: 3
 - status information STRM 2: 1
 - write-side put procedure STRM 8: 3
- STREAMS queue
 - definition STRM 2: 3
 - equeue structure STRM 5: 22
 - flags STRM 5: 21
 - overview STRM 2: 9
 - qband structure STRM 5: 22
 - queue structure STRM 5: 19
 - structures STRM A: 1–6
 - using equeue information STRM 5: 24
 - using qband information STRM 5: 24
 - using queue information STRM 5: 21
- STREAMS scheduler, in service procedure STRM 4: 2
- STREAMS structures DDRM 4: 1–2
- STREAMS tunable parameters SAG 8: 76–77

- STREAMS utility routines STRM
 - C: 1-21
- STREAMS-based console driver
 - description STRM G: 10-14
 - open STRM G: 12
 - read-side processing STRM G: 12
 - read-side service procedure STRM G: 13
 - write-side processing STRM G: 13
- STREAMS-based pipe STRM 2: 1
 - atomic write STRM 11: 5
 - basic operations STRM 11: 1-6
 - creation STRM 3: 8
 - creation errors STRM 11: 2
 - definition STRM 11: 1
 - PIPE_BUF STRM 11: 5
- STREAMS-based ports driver
 - close STRM G: 5
 - description STRM G: 2-9
 - interrupt procedure STRM G: 8
 - ioctl STRM G: 6
 - open STRM G: 4
 - read-side service procedure STRM G: 9
 - write-side put procedure STRM G: 5
 - write-side service procedure STRM G: 7
- STREAMS-based pseudo-terminal
 - subsystem (see pseudo-tty subsystem)
- STREAMS-based sxt driver STRM
 - 12: 3
- STREAMS-based terminal subsystem
 - (see tty subsystem)
- STREAMS-based xt driver
 - close control channel STRM G: 26
 - close normal window STRM G: 26
 - data flow STRM G: 26
 - description STRM G: 15-34
 - encoded transmission STRM G: 33
 - error checking STRM G: 33
 - flow control STRM G: 32
 - open control channel STRM G: 24
 - open normal window STRM G: 25
 - scanning STRM G: 33
 - set up STRM G: 15
 - streamio(7) STRM G: 31
 - termio(7) STRM G: 30
 - windowing ioctl STRM G: 30
 - X type ioctl STRM G: 31
- streamtab(D4DK) DDRM
- strerr(1M) SARM; STRM
- strerror(3C) PRM
- strftime(3C) PRM
- strftime(4) PRM; SARM
- string(3) BSD
- string(3C) PRM
- string.h CGP 2: 35 (see also header files)
- strings MG A: 13
 - and string functions, awk(1) UG 10: 24
 - constants CGP 3: 9
 - literals CGP 3: 9
 - routines CGP 2: 35-37, 40-41
- strings(1) URM
- strings.h header file BSD/XNX 1: 34
- strioc1 structure STRM 3: 15, A: 12
- strip MG A: 13
- strip(1) CGP 1: 12, 6: 3; PRM
- strlen MG A: 23
- strlog(D3DK) DDRM
- STRMSGSZ parameter SAG 8: 76; STRM E: 5
- strncasecmp(3) BSD/XNX 1: 28
- strncat MG A: 23

- strncmp** MG A: 23
strncpy MG A: 23
stroptions structure STRM A: 13
strpbrk MG A: 23
strqget(D3DK) DDRM
strqset(D3DK) DDRM
strchr MG A: 23
strspn MG A: 23
STRTHRESH parameter SAG 8: 77;
 STRM E: 6
strtod MG A: 23
strtod(3C) PRM
strtok MG A: 23
strtol MG A: 23
strtol(3C) PRM
strtty structure STRM G: 3
struct.h header file BSD/XNX 1: 34
structure (struct)
 declaration CGP 3: 22–23
 initialization CGP 3: 44–45
strxfrm(3C) PRM
stty MG A: 10
stty(1) BSD; BSD/XNX 1: 25; CHAR
 1: 11, A: 9; SAG 13: 52; UG 2: 16,
 9: 90–91; URM
 with LP SAG 9: 123–124
sttydefs(1M) SAG 13: 49–51; SARM
stune file ISDG 3: 41
stune(4) SARM
su MG A: 10
su(1M) SAG 12: 3; SARM; URM
su(4) SARM
subdirectory UG 3: 2
subnets
 number NUAG 2: 3
 setting up NUAG 4: 17
subwin MG A: 24
subwin(3X) CHAR 8: 9

su_log system log SAG 12: 3, 17
sulogin MG A: 10
sulogin(1M) SARM
sum MG A: 10
sum(1) BSD; BSD/XNX 1: 17; URM
sunfloatingpoint.h header file
 BSD/XNX 1: 34
sunfp.h header file BSD/XNX 1: 34
suspending files (FACE) UG 5: 96–97
svc_create function NI 7: 2
svc_dg_create function NI 7: 4
svc_reg function NI 7: 4
svc_tli_create function NI 7: 3
svc_tp_create function NI 7: 3
svc_unreg function NI 7: 4
svc_vc_create function NI 7: 4
sv_machine SCSI 4: 15
SVMMLIM parameter SAG 8: 75
sv_modes SCSI 4: 15
SVR4 (see UNIX System V Release 4)
sv_release SCSI 4: 15
swab MG A: 24
swab(3C) PRM
swap disk partitions SAG A: 3
 changed for Release 4.0 MG 5: 23
swap space, increase SAG 15: 24
swap(1M) SARM
swapadd MG A: 29
swapctl MG A: 10
swapctl(2) PRM
switch statement CGP 3: 48–49
sxt(7) SARM; STRM
symbolic debugger (see **sdb(1)**)
symbolic links SS 2: 23–24, 6: 1; UG
 1: 12, 3: 2
 access SS 6: 8
 and pre-SVR4.0 systems SS 6: 11
 changed for Release 4.0 MG 5: 12

