

USP LAB EXAM

LAB 1

Enter these commands at the UNIX prompt, and take snapshot of the output and submit:

Echo hello world

```
preksha@ubuntu:~/Desktop$ echo hello world  
hello world
```

Passwd

```
preksha@ubuntu:~/Desktop$ passwd  
Changing password for preksha.  
Current password:  
New password:  
Retype new password:  
passwd: password updated successfully
```

Date

```
preksha@ubuntu:~/Desktop$ date  
Fri 23 Apr 2021 06:30:29 AM PDT
```

Hostname

```
preksha@ubuntu:~/Desktop$ hostname  
ubuntu
```

Arch

```
preksha@ubuntu:~/Desktop$ arch  
x86_64
```

Uname-a

```
preksha@ubuntu:~/Desktop$ uname -a  
Linux ubuntu 5.4.0-48-generic #52-Ubuntu SMP Thu Sep 10 10:58:49 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

Time sleep

```
preksha@ubuntu:~/Desktop$ time sleep
sleep: missing operand
Try 'sleep --help' for more information.

real    0m0.004s
user    0m0.003s
sys     0m0.000s
```

dmesg | more (you may need to press q to quit)

```
preksha@ubuntu:~/Desktop$ dmesg | more
[...]
0.000000] Linux version 5.4.0-48-generic (buildd@lcy01-amd64-010) (gcc version 9.3.0 (Ubuntu 9.3.0-10ubuntu2)) #52-Ubuntu SMP Thu Sep 10 10:58:49 UTC 2020 (Ubuntu 5.4.0-48.52-generic 5.4.0)
0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-48-generic root=UUID=01890f3a-fe82-4f7b-ae52-0eecd487d76e ro find_preseed=/preseed.cfg auto noskip priority=critical locale=en_US quiet
0.000000] KERNEL supported cpus:
0.000000]   Intel GenuineIntel
0.000000]   AMD AuthenticAMD
0.000000]   Hygon HygonGenUine
0.000000]   Centaur CentaurHells
0.000000]   HiSilicon HiSilicon
0.000000]   Disabled fast string operations
0.000000] x86/fpu Supporting XSAVE feature 0x0001: 'x87 floating point registers'
0.000000] x86/fpu Supporting XSAVE feature 0x0002: 'SSE registers'
0.000000] x86/fpu Supporting XSAVE feature 0x0004: 'AVX registers'
0.000000] x86/fpu Enabled XSAVE feature 0x0001, size 0x7, context size is 832 bytes, using 'compacted' format.
0.000000] x86-prov: physical RAM map:
0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009ffff] usable
0.000000] BIOS-e820: [mem 0x000000000009e800-0x000000000009ffff] reserved
0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000ffff] reserved
0.000000] BIOS-e820: [mem 0x000000000000100000-0x0000000000ffff] reserved
0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000ffff] usable
0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000ffff] ACPi data
0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000ffff] reserved
0.000000] NX (Execute Disable) protection: active
0.000000] SMBIOS present
0.000000] 00: VMware, Inc., VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 07/22/2020
0.000000] vmware: hypercall mode: 0x82
0.000000] Hypervisor detected: VMware
0.000000] vmware: TSC freq read from hypervisor : 1991.998 MHz
0.000000] vmware: Host bus clock speed read from hypervisor : 6660000 Hz
0.000000] vmware: using sched offset of 81526620940 ns
0.000000] tsc: Detected a 1991.998 MHz processor
0.000002] 00:00:00:00:00:00-0x000000000000ffff] usable ==> reserved
0.000564] e820: Reserve [mem 0x0000a000-0x000ffff] usable
0.000569] last_pfn = 0x140000 max_arch_pfn = 0x400000000
0.000620] MTRR default type: uncachable
0.000627] MTRR fixed ranges enabled:
0.000629]   00000-0FFFF write-back
0.000630]   A0000-BFFFF uncachable
0.000631]   C0000-CFFFF write-protect
0.000632]   D0000-DFFFF write-protect
0.000633] MTRR variable ranges enabled:
0.000634]   0 base 000000000000 mask 1FE000000000 write-back
```

Uptime

```
preksha@ubuntu:~/Desktop$ uptime
06:38:37 up 26 min,  1 user,  load average: 0.01, 0.02, 0.14
```

Who am i

```
preksha@ubuntu:~/Desktop$ whoami
preksha
```

Who

```
preksha@ubuntu:~/Desktop$ who
preksha  :0          2021-04-23 06:13 (:0)
```

Id

```
preksha@ubuntu:~/Desktop$ id
uid=1000(preksha) gid=1000(preksha) groups=1000(preksha),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare)
```

Last

```
preksha@ubuntu:~/Desktop$ last
preksha  :0          :0                      Fri Apr 23 06:13  still logged in
reboot   system boot  5.4.0-48-generic  Fri Apr 23 06:12  still running
preksha  :1          :1                      Tue Apr  6 23:54 - down  (00:38)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:44 - 00:32  (00:48)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:42 - 23:43  (00:00)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:41 - 23:42  (00:00)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:40 - 23:40  (00:00)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:37 - 23:39  (00:01)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:36 - 23:37  (00:01)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:34 - 23:35  (00:00)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:29 - 23:35  (00:06)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:23 - 23:24  (00:00)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:22 - 23:22  (00:00)
reboot   system boot  5.4.0-48-generic  Tue Apr  6 23:07 - 23:19  (00:11)
preksha  :0          :0                      Fri Sep 25 01:43 - down  (14+00:26)
reboot   system boot  5.4.0-48-generic  Fri Sep 25 01:42 - 02:10 (14+00:28)
preksha  :0          :0                      Wed Sep 23 04:30 - down  (00:21)
reboot   system boot  5.4.0-48-generic  Wed Sep 23 04:27 - 04:51 (00:24)

wtmp begins Wed Sep 23 04:27:27 2020
```

Finger

```
preksha@ubuntu:~/Desktop$ finger
Login      Name      Tty      Idle  Login Time  Office      Office Phone
preksha    Preksha Sannecy  *:0           Apr 23 06:13 (:0)
```

W

