



10. Important_Linux Command



TechData-Infinity-Devops with MultiCloud



find –

The find command is used to find a particular file within a directory. It also supports various options to find a file such as byname, by type, by date, and more.

The following symbols are used after the find command:

(.) : For current directory name

(/) : For root

find . -name "*.txt"

```
root@ip-172-31-8-13:~# ls
file1.tar.bz2  file1.tar.gz  file1.txt  file2.txt  snap
root@ip-172-31-8-13:~# find . -name "*.txt"
./file1.txt
./file2.txt
root@ip-172-31-8-13:~#
```

locate –

The locate command is used to search a file by file name. It is quite similar to find command; the difference is that it is a background process. It searches the file in the database, whereas the find command searches in the file system. It is faster than the find command. To find the file with the locates command, keep your database updated.

locate <file name>

```
root@ip-172-31-8-13:~# locate file1
/root/file1.tar.bz2
/root/file1.tar.gz
/root/file1.txt
root@ip-172-31-8-13:~#
```

date –

date

```
root@ip-172-31-8-13:~# date
Fri Jul 8 14:02:47 UTC 2022
root@ip-172-31-8-13:~#
```

TechData-Infinity-Devops with MultiCloud



df -

The df command is used to display the disk space used in the file system. It displays the output as in the number of used blocks, available blocks, and the mounted directory.

df -h

```
root@ip-172-31-42-137:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        7.6G  1.6G  6.1G  20% /
tmpfs            484M   0    484M   0% /dev/shm
tmpfs            194M  828K  193M   1% /run
tmpfs            5.0M   0    5.0M   0% /run/lock
/dev/xvda15      105M  5.3M  100M   5% /boot/efi
tmpfs            97M   4.0K   97M   1% /run/user/1000
root@ip-172-31-42-137:~#
```

du -

If you want to check how much space a file or a directory takes, the **du (Disk Usage)** command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the **-h** argument to the command line.

```
root@ip-172-31-42-137:/var# du -h
4.0K    ./mail
4.0K    ./snap/core18/common
4.0K    ./snap/core18/2409
12K     ./snap/core18
4.0K    ./snap/lxd/common/lxd
4.0K    ./snap/lxd/common/lxd-user
16K     ./snap/lxd/common
4.0K    ./snap/lxd/22923
24K     ./snap/lxd
4.0K    ./snap/amazon-ssm-agent/common
4.0K    ./snap/amazon-ssm-agent/5656
12K     ./snap/amazon-ssm-agent
4.0K    ./snap/core20/common
4.0K    ./snap/core20/1518
12K     ./snap/core20
4.0K    ./snap/snapd/common
4.0K    ./snap/snapd/16010
12K     ./snap/snapd
76K     ./snap
```

TechData-Infinity-Devops with MultiCloud



mount –

The mount command is used to connect an external device file system to the system's file system.

mount -t type <device> <directory>

```
root@ip-172-31-8-13:~# mount
/dev/xvda1 on / type ext4 (rw,relatime,discard,errors=remount-ro)
devtmpfs on /dev type devtmpfs (rw,relatime,size=488900k,nr_inodes=122225,mode=755,inode64)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,size=198228k,nr_inodes=819200,mode=755,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursive...)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
```

mail –

mail -s "Subject" <recipient address>

```
root@ip-172-31-8-13:~# mail -s "welcome to Young Minds" ygminds73@gmail.com
Cc:
Welcome to Young Minds.

You will learn AWS/DevOps

root@ip-172-31-8-13:~#
```

Man –

man command in Linux is used to display the user manual of any command that we can run on the terminal.

```
root@ip-172-31-42-137:~# man ls
root@ip-172-31-42-137:~#
```

ps aux–

The **ps** (process status) command is one of the most frequently used commands in Linux. Usually it is used to get the more and detailed information about a specific process or all processes.

“aux” print all the running process in system regardless from where they have been executed.

TechData-Infinity-Devops with MultiCloud



```
root@ip-172-31-42-137:~# ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1   0.0  1.2 167412 12816 ?        Ss   05:36   0:06 /sbin/init
root           2   0.0  0.0      0      0 ?        S    05:36   0:00 [kthreadd]
root           3   0.0  0.0      0      0 ?        I<   05:36   0:00 [rcu_gp]
root           4   0.0  0.0      0      0 ?        I<   05:36   0:00 [rcu_par_gp]
root           5   0.0  0.0      0      0 ?        I<   05:36   0:00 [netns]
root           7   0.0  0.0      0      0 ?        I<   05:36   0:00 [kworker/0:0H]
root           8   0.0  0.0      0      0 ?        I    05:36   0:00 [kworker/0:1-]
root          10   0.0  0.0      0      0 ?        I<   05:36   0:00 [mm_percpu_wq]
root          11   0.0  0.0      0      0 ?        S    05:36   0:00 [rcu_tasks_ru]
root          12   0.0  0.0      0      0 ?        S    05:36   0:00 [rcu_tasks_tr]
```

kill-

If you have an unresponsive program, you can terminate it manually by using the kill command. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

There is a total of sixty-four signals that you can use, but people usually only use two signals:

SIGTERM (15) — requests a program to stop running and gives it some time to save all of its progress. If you don't specify the signal when entering the kill command, this signal will be used.