- content of SS 6: 1
- copy SS 6: 8-9
- create SS 6: 3, 6-7
- definition of SS 6: 1
- examples of creating SS 6: 7-8
- link SS 6: 9-10
- looping with SS 6: 6
- move SS 6: 10
- properties of SS 6: 3-5
- referenced file SS 6: 3
- remove SS 6: 3, 8
- uses of SS 6: 1-2
 - with RFS SS 6: 12-14
- symlink(2) PRM; SS 6: 6
- sync MG A: 10, 24
- sync(1M) SARM; URM
- sync(2) PRM
- synchronization SS 7: 3
- synchronous input/output, in polling STRM 6: 1
- synchronous mode NI 4: 14-15
- syntax
 - command line UG 1: 6-9
 - errors (FMLI) CHAR 2: 9
- sys login SAG 12: 19, 16: 19
- SYS parameter SAG 8: 70
- sysadm(1M) SAG C: 1-2; SARM
 - creating and changing menus SAG D: 15-20
 - customizing SAG D: 1-5
 - diagnostics SAG 4: 2
 - express mode SAG C: 35
 - forms SAG C: 27-28, 30, D: 18-20
 - frames SAG C: 4
 - function keys SAG C: 26
 - interface hierarchy SS 9: 6
 - interface modification SS 9: 1
 - login SAG 12: 18
 - main menu SAG C: 37
 - messages SAG C: 9
 - tasks SAG C: 9, 27, 33, D: 21-26
 - testing menus SAG D: 18
 - used to administer NFS NUAG D: 1-5
 - writing help messages SAG D: 6-14
- sysadmin MG A: 10
- sysadmsh MG A: 28
- syscall(3) BSD; BSD/XNX 1: 32
- syscon STRM G: 10
- sysconf(3C) PRM
 - _sysconfig NI 13: 21, 23-24, 26
 - _sysconfig file (SAF) SAG 13: 7-8, 18, 20, 23
- sysdump command SAG 4: 28
- sys_errlist MG A: 24
- sysexits.h header file BSD/XNX 1: 34
- Sysfiles file (BNU) SAG 7: 57
 - setup SAG 7: 23
- sysfs(2) PRM
- sysi86(2) PRM
- sysinfo(2) PRM
- sysinit entries (in inittab file) SAG 6: 17
- syslog(3) BSD
- syslog.conf(4) BSD
- syslogd(1M) BSD
- SYS_NAME parameter SAG 8: 77
- sys_nerr MG A: 24
- system MG A: 24
- system buffers ISDG 3: 3
- system calls CGP 2: 43-44; SS 2: 2-18
 - advanced IO SS 2: 4
 - basic IO SS 2: 3
 - error handling SS 2: 2-3
 - IPC SS 4: 1-87
 - list file system control SS 2: 12

list IPC SS 2: 10
list memory management SS 2: 11
list miscellaneous SS 2: 18
signals SS 2: 12-17
terminal IO SS 2: 5
user processes SS 2: 5-10
XENIX compatibility BSD/XNX 1: 6-7
system configuration SAG 8: 52-62
system dump ISDG 3: 69; SAG 4: 27
system errors NI 4: 15
system file ISDG 3: 14, 41
system file ISDG 3: 50
system files SAG 8: 52
system history log (BNU) SAG 7: 66
system initialization SAG 6: 15, 17
system log SAG 1: 13
system logins SAG 12: 19
system name SAG 16: 20-21
 display SAG 6: 40
system panics ISDG 3: 68
system profile SAG 17: 22
system reconfiguration SAG 8: 54
 sample SAG 8: 57
system restore SAG 11: 17-24
system scheduler class SAG 10: 5; SS
 5: 5
system security SAG 17: 5
system setup
 overview SAG 16: 4-5
 shell interface SAG 16: 1-3
 sysadm(1M) interface SAG 16: 1-3
 testing SAG 16: 12-13
system state SAG 6: 13-26, 16: 5
 changing SAG 6: 18, 20
 default SAG 6: 15
 definitions SAG 6: 15
 directories SAG 6: 25
 files SAG 6: 26

firmware SAG 6: 24
firmware state SAG 4: 7
multi-user SAG 6: 15, 22
reboot SAG 6: 24
RFS SAG 6: 23
single-user SAG 6: 15, 21
System Performance Analysis Utilities
 (SPAU) SAG 8: 9-11
system(3S) PRM
system(4) SAG 6: 7
Systems file (BNU) SAG 7: 42-47
 and cu command UG 12: 19-20
 and uucp command UG 12: 5
systty STRM G: 10

T

tabs(1) URM
t_accept(3N) NI
taddr2uaddr function NI 5: 10, 12: 17
tail MG A: 10
tail(1) URM
talk(1) NUAG; URM
talkd(1M) NUAG; SARM
t_alloc(3N) NI
TAM CHAR D: 1-11
 compiling and running CHAR D: 2
 how the library works CHAR D: 4-11
 library CHAR 6: 6
 tips for polishing programs CHAR
 D: 3
 transition keyboard subsystem
 CHAR D: 7
 translation of calls CHAR D: 4
tam(3X) CHAR
tan MG A: 24
tanh MG A: 24

