Subject: Unix and Shell Programming

Course Code: BCAC601   Semester: 6

Module 1

Introduction to UNIX

Unix is a computer Operating System which is capable of handling activities from multiple users at the same time. The development of Unix started around 1969 at AT&T Bell Labs by Ken Thompson and Dennis Ritchie.

What is Unix?

The Unix operating system is a set of programs that act as a link between the computer and the user.

The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the operating system or the kernel.

Users communicate with the kernel through a program known as the shell. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

Unix was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna at Bell Labs.

There are various Unix variants available in the market. Solaris Unix, AIX, HP Unix and BSD are a few examples. Linux is also a flavor of Unix which is freely available.

Several people can use a Unix computer at the same time; hence Unix is called a multiuser system.

A user can also run multiple programs at the same time; hence Unix is a multitasking environment.

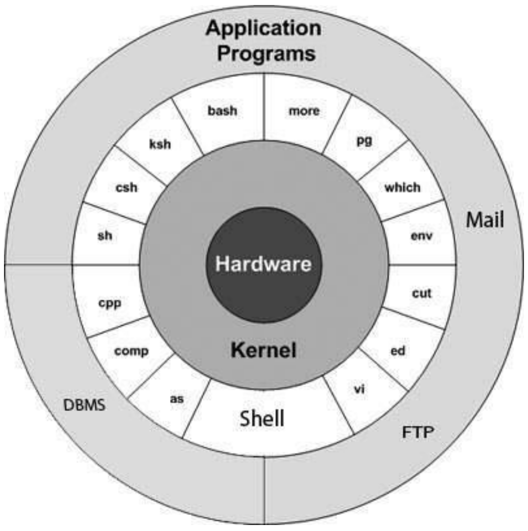
Difference between Linux and Unix

Comparison	Linux	Unix
Definition	It is an open-source operating system which is *freely available to everyone*.	It is an operating system which *can be only used by its copyrighters*.
Examples	It has different distros like Ubuntu, Redhat, Fedora, etc	IBM AIX, HP-UX and Sun Solaris.
Users	Nowadays, Linux is in great demand. Anyone can use Linux whether a home user, developer or a student.	It was developed mainly for servers, workstations and mainframes.
Usage	Linux is used everywhere from servers, PC, smartphones, tablets to mainframes and	It is used in servers, workstations and PCs.

	supercomputers.	
Cost	Linux is freely distributed, downloaded, and distributed through magazines also. And priced distros of Linux are also cheaper than Windows.	Unix copyright vendors decide different costs for their respective Unix Operating systems.
Development	As it is open source, it is developed by sharing and collaboration of codes by world-wide developers.	Unix was developed by AT&T Labs, various commercial vendors and non-profit organizations.
Manufacturer	Linux kernel is developed by the community of developers from different parts of the world. Although the father of Linux, Linus Torvalds oversees things.	Unix has three distributions IBM AIX, HP-UX and Sun Solaris. Apple also uses Unix to make OSX operating system.
GUI	Linux is command based but some distros provide GUI based Linux. Gnome and KDE are mostly used GUI.	Initially it was command based OS, but later Common Desktop Environment was created. Most Unix distributions use Gnome.
Interface	The default interface is BASH (Bourne Again SHell). But some distros have developed their own interfaces.	It originally used Bourne shell. But is also compatible with other GUIs.
File system support	Linux supports more file system than Unix.	It also supports file system but lesser than Linux.
Coding	Linux is a Unix clone, behaves like Unix but doesn't contain its code.	Unix contain a completely different coding developed by AT&T Labs.
Operating system	Linux is just the kernel.	Unix is a complete package of Operating system.
Security	It provides higher security. Linux has about 60-100 viruses listed till date.	Unix is also highly secured. It has about 85-120 viruses listed till date.
Error detection and solution	As Linux is open-source, whenever a user post any kind of threat, developers from all over the world start working on it. And hence, it provides faster solution.	In Unix, users have to wait for some time for the problem to be resolved.

Unix Architecture

Here is a basic block diagram of a Unix system −



The main concept that unites all the versions of Unix is the following four basics −

Kernel − The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

Shell − The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.

Commands and Utilities − There are various commands and utilities which you can make use of in your day-to-day activities. cp, mv, cat and grep, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.

Files and Directories − All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the filesystem.
In Unix a **file** is just a destination for or a source of a stream of data. Thus, a printer, for example, is a file and so is the screen. A **process** is a program that is currently running. So a process may be associated with a file.

