

p-System™

Supplement for the

Corvus Concept Workstation

SofTech Microsystems, Inc.
San Diego, California

January, 1984

1-240.41.A

Copyright ©1983, 1984 by SofTech Microsystems Inc.

All rights reserved. No part of this work may be reproduced in any form or by any means or used to make a derivative work (such as a translation, transformation, or adaptation) without the written permission of SofTech Microsystems, Inc.

p-System is a trademark of SofTech Microsystems Inc.

UCSD and UCSD Pascal are registered trademarks of the Regents of the University of California. Use thereof in conjunction with any goods or services is authorized by specific license only, and any unauthorized use is contrary to the laws of the State of California.

Printed in the United States of America.

Disclaimer

This document and the software it describes are subject to change without notice. No warranty, expressed or implied covers their use. Neither the manufacturer nor the seller is responsible or liable for any consequences of their use.

T A B L E
O F
C O N T E N T S

INTRODUCTION 1
HARDWARE CONFIGURATIONS 3
BRINGING UP THE p-SYSTEM 5
 Backing Up the System 5
 Booting the p-System 7
 Keyboard Considerations 9
 Extended Memory and Setup Factors 10
FLOATING POINT CONFIGURATION 11
TURTLEGRAPHICS CONFIGURATION 15

NATIVE CODE PROCEDURES 17
DISK FILE CONFIGURATIONS 20

INTRODUCTION

This document provides a brief introduction to the p-System Version IV.1 as implemented on the Corvus Concept Workstation. It is intended to accompany the standard p-System documentation, which includes the Operating System Reference Manual, and Personal Computing with the UCSD p-System.

Chapter 1 of the Operating System Reference Manual gives an overview of the system. Personal Computing with the UCSD p-System is an excellent general introduction to the use of the system.

Personal Computing with the UCSD p-System will help you become familiar with the Filer and Screen-Oriented Editor. A full description of these programs is given in Chapters 4 and 5 of the Operating System Reference Manual.

The use of UCSD Pascal is further described in the manual The UCSD Pascal Handbook by Randy Clark and Stephen Koehler. This manual serves as a useful reference guide for users of UCSD Pascal and the p-System.

On the Corvus Concept Workstation, the p-System runs in conjunction with the Corvus Concept Operating System (CCOS). The CCOS must be booted and operational before the p-System can be booted. However, once the p-System is booted, the CCOS is used only for low level I/O support. It will not be visible while the p-System is running.

Familiarity with the CCOS is required in order to install the p-System and create an appropriate environment for it. The following Corvus Concept documents will be helpful for the installation of the p-System: The Corvus Concept, Personal Workstation User Guide and The Corvus Concept, System Managers Guide.

The specific CCOS utilities used to install and use the p-System are the Constellation II Volume Manager to create p-System volumes on the Winchester Disk, the Access Manager to mount volumes, the File Manager to copy files from one volume to another, and the System Utilities Manager to assign a device driver to the printer.

HARDWARE CONFIGURATIONS

The p-System as implemented on the Corvus Concept Workstation requires the following minimum hardware configuration:

- Corvus Concept Workstation, which includes:
 - 256K bytes of main memory
 - Video Display Unit
 - Keyboard
- Floppy Disk Drive
- Corvus Winchester Disk Drive

In addition to this minimum configuration, the p-System can support additional main memory and peripherals. Since the Corvus Concept Operating System is used to provide low level I/O support, most peripherals that it supports can be used with the p-System. Following is the maximum configuration that the p-System will support, with an indication of the p-System unit number used to access the devices:

- Up to six disk volumes, which can be chosen from any Corvus volume that has a p-System (type 1) directory. The disk volumes are assigned p-System device numbers 4, 5, and 9 through 12. The RAM disk volume is available on systems with 512K bytes of main memory.
- A console port accessed as CONSOLE: and SYSTEM: (p-System devices 1 and 2, respectively).

- A printer port accessed as PRINTER: (p-System device 6).
- A remote, serial port accessed as REMIN: and REMOUT: (p-System devices 7 and 8, respectively).

BRINGING UP THE p-SYSTEM

The following sections describe a step by step approach to bringing the p-System up for the first time on the Corvus Concept Workstation. The important activities described are how to backup the system disks, how to install the system, and how to boot the system. Also, specific information about keyboard use is given.

There will already be some p-System files present on the Corvus Concept Workstation. These existing files are meant for executing applications only, and should be ignored for the purposes of this manual.