- tape MG A: 28
- tar MG A: 10
- tar(1) URM
- Target controller reset SCSI 1: 9
- target driver SCSI 4: 17
- task
 - action file SS 9: 11
 - change entry SS 9: 28
 - create entry SS 9: 27
 - definition form SS 9: 30
 - delete entry SS 9: 33
- t_bind(3N) NI
- tbl(1) BSD; BSD/XNX 1: 16
- t_buf structure STRM G: 3
- t_close(3N) NI
- t_connect(3N) NI
- tcopy(1) BSD; BSD/XNX 1: 23
- TCP (Transmission Control Protocol)
 - NI 12: 14, 19; NUAG 1: 3, 7
- tcp transport type NI 5: 9
- TCP(7) NI; NUAG; SARM
- TCP/IP NI 5: 4, 8-9; NUAG 1: 3-6; SAG F: 8
 - administration NUAG 1: 10
 - application layer NUAG 1: 8-9
 - changed for Release 4.0 MG 5: 34
 - data link layer NUAG 1: 6
 - network addressing NI 12: 14
 - network layer NUAG 1: 7
 - physical layer NUAG 1: 6
 - protocol layers NUAG 1: 4-9
 - transport layer NUAG 1: 7-8
- tcpip.so file NI 12: 14
- tcsetpgrp(3C) PRM
- tdelete MG A: 24
- tee MG A: 10
- tee(1) URM
- telinit MG A: 10
- tellmdir MG A: 24
- telnet(1) NUAG 1: 8, 24: 5, 25: 7; URM
 - escape character NUAG 24: 7
- telnetd(1M) NUAG; SARM
- tempnam MG A: 24
- TERM environment variable CHAR 6: 7;
 - SAG 17: 28; UG 3: 36, 9: 93, G: 1
- term(4) CHAR; SARM
- term(5) CHAR; SARM
- termcap CHAR 13: 11, 19-20
- termcap MG A: 24
- term.h CHAR 13: 4
- terminal
 - call (see ct)
 - configuration UG 2: 2, 9: 90-91
 - keyboard UG 2: 3
 - options UG 2: 2, 9: 90-91
 - special keys UG 2: 6-8
 - tty subsystem changes with Release 4.0 MG 5: 21
 - type UG 2: 2, 3: 36, 5: 2-3, G: 1-3
- Terminal Access Method (see TAM)
- terminal display attributes
 - reset CHAR 2: 47
 - table of CHAR 2: 47
 - using CHAR 2: 47-49
- terminal independence (FMLI) CHAR 1: 10-11
- terminal line settings SAG 13: 46
 - hunt sequence SAG 13: 51
 - status SAG 13: 49
 - stty(1) SAG 13: 52
- terminfo MG A: 24
- TERMINFO environment variable SAG 17: 27
- terminfo(4) CHAR 13: 1-20; SARM
 - basic capabilities CHAR 13: 13
 - compare descriptions CHAR 13: 19

- compile and run a program CHAR
 13: 5
 compile description CHAR 13: 17
 convert from `termcap` CHAR 13: 19
 database CHAR 1: 10, 13: 9–20, A: 1
 example program CHAR 13: 5
 keyboard-entered capabilities CHAR
 13: 15
 learn capabilities CHAR 13: 10
 name terminal CHAR 13: 9
 parameter string capabilities CHAR
 13: 15
 print descriptions CHAR 13: 19
 routines CHAR 13: 3–8
 screen-oriented capabilities CHAR
 13: 14
 specify capabilities CHAR 13: 11
 terminal descriptions CHAR 13: 9
 test description CHAR 13: 18
 use by LP UG 8: 16
 use with LP print service SAG
 9: 124–127
`termio(7)` SAG 13: 47; STRM 6: 11
 default flag values STRM 12: 3
 xt driver STRM G: 30
`termios(2)` PRM
`termiox(7)` SARM
 support STRM 12: 12
`t_error(3N)` NI
`test` MG A: 10
`test(1)` BSD; BSD/XNX 1: 17; UG
 9: 76–79; URM
`test(1F)` CHAR 2: 41, , C: 1
`testb(D3DK)` DDRM
 text descriptor CHAR 2: 24, 26, 3: 61,
 4: 4
 text editing UG 4: 3–7 (see also `ed(1)`
 and `vi(1)`)
 command mode UG 4: 4
 input mode UG 4: 4
 text frames (FACE)
 editing UG 5: 35
 function keys UG 5: 35
 navigation UG 5: 35
 text frames (FMLI)
 definition CHAR 1: 9
 descriptor definitions CHAR 3: 56–61
 editing CHAR 1: 20
 examples of CHAR 3: 62–66
 graphic characters in CHAR 2: 47
 header CHAR A: 14
 lists of descriptors CHAR 2: 23–25
 navigation in CHAR 1: 20
 order of descriptors CHAR 3: 56
 positioning CHAR 3: 57
 scrollable CHAR 1: 9
 text segment CGP 2: 2, 22–25, 27 (see
 also object files)
 TEXT variable CHAR 2: 12
`tfind` MG A: 24
`t_free(3N)` NI
`tftp(1)` NUAG 1: 9, 25: 6; URM
`tftpd(1M)` NUAG; SARM
`tgetent` MG A: 24
`tgetflag` MG A: 24
`t_getinfo(3N)` NI
`tgetnum` MG A: 24
`t_getstate(3N)` NI
`tgetstr` MG A: 24
`tgoto` MG A: 24
`tic` MG A: 10
`tic(1M)` CHAR; SARM
 ticks, definition ISDG 3: 6
`ticlts(7)` NI
`tid` MG A: 28
`time` MG A: 13, 24

