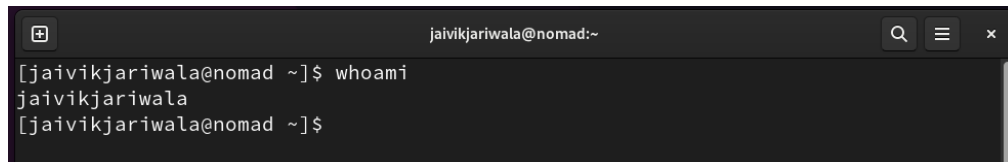


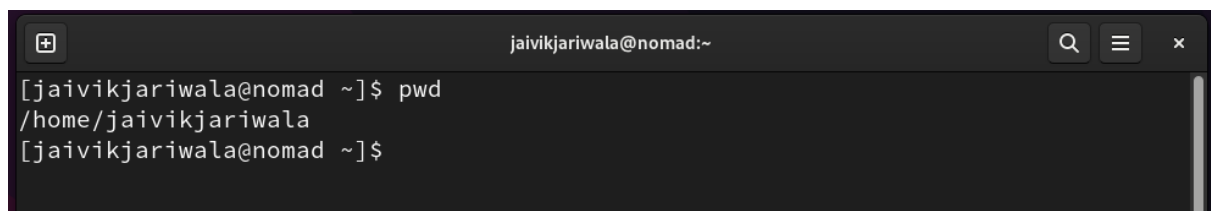
Perform the following Linux Fedodra command operations

1. whoami



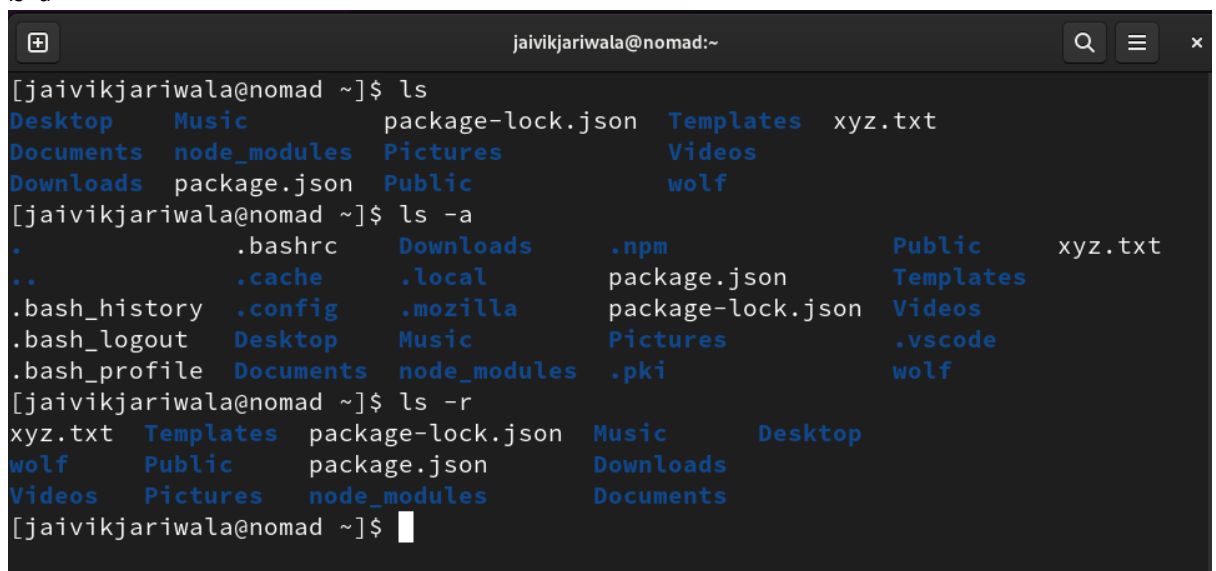
```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ whoami  
jaivikjariwala  
[jaivikjariwala@nomad ~]$
```

2. pwd



```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ pwd  
/home/jaivikjariwala  
[jaivikjariwala@nomad ~]$
```

3. is
4. is -r
5. is -a



```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ ls  
Desktop    Music      package-lock.json  Templates  xyz.txt  
Documents  node_modules Pictures      Videos  
Downloads  package.json Public        wolf  
[jaivikjariwala@nomad ~]$ ls -a  
.  
..  
.bash_history .cache .local .npm Public xyz.txt  
.bash_logout .config .mozilla package.json Templates  
.bash_profile Desktop Music Pictures Videos  
Documents node_modules .pki wolf  
[jaivikjariwala@nomad ~]$ ls -r  
xyz.txt Templates package-lock.json Music Desktop  
wolf Public package.json Downloads  
Videos Pictures node_modules Documents  
[jaivikjariwala@nomad ~]$
```

6. history

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ history  
1 sudo rpm  
2 sudo rpm --import http://packages.microsoft.com/keys/microsoft.asc  
3 sudo rpm  
4 npm install  
5 install npz  
6 install npx  
7 npm install npz
```

7. clear

```
72 history  
73 clear  
74 history  
[jaivikjariwala@nomad ~]$ clear  
  
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$
```

8. echo

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ echo  
  
[jaivikjariwala@nomad ~]$
```

9. touch

```
jaivikjariwala@nomad:~  
wc: kp.txt: No such file or directory  
[jaivikjariwala@nomad ~]$ ls  
Desktop Downloads node_modules package-lock.json Public Videos  
Documents Music package.json Pictures Templates wolf  
[jaivikjariwala@nomad ~]$ cd text  
bash: cd: text: No such file or directory  
[jaivikjariwala@nomad ~]$ touch  
touch: missing file operand  
Try 'touch --help' for more information.  
[jaivikjariwala@nomad ~]$ touch abc  
[jaivikjariwala@nomad ~]$ ls  
abc Documents Music package.json Pictures Templates wolf  
Desktop Downloads node_modules package-lock.json Public Videos  
[jaivikjariwala@nomad ~]$ rm abc  
[jaivikjariwala@nomad ~]$ ls  
Desktop Downloads node_modules package-lock.json Public Videos  
Documents Music package.json Pictures Templates wolf  
[jaivikjariwala@nomad ~]$ mv abc.txt  
mv: missing destination file operand after 'abc.txt'  
Try 'mv --help' for more information.  
[jaivikjariwala@nomad ~]$ ls  
Desktop Downloads node_modules package-lock.json Public Videos  
Documents Music package.json Pictures Templates wolf  
[jaivikjariwala@nomad ~]$
```

10. rm
11. mkdir

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ mkdir wolf  
[jaivikjariwala@nomad ~]$ ls  
Desktop  Downloads  node_modules  package-lock.json  Public  Videos  
Documents Music      package.json  Pictures           Templates  wolf  
[jaivikjariwala@nomad ~]$
```

12. rmdir
13. mv

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ mkdir wolf  
[jaivikjariwala@nomad ~]$ ls  
Desktop  Downloads  node_modules  package-lock.json  Public  Videos  
Documents Music      package.json  Pictures           Templates  wolf  
[jaivikjariwala@nomad ~]$ rmdir lonewolf  
rmdir: failed to remove 'lonewolf': No such file or directory  
[jaivikjariwala@nomad ~]$ ls  
Desktop  Downloads  node_modules  package-lock.json  Public  Videos  
Documents Music      package.json  Pictures           Templates  wolf  
[jaivikjariwala@nomad ~]$ tree plan  
plan [error opening dir]  
  
0 directories, 0 files  
[jaivikjariwala@nomad ~]$
```

14. cd

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ cd  
[jaivikjariwala@nomad ~]$
```

15. cmp

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ cd  
[jaivikjariwala@nomad ~]$ ls  
Desktop  Downloads  node_modules  package-lock.json  Public  Videos  
Documents Music      package.json  Pictures           Templates  
[jaivikjariwala@nomad ~]$
```

16. cat

```
jaivikjariwala@nomad:~ — cat  
[jaivikjariwala@nomad ~]$ cat  
file 1  
file 1  
clear  
clear  
  stopped  
  stopped  
[1]+ Stopped  
[1]+ Stopped
```

17. cal

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ echo  
[jaivikjariwala@nomad ~]$ cal  
January 2023  
Su Mo Tu We Th Fr Sa  
1 2 3 4 5 6 7  
8 9 10 11 12 13 14  
15 16 17 18 19 20 21  
22 23 24 25 26 27 28  
29 30 31
```

18. cal -ycal 2018

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ cal 2023  
2023  
  
January February March  
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa  
1 2 3 4 5 6 7 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  
8 9 10 11 12 13 14 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28  
15 16 17 18 19 20 21 12 13 14 15 16 17 18 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
22 23 24 25 26 27 28 19 20 21 22 23 24 25 26 27 28 29 30 31  
29 30 31 26 27 28 26 27 28 29 30 31  
  
April May June  
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa  
1 1 2 3 4 5 6 1 2 3  
2 3 4 5 6 7 8 7 8 9 10 11 12 13 4 5 6 7 8 9 10  
9 10 11 12 13 14 15 14 15 16 17 18 19 20 11 12 13 14 15 16 17  
16 17 18 19 20 21 22 21 22 23 24 25 26 27 18 19 20 21 22 23 24  
23 24 25 26 27 28 29 28 29 30 31 25 26 27 28 29 30  
30  
  
July August September  
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa  
1 1 2 3 4 5 1 2  
2 3 4 5 6 7 8 6 7 8 9 10 11 12 3 4 5 6 7 8 9  
9 10 11 12 13 14 15 13 14 15 16 17 18 19 10 11 12 13 14 15 16  
16 17 18 19 20 21 22 20 21 22 23 24 25 26 17 18 19 20 21 22 23  
23 24 25 26 27 28 29 27 28 29 30 31 24 25 26 27 28 29 30  
30 31  
  
October November December  
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa  
1 2 3 4 5 6 7 1 2 3 4 1 2  
8 9 10 11 12 13 14 5 6 7 8 9 10 11 3 4 5 6 7 8 9  
15 16 17 18 19 20 21 12 13 14 15 16 17 18 10 11 12 13 14 15 16  
22 23 24 25 26 27 28 19 20 21 22 23 24 25 17 18 19 20 21 22 23  
29 30 31 26 27 28 29 30 24 25 26 27 28 29 30  
31
```

19. passwd

```
jaivikjariwala@nomad:~ — passwd
[jaivikjariwala@nomad ~]$ passwd
Changing password for user jaivikjariwala.
Current password:
Current Password:
```

20. grep

```
jaivikjariwala@nomad:~ — cat
[jaivikjariwala@nomad ~]$ grep
Usage: grep [OPTION]... PATTERNS [FILE]...
Try 'grep --help' for more information.
[jaivikjariwala@nomad ~]$ cat > xyz.txt
^s
[2]+ Stopped
cat>xyz.txt
```

21. free

```
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ cd
[jaivikjariwala@nomad ~]$ ls
Desktop Downloads node_modules package-lock.json Public Videos
Documents Music package.json Pictures Templates
[jaivikjariwala@nomad ~]$ free
              total            used             free       shared  buff/cache   available
Mem:      15989212       1214732      13204368        380336       1570112      14098504
Swap:      8388604              0         8388604
```

22. uname

```
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ uname
Linux
[jaivikjariwala@nomad ~]$ uname -a
Linux nomad 6.0.7-301.fc37.x86_64 #1 SMP PREEMPT_DYNAMIC Fri Nov 4 18:35:48 UTC
2022 x86_64 x86_64 x86_64 GNU/Linux
[jaivikjariwala@nomad ~]$
```

