What Is an Operating System?

An Operating System (which is software) is a complex master control program whose principal function is to use the resources of a computer efficiently.

The operating system is always there, waiting to serve you and to manage the resources of your computer.

In Unix, the operating system can be divided into three layers: utilities, shell, and kernel.

Utilities: The utilities of an operating system are the standard commands and programs associated with the operating system.

Shell: A shell is a program that runs other programs.

Kernel: The kernel is a collection of software that provides the basic capabilities of the operating system.

"Unix" Is the Name of a Culture

Unix means much more than a family of operating systems.

In using Unix, we will learn to approach and solve problems by combining simple programs into elegant structures.

The Unix Connection

Host: The main computer that actually does most of the work.

Character Terminal:

A character terminal has nothing more than a screen and a keyboard, and can display only characters.

Graphics Terminal:

It can display everything that can be drawn on a screen using small dots: pictures, geometric shapes, and so on.

Most graphics terminals have a mouse and are designed to be used with a graphical user interface.

Console:

A display screen and a keyboard that are part of the host computer itself.

A Console is just another terminal.

What Happens When You Press a Key?

Each time you press a key, a signal is sent to the host. The host responds by sending its own signal back to your terminal telling it to display the appropriate character on the screen.

If the host computer is far away, you might not see the letters appear on the screen immediately after you press keys.

Network Connections

Network

A network refers to two or more computers connected together. People connect computers into networks in order to share resources.

Local Area Network (LAN)

When computers are connected directly by using some type of cable, we call the network LAN.

Wide Area Network (WAN)

Many LANs are connected to other networks, forming a bigger network that is called WAN.

Backbone

A high-speed link that ties together the smaller LANs into one large wide area network.

Gateways

Some computers, called gateways, will act as the links between the campus network and the outside world.

Internet

Around the world, the major wide area networks are connected to a system known as the Internet. Any computer on the Internet can connect to any other computer on the Internet.

Client-Server Relationship

Server

In network terminology, any program that offers a resource is called a server.

A program that provides access to files over the network is called a file server; A program that coordinates the printing of data using different printers is called a print server.

Sometimes the name server is used to refer to a real computer too (i.e., mail server, news server, …).

Client

A program that uses a resource is called a client.

Unix system programmers often talk about the connection between a client program and a server as the Client-Server Relationship.

VT-100 Terminal:

A very old terminal made by Digital Equipment Corporation.

Starting to Use Unix

System Manager/Administrator:

All Unix systems require administration and maintenance. The person who performs these duties is called the system manager or system administrator.

Userid: A name that identifies you to the system.

Password: A secret code that you must type in each time you use the system.

Account: Once you have permission to use a system, we say that you have a Unix account on that computer.

Logging In: Starting work with Unix

Logging Out: Stopping working with Unix

1. *logout* 2. *exit* 3. *login* 4. *Ctrl-D* (for some systems)

Getting Down to Work: The Shell Prompt

The program that reads and interprets your commands is called a "shell". When the shell is ready for you to type the next command, it will display a "prompt".

If you use the C-Shell, your prompt will be a `%`.

If you system manager has customized your environment, the prompt may be somewhat different, i.e., mars.acs.uwinnipeg.ca>.

With a Bourne shell, a `$` may be your prompt.

Upper- and Lowercase

Unix is case sensitive.

Some cases to use uppercase letters:

1. Passwords
2. Environment variables (TERM, HOME, …)
3. Writing programs
4. Electronic mail (E-mail) address

   Example: uwinnipeg.ca vs. Uwinnipeg.ca

Who Has Been Using Your Account: *last*

The command *last* can display login and logout information about users and terminals.

Example:

```
mars% last
sliao    pts/1        wnpgmb0426w-ds02 Sat Sep  2 15:40   still logged in
nischal  pts/1        acs-3d07b-f01.uw Fri Sep  1 12:23 - 12:24  (00:00)
acs2941  pts/1        acs-3d07b-f01.uw Fri Sep  1 12:22 - 12:23  (00:00)
. . .
aulakh-s pts/4        142.161.217.182  Sun Mar 19 03:28 - 04:03  (00:34)

wtmp begins Sun Mar 19 03:27:57 2023
mars%
```

passwd: You can change your password by using the command *passwd*.

Userid vs. User

A user is a person who utilizes a Unix system in some way. Unix itself only knows about userids. If someone logs in with your userid, Unix has no way of knowing whether or not it is really you.

Example:

```
mars% finger
Login     Name          Tty      Idle  Login Time   Office     Office Phone   Host
sliao     Simon Liao    pts/1          Sep  2 15:40
(wnpgmb0426w-ds02-202-50-88.dynamic.bellmts.net)
mars%
```

The Superuser Userid: root

Within Unix, all userids are considered equal, except the superuser root.

Using the Keyboard with Unix

TTYs: When the Unix was first developed, the programmers used Teletype ASR33 terminals. The terminals had letters, numbers, and a "Control" key. TTY (Teletype) quickly became a way to refer to any terminal.

tty is a command to display the name of your terminal.

```
mars% tty
/dev/pts/1
mars%
```

stty is a command to set up your terminal.

Another convention derived from Teletypes is how we use the word "print". Teletype printed output on paper. But now the same information would be displayed on a screen.

"Print" means "Display"

Examples: appreciate

pwd	Print Working Directory
lpr	Line Printer

How to Deal with Different Types of Terminals?

For older Unix systems, the descriptions of all different types of terminals into a single file, termcap database. The newer Unix systems should use the terminfo database and associated libraries.

How Does Unix Know What Terminal You Are Using?

There is a global variable named TERM whose value is the type of terminal you are using.

```
mars% echo $TERM
vt100
mars%
```

Understanding Your Keyboard

Unix must work with any terminals and there is no such thing as a standard keyboard. As a solution, Unix defines standard codes that are mapped into different keyboards.