- time of day SAG 16: 12, 15; UG 2: 17
- time slice, real-time process SS 5: 20
- time(1) URM
- time(2) PRM
- timeout () call ISDG 3: 6
- timeout () function ISDG 3: 31
- timeout(D3DK) DDRM
- timers, high resolution and changes
 - for Release 4.0 MG 5: 29
- times MG A: 24
- times(2) PRM
- times(3) BSD/XNX 1: 30
- times(3C) BSD
- time-sharing
 - process priority SAG 10: 11–13
 - scheduler class SAG 10: 4; SS 5: 4
 - scheduler parameter table SAG 10: 5, 10–13; SS 5: 5
 - scheduler policy SAG 10: 10–13
- timex(1) SAG 8: 40; URM
- timezone(3) BSD/XNX 1: 28
- timezone(3C) BSD
- timezone(4) PRM; SARM
- timod(7) STRM
- tirdwr(7) NI 2: 47–50; STRM
- title descriptor CHAR 2: 24, 26, 3: 61, 4: 4
- title line (FACE) UG 5: 4
- TLI (Transport-Level Interface) (see transport interface)
- t_listen(3N) NI
- t_look(3N) NI
- tmpfile MG A: 24
- tmpfile(3S) PRM
- tmp_mnt, special directory NUAG 19: 2
- tmpnam MG A: 24
- tmpnam(3S) PRM
- tnamed(1M) NUAG; SARM
- toascii MG A: 24
- toggle descriptor CHAR 2: 27, 4: 11
- toggle CHAR 2: 35
- togslk command (FMLI) CHAR 2: 35, A: 15
- tokens CGP 3: 5–12; CTG15–19
 - preprocessing CGP 3: 11–12
- _tolower MG A: 24
- tolower MG A: 24
- t_open(3N) NI
- top_row(3X) CHAR 10: 62
- t_optmgmt(3N) NI
- tosmtp(1M) SARM
- touch MG A: 10
- touch(1) URM
- touchwin MG A: 24
- _toupper MG A: 24
- toupper MG A: 24
- tput MG A: 10
- tput(1) CHAR 1: 11; URM
- tputs MG A: 24
- tr MG A: 10
- tr(1) BSD; BSD/XNX 1: 14; URM
- trace driver ISDG 3: 68
- track (disk) SAG 4: 38
- transfer files (see send files)
- transfer log (BNU) SAG 7: 67
- transparent ioctl
 - M_COPYIN example STRM 7: 18–21
 - M_COPYOUT example STRM 7: 22–24
 - messages STRM 7: 17
 - processing STRM 7: 14–29
- transport endpoints NI 2: 9, 48, 52–60, 64, 69, 4: 1
 - polling multiple endpoints NI 2: 52, 54
- transport interface NI 4: 1, 9
 - datagrams NI 4: 11–13

- error handling NI 4: 15
 examples of NI 2: 71-87
 local management NI 2: 8-10, 16-23, 40-42
 local management common state table NI 2: 66
 states NI 2: 61
 transport layer, TCP/IP NUAG 1: 7-8
 transport provider NI 2: 6, 42, 45, 47-48, 52, 62, 69-70; NUAG 10: 7
 set name NUAG 11: 9, 12: 4
 transport user NI 2: 6, 65
 transport-level programming NI 2: 1-90
 t_rcv(3N) NI
 t_rcvconnect(3N) NI
 t_rcvdis(3N) NI
 t_rcvrel(3N) NI
 t_rcvudata(3N) NI
 t_rcvuderr(3N) NI
 trig(3M) PRM
 trigraph sequences CGP 3: 11; CTG15
 troff(1) BSD; BSD/XNX 1: 16
 trouble report SAG 17: 33
 trpt(1M) NUAG; SARM
 true MG A: 10
 true(1) URM
 truncate(3C) PRM
 truss(1) URM
 ts_dptbl(4) SAG 10: 10-13, 17; SARM
 TSDU (transport service data unit) NI 2: 32-33, 69
 tsearch MG A: 24
 tsearch(3C) PRM
 tset MG A: 10
 tset(1) BSD; BSD/XNX 1: 14; XNX
 ts_kmdpris (kernel-mode scheduler parameter table) SAG 10: 13-14
 TSMAXUPRI parameter SAG 8: 79
 t_snd(3N) NI
 t_snddis(3N) NI
 t_sndrel(3N) NI
 t_sndudata(3N) NI
 tsort MG A: 13
 tsort(1) PRM
 t_sync(3N) NI
 ttcompat(7) SARM
 tty MG A: 10
 tty subsystem
 benefits STRM 12: 1
 changed for Release 4.0 MG 5: 21
 description STRM 12: 1-14
 hardware emulation module STRM 12: 13-14
 ldterm(7) STRM 12: 3
 setup STRM 12: 2
 tty(1) URM
 tty(7) SARM
 ttyadm(1M) SAG 13: 11, 17, 35; SARM
 ttydefs SAG 13: 2, 32, 46-49, 54
 add records SAG 13: 50-51
 remove records SAG 13: 52
 ttydefs(4) NI
 ttymon NI 13: 1
 ttymon(1M) SAG 6: 16, 13: 1, 10, 17, 31-35; SARM
 add port monitor SAG 13: 38-39
 add service SAG 13: 39-40
 and ps(1) SAG 13: 44-45
 and Service Access Facility SAG 13: 33-36
 and who(1) SAG 13: 43-44
 configuration files SAG 13: 42
 debugging SAG 13: 45
 default configuration SAG 13: 35
 disable service SAG 13: 41

enable service SAG 13: 40-41
express mode SAG 13: 42, 45
log file SAG 13: 45
port monitor status SAG 13: 36
port status SAG 13: 37-38
remove port monitor SAG 13: 39
remove service SAG 13: 40
service status SAG 13: 37
ttyname MG A: 24
ttyname(3C) PRM
ttyslot MG A: 24
ttyslot(3C) PRM
ttysrch(4) SARM
tunable parameters SAG 8: 3, 63-84
 kernel SAG 8: 70-71
 log driver SAG 8: 79
 message SAG 8: 81-82
 modifying SAG 8: 55
 paging SAG 8: 72-73
 RFS NUAG 12: 79-83
 scheduler SAG 10: 8-9
 semaphores SAG 8: 82-83
 shared memory SAG 8: 83-84
 STREAMS SAG 8: 76-77
 XENIX compatibility BSD/XNX 1: 2
tunable system parameters
 defining ISDG 3: 61
 enabling ISDG 3: 61
 files containing ISDG 3: 60
 modifying ISDG 3: 60
t_unbind(3N) NI
tunefs(1M) SARM
twalk MG A: 24
type casts (FMLI) CHAR 2: 3-5
 descriptor evaluation time CHAR
 2: 4-5
 file names CHAR 2: 2-4
type conversions CGP 3: 2, 31-33

type qualifiers CGP 3: 22; CTG20-23
typedef CGP 3: 26, 29
types CGP 3: 21-24, 27-30
 compatible and composite
 CTG44-47
 incomplete CTG40-43
types(5) PRM
typing conventions UG 2: 4-9
TZ Environment Variable ISDG F: 4
tzset MG A: 25

