Department of Computer Science and Engineering (Data Science)

Experiment 1
(Linux Commands)

Aim: Explore the internal commands of Linux

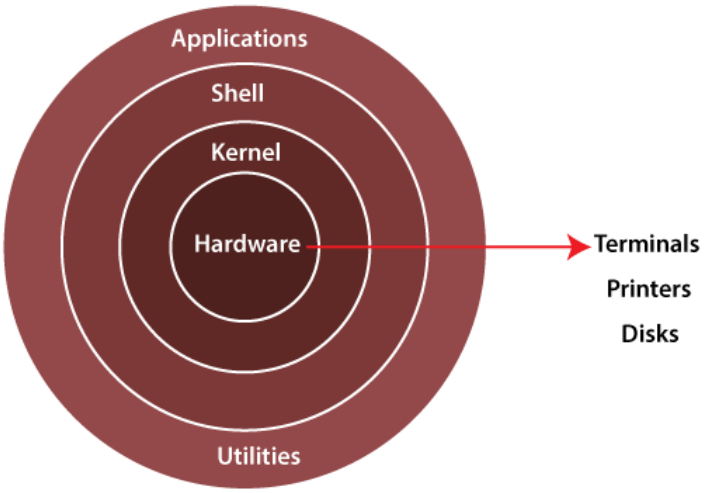
Do the following

1. Display top 10 processes in descending order
2. Display processes with highest memory usage.
3. Display current logged in user and logname.
4. Display current shell, home directory, operating system type, current path setting, current working directory.
5. Display OS version, release number, kernel version.
6. Illustrate the use of sort, grep, awk, etc

Theory:

Linux operating system: An operating system can be described as an interface among the computer hardware and the user of any computer. It is a group of software that handles the resources of the computer hardware and facilitates basic services for computer programs.

Architecture of Linux system



Kernel:- The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS. The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system. There are some of the important kernel types which are mentioned below:

- Monolithic Kernel
- Micro kernels
- Exo kernels
- Hybrid kernels

Shell:- The shell is the Linux command line interpreter. It is an interface among the kernel and user. It can take commands through the user and runs the functions of the kernel.

Linux Commands

A Linux command is a program or utility that runs on the command line. A command line is an interface that accepts lines of text and processes them into instructions for computer.

Department of Computer Science and Engineering (Data Science)

(I) File and Directory Related commands

1) pwd

This command prints the current working directory

2) ls

This command displays the list of files in the current working directory.
$ls –l Lists the files in the long format
$ls –t Lists in the order of last modification time
$ls –d Lists directory instead of contents
$ls -u Lists in order of last access time

3) cd

This command is used to change from the working directory to any other directory specified.
$cd directoryname

4) cd ..

This command is used to come out of the current working directory.
$cd ..

5) mkdir

This command helps us to make a directory.
$mkdir directoryname

6) rmdir

This command is used to remove a directory specified in the command line. It requires the
specified directory to be empty before removing it.
$rmdir directoryname

7) cat

This command helps us to list the contents of a file we specify.
$cat [option][file]
cat > filename – This is used to create a new file.
cat >>filename – This is used to append the contents of the file
Eg:
cat file1
cat file1 file2 > all
cat file1 >> file2

8) cp

This command helps us to create duplicate copies of ordinary files.
$cp source destination

9) mv

This command is used to move files.
$mv source destination

10) ln

This command is to establish an additional filename for the same ordinary file.
$ln firstname secondname

11) rm

This command is used to delete one or more files from the directory.
$rm [option] filename
$rm –i Asks the user if he wants to delete the file mentioned.
$rm –r Recursively delete the entire contents of the directory as well as the directory itself.

