Session 04:

**PWD : present working directory**

**ls -l long list (ll) - print in alphabetical order**

**ls -lr print in reverse order**

**ls -lt last modified time**

**ls -ltr reverses order with recently created**

**Absalute path :**

**Start with /**

**Eg ; Cd /var/log**

**/home/user/ducuments**

**No matter where you are in the system , the path will always point to the mentioned location**

**Relative path :**

**It doen’t start with /**

**If we are in /home/user and we need to type cd doument to reach document**

**Create/remove a directory :**

**mkdir - make a directory**

**rmdir - remove directory , if not empty**

**rm -r <directory name> : remove even if the directory not empty (recursive)**

**Create/delete a file :**

**touch filename.txt**

**touch filename01 filename02 filename03**

**touch filename{01..10}.txt**

**Cat > filename.txt -> add the content -> ctrl+D**

**List the file item**

**Cat <fiename>**

**Edit the file :**

**Copy : cp : cp <filename> <foldername>**

**Copy all file : cp \* destination\_directory**

**cp \*.txt /destination/directory/ - copy all .txt file to destination**

**cp /source/directory/\*.log /destination/directory/**

**✅ If you want to copy everything inside /source/directory into /destination/directory/ → use:**

**cp -r /source/directory/. /destination/directory/**

**✅ If you want to copy the whole folder itself → use:**

**cp -r /source/directory /destination/directory/**

**Move : mv <filename> <foldername> , similar to cut and paste**

**mv /source/directory/file.txt /destination/directory/**

#### ****Move a full directory:****

**mv /source/directory/ /destination/directory/**

**Can edit using cat and Vi/Vim/nano editer**

**Rename :**

**mv old\_name new\_name**

**VI:**

## **Insert Mode Commands (type text)**

| **Command** | **Action** |
| --- | --- |
| i | Insert before cursor |
| I | Insert at the beginning of the line |
| a | Append after cursor |
| A | Append at the end of the line |
| o | Open a new line below |
| O | Open a new line above |

## 💾 **Save / Quit Commands (press** : **first)**

| **Command** | **Action** |
| --- | --- |
| :w | Save (write) file |
| :q | Quit |
| :wq or ZZ | Save and quit |
| :q! | Quit without saving |
| :x | Save and quit (same as :wq) |

**Open with VIM**

**Vim <filename>**

**Grep:**

**Filter the specific things**

**Grep “word” <filename>**

**cat <filename> | grep <specific word>**

**Log filter eg : /var/log$ cat syslog | grep "man-db"**

**Create a hidden file/directory and find them**

**File : touch .<filename> , cat >.<filename>**

**Directory : mkdir .<dirname>**

**Printn hidden files/directories : ls -a**

**Eg : tmp$ cat >.hiddensecret**

**/tmp$ ls -a**

**Find:**

**find / -name "filename"**

**Count :**

**compgen -c | wc -l**

**ls | wc -l**

**Wc : word cound**

**l : Lines only**

**Disk details:**

**Df -h**

**File permission :**

| **Permission** |  |
| --- | --- |

|  |  |
| --- | --- |
| r | Read |

|  |  |
| --- | --- |
| w | Write |

|  |  |
| --- | --- |
| x | Execute |

**ls -l <filename> to see the permission**

| **Permission** | **Value** |
| --- | --- |
| Read (r) | 4 |
| Write (w) | 2 |
| Execute(x) | 1 |

**chmod 755 file**

**means:**

**User: 7 (4+2+1) → rwx**

**Group: 5 (4+0+1) → r-x**

**Others: 5 (4+0+1) → r-x**

**Full permission chmod 777**

**Directory permission:**

**ls -ld <directory\_name>**

**File permission**

**ls -l <filename>**

**Chmod 777 <directry\_name>**

**Users and Groups**

**Create new user : sudo useradd -m <username>**

**Set password : sudo passwd <username>**

**Group creation : sudo groupadd <groupname>**

**Add user to group : sudo usermod -aG <groupname> <username>**

Group exists - /etc/group(file)

azureuser@test:/etc$ grep "mygroup" group

mygroup:x:1002:user1

**Shell** is a program that provides an interface for users to interact with the operating system. In simpler terms, the shell acts as a command interpreter, translating the text-based input into actions.