```
preksha@ubuntu:~/Desktop$ w
06:46:48 up 34 min, 1 user, load average: 0.04, 0.02, 0.08
USER     TTY      FROM      LOGIN@  IDLE    JCPU   PCPU WHAT
preksha  :0      :0      06:13 ?xdm?  1:51  0.06s /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
preksha@ubuntu:~/Desktop$
```

Echo

```
preksha@ubuntu:~/Desktop$ echo
```

Man "automatic door"

```
preksha@ubuntu:~/Desktop$ man "automatic door"
No manual entry for automatic door
```

Lost

```
preksha@ubuntu:~/Desktop$ lost
Command 'lost' not found, did you mean:

  command 'gost' from snap gost (2.11.1)
  command 'gost' from deb gost (0.1.0+git20181204.5afeda5e-1)
  command 'host' from deb bind9-host (1:9.16.1-0ubuntu2.3)
  command 'last' from deb util-linux (2.34-0.1ubuntu9.1)
  command 'most' from deb most (5.0.0a-4)
  command 'lout' from deb lout (3.39-3)

See 'snap info <snapname>' for additional versions.

preksha@ubuntu:~/Desktop$ █
```

Top

```
preksha@ubuntu:~/Desktop$ top
      Tasks: 279 total, 2 running, 277 sleeping, 0 stopped, 0 zombie
 %Cpu(s): 0.2 us, 0.2 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 sl, 0.0 st
 Mem: 3908.4 total, 2151.4 free, 921.9 used, 835.1 buff/cache
 Swap: 2048.0 total, 0.0 free, 0.0 used, 2741.9 avail Mem

 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 66 root 20 0 166680 7256 6320 R 0.3 0.2 0:06.13 vntoolsd
 68 avahi 20 0 8500 3476 3148 S 0.3 0.1 0:00.38 avahi-daemon
 142 preksha 20 0 270364 58068 33888 S 0.3 1.5 0:14.19 Xorg
 165 preksha 20 0 3789328 230132 96360 S 0.3 5.0 0:21.35 gnome-shell
 1798 preksha 20 0 292232 41924 30460 S 0.3 1.8 0:06.60 vntoolsd
 2111 preksha 20 0 963660 53408 39332 S 0.3 1.3 0:14.74 gnome-terminal-
 1 root 20 0 103292 12960 8464 S 0.0 0.3 0:09.67 systemd
 2 root 20 0 0 0 0 S 0.0 0.0 0:00.04 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
 6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd
 9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq
 10 root 20 0 0 0 0 S 0.0 0.0 0:00.25 ksoftirqd/0
 11 root 20 0 0 0 0 I 0.0 0.0 0:00.96 rcu_sched
 12 root rt 0 0 0 0 S 0.0 0.0 0:00.07 migration/0
 13 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0
 14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
 15 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
 16 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/1
 17 root rt 0 0 0 0 S 0.0 0.0 0:01.05 migration/1
 18 root 20 0 0 0 0 S 0.0 0.0 0:00.48 ksoftirqd/1
 20 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/1:0H-kblockd
 21 root 20 0 0 0 0 S 0.0 0.0 0:00.01 kdevtmpfs
 22 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns
 23 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rCU_tasks_kthre
 24 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kauditd
 26 root 20 0 0 0 0 S 0.0 0.0 0:00.00 khungtaskd
 27 root 20 0 0 0 0 S 0.0 0.0 0:00.00 oom_reaper
 28 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback
 29 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kcompactd0
 30 root 25 5 0 0 0 S 0.0 0.0 0:00.00 ksmd
 31 root 39 19 0 0 0 S 0.0 0.0 0:00.00 khugepaged
 78 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kIntegrityd
 79 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd
 80 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 blkcg_punt_bio
 82 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 tpm_dev_wq
 83 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 ata_sff
 84 root 20 0 0 0 0 I 0.0 0.0 0:00.00 md
 85 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 edac-poller
 86 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 devfreq_wq
 87 root rt 0 0 0 0 S 0.0 0.0 0:00.00 watchdog
 90 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kswapd0

 84 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 md
 85 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 edac-poller
 86 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 devfreq_wq
 87 root rt 0 0 0 0 S 0.0 0.0 0:00.00 watchdog
 90 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kswapd0

preksha@ubuntu:~/Desktop$ █
```

Clear



Cal 2000

```

preksha@ubuntu:~/Desktop$ cal 2000
          2000
      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           1   2   3   4   5           1   2   3   4
 2   3   4   5   6   7   8   6   7   8   9   10  11  12   5   6   7   8   9   10  11
 9  10  11  12  13  14  15  13  14  15  16  17  18  19  12  13  14  15  16  17  18
16  17  18  19  20  21  22  20  21  22  23  24  25  26  19  20  21  22  23  24  25
23 24 25 26 27 28 29  27 28 29               26 27 28 29 30 31
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           1   2   3   4           1   2
 1   2   3   4   5   6   7   5   6   7   8   9   10  11   3   4   5   6   7   8   9
 8  9  10  11  12  13  14  12  13  14  15  16  17  18  10  11  12  13  14  15  16
15 16 17 18 19 20 21  19 20 21 22 23 24 25  17 18 19 20 21 22 23
22 23 24 25 26 27 28  26 27 28 29 30               24 25 26 27 28 29 30
29 30 31               31

```

Cal 9 1752

```

preksha@ubuntu:~/Desktop$ cal 9 1752
September 1752
Su Mo Tu We Th Fr Sa
      1   2   14  15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30

```

bc -l (type quit or press Ctrl-d to quit)

