18/02/2025

Remove an empty directory:

```
root@Goosari:~# mkdir a
root@Goosari:~# rmdir -p a
root@Goosari:~# ls -lrt
total 40
drwxr-xr-x 2 root root 4096 Feb 18 09:37 practcie
-rwxr--r-- 1 root root   21 Feb 18 11:17 code.sh
-rwxr--r-- 1 root root   21 Feb 18 11:20 greet.sh
-rw-r--r-- 1 root root   35 Feb 18 11:29 name.txt
-rwx------ 1 root root  156 Feb 18 11:47 first.sh
-rwx------ 1 root root  246 Feb 18 12:27 second.sh
-rwx------ 1 root root  197 Feb 18 12:37 third.sh
-rwx------ 1 root root  159 Feb 18 13:15 forth.sh
-rw-r--r-- 1 root root   78 Feb 18 14:44 data.txt
drwxr-xr-x 2 root root 4096 Feb 18 20:20 practice
```

Remove directory by recursively forcing:

```
root@Goosari:~# rm -rf a
```

Uname command: Prints system information

```
root@Goosari:~# uname -a
Linux Goosari 5.15.167.4-microsoft-standard-WSL2 #1 SMP Tue Nov 5 00:21:55 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
root@Goosari:~# uname -s
Linux
root@Goosari:~# uname -n
Goosari
root@Goosari:~# uname -r
5.15.167.4-microsoft-standard-WSL2
root@Goosari:~# uname -m
x86_64
```

- uname -a : print all details except omit -p and -i if unknown
    - printed in the format
        - *kernel name*
        - *nodename*
        - *kernel release*
        - *kernel version*
        - *machine*
        - *processor*
        - *hardware platform*
        - *operating system*
- uname -s : print kernel name
- uname -n : print network node host name
- uname -r : print kernel release
- uname -m : print machine hardware name

Read the content of the file:

```
root@Goosari:~# vi content.txt
root@Goosari:~# cat content.txt
Khushi
Goosari
Windows
Linux
z
y
x
```

cat command: Concatenate files and print on the standard output

Display information about a selection of the active processes. If you want a repetitive update of the selection and the displayed information, use top instead.

```
For more details see ps(1).
root@Goosari:~# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.1  21668 12780 ?        Ss   03:37   0:00 /sbin/init
root           2  0.0  0.0   2776  1920 ?        Sl   03:37   0:00 /init
root           6  0.0  0.0   2776    68 ?        Sl   03:37   0:00 plan9 --control-socket 7 --log-level 4 --server-fd 8
root          51  0.0  0.1  34060 12584 ?        S<s  03:38   0:00 /usr/lib/systemd/systemd-journald
root          93  0.0  0.0  23968  5880 ?        Ss   03:38   0:00 /usr/lib/systemd/systemd-udevd
systemd+     136  0.0  0.1  21452 11880 ?        Ss   03:38   0:00 /usr/lib/systemd/systemd-resolved
systemd+     138  0.0  0.0  91020  6520 ?        Ssl  03:38   0:00 /usr/lib/systemd/systemd-timesyncd
root         144  0.0  0.0   4236  2604 ?        Ss   03:38   0:00 /usr/sbin/cron -f -P
message+     145  0.0  0.0   9528  5152 ?        Ss   03:38   0:00 @dbus-daemon --system --address=systemd: --nofork --n
root         156  0.0  0.1  17976  8372 ?        Ss   03:38   0:00 /usr/lib/systemd/systemd-logind
root         159  0.0  0.2 1755840 17952 ?       Ssl  03:38   0:00 /usr/libexec/wsl-pro-service -vv
root         164  0.0  0.0   3160  1188 hvc0     Ss+  03:38   0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 1152
root         187  0.0  0.0   3116  1156 tty1     Ss+  03:38   0:00 /sbin/agetty -o -p -- \u --noclear - linux
syslog       188  0.0  0.1 222508 11516 ?        Ssl  03:38   0:00 /usr/sbin/rsyslogd -n -iNONE
root         197  0.0  0.2 107008 22228 ?        Ssl  03:38   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unatt
mysql        241  0.6  5.3 2375900 417664 ?      Ssl  03:38   0:07 /usr/sbin/mysqld
root         335  0.0  0.0   2784   204 ?        Ss   03:38   0:00 /init
root         336  0.0  0.0   2784   208 ?        S    03:38   0:00 /init
root         337  0.0  0.0   6072  5328 pts/0    Ss   03:38   0:00 -bash
root         338  0.0  0.0   6664  4492 pts/1    Ss   03:38   0:00 /bin/login -f
root         434  0.0  0.1  20264 11468 ?        Ss   03:38   0:00 /usr/lib/systemd/systemd --user
root         435  0.0  0.0  21148  1708 ?        S    03:38   0:00 (sd-pam)
root         446  0.0  0.0   6072  5276 pts/1    S+   03:38   0:00 -bash
root         568  0.4  1.0  88380 79052 pts/0    S+   03:53   0:00 apt install plocate
root         631  0.0  0.0   8452  4804 pts/2    Ss+  03:53   0:00 /usr/bin/dpkg --status-fd 39 --configure --pending
root         632  0.0  0.0   2800  1108 pts/2    S+   03:53   0:00 /bin/sh /var/lib/dpkg/info/plocate.postinst configure
root         646  5.9  2.2 188228 176732 pts/2   Dl+  03:53   0:13 /usr/sbin/updatedb.plocate
root         652  0.0  0.0   2784   208 ?        Ss   03:55   0:00 /init
root         653  0.0  0.0   2784   212 ?        S    03:55   0:00 /init
root         659  0.0  0.0   6072  5192 pts/3    Ss   03:55   0:00 -bash
root         679  0.0  0.0   8332  4312 pts/3    R+   03:57   0:00 ps aux
```