Backing Up the System

The first thing you should do is backup the p-System disks. The easiest way to do that is to copy the disks onto the Winchester Disk. The Winchester Disk copy should then be used as the working version and the master disks should be saved as a backup copies.

The system release consists of two master disks: SYSTEM and a compiler disk, either PASCAL or FORTRAN. The following steps should be followed to copy a master disk to the Winchester Disk:

1. Use the CCOS Constellation II Volume Manager to create a volume. It is recommended that the volume created for the SYSTEM disk be 1024 blocks in size. The volumes created for the other disks need only be 512 blocks in size. Be certain it is a p-System volume.
2. Use the CCOS Constellation II Access Manager to mount the volume you just created.
3. Put the master disk to be copied in the floppy disk drive.
4. Use the CCOS Utility File Manager to copy the entire contents of the master disk to the newly created Winchester Disk volume. Use the wild card feature of the file manager to copy the entire contents of the master disk. For example, to copy the SYSTEM master disk to the Winchester Disk volume PSYSTEM, do the following after entering the File Manager:

Press the CopyFile command key

Enter /SYSTEM/* <return>

Enter /PSYSTEM <return>

Refer to the Corvus Concept Documentation for details on using the CCOS utilities.

Throughout this document the names of the master disks are used in the instructions for setting up the system. You should substitute the name of the appropriate backup volume when you follow these instructions. The master disks should not be used, because a mistake in setting up the system could render some of the master disks unusable.

Booting the p-System

To boot the p-System, you must first boot the CCOS and establish the appropriate environment for running the p-System. After that has been done, simply press the PSYSTEM function key on the primary function key level. This will cause the p-System bootstrap program to be loaded and executed. When it is complete, the p-System will be operational, with its main command line at the top of the screen. To terminate the p-System, type "H" (for Halt) and you will be returned to the CCOS. The following steps should be followed to boot the p-System:

1. If you want to use a printer with the p-System, you must assign a device driver to the printer and set the printer's parameters. This is done using the CCOS system utilities. See The Corvus Concept, Personal Workstation User Guide for instructions on the use of the CCOS system utilities.

2. Type the name of the volume where you copied SYSTEM and then the function key [SetVol]. This is the boot volume, and it must at least contain the following files:

```
SYSTEM.INTERP
SYSTEM.PASCAL
SYSTEM.MISCINFO
```

3. Press the [PSYSTEM] function key.
4. All volumes that you have the option to mount are displayed on the screen. If there is enough memory, the RAM disk option will be available. If you did not mount the PRINTER: driver, there will be a warning message. If you have access to more than five p-System volumes, you will be prompted for which volumes to mount. If you have access to five or less volumes, all volumes will be mounted. If you make a mistake during this process, press the <esc> key and you will exit to the CCOS Dispatcher.

If you have a system with 512K bytes of main memory, one of the six p-System units will be a RAM disk volume. In this case, you may enter up to five other volumes. The RAM disk volume will be assigned the p-System unit number five. The RAM disk volume acts as a normal disk drive, but it is much faster because it is in memory and it is temporary, e.g., it will be erased upon leaving the p-System.

5. The p-System bootstrap process will now begin. The system will be loaded into memory and initialized. When the p-System comes up, it will use the character set and screen window that was in effect at the start of the boot process. If you wish to change either of these, use the CCOS System Manager Utility or Window Manager Utility to do so.

Keyboard Considerations

There are several keys and control characters which are frequently used when running the p-System. The following table identifies the special keys and control characters and their function on the Corvus Concept:

CONTROL-C	=	ETX, Editor accept, and CONSOLE end of file
CONTROL-F	=	Flush
CONTROL-R	=	alpha-lock
CONTROL-S	=	Stop CONSOLE output
CONTROL-Q	=	Resume CONSOLE output
CONTROL-Z	=	Editor character insert key
CONTROL-X	=	Editor character delete key
ESCAPE	=	System escape key (must be pressed twice)

With the exception of stop and resume CONSOLE output, the p-System's Setup utility may be used to change the keys used to perform the above functions. See Chapter 5 of the Operating System Reference Manual on using the Setup utility.

Extended Memory and Setup Factors

The p-System, as shipped on the release disks, is configured as an extended memory system. When it is booted, the system determines how much memory is available and sets the code pool base and code pool size accordingly. The Setup utility allows you to specify whether extended memory is to be used. If you desire to run the p-System as a non-extended memory system with an internal code pool, use the Setup utility to set the system parameter HAS EXTENDED MEMORY to FALSE. The Setup utility also lets you specify values for the code pool size and base address. However, the Corvus Concept implementation of the p-System ignores these values because it dynamically determines the best values for them.