23. uname -a

24. uname -s

```
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ uname -s
Linux
[jaivikjariwala@nomad ~]$
```

25. `uname -n`

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ uname -n  
nomad  
[jaivikjariwala@nomad ~]$
```

26. `group`

```
jaivikjariwala@nomad:/  
[jaivikjariwala@nomad home]$ group  
bash: group: command not found...  
[jaivikjariwala@nomad home]$ groups wolf  
groups: 'wolf': no such user  
[jaivikjariwala@nomad home]$ group liveuser  
bash: group: command not found...  
[jaivikjariwala@nomad home]$ groups liveuser  
groups: 'liveuser': no such user  
[jaivikjariwala@nomad home]$ chomd  
bash: chomd: command not found...  
Similar command is: 'chmod'  
[jaivikjariwala@nomad home]$ cd /  
[jaivikjariwala@nomad /]$ cd /home/Downloads/kp/  
bash: cd: /home/Downloads/kp/: No such file or directory  
[jaivikjariwala@nomad /]$ cd /home/jaivikjariwala/Downloads/jj/  
bash: cd: /home/jaivikjariwala/Downloads/jj/: No such file or directory  
[jaivikjariwala@nomad /]$
```

27. `comm`

```
jaivikjariwala@nomad:/home  
[jaivikjariwala@nomad ~]$ comm  
comm: missing operand  
Try 'comm --help' for more information.  
[jaivikjariwala@nomad ~]$ cd /  
[jaivikjariwala@nomad /]$ cd /home/  
[jaivikjariwala@nomad home]$ cd/home/user/downloads/kp/  
bash: cd/home/user/downloads/kp/: No such file or directory  
[jaivikjariwala@nomad home]$ comm abc.txt kp.txt  
comm: abc.txt: No such file or directory  
[jaivikjariwala@nomad home]$
```

28. `date`

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ date  
Thursday 19 January 2023 10:29:05 PM EST  
[jaivikjariwala@nomad ~]$
```

29. `date -d`

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ date -d yesterday  
Wednesday 18 January 2023 10:36:40 PM EST  
[jaivikjariwala@nomad ~]$
```

30. `chmod`

31. `wc`

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ wc kp.txt  
wc: kp.txt: No such file or directory  
[jaivikjariwala@nomad ~]$ ls  
Desktop Downloads node_modules package-lock.json Public Videos  
Documents Music package.json Pictures Templates wolf  
[jaivikjariwala@nomad ~]$ cd text  
bash: cd: text: No such file or directory  
[jaivikjariwala@nomad ~]$ touch  
touch: missing file operand  
Try 'touch --help' for more information.  
[jaivikjariwala@nomad ~]$ touch abc  
[jaivikjariwala@nomad ~]$ ls  
abc Documents Music package.json Pictures Templates wolf  
Desktop Downloads node_modules package-lock.json Public Videos  
[jaivikjariwala@nomad ~]$
```

Perform the following Linux Fedodra command operations

1. ifconfig : Display network interfaces and IP addresses.

```

jaivikjariwala@nomad:~$ ifconfig
eno2: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 58:11:22:3a:8b:79 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0x60100000-60120000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 26 bytes 2861 (2.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26 bytes 2861 (2.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 56:7e:e2:58:d0:61 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[jaivikjariwala@nomad ~]$

```

2. kill: Kill active processes by process ID or name

```

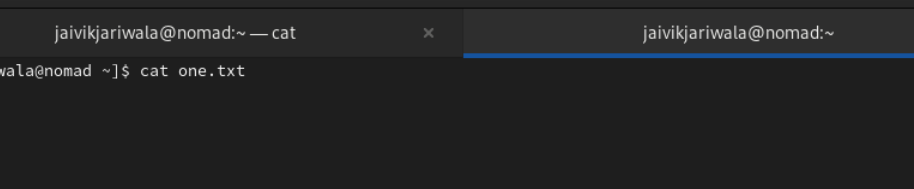
jaivikjariwala@nomad:~$ type -a kill
kill is a shell builtin
kill is /usr/bin/kill
[jaivikjariwala@nomad ~]$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
[jaivikjariwala@nomad ~]$ pidof firefox
9670 9639 9596 9499 9379 9319 9316 9264 9261 9228 9174 9156 9154 5206 5159 5121 4985
[jaivikjariwala@nomad ~]$ kill -9 $(pidof firefox)
[jaivikjariwala@nomad ~]$

```


3. mount : Mount file systems in Linux

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ mount  
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)  
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)  
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=1048576,  
mode=755,inode64)  
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)  
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)  
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,  
ptmxmode=000)  
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=3197844k,nr_inodes=819200,  
mode=755,inode64)  
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,  
nsdelegate,memory_recursiveprot)  
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)  
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)  
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)  
/dev/nvme0n1p8 on / type btrfs (rw,relatime,seclabel,compress=zstd:1,ssd,space_cache=v2,  
subvol=257,subvol=/root)  
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)  
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=35,pgrp=1,timeout=0,  
minproto=5,maxproto=5,direct,pipe_ino=22409)  
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)  
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)  
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)  
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)  
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)  
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)  
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,size=7994608k,nr_inodes=1048576,  
inode64)  
/dev/nvme0n1p8 on /home type btrfs (rw,relatime,seclabel,compress=zstd:1,ssd,space_cache=v2,  
subvol=256,subvol=/home)  
/dev/nvme0n1p7 on /boot type ext4 (rw,relatime,seclabel,stripe=32)  
/dev/nvme0n1p1 on /boot/efi type vfat (rw,relatime,fmask=0077,dmask=0077,codepage=437,  
iocharset=ascii,shortname=winnt,errors=remount-ro)  
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)  
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=1598920k,  
nr_inodes=399730,mode=700,uid=1000,gid=1000,inode64)  
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,  
group_id=1000)  
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,  
group_id=1000)
```

4. sort : Linux command to sort the content of a file while outputting



```
jaivikjariwala@nomad:~ — cat x jaivikjariwala@nomad:~ x
```

```
[jaivikjariwala@nomad ~]$ cat one.txt
hi
wolf
pup
howl
die
stress
depression
--Stopped
[jaivikjariwala@nomad ~]$ sort one.txt
depression
die
hi
howl
pup
--Stopped
stress
wolf
[jaivikjariwala@nomad ~]$
```

5. export : Export environment variables in Linux

```

[+]
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ export [-f][-n][name=value...]
bash: export: `[-f][-n][name=value]...': not a valid identifier
[jaivikjariwala@nomad ~]$ export -p
declare -x COLORTERM="truecolor"
declare -x DBUS_SESSION_BUS_ADDRESS="unix:path=/run/user/1000/bus"
declare -x DEBUGINFOD_URLS="https://debuginfod.fedoraproject.org/"
declare -x DESKTOP_SESSION="gnome"
declare -x DISPLAY=":0"
declare -x EDITOR="/usr/bin/nano"
declare -x GDMSESSION="gnome"
declare -x GDM_LANG="en_IN.UTF-8"
declare -x GNOME_SETUP_DISPLAY=":1"
declare -x GNOME_TERMINAL_SCREEN="/org/gnome/Terminal/screen/36005707_6568_4754_b0aa_8ad042751b82"
declare -x GNOME_TERMINAL_SERVICE=":1.93"
declare -x HISTCONTROL="ignoredups"
declare -x HISTSIZE="1000"
declare -x HOME="/home/jaivikjariwala"
declare -x HOSTNAME="nomad"
declare -x LANG="en_IN.UTF-8"
declare -x LESSOPEN="||/usr/bin/lesspipe.sh %s"
declare -x LOGNAME="jaivikjariwala"
declare -x LS_COLORS="rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;37;41:su=37;41:sg=3
0;43:ca=00:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lzh=01
;31:*.lzm=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.gz=01;31:*.gz=01;31:*.lrz=01;31:*.lzo=01
;31:*.xz=01;31:*.zst=01;31:*.tzt=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.w
ar=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31
:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.avif=01;35:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.
pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=
01;35:*.mov=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;
35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl
=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.
m4a=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.oga=01;36:*.opus=01;36:*.spx=01
;36:*.xspf=01;36:*.x=00;90:*.x=00;90:*.bak=00;90:*.old=00;90:*.orig=00;90:*.part=00;90:*.rej=00;90:*.swp=00;90:*.tmp=00;90:*.dpgk-dist=00;90:
*.dpkg-old=00;90:*.ucf-dist=00;90:*.ucf-new=00;90:*.ucf-old=00;90:*.rpmnew=00;90:*.rpmorig=00;90:*.rpmsave=00;90:"
declare -x MAIL="/var/spool/mail/jaivikjariwala"
declare -x OLDPWD
declare -x PATH="/home/jaivikjariwala/.local/bin:/home/jaivikjariwala/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin"
declare -x PWD="/home/jaivikjariwala"
declare -x QT_IM_MODULE="ibus"
declare -x SESSION_MANAGER="local/unix:@/tmp/.ICE-unix/2107,unix/unix:/tmp/.ICE-unix/2107"
declare -x SHELL="/bin/bash"
declare -x SHLVL="1"
declare -x SSH_AUTH_SOCK="/run/user/1000/keyring/ssh"
declare -x SYSTEMD_EXEC_PID="2162"
declare -x TERM="xterm-256color"
declare -x USER="jaivikjariwala"
declare -x USERNAME="jaivikjariwala"
declare -x VTE_VERSION="7001"
declare -x WAYLAND_DISPLAY="wayland-0"
declare -x XAUTHORITY="/run/user/1000/.mutter-Xwaylandauth.XBh3Y1"
declare -x XDG_CURRENT_DESKTOP="GNOME"
declare -x XDG_DATA_DIRS="/home/jaivikjariwala/.local/share/flatpak/exports/share:/var/lib/flatpak/exports/share:/usr/local/share:/usr/sha
re/"
declare -x XDG_MENU_PREFIX="gnome-"
declare -x XDG_RUNTIME_DIR="/run/user/1000"
declare -x XDG_SESSION_CLASS="user"

```

6. ssh : Secure Shell command in Linux

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ ssh localhost  
ssh: connect to host localhost port 22: Connection refused  
[jaivikjariwala@nomad ~]$
```

7. zip : Zip files in Linux

```
jaivikjariwala@nomad:~ — unzip xyz.zip  
[jaivikjariwala@nomad ~]$ ssh localhost  
ssh: connect to host localhost port 22: Connection refused  
[jaivikjariwala@nomad ~]$ zip myfile.zip wolf.txt  
zip warning: name not matched: wolf.txt  
  
zip error: Nothing to do! (myfile.zip)  
[jaivikjariwala@nomad ~]$ ls  
Desktop Downloads node_modules package.json Pictures Templates wolf1  
Documents Music one.txt package-lock.json Public Videos xyz.txt  
[jaivikjariwala@nomad ~]$ zip xyz.zip xyz.txt  
adding: xyz.txt (stored 0%)  
[jaivikjariwala@nomad ~]$ unzip xyz.zip  
Archive: xyz.zip  
replace xyz.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

8. unzip : Unzip files in Linux

9. ps : Display active processes

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ ps  
  PID TTY          TIME CMD  
 15475 pts/1    00:00:00 bash  
 17127 pts/1    00:00:00 ps  
[jaivikjariwala@nomad ~]$
```

10. uname : Linux command to get basic information about the OS

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ ps  
  PID TTY          TIME CMD  
 15475 pts/1    00:00:00 bash  
 17127 pts/1    00:00:00 ps  
[jaivikjariwala@nomad ~]$ uname  
Linux  
[jaivikjariwala@nomad ~]$
```

11. chown : Command for granting ownership of files or folders