U

uaddr2taddr function NI 5: 10, 12: 18
uadmin MG A: 25
uadmin(1M) SARM
uadmin(2) PRM
ualarm(3) BSD; BSD/XNX 1: 27
u-block, real-time process SS 5: 34
ucontext(5) PRM
udp transport type NI 5: 9
UDP (User Datagram Protocol) NI
 5: 8-9; NUAG 1: 3, 7
UDP(7) NI; NUAG; SARM
UDSP ISDG 3: 57
ufsdump(1M) SARM
ufs-FSType SAG 5: 12-18
 boot block SAG 5: 14
 cylinder group map SAG 5: 12
 disk block addresses SAG 5: 16
 free blocks SAG 5: 18
 indirect addressing SAG 5: 16
 inodes SAG 5: 15
 storage blocks SAG 5: 17
 summary information block SAG
 5: 12
 super block SAG 5: 15

- UFSNINODE parameter SAG 8: 80
 ufs/quota.h header file BSD/XNX
 1: 35
 ufsrestore(1M) SARM
 uio(D4DK) DDRM
 uiomove(D3DK) DDRM
 ul(1) BSD; BSD/XNX 1: 16
 ulimit MG A: 25
 ulimit(2) PRM
 umask MG A: 10, 25
 umask(1) SAG 12: 24, 17: 28; URM
 umask(2) PRM
 umount MG A: 10, 25
 umount(1M) NUAG 9: 15, 15: 15; SAG
 6: 15
 umount(2) PRM
 umountall command NUAG 9: 18,
 15: 18; SAG 4: 41
 uname MG A: 10, 25
 uname(1) SAG 6: 40, 16: 20–21; UG
 11: 7–9; URM
 uname(2) PRM
 unbootable operating system
 recovery SAG 8: 60
 unbufcall(D3DK) DDRM
 unctrl MG A: 25
 #undef CGP 3: 14
 undefined symbols CGP 2: 10
 undelete command (FACE) UG 5: 55,
 C: 12
 underlining attribute CHAR 2: 47
 unget MG A: 13
 unget(1) CGP 10: 12, 17; PRM
 ungetc MG A: 25
 ungetc(3S) PRM
 undef(1) BSD; BSD/XNX 1: 19
 union
 declaration CGP 3: 23
 initialization CGP 3: 44
 uniq MG A: 10
 uniq(1) URM
 unique connection (STREAMS) STRM
 11: 11–13
 unistd(4) SARM
 unistd.h(4) BSD/XNX 1: 35
 unit numbers ISDG 3: 17
 units MG A: 10
 units(1) URM
 universal address NI 5: 3–4, 6, 10–11,
 13, 9: 26, 28–30
 unix SAG 6: 7–12
 configuring SAG 8: 52–62
 UNIX System
 accessing from FACE UG C: 24
 accessing from FMLI CHAR 1: 24–25
 authentication NUAG 18: 2
 files SS 6: 1
 general description UG 1: 1–9
 interrupting commands CHAR A: 4
 UNIX system account UG 2: 10
 UNIX System V
 feature evolution MG 3: 1–5, 4: 1–14
 history MG 2: 1
 Release 2.0 features MG 3: 2–3
 Release 2.1 features MG 3: 4–5
 Release 3.0 features MG 4: 2–6
 Release 3.1 features MG 4: 7–10
 Release 3.2 features MG 4: 11–14
 Release 4 features MG 5: 1–45
 UNIX System V Release 4
 changes (from Release 2) MG 3: 1–5
 changes (from Release 3) MG 4: 1–14
 new features MG 5: 1–45
 unix-system command
 FACE UG C: 24
 FMLI CHAR 2: 35

- UNIX-to-UNIX copy (see uucp)
- UNIX-to-UNIX execution (see uux)
- unlink MG A: 25
- unlink(2) PRM
- unlinkb(D3DK) DDRM
- unlockpt(3C) STRM 12: 26
 - with pseudo-tty driver STRM 12: 22
- unmountfsys login SAG 12: 18
- unpack MG A: 10
- unpost_form(3X) CHAR 11: 56
- unpost_menu(3X) CHAR 10: 40
- unset(1F) CHAR 2: 41, C: 1
- unshare(1M) NUAG 9: 5, 15: 5; SARM
- unshareall(1M) NUAG 9: 8, 15: 8
- unsigned CGP 3: 2, 21, 31-33
- untimeout(D3DK) DDRM
- update command
 - FACE UG C: 15
 - FMLI CHAR 2: 36, A: 13-14
- update driver software package ISDG 3: 57
- updaters(4) NI
- upgrade to SVR4 (see UNIX System V Release 4)
- upper multiplexor STRM 2: 17
- upper Stream STRM 2: 15
- upstream STRM 2: 3
- uptime(1) BSD; BSD/XNX 1: 24
- ureadc(D3DK) DDRM
- use_incorrect_pre4.0_behavior
 - descriptor CHAR 2: 13, 4: 12, A: 10
- user
 - adding SAG 17: 8
 - communication SAG 17: 32
 - ID UG 3: 57
 - identification bits SAG 12: 25-28
 - list information SAG 17: 20
 - permissions UG 3: 50
 - profile SAG 12: 3, 17: 25
 - removing SAG 17: 14
 - requests SAG 17: 33
 - user interface ISDG 3: 39
 - user interfaces, changed for Release 4.0 MG 5: 38
 - user priority SS 5: 7
 - user privileges ISDG 3: 39
 - useracc(D3DK) DDRM
 - useradd(1M) SARM
 - user-defined variables (FMLI) CHAR 2: 10-11
 - userdel(1M) SARM
 - usermod(1M) SARM
 - users(1) BSD; BSD/XNX 1: 25
 - usleep(3) BSD; BSD/XNX 1: 27
 - /usr
 - ccs/lib CGP 2: 12-13
 - directories SAG B: 25-27
 - disk partition SAG A: 3
 - file system changed for Release 4.0 MG 5: 16
 - files SAG B: 28-29
 - include CGP 2: 8-9
 - lib CGP 2: 12, 19-20
 - ustat MG A: 25
 - ustat(2) PRM
 - utilities (FMLI), built-in CHAR 2: 39-41, A: 17, C: 1
 - utime MG A: 25
 - utime(2) PRM
 - utimes(3) BSD; BSD/XNX 1: 32
 - utmp(4) PRM; SAG 13: 11; SARM
 - utmpname MG A: 25
 - utmpx(4) PRM; SARM
 - uuccheck(1M) SARM
 - uucico MG A: 11
 - uucico(1M) SAG 7: 17, 19; SARM; UG 12: 5-6

