

Name: Kiran Yadav

Email: kiranyadav1102003@gmail.com

Course Name: Devops and Cloud Computing Cours

Assignment Name: Linux

Git Link: <https://github.com/Hydra-Dev110/Linux>

Drive Link:

https://docs.google.com/document/d/1xnSN4sTER_nrXpKICDnhrCetZNbWpCcLmJS-QL3gtgE/edit?usp=sharing

1. What is Linux?

-> Linux is an operating system unix like operating system kernel that servers as the foundation for a wide range of operating systems known as Linux distributions(or distros). These operating systems are used on a wide variety of devices from servers and desktops to smartphone and embedded systems.

Key Features:

- Open Source
- Stable and Secure
- Multitasking and Multiuserds
- Customizable.

2. What is the difference between Hard Link & Soft Link?

-> **Hard Link** : It is essentially another name(alian) for an existing file. It points directly to the file's inode(the data

structure that stores the actual file content and metadata).

Key Characteristics:

- Stores the same inode number as the original file.
- No distinction between the original file and the hard link both are equal.
- If the original file deleted the data still exists as long as one hard link remains.
- Cannot span across different file systems or partitions.

- Cannot link to directories(to prevent loops in the filesystem).

Example : In original.txt hardlink.txt

Soft Link(Symbolic Link or Symlink): A Soft Link is a special file that contain a path to another file or directory. It like

a shortcut.

Key Characteristics:

- Points the file name/path, not the inode
- Different inode number than the original file.
- Can link directories.
- Can span across file systems or partitions.
- If the original file is deleted, the symlink becomes broken or dangling link.

3.What is a Kernel in Linux?

-> In Linux(and all operating systems) the kernel is the core component that acts as a bridge between the hardware and the software. It manages systems, resources and allows software to communicate with the hardware in a controlled and efficient manner.

Key Responsibilities of kernel.

1. Process Management.
2. Memory Management.
3. Device Management.
4. File System Management.
5. System Calls Interface.

4. How do you create a user account?

-> Using adduser:

```
hydra_02@DESKTOP-1FI938H:/$ sudo adduser abhi
Adding user 'abhi' ...
Adding new group 'abhi' (1005) ...
Adding new user 'abhi' (1002) with group 'abhi' ...
Creating home directory '/home/abhi' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for abhi
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

5. What is the 'grep' command used for in Linux?

-> The grep command in Linux is used to search for text patterns in files or command output.

- Finds lines in a file that match a word, phrase, or pattern.
- Supports **regular expressions** for advanced searching.
- Can be used with other commands to **filter output**.

-Basic syntax of grep

grep[options] pattern filename

Example :- grep "Hello" testfile.txt

6. Step1: Create user p1

Step2: He should be part of 3 groups g1,g2,g3.

Step3: whenever he creates a file automatically in the group section of file grp g1 should come.

-> Step 1: Create the user p1

```
hydra_02@DESKTOP-1FI938H:/mnt/c/WINDOWS/system32$ sudo adduser p1
Adding user `p1' ...
Adding new group `p1' (1001) ...
Adding new user `p1' (1001) with group `p1' ...
Creating home directory `/home/p1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for p1
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
```

Step 2: Create groups and add p1 to them

```
hydra_02@DESKTOP-1FI938H:~$ sudo groupadd g1
hydra_02@DESKTOP-1FI938H:~$ sudo groupadd g2
hydra_02@DESKTOP-1FI938H:~$ sudo groupadd g3
hydra_02@DESKTOP-1FI938H:~$ sudo usermod -g g1 -aG g2,g3 p1
```

Step 3: Ensuring files created by p1 belongs to g1

```
hydra_02@DESKTOP-1FI938H:~$ su - p1
Password:
p1@DESKTOP-1FI938H:~$ touch testfile
p1@DESKTOP-1FI938H:~$ ls -l testfile
-rw-r--r-- 1 p1 g1 0 May 15 19:19 testfile
```

7. You suspect that a particular process is consuming excessive CPU resources on your Linux server. How would you identify and terminate this process?

-> To identify the Process using High CPU (use top for monitoring)

```
top - 07:43:22 up 1:28, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 37 total, 1 running, 36 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 5812.9 total, 5289.3 free, 349.1 used, 174.4 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 5254.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	165808	11264	8388	S	0.3	0.2	0:00.88	systemd
2	root	20	0	2776	1920	1796	S	0.0	0.0	0:00.22	init-systemd(Ub
9	root	20	0	2776	132	132	S	0.0	0.0	0:00.00	init
62	root	19	-1	105196	29592	28544	S	0.0	0.5	0:00.53	systemd-journal
91	root	20	0	21964	5700	4456	S	0.0	0.1	0:00.39	systemd-udev
93	systemd+	20	0	25664	12548	8116	S	0.0	0.2	0:00.22	systemd-resolve
94	systemd+	20	0	89364	6568	5764	S	0.0	0.1	0:00.36	systemd-timesyn
166	root	20	0	4388	2704	2468	S	0.0	0.0	0:00.02	cron
167	message+	20	0	8584	4696	4152	S	0.0	0.1	0:00.26	dbus-daemon
173	root	20	0	30136	19112	10320	S	0.0	0.3	0:00.15	networkd-dispat
174	syslog	20	0	222404	5128	4568	S	0.0	0.1	0:00.08	rsyslogd
177	root	20	0	15332	7596	6624	S	0.0	0.1	0:00.20	systemd-logind
192	root	20	0	3240	1120	1028	S	0.0	0.0	0:00.00	agetty
197	root	20	0	3196	1116	1028	S	0.0	0.0	0:00.00	agetty
207	root	20	0	107164	21180	13076	S	0.0	0.4	0:00.11	unattended-upgr
271	root	20	0	2784	204	80	S	0.0	0.0	0:00.00	SessionLeader
272	root	20	0	2784	208	80	S	0.0	0.0	0:00.32	Relay(273)
273	hydra_02	20	0	6120	5088	3332	S	0.0	0.1	0:00.06	bash
274	root	20	0	7528	4936	4016	S	0.0	0.1	0:00.01	login
325	hydra_02	20	0	16932	9352	7884	S	0.0	0.2	0:00.07	systemd
326	hydra_02	20	0	168736	3352	16	S	0.0	0.1	0:00.00	(sd-pam)
331	hydra_02	20	0	6100	4752	3192	S	0.0	0.1	0:00.01	bash
430	root	20	0	8096	4736	3896	S	0.0	0.1	0:00.03	su
432	pl	20	0	16040	9388	7920	S	0.0	0.2	0:00.26	systemd
433	pl	20	0	168736	3352	16	S	0.0	0.1	0:00.00	(sd-pam)
438	pl	20	0	6116	5096	3344	S	0.0	0.1	0:00.01	bash
490	root	20	0	8092	4684	3844	S	0.0	0.1	0:00.03	su
491	hydra_02	20	0	6116	5160	3404	S	0.0	0.1	0:00.04	bash
576	root	20	0	8092	4744	3904	S	0.0	0.1	0:00.04	su
577	pl	20	0	6116	5096	3348	S	0.0	0.1	0:00.03	bash
680	root	20	0	8092	4736	3896	S	0.0	0.1	0:00.06	su
681	hydra_02	20	0	6212	5232	3472	S	0.0	0.1	0:00.14	bash
808	root	20	0	8092	4672	3832	S	0.0	0.1	0:00.03	su
809	abhi	20	0	6248	5196	3440	S	0.0	0.1	0:00.04	bash

To terminate the process(kill (PID))

Kill (PID) will terminate the process.

Example: kill 5302.