```

preksha@ubuntu:~/Desktop$ bc -l
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
q
0
^C
(interrupt) use quit to exit.
quit
preksha@ubuntu:~/Desktop$ 

```

```
echo 5+4 | bc -l
```

```
preksha@ubuntu:~/Desktop$ echo 5+4 | bc -l  
9  
preksha@ubuntu:~/Desktop$
```

Yes please

History

```
preksha@ubuntu:~/Desktop$ history
1 ls
2 ls -l
3 pwd
4 man
5 man ls
6 ls
7 ls -c
8 man date
9 date
10 man date
11 cal
12 -m
13 cal -h
14 clear
15 pwd
16 ls
17 ls -s
18 clear
19 history
20 clear
21 man
22 man man
23 who
24 man who
25 who -a
26 who -b
27 who -d
28 bc
29 du
30 df
31 whoami
32 who
33 ping www.google.com
34 clear
35 ping
36 bc
37 b
38 q
39 du
40 who
41 clear
42 ping www.google.com
43 clear
44 pwd
45 ls
46 mkdir abc
47 ls
48 cd abc
```

```

48 cd abc
49 ls
50 cat
51 cat > f1
52 ls
53 cat > f1
54 cat f1
55 cat > f1
56 cat f1
57 cat > f2
58 cat > f3
59 ls
60 cat f2 f3
61 cat f2 f3 > concatfile
62 ls
63 cat concatfile
64 rmdir abc
65 sudo apt install build-essential dkms linux-headers-$(uname -r)
66 apt-get update
67 --fix-missing
68 ls
69 sudo apt install build-essential dkms linux-headers-$(uname -r)
70 echo hello world
71 passwd
72 date
73 hostname
74 arch
75 uname-a
76 uname - a
77 dmesg|more
78 uname -a
79 uptime
80 who am i
81 dmesg|more
82 who
83 id
84 last
85 finger
86 who am i
87 w
88 top
89 echo
90 whoami
91 man "automatic door"
92 lost
93 clear
94 cal 2000
95 cal 9 1752
96 bc -| q
97 echo 5+4 |bc -| q
98 yes please
99 time sleep
100 history
101 bc -l
102 echo 5+4 | bc -l
103 sudo apt install finger
104 finger
105 man "automatic door"
106 lost
107 history
preksha@ubuntu:~/Desktop$ █

```

LAB 2

Try the following command sequence and submit with snapshot of output

1. cd
2. pwd
3. cd ..
4. pwd
5. ls -al

6. cd ..

7. pwd (what happens now)

8. cd /etc

9. ls -al | more

10. cat passwd

11. cd ~

12. pwd

COMBINED OUTPUT OF THE COMMANDS:

```
preksha@ubuntu:~/Desktop$ cd
preksha@ubuntu:~$ pwd
/home/preksha
preksha@ubuntu:~$ cd ..
preksha@ubuntu:/home$ pwd
/home
preksha@ubuntu:/home$ ls -al
total 12
drwxr-xr-x 3 root root 4096 Sep 23 2020 .
drwxr-xr-x 20 root root 4096 Sep 23 2020 ..
drwxr-xr-x 16 preksha preksha 4096 Apr 23 22:22 preksha
preksha@ubuntu:/home$ cd ..
preksha@ubuntu:/$ pwd
/
preksha@ubuntu:/$ cd /etc
preksha@ubuntu:/etc$ ls -al | more
total 1116
drwxr-xr-x 130 root root 12288 Apr 23 06:29 .
drwxr-xr-x 20 root root 4096 Sep 23 2020 ..
drwxr-xr-x 3 root root 4096 Jul 31 2020 acpi
-rw-r--r-- 1 root root 3028 Jul 31 2020 adduser.conf
drwxr-xr-x 3 root root 4096 Jul 31 2020 alsa
drwxr-xr-x 2 root root 4096 Sep 23 2020 alternatives
-rw-r--r-- 1 root root 401 Jul 16 2019 anacrontab
-rw-r--r-- 1 root root 433 Oct 1 2017 apg.conf
drwxr-xr-x 5 root root 4096 Jul 31 2020 apt
drwxr-xr-x 3 root root 4096 Jul 31 2020 apparmor
drwxr-xr-x 7 root root 4096 Apr 6 23:13 apparmor.d
drwxr-xr-x 4 root root 4096 Oct 9 2020 apport
-rw-r--r-- 1 root root 769 Jan 18 2020 appstream.conf
drwxr-xr-x 7 root root 4096 Sep 23 2020 apt
drwxr-xr-x 3 root root 4096 Jul 31 2020 avahi
-rw-r--r-- 1 root root 2319 Feb 25 2020 bash.bashrc
-rw-r--r-- 1 root root 45 Jan 25 2020 bash_completion
drwxr-xr-x 2 root root 4096 Oct 9 2020 bash_completion.d
-rw-r--r-- 1 root root 367 Apr 14 2020 bindresvport.blacklist
drwxr-xr-x 2 root root 4096 Apr 22 2020 binfmt.d
drwxr-xr-x 2 root root 4096 Jul 31 2020 bluetooth
-rw-r---- 1 root root 33 Jul 31 2020 brlapi.key
drwxr-xr-x 7 root root 4096 Jul 31 2020 brltty
-rw-r--r-- 1 root root 26916 Mar 3 2020 brltty.conf
drwxr-xr-x 3 root root 4096 Jul 31 2020 ca-certificates
-rw-r--r-- 1 root root 5714 Jul 31 2020 ca-certificates.conf
-rw-r--r-- 1 root root 5713 Jul 31 2020 ca-certificates.conf.dpkg-old
drwxr-xr-x 2 root root 4096 Jul 31 2020 calendar
drwxr-s--- 2 root dip 4096 Apr 6 23:12 chatscripts
drwxr-xr-x 2 root root 4096 Sep 25 2020 console-setup
drwxr-xr-x 2 root root 4096 Jul 31 2020 cracklib
drwxr-xr-x 2 root root 4096 Sep 23 2020 cron.d
```