Some codes:

erase:	Erase the last character that you typed.
werase:	Erase the last word you typed.
kill:	Erase the whole line.
intr:	Abort the program that is currently running (interrupt).
quit:	quit is designed for advanced programmers. When you stop a program with quit, it not only stops the program, but also makes a copy of the contents of memory at that instant.
stop:	Pause the screen display.
start:	Restart the screen display.
eof:	End of file

Checking the Special Keys for Your Terminals: *stty*

To check how your Unix system uses your particular terminal, you can use the *stty* (set terminal) command.

```
mars% stty
speed 38400 baud; line = 0;
kill = ^X;
-brkint -imaxbel
mars%
```

You can set some special keys in your .login file.

For example, add

```
stty kill ^X
```

to your .login file will change the kill key to ^X.

Summary of some Keyboard Codes:

Code	Key	Purpose
intr	^C	stop a program that is running
erase	<Backspace>,<Delete>	erase the last character typed
werase	^W	erase the last word typed
kill	^X, ^U	erase the entire line
quit	^\	stop a program, save **core** file
stop	^S	pause the screen display
start	^Q	restart the screen display
eof	^D	indicate there is no more data

Teletype (ASR33) Control Signals

<Ctrl-H>: Caused the print carriage to back up a single space before printing the next character.

<Ctrl-M>: Moved the print carriage to the beginning of the line.

<Ctrl-J>: Moved the paper up one line.

<Ctrl-M> <Ctrl-J>: Would position the carriage and the paper at the beginning of the next line.

<Ctrl-I>: Tab Setting

How Teletype Control Signals are Used by Unix

^H: When you press the <Backspace> key, Unix interprets the signal as being a ^H.

^I: When you press the <Tab> key, Unix interprets it as a ^I.

^M: Signal that you have reached the end of a line. (return)

^J: Mark the end of each line. (newline)

Unix treats the data typed at the keyboard the same as data read from a file.

When you display data, each *newline* (^J) is changed by Unix into a *return newline* (^M^J) combination.

Q. Can you press ^J instead of <Return> to enter a command at any time?

A. Yes!

Programs to Use Right Away

date The date command will display the current time and date.

Example:

```
mars% date
Tue Sep  5 15:29:51 CDT 2023
mars%
```

cal The cal command displays a calendar.

Examples:

```
mars% cal
   September 2023
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

mars%
```

calendar

Unix does have a calendar command, which is different from cal.

The calendar program offers a reminder service based on a file named calendar in your home directory. The program calendar will check this file and display all the lines that have today's and tomorrow's date.

Example:

```
pearl% cat calendar
January 5       Day1
January 6       Day2
January 7       Day3
January 8       Day4
pearl% date
Thu Jan  5 17:13:26 CST 2019
pearl% calendar
Jan 05  Day1
Jan 06  Day2
pearl%
```

uptime

The uptime command will tell you that how long your particular computer has been up.

Example:

```
mars% uptime
 15:30:55 up 20 days,  5:49,  3 users,  load average: 0.00, 0.01, 0.05
mars%
```

In this case, **mars** has been up for 20 day, 5 hour and 49 minutes, and there are 3 users currently logged in. The last three numbers show the average number of jobs in the run queue over the last 1, 5 and 15 minutes, respectively.

hostname

The hostname command will display the name of the system you are using.

Example:

```
mars% hostname
mars-acs-uwinnipeg-ca
mars%
```

The Online Unix Manual

Unix comes with a large, built-in manual that is accessible at any time from your terminal.

The Online Manual is a collection of files, stored on disk, each of which contains the documentation about one Unix command or topic.

The Online Manual can be accessed at any time by using the *man* command.

Examples:

```
man cp
man man
man mv lpr ln
```

How Is the Online Manual Organized?

Section

1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in /dev)
5 File formats and conventions (e.g. /etc/passwd)
6 Games
7 Miscellaneous (including macro packages and conventions)
8 System administration commands (usually only for root)
9 Kernel routines (nonstandard)

The most important section is Section 1. Section 2, 3, 4, and 5 may be important to programmers.

The following conventions apply to the SYNOPSIS section and can be used as a guide in other sections:

bold text	type exactly as shown
italic text	replace with appropriate argument
[-abc]	any or all arguments within [ ] are optional
-a | -b	options delimited by | cannot be used together
argument …	argument is repeatable
[expression] …	entire expression within [ ] is repeatable
…	

Examples:

`man kill`

will show the description of **`kill`** that resides in Section 1 of the manual;

`man -s 2 kill`

will show the description of **`kill`** that resides in Section 2 of the manual;

`man -s 7 man`

will show the description of **`man`** in Section 7;

`man umask`

will show the description of **`umask`** that resides in Section 1 of the manual:

```
mars% man kill

KILL(1)                 User Commends                        KILL(1)

NAME
       kill - terminate a process

SYNOPSIS
       kill [-s signal|-p] [-q sigval] [-a] [--] pid...
       kill -l [signal]
.
.
.
```

At the end of the kill man pages are the following a few lines, which tell us there are other pages related to this one:

```
SEE ALSO
       bash(1), tcsh(1), kill(2), sigvec(2), signal(7)

AUTHOR
       Taken from BSD 4.4. The ability to translate process names to
process ids was added by Salvatore Valente …

AVAILABILITY
       The kill command is part of the util-linux package and is available
from Linux Kernel Archive …
```

NAME: This is what the command is all about.

SYNOPSIS: Official explanation of how to enter the command.

DESCRIPTION: Could be divided into two separate sections: Description & Options.

FILES: This section shows the names of the files that are used by this command.

SEE ALSO: It shows you other places to look in the manual for more information.