Week 1

Basic Unix commands

Introduction to Unix

An OS providing both command line interface (CLI) and graphical user interface (GUI) based interaction integrated by Dennis Ritchie, Douglas Mcllroy, Joe Ossanna Brian Kernighan, and Ken Thompson at Bell laboratory in 1970 called a multi-tasking operating system permitting two or more users to simultaneously operate on the operating system and offers commands for the users for interacting with the application from command line interface, such as sudo command, chmod command, su command, mv command, rm command, vi command, cat command, rmdir command, mkdir command, clear command, and ls comma nd, which can be used to implement complex tasks.

Hence, it's recommended by professionals who operate with servers; it's also recommended to learn how the command-line-based operating system works. Many large and complex applications that utilize Unix to execute because of its aspect to handle the processes easily. It's a bit faster and provides a nice user experience when compared with the Windows operating system.

Process of Unix:

A program runs in a process. Each time when a program or command is run, a fresh process is established. The process is running for as long as the command is in an active state. For instance, if we are running the command, i.e., cat, the cat process is generated.

The kernel assigns a special identification number known as the PID or process identification number, which ranges between 0 to 32,767 every time a fresh process is established. Other process properties include their GID (group related to the process), UIS (user id owns the process), TTY (controlling terminal through where they're launched), and PPID (the parent PID).

Types in Unix

o Foreground Process: A foreground process is launched through a terminal and denies further commands until it ends. By default, the stdout and stdin are associated with the terminal in such a process.

o Background Process: This process is launched through a terminal but is executed in the background. Hence, permitting further commands while it executes. The stdout and stdin should be typically redirected, so they do not interfere with another foreground process.

o Daemon Process: It's a process that's not related to the terminal session. Usually, such processes are released for system services like printing and networking.

Unix users commands

These commands allow you to get basic information about Unix users in your environment.

whoami – show your username

id – print user identity

groups – show which groups user belongs to

passwd – change user password

who – find out who is logged into the system

last – show history of logins into the system

Unix file operations

Navigating filesystem and managing files and access permissions:

nslookup google.com – server address

cp – copy files (work in progress)

rm – remove files and directories (work in progress)

mv – rename or move files and directories to another location

echo – "hello world"

chown – change file/directory ownership

Unix directory management commands

Navigating filesystems and managing directories:

- cd – change directory
- pwd – confirm current directory
- mkdir – make new directory
- rmdir – remove directories in Unix

Main features of unix :

multiuser - More than one user can use the machine

Multitasking- More than one program can be run at a time.

Portability – This means the operating system can be easily converted to run on different browsers.

Control Commands:

These commands are a two-key combination where a letter is pressed simultaneously with the 'Ctrl' key.

- **Control-C:** This command terminates the currently running foreground process.
- **Control-D:** This command terminates the currently running login or terminal
- **Control-Z:** This command suspends the currently running foreground process to the background.

Command	**ps - displays a snapshot of all current processes**
Common Syntax	$ ps [options]
Example	$ ps -ef Show every process running, formatted as a table

Command	**top - displays a live status of current processes**
Common Syntax	$ top [options]
Example	$ top Show a live view of all current processes

Command	**bg - resume a background suspended a job**
Common Syntax	$ bg [job_spec …]

Example	$ xterm
	Ctrl-Z
	$ bg
	Continue running a job that was previously suspended (using Ctrl-Z) in the b

Command	**clear – clear a terminal screen**
Common Syntax	$ clear
Example	$ clear
	Clear all prior text from the terminal screen
Command	**history – print history of commands in the current session**
Common Syntax	$ history [options]
Example	$ history

#1) touch: Create a new file or update its timestamp.

- Syntax: touch [OPTION]…[FILE]
- Example: Create empty files called 'file1' and 'file2'
- $ touch file1 file2

#2) cat: Concatenate files and print to stdout.

- Syntax: cat [OPTION]…[FILE]
- Example: Create file1 with entered cotent
- $ cat > file1
- Hello
- ^D

#3) cp: Copy files

- Syntax: cp [OPTION]source destination
- Example: Copies the contents from file1 to file2 and the contents of file1 are retained
- $ cp file1 file2

#4) mv: Move files or rename files

- Syntax: mv [OPTION]source destination
- Example: Create empty files called 'file1' and 'file2'
- $ mv file1 file2

#5) rm: Remove files and directories

- Syntax: rm [OPTION]…[FILE]
- Example: Delete file1

- $ rm file1

#6) mkdir: Make a directory

- Syntax: mkdir [OPTION] directory
- Example: Create a directory called dir1
- $ mkdir dir1

#7) rmdir: Remove a directory

- Syntax: rmdir [OPTION] directory
- Example: Create empty files called 'file1' and 'file2'
- $ rmdir dir1

#8) cd: Change directory

- Syntax: cd [OPTION] directory
- Example: Change working directory to dir1
- $ cd dir1

#9) pwd: Print the present working directory

- Syntax: pwd [OPTION]
- Example: Print 'dir1' if a current working directory is dir1

#10) ls: List directory contents

- Syntax: ls [OPTION] [FILE]
- Example: list all (including hidden files) directory contents, in long format, sorted by
- $ ls -alt
    - $ls -l
    - $ls -ltr
    - $ls -a

#11) which: Locate a command

- Syntax: which [-a] filename
- Example: List all paths from where 'cat' can run
- $ which -a cat

#12) man: Interface for working with the online reference manuals.

- Syntax: man [-s section] item
- Example: Show the manual page for the 'cat' command
- $ man cat

#13) su: Change user-id or become super-user.

- Syntax: su [options] [username]
- Example: Change user-id to 'user1' (if it exists)
- $ su user1

#14) sudo: Execute a command as some other user or super-user

- Syntax: sudo [options] [command]
- Example: Get a file listing of an unlisted directory
- $ sudo ls /usr/local/protected

#15) find: Used to search for files and directories as mentioned in the 'expression'

- Syntax: find [starting-point] [expression]
- Example: In '/usr' folder, find character device files, of the name 'backup'
- $ find /usr -type c -name backup

#16) du: Estimate disk usage is blocks

- Syntax: du [options] [file]
- Example: Show the number of blocks occupied by files in the current directory
- $ du

#17) df: Show the number of free blocks for the mounted file system

- Syntax: df [options] [file]
- Example: Show the number of free blocks in local file systems
- $ df -l

#18) cal: Displays the calendar.

- Syntax: cal [[month] year]
- Example: display the calendar for April 2018
- $ cal 4 2018

#19) date: Displays the system date and time.

- Syntax: date [+format]
- Example: Display the date in dd/mm/yy format
- $ date +%d/%m/%y

#20) who: Displays the list of users currently logged in

- Syntax: who [option] … [file][arg1]

- Example: List all currently logged-in users
- $ who

#21) whoami: Displays the user id of the currently logged-in user.

- Syntax: whoami [option]
- Example: List currently logged-in user
- $ whoami

Questions:

1) How to Create a File?

2) How to view the list of Commands executed?

3) How to view your username in Unix?

4) List some of the common and most widely used UNIX commands.

5) How to Remove files using unix commands?

6) Create a RGUKT new file

7) How can you use grep along with regular expressions in a shell script to extract lines from a log file that match a specific pattern, such as timestamps?