

Unix Network Programming-Introduction

UNIX

UNIX is CUI operating system. Operating system is an interface between hardware and applications software's. It serves as the operating system for all types of computers, including single-user personal computers and engineering workstations, multi-user microcomputers, mini-computers, mainframes and super computers as well as special-purpose devices.

History of UNIX:-

MuliUser:-

Multi user operating system means more than one user shares the same system resources (hard disk, memory, printer, application software etc.,) at the same time.

Multi tasking:-

Another highlight of UNIX is that it is Multitasking, implying that it is capable of carrying out more than one job at the same time. Depending on the priority the task, the operating system appropriately allots small time slots to each foreground and background task.

Programming Facility:-

UNIX operating system provides shell. Shell works like a programming language. It provides commands and keywords. By running these two, user can prepare efficient program.

Portability:-

One of the main reasons for the universal popularity of unix is that it can be ported to almost any computer system, with only the bare minimum of adoptions to suit the given computer architecture.

Communication:-

UNIX provides electronic mail. The communication may be within the network of a single main computer, or between two or more such computer networks. The user can easily exchange mail data, programs thro such networks.

Security:-

UNIX provides three levels of security to protect data. The first is provided by assigning passwords and login names to individual users ensuring that not anybody can come and have access to your work.

At the file level, there are read, write and execute permissions to each file which decide who can access a particular file, who can modify it and who can execute it.

Lastly, there is file encryption. This utility encodes your file into an unreadable format, so that even if some one succeeds in opening it, your secrets are safe.

Open system:-

Unix Network Programming-Introduction

The source code for the UNIX system and not just the executable code, has been made available to users and programmers. Because of this many people have been able to adapt the UNIX system in different ways.

System Calls:-

Programs interact with the kernel through approximately 100 system calls. System calls tell the kernel to carry out various tasks for the program, such as opening a file, writing a file, obtaining information about a file, executing a program, terminating a process, changing the priority of a process and getting the time of day.

Different implementations of unix system have compatible system calls with each call having the same functionality.

Help facility:-

UNIX provides manual pages for UNIX commands.

Shell:-

The shell reads your commands and interprets them as requests to execute a program or programs, which it then arranges to have carried out. Because the shell plays this role, it is called a command interpreter. As a programming language, it permits you to control how and when commands are carried out. Shell acts an interface between user and the kernel.

Kernel:-

The kernel is the part of the operating system that interacts directly with the hardware of a computer, through device drivers that are built into the kernel. The major functions of the kernel are to manage computer memory, to control access to the computer, to maintain the file system, to handle interrupts, to handle errors, to perform input and output services (which allows computers to interact with terminals, storage devices and printers) and to allocate the resources of the computer among users. Programs interact with the kernel through approximately 100 system calls. System calls tell the kernel to carry out various tasks for the program, such as opening a file, writing to a file, obtaining information about a file, executing a program, terminating a process, changing the priority of a process and getting the time of day.

UNIX file system

A file is the basic structure used to store information on the UNIX system. The UNIX file system provides a logical method for organizing, storing, retrieving, manipulating and information. Within the UNIX there are three different types of files.

- 1) Regular files
- 2) Directory files
- 3) Special files

1. Ordinary files:-

As a user, the information that you work with will be stored as an ordinary file. Ordinary files are aggregates of characters that are treated as a unit by the UNIX system. An ordinary file can contain normal ASCII characters such as a text for manuscripts or programs. Ordinary file can be created, changed or deleted as you wish.

2. Directory Files:-

Directory is a file that holds other files and contains information about the locations and attributes of these other files.

The Hierarchical File Structure

The UNIX file system called as hierarchical file system. In fact, within the UNIX system, there is no limit to the number of files and directories you can create in a directory that you own.

Files systems of this type are often called tree-structured file systems, because each directory allow you to branch off into other directories and files.

/ (root):- This is the root directory of the file system, the main directory of the entire file system, and the root directory for the super user.

/bin:- bin stands for binary. This directory contains executable files for most of the Unix commands. UNIX commands can be either C programs or shell programs. Shell programs are nothing but a collection of several UNIX commands.

/etc: - This contains system administration and configuration databases.

/dev: - This contains the special device files that includes terminals, printers and storage devices.

/lib: - This directory contains all the library functions provided by UNIX for programmers.

/sbin: - This contains programs used in booting the system and in system recovery.

/opt: - This is the root for the sub tree containing the add-on application packages.

/home:-This contains the home directories and files of all users. If your log name is VIJAY, your default home directory is /home/VIJAY.

/tmp: - This contains all temporary files used by the UNIX system.

/mnt: - Contains entries for removable media such as CD-ROMs.

/UNIX: - This directory contains UNIX kernel programming.