```
jaivikjariwala@nomad:~ — sudo chown test sample
[jaivikjariwala@nomad ~]$ chown NewUser FILE
chown: invalid user: 'NewUser'
[jaivikjariwala@nomad ~]$ chown Nomad FILE
chown: invalid user: 'Nomad'
[jaivikjariwala@nomad ~]$ chown jaivikjariwala@nomad FILE
chown: invalid user: 'jaivikjariwala@nomad'
[jaivikjariwala@nomad ~]$ chown test sample
chown: invalid user: 'test'
[jaivikjariwala@nomad ~]$ chown --version
chown (GNU coreutils) 9.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie and Jim Meyering.
[jaivikjariwala@nomad ~]$ sudo chown test sample
[sudo] password for jaivikjariwala:
```

12. wget : Direct download files from the internet

```
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ wget http://example.com/sample.php
--2023-01-25 04:51:37-- http://example.com/sample.php
Resolving example.com (example.com)... 2606:2800:220:1:248:1893:25c8:1946, 93.184.216.34
Connecting to example.com (example.com)|2606:2800:220:1:248:1893:25c8:1946|:80..
. connected.
HTTP request sent, awaiting response... 404 Not Found
2023-01-25 04:51:38 ERROR 404: Not Found.
```

13. ufw : Firewall command

```
[jaivikjariwala@nomad ~]$ ufw
bash: ufw: command not found...
Install package 'ufw' to provide command 'ufw'? [N/y] y

* Waiting in queue...
The following packages have to be installed:
ufw-0.35-26.fc37.noarch      Uncomplicated Firewall
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
* Waiting in queue... Failed to install packages: Curl error (6): Couldn't resolve host name for https://mirrors.fedoraproject.org/metalink?repo=fedora-37&arch=x86_64 [Could not resolve host: mirrors.fedoraproject.org]
```

14. traceroute : Trace all the network hops to reach the destination

```
jaivikjariwala@nomad:~$ traceroute
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [
  -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ] [ -w MAX,
  HERE,NEAR ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [ --fwmark=num ] hos
  t [ packetlen ]
Options:
  -4                      Use IPv4
  -6                      Use IPv6
  -d --debug              Enable socket level debugging
  -F --dont-fragment      Do not fragment packets
  -f first_ttl --first=first_ttl
                          Start from the first_ttl hop (instead from 1)
  -g gate,... --gateway=gate,...
                          Route packets through the specified gateway
                          (maximum 8 for IPv4 and 127 for IPv6)
  -I --icmp               Use ICMP ECHO for tracerouting
  -T --tcp                Use TCP SYN for tracerouting (default port is 80)
  -i device --interface=device
                          Specify a network interface to operate with
  -m max_ttl --max-hops=max_ttl
                          Set the max number of hops (max TTL to be
                          reached). Default is 30
  -N squeries --sim-queries=squeries
                          Set the number of probes to be tried
                          simultaneously (default is 16)
  -n                      Do not resolve IP addresses to their domain names
  -p port --port=port     Set the destination port to use. It is either
                          initial udp port value for "default" method
                          (incremented by each probe, default is 33434), or
                          initial seq for "icmp" (incremented as well,
                          default from 1), or some constant destination
                          port for other methods (with default of 80 for
```

15. service : Linux command to start and stop services

```
jaivikjariwala@nomad ~]$ service
Usage: service < option > | --status-all | [ service_name [ command | --full-restart ] ]
jaivikjariwala@nomad ~]$
```

16. alias : Create custom shortcuts for your regularly used commands

```
jaivikjariwala@nomad:~$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias xzgrep='xzgrep --color=auto'
alias xzfgrep='xzfgrep --color=auto'
alias xzgrep='xzgrep --color=auto'
alias zegrep='zegrep --color=auto'
alias zfgrep='zfgrep --color=auto'
alias zgrep='zgrep --color=auto'
jaivikjariwala@nomad ~]$
```

17. dd : Majorly used for creating bootable USB sticks

```
jaivikjariwala@nomad:~ — dd
[jaivikjariwala@nomad ~]$ dd
ibm
asosrc
```

18. whereis : Locate the binary, source, and manual pages for a command

```
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ whereis -V
whereis from util-linux 2.38.1
[jaivikjariwala@nomad ~]$
```

19. whatis : Find what a command is used for

```
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ whereis -V
whereis from util-linux 2.38.1
[jaivikjariwala@nomad ~]$ whatis -version
whatis what?
[jaivikjariwala@nomad ~]$
```

20. diff : Find the difference between two files

```
jaivikjariwala@nomad:~
[jaivikjariwala@nomad ~]$ diff --help
Usage: diff [OPTION]... FILES
Compare FILES line by line.

Mandatory arguments to long options are mandatory for short options too.
  --normal                output a normal diff (the default)
  -q, --brief             report only when files differ
  -s, --report-identical-files report when two files are the same
  -c, -C NUM, --context[=NUM] output NUM (default 3) lines of copied context
  -u, -U NUM, --unified[=NUM] output NUM (default 3) lines of unified context
  -e, --ed               output an ed script
  -n, --rcs              output an RCS format diff
  -y, --side-by-side      output in two columns
  -W, --width=NUM         output at most NUM (default 130) print columns
  --left-column           output only the left column of common lines
  --suppress-common-lines do not output common lines

  -p, --show-c-function    show which C function each change is in
  -F, --show-function-line=RE show the most recent line matching RE
  --label LABEL           use LABEL instead of file name and timestamp
                        (can be repeated)
```

21. In : Create symbolic links (shortcuts) to other files

```
[jaivikjariwala@nomad ~]$ In
bash: In: command not found...
[jaivikjariwala@nomad ~]$
```

22. top : View active processes live with their system usage

```
jaivikjariwala@nomad:~ — top
top - 14:28:46 up 13 min, 1 user, load average: 0.24, 0.29, 0.22
Tasks: 384 total, 1 running, 383 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.4 us, 0.2 sy, 0.0 ni, 99.2 id, 0.0 wa, 0.2 hi, 0.0 si, 0.0 st
MiB Mem : 15614.5 total, 12031.6 free, 1263.8 used, 2319.1 buff/cache
MiB Swap: 8192.0 total, 8192.0 free, 0.0 used, 13614.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3971	jaivikj+	20	0	772152	57936	44636	S	5.3	0.4	0:10.18	gnome-terminal-
2146	jaivikj+	20	0	8018936	216588	125552	S	3.7	1.4	1:09.19	gnome-shell
513	root	-51	0	0	0	0	S	1.0	0.0	0:02.76	irq/79-ASUP1205:00
901	systemd+	20	0	16156	7932	7004	S	0.3	0.0	0:02.21	systemd-oomd
2415	root	20	0	259640	29744	8056	S	0.3	0.2	0:00.58	sssd_kcm
5120	jaivikj+	20	0	224996	3880	3040	R	0.3	0.0	0:00.03	top
1	root	20	0	172164	17840	11060	S	0.0	0.1	0:01.74	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	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	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
9	root	20	0	0	0	0	I	0.0	0.0	0:01.48	kworker/u40:0-btrfs-endio-write
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
16	root	20	0	0	0	0	I	0.0	0.0	0:00.99	rcu_preempt
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.01	migration/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
21	root	rt	0	0	0	0	S	0.0	0.0	0:00.11	migration/1

23. useradd : Add new user or change existing users data

```
jaivikjariwala@nomad:~  
; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 62011  
; flags: qr aa rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1  
  
; OPT PSEUDOSECTION:  
EDNS: version: 0, flags:; udp: 65494  
; QUESTION SECTION:  
.                IN      NS  
  
; Query time: 1 msec  
; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)  
; WHEN: Thu Feb 02 16:07:27 EST 2023  
; MSG SIZE rcvd: 28  
  
jaivikjariwala@nomad ~]$ talk  
bash: talk: command not found...  
Install package 'talk' to provide command 'talk'? [N/y] y  
  
* Waiting in queue...  
The following packages have to be installed:  
talk-0.17-66.fc37.x86_64      Talk client for one-on-one Internet chatting  
Proceed with changes? [N/y] y
```

24. man : Access manual pages for all Linux commands

25. cp : Similar usage as mv but for copying files in Linux

```
jaivikjariwala@nomad:~  
jaivikjariwala@nomad ~]$ cp xyz.txt  
cp: missing destination file operand after 'xyz.txt'  
Try 'cp --help' for more information.  
jaivikjariwala@nomad ~]$ cp wolf1  
cp: missing destination file operand after 'wolf1'
```

26. ln : Create symbolic links (shortcuts) to other files

```
jaivikjariwala@nomad:~  
jaivikjariwala@nomad ~]$ ln  
bash: ln: command not found...  
jaivikjariwala@nomad ~]$
```


27. netstat : netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc.

```
jaivikjariwala@nomad:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      1 nomad:39044             10.30.1.14:domain      SYN_SENT
^[[3~udp        0      0 nomad:bootpc            1.1.1.1:bootps        ESTABLISHED

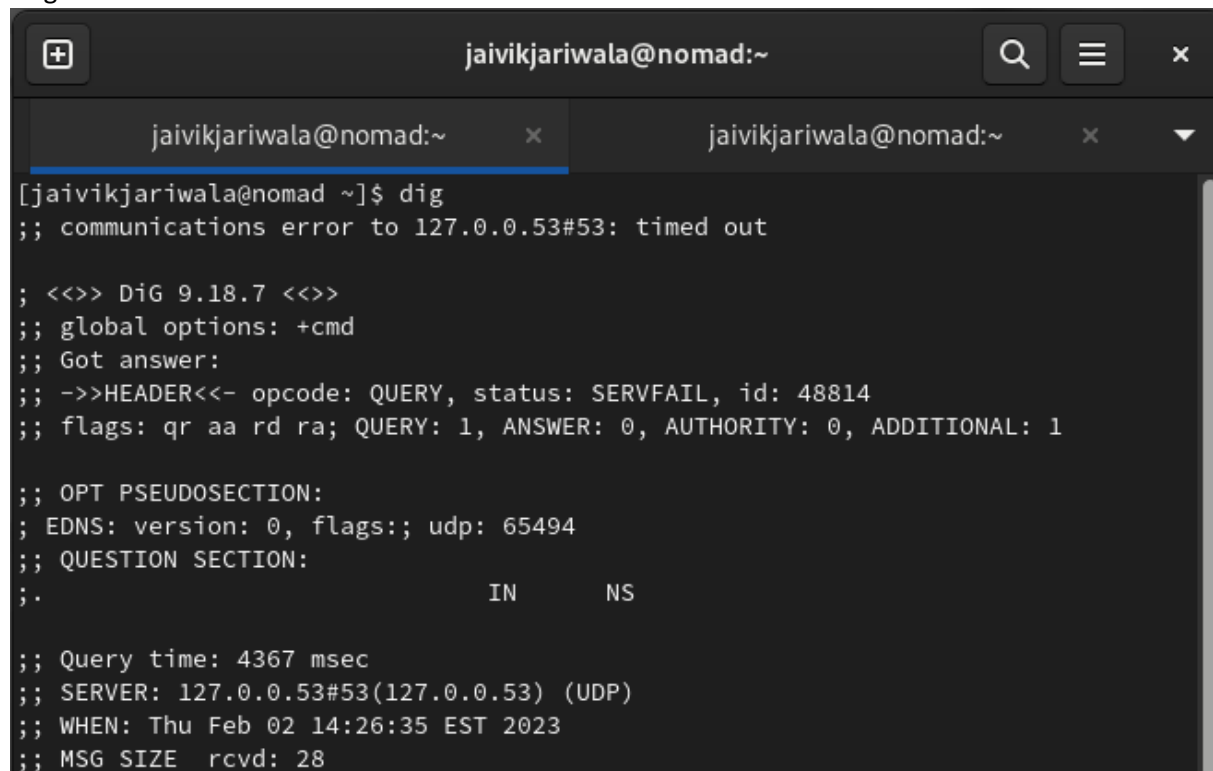
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State         I-Node  Path
unix   3      [ ]      STREAM    CONNECTED    41743    /run/systemd/journal/stdout
unix   3      [ ]      DGRAM     CONNECTED    19099    /run/systemd/journal/stdout
unix   3      [ ]      STREAM    CONNECTED    43839    /run/systemd/journal/stdout
unix   3      [ ]      STREAM    CONNECTED    40205    /run/user/1000/bus
unix   3      [ ]      STREAM    CONNECTED    26854    /run/user/1000/bus
unix   3      [ ]      STREAM    CONNECTED    37861    /run/user/1000/bus
unix   3      [ ]      STREAM    CONNECTED    37750    /run/user/1000/wayland-0
unix   3      [ ]      STREAM    CONNECTED    24865    /run/user/1000/wayland-0
unix   3      [ ]      STREAM    CONNECTED    16304    /run/systemd/journal/stdout
unix   3      [ ]      STREAM    CONNECTED    37679    /run/user/1000/pipewire-0
unix   3      [ ]      DGRAM     CONNECTED    40992    /run/systemd/journal/stdout
unix   3      [ ]      STREAM    CONNECTED    25672    /run/systemd/journal/stdout
unix   3      [ ]      STREAM    CONNECTED    55772    /run/systemd/journal/stdout
unix   3      [ ]      STREAM    CONNECTED    33831    /run/systemd/journal/stdout
unix   3      [ ]      STREAM    CONNECTED    37866    /run/user/1000/bus
unix   3      [ ]      STREAM    CONNECTED    35267    /run/user/1000/bus
```

