
Glossary

alphanumeric

Characters: letters (*alpha*) and numbers (*numeric*), including punctuation characters (such as `_` and `?`).

click

Depress and quickly release a mouse button; double- and triple-click imply depressing and releasing a mouse button two or three times, respectively, within a short period. You'll usually click with the first mouse button (which is the left mouse button for righthanded users—or the opposite if your mouse has been configured for a lefthanded user). See also *point*.

clipboard

A temporary storage area for X Window System programs, used for transferring text (“copying” and “pasting” text) between programs.

command

A command is an instruction that you can give to a program running on the Unix system. For instance, you can type a program's name and arguments on a command line, at a shell prompt; this command asks the shell to run that program. (The shell is a program itself; see *shell*.) Once a program starts running, it may accept commands of its own. For example, a text editor has commands for deleting and adding text to the file it's editing.

The terms *command* and *program* are used almost interchangeably, probably because the program name is typed first on a command line (at a shell prompt). Shells have some *built-in* commands that don't start a separate program running; one of these is `cd`, which changes the shell's working directory.

cracker

A malicious person who tries to break into computer systems (usually via a network), disrupt computers and networks, steal secrets (like passwords and credit card numbers), and other antisocial behavior.

Popular media often call these people *hackers*. But, to most computer programmers, a hacker is someone who enjoys computing and programming, and may be an expert at some area of it. (For instance, a *Perl hacker* is someone who's good at programming in the Perl language.)

desktop

The part of a display that's "behind" (not enclosed within) the windows, icons, and other items on the display. Also called the *root window*.

directory

A list of files and/or other directories. A directory is actually a special kind of file that has names and locations of other files and directories. See also *working directory*.

display

One meaning of *display* is to make something visible, as in "the command displays its result." In the X Window System, a *display* is the viewable area output by the X *display server*. Usually this is a single terminal screen, but X can be configured to use multiple screens as part of the same display.

Multiscreen X displays aren't common, though, and sentences like "the result is displayed on the display" are clumsy. To avoid confusion, we use the term *screen* for the visual output of your computer—whether it's an alphanumeric terminal or a graphical workstation. See also *screen*.

drag

As in *drag an object*, i.e., a window or an icon, means to point to the object and then depress and hold down (usually) the first mouse button while moving the pointer to a new location, where the mouse button is released.

Some Unix desktop environments support "drag and drop," which means dragging one object and dropping it over another object. For example, to print a file, you could drag the file's icon and drop it onto a printer icon.

Free Software Foundation (FSF)

An organization formed in 1985 that works for the right of computer users to study, copy, modify, and redistribute computer programs. The FSF also distributes free software. See <http://www.fsf.org/>; see also *GNU*.

GNOME

A project to develop a free desktop environment (a window system and more) for free operating systems. See <http://www.gnome.org/>; see also *KDE*.

GNU

A project, started in 1984, to develop a completely free Unix-like operating system: the GNU system. GNU stands for "GNU's Not Unix"; it is pronounced "guh-NEW." See also *Free Software Foundation*.

KDE

A desktop environment (a window system and more), as well as a family of application programs, for Unix-like workstations. See <http://www.kde.org/>; see also *GNOME*.

mouse pointer

The graphic symbol that appears on the output display and moves under the control of the mouse, trackball, or keyboard input to the window system.

In the X Window System, the pointer is actually called a *cursor*. But we use the term “pointer” in this book to distinguish the cursor under control of the mouse from other cursors that you’ll sometimes see (such as the “I-beam” cursor in an *xterm* window).

multitasking

An operating system that can run more than one program at a time is said to be a *multitasking OS*. The programs don’t actually all run simultaneously: the OS can divide the computer’s time between the different programs, very rapidly, so that they all *appear* to run at the same time. The system can still be overloaded, and run slowly, if too many programs are trying to run at once.

Unix has always been multitasking. MS-DOS (an early Microsoft OS) was not.

pathname

The location of a file or directory in a Unix filesystem: a series of names separated by slash (/) characters. Pathnames can be *absolute* (starting with a slash character, which means they begin at the filesystem’s root directory) or *relative* (not starting with a slash, which means the pathname starts from the current working directory). See also the section “The Unix Filesystem” in Chapter 3.

point

As in “point a mouse,” means to position the mouse pointer at a specified place or location within a window or other part of a window system display. See also *click*, *drag*.

program

A set of instructions to the computer, written by a programmer, and stored in a file. The program is executed when you type its name as the first word on a command line, at a shell prompt—or when you choose the program from a menu or icon in a window system. Unix runs a program as a *process*, which you can suspend or terminate using job control, an interrupt key, or the *kill* command.

root (user and directory)

Unix systems have an account named *root*, also called the “superuser,” that has no protections or restrictions. System administrators and staff use this account to make changes to the system’s configuration and operation.