(II) Process and status information commands

1) who

This command gives the details of who all have logged in to the UNIX system currently.
$ who

2) who am i

This command tells us as to when we had logged in and the system's name for the
connection being used.
$who am i
8

3) date

This command displays the current date in different formats.
+%D mm/dd/yy +%w Day of the week
+%H Hr-00 to 23 +%a Abbr.Weekday
+%M Min-00 to 59 +%h Abbr.Month
+%S Sec-00 to 59 +%r Time in AM/PM
+%T HH:MM:SS +%y Last two digits of the year

4) echo

This command will display the text typed from the keyboard.
$echo
Eg: $echo Have a nice day
O/p Have a nice day

(III) Text related commands

1. head

This command displays the initial part of the file. By default it displays first ten lines of the
file.
$head [-count] [filename]

2. tail

This command displays the later part of the file. By default it displays last ten lines of the
file.
$tail [-count] [filename]

3. wc

This command is used to count the number of lines, words or characters in a file.
wc -l <filename> print the line count
wc -c <filename> print the byte count
wc -m <filename> print the character count
wc -L <filename> print the length of longest line
wc -w <filename> print the word count

4. find

The find command is used to locate files in a directory and in a subdirectory.
The –name option
This lists out the specific files in all directories beginning from the named directory. Wild cards can be used.

The –type option
This option is used to identify whether the name of files specified are ordinary files or directory files. If the name is a directory, then use "-type d" and if it is a file then use "-type f".
The –mtime option
This option will allow us to find that file which has been modified before or after a specified time. The various options available are –mtime n(on a particular day),-mtime +n(before a particular day),-mtime –n(after a particular day)
The –exec option
This option is used to execute some commands on the files that are found by the find command.

(IV) File Permission commands

1) chmod
Changes the file/directory permission mode:
$chmod [option] mode files

options:
-R Descend directory arguments recursively while setting modes.
-f Suppress error messages if command fails.
mode:
Who u=user, g=group, o=other, a=all (default)
Opcode
+ means add permission
— means remove permission
= means assign permission and remove the permission of unspecified fields
Permission r=Read, w=write, x=Execute

Eg. $ chmod 777 file1
Gives full permission to owner, group and others
$ chmod o-w file1
Removes write permission for others.

2) chgrp
chgrp user file Makes file belong to the group user.

3) chown
chown cliff file Makes cliff the owner of file.
chown -R cliff dir Makes cliff the owner of dir and everything in its directory tree.

(V) Other Useful Commands:

1) exit - Ends your work on the UNIX system.
9

2) Ctrl-l or clear
Clears the screen.

3) Ctrl-c
Stops the program currently running.

4) Ctrl-z
Pauses the currently running program.

5) man COMMAND
Looks up the UNIX command COMMAND in the online manual pages.

6) history
List all commands typed so far.

7) more FILE
Display the contents of FILE, pausing after each screenful.
There are several keys which control the output once a screenful has been printed.
<enter> Will advance the output one line at a time.
<space bar> Will advance the output by another full screenful.
"q" Will quit and return you to the UNIX prompt.

8) less FILE
"less" is a program similar to "more", but which allows backward movement in the file as well as forward movement.

9) lpr FILE
To print a UNIX text or PostScript file, type the following command at the system prompt:

(VI) Meta characters

Some special characters, called metacharacters may be used to specify multiple filenames. These characters substitute filenames or parts of filenames.

1) The "*"
This character is used to indicate any character(s)

2) $ cat ap*
This displays the contents of all files having a name starting with ap followed by any number of characters.

3) The "?" This character replaces any one character in the filename.
$ ls ?st
list all files starting with any character followed by st.

4) The [] These are used to specify range of characters.
$ ls [a-z]pple
Lists all files having names starting with any character from a to z.

(VII) Absolute path and relative path

Generally, if a command is given it will affect only the current working directory. For example, the following command will create a directory named curr in the current working directory.