28. nslookup : A network utility program used to obtain information about Internet servers. As its name suggests, the utility finds name server information for domains by querying DNS.

```
jaivikjariwala@nomad:~ — nslookup
jaivikjariwala@nomad:~$ nslookup
> ls
The 'ls' command is not implemented.
> wolf1
;; communications error to 127.0.0.53#53: timed out
Server:         127.0.0.53
Address:        127.0.0.53#53

** server can't find wolf1.pdu.ac.in: SERVFAIL
>
```

29. dig : dig is a tool for querying DNS nameservers for information about host addresses, mail exchanges, nameservers, and related information. This tool can be used from any Linux (Unix) or Macintosh OS X operating system. The most typical use of dig is to simply query a single host.



```
[jaivikjariwala@nomad ~]$ dig
;; communications error to 127.0.0.53#53: timed out

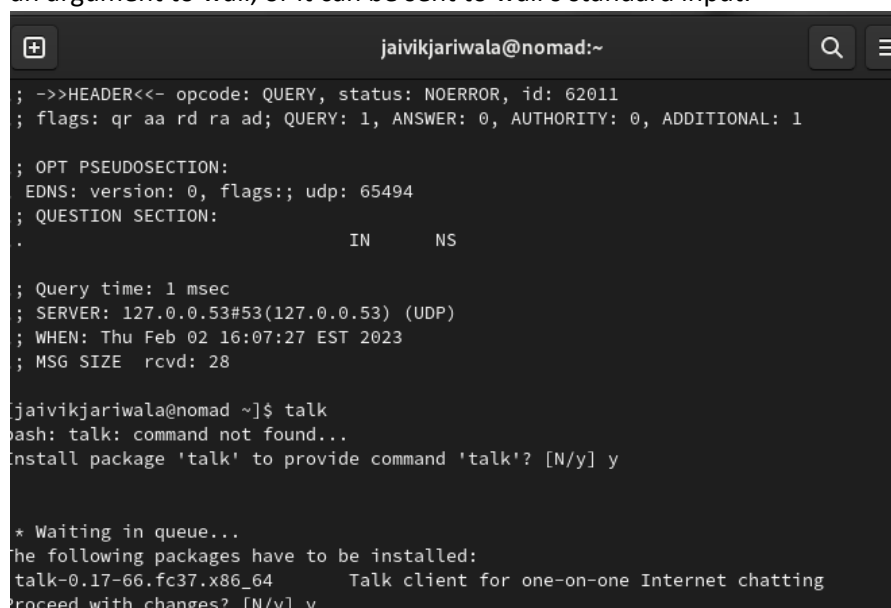
; <<>> DiG 9.18.7 <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 48814
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; udp: 65494
;; QUESTION SECTION:
;                               IN      NS

;; Query time: 4367 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu Feb 02 14:26:35 EST 2023
;; MSG SIZE rcvd: 28
```

30. uptime : You have just connected to your Linux Server Machine and founds Something unusual or malicious, what you will do? Guessing.... NO, definitely not you could run uptime to verify what happened actually when the server was unattended.

31. wall : one of the most important command for administrator, wall sends a message to everybody logged in with their mesg permission set to “yes”. The message can be given as an argument to wall, or it can be sent to wall’s standard input.



```
[jaivikjariwala@nomad ~]$ talk
bash: talk: command not found...
install package 'talk' to provide command 'talk'? [N/y] y

* Waiting in queue...
the following packages have to be installed:
talk-0.17-66.fc37.x86_64      Talk client for one-on-one Internet chatting
proceed with changes? [N/y] y
```

32. **mesg** : Lets you control if people can use the “write” command, to send text to you over the screen.

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ mesg  
is y  
[jaivikjariwala@nomad ~]$
```

33. **write** : Let you send text directly to the screen of another Linux machine if ‘mesg’ is ‘y’.

```
jaivikjariwala@nomad ~]$ write  
Try 'write --help' for more information.  
jaivikjariwala@nomad ~]$ write wolf1  
write: wolf1 is not logged in
```

34. **talk** : An enhancement to write command, talk command lets you talk to the logged in users.

```
jaivikjariwala@nomad:~ — bash  
[jaivikjariwala@nomad ~]$ talk  
bash: talk: command not found...
```

35. **w** : what command ‘w’ seems you funny? But actually it is not. t’s a command, even if it’s just one letter long! The command “w” is a combination of uptime and who commands given one immediately after the other, in that order.

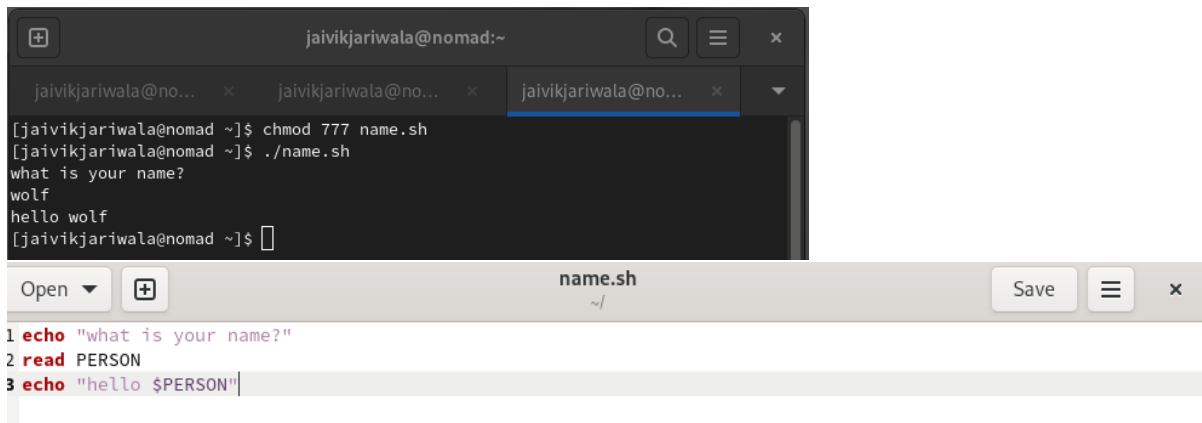
36. **rename** : As the name suggests, this command rename files. rename will rename the specified files by replacing the first occurrence from the file name.

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ free  
              total        used        free      shared  buff/cache   available  
Mem:          15989212     1264344     12430512        405872       2294356       14003804  
Swap:           8388604             0         8388604  
[jaivikjariwala@nomad ~]$ w  
14:23:04 up 7 min,  1 user,  load average: 0.32, 0.44, 0.24  
USER      TTY      LOGIN@  IDLE   JCPU   PCPU   WHAT  
jaivikja  tty2      14:16   7:48   0.02s  0.02s  /usr/libexec/gnome-session-bina  
[jaivikjariwala@nomad ~]$
```

37. **free** : Keeping track of memory and resources is as much important, as any other task performed by an administrator, and ‘free’ command comes to rescue here.

```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ free  
              total        used        free      shared  buff/cache   available  
Mem:          15989212     1264344     12430512        405872       2294356       14003804  
Swap:           8388604             0         8388604  
[jaivikjariwala@nomad ~]$
```

1) Write a shell script to print your name.



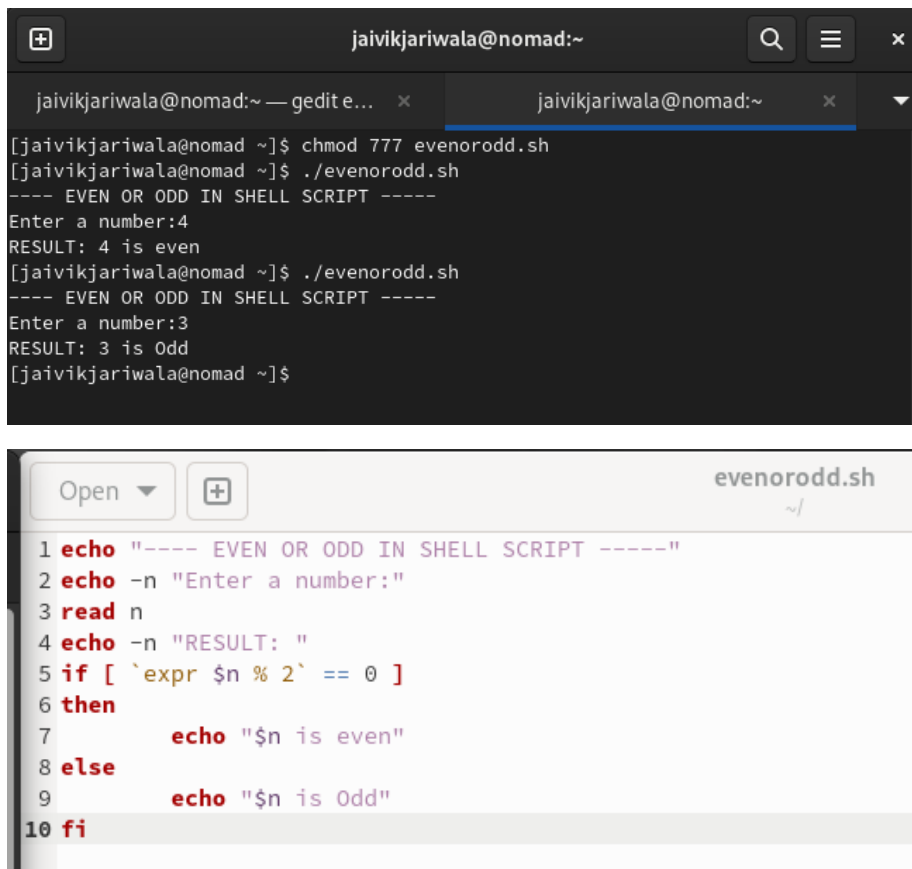
The image shows a terminal window and a code editor. The terminal window displays the following commands and output:

```
[jaivikjariwala@nomad ~]$ chmod 777 name.sh
[jaivikjariwala@nomad ~]$ ./name.sh
what is your name?
wolf
hello wolf
[jaivikjariwala@nomad ~]$
```

The code editor shows the content of the file `name.sh`:

```
1 echo "what is your name?"
2 read PERSON
3 echo "hello $PERSON"
```

2) Write a shell script to find whether a number is even or odd.