```
drwxr-xr-x  2 root root  4096 Sep 23  2020 cron.d
drwxr-xr-x  2 root root  4096 Oct  9  2020 cron.daily
drwxr-xr-x  2 root root  4096 Jul 31  2020 cron.hourly
drwxr-xr-x  2 root root  4096 Jul 31  2020 cron.monthly
-rw-r--r--  1 root root  1042 Feb 13  2020 crontab
drwxr-xr-x  2 root root  4096 Jul 31  2020 cron.weekly
drwxr-xr-x  5 root lp   4096 Apr 24 08:36 cups
drwxr-xr-x  2 root root  4096 Jul 31  2020 cupshelpers
drwxr-xr-x  4 root root  4096 Jul 31  2020 dbus-1
drwxr-xr-x  4 root root  4096 Jul 31  2020 dconf
-rw-r--r--  1 root root  2969 Aug  3  2019 debconf.conf
-rw-r--r--  1 root root  13 Dec  5  2019 debian_version
drwxr-xr-x  3 root root  4096 Apr  6 23:14 default
-rw-r--r--  1 root root  604 Sep 15  2018 deluser.conf
drwxr-xr-x  2 root root  4096 Jul 31  2020 depmod.d
drwxr-xr-x  4 root root  4096 Jul 31  2020 dhcp
preksha@ubuntu:/etc$ cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
```

```
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:KernelOops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
preksha:x:1000:1000:Preksha Sannecy,,,:/home/preksha:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
```

13. Change back into your home directory.

```
preksha@ubuntu:/etc$ cd ~
preksha@ubuntu:~$ pwd
/home/preksha
preksha@ubuntu:~$ cd ..
preksha@ubuntu:/home$ mkdir work
mkdir: cannot create directory 'work': Permission denied
preksha@ubuntu:/home$ ls
preksha
preksha@ubuntu:/home$ cd
```

14. Make subdirectories called work and play.

```
preksha@ubuntu:/home$ cd
preksha@ubuntu:~$ mkdir work
preksha@ubuntu:~$ mkdir play
preksha@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures play Public Templates Videos work
```

15. Delete the subdirectory called work.

```
preksha@ubuntu:~$ rmdir work
preksha@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures play Public Templates Videos
```

16. Copy file /etc/passwd into your home directory.

```
preksha@ubuntu:~$ cp /etc/passwd "HOME"
preksha@ubuntu:~$ ls
Desktop Documents Downloads HOME Music Pictures play Public Templates Videos
preksha@ubuntu:~$ cp /etc/passwd "$HOME"
preksha@ubuntu:~$ ls
Desktop Documents Downloads HOME Music passwd Pictures play Public Templates Videos
```

17. Move it into the subdirectory play

```
preksha@ubuntu:~$ mv passwd play
preksha@ubuntu:~$ ls
Desktop Documents Downloads HOME Music Pictures play Public Templates Videos
preksha@ubuntu:~$ cd play
preksha@ubuntu:~/play$ ls
passwd
```

LAB 3

1. Create a directory.

```
preksha@ubuntu:~/Desktop$ cd  
preksha@ubuntu:~$ mkdir xyz  
preksha@ubuntu:~$ cd xyz  
preksha@ubuntu:~/xyz$ mkdir subdir  
preksha@ubuntu:~/xyz$ pwd  
/home/preksha/xyz
```

2. Create a subdirectory in the directory created.

```
preksha@ubuntu:~/xyz$ mkdir subdir  
preksha@ubuntu:~/xyz$ pwd  
/home/preksha/xyz  
preksha@ubuntu:~/xyz$ cd subdir  
preksha@ubuntu:~/xyz/subdir$ pwd  
/home/preksha/xyz/subdir
```

3. Change your current directory to the subdirectory.

```
preksha@ubuntu:~$ mkdir dir  
preksha@ubuntu:~$ ls  
Desktop Documents HOME Pictures Public Videos  
dir Downloads Music play Templates xyz  
preksha@ubuntu:~$ mkdir dir/subdir  
preksha@ubuntu:~$ ls  
Desktop Documents HOME Pictures Public Videos  
dir Downloads Music play Templates xyz  
preksha@ubuntu:~$ cd dir/subdir  
preksha@ubuntu:~/dir/subdir$ cd ..
```

4. Display the calendar of the month.

```
preksha@ubuntu:~/xyz$ cal 4 2021  
April 2021  
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
```

5. Get a directory listing the parent directory.

```
preksha@ubuntu:~/dir$ ls  
subdir
```

6. How many users were logged onto your device.

```
preksha@ubuntu:~/xyz$ who  
preksha :0          2021-04-25 10:17 (:0)  
preksha@ubuntu:~/xyz$ who -u  
preksha :0          2021-04-25 10:17      ?           1539 (:0)
```

7. Display your name in the form of a banner.

8. Display the name of device name of your terminal.

```
preksha@ubuntu:~/xyz$ tty  
/dev/pts/0
```

9. Move to the root directory.

```
preksha@ubuntu:~/dir/subdir$ cd ..
preksha@ubuntu:~/dir$ cd ..
preksha@ubuntu:~$ cd ..
preksha@ubuntu:/home$ cd ..
```

LAB 4

1. List all the processes that are presently running.

```
preksnaguuntu:~/Desktop$ ps  
 PID TTY      TIME CMD  
 3282 pts/0    00:00:00 bash  
 3288 pts/0    00:00:00 ps
```

2. List the text files in your current directory.

