

Linux Commands

1. LS Command

What is LS?

ls is a basic Linux command used to list files and directories in the current working directory.

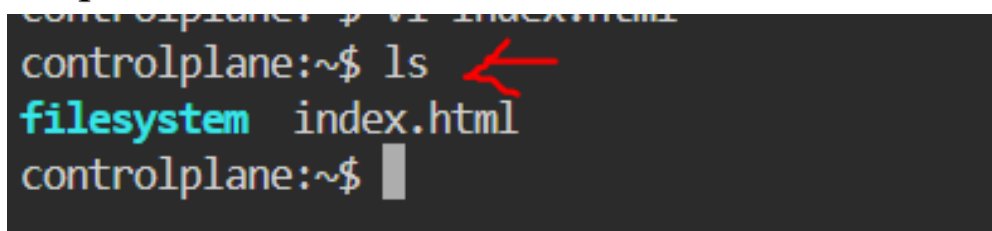
Why LS?

The ls command helps users see what files and folders are present before performing operations like opening, editing, or deleting them.

Purpose

- List files and directories
- View file structure
- Confirm file creation or deletion

Output:

A terminal window with a dark background. The prompt is 'controlplane:~\$'. The command 'ls' is entered, and the output is 'filesystem index.html'. A red arrow points from the 'ls' command to the output. The prompt 'controlplane:~\$' is shown again at the bottom.

```
controlplane:~$ ls  
filesystem index.html  
controlplane:~$
```

2. touch index.html

What is touch?

touch is a Linux command used to create an empty file.

Why touch?

It allows users to quickly create new files without opening an editor.

Purpose

- Create empty files
- Update timestamp of existing files

Output:

```
controlplane:~$ touch index.html
controlplane:~$ ls
filesystem index.html
```

3. cat index.html

What is cat?

cat displays the content of a file in the terminal.

Why cat?

Used to quickly read file contents without opening an editor.

Purpose

- View file contents
- Combine files

Output:

```
controlplane:~$ cat index.html
controlplane:~$ ls
```

4. vi login.html

What is vi?

vi is a text editor used inside the Linux terminal.

Why vi?

Allows editing files directly from the command line.

Purpose

- Create and edit text files

```
~
~
~
~
~
~
~
"index.html" 0L, 0B
```

What is mv?

mv is used to move or rename files and directories.

Why mv?

Helps in organizing files and changing file names.

Purpose

- Rename files
- Move files to another directory

mv is used to move or rename files and directories.
Why mv?
Helps in organizing files and changing file names.
Purpose

Why mv?

Helps in organizing files and changing file names.

Purpose

- Rename files
- Move files to another directory

```
mv index.html  
ls
```

What is mkdir?

mkdir creates a new directory (folder).

Why mkdir?

Used to organize files into folders.

Purpose

- Create directories

mkdir creates a new directory (folder).
Why mkdir?
 Used to organize files into folders.
 Purpose

Used to organize files into folders.

Purpose

- Create directories

Output:

```
controlplane:~$ mkdir wipro-devops-training
controlplane:~$ ls
filesystem  index.html  wipro-devops-training
```

7. cd DevOps/

What is cd?

cd changes the current directory.

Why cd?

Used to move between folders in the file system.

Purpose

- Navigate directories

Output:

```
controlplane:~$ cd DevOps
controlplane:~/DevOps$
```

8. pwd

What is pwd?

pwd shows the present working directory.

Why pwd?

Helps users know their exact location in the file system.

Purpose

- Display current directory path

Output:

```
controlplane:~$ pwd
/home/controlplane
```

9. rmdir DevOps/

What is rmdir?

rmdir removes an empty directory.

Why rmdir?

Used to delete folders that are no longer needed.

Purpose

- Remove empty directories

Output:

```
controlplane:~$ rmdir DevOps/  
controlplane:~$ ls  
filesystem wipro-devops-training
```

10. whoami Command

What is whoami?

whoami is a Linux command used to display the name of the currently logged-in user.

Output:

```
filesystem index.html wipro-devops-training  
controlplane:~$ whoami  
root
```

11. history

What is history?

history shows previously executed commands.

Why history?

Helps users track and reuse past commands.

Output:

```
wget https://d1cnf.apache.org
la
ping www.google.com
ping www.amazon.com
history
plane:~$
```

12.PING Command

What is ping?

ping is a Linux command used to check network connectivity between your system and another system (server/website).

Why ping?

It helps to verify whether a server or website is reachable and how fast the network response is. Output:

```
controlplane:~$ ping google.com
PING google.com (172.217.20.46) 56(84) bytes of data:
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=1 ttl=116 time=16.8 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=2 ttl=116 time=12.7 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=3 ttl=116 time=13.3 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=4 ttl=116 time=13.9 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=5 ttl=116 time=15.5 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=6 ttl=116 time=12.7 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=7 ttl=116 time=13.9 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=8 ttl=116 time=17.0 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=9 ttl=116 time=13.0 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=10 ttl=116 time=12.7 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=11 ttl=116 time=13.1 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=12 ttl=116 time=12.5 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=13 ttl=116 time=13.8 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=14 ttl=116 time=15.0 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=15 ttl=116 time=13.0 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=16 ttl=116 time=13.8 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=17 ttl=116 time=12.7 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=18 ttl=116 time=13.2 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=19 ttl=116 time=13.0 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=20 ttl=116 time=15.0 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=21 ttl=116 time=12.9 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=22 ttl=116 time=13.6 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=23 ttl=116 time=13.7 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=24 ttl=116 time=13.1 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=25 ttl=116 time=13.2 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=26 ttl=116 time=12.9 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=27 ttl=116 time=12.8 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=28 ttl=116 time=13.2 ms
64 bytes from par10s09-in-f46.1e100.net (172.217.20.46): icmp_seq=29 ttl=116 time=12.9 ms
```

TOP Command

What is top?

top is a Linux command used to display real-time system processes and resource usage.