Put the multiple commands and provide the extension as **sh**

**Bash** (Bourne Again Shell) : **Bash** is a specific type of shell, and it is the most commonly used shell in Linux and macOS systems. It’s an improved version of the original **Bourne Shell (sh)**

Sample script :

Create a file : cat >myscript.sh

ls

date

Ctrl+D

chmod +x myscript.sh - execute permission

./myscript.sh

Web deployment :

sudo apt update

sudo apt install nginx -y (-y : pre confirmation)

sudo systemctl status nginx : check nginx status

[ systemctl is a **command-line utility**

|  |  |
| --- | --- |
| systemctl start nginx | Starts the nginx service. |

|  |  |
| --- | --- |
| systemctl stop nginx | Stops the nginx service. |

|  |  |
| --- | --- |
| systemctl restart nginx | Restarts the nginx service. |

|  |  |
| --- | --- |
| systemctl status nginx | Shows the status of the nginx service (running, failed, etc.). |

]

git clone <https://github.com/Soorajskr/Starbucks.git>

cd Starbucks

sudo cp -r \* /var/www/html/

ls /var/www/html

Allow port 80 in NIC level

<http://publicip>

# **Basic System Commands**

### **lscpu**

* ****Description:**** Displays detailed information about the CPU architecture.
* ****Usage:****

lscpu

* ****Output:****
  + CPU architecture (x86\_64, i386, etc.)
  + Number of CPUs
  + Model name
  + CPU speed (MHz)
  + Cache sizes
  + Vendor ID

### **free -h**

* ****Description:**** Displays the amount of free and used memory in the system (RAM) in a human-readable format.
* ****Usage:****

free -h

* ****Output:****
  + Total, used, and free memory
  + Shared memory
  + Buffers/cache memory
  + Available memory

### **df -h**

* ****Description:**** Reports the amount of disk space used and available on mounted filesystems in a human-readable format.
* ****Usage:****

df -h

* ****Output:****
  + Filesystem name
  + Size of each filesystem
  + Used space
  + Available space
  + Percentage of used space
  + Mount point

### **cat /etc/os-release**

* ****Description:**** Displays information about the operating system.
* ****Usage:****

cat /etc/os-release

* ****Output:****
  + NAME: The name of the OS (e.g., Ubuntu, Fedora)
  + VERSION: The version of the OS
  + ID: The ID of the OS
  + PRETTY\_NAME: A human-readable name of the OS
  + Other OS-related details

### **hostname -I**

* ****Description:**** Shows the IP address(es) assigned to the machine.
* ****Usage:****

hostname -I

* ****Output:****
  + The machine's IP address(es)

### **uptime**

* ****Description:**** Shows how long the system has been running and the system load.
* ****Usage:****

uptime

* ****Output:****
  + Current time
  + How long the system has been up
  + Number of users currently logged in
  + System load averages for the past 1, 5, and 15 minutes

### **top**

* ****Description:**** Provides a dynamic real-time view of the system's running processes.
* ****Usage:****

top

* ****Output:****
  + PID: Process ID
  + USER: User that owns the process
  + PR: Priority of the process
  + NI: Nice value of the process
  + VIRT: Virtual memory used by the process
  + RES: Resident memory used by the process
  + SHR: Shared memory used by the process
  + S: Process status (running, sleeping, etc.)
  + %CPU: CPU usage
  + %MEM: Memory usage
  + TIME+: Total CPU time used by the process
  + COMMAND: Command name/line that started the process

### **ps**

* ****Description:**** Displays information about the currently running processes.
* ****Usage:****

ps [options]

* ****Common Options:****
  + ps aux: Shows detailed information about all running processes.
  + ps -ef: Displays all the processes in a full-format listing.
* ****Output:****
  + USER: User that owns the process
  + PID: Process ID
  + %CPU: CPU usage
  + %MEM: Memory usage
  + VSZ: Virtual memory size
  + RSS: Resident set size (physical memory used)
  + TTY: Terminal associated with the process
  + STAT: Process state
  + START: Time the process started
  + TIME: Cumulative CPU time
  + COMMAND: Command that started the process

### **kill -9**

* ****Description:**** Forces the termination of a process using its PID (Process ID).
* ****Usage:****

kill -9 [PID]

* ****Example:****

kill -9 1234

* ****Note:**** Use ps or top to find the PID of the process you want to kill.

pkill : kill the process

Eg: pkill sleep : sleep is one process

### **du -h**

* ****Description:**** Estimates file and directory space usage in a human-readable format.
* ****Usage:****

du -h [directory]

* ****Example:****

du -h /var/log

* ****Output:****
  + The size of each directory and subdirectory within the specified directory in human-readable format (KB, MB, GB)

These commands are essential for monitoring and managing system performance, processes, and resources in a Linux environment.