UNIX /LINUX

Importance of Unix for Test Engineers:

ü Most of the Projects Build is deployed in Unix Servers. So it is Tester responsibility to understand Deployment instructions in UNIX.

ü Understanding UNIX will give added advantage to understand the Functionality of more security applications like Banking and Insurance

ü All Product based companies recruiting Test Engineers based on UNIX Knowledge.

Contents

1 Getting Started
History of UNIX
Features of UNIX
  Multiuser Capability
  Multitasking Capability
  Communication
  Security
  Portability
UNIX System Organization
  Shell
  Kernel
  Functions of Kernel
The First Faltering Steps
  who am i
  who
  pwd
  logname
  date
  cal
2 Unix File System
Creating files
  touch
  cat
Copy a file
  cp
Rename a file
  mv
Listing files and directories

ls
Changing file permissions
chmod
Removing a file
rm
Directory related commands
mkdir
rmdir
cd
3 Essential Unix Commands
passwd
File related commands
wc
sort
cut
grep
fgrep
Viewing files
head
tail
4 Process in Unix
What is running right now?
Background processes
Killing a process

History of UNIX:

UNIX is a CUI (Command Unser Interface) operating system which was first developed in the 1960s. **Operating System:** An operating system can be defined as the software that controls the H/W resources of the computer and provides an environment under which programs can run.

UNIX is almost 45 year old OS. Before development of UNIX OS at AT & T Bell labs, s/w team lead by Ken Thomson, Dennis Ritchie and Rudd Canday worked on MULTICS project (Multi Information Computing System) .Initially, MULTICS was developed for only two users. Based on the same concept in 1969, UNICS (Uniplexed Information Computing System) OS was developed for 100's of users. In 1973 named as UNIX. It is open source OS.

Linux almost had same Unix Like feature for e.g.

- Like UNIX, Linux is also written is C.
- Like Unix, Linux is also the Multi-user/Multitasking OS
- Like Unix, Linux runs on different hardware platform (Portable)

Flavours of UNIX:

- Aix by IBM
- Macos by apple
- Red hat linus by red hat s/w
- Solaries by sun solaries

Multitasking Capability

Performing tasks simultaneously rather than sequentially. A multi tasking operating system allows more than one program to be running at a time

Communication

Communication between different terminals

Security

UNIX provides 3 levels of security to protect data.

ü Assigning passwords and login names to individual users.

ü At file level

ü File encryption utility.

Portability:

It can be ported (Transfer from one system to another) to almost any computer system.

The First Faltering Steps:

When you try to access your system, UNIX will display a prompt that looks something like this:

Login:

Password:

$who am i

It displays current user name, terminal number, date and time at which you logged in.

$who

Aa1 tty3a Jan 16 01:25

Ravi tty6c May 22 15:10

Ramana tty3b June 18 10:19

It displays login name, terminal number/serial port, date & time when logged in. note that this shown only for users who are currently logged in.

$pwd

It displays the present working directory.

$logname: It prints user's login name

$date: it displays system date and time (current date and time)

$cal 9 2003

It will display calendar of September 2003.

$cal 2010

It will display calendar of entire year 2010.

UNIX File System:

A file is the basic structure used to store information on the UNIX system. All utilities, applications, data in UNIX is stored as files. Even a directory is treated as a file which contains several other files. An UNIX file system resembles an upside down tree. File system begins with a directory called **root.** The root directory is denoted as slash (/).

Creating files:

$touch sample

This creates a file called sample. **The size of the file would be zero bytes** since **touch does not allow you to store anything in a file.**

Then does touch serve any purpose? Yes, to create several empty files quickly.

$touch sample1 sample2 sample3 sample4

But what if you want to store a few lines in a file. Just type the command

$cat > sample1

******

******

******

Ctrl + d

To append data to the existing file.

$cat >> sample1

Ctrl+d

To view the contents of an existing file.

$cat filename

Copy a file:

Syntax: cp source file target file

$cp sample1 sample2

This will copy contents of sample1 into a sample2. If sample2 already existed it overwrites.

$cp –i sample1 sample2 à if sample2 already existed then it asks the confirmation.

Rename a file:

If you want to rename the file test to sample we would say:

$mv test sample

mv command also has the power to rename directories.

