**LINUX NOTES: MODULE 1**

Components of Linux system:

1. Bootloader: starting the Linux kernel
2. Kernel: software that interfaces directly with the computer hardware.
3. Init System: first process to run once the kernel is loaded. Init is a daemon program that acts as the parent process to all other processes running on the system.
4. Daemons: program that runs in the background. Example: web server running on a Linux server depends on a daemon, usually named httpd. We cannot interact with a daemon process.
5. Graphical Server
6. Desktop environment
7. Applications: downloaded after Linux installation.

COMMANDS

1. Pwd: print current working directory
2. Cd: change directory
3. Ls: list files and directories
4. Touch: to create a new file. Without using touch also we can create new file when we do nano other the file.
5. Mkdir: to create a new directory.
6. Rm: delete file and directory. You can never recover file.

For file: rm filename

For directory: rm -r directory\_name

1. Cp: copies a file or directory. Original file also exists.
2. Mv: moving or directories. Original file is lost.
3. Cat: to print
4. Head: print first 10 lines by default. Else head -n 6 filenames.(-n argument to specify number)
5. Tail: print last 10 lines by default. Else tail -n 6 filename.
6. More: print one screen plus option of navigation forward.
7. Less: print one screen plus option of navigation forward and backward.
8. Echo: to print
9. Grep: grep “thing” file.
10. Sed: replace. sed 's/<text\_to\_replace>/<replacement\_text>/' <file\_name>
11. Sort: sorts file lexicographically. Contents of file do not change.

To sort numbers:

Asc use -n

Dec use -nr

1. Uniq: prints only unique elements.
2. Whoami: to find and print current user
3. Useradd/groupadd: create new user and group
4. Passwd/.gpasswd: add or modify pass
5. Usermod: modify user
6. Userdel/groupdel: del user/ del group

PERMISSIONS OF A FILE

---(user) ---(group) ---(others)

Rwe(user)

1. Chmod: to change permissions
2. Chown: to change owner of file or directory. Syntax: chown <new\_owner> <file\_name>
3. Chgrp: to change group owner. Syntax: chown jaisa

PROCESS MANAGEMENT:

1. Ps: process status. Syntax: ps or ps aux or ps -p 98(id) or ps | grep -I ‘java’
2. Top: process running in real time.

MEMORY MANAGEMENT:

1. Free: free -h. displays total, free, used memory in ram.
2. Df: disk free.
3. Du: disk usage. Display usage of files and directories.

KERNEL VS USER MODE

KERNEL USER MODE

|  |  |  |
| --- | --- | --- |
| **Kernel-mode vs User mode** | In kernel mode, the program has direct and unrestricted access to system resources. | In user mode, the application program executes and starts. |
| **Interruptions** | In Kernel mode, the whole operating system might go down if an interrupt occurs | In user mode, a single process fails if an interrupt occurs. |
| **Modes** | Kernel mode is also known as the master mode, privileged mode, or system mode. | User mode is also known as the unprivileged mode, restricted mode, or slave mode. |
| **Virtual address space** | In kernel mode, all processes share a single virtual address space. | In user mode, all processes get separate virtual address space. |
| **Level of privilege** | In kernel mode, the applications have more privileges as compared to user mode. | While in user mode the applications have fewer privileges. |
| **Restrictions** | As kernel mode can access both the user programs as well as the kernel programs there are no restrictions. | While user mode needs to access kernel programs as it cannot directly access them. |
| **Mode bit value** | The mode bit of kernel-mode is 0. | While; the mode bit of user-mode is 1. |

THE SHELLS

1. The Bourne Shell (sh): the first shell.

Disadvantage: cannot recall commands, cannot handle logical and arithmetic operations.

1. The GNU Bourne Again shell: u can recall commands
2. The C Shell(csh): arithmetic possible and command history bhi h
3. The Korn Shell(ksh): similar to c prog lang
4. The Z Shell(zsh): modern shell with customization

ENVIRONMENT VARIABLES:

1. Global: is defined in a terminal can be accessed from anywhere in that particular environment which exists in the terminal. That means it can be used in all kind of scripts, programs or processes running in the environment bound by that terminal.

Print: env

1. Local: is defined in a terminal cannot be accessed by any program or process running in the terminal. It can only be accessed by the terminal (in which it was defined) itself.
2. Access both: echo $NAME
3. Print both: set
4. Set a env: set NAME=value
5. Unset: unset NAME