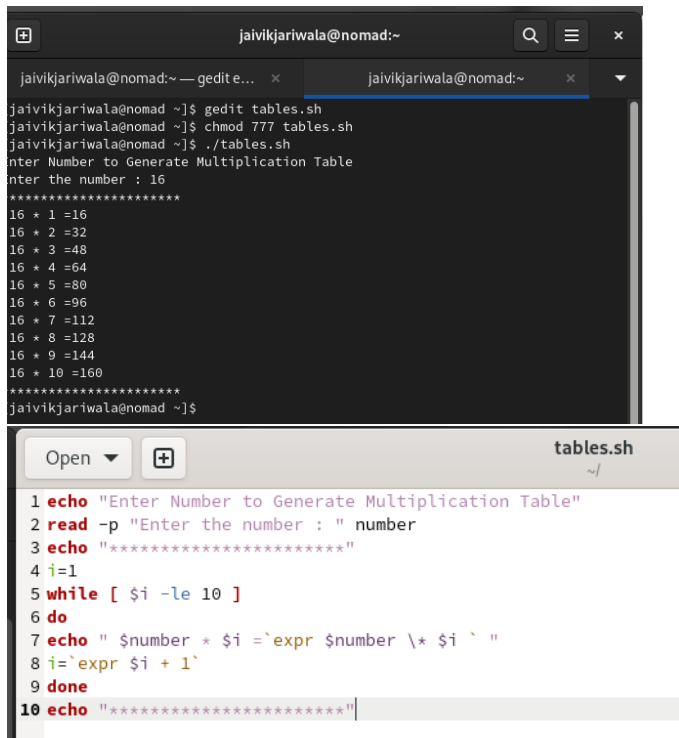
The image shows a terminal window and a code editor. The terminal window displays the following commands and output:

```
[jaivikjariwala@nomad ~]$ chmod 777 evenorodd.sh
[jaivikjariwala@nomad ~]$ ./evenorodd.sh
---- EVEN OR ODD IN SHELL SCRIPT ----
Enter a number:4
RESULT: 4 is even
[jaivikjariwala@nomad ~]$ ./evenorodd.sh
---- EVEN OR ODD IN SHELL SCRIPT ----
Enter a number:3
RESULT: 3 is Odd
[jaivikjariwala@nomad ~]$
```

The code editor shows the content of the file `evenorodd.sh`:

```
1 echo "---- EVEN OR ODD IN SHELL SCRIPT ----"
2 echo -n "Enter a number:"
3 read n
4 echo -n "RESULT: "
5 if [ `expr $n % 2` == 0 ]
6 then
7     echo "$n is even"
8 else
9     echo "$n is Odd"
10 fi
```

3) Write a script to print a table of a given number.

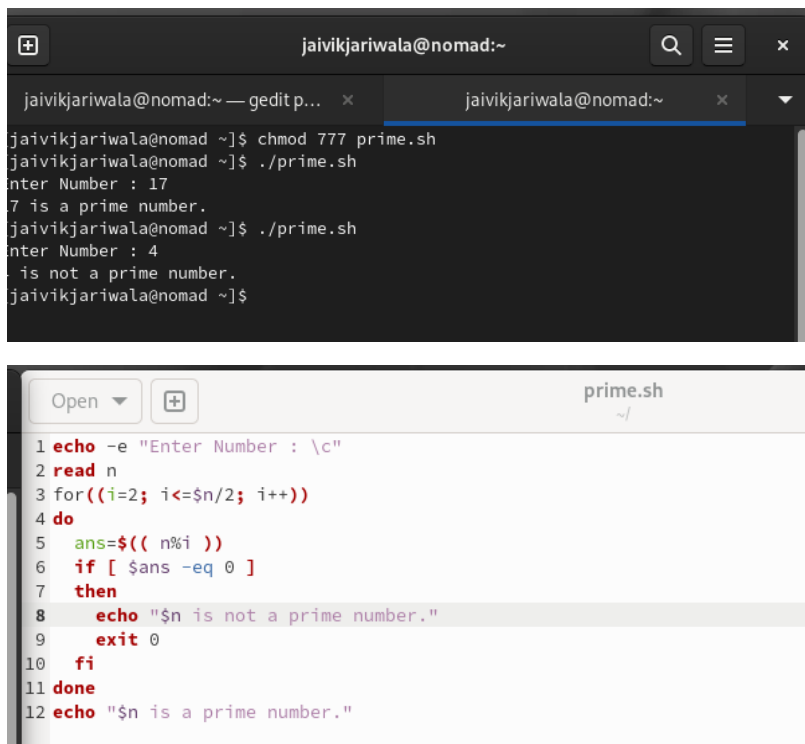


The image shows a terminal window and a code editor. The terminal window displays the execution of a script named `tables.sh`. The user enters the number 16, and the script prints a multiplication table for 16, showing products from 16 * 1 to 16 * 10. The code editor shows the script's content, which uses a `while` loop to generate the table.

```
jaivikjariwala@nomad:~$ gedit tables.sh
jaivikjariwala@nomad:~$ chmod 777 tables.sh
jaivikjariwala@nomad:~$ ./tables.sh
Enter Number to Generate Multiplication Table
Enter the number : 16
*****
16 * 1 =16
16 * 2 =32
16 * 3 =48
16 * 4 =64
16 * 5 =80
16 * 6 =96
16 * 7 =112
16 * 8 =128
16 * 9 =144
16 * 10 =160
*****
jaivikjariwala@nomad:~$
```

```
1 echo "Enter Number to Generate Multiplication Table"
2 read -p "Enter the number : " number
3 echo "*****"
4 i=1
5 while [ $i -le 10 ]
6 do
7 echo " $number * $i ="`expr $number \* $i ` "
8 i=`expr $i + 1`
9 done
10 echo "*****"
```

4) Write a shell script to check whether a given no. is prime or not.

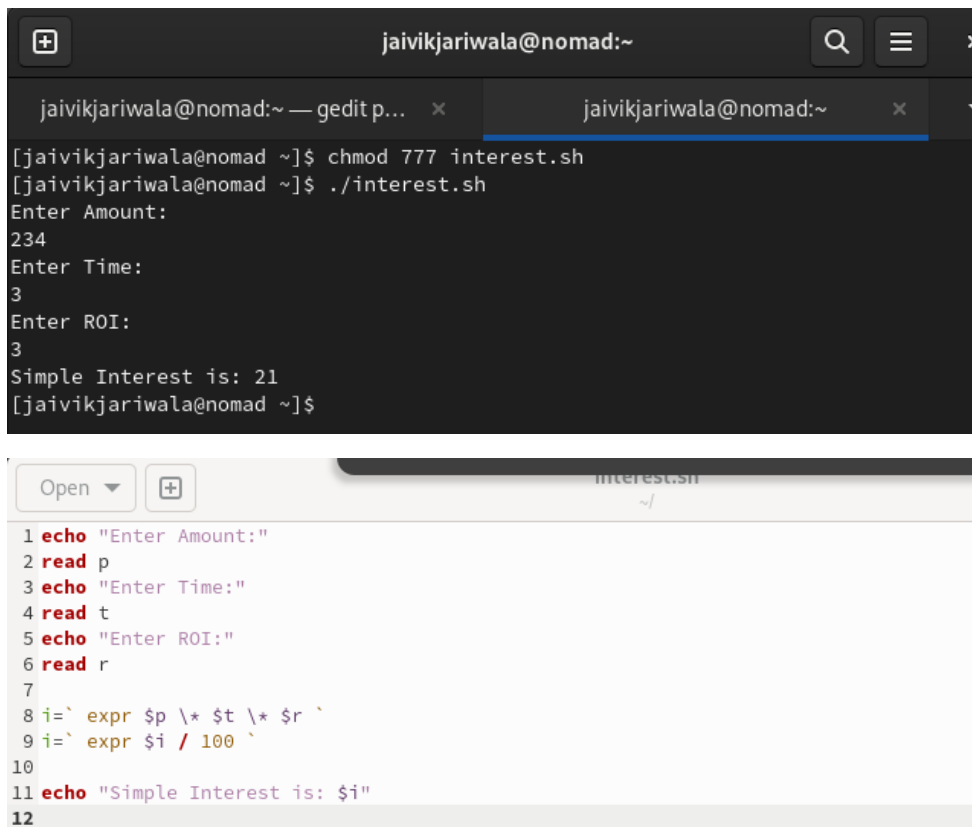


The image shows a terminal window and a code editor. The terminal window displays the execution of a script named `prime.sh`. The user enters the number 17, and the script outputs "17 is a prime number." The user then enters the number 4, and the script outputs "4 is not a prime number." The code editor shows the script's content, which uses a `for` loop to check for divisors of the input number.

```
jaivikjariwala@nomad:~$ chmod 777 prime.sh
jaivikjariwala@nomad:~$ ./prime.sh
Enter Number : 17
17 is a prime number.
jaivikjariwala@nomad:~$ ./prime.sh
Enter Number : 4
4 is not a prime number.
jaivikjariwala@nomad:~$
```

```
1 echo -e "Enter Number : \c"
2 read n
3 for((i=2; i<=$n/2; i++))
4 do
5 ans=$(( n%i ))
6 if [ $ans -eq 0 ]
7 then
8 echo "$n is not a prime number."
9 exit 0
10 fi
11 done
12 echo "$n is a prime number."
```

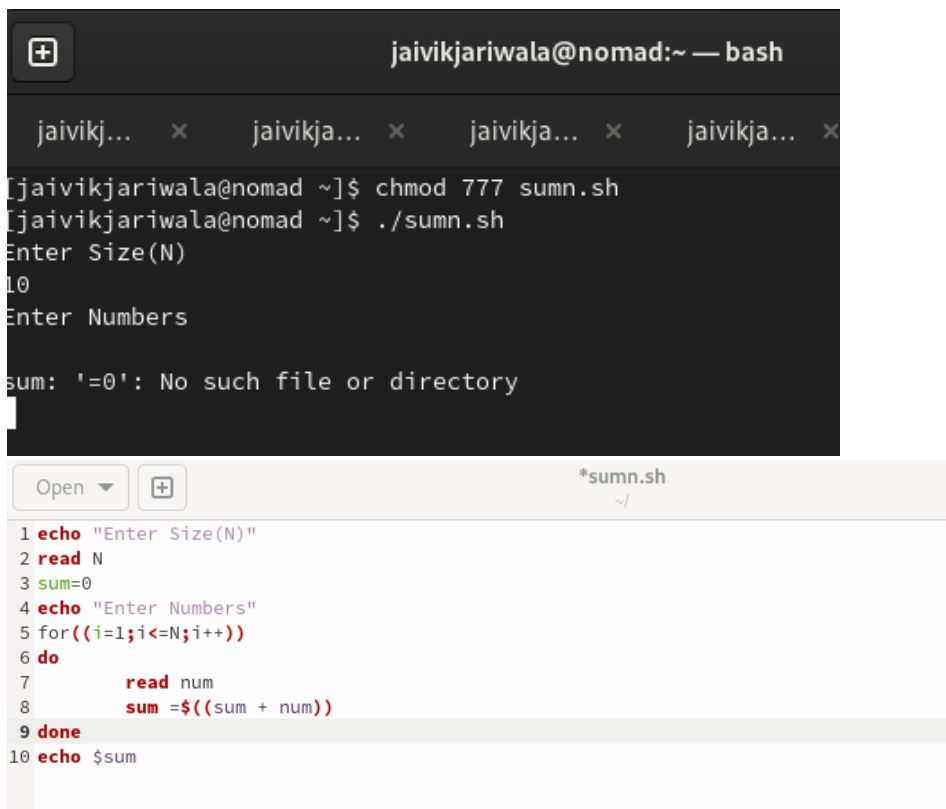
5) Write a shell script to find the simple interest.