```
preksha@ubuntu:~/Desktop$ mkdir ex.txt
preksha@ubuntu:~/Desktop$ ls *.txt
preksha@ubuntu:~/Desktop$ ls
a abc b c concatenatefile ex.txt f1 f2 f3
preksha@ubuntu:~/Desktop$ mkdir ex1.txt
preksha@ubuntu:~/Desktop$ ls *.txt
ex1.txt:
ex.txt:
```

3. Make a copy of any text file.

```
preksha@ubuntu:~/Desktop$ cp -r ex.txt ex2.txt  
preksha@ubuntu:~/Desktop$ cd ex2.txt  
preksha@ubuntu:~/Desktop/ex2.txt$ ls
```

4. Rename one of your text files in the current directory.

```
preksha@ubuntu:~/Desktop$ mv ex2.txt ex3.txt  
preksha@ubuntu:~/Desktop$ ls  
a abc b c concatfile ex1.txt ex3.txt ex.txt f1 f2 f3
```

5. Delete an unneeded copy of a file.

```
preksha@ubuntu:~/Desktop$ rmdir ex3.txt  
preksha@ubuntu:~/Desktop$ ls  
a abc b c concatfile ex1.txt ex.txt f1 f2 f3
```

LAB 5:

Set permissions on all of your files and directories to those that you want. You may want to give read permission on some of your files and directories to members of your group.

```
preksha@ubuntu:~/Desktop$ ls -l
total 48
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 a
drwxrwxr-x 2 preksha preksha 4096 Sep 25 2020 abc
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 c
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
drwxrwxr-x 2 preksha preksha 4096 Apr 27 07:38 ex1.txt
drwxrwxr-x 2 preksha preksha 4096 Apr 27 23:21 ex.c
drwxrwxr-x 2 preksha preksha 4096 Apr 27 07:38 ex.txt
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 44 Apr 27 23:38 user.txt
preksha@ubuntu:~/Desktop$ chmod g=rx abc
preksha@ubuntu:~/Desktop$ ls -l
total 48
drwxr-xr-x 2 preksha preksha 4096 Apr 27 00:22 a
drwxr-xr-x 2 preksha preksha 4096 Sep 25 2020 abc
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 c
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
drwxrwxr-x 2 preksha preksha 4096 Apr 27 07:38 ex1.txt
drwxrwxr-x 2 preksha preksha 4096 Apr 27 23:21 ex.c
drwxrwxr-x 2 preksha preksha 4096 Apr 27 07:38 ex.txt
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 44 Apr 27 23:38 user.txt
preksha@ubuntu:~/Desktop$
```

Create a number of hierarchically related directories and navigate through them using a combination of absolute path names (starting with "/") and relative pathnames.

```
preksha@ubuntu:~/Desktop$ cd bollywood/movie/comedy
preksha@ubuntu:~/Desktop/bollywood/movie/comedy$
```

Try using wildcards ("*" and possibly "?").

```
preksha@ubuntu:~/Desktop$ ls -l *
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
preksha@ubuntu:~/Desktop$ ls concat?file
concatfile
preksha@ubuntu:~/Desktop$
```

Put a listing of the files in your directory into a file called file list. (Then delete it!)

```
preksha@ubuntu:~/Desktop$ ls -l *e
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatefile
preksha@ubuntu:~/Desktop$ cat *.txt > merged-file
cat: ex1.txt: Is a directory
cat: ex.txt: Is a directory
preksha@ubuntu:~/Desktop$ ls
a abc b c concatefile ex1.txt ex.c ex.txt f1 f2 f3 merged-file user.txt
preksha@ubuntu:~/Desktop$ cat merged-file
With death being the only certainty of life, life is just a bit too short for regrets.
Let's live while we're alive.
Let's love while we're alive.
Let's live and not just survive.
preksha@ubuntu:~/Desktop$ rm merged-file
preksha@ubuntu:~/Desktop$ ls
a abc b c concatefile ex1.txt ex.c ex.txt f1 f2 f3 user.txt
preksha@ubuntu:~/Desktop$
```

LAB 6:

Type the command `ls -l` and examine the format of the output. Pipe the output of the command `ls -l` to the word count `wc` to obtain a count of the number of files in your directory.

```
preksha@ubuntu:~/Desktop$ ls -l
total 52
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 a
drwxr-xr-x 2 preksha preksha 4096 Sep 25 2020 abc
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
drwxrwxr-x 3 preksha preksha 4096 Jun 12 09:06 bollywood
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 c
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatefile
drwxrwxr-x 2 preksha preksha 4096 Apr 27 07:38 ex1.txt
drwxrwxr-x 2 preksha preksha 4096 Apr 27 23:21 ex.c
drwxrwxr-x 2 preksha preksha 4096 Apr 27 07:38 ex.txt
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 180 Jun 11 23:04 user.txt
preksha@ubuntu:~/Desktop$ ls -l | wc
      14      119     690
```

Use `cut` to strip away the reference material and leave just the text field.

```
preksha@ubuntu:~/Desktop$ cut user.txt -d ':' -f 1,3
With death being the only certainty of life, life is just a bit too short for regrets.
Let's live while we're alive.
Let's love while we're alive.
Let's live and not just survive.
preksha@ubuntu:~/Desktop$ cut user.txt -f 1,20 --complement
With death being the only certainty of life, life is just a bit too short for regrets.
Let's live while we're alive.
Let's love while we're alive.
Let's live and not just survive.
preksha@ubuntu:~/Desktop$ echo 'preksha' | cut -b 6
h
preksha@ubuntu:~/Desktop$
```

```
preksha@ubuntu:~/Desktop$ echo 'hello how are you doing' | cut -d ' ' -f 1,2
hello how
preksha@ubuntu:~/Desktop$ echo 'hello how are you doing' | cut -d ' ' -f 1
hello
preksha@ubuntu:~/Desktop$
```

Use tr to strip away any tags that are actually in the text (e.g., attached to the words), so that you are left with just the words.