FLOATING POINT CONFIGURATION

The system, as it is configured initially, supports 4-word floating point (real) numbers. It may be configured to support 2-word or 4-word real numbers or no real numbers at all. Four-word real numbers provide the most precision but also require the most data space for their representation. Two-word real numbers may be desired for applications where high precision is not required and program data space is in short supply. A system with no real number support at all may be configured.

To change the real number support, the following steps must be taken:

1. Install the correct version of REALOPS in the operating system file (SYSTEM.PASCAL); and
2. Select the p-machine emulator that provides the desired real number support and copy it to the boot disk as SYSTEM.INTERP.

These steps should only be done after the master disks have been backed up. A mistake in these steps could render them unusable.

Prior to changing the real number support, you should decide which level of real number support you want. If you choose 2-word reals, the files REALOPS2.CODE and 2WORD.INTERP should be used in the following steps. If you choose 4-word reals, the files REALOPS4.CODE and 4WORD.INTERP should be used in the following steps. If you choose no real number support at all, the file 0WORD.INTERP should be used in the following steps. Since the REALOPS unit of the operating system is not used in a no-reals system, you need not change the operating system if you select this level of real number support.

To install REALOPS in the operating system, the Library utility is used. You should make sure that there is enough room on the boot disk to create a new system (approximately 125 blocks). The following steps should be followed:

1. Set the current volume to PSYSTEM (the volume that is a copy of the SYSTEM disk) and boot the p-System.
2. Execute LIBRARY. This invokes the Library utility.
3. Respond "NEWSYS.CODE" to the prompt for the output file name.
4. Respond "SYSTEM.PASCAL" to the prompt for the input file name.

5. Selectively copy all units and segments, except the REALOPS unit, from the input file to the output file. This is done by entering "S" at the Library utility's command line then responding "Y" to the prompt to copy each unit or segment and "N" to the prompt to copy the REALOPS unit.
6. Enter "N" to get a new input file. Then enter the name of the REALOPS file you selected (prefixed with "SYSTEM:", i.e., "SYSTEM:REALOPS2.CODE").
7. Enter "T" to inhibit the copying of interface text to the output file.
8. Enter "E" to copy every segment.
9. Enter "Q" to quit the Library utility.
10. Enter <return> to the prompt for a copyright notice.
11. Execute the Filer by typing "F" from the main command line.
12. Enter "C" to use the C(hange command of the Filer.
13. Respond "NEWSYS.CODE" to the prompt requesting the name of the file to change.
14. Respond "SYSTEM.PASCAL" to the prompt requesting the new name of the file.

15. Respond "Y" to the question "DELETE OLD SYSTEM.PASCAL?".

There will now be a new operating system in the boot volume that contains support for real numbers and transcendental functions.

To install a new p-machine emulator with the correct real number module, the following steps should be performed:

1. Execute the Filer by typing "F" from the main command line.
2. Enter "T" to use the transfer command.
3. When the Filer prompts for the name of the file to transfer, enter the name of the p-machine emulator that contains the desired real number support prefixed with "SYSTEM:" (i.e., SYSTEM:4WORD.INTERP, SYSTEM:2WORD.INTERP, or SYSTEM:0WORD.INTERP).
4. When the Filer prompts for the name of the destination file, enter "SYSTEM.INTERP". Respond "Y" to the question "DELETE OLD SYSTEM.INTERP?".

There will now be a new p-machine emulator on the boot volume that contains the desired real number support.

Reboot the system to make the new real number system operational.

TURTLEGRAPHICS CONFIGURATION

The Turtlegraphics routines as initially shipped with the system use 4-word real numbers. If you change the real number support of the system, as described in the "Floating Point Configuration" section, you must also change the Turtlegraphics routines. If you use REALOPS2.CODE, you must use TURTLE2.CODE, and REALOPS4.CODE uses TURTLE4.CODE. To install Turtlegraphics in SYSTEM.LIBRARY, the Library utility is used. You should make sure there is enough room on the boot volume to create a new file (approximately 35 blocks). The following steps should be followed:

1. Set the current volume to PSYSTEM (the volume that is a copy of SYSTEM disk).
2. Execute Library. This invokes the Library utility.
3. Respond "NEWLIB.CODE" to the prompt for the output file name.
4. Respond "SYSTEM.LIBRARY" to the prompt for the input file name.
5. Selectively copy all units and segments, except the TURTLEGR unit, from the input file to the output file. This is done by entering "S" at the Library utility's command line then responding "Y" to the prompt to copy each unit or segment and "N" to the prompt to copy the TURTLEGR unit.