The image shows a terminal window and a script editor. The terminal window, titled 'jaivikjariwala@nomad:~', shows the execution of a script named 'interest.sh'. The user sets permissions with 'chmod 777 interest.sh' and runs './interest.sh'. The script prompts for 'Amount' (234), 'Time' (3), and 'ROI' (3), then calculates and displays 'Simple Interest is: 21'. The script editor, titled 'interest.sh', shows the following code:

```
1 echo "Enter Amount:"
2 read p
3 echo "Enter Time:"
4 read t
5 echo "Enter ROI:"
6 read r
7
8 i=`expr $p \* $t \* $r `
9 i=`expr $i / 100 `
10
11 echo "Simple Interest is: $i"
12
```

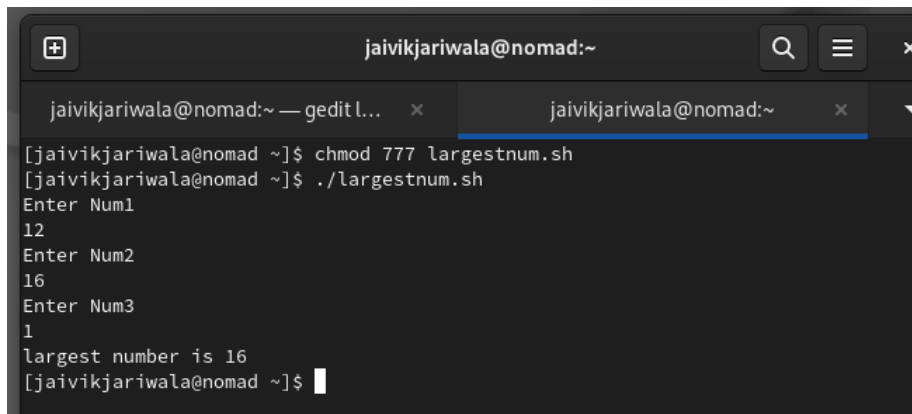
6) Write a shell script to find sum of “n” numbers.



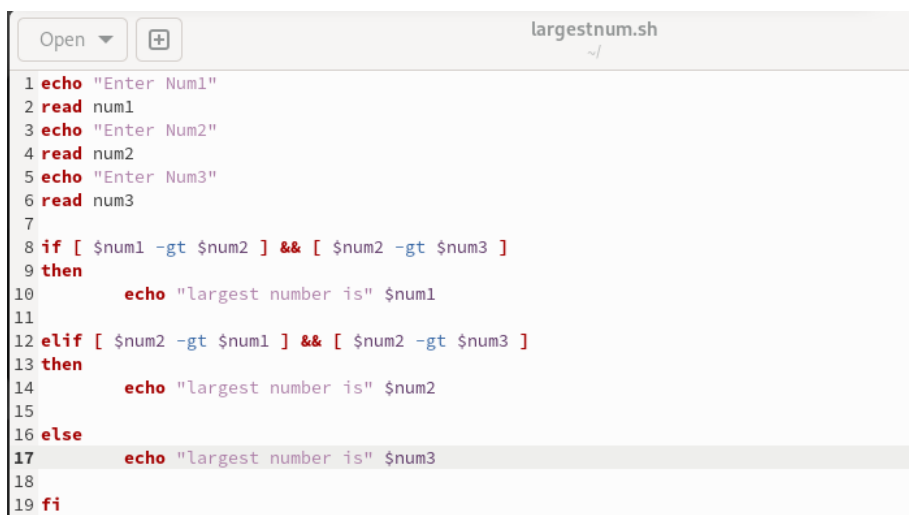
The image shows a terminal window and a script editor. The terminal window, titled 'jaivikjariwala@nomad:~ — bash', shows the execution of a script named 'sumn.sh'. The user sets permissions with 'chmod 777 sumn.sh' and runs './sumn.sh'. The script prompts for 'Size(N)' (10) and 'Numbers'. It then displays an error message: 'sum: '=': No such file or directory'. The script editor, titled '*sumn.sh', shows the following code:

```
1 echo "Enter Size(N)"
2 read N
3 sum=0
4 echo "Enter Numbers"
5 for((i=1;i<=N;i++))
6 do
7     read num
8     sum=$((sum + num))
9 done
10 echo $sum
```

7) Write a shell script to find the largest number of three numbers.



```
jaivikjariwala@nomad:~  
[jaivikjariwala@nomad ~]$ chmod 777 largestnum.sh  
[jaivikjariwala@nomad ~]$ ./largestnum.sh  
Enter Num1  
12  
Enter Num2  
16  
Enter Num3  
1  
largest number is 16  
[jaivikjariwala@nomad ~]$
```



```
largestnum.sh  
~/  
1 echo "Enter Num1"  
2 read num1  
3 echo "Enter Num2"  
4 read num2  
5 echo "Enter Num3"  
6 read num3  
7  
8 if [ $num1 -gt $num2 ] && [ $num2 -gt $num3 ]  
9 then  
10     echo "largest number is" $num1  
11  
12 elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]  
13 then  
14     echo "largest number is" $num2  
15  
16 else  
17     echo "largest number is" $num3  
18  
19 fi
```

Write a menu driven shell script, which will print the following menu and execute the given task.

- ❖ Display a calendar of current month
- ❖ Display today's date and time
- ❖ Display username those are currently logged in the system
- ❖ Display your name at the given x,y position.
- ❖ Display your terminal number

```
1  echo "Menu"
2  echo "1) Display Calendar of current month"
3  echo "2) Display Today's date and time"
4  echo "3) Display Username who is currently logged in"
5  echo "4) Display your name at given x,y position"
6  echo "5) Display your terminal number"
7  echo "6) exit"
8  echo "Enter your choice:"
9  read ch
10 case $ch in
11 1) cal;;
12 2) date;;
13 3) who;;
14 4) row=$(tput lines)
15    col=$(tput cols)
16    echo "Number of rows $row Cols=$col"
17    echo "Enter desired x,y position"
18    echo "x position="
19    read x
20    echo "y position="
21    read y
```

```
Menu
1) Display Calendar of current month
2) Display Today's date and time
3) Display Username who is currently logged in
4) Display your name at given x,y position
5) Display your terminal number
6) exit
Enter your choice:
1
```


Write a shell script which will generate first n Fibonacci numbers such as :1, 1, 2, 3, 5, 13,....etc

```
1 fibo()
2 {
3     no=$1
4     if [ $no -eq 1 ]; then
5         return 0
6     elif [ $no -eq 2 ]; then
7         return 1
8     else
9         a1=`expr $no - 1`
10        fibo $a1
11        a=$(echo $?)
12
13        b1=`expr $no - 1`
14        fibo $b1
15        b=$(echo $?)
16
17        c=`expr $a + $b`
18        return $c
19    fi
20 }
```

```
20 }
21
22 clear
23
24 echo -e "Enter number of terms : \c"
25 read n
26
27 if [ $n -gt 0 ]; then
28     for (( i=1; i<=$n; i++ ))
29     do
30         fibo $i
31         echo $?
32     done
33 else
34     echo -e "Invalid input."
35 fi
36
37
38
```

```
Enter number of terms : 6
0 1 1 2 3 5
```

Shell Script to print half pyramids using numbers

```
1 rows=5
2 for((i=rows; i>=1; i--))
3 do
4     for((j=1; j<=i; j++))
5     do
6         echo -n "$j "
7     done
8     echo
9 done
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Write a shell script to find the reverse of a given number.

```
1 read -p "Enter a number: " number
2 temp=$number
3 while [ $temp -ne 0 ]
4 do
5     reverse=$reverse$((temp%10))
6     temp=$((temp/10))
7 done
8 echo "Reverse of $number is $reverse"
```

```
Enter a number: 143
Reverse of 143 is 341
```

Write a shell script to find the sum of two floating point numbers

```
1 echo enter a and b
2 read a b
3 c=`echo $a+$b | bc`
4 echo $c
```

enter a and b
4.5 6.7
11.2

Write a shell script that changes text to uppercase

```
1 echo "enter the string:"
2 read string
3 echo "the string in uppercase is:"
4 echo ${string^^}
```

```
enter the string:
rashida
the string in uppercase is:
RASHIDA
```

Write a shell script which prints "invalid no. of arguments" if more than 5 command line arguments otherwise print "valid no. of arguments"

```
6 7 8 5 4
valid no. of arguments
```

```
1 echo $#
2
3 if [ $# -le 5 ]
4 then
5     echo "valid no of arguments"
6 else
7     echo "Invalid no of arguments"
8
9 fi
```

Write a shell script to make the following operations menu based:

❖ Addition ❖ Subtraction ❖ Multiplication ❖ Division

```
echo -e "Addition: \nSubtraction: -\nMultiplication: x\nDivision: /\n"
read op

case $op in
    +) c=expr $a $b
       echo "Sum of $a and $b is $c";;
    -) c=expr $a - $b
       echo "Difference of $a and $b is $c";;
    x) c="expr $a * $b"
       echo "Product of $a and $b is $c";;
    /) c="expr $a / $b";;
       echo "Division of $a and $b is $c";;
    .) echo "Invalid Operator"
       exit;;
    *)
esac
```

Enter the first number

23

Enter the second number

45

Enter the operator:

Addition: +

Subtraction:

Multiplication: x

Division: /

Sum of 23 and 45 is 68

Write a shell script to find the sum of all digits for a given number

```
1  echo -n "Enter number: "  
2  read n  
3  sd=0  
4  sum=0  
5  while [ $n -gt 0 ]  
6  do  
7      sd=$(( $n % 10 ))  
8      n=$(( $n / 10 ))  
9      sum=$(( $sum + $sd ))  
10  
11 done  
12 echo "Sum of all digit is $sum"  
13
```

```
Enter number: 67  
  
Sum of all digit is 13
```

Write a shell script to find the factorial of a given number.