1) **$ mkdir curr**

The directory can also be created elsewhere in the file system using the absolute and relative path.If the path is given with respect to the root directory, then it is called **full path or absolute path**
$ mkdir /home/it2006/it2k601/curr

The full path always starts with the /, which represents the root directory.
If the path is given with respect to the current working directory or parent directory then it is called **relative path.**
$ mkdir ../curr

The above command will create a directory named curr in the parent directory.
$ mkdir ./first/curr

The above command will create a directory named curr inside first directory, where the directory first is located in the current working directory.
Note "." Represents current directory and ".." represents parent directory.

(VIII) PIPES AND FILTERS

In UNIX commands were created to perform single tasks only. If we want to perform multiple tasks we can go for pipes and filters.

PIPES

A pipe is a mechanism which takes the output of a command as its input for the next command. For example, list the numbers of users currently login in the system and then sort it.
ls who | sort

$who | wc –l
$cat text.c | head –3

FILTERS

Filters are used to extract the lines, which contain a specific pattern, to arrange the contents of a file in a sorted order, to replace existing characters with some other characters, etc.

1. Sort filter

The sort filter arranges the input taken from the standard input in alphabetical order. The
sort command when used with "-r" option will display the input taken from the keyboard in the reverse alphabetical order. When used with "-n" option arranges the numbers, alphabets and special characters according to their ASCII value. If we want to sort on any one field, then sort provides us with an option called "+pos1 –pos2" option.

2.Grep filter

This command is used to search for a particular pattern from a file or from standard input and display those lines on the standard output. Grep stands for "Global search for regular expression".
There are diverse options available with grep command.
-v displays only those lines which do not match the pattern specified.
-c displays only the count of those lines which match the pattern specified

-n displays matched lines with line numbers
-i displays matched pattern ignoring case distinction
Eg: grep -c tech file1

3.Uniq filter

The uniq filter compares adjacent lines in the sorted input file and when used with different options displays single and multiple occurrences.
-d displays only the lines which are duplicated in the input file.
-u displays only the lines with single occurrences.

4.Pg and more filter

These commands display the output of the command on the screen page by page. The difference between pg and more filter is that the viewing screen of the latter can be done by pressing space bar while that of the former is done by pressing enter.

5.Cut command

One particular field from any file or from output of any command can be extracted and displayed using this cut command. One particular character can also be extracted using the –c option of this command.

6.Tr command

This command is used to translate characters taken from the standard input. This command when used with "-s" option is used to squeeze multiple spaces into a single space.

Additional Commands

rmdir

This is the opposite of 'mkdir'- which is used to delete the directories. It should be pointed out that in order to use it, the directory must be empty.
'rm' command for removing files in the directory.

who

This is used to find out who's working on this system. Linux is a multiuser system.
For example, if we logged in as 'root' but are working as 'nitin'.
root tty1 May 20 09:48 'root' started working on terminal 1 on May 20 at 9:48
nitin tty2 May 20 10:05 *nitin* started working on terminal 2 at 10:05

whoami

It is a little program that tells us user names.
$whoami

ps

It gives us a list of the processes running on the system.
$ps

editing files:

vi filename Edit a file using the vi editor. All UNIX systems will have vi in some form.
emacs filename Edit a file using the emacs editor. Not all systems will have emacs.

***man* command**

Shows manual (description) for given command.
$man ls

Department of Computer Science and Engineering (Data Science)

Lab Assignments to complete in this session

1. Display top 10 processes in descending order

ps|head

2. Display processes with highest memory usage.

ps -eo pid,%mem --sort=-%mem |head

3. Display current logged in user and logname.

Whoami
who | cut -d' ' -f1 | sort | uniq
id

4. Display current shell, home directory, operating system type, current path setting, current working directory.

$SHELL
$ cd ~
lsb_release -a
cat /proc/version
$PATH
PWD

5. Display OS version, release number, kernel version.

lsb_release –a à OS version on Linux
uname –r à kernel version.

6. Illustrate the use of sort, grep, awk, etc.

sort file1.txt
grep “datascience” FE_List.xml