dial-up password SAG 12: 10-12
 uuclean MG A: 11
 uucleanup(1M) SARM
 uucp MG A: 11
 uucp login SAG 16: 19
 uucp(1C) UG 12: 2-7; URM
 dial-up password SAG 12: 10-12
 login SAG 12: 19
 logins SAG 7: 19, 24
 logs (see BNU logs)
 network administration (see BNU)
 uucppublic directory UG 12: 12-15
 uuencode(1C) URM
 uuglist(1C) URM
 uuinstall MG A: 28
 uulog MG A: 11
 uuname MG A: 11
 uuname(1) UG 11: 7-8
 uupick MG A: 11
 uupick(1C) UG 12: 12-15
 uusched(1M) SAG 7: 20; SARM; UG
 12: 6
 uustat MG A: 11
 uustat(1C) UG 12: 10-12; URM
 uusub MG A: 28
 uuto MG A: 11
 uuto(1C) UG 12: 8-10; URM
 Uutry(1M) SARM
 uux MG A: 11
 uux(1C) UG 12: 24-26; URM
 uuxqt(1M) SAG 7: 17, 20; SARM
 uwritec(D3DK) DDRM

V

vacation(1) BSD; BSD/XNX 1: 22; UG
 11: 14; URM

val MG A: 13
 val(1) CGP 10: 12, 35; PRM
 valid descriptor CHAR 2: 22, 3: 43,
 A: 2-3, 13
 value descriptor CHAR 2: 22, 3: 44,
 A: 13
 values(5) PRM
 /var
 changed for Release 4.0 MG 5: 18
 directories SAG B: 30-33
 disk partition SAG A: 3
 files SAG B: 34-37
 mail SAG F: 3
 spool/smtpq/LOG SAG F: 9
 varargs MG A: 25
 varargs(5) CTG3; PRM
 variables (FMLI) CHAR 2: 10-14
 built-in CHAR 2: 11-12
 const cast CHAR 2: 4
 evaluation of CHAR 2: 12-14
 notation CHAR 2: 13
 vary cast CHAR 2: 4
 vc(1) CGP 10: 12, 36; PRM
 vedit MG A: 11
 VER parameter SAG 8: 70
 verifiers, in RPC authentication NUAG
 18: 2
 ver_no(D4I) SCSI 4: 17
 Version number (of host adapter)
 SCSI 2: 4, 4: 15
 Version number structure SCSI 4: 15
 vfork(2) PRM
 vfprintf MG A: 25
 VFS
 architecture SS 6: 2
 changed for Release 4.0 MG 5: 9
 vfstab(4) NUAG 9: 13, 14: 7, 15: 13,
 17: 6, 18: 8; SAG 5: 30; SARM

vi MG A: 11
vi(1) UG 7: 1-3; URM
 add text UG 7: 16-17, 50-54
 change text UG 7: 63-69
 command mode UG 7: 8
 copy text UG 7: 71-74
 create text UG 7: 7-9
 cursor movement UG 7: 10-14, 22-38
 delete text UG 7: 15-16, 56-61, 81
 edit multiple files UG 7: 87
 environment UG 7: 5
 global substitute UG 7: 82-83
 input mode UG 7: 8
 join lines UG 7: 76
 line addressing UG 7: 81
 line editing mode UG 7: 79
 line numbers UG 7: 42-43
 move around in file UG 7: 39-47
 move text UG 7: 70
 patterns UG 7: 43-47
 quick reference UG E: 1-9
 quit UG 7: 18-20, 84-86
 read file UG 7: 81
 read-only mode UG 7: 88
 recover from hangup UG 7: 87
 redraw screen UG 7: 77
 repeat command UG 7: 76
 scroll through text UG 7: 39-41
 search for character UG 7: 24-26
 search for text UG 7: 43-47
 shell escape UG 7: 79-80
 substitute text UG 7: 63-69, 82-83
 terminal configuration UG 7: 4-5
 transpose characters UG 7: 70
 undo previous command UG
 7: 57-58
 upper-case, lower-case change UG
 7: 77

 write text to a file UG 7: 80
 yank text UG 7: 71-74
view MG A: 11
virtual addressing CGP 13: 47-53
virtual memory SS 7: 1-3 (see also
 memory management)
virtual system console (see syscon)
visible transport type NI 5: 9
VM (virtual memory) (see memory
 management)
vmstat MG A: 28
void CGP 3: 21
volatile CGP 3: 22; CTG20-23, 45
volcopy(1M) SARM
 restore service SAG 11: 5
vprintf MG A: 25
vprintf(3S) PRM
vsh MG A: 28
vsig(1F) CHAR, C: 1
vsprintf MG A: 25
VTOC (Volume Table of Contents)
 SAG 4: 42, 15: 10-11, 15-18
 diskette SAG 15: 10
 partitioning SAG 15: 10-12
 SCSI SAG 15: 11
vtop(D3D) DDRM