/usr: - This contains other accessible directories such as /usr/lib and /usr/bin

/var: - This contains the directories of all files that vary among systems. These include files that log system activity, accounting files, mail files, application packages, backup files for editors and many other types of files that vary from system to system.

Basic Commands

1. \$log name: It displays the current user name.
2. \$pwd: It displays current working directory.
3. \$clear: It clears the screen.

4. \$ exit: To logout from current user.
5. \$ date: It displays the system date and time
Sat jun 11 04:10:40 IST 2009
6. \$who am i: It displays current username, terminal number, date and time at which you logged into the system.
7. \$who: To displays the information about all the users who have logged into the system currently. i.e each user login name , terminal number, dat and time that the person logged in.
8. \$finger: It displays complete information about all the users who are logged in.
9. \$ cal: It displays the previous month, current month and next month calendar.
10. \$ cal year: It displays the given year calendar.
11. \$ cal month year: It displays the given month calendar only.
12. # init: To change system run levels.
 - #init 0: To shut down the system.
 - #init 1: To bring the system to single user mode.
 - #init 2: To bring the system to multi user mode with no resource shared.
 - #init 3: To bring the system multi user mode with source shared.
 - #init 6: Halt and reboot the system to the default run level.
13. \$banner "VIJAYA CHANDRA ": It prints a message in large letters.

Creating files:

These are two commands to create files: touch and cat

1. \$ touch filename: It creates zero bytes file size.

Eg: \$touch sample.

The size of sample file is zero bytes.

Touch does not allow you to store anything in a file. It is used for to create several empty files quickly.

Eg: touch file1 file2 file3

2. Cat command

Syntax: \$cat>filename

Ctrl+d { to close file}

\$Cat >> sample

To append data to the file.

Ctrl+d

3. \$ cat < filename (or) \$cat filename: To open a file.

Eg: \$cat sample

It displays sample file contents.

Eg: \$cat file1 file2 file3

It displays file1 contents followed by file2 then followed by file3.

Removing file:

4. rm command: To remove the given file.

Syntax: \$rm filename

\$rm sample: It removes sample file.

\$rm -I filename: It asks confirmation before deleting the file.

\$rm file1 file2 file3: It removes three files.

\$rm *: It removes all files in current directory.

Creating Directory:

5. Mkdir command: To make a directory.

Syntax:

\$mkdir directory name

Creating multiple directories

\$mkdir dir1 dir2 dir3.....dirn.

Change directory:

Syntax: \$cd directory name

\$cd..: To change into parent directory.

\$cd\: To change to root directory.

Remove directory:

1. \$rmdir directory name: To delete a directory but directory should be empty.
2. \$rm -r directory name: It removes all files and sub directories, sub directory files including directory.

Copy a file:

\$cp file1 file2: This will copy the contents of file1 into a file2. It file2 is already existed it overwrites.

\$cp -I file1 file2: If file2 is already existed then it asks the confirmation.

Rename a file:

\$mv old filename new filename: To rename a file.

Comparison of files:

\$cmp file1 file2: It compares file1 and file2. If both file contents are same number output if files are different then it displays line number and character location.

Word count:

\$wc filename: It counts number of lines, words and character and displays.

\$wc file1

5 10 70 file1

In above output, 5 represents as total number of lines, 10 represents as total number of words, 70 represents as total number of characters.

\$wc -l file1: It displays only line count.
\$wc -w file1: It displays only word count.
\$wc -c file1: It displays only character count.
\$wc -lw file1: It displays only line and word count.
\$wc -wc file1: It displays only word and character count.
\$wc -lc file1: It displays only line and character count.

Listing of files:

ls: It displays list of files and directories.
\$ls -x: It displays width wise.
\$ls |pg: It displays list of files and directories pages wise.
\$ls -x |pg: It displays list of files and directories pages wise and width.
\$ls -a: It displays files and directories including . and.. Hidden files.
\$ls -r: It displays files and directories in reverse order i.e descending order.
\$ls -t: It displays files and directories based on date and time of creation i.e last file to first file.
\$ls -rt: It displays file and directories based on date and time of creation but in reverse order. i.e first file to last file.
\$ls -i: It displays files and directories along with in node number.
\$ls -l: It displays files and directories along in long list format.
Wild card character or meta characters: (*,?,[],-)
\$ls a* : It displays all file, starting letter is a.
\$ls b*k : It displays all file starting letter is b and ending letter is k.
\$ls *k: It displays all files ending letter is k.
\$ls b?k: It displays all 4 length filenames but starting letter is b and ending letter is k.
\$ls [aeiou]*:It displays all files but first character of the filename to be listed must be any of the letters given within square brackets and the remaining can be anything.
\$ls [!aeiou]*: It displays all files whose first character is anything other than a vowel.
\$ls [k-v]*: It displays all files whose starting letter is between k and v.