Why top?

It helps to monitor CPU usage, memory usage, and running processes in real time.

Output:

```
top - 06:23:19 up 10 min, 1 user, load average: 0.35, 0.63, 0.51
Tasks: 201 total, 1 running, 200 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3.8 us, 2.4 sy, 0.0 ni, 91.4 id, 1.7 wa, 0.0 hi, 0.3 si, 0.3 st
MiB Mem : 2246.3 total, 305.0 free, 1041.5 used, 1071.8 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 1204.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1894	root	20	0	1512940	275200	42496	S	2.3	12.0	0:24.53	kube-apiserver
1662	root	20	0	1896792	66560	30208	S	1.7	2.9	0:12.04	kubelet
1968	root	20	0	11.2g	64092	18176	S	1.7	2.8	0:11.14	etcd
2000	root	20	0	1298312	83876	38400	S	1.3	3.6	0:09.08	kube-controller
3430	root	20	0	1401680	58264	43904	S	1.0	2.5	0:05.69	calico-node
2167	root	0	-20	1231188	12180	4480	S	0.7	0.5	0:01.84	runtime-info-se
887	root	20	0	1938732	45568	16640	S	0.3	2.0	0:04.28	containerd
1927	root	20	0	1275700	37120	18176	S	0.3	1.6	0:05.87	kube-scheduler
8521	root	20	0	10232	6016	3840	R	0.3	0.3	0:00.04	top
1	root	20	0	22612	13540	9444	S	0.0	0.6	0:04.51	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-slub_
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-netns
8	root	20	0	0	0	0	I	0.0	0.0	0:00.07	kworker/0:0-events
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
11	root	20	0	0	0	0	I	0.0	0.0	0:00.21	kworker/u2:0-events_unbound
12	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-mm_pe
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
16	root	20	0	0	0	0	S	0.0	0.0	0:00.34	ksoftirqd/0
17	root	20	0	0	0	0	I	0.0	0.0	0:00.49	rcu_preempt
18	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
19	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0

df Command

What is df?

df stands for Disk Filesystem.

It is a Linux command used to check disk space usage on the system.

Why df?

It helps users understand how much storage space is used and how much is free on each disk partition. This is very important in servers and Kubernetes environments to avoid running out of disk space.

```
controlplane:~$ df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
tmpfs	230020	2560	227460	2%	/run
/dev/vda1	19221248	10319912	8884952	54%	/
tmpfs	1150084	84	1150000	1%	/dev/shm
tmpfs	5120	0	5120	0%	/run/lock
/dev/vda16	901520	119048	719344	15%	/boot
/dev/vda15	106832	6250	100582	6%	/boot/efi
tmpfs	230016	96	229920	1%	/run/user/112
shm	65536	0	65536	0%	/run/containerd/io.containerd.grpc.v1.cri/sandboxes/0baaa130602bc11a5cea79e419a498a81e326abe0c8d72f70d5cb1ccac6cdd12/shm
shm	65536	0	65536	0%	/run/containerd/io.containerd.grpc.v1.cri/sandboxes/cf858efe277a34f3ac90626356b575ca71bbc749ccf03b4faedb3cbcece32360/shm
shm	65536	0	65536	0%	/run/containerd/io.containerd.grpc.v1.cri/sandboxes/6e9274dc047271884901b5154d4f76a17e39e8aab685615916d0a95117dab281/shm
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/cf858efe277a34f3ac90626356b575ca71bbc749ccf03b4faedb3cbcece32360/rootfs
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/0baaa130602bc11a5cea79e419a498a81e326abe0c8d72f70d5cb1ccac6cdd12/rootfs
shm	65536	0	65536	0%	/run/containerd/io.containerd.grpc.v1.cri/sandboxes/99c151d1330afffb0f06a34dc7d35eae0aa6f913a8063515579cf277795b1683/shm
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/99c151d1330afffb0f06a34dc7d35eae0aa6f913a8063515579cf277795b1683/rootfs
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/6e9274dc047271884901b5154d4f76a17e39e8aab685615916d0a95117dab281/rootfs
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/f4db0317012e3eb5239ddf8b37980096e70037f3ea72a9245c81ad03793435de/rootfs
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/cbd5a60a621f4b9097923806647106ad7873ff487af2b9ad93afa2837d0483d3/rootfs
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/bba1544e565c099aab4f7402d575058d714546dfef2ab43d6c9c686519a22f6c/rootfs
overlay	19221248	10319912	8884952	54%	/run/containerd/io.containerd.runtime.v2.task/k8s.io/2e44e5cd5988a7a009de4e786c9addda9e382863fc25057d8ef1744159bddfdd/rootfs
tmpfs	2097416	12	2097404	1%	/var/lib/kubelet/pods/9854fc42-dbc9-449e-a6ea-8be77d420ada/volumes/kubernetes.io~projected/kube-