W

w(1) BSD; BSD/XNX 1: 25
waddch MG A: 25
waddstr MG A: 25
wait MG A: 11, 25
wait(1) URM
wait(2) PRM
wait(3) BSD
wait3(3) BSD/XNX 1: 32

- waitid(2) PRM
 waitpid(2) PRM
 waitsem MG A: 25
 waitsem(2) XNX
 wakeup() call ISDG 3: 5
 wakeup() function ISDG 3: 23
 wakeup(D3DK) DDRM
 wall MG A: 11
 wall(1M) SAG 6: 4, 20, 34, 17: 32; SARM
 WARNING error messages SAG E: 6-11
 wastebasket (FACE)
 deleting files and file folders UG
 5: 9, 53
 undeleting files and file folders UG
 5: 54
 wc MG A: 11
 wc(1) UG 3: 28, 47-49; URM
 wclear MG A: 25
 wclrtobot MG A: 25
 wclrtoeol MG A: 25
 wd(7) SARM
 wdelch MG A: 25
 wdeleteln MG A: 25
 weak symbols CGP 2: 29-30, 13: 26-27
 werase MG A: 25
 wgetch MG A: 25
 wgetstr MG A: 25
 what MG A: 11
 what(1) CGP 10: 12, 33-34; PRM
 whatis(1) BSD
 which(1) BSD; BSD/XNX 1: 18
 while statement CGP 3: 49
 who MG A: 11
 who(1) SAG 6: 41; UG 2: 17, 9: 44; URM
 whoami(1) BSD; BSD/XNX 1: 25
 whodo MG A: 11
 whodo(1M) SARM
 whois(1) NUAG; URM
 wide character constants CGP 3: 9;
 CTG26-27
 wide characters CGP 3: 9; CTG25-27
 wide string literals CTG26-27
 widget SS 1: 11
 wildcard address NI 3: 48
 winch MG A: 25
 windowing terminal commands
 STRM G: 28
 windows CHAR 8: 1-13; UG G: 5-9
 create CHAR 8: 8
 output and input CHAR 8: 2
 window_text descriptor CHAR 2: 28,
 4: 8
 winsch MG A: 25
 winsertln MG A: 25
 wmove MG A: 25
 wnoutrefresh(3X) CHAR 8: 3
 work area (FACE) UG 5: 5
 work area (FMLI) CHAR 1: 4
 working descriptor CHAR 2: 26, 4: 5-6
 working icon CHAR 1: 3, 4: 6
 WORM (write-once-read-many) dev-
 ice SAG 15: 5
 wprintw MG A: 25
 wrap descriptor CHAR 2: 22, 24, 3: 45,
 61, A: 14
 wrapping (FMLI)
 navigation keys CHAR B: 1-6
 word CHAR 3: 45, 61, 63
 WR(D3DK) DDRM
 wrefresh MG A: 25
 write MG A: 11, 26
 write lock SS 3: 2, 4, 9
 write(1) URM
 write(2) NI 2: 47-48; PRM
 write(D2DK) DDRM
 write-once-read-many (WORM) dev-
 ice SAG 15: 5

write-side
 console driver processing STRM
 G: 13
 definition STRM 2: 3
 ldterm(7) STRM 12: 7
 ports driver put procedure STRM
 G: 5
 ports driver service procedure
 STRM G: 7
 put procedure STRM 8: 3
 xt driver STRM G: 28
 wscanw MG A: 26
 wstandend MG A: 26
 wstandout MG A: 26
 wstat(5) PRM
 wtinit(1M) SARM

X

X11/NeWS, changed for Release 4.0
 MG 5: 39
 xargs MG A: 11
 xargs(1) URM
 XDR (External Data Representation)
 NI 5: 3, 6, 6: 1-3, 12, 14, 8: 1
 array, fixed length NI 8: 12
 array, variable length NI 8: 13
 basic block size NI 8: 1
 block size NI 8: 1
 boolean NI 8: 5
 changed for Release 4.0 MG 5: 36
 constant NI 8: 17
 data, optional NI 8: 18
 data types NI 8: 3
 discriminated union NI 8: 15
 double-precision floating-point
 integer NI 8: 8

enumeration NI 8: 5
 fixed-length array NI 8: 12
 fixed-length opaque data NI 8: 9
 floating-point integer NI 8: 7
 hyper integer NI 8: 6
 integer NI 8: 3
 integer, double-precision floating
 point NI 8: 8
 integer, floating point NI 8: 7
 integer, hyper NI 8: 6
 integer, unsigned NI 8: 4
 language NI 5: 3, 6: 30, 8: 20
 language, notation NI 8: 20
 language, syntax NI 8: 21, 23
 memory allocation NI 7: 69
 opaque data, fixed length NI 8: 9
 opaque data, variable length NI
 8: 10
 optional data NI 8: 18
 string NI 8: 11
 structure NI 8: 14
 typedef NI 8: 17
 unsigned integer NI 8: 4
 variable-length array NI 8: 13
 variable-length opaque data NI 8: 10
 void NI 8: 16
 xdr(3N) NI
 xdr_admin(3N) NI
 xdr_complex(3N) NI
 xdr_create(3N) NI
 xdr_simple(3N) NI
 XENIX, shared data SAG 8: 78
 XENIX compatibility BSD/XNX 1: 1-7
 backup(1) BSD/XNX 1: 4
 file management BSD/XNX 1: 4, 6
 file operations unified in Release
 4.0 MG 5: 8
 installing XENIX BSD/XNX 1: 4