6. Enter "N" to get a new input file. Then enter the name of the Turtlegraphics file you selected (i.e., "TURTLE2.CODE" or "TURTLE4.CODE").
7. Enter "E" to copy every segment.
8. Enter "Q" to quit the Library utility.
9. Enter <return> to the prompt for a copyright notice.
10. Execute the Filer by typing "F" from the main command line.
11. Enter "C" to use the C(hange command of the Filer.
12. Respond "NEWLIB.CODE" to the prompt requesting the name of the file to change.
13. Respond "SYSTEM.LIBRARY" to the prompt requesting the new name of the file.
14. Respond "Y" to the question "DELETE OLD SYSTEM.LIBRARY?".

The correct Turtlegraphics routines will now be in the file "SYSTEM.LIBRARY".

Refer to the Program Development Reference Manual for directions on use of Turtlegraphics.

NATIVE CODE PROCEDURES

This section describes the conventions for writing assembly procedures to be called from Pascal or another high-level language. This section is only important if you are writing assembly code.

An assembly procedure is called via a JSR instruction, so it should expect a double word return address on the stack when it is called. Return to the host using an RTS instruction.

Registers A0-A2 and D0-D7 are available for use. Registers A3-A7 must be restored to the values at call-time if they are used.

Since pointers within the p-machine are byte offsets from a base register (A6), .PUBLIC references to Pascal variables will generate an offset, not the actual address, of the variable. To access an external variable, use this offset as a displacement from A6. For example:

```
ADDQ.W    #1,ABC(A6)
```

will increment the Pascal variable ABC.

A variable parameter is a p-machine pointer to the parameter, so it is also accessed as above. For example, a variable parameter may be accessed as follows:

```
MOVEQ    #0,D7          ; clear D7.L
MOVE.W   4(SP),D7       ; load the pointer (parameter)
ADDQ.W   #1,0(A6,D7.L) ; increment the variable
```

References to variables in other assembly procedures (via a .REF) may be accessed as above, provided the segment the procedures are in is located in the data area (i.e., it is not a RELPROC).

Here is a list of the register values available to the assembly procedure on entry.

- A2 - base of current segment
- A3 - base of PME
- A4 - p-machine program counter
- A6 - pointer to data area
- A7 - stack pointer

The symbol .INTERP (used to access items in the PME) is ignored. Instead, accesses should be made relative to A3 (the base of the p-machine). The following entry point is available to the assembly code programmer:

Routine	Offset	Parameters
XEQERR	04E	D0.W - execution error number

XEQERR may be used to cause an execution error to be recognized from assembly language. XEQERR should be jumped to, not called. Before jumping to XEQERR, the stack should be clear of all parameters (including the return word), and all registers should be restored. This routine is normally used for system work.

DISK FILE CONFIGURATIONS

The standard system is shipped to you on the SYSTEM disk. In the standard shipment, you will also get a Pascal or FORTRAN-77 compiler disk named PASCAL or FORTRAN, respectively. There are additional disks which are not part of the standard shipment but which can be purchased separately. They are PASCAL, FORTRAN, BASIC, ASM, and SFTCH. Following is a list of the files that are on each of these disks.

SYSTEM: {Standard Disk}

SYSTEM.INTERP
 SYSTEM.PASCAL
 SYSTEM.MISCINFO
 SYSTEM.FILER
 SYSTEM.EDITOR
 SYSTEM.LIBRARY
 TURTLE2.CODE
 TURTLE4.CODE
 SYSTEM.FONT
 LIBRARY.CODE
 DISKSIZE.CODE
 ABSWRITE.CODE
 COPYDUPDIR.CODE
 MARKDUPDIR.CODE
 PRINT.CODE
 QUICKSTART.CODE
 REALCONV.CODE
 RECOVER.CODE
 REMTALK.CODE
 SETUP.CODE
 REALOPS2.CODE
 REALOPS4.CODE
 0WORD.INTERP
 2WORD.INTERP
 4WORD.INTERP

DECODE.CODE
 XREF.CODE
 COMPRESS.CODE
 KERNEL.CODE
 COMMANDIO.CODE
 SCREENOPS.CODE
 DIR.INFO.CODE
 FILE.INFO.CODE
 SYS.INFO.CODE
 WILD.CODE
 68000.ERRORS
 68000.NCG.CODE
 68000.OPCODES
 CHKSUMOPS.CODE
 ERRORHANDL.CODE
 PATCH.CODE
 PEDGEN.CODE
 SYSTEM.ASSMBLER
 SYSTEM.LINKER

