

CST 126 – LESSON 2

Touring Essential Programs
Command Basics

Overview

- Computer Hardware and Unix Architecture
- Playing the shell game – instructions to the shell
- Command structure – Arguments, options, etc.
- Understand processes.
- Navigating the file system.
- Examining and managing files.
- Accessing the programmer’s manual.
- Accessing Internet resources.

Computer System Hardware

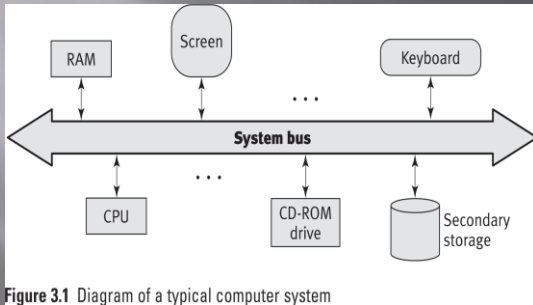


Figure 3.1 Diagram of a typical computer system

Unix Software Architecture

- Device Driver Layer
 - Mouse driver, printer driver
- The UNIX Kernel
 - Process Management
 - File Management
 - Main Memory Management
 - Disk Management
- The System Call Interface
 - Entry points to Kernel
- Language Libraries
 - C, C++, Java, FORTRAN etc.
- UNIX Shell
- Applications
 - Compilers, word processors, spreadsheets, ftp, telnet, Web browser etc.

UNIX Software Architecture (Cont.)

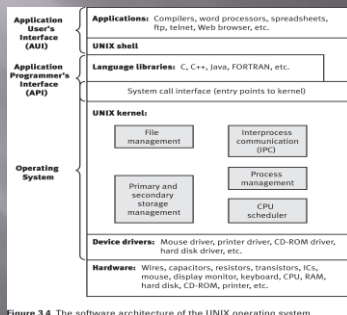


Figure 3.4 The software architecture of the UNIX operating system

Unix Shells

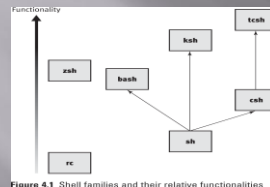


Figure 4.1 Shell families and their relative functionalities

SHELL	DEFAULT PROMPT	NAME
sh	\$	Bourne Shell (the original)
bash	\$	Bourne Again Shell
ksh	\$	Korn Shell
csh	%	C Shell
tcsh	%	T C Shell

Shell Setups and Similarities

TABLE 3.2 Shell Configuration Files

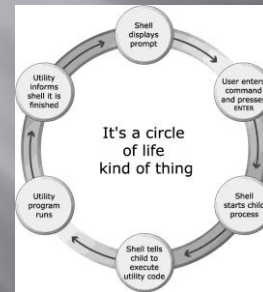
UNIX Shell	Name of the Configuration File(s)
Bourne (sh), Korn (ksh)	/etc/profile*, ~/.profile
Korn (ksh)	~/.kshrc
C (csh)	~/.login, ~/.cshrc
Bash (bash)	~/.bashrc, ~/.bash_profile
Z (zsh)	~/.zshrc
TC (tcsh)	~/.tcshrc

TABLE 4.2 Shell Similarities

Function	Description
Execution	The ability to execute programs and commands
I/O Handling	The control of program and command input and output
Programming	The ability to execute sequences of programs and commands

What shell are we running? Type "echo \$SHELL" at the command prompt.

Issuing Commands to a Shell



How the shell works

Basic Commands Introduced in Chapter Two

Command	Command	Command	Command
whoami	date	hostname	clear
cal	ps	ls	who
more	wc	sh	history
alias	unalias	pwd	mkdir
cd	cat	head	tail
cat	cp	rm	lp
lpr	man	mv	

Structure of a UNIX command

\$ command [[-] option (s)] [option argument (s)] [command argument (s)]

Examples:

```

❑ $ ls
❑ $ ls -la
❑ $ ls -la m*
❑ $ lpr -Pspr -n 3 proposal.ps

```

* In some circles you will hear these options, option arguments, and command arguments referred to as parameters. Call them what you will, just make sure you understand what you are referencing.

Executing Commands - Passing Information to a Utility

- ❑ When any information is passed to a utility, the shell runs the utility, and passes the information that comes after the command to the utility.
- ❑ Information passed to the utility is called an **arguments or options**.
- ❑ **Arguments or options** provide instructions to utilities.
- ❑ Multiple **arguments or options** can also be passed to a utility.

Redirecting Output from a Utility

- ❑ By default, the results provided by any utility are displayed on the user's screen.
- ❑ Instructions can be given to redirect the output of a utility to a file.
- ❑ The command to redirect the output of a file is "utility > filename".
- ❑ The > is the instruction to redirect the output to a new file.

Redirecting Output from a Utility

- ❑ The double redirect (`>>`) can be used to append the output of a utility to the end of an existing file.
- ❑ The `|` (pipe) command can be used to redirect the output of one utility to another utility.
- ❑ The argument following the pipeline must be a utility.