```
Nature's first green is gold
preksha@ubuntu:~/Desktop$ echo "Nature's first green is gold" | tr " " "\n"
Nature's
first
green
is
gold
preksha@ubuntu:~/Desktop$
```

Set a file to be read-only with the chmod (from change mode) command. Interpret the file permissions displayed by the ls –l command.

```
preksha@ubuntu:~/Desktop$ chmod 400 ex.txt
preksha@ubuntu:~/Desktop$ ls -l
total 52
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 a
drwxr-xr-x 2 preksha preksha 4096 Sep 25 2020 abc
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
drwxrwxr-x 3 preksha preksha 4096 Jun 12 09:06 bollywood
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 c
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatefile
drwxrwxr-x 2 preksha preksha 4096 Apr 27 07:38 ex1.txt
drwxrwxr-x 2 preksha preksha 4096 Apr 27 23:21 ex.c
dr----- 2 preksha preksha 4096 Apr 27 07:38 ex.txt
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 180 Jun 11 23:04 user.txt
preksha@ubuntu:~/Desktop$
```

Delete one or more directories with the rmdir (from remove directory) command. See what happens if the directory is not empty. Experiment (carefully!) with the rm -r command to delete a directory and its content.

```
preksha@ubuntu:~/Desktop$ ls
a abc b bollywood c concatefile ex.txt f1 f2 f3 test.c text.cpp text.txt user.txt
preksha@ubuntu:~/Desktop$ rmdir text.txt
preksha@ubuntu:~/Desktop$ ls
a abc b bollywood c concatefile ex.txt f1 f2 f3 test.c text.cpp user.txt
preksha@ubuntu:~/Desktop$ rmdir text.cpp
preksha@ubuntu:~/Desktop$ rm -r test.c
preksha@ubuntu:~/Desktop$ ls
a abc b bollywood c concatefile ex.txt f1 f2 f3 user.txt
preksha@ubuntu:~/Desktop$
```

Experiment with redirecting command output (e.g., ls -l >file1). Try ">>" instead of ">"

with an existing text file as the output.

```
preksha@ubuntu:~/Desktop$ touch baba.txt
preksha@ubuntu:~/Desktop$ cat baba.txt
preksha@ubuntu:~/Desktop$ cat > baba.txt
ba balck sheep,
have you any wool,
yes sir yes sir,
three bags full...
^Z
[1]+  Stopped                  cat > baba.txt
preksha@ubuntu:~/Desktop$ echo "this is a poem" > baba.txt
preksha@ubuntu:~/Desktop$ cat baba.txt
this is a poem
preksha@ubuntu:~/Desktop$ echo "first new line" > baba.txt
preksha@ubuntu:~/Desktop$ cat baba.txt
first new line
preksha@ubuntu:~/Desktop$ echo "second new line" > baba.txt
preksha@ubuntu:~/Desktop$ cat baba.txt
second new line
preksha@ubuntu:~/Desktop$ echo "third new line" >> baba.txt
preksha@ubuntu:~/Desktop$ cat baba.txt
second new line
third new line
preksha@ubuntu:~/Desktop$ ls -l > baba.txt
preksha@ubuntu:~/Desktop$ cat baba.txt
total 64
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 a
drwxr-xr-x 2 preksha preksha 4096 Sep 25 2020 abc
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
-rw-rw-r-- 1 preksha preksha 0 Jun 12 10:45 baba.txt
drwxrwxr-x 3 preksha preksha 4096 Jun 12 09:06 bollywood
drwxrwxr-x 2 preksha preksha 4096 Jun 12 09:34 c
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
dr----- 2 preksha preksha 4096 Apr 27 07:38 ex.txt
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 3 Jun 12 10:37 poem1
drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:32 poem1.txt
-rw-rw-r-- 1 preksha preksha 87 Jun 12 10:36 poem2
drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:33 poem2.txt
-rw-rw-r-- 1 preksha preksha 112 Jun 12 10:41 twinkle.txt
-rw-rw-r-- 1 preksha preksha 180 Jun 11 23:04 user.txt
total 68
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 a
drwxr-xr-x 2 preksha preksha 4096 Sep 25 2020 abc
drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
-rw-rw-r-- 1 preksha preksha 918 Jun 12 10:45 baba.txt
drwxrwxr-x 3 preksha preksha 4096 Jun 12 09:06 bollywood
drwxrwxr-x 2 preksha preksha 4096 Jun 12 09:34 c
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
dr----- 2 preksha preksha 4096 Apr 27 07:38 ex.txt
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 3 Jun 12 10:37 poem1
drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:32 poem1.txt
-rw-rw-r-- 1 preksha preksha 87 Jun 12 10:36 poem2
drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:33 poem2.txt
-rw-rw-r-- 1 preksha preksha 112 Jun 12 10:41 twinkle.txt
-rw-rw-r-- 1 preksha preksha 180 Jun 11 23:04 user.txt
```

See whether upper-case versions of any of these commands work as well as the lower-case versions.