ps - report a snapshot of the current processes.

Display a simple calendar in traditional format:

```
root@Goosari:~# ncal
    February 2025
Su     2  9 16 23
Mo     3 10 17 24
Tu     4 11 18 25
We     5 12 19 26
Th     6 13 20 27
Fr     7 14 21 28
Sa  1  8 15 22
```

- ncal: print current month calendar
- ncal -y “2024”: print whole year calendar

whereis command: locates the binary, source and manual files for the specified command names.

```
root@Goosari:~# plocate "17-02-2025".txt
/mnt/c/Users/khush/AppData/Roaming/Microsoft/Windows/Recent/17-02-2025.txt.lnk
/mnt/c/Users/khush/AppData/Roaming/Notepad++/backup/17-02-2025.txt@2025-02-17_231627
/mnt/c/Users/khush/Desktop/Mthree training/17-02-2025.txt
/mnt/c/Users/khush/OneDrive/Desktop/Mthree/17-02-2025.txt
root@Goosari:~# whereis python
python:
root@Goosari:~#
root@Goosari:~# whereis Python
Python:
root@Goosari:~# whereis java
java: /usr/share/java
root@Goosari:~# whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
```

wc command: Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified.

```
root@Goosari:~/practice# vi 2.txt
root@Goosari:~/practice# cat 2.txt
Khushi Goosari
How are you?
root@Goosari:~/practice# wc -l 2.txt
2 2.txt
root@Goosari:~/practice# wc -w 2.txt
5 2.txt
root@Goosari:~/practice# wc -c 2.txt
28 2.txt
root@Goosari:~/practice# man wc
root@Goosari:~/practice# wc -m 2.txt
28 2.txt
root@Goosari:~/practice# wc *
 0  0  0 1.txt
 2  5 28 2.txt
 0  0  0 3.txt
 0  0  0 4.txt
 0  0  0 5.txt
 2  5 28 total
root@Goosari:~/practice#
```

- wc -l: Count number of lines
- wc -w: Count number of words
- wc -c: Count number of bytes
- wc -m: Count number of characters

Assign string to variable name and print the text “Hello, <name>”

```
root@Goosari:~# name="Khushi"
root@Goosari:~# echo "Hello, $name"
Hello, Khushi
root@Goosari:~# whoami
root
root@Goosari:~#
```

Whoami command: Prints effective user name

Find command: Search for files in a directory hierarchy

```
root@Goosari:~# find -name "*txt"
./practice/4.txt
./practice/3.txt
./practice/1.txt
./practice/2.txt
./practice/5.txt
root@Goosari:~# find -type d
.
./.ssh
./practcie
./practice
./.cache
root@Goosari:~# find . -name "*.tmp" -exec rm {} \;
root@Goosari:~#
```

Summarize device usage of the set of FILEs, recursively for directories

```
root@Goosari:~# du -sh .
108K    .
root@Goosari:~# du -sh *
4.0K    code.sh
4.0K    content.txt
4.0K    data.txt
4.0K    first.sh
4.0K    forth.sh
4.0K    greet.sh
4.0K    name.txt
4.0K    practcie
12K     practice
4.0K    second.sh
4.0K    third.sh
root@Goosari:~#
```

- -s: summarize
- -h: print in human readable format

Print the first 2 lines and last 2 lines of file.

```
root@Goosari:~# cat content.txt
Khushi
Goosari
Windows
Linux
z
y
x
root@Goosari:~# head -2 content.txt
Khushi
Goosari
root@Goosari:~# tail -2 content.txt
y
x
root@Goosari:~# head -3 content.txt | tail -1
Windows
root@Goosari:~#
```

head -3 content.txt | tail -1: Take the 1st three lines and print the last line from that output