PASCAL: {Optional Disk}

PASCAL.COMPILER
 SYSTEM.SYNTAX

BASIC: {Optional Disk}

BASIC.R4.CODE
 BASIC.R2.CODE
 BLIB.R4.CODE
 BLIB.R2.CODE

FORTRAN: {Optional Disk}

FORTTRAN.R2.CODE
 FORTTRAN.R4.CODE
 FORTLIB.R2.CODE
 FORTLIB.R4.CODE

ASM: {Optional Disk containing
assemblers}

- 11.ERRORS
- Z80.ERRORS
- 6500.ERRORS
- 6809.ERRORS
- 8080.ERRORS
- 9900.ERRORS
- 11.OPCODES
- Z80.OPCODES
- 6500.OPCODES
- 6809.OPCODES
- 8080.OPCODES
- 9900.OPCODES
- ASM6809.CODE
- ASM8080.CODE
- ASM9900.CODE
- ASM11.CODE
- ASMZ80.CODE
- ASM6500.CODE
- 8086.ERRORS
- 8086.OPCODES
- 8087.FOPS
- ASM8086.IV.CODE
- 68000.ERRORS
- 68000.OPCODES
- ASM68000.CODE

SFTCH: {Optional Disk
containing SofTeach}

- TEST6A.CODE
- TEST6C.CODE
- TEST6D.CODE
- TEST6E.CODE
- TEST6B.CODE
- TEST6F.CODE
- TEST13A.CODE
- TEST13B.CODE
- TEST13C.CODE
- TEST13D.CODE
- TEST13E.CODE
- TEST14A.CODE
- TEST14B.CODE
- TEST14C.CODE
- TEST14D.CODE
- TEST14E.CODE
- TEST15A.CODE
- TEST15B.CODE
- TEST15C.CODE
- TEST15D.CODE
- TEST15E.CODE
- AQUIZ.CODE
- QUIZ6A.TEXT
- QUIZ6B.TEXT
- QUIZ6C.TEXT
- QUIZ6D.TEXT
- QUIZ6E.TEXT
- QUIZ6F.TEXT
- QUIZ13A.TEXT
- QUIZ13B.TEXT
- QUIZ13C.TEXT
- QUIZ13D.TEXT
- QUIZ13E.TEXT
- QUIZ14A.TEXT
- QUIZ14B.TEXT
- QUIZ14C.TEXT
- QUIZ14D.TEXT
- QUIZ14E.TEXT

QUIZ15A.TEXT
 QUIZ15B.TEXT
 QUIZ15C.TEXT
 QUIZ15D.TEXT
 QUIZ15E.TEXT
 QUIZ1.DATA
 QUIZ2.DATA
 QUIZ3.DATA
 QUIZ4.DATA
 QUIZ5.DATA
 QUIZ7.DATA
 QUIZ8.DATA
 QUIZ9.DATA
 QUIZ10.DATA
 QUIZ11.DATA
 USERLIB.TEXT
 ALT15C.CODE
 ALT154C.CODE

The remainder of this section discusses some strategies for configuring files on volumes. You may want to experiment with various ways to organize the system files on the volumes you use. This section describes the constraints that must be met, and makes a few useful suggestions.

There are certain system files which must be on the boot volume. Other system files may be placed on other volumes. In a standard configuration, the following system files (if they exist) must be on the boot volume:

SYSTEM.INTERP
 SYSTEM.PASCAL
 SYSTEM.MISCINFO
 SYSTEM.LIBRARY
 SYSTEM.WRK.TEXT
 SYSTEM.WRK.CODE
 SYSTEM.STARTUP
 SYSTEM.SYNTAX
 SYSTEM.FONT

Of these files, the ones which may cause space difficulties are the workfiles (SYSTEM.WRK.TEXT and SYSTEM.WRK.CODE). They are constantly being modified, expanded, created, and destroyed as new versions of a program or other text are produced. Since they must be stored on the boot volume, it is critical to leave as much room for them as possible. It is recommended that you allocate 1024 blocks for the volumes you create that are to be used as boot volumes. If this cannot be done, you can create extra work space on your boot volumes by moving the system files which do not have to be on the boot volume (SYSTEM.COMPILER, SYSTEM.EDITOR, and so forth) to another volume.