```
preksha@ubuntu:~/Desktop$ LS -L
LS: command not found
preksha@ubuntu:~/Desktop$ CAT baba.txt
CAT: command not found
preksha@ubuntu:~/Desktop$ Cat baba.txt

Command 'Cat' not found, did you mean:

  command 'cat' from deb coreutils (8.30-3ubuntu2)
  command 'pat' from deb dist (1:3.5-236-0.1build1)
  command 'bat' from deb bacula-console-qt (9.4.2-2ubuntu5)
  command 'dat' from deb liballegro4-dev (2:4.4.3.1-1)
  command 'iat' from deb iat (0.1.3-7build1)
  command 'at' from deb at (3.1.23-1ubuntu1)

Try: sudo apt install <deb name>

preksha@ubuntu:~/Desktop$ Echo "second line" >> baba.txt

Command 'Echo' not found, did you mean:

  command 'echo' from deb coreutils (8.30-3ubuntu2)

Try: sudo apt install <deb name>

preksha@ubuntu:~/Desktop$ MKdir hi.txt
MKdir: command not found
preksha@ubuntu:~/Desktop$ Rmdir hi.txt

Command 'Rmdir' not found, did you mean:

  command 'mdir' from deb mtools (4.0.24-1)
  command 'rmdir' from deb coreutils (8.30-3ubuntu2)
  command 'mmdir' from deb simh (3.8.1-6)

Try: sudo apt install <deb name>

preksha@ubuntu:~/Desktop$ Chmod 444 baba.txt

Command 'Chmod' not found, did you mean:

  command 'chmod' from deb coreutils (8.30-3ubuntu2)

Try: sudo apt install <deb name>
```

Use the who command to see users logged into the system. Pipe the output of the who command to the sort command. Search for your login name in who file using the grep command.

```
preksha@ubuntu:~/Desktop$ who
preksha :0          2021-06-11 21:47 (:0)
preksha@ubuntu:~/Desktop$ who | sort
preksha :0          2021-06-11 21:47 (:0)
preksha@ubuntu:~/Desktop$ who | grep preksha
preksha :0          2021-06-11 21:47 (:0)
preksha@ubuntu:~/Desktop$
```

LAB 7:

Compare two text files with the diff command.

```
preksha@ubuntu:~/Desktop$ diff baba.txt baba1.txt
1,36d0
< total 64
< drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 a
< drwxr-xr-x 2 preksha preksha 4096 Sep 25 2020 abc
< drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
< -rw-rw-r-- 1 preksha preksha 0 Jun 12 10:45 baba.txt
< drwxrwxr-x 3 preksha preksha 4096 Jun 12 09:06 bollywood
< drwxrwxr-x 2 preksha preksha 4096 Jun 12 09:34 c
< -rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
< dr----- 2 preksha preksha 4096 Apr 27 07:38 ex.txt
< -rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
< -rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
< -rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
< -rw-rw-r-- 1 preksha preksha 3 Jun 12 10:37 poem1
< drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:32 poem1.txt
< -rw-rw-r-- 1 preksha preksha 87 Jun 12 10:36 poem2
< drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:33 poem2.txt
< -rw-rw-r-- 1 preksha preksha 112 Jun 12 10:41 twinkle.txt
< -rw-rw-r-- 1 preksha preksha 180 Jun 11 23:04 user.txt
< total 68
< drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 a
< drwxr-xr-x 2 preksha preksha 4096 Sep 25 2020 abc
< drwxrwxr-x 2 preksha preksha 4096 Apr 27 00:22 b
< -rw-rw-r-- 1 preksha preksha 918 Jun 12 10:45 baba.txt
< drwxrwxr-x 3 preksha preksha 4096 Jun 12 09:06 bollywood
< drwxrwxr-x 2 preksha preksha 4096 Jun 12 09:34 c
< -rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
< dr----- 2 preksha preksha 4096 Apr 27 07:38 ex.txt
< -rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
< -rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
< -rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
< -rw-rw-r-- 1 preksha preksha 3 Jun 12 10:37 poem1
< drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:32 poem1.txt
< -rw-rw-r-- 1 preksha preksha 87 Jun 12 10:36 poem2
< drwxrwxr-x 2 preksha preksha 4096 Jun 12 10:33 poem2.txt
< -rw-rw-r-- 1 preksha preksha 112 Jun 12 10:41 twinkle.txt
< -rw-rw-r-- 1 preksha preksha 180 Jun 11 23:04 user.txt
preksha@ubuntu:~/Desktop$ █
```

Count lines, words, and characters in a file with the wc command.

```
preksha@ubuntu:~/Desktop$ wc baba.txt
 36 310 1836 baba.txt
preksha@ubuntu:~/Desktop$ wc baba1.txt
0 0 0 baba1.txt
preksha@ubuntu:~/Desktop$ wc -l baba.txt
36 baba.txt
preksha@ubuntu:~/Desktop$ wc -w baba.txt
310 baba.txt
preksha@ubuntu:~/Desktop$ wc -c baba.txt
1836 baba.txt
preksha@ubuntu:~/Desktop$
```

Display your current environment variables with the following command: `set` or `env`.

```
brahakarubuntu:~/Desktop$ env  
SHELL=/bin/bash  
SESSION_MANAGER=local/ubuntu:@tmp/.ICE-unix/ubuntu:/tmp/.ICE-unix/1636  
QT_ACCESSIBILITY=1  
COLORTERM=truecolor  
XDG_CURRENT_DESKTOP=xdg-ubuntu/xdg  
XDG_MENU_PREFIX=x-  
GNOME_DESKTOP_SESSION_ID=this-is-deprecated  
GTK_IM_MODULE=ibus  
QT4_IM_MODULE=ibus  
GNOME_SHELL_SESSION_MODE=ubuntu  
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh  
DISPLAY=:0  
DESKTOP_SESSION=ubuntu  
SSH_AGENT_PID=1538  
QT_QPA_PLATFORM=gallium-bridge  
PWD=/home/preksha/Desktop  
LOGNAME=preksha  
XDG_RUNTIME_DIR=/tmp/.X-1000-OpJ-ubuntu  
XDG_SESSION_TYPE=x11  
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1  
XAUTHORITY=/run/user/1000/gdm/Xauthority  
GJS_DEBUG_TOPICS=JS_ERROR;JS_LOG  
WINDOWPATH=2  
HOME=/home/preksha  
XDG_CONFIG_DIRS=/etc/xdg:  
TK_CONFIG_PHASE=1  
LANG=en_US.UTF-8  
XDG_CURRENT_DESKTOP=GNOME  
VTE_VERSION=6003  
XDG_SESSION_TYPE=x-  
XDG_CURRENT_DESKTOP=x-  
INVOCATION_ID=38f19657883ab49e021f9ccf3410a72  
MANAGERPID=1360  
CLUTTER_IM_MODULE=ibus  
GJS_DEBUG_OUTPUT=stderr  
LESSCLOSE=/usr/bin/lesspipe %s %s  
XDG_SESSION_CLASS=user  
TERM=xterm-256color  
LESSOPEN=| /usr/bin/lesspipe - %s
```