```
1  echo "Enter a number"  
2  read num  
3  fact=1  
4  while [ $num -gt 1 ]  
5  do  
6      fact=$((fact * num))  
7      num=$((num - 1))  
8  done  
9  echo Factorial=$fact
```

```
Enter a number: 7  
Factorial=5040
```

Scheduling Algorithm

```
def fcfs(processes):  
    #initialize the waiting time and turn around time  
    time=0  
    waiting_time=0  
    completion_time=0  
    turn_around_time=0  
  
    #execute the processes one by one in fcfs  
    for p in processes:  
        #update waiting time for current process  
        waiting_time += time  
        #execute the current process  
        print("executing process", p.pid, "with burst time" , p.burst_time ,  
"with arrival time", p.pid)  
        time += p.burst_time  
        #calculate turnaround time for current process  
        turn_around_time += time  
        #calculate completion time for current process  
        completion_time += time  
        #print the current process details  
        print("waiting time", waiting_time)  
        print("turn around time", turn_around_time)  
        print("completion time", completion_time)  
  
    #print the total burst and total average time  
    total_waiting_time = waiting_time  
    average_waiting_time = total_waiting_time / len(processes)  
    print("total waiting time", total_waiting_time)  
    print("average waiting time", average_waiting_time)
```

```
fcfs(processes)
```

executing process 1 with burst time 10 with arrival time 1
waiting time 0
turn around time 10
completion time 10
executing process 2 with burst time 5 with arrival time 2
waiting time 10
turn around time 25
completion time 25
executing process 3 with burst time 8 with arrival time 3
waiting time 25
turn around time 48
completion time 48
executing process 4 with burst time 3 with arrival time 4
waiting time 48
turn around time 74
completion time 74
total waiting time 48
avarage waiting time 12.0

```

# Define a function to execute the processes using SJF scheduling algorithm
def sjf(processes):
    # Sort the processes by their burst time in non-decreasing order
    processes.sort(key=lambda p: p.burst_time)

    # Initialize the time and waiting time variables
    time = 0
    waiting_time = 0
    completion_time = 0
    turn_around_time = 0

    # Execute the processes one by one in SJF order
    for p in processes:
        #update waiting time for current process
        waiting_time += time
        #execute the current process
        print("executing process", p.pid, "with burst time" , p.burst_time ,
"with arrival time", p.pid)
        time += p.burst_time
        #calculate turnaround time for current process
        turn_around_time += time
        #calculate completion time for current process
        completion_time += time
        #print the current process details
        print("waiting time", waiting_time)
        print("turn around time", turn_around_time)
        print("completion time", completion_time)

    # Print the total waiting time and average waiting time
    total_waiting_time = waiting_time
    average_waiting_time = total_waiting_time / len(processes)
    print("Total waiting time:", total_waiting_time)
    print("Average waiting time:", average_waiting_time)

sjf(processes)

```

executing process 4 with burst time 3 with arrival time 4
waiting time 0
turn around time 3
completion time 3
executing process 2 with burst time 5 with arrival time 2
waiting time 3
turn around time 11
completion time 11
executing process 3 with burst time 8 with arrival time 3
waiting time 11
turn around time 27
completion time 27
executing process 1 with burst time 10 with arrival time 1
waiting time 27
turn around time 53
completion time 53
Total waiting time: 27
Average waiting time: 6.75


```

# Define a function to execute the processes using Round Robin scheduling
algorithm
def round_robin(processes, quantum):
    # Initialize the waiting time, turnaround time, completion time, and time
    variables
    waiting_time = [0] * len(processes)
    turnaround_time = [0] * len(processes)
    completion_time = [0] * len(processes)
    time = 0

    # Create a copy of the processes list to keep track of the remaining burst
    time of each process
    remaining_burst_time = [p.burst_time for p in processes]

    # Execute the processes one by one in Round Robin order
    while True:
        all_processes_completed = True
        for i, p in enumerate(processes):
            if remaining_burst_time[i] > 0:
                # Calculate the time slice for the current process
                time_slice = min(remaining_burst_time[i], quantum)

                # Execute the current process for the time slice
                print("Executing process", p.pid, "for", time_slice, "units of
time")

                remaining_burst_time[i] -= time_slice
                time += time_slice

            # Check if the current process has completed
            if remaining_burst_time[i] == 0:
                # Calculate turnaround time and completion time for the
current process

                turnaround_time[i] = time - p.arrival_time
                completion_time[i] = time

                # Print the waiting time, turnaround time, and completion
time for the current process
                print("Process", p.pid, "completed at time",
completion_time[i])
                print("Waiting time for process", p.pid, "is",
waiting_time[i])
                print("Turnaround time for process", p.pid, "is",
turnaround_time[i])
                print("Completion time for process", p.pid, "is",
completion_time[i])

```

```

        else:
            # Update the waiting time for the current process
            waiting_time[i] = time - p.arrival_time

        all_processes_completed = False

        # Check if all processes have completed
        if all_processes_completed:
            break

        # Calculate the total waiting time and average waiting time for all
        # processes
        total_waiting_time = sum(waiting_time)
        average_waiting_time = total_waiting_time / len(processes)

        # Print the total and average waiting time
        print("Total waiting time:", total_waiting_time)
        print("Average waiting time:", average_waiting_time)

```

```

quantum = 2
round_robin(processes, quantum)
Executing process 4 for 2 units of time
Executing process 2 for 2 units of time
Executing process 3 for 2 units of time
Executing process 1 for 2 units of time
Executing process 4 for 1 units of time
Process 4 completed at time 9
Waiting time for process 4 is -1
Turnaround time for process 4 is 6
Completion time for process 4 is 9
Executing process 2 for 2 units of time
Executing process 3 for 2 units of time
Executing process 1 for 2 units of time
Executing process 2 for 1 units of time
Process 2 completed at time 16
Waiting time for process 2 is 10
Turnaround time for process 2 is 15
Completion time for process 2 is 16
Executing process 3 for 2 units of time
Executing process 1 for 2 units of time
Executing process 3 for 2 units of time
Process 3 completed at time 22
Waiting time for process 3 is 16
Turnaround time for process 3 is 20
Completion time for process 3 is 22
Executing process 1 for 2 units of time
Turnaround time for process 1 is 26
Completion time for process 1 is 26
Total waiting time: 49
Average waiting time: 12.25

```

```

def srtf(processes):
    # Initialize the waiting time, turnaround time, completion time, and time
    variables
    waiting_time = [0] * len(processes)
    turnaround_time = [0] * len(processes)
    completion_time = [0] * len(processes)
    time = 0

    # Create a copy of the processes list to keep track of the remaining burst
    time of each process
    remaining_burst_time = [p.burst_time for p in processes]

    # Execute the processes one by one in Shortest Remaining Time First order
    while True:
        # Find the process with the shortest remaining burst time
        shortest_time = float('inf')
        shortest_index = None
        for i, p in enumerate(processes):
            if remaining_burst_time[i] > 0 and remaining_burst_time[i] <
shortest_time:
                shortest_time = remaining_burst_time[i]
                shortest_index = i

        if shortest_index is None:
            # All processes completed
            break

        # Execute the process with the shortest remaining burst time
        p = processes[shortest_index]
        print("Executing process", p.pid, "for",
remaining_burst_time[shortest_index], "units of time")
        remaining_burst_time[shortest_index] = 0
        time += remaining_burst_time[shortest_index]

        # Calculate turnaround time and completion time for the current
        process
        turnaround_time[shortest_index] = time - p.arrival_time
        completion_time[shortest_index] = time

        # Print the waiting time, turnaround time, and completion time for the
        current process
        print("Process", p.pid, "completed at time",
completion_time[shortest_index])
        print("Waiting time for process", p.pid, "is",
waiting_time[shortest_index])
        print("Turnaround time for process", p.pid, "is",
turnaround_time[shortest_index])

```

```

        print("Completion time for process", p.pid, "is",
completion_time[shortest_index])

        # Update the waiting time for all remaining processes
        for i, p in enumerate(processes):
            if i != shortest_index and remaining_burst_time[i] > 0:
                waiting_time[i] += remaining_burst_time[shortest_index]

        # Calculate the total waiting time and average waiting time for all
processes
        total_waiting_time = sum(waiting_time)
        average_waiting_time = total_waiting_time / len(processes)

        # Print the total and average waiting time
        print("Total waiting time:", total_waiting_time)
        print("Average waiting time:", average_waiting_time)

```

```

srtf(processes)

```

```

Executing process 4 for 3 units of time
Process 4 completed at time 0
Waiting time for process 4 is 0
Turnaround time for process 4 is -3
Completion time for process 4 is 0
Executing process 2 for 5 units of time
Process 2 completed at time 0
Waiting time for process 2 is 0
Turnaround time for process 2 is -1
Completion time for process 2 is 0
Executing process 3 for 8 units of time
Process 3 completed at time 0
Waiting time for process 3 is 0
Turnaround time for process 3 is -2
Completion time for process 3 is 0
Executing process 1 for 10 units of time
Process 1 completed at time 0
Waiting time for process 1 is 0
Turnaround time for process 1 is 0
Completion time for process 1 is 0
Total waiting time: 0
Average waiting time: 0.0

```

```

class Process:
    def __init__(self, pid, arrival_time, burst_time, priority):
        self.pid = pid
        self.arrival_time = arrival_time
        self.burst_time = burst_time
        self.priority = priority
        self.remaining_time = burst_time

    def __lt__(self, other):
        return self.priority < other.priority

def priority_scheduling(processes):
    time = 0
    n = len(processes)
    completed = 0
    waiting_time = 0
    turnaround_time = 0
    response_time = [0] * n
    remaining_time = [p.burst_time for p in processes]

    while completed != n:
        highest_priority_process = None
        for i in range(n):
            if processes[i].arrival_time <= time and remaining_time[i] > 0:
                if highest_priority_process is None or processes[i] <
processes[highest_priority_process]:
                    highest_priority_process = i

        if highest_priority_process is None:
            time += 1
            continue

        response_time[highest_priority_process] = time -
processes[highest_priority_process].arrival_time

        remaining_time[highest_priority_process] -= 1
        time += 1

        if remaining_time[highest_priority_process] == 0:
            completed += 1
            waiting_time += time -
processes[highest_priority_process].arrival_time -
processes[highest_priority_process].burst_time
            turnaround_time += time -
processes[highest_priority_process].arrival_time

    print("Average waiting time =", waiting_time / n)
    print("Average turnaround time =", turnaround_time / n)

```

```
print("Average response time =", sum(response_time) / n)

if __name__ == '__main__':
    processes = [
        Process(1, 0, 7, 2),
        Process(2, 2, 4, 3),
        Process(3, 4, 1, 1),
        Process(4, 5, 4, 4),
        Process(5, 6, 3, 2),
    ]
    priority_scheduling(processes)
```

Average waiting time = 4.4
Average turnaround time = 8.2
Average response time = 7.2

Disk Scheduling Algorithm

```
import matplotlib.pyplot as plt

# FCFS disk scheduling
def fcfs(head, queue):
    total_movement = 0
    for i in range(len(queue)):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    return total_movement

# SSTF disk scheduling
def sstf(head, queue):
    total_movement = 0
    while queue:
        queue.sort(key=lambda x: abs(x-head))
        diff = abs(head - queue[0])
        total_movement += diff
        head = queue[0]
        queue.pop(0)
    return total_movement

# SCAN disk scheduling
def scan(head, queue):
    queue.sort()
    total_movement = 0
    if head not in queue:
        print("Head not in queue.")
        return 0
    for i in range(queue.index(head) + 1):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    total_movement += abs(head - queue[0])
    head = queue[0]
    for i in range(queue.index(head), len(queue)):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    return total_movement
```

```

# C-SCAN disk scheduling
def cscan(head, queue):
    queue.sort()
    total_movement = 0
    if head not in queue:
        print("Head not in queue.")
        return 0
    for i in range(queue.index(head) + 1):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    total_movement += abs(head - queue[0])
    head = queue[0]
    total_movement += abs(head - queue[-1])
    head = queue[-1]
    for i in range(0, queue.index(head)):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    return total_movement