Reissuing Commands

- ❑ The `!!` command or the `!#` command can be used to re-execute a previously entered command.
- ❑ The `!!` command works only on a csh, a tch, or a bash shell.
- ❑ The `!#` command works only on a Korn shell.
- ❑ The sh shell does not allow a user to re-execute previously entered commands.

Reissuing Commands

- ❑ The Up arrow key can be used to display previously entered commands one at a time.
- ❑ The shell keeps track of the commands that we issue at the prompt.
- ❑ The `history` command can be used to provide a list of all the commands entered previously, and every command has a number associated with it.

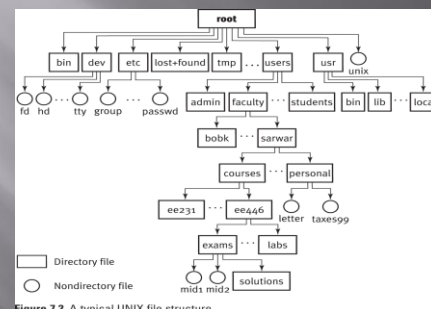
Using Nicknames for Commands

- ❑ An `alias` or alternate name can be used for commands that are hard to remember.
- ❑ The `alias` command can be used to provide a list of all the current aliases.
- ❑ The command `unalias alias name` can be used to remove an alias.

Listing Processes

- ❑ A **process** is an instance of a running program code. It is often referred to as a program in execution.
- ❑ The `ps` command can be used to obtain a list all the current **processes**, with some information about each **process**.
- ❑ The output displays the process id, the port, CPU time, and the code that the process is running.

Typical Unix File System Structure



Navigating the File System

- ❑ The arrangement of files and folders in a system is called the file system.
- ❑ The “pwd” (present working directory) command can be used to get the location of the user’s home directory.
- ❑ The result displays the path from the root to the present working directory.
- ❑ The topmost directory is called the “root,” and is identified as /.
- ❑ The “ls /” command can be used to obtain a list of all the files and directories in the root directory.

Navigating the File System

- ❑ The “mkdir” (make directory) command is used to create a new directory.
- ❑ The command requires one argument – the name to be given to the directory.
- ❑ The “-F” option can be used with the ‘ls’ command to include a / (slash) after all the directory names, and also to identify other kinds of files.

Navigating the File System

- ❑ The “cd” command can be used to make a directory the current directory.
- ❑ The command requires one parameter – the name of the directory that needs to be made the current directory.
- ❑ The cd command is used to return to the home directory. It does not require any parameters.

Navigating the File System

TABLE 7.2 Some Important Hidden Files and Their Purposes

File Name	Purpose
.	Present working directory
..	Parent of the present working directory
.addressbook	Address book for pine
.bashrc	Setup for Bash shell
.cshrc	Setup for C shell
.exrc	Setup for vi
.login	Setup for shell if c or tc shells are the login shells; executed at login time
.mailrc	Setup and address book for mail and mailx
.profile	Setup for shell if Bourne or Korn shell is the login shell; executed at login time

Navigating the File System

TABLE 7.3 Summary of the Output of the ls -l Command (fields are listed left to right)

Field	Meaning
First letter of first field	File type: - ordinary file b block special file c character special file d directory l link p named pipe (FIFO) s socket
Remaining letters of first field	Access permissions for owner, group, and others
Second field	Number of links
Third field	Owner’s login name
Fourth field	Owner’s group name (can also be a number)
Fifth field	File size in bytes
Sixth, seventh, and eighth field	Date and time of last modification
Ninth field	File name

```
$ ls -l ~/courses/ee446/exams
-rwxr--r-- 1 satvar faculty 163 Mar 16 11:10 mid1
-rwxr--r-- 1 satvar faculty 163 Apr 11 14:34 mid22
drwxr-xr-x 1 satvar faculty 163 May 12 23:44 solutions
$
```

Accessing the Programmer’s Manual (Man Pages)

- ❑ The UNIX and Linux systems include an extensive collection of powerful utility programs, system features, application languages, and support libraries.
- ❑ The UNIX programmer’s manual provides the information needed to employ the exact syntax of a particular option or command format for a utility.

Accessing the Programmer's Manual

- ❑ The manual contains a detailed documentation on the uses and functions of utility programs, application programs, and libraries.
- ❑ The manual also contains information on UNIX system files and system programming libraries.
- ❑ It also includes supplementary information on related special files and commands for each entry.

Accessing the Programmer's Manual

- ❑ The man command can be used to provide an online manual entry for a utility or a command.
- ❑ The command requires one argument - the name of the utility or the command.
- ❑ The "man -k" command can be used to search the manual pages' descriptions for keywords.

Summary

- ❑ The shell is a process that interprets the commands entered by the user.
- ❑ The file system is a collection of files and folders.
- ❑ We can work on many utilities, print files, and access system folders from a shell.
- ❑ Online manual pages provide a detailed description of system utilities, files, and functionalities.