SHELL SCRIPT

Create a shell script to initialize 2 variables and print it.

```
var_1="Khushi"
var_2="Goosari"
echo "$var_1 $var_2"
unset var_1
echo "$var_1"
readonly var_2
var_2='khushigoosari'
~
~
```

- Unset removes a variable or function
- Readonly makes a variable or function immutable

```
root@Goosari:~# vi first.sh
root@Goosari:~# ./first.sh
Khushi Goosari

./first.sh: line 8: unset: var_2: cannot unset: readonly variable
root@Goosari:~# ./first.sh
Khushi Goosari
```

Example 2:

```
var_name='Khushi'
var_age='22'
echo "Name is $var_name and age is $var_age"

var_blood_grp='A+'
echo "Error modifying readonly varaiable please dont modify it "
echo var_blood_group="b+"
echo

unset var_age
echo "Age is after unsetting $var_age"
~
~
```

```
root@Goosari:~# vi second.sh
root@Goosari:~# ./second.sh
Name is Khushi and age is 22
Error modifying readonly varaiable please dont modify it
var_blood_group=b+

Age is after unsetting
root@Goosari:~#
```

Conditional Statement:

Write a Bash script that prints a greeting message (Good morning, Good afternoon, or Good evening) based on the current hour using the date command.

```
time=$(date +%H)
echo $time
if [ $time -lt 12 ];then
        message='Good morning user'
elif [ $time -lt 18 ];then
        message='Good afternoon user'
else
        message='Good evening user'
fi
echo $message $time
~
```

```
root@Goosari:~# vi third.sh
root@Goosari:~# ./third.sh
21
Good evening user 21
root@Goosari:~#
```

```
root@Goosari:~# sudo timedatectl set-timezone Asia/Kolkata
root@Goosari:~# timedatectl
               Local time: Tue 2025-02-18 12:39:11 IST
           Universal time: Tue 2025-02-18 07:09:11 UTC
                 RTC time: Tue 2025-02-18 07:09:12
                Time zone: Asia/Kolkata (IST, +0530)
System clock synchronized: no
              NTP service: active
          RTC in local TZ: no
root@Goosari:~# ./third.sh
12
Good afternoon user 12
root@Goosari:~# vi data.txt
```

Looping:

Write a Bash script that uses a while loop to print string 4 times and a for loop to iterate through numbers 1 to 9, stopping at 5 using a break statement.

```
i=1
while [ $i -lt 5 ];
do
   echo "Khushi"
   i=`expr $i + 1`
done
for a in 1 2 3 4 5 6 7 8 9
do
        if [ $a == 5 ]
        then break
        fi
        echo "iteration is $a"
done
~
~
```

```
root@Goosari:~# vi forth.sh
root@Goosari:~# ./forth.sh
Khushi
Khushi
Khushi
Khushi
iteration is 1
iteration is 2
iteration is 3
iteration is 4
root@Goosari:~# vi forth.sh
```

AWK:

```
root@Goosari:~# vi data.txt
root@Goosari:~# cat data.txt
Khushi 22 engineer
kgk 25 civil
goosari 27 electric
root@Goosari:~# awk '{print $1}' data.txt
Khushi
kgk
goosari
root@Goosari:~# awk '{print "name " $1, "profession " $3}' data.txt
name Khushi profession engineer
name kgk profession civil
name goosari profession electric
root@Goosari:~# awk '/civil/' data.txt
kgk 25 civil
root@Goosari:~# awk '/civil/ {print $1}' data.txt
kgk
root@Goosari:~# awk '$2 > 25 {print 1$, "is older than 25"}' data.txt
awk: cmd. line:1: $2 > 25 {print 1$, "is older than 25"}
awk: cmd. line:1:                 ^ syntax error
root@Goosari:~# awk '$2 > 25 {print $1, "is older than 25"}' data.txt
goosari is older than 25
root@Goosari:~# awk '{if ($2 > 25) print $1, "is older than 25"; else print $1, "is younger than 25"}' data.txt
Khushi is younger than 25
kgk is younger than 25
goosari is older than 25
root@Goosari:~#
```

Pattern scanning and processing language

Tar:

Used to create, extract, and manipulate archive files

```
root@Goosari:/mnt/c/Users/khush/Desktop# tar -cvf archive.tar Test/
Test/
Test/1.txt
Test/2.txt
Test/3.txt
Test/4.txt
Test/5.txt
root@Goosari:/mnt/c/Users/khush/Desktop#
```

- -c : create a new archive
- -v : Verbose mode (lists files being added)
- -f : File name specification (i.e. archive.tar)

```
root@Goosari:/mnt/c/Users/khush/Desktop# tar -czvf archive.tar.gz Test/
Test/
Test/1.txt
Test/2.txt
Test/3.txt
Test/4.txt
Test/5.txt
```