SIGKILL (9) — forces programs to stop immediately. Unsaved progress will be lost.

Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to kill. If you don't know the PID, simply run the command `ps ux`.

After knowing what signal you want to use and the PID of the program, enter the following syntax:

kill [signal option] PID

```
root@ip-172-31-42-137:~# ps -aux | grep nginx
root          3032  0.0  0.1  55212  1676 ?        Ss   08:44   0:00 nginx: master process /usr/sbin/nginx -g dae
mon on; master_process on;
www-data      3033  0.0  0.5  55844  5648 ?        S    08:44   0:00 nginx: worker process
root          3035  0.0  0.2   7004  2196 pts/3    S+   08:44   0:00 grep --color=auto nginx
root@ip-172-31-42-137:~# kill -9 3033
root@ip-172-31-42-137:~# ps -aux | grep nginx
root          3032  0.0  0.1  55212  1676 ?        Ss   08:44   0:00 nginx: master process /usr/sbin/nginx -g dae
mon on; master_process on;
www-data      3037  0.0  0.5  55844  5648 ?        S    08:44   0:00 nginx: worker process
root          3040  0.0  0.2   7004  2176 pts/3    S+   08:44   0:00 grep --color=auto nginx
root@ip-172-31-42-137:~#
```

Wget-

The Linux command line is super useful — you can even download files from the internet with the help of the **wget** command. To do so, simply type **wget** followed by the download link.

```
root@ip-172-31-42-137:~# wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.17.2.tar.xz
--2022-07-10 08:54:21-- https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.17.2.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.153.176, 2a04:4e42:24::432
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.153.176|:443... connected.
HTTP request sent, awaiting response...
```


TechData-Infinity-Devops with MultiCloud



Uname –

The **uname** command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

```
root@ip-172-31-42-137:~# uname
Linux
root@ip-172-31-42-137:~# uname -r
5.15.0-1011-aws
root@ip-172-31-42-137:~# |
```

History –

When you've been using Linux for a certain period of time, you'll quickly notice that you can run hundreds of commands every day. As such, running **history** command is particularly useful if you want to review the commands you've entered before.

```
root@ip-172-31-42-137:~# history
 1  clear
 2  apt install nginx
 3  clear
 4  systemctl start nginx
 5  systemctl status nginx
 6  clear
 7  id
 8  clear
 9  ps aux
10  clear
11  systemctl status nginx
12  ps -aux | nginx
```

Top –

As a terminal equivalent to Task Manager in Windows, the **top** command will display a list of running processes and how much CPU each process uses. It's very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

TechData-Infinity-Devops with MultiCloud



```

root@ip-172-31-42-137:~# top
top - 09:09:18 up 3:33, 4 users, load average: 0.00, 0.00, 0.00
Tasks: 107 total, 1 running, 106 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 967.9 total, 204.8 free, 198.4 used, 564.7 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 579.7 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 3077 root        20   0   10904    3936   3228 R   0.3   0.4   0:00.01 top
    1 root        20   0  167412   12824   8192 S   0.0   1.3   0:06.54 systemd
    2 root        20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
    7 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
    8 root        20   0      0      0      0 I   0.0   0.0   0:00.04 kworker/0:1-cgroup_destroy
   10 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
   11 root        20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
   12 root        20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
   13 root        20   0      0      0      0 S   0.0   0.0   0:00.12 ksoftirqd/0
   14 root        20   0      0      0      0 I   0.0   0.0   0:00.53 rcu_sched
   15 root         0 -20      0      0      0 S   0.0   0.0   0:00.00 migration/0
  
```

- **PID:** Shows task's unique process id.
- **PR:** The process's priority. The lower the number, the higher the priority.
- **VIRT:** Total virtual memory used by the task.
- **USER:** User name of owner of task.
- **%CPU:** Represents the CPU usage.
- **TIME+:** CPU Time, the same as 'TIME', but reflecting more granularity through hundredths of a second.
- **SHR:** Represents the Shared Memory size (kb) used by a task.
- **NI:** Represents a Nice Value of task. A Negative nice value implies higher priority, and positive Nice value means lower priority.
- **%MEM:** Shows the Memory usage of task.
- **RES:** How much physical RAM the process is using, measured in kilobytes.
- **COMMAND:** The name of the command that started the process