$mv olddir newdir

Note: Moving a file implies removing it from its current location and copying it at a new location.

mv file1 file2 newdir

Listing files and directories

$ls –l

ls -> Show contents of working directory
ls file1 -> list file1, if it exists in working directory
ls dir1 -> show contents of the directory dir1
ls -a -> shows all your files, including hidden ones
ls -al -> give detailed listing of contents
ls *.doc - show all files with suffix ".doc"
ls -lt -> Time of last modification will come first (last modified/created files display first on the screen)
ls –ltr -> Time of last modification will come last.

Changing file permissions:

chmod: chmod is the command to change file permissions or directory permissions.

Permissions	**weight**
r- read	4
w- write	2
x- execute	1

Ex: $chmod 700 filename
u for user or owner
g for group
o for others

Removing a file:

$rm file1

It removes file1, if file permissions permit.

Remove multiple files:

$rm file1 file2 file3

$rm –i filename à i- interactively

Directory related commands:

$mkdir dir1

Make/create directory called dir1 in your working directory.

$mkdir dir1 dir2 dir3 dir4

To create multiple directories.

$mkdir –p dir1/dir2/dir3/dir4

Creates all the parent directories specified in the given path.

$rmdir dir1

It removes directory dir1.

Note: Directories must be empty before you remove them.

To recursively remove nested directories, use the rm command with the **-r option**:

$rm -r directory name

Changing directory:

$cd dirname

$cd .. à To change into parent directory

Essential Unix Commands:

$ passwd

Updates a user authentication.

File related commands:

$wc filename

This command is used to count the number of lines, words & characters from a file.

Options

-l à Lines

-w à words

-c à characters

$wc –l filename

$wc –w filename

$wc –lw filename

$wc –c filename

$wc –l file1 file2 file3

sort command:

1. Sort command can be used for sorting the contents of a file.
2. It can merge multiple sorted files and store the result in the specified output file.
3. Sort can display unique lines.

$sort myfile1

$sort file1 file2 file3

$sort –o myresuly file1 file2 file3 **à** here, with –o option write result to myresult instead of standard output

$sort –u –o result file1 file2 file3 **à -**u option is to display **unique** lines

$sort –m file1 file2 -m à Merge file1 content with file2.

cut Command:

Like sort, cut is also a filter. It cuts or picks up a given number of characters or fields from the specified file. (Here, cut command assumes that fields are separated by tab character).

$cut –f 2 file1

It displays second filed in file1.

$cut –f 2,4 file1

It displays 2,4th fields in file1.

$cut –f 1-5 file1

It displays 1 to 5th fields in file1.

Let us say, each piece of information is separated by a “,” then command would be
$cut –f 1-5 –d”,” file2
It displays 1 to 5th fields in file12.
$cut –c 1-3,5-8 abc
c: character by character.
It displays 1-3 characters and 5-8 characters from file **abc.**

grep command:

Globally search a regular expression.
Syntax: grep "word-to-find" {file-name}.
$grep hyderabad sample1
grep will locate all lines for the " hyderabad " pattern and print all (matched) such line(s) on-screen.
Options
-c à it returns only number of matches.
-i à ignores case while searching.
-v à returns lines that do not match the test.

fgrep Command:

It is almost similar to grep, but by using fgrep you can search for multiple patterns. But it doesn’t allow you to use regular expressions.
$fgrep “string1
> string 2
> string 3” filename

Viewing files:

So far we have used the cat command to view the contents of a file. However, if the file is large in size then the matter would naturally scroll off the screen. To overcome scroll off the screen **head** and **tail** commands help in viewing lines at the beginning or end of the file.
head: Head prints the first N number of data lines of the given input. By default, it prints first 10 lines of each given file.
Syntax: head –n filename
$head -20 file1 à it displays first 20 lines from file1
tail: Tail prints the last N number of lines from given input. By default, it prints last 10 lines of each given file.
Syntax: tail -5 filename
Command: tail -5 file1 à it displays last 5 lines from file1.

Process in UNIX:

Process is kind of program or task carried out by your PC.
"An instance of running command is called **process** and the number printed by shell is called **process-id (PID)**, this PID can be used to refer specific running process."

What is running right now:

$ps

To see currently running process at your terminal.

$ps –a à processes of all the users.

Background processes:

To run command in background, you end it with an &.

Command: cp file1 file2 &

Killing a process:

Kill command is used to terminate the process or kill the process.

Syntax: kill pid