A Unix filesystem is like an upside-down tree with a branching structure of directories inside directories. The first directory, where the filesystem starts, is called the *root directory*. Figure 3-1 is a filesystem diagram showing the root directory and a small part of a filesystem.

screen

The area of a terminal (usually glass or plastic) that shows computer output. See also *display* and *terminal*.

session

When two programs, or two users running programs, communicate across a network, they typically start the communication by doing a certain thing—for instance, by logging in. The communication continues until it's completed (or, possibly, aborted before it completes)—for instance, by logging out. The entire process, from start to completion, is called a *session*.

shell

A program that runs other programs. There are several different kinds of shells, each with its own command-line syntax; some of the most common are **bash**, **tsh** and **ksh**. All shells do the same basic job: reading commands that you type interactively at a shell prompt, or reading commands noninteractively from a program file called a *shell script*.

When you start using a terminal (by logging in) or a terminal window (by starting a program such as **xterm**), a shell program begins to run and prints a shell prompt. When you terminate that shell (by typing **exit** or **[CTRL-D]** at a prompt), you're logged out from that terminal; a terminal window will close.

syntax

The rules for, or the format of, the characters you use to make a command or other computer input. For example, the syntax of a Unix command line is explained in the section “Syntax of Unix Command Lines” of Chapter 1.

terminal

Computer hardware that provides a way to input data to, and display output from, an operating system and programs running under it. Usually the input hardware is a keyboard and the output is a glass or plastic screen. For the purposes of this book, there are two types of screens or terminals, *alphanumeric* and *graphical*.

An alphanumeric terminal can only display text, can't run a window system, and usually doesn't have a mouse or other pointing device.* An alphanumeric terminal displays alphanumeric characters—and possibly simple graphics (lines, boxes and maybe a few special symbols). An alphanumeric terminal can't handle a window system and typically doesn't have a mouse or other pointing device; if the cursor can be moved around the screen, it's probably done with arrows or other keys on the keyboard. See also *alphanumeric*.

* Before the widespread use of glass terminals (when data transmission rates were slow) it was common to use a teletype as both the input and output hardware. This is why Unix terminals are often called *tjys*.

A graphical terminal can usually run a window system—with arbitrary-sized windows, images (photographs and other graphics), sound, etc. Graphical terminals are typically *bitmapped*, which means that each *pixel* (dot of color) can be individually controlled by the computer—as opposed to an alphanumeric terminal, where the terminal itself chooses which dots to turn on and off to make letters, numbers, and other characters that the computer has told it to create.

terminal window

A window, on a window system, made by a terminal emulation program such as **xterm**, GNOME Terminal, or **konsole**. It's an interface like an alphanumeric terminal—with a shell prompt where you can type command lines from your keyboard and can see any text that those programs output. In most cases, a mouse or other pointing device is useless inside a terminal window—though it works at the borders of the window (to minimize the window, move it, etc.) just as on other windows.

titlebar

The part of the window border above a window. It shows the window's title. It also has buttons and/or menus that control characteristics of the window, such as minimizing the window or lowering the window to the bottom of a window stack. Figure 2-4 shows a titlebar.

virtual consoles

Virtual consoles, available on Linux and other PC operating systems, let you access several different fullscreen login sessions on the same screen, independent of any window system. Just after a reboot, if you get a “login:” prompt (as in Example 1-1), you'll be using the first virtual console. To use other virtual consoles, hold down the **CTRL** key and the left **ALT** key, then press one of the function keys **F1** (for the first console) through **F6** (for the sixth). Each of those function keys will bring up a separate login session. (Once you've started the X Window System, **CTRL-ALT-F7** may take you to the X display.) Use each virtual console for whatever you want—just remember to log out of each when you're done!

window

An area of an output display often smaller in size than the maximum size of the display screen.

If a window manager program is running, a window usually will have a well-defined border, a title, and other characteristics. The window manager lets you move and resize a window as well.

working directory

When you give Unix a *relative* pathname to a file or subdirectory, the working directory is the starting point—the directory where that relative pathname starts. Here are two examples:

If your working directory is */home/joe/food* and you type the command **less recipes/fish**, Unix opens the file */home/joe/food/recipes/fish*. (Your working directory is still */home/joe/food*.)

If you type the command “`ls ..`” from any working directory, you get a listing of the files in your parent directory. That command uses the relative path-name to the parent directory (`..`). So if your working directory is `/home/joe/food`, that command would list the parent directory `/home/joe`. Or, if your working directory is `/home/joe`, that same command would list the parent directory `/home`.

Each process running on a Unix system has its own working directory, which the program can change at any time. For instance, you can give the shell the command `cd` to change its working directory.

x86 processor

Since the 1980s, the Intel Corporation has been building a family (series) of microprocessors (which are used in computer *CPUs*, Central Processing Units) whose model numbers end in the number 86. The first was the 8086; then came the 80286 (the 80186 wasn't as widely used); next was the very popular 80386; and so on. Many operating systems run only on a certain family of microprocessors. Microsoft Windows, for instance, is primarily designed for the x86 family; recent versions won't work on a processor earlier than the 80586. Unix-like operating systems run on many different microprocessor families, but the x86 is one of the most popular—especially for Linux, which works well with an 80386.

xterm program

A program that runs under the X Window System. It makes a terminal window (called an xterm window) in which a Unix login session runs.