System calls in Unix are used for file system control, process control, inter process communication etc. Access to the Unix kernel is only available through these system calls. Generally, system calls are similar to function calls, the only difference is that they remove the control from the user process.

Features of UNIX

Unix is an operating system, so it has all the features that the OS must-have. UNIX also looks at a few things in a different way than other OS. Features of UNIX are listed below:

1. Multiuser System:

Unix provides multiple programs to run and compete for the attention of the CPU. This happens in 2 ways:

i) Multiple users running multiple jobs
ii) Single user running multiple jobs

In UNIX, resources are actually shared between all the users, so-called a multi-user system. For doing so, computer give a time slice (breaking unit of time into several segments) to each user. So, at any instant of time, only one user is served but the switching is so fast that it gives an illusion that all the users are served simultaneously.

2. Multitask System:

A single user may run multiple tasks concurrently. Example: Editing a file, printing another on the printer & sending email to a person, and browsing the net too at the same time. The Kernel is designed to handle user's multiple needs.

The important thing here is that only one job can be seen running in the foreground, the rest all seems to run in the background. Users can switch between them, terminate/suspend any of the jobs.

3. The Building-Block Approach:

The Unix developers thought about keeping small commands for every kind of work. So, Unix has so many commands, each of which performs one simple job only. You can use 2 commands by using pipes ('|'). Example: $ ls | wc Here, | (pipe) connects 2 commands to create a pipeline. This command counts the number of files in the directory. These types of connected commands that can filter/manipulate data in other ways are called filters.

Nowadays, many UNIX tools are designed in a way that the output of 1 can be used as an input for the others. We can create a large number of combinations by connecting a number of tools.

4. The UNIX Toolkit:

Unix has a kernel but the kernel alone can't do much that could help the user. So, we need to use the host of applications that usually come along with the UNIX systems. The applications are quite diversified. General-purpose tools, text manipulation utilities (called filters), compilers and interpreters, networked programs, and system administration tools are all included. With every UNIX release, new tools are being added and the older ones are modified/ removed.

5. <u>Pattern Matching:</u>

Unix provides very sophisticated pattern matching features. The meta-char '*' is a special character used by the system to match a number of file names. There are several other metachars in UNIX. The matching is not confined to only filename. Advanced tools use a regular expression that is framed with the characters from this set.

6. <u>Programming Facility:</u>

Unix provides shell which is also a programming language designed for programmers, not for casual end-users. It has all the control structures, loops, and variables required for programming purposes. These features are used to design the shell scripts (programs that can invoke the UNIX commands).
Many functions of the system can be controlled and managed by these shell scripts.

7. <u>Documentation:</u>

It has a 'man' command that stands for the manual, which is the most important reference for any commands and their configuration files. Apart from the offline documentation, there is a vast number of resources available on the Internet.

What is POSIX (Portable Operating System Interface)?

POSIX (Portable Operating System Interface) is a set of standard operating system interfaces based on the Unix operating system. The most recent POSIX specifications -- IEEE Std 1003.1-2017 -- defines a standard interface and environment that can be used by an operating system (OS) to provide access to POSIX-compliant applications. The standard also defines a command interpreter (shell) and common utility programs. POSIX supports application portability at the source code level so applications can be built to run on any POSIX compliant OS.

A brief history of the POSIX standard

The POSIX interfaces were originally developed under the auspices of IEEE. However, the POSIX standard is now being developed and maintained by the Austin Common Standards Revision Group, commonly referred to as the Austin Group.

The Austin Group is a joint working group made up of members from IEEE, The Open Group and Joint Technical Committee 1 of the International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). IEEE owns the POSIX trademark. The Open Group, which owns the Unix trademark, is a global consortium that develops technology standards.

Internal and external commands

The UNIX system is command-based i.e. things happen because of the commands that you key in. All UNIX commands are seldom more than four characters long. They are grouped into two categories:

Internal Commands: Commands which are built into the shell. For all the shell built-in commands, execution of the same is fast in the sense that the shell doesn't have to search the given path for them in the PATH variable, and also no process needs to be spawned for executing it.
Examples: source, cd, fg, etc.

External Commands: Commands which aren't built into the shell. When an external command has to be executed, the shell looks for its path given in the PATH variable, and also a new process has to be spawned and the command gets executed. They are usually located in /bin or /usr/bin. For example, when you execute the "cat" command, which usually is at /usr/bin, the executable /usr/bin/cat gets executed.
Examples: ls, cat etc.