# LOOK disk scheduling
def look(head, queue):
    queue.sort()
    total_movement = 0
    if head not in queue:
        print("Head not in queue.")
        return 0
    for i in range(queue.index(head) + 1):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    for i in range(queue.index(head), len(queue)):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    for i in range(queue.index(head) - 1, -1, -1):
        diff = abs(head - queue[i])
        total_movement += diff
        head = queue[i]
    return total_movement

```



```

# C-LOOK disk scheduling
def clook(head, queue):
    queue.sort()
    total_movement = 0
    for i in range(len(queue)):
        if queue[i] >= head:
            total_movement += abs(head - queue[i])
            head = queue[i]
            break
    for i in range(len(queue)):
        if queue[i] < head:
            total_movement += abs(head - queue[i])
            head = queue[i]
    for i in range(len(queue)-1, -1, -1):
        if queue[i] >= head:
            total_movement += abs(head - queue[i])
            head = queue[i]
    for i in range(len(queue)-2, -1, -1):
        if queue[i] < head:
            total_movement += abs(head - queue[i])
            head = queue[i]
    return total_movement

# Sample disk queue and head position
queue = [86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130]
head = 143

# Calculate total head movement for each algorithm
fcfs_total = fcfs(head, queue)
sstf_total = sstf(head, queue.copy())
scan_total = scan(head, queue.copy())
cscan_total = cscan(head, queue.copy())
look_total = look(head, queue.copy())
clook_total = clook(head, queue.copy())

print("FCFS total head movement:", fcfs_total)
print("SSTF total head movement:", sstf_total)
print("SCAN total head movement:", scan_total)
print("C-SCAN total head movement:", cscan_total)
print("LOOK total head movement:", look_total)
print("C-LOOK total head movement:", clook_total)

plt.bar(["FCFS", "SSTF", "SCAN", "C-SCAN", "LOOK", "C-LOOK"],
        [fcfs_total, sstf_total, scan_total, cscan_total, look_total, clook_total])
plt.title("Disk Scheduling Algorithms")
plt.ylabel("Total Head Movement")
plt.show()

```

Output :

```
Head not in queue.  
Head not in queue.  
Head not in queue.  
Head not in queue.  
FCFS total head movement: 7081  
SSTF total head movement: 1745  
SCAN total head movement: 0  
C-SCAN total head movement: 0  
LOOK total head movement: 0  
C-LOOK total head movement: 0
```

Page Replacement

FIFO Algorithm

```
def fifo_page_replacement(pages, frames):  
    # initialize page fault counter  
    page_faults = 0  
    # initialize list of frames  
    frame_list = []  
    # iterate over pages  
    for page in pages:  
        # if page is not in frames  
        if page not in frame_list:  
            # if there is still space in frames  
            if len(frame_list) < frames:  
                frame_list.append(page)  
            # if there is no space in frames  
            else:  
                # remove the oldest page in frames  
                frame_list.pop(0)  
                # add the new page to frames  
                frame_list.append(page)  
            # increment page fault counter  
            page_faults += 1  
    return page_faults
```

```
pages = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
frames = 3  
page_faults = fifo_page_replacement(pages, frames)  
print("Number of page faults:", page_faults)
```

Number of page faults: 10

Optimal Algorithm

```
def optimal_page_replacement(pages, frames):
    # initialize page fault counter
    page_faults = 0
    # initialize list of frames
    frame_list = []
    # iterate over pages
    for page in pages:
        # if page is not in frames
        if page not in frame_list:
            # if there is still space in frames
            if len(frame_list) < frames:
                frame_list.append(page)
            # if there is no space in frames
            else:
                # initialize dictionary of indices
                indices = {}
                # iterate over frames
                for f in frame_list:
                    # if page is not in the remaining pages
                    if f not in pages[pages.index(page):]:
                        # store the index
                        indices[f] = 1000000
                    # if page is in the remaining pages
                    else:
                        # store the index of its next occurrence
                        indices[f] = pages.index(f, pages.index(page))
                # remove the frame with the maximum index
                frame_list.remove(max(indices, key=indices.get))
                # add the new page to frames
                frame_list.append(page)
            # increment page fault counter
            page_faults += 1
    return page_faults
```

```
pages = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
frames = 3
page_faults = optimal_page_replacement(pages, frames)
print("Number of page faults:", page_faults)
```

Number of page faults: 10

LRU

```
from collections import deque

def lru_page_replacement(pages, frames):
    # initialize page fault counter
    page_faults = 0
    # initialize list of frames
    frame_list = []
    # initialize queue to keep track of the least recently used pages
    page_queue = deque()
    # iterate over pages
    for page in pages:
        # if page is not in frames
        if page not in frame_list:
            # if there is still space in frames
            if len(frame_list) < frames:
                frame_list.append(page)
            # if there is no space in frames
            else:
                # remove the least recently used page from frames
                lru_page = page_queue.popleft()
                frame_list.remove(lru_page)
                # add the new page to frames
                frame_list.append(page)
            # add the new page to the least recently used queue
            page_queue.append(page)
            # increment page fault counter
            page_faults += 1
        # if page is already in frames
        else:
            # remove page from the least recently used queue
            page_queue.remove(page)
            # add page to the end of the least recently used queue
            page_queue.append(page)
    return page_faults
```

```
pages = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
frames = 3
page_faults = lru_page_replacement(pages, frames)
print("Number of page faults:", page_faults)
```

Number of page faults: 10

LFU

```
from collections import defaultdict

def lfu_page_replacement(pages, frames):
    # initialize page fault counter
    page_faults = 0
    # initialize list of frames
    frame_list = []
    # initialize dictionary to keep track of the frequency of pages in frames
    page_freq = defaultdict(int)
    # iterate over pages
    for page in pages:
        # if page is not in frames
        if page not in frame_list:
            # if there is still space in frames
            if len(frame_list) < frames:
                frame_list.append(page)
            # if there is no space in frames
            else:
                # find the least frequently used page in frames
                min_freq = min(page_freq.values())
                lfu_pages = [p for p in frame_list if page_freq[p] ==
min_freq]

                lfu_page = lfu_pages[0]
                for p in lfu_pages:
                    if page_freq[p] < page_freq[lfu_page]:
                        lfu_page = p
                frame_list.remove(lfu_page)
                # add the new page to frames
                frame_list.append(page)
                page_freq.clear()
            # increment frequency of new page
            page_freq[page] += 1
            # increment page fault counter
            page_faults += 1
        # if page is already in frames
        else:
            # increment frequency of page
            page_freq[page] += 1
    return page_faults
```

```
pages = [1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5]
frames = 3
page_faults = lfu_page_replacement(pages, frames)
print("Number of page faults:", page_faults)
```

Number of page faults: 8

Deadlocks and Concurrency

Producer Consumer

```
import threading
import time
import random
from queue import Queue

# Define the shared queue with a maximum size
queue = Queue(maxsize=5)

# Define a function to simulate the producer thread
def producer():
    while True:
        # Generate a random number and add it to the queue
        item = random.randint(1, 10)
        queue.put(item)
        print(f"Producer added {item} to the queue")
        # Sleep for a random amount of time
        time.sleep(random.uniform(0.5, 1.5))

# Define a function to simulate the consumer thread
def consumer():
    while True:
        # Get an item from the queue
        item = queue.get()
        print(f"Consumer removed {item} from the queue")
        # Sleep for a random amount of time
        time.sleep(random.uniform(0.5, 1.5))
        # Indicate that the task is complete
        queue.task_done()

# Create and start the producer and consumer threads
producer_thread = threading.Thread(target=producer)
consumer_thread = threading.Thread(target=consumer)
producer_thread.start()
consumer_thread.start()

# Wait for the threads to complete (which they never will in this example)
producer_thread.join()
consumer_thread.join()
```


Output exceeds the size limit. Open the full output data in a text editor

Producer added 1 to the queue

Consumer removed 1 from the queue

Producer added 9 to the queueConsumer removed 9 from the queue

Producer added 4 to the queue

Consumer removed 4 from the queue

Producer added 2 to the queue

Consumer removed 2 from the queue

Producer added 3 to the queue

Consumer removed 3 from the queue

Producer added 2 to the queueConsumer removed 2 from the queue

Producer added 4 to the queueConsumer removed 4 from the queue

Producer added 5 to the queue

Consumer removed 5 from the queue

Producer added 10 to the queueConsumer removed 10 from the queue

Producer added 4 to the queue

Consumer removed 4 from the queue

Producer added 2 to the queue

Consumer removed 2 from the queue

Producer added 2 to the queue

Consumer removed 2 from the queue

Producer added 2 to the queue

...

Producer added 6 to the queue

Consumer removed 6 from the queue

Producer added 4 to the queue

Producer added 1 to the queue

Banker's Algorithm

```
# Define the available resources and maximum resource needs for each process
available = [3, 3, 2]
maximum = [
    [7, 5, 3],
    [3, 2, 2],
    [9, 0, 2],
    [2, 2, 2],
    [4, 3, 3]
]

# Define the currently allocated resources for each process
allocation = [
    [0, 1, 0],
    [2, 0, 0],
    [3, 0, 2],
    [2, 1, 1],
    [0, 0, 2]
]

# Define the remaining need for each process
need = [
    [7, 4, 3],
    [1, 2, 2],
    [6, 0, 0],
    [0, 1, 1],
    [4, 3, 1]
]

# Define a function to check if a state is safe
def is_safe(available, allocation, need):
    # Create copies of the available, allocation, and need matrices
    work = available.copy()
    finish = [False] * len(allocation)
    allocation = [row.copy() for row in allocation]
    need = [row.copy() for row in need]

    # Iterate through the processes
    while True:
        # Find a process that can be completed
        found = False
        for i in range(len(allocation)):
            if not finish[i] and all(need[i][j] <= work[j] for j in
range(len(available))):
                found = True
                # Release the allocated resources
                for j in range(len(available)):
                    work[j] += allocation[i][j]
```

```

        finish[i] = True
        break
    # If no process can be completed, exit the loop
    if not found:
        break

# If all processes have been completed, the state is safe
return all(finish)

# Define a function to request resources
def request_resources(process, request, available, allocation, need):
    # Check if the request is valid
    if any(request[i] > need[process][i] or request[i] > available[i] for i in
range(len(available))):
        return False

    # Attempt to allocate the resources
    for i in range(len(available)):
        available[i] -= request[i]
        allocation[process][i] += request[i]
        need[process][i] -= request[i]

    # Check if the new state is safe
    if is_safe(available, allocation, need):
        return True
    else:
        # If the new state is not safe, revert the allocation
        for i in range(len(available)):
            available[i] += request[i]
            allocation[process][i] -= request[i]
            need[process][i] += request[i]
        return False

# Test the request_resources function with a sample request
print(f"Initial state: Available = {available}, Allocation = {allocation},
Need = {need}")
if request_resources(1, [1, 0, 2], available, allocation, need):
    print("Request granted")
    print(f"New state: Available = {available}, Allocation = {allocation},
Need = {need}")
else:
    print("Request denied")
    print(f"State unchanged: Available = {available}, Allocation =
{allocation}, Need = {need}")

```

Initial state:

Available = [3, 3, 2],

Allocation = [[0, 1, 0], [2, 0, 0], [3, 0, 2], [2, 1, 1], [0, 0, 2]],

Need = [[7, 4, 3], [1, 2, 2], [6, 0, 0], [0, 1, 1], [4, 3, 1]]

Request granted

New state:

Available = [2, 3, 0],

Allocation = [[0, 1, 0], [3, 0, 2], [3, 0, 2], [2, 1, 1], [0, 0, 2]],

Need = [[7, 4, 3], [0, 2, 0], [6, 0, 0], [0, 1, 1], [4, 3, 1]]