IPC BSD/XNX 1: 6
 master.d/kernel file BSD/XNX 1: 1
 master.d/xnamfs file BSD/XNX 1: 3
 PATH environment variable
 BSD/XNX 1: 1
 process management BSD/XNX 1: 6
 running programs on System V
 BSD/XNX 1: 1
 semaphores BSD/XNX 1: 3
 shared data parameters BSD/XNX
 1: 2
 shell utilities BSD/XNX 1: 5
 system calls BSD/XNX 1: 6-7
 terminal management BSD/XNX 1: 5
 timezone parameters BSD/XNX 1: 2
 tunable parameters BSD/XNX 1: 2
 xferstats (transfer) log (BNU) SAG
 7: 67
 xfsck(1M) SARM
 xinstall MG A: 11
 xinstall(1M) SARM; XNX
 xlist MG A: 26
 xref MG A: 13
 xrestore MG A: 11
 xrestore(1M) SARM
 XSDSEGS parameter SAG 8: 78
 XENIX compatibility BSD/XNX 1: 2
 XSDSLOTS parameter SAG 8: 78
 XENIX compatibility BSD/XNX 1: 2
 xstr MG A: 13
 xt(7) SARM; STRM
 xt_chan structure STRM G: 23
 xtctl structure STRM G: 22
 xterm XWIN terminal emulator CHAR
 A: 10, 19
 xt_msg structure STRM G: 24
 xtproto(5) PRM; SARM
 xts(1M) SARM

xtt(1M) SARM
 XWIN SS 1: 11
 changed for Release 4.0 MG 5: 39

Y

y0 MG A: 26
 y1 MG A: 26
 yacc MG A: 13
 yacc(1) CGP 1: 10, 12: 1-55; PRM; SS
 1: 7
 definitions CGP 12: 9-11
 disambiguating rules CGP 12: 17-27
 error handling CGP 12: 28-30
 how to write source CGP 12: 4-9
 library CGP 11: 24, 12: 31-32
 parser actions CGP 12: 12-16
 routines CGP 12: 37
 source format CGP 12: 4
 symbols CGP 12: 4-9
 typing CGP 12: 39-40
 usage examples CGP 12: 41-55
 use with lex(1) CGP 11: 17, 22-24,
 12: 1-4, 9-11, 31-32
 use with sdb(1) CGP 12: 32
 yylex() CGP 12: 31
 yyparse() CGP 12: 31-32
 Yellow Pages Service (see YP)
 yes MG A: 11
 yes(1) XNX
 yn MG A: 26
 YP (Yellow Pages Service)
 binding NI 11: 5-6
 client NI 11: 5, 21-22
 debugging NI 11: 37-46
 domain NI 11: 3, 7-8
 machine types NI 11: 3

- maps (see maps (YP))
- master server NI 11: 15-17
- servers NI 11: 3-6, 31-32
- slave server NI 11: 18-21
- steps to implement NI 11: 7-22
- turning off NI 11: 46
- ypbind(1M) NI 11: 2-6
- ypcat(1) NI
- ypclnt(3N) NI
- ypfiles(4) NI
- ypinit(1M) NI 11: 15-22
- ypmake(1M) NI
- ypmatch(1) NI
- yppoll(1M) NI
- yppush(1M) NI
- ypserv(1M) NI 11: 2-6
- ypset(1M) NI
- ypupdate(3N) NI
- ypupdated(1M) NI
- ypwhich(1) NI
- ypxfr(1M) NI 11: 27-29

Z

- zdump(1M) SARM
- zero(7) SARM; SS 7: 8
- zic(1M) SARM



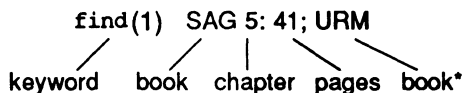
INDEX REFERENCE CARD

INDEX REFERENCE CARD

USING THE MASTER INDEX

SUBJECT INDEX

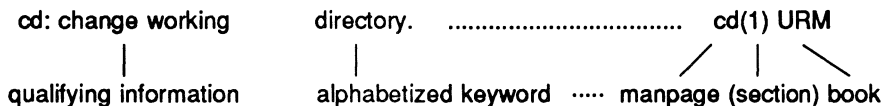
EXAMPLE:



* Some books hold alphabetized manual pages; chapters and page numbers do not apply. When a book acronym is given without chapter and page numbers, it means that you can find a manual page for the index entry in the book section given in parentheses. In the example above, look for the **find** page in section (1) of the URM.

PERMUTED INDEX

EXAMPLE:



The first two columns of the permuted index are built with phrases taken from the NAME line on each manual page. Look in the middle column for the word you are interested in, then read the complete phrase by starting with the manual page name, which may appear in the left or middle column; long phrases wrap around to the beginning of the left column. See page 3-1 for more details.

BOOK ACRONYMS

BSD	BSD Compatibility Guide
CGP	Programmer's Guide: ANSI C and Programming Support Tools
CHAR	Programmer's Guide: Character User Interface
CTG	ANSI C Transition Guide
DDRM	Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual
ISDG	Integrated Software Development Guide
MBRMAN	MULTIBUS Reference Manual
MG	Migration Guide
MOUSE	Mouse Driver Administrator's Guide
NI	Programmer's Guide: Networking Interfaces
NUAG	Network User's and Administrator's Guide
PRM	Programmer's Reference Manual
PSX	Programmer's Guide: POSIX Conformance
SAG	System Administrator's Guide
SARM	System Administrator's Reference Manual
SCSI	Programmer's Guide: SCSI Driver Interface
SS	Programmer's Guide: System Services and Application Packaging Tools
STRM	Programmer's Guide: STREAMS
TAI	Transport Application Interface Guide
UG	User's Guide
URM	User's Reference Manual
XNX	XENIX Compatibility Guide

320-708

**UNIX
PRESS**

A Prentice Hall Title

ISBN 0-13-957515-4