Password changing (password)

Change Password

All Unix systems require passwords to help ensure that your files and data remain your own and that the system itself is secure from hackers and crackers. Following are the steps to change your password −

Step 1 − To start, type password at the command prompt as shown below.

Step 2 − Enter your old password, the one you're currently using.

Step 3 − Type in your new password. Always keep your password complex enough so that nobody can guess it. But make sure, you remember it.

Step 4 − You must verify the password by typing it again.

```
$ passwd
Changing password for amrood
(current) Unix password:******
New UNIX password:*******
Retype new UNIX password:*******
passwd: all authentication tokens updated  successfully
```

Note − We have added asterisk (*) here just to show the location where you need to enter the current and new passwords otherwise at your system. It does not show you any character when you type.

Knowing who are logged in (who)

who command is used to find out the following information:
1. Time of last system boot.
2. Current run level of the system 3. List of logged in users and
   more.

Description: The who command is used to get information about currently logged in user on to system.

Syntax: $who [options] [filename]

Examples:
1. The who command displays the following information for each user currently logged in to the system if no option is provided:

Login name of the users
Terminal line numbers
Login time of the users in to system Remote
host name of the user hduser@mahesh-
Inspiron-3543:~$ who

```
hduser   tty7        2018-03-18 19:08 (:0)
hduser@mahesh-Inspiron-3543:~$
```

whoami command

whoami command is used both in Unix Operating System and as well as in Windows Operating System.

It is basically the concatenation of the strings “who”,”am”,”i” as whoami.
It displays the username of the current user when this command is invoked.
It is similar as running the id command with the options -un.
The earliest versions were created in 2.9 BSD as a convenience form for who am i, the Berkeley Unix who command’s way of printing just the logged in user’s identity. The GNU version was written by Richard Mlynarik and is part of the GNU Core Utilities (coreutils).

Syntax:

```
akfrgs@HP~: whoami
```

System information using uname

The command ‘uname‘ displays the information about the system.

Syntax:

uname [OPTION]

Options and Examples

1. -a option: It prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system.

Syntax:
```
$uname  -a
```

2. -s option: It prints the kernel name.

Syntax:
```
$uname  -s
```

3. -n option: It prints the hostname of the network node(current computer).

Syntax:
```
$uname  -n
```

4. -r option: It prints the kernel release date.

Syntax:
```
$uname  -r
```

5. -v option: It prints the version of the current kernel.

Syntax:
```
$uname  -v
```

6. -m option: It prints the machine hardware name.

Syntax:
```
$uname  -m
```

7. -p option: It prints the type of the processor.

Syntax:

```
$uname  -p
```

8. -I option: It prints the platform of the hardware.

Syntax:

```
$uname  -i
```

9. -o option: It prints the name of the operating system.

Syntax :

```
$uname  -o
```

File name of terminal connected to the standard input (tty)

Linux operating system represents everything in a file system, the hardware devices that we attach are also represented as a file. The terminal is also represented as a file. There a command exists called tty which displays information related to terminal. The tty command of terminal basically prints the file name of the terminal connected to standard input. tty is short of teletype, but popularly known as a terminal it allows you to interact with the system by passing on the data (you input) to the system, and displaying the output produced by the system.

Syntax:

```
tty [OPTION]....
```

Example:
[i] tty - -version
[ii] sudo tty

Who is Logged in?

Sometime you might be interested to know who is logged in to the computer at the same time. There are three commands available to get you this information, based on how much you wish to know about the other users: users, who, and w.

```
$ users
 amrood bablu qadir

$ who amrood ttyp0 Oct 8 14:10
(limbo) bablu  ttyp2 Oct 4 09:08
(calliope)
qadir  ttyp4 Oct 8 12:09 (dent)

$
```

Try the w command on your system to check the output. This lists down information associated with the users logged in the system.

Logging Out

When you finish your session, you need to log out of the system. This is to ensure that nobody else accesses your files.

To log out

Just type the logout command at the command prompt, and the system will clean up everything and break the connection.

System Shutdown

The most consistent way to shut down a Unix system properly via the command line is to use one of the following commands –

Sr.No.	Command & Description
1	Halt Brings the system down immediately
2	init 0 Powers off the system using predefined scripts to synchronize and clean up the system prior to shutting down
3	init 6 Reboots the system by shutting it down completely and then restarting it
4	poweroff Shuts down the system by powering off
5	Reboot Reboots the system
6	shutdown Shuts down the system