```
BASH_TCPSIZE=complete_fullquote:expand_aliases:extglob:extquote:force_fignore:globasclranges:hstappended:interactive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=(0)
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION_VERSINFO=(0)=2 [1]=10"
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=(0)=5 [1]=0 [2]=17 [3]=1 [4]=release [5]=x86_64-pc-linux-gnu")
BASH_VERSION='5.0.17(1)-release'
CLUTTER_IM_MODULE=ibus
COLORTERM=truecolor
DASH_ECHO=0
DISPLAY=:0
DRTBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
DESKTOP_SESSION=ubuntu
DIRSTACK=()
DISPLAY=:0
EUID=1000
GDMSESSION=ubuntu
GJS_DEBUG_OUTPUT=stderr
GJS_DESKTOP_TOPIC_ERROR_LOG='NONE' DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SHELL_SESSION_MODE=ubuntu
GNOME_TERMINAL_SCREEN=/org/gnome/terminal/screen/645a56e8_d4d6_4446_8769_8babee662ed2
GNOME_TERMINAL_SERVICE=:1.156
GPG_AGENT_INFO=/run/user/1006/gnupg/S.gpg-agent:0:1
GROUPS=()
GTK_IM_MODULE=ibus
GTK_MODULES=gailatk-bridge
HISTCONTROL=ignoreboth
HISTFILE=/home/preksha/.bash_history
HISTFILESIZE=2000
HISTSIZE=1000
HISTTIMEFORMAT=%F %R
HOSTNAME=preksha
HOSTTYPE=ubuntu
HOSTTYPE=x86_64
IFS=$' \t\n'
IM_CONFIG_PHASE=1
INCLUDE_GNOME=false
INPUT_SOURCE=8fb7883ab496ea21f9fcf3416a72
JOBS=1
JOURNAL_STREAM=8:150941
LANG=en_US.UTF-8
LESSCLOSE=/usr/bin/lesspipe %s %s'
LESSOPEN '| /usr/bin/lesspipe %s'
LINES=49
LOGNAME=preksha
```

```

vars=$(_compgen -A variable -P '$' -- "${BASH_REMATCH[3]}");
if [[ ${#vars[@]} -eq 1 && -n $arrs ]]; then
    compopt -o nospace;
    COMPREPLY+=("${arrs[*]}");
else
    COMPREPLY+=("${vars[*]}");
fi;
else
    COMPREPLY=(${_compgen -A variable -P '$' -- "${BASH_REMATCH[3]}"});
    ft;
    return 0;
else
    if [[ $cur =~ ^(\${![#]}?[A-Za-z0-9_-]*\{[^ ]*\})$ ]]; then
        local IFS=
    COMPREPLY=(${_compgen -W "$printf %s\n${!'\${BASH_REMATCH[2]}'[@]}'"
    if [[ ${#BASH_REMATCH[3]} == [0*] ]]; then
        COMPREPLY+=("${BASH_REMATCH[1]}${BASH_REMATCH[2]}${BASH_REMATCH[3]}");
    fi;
    _trim_colon_completions "$cur";
    return 0;
else
    if [[ $cur =~ ^\${![#]}?[A-Za-z0-9_-]*\{.*\}$ ]]; then
        COMPREPLY+=("$cur");
        _trim_colon_completions "$cur";
        return 0;
    else
        case ${prev} in
            T2)
                cur=/usr/share/zoneinfo/$cur;
                _filedir;
                for i in "${!COMPREPLY[@]}";
                do
                    if [[ ${COMPREPLY[i]} == *.tab ]]; then
                        unset 'COMPREPLY[i]';
                        continue;
                    else
                        if [[ -d ${COMPREPLY[i]} ]]; then
                            COMPREPLY[i]+=('/');
                            compopt -o nospace;
                        fi;
                        COMPREPLY[i]=$(COMPREPLY[i]#/usr/share/zoneinfo);
                    done;
                    return 0
                ;;
            TERM)
                _terms;
                return 0
            ;;
        esac
    fi;
    _terms;
    return 0
fi;
;;
TERM)
    _terms;
    return 0
;;

```

Concatenate all files in a directory redirected to /dev/null and redirecting standard error to "errorFile"?

```
preksha@ubuntu:~/Desktop$ cat *.txt >/dev/null
cat: ex.txt: Is a directory
cat: poem1.txt: Is a directory
cat: poem2.txt: Is a directory
preksha@ubuntu:~/Desktop$ baba.txt 2>errorfile
preksha@ubuntu:~/Desktop$
```

Display information on yourself or another user with the finger command.

```
preksha@ubuntu:~/Desktop$ finger preksha
Login: preksha                                Name: Preksha Sannecy
Directory: /home/preksha                         Shell: /bin/bash
On since Fri Jun 11 21:47 (PDT) on :0 from :0 (messages off)
No mail.
No Plan.
preksha@ubuntu:~/Desktop$
```

If you wish, experiment with sending and receiving mail using the pine email program.
Delete all the files in the current directory whose name ends in ".bak".

```
preksha@ubuntu:~/Desktop$ mkdir abak
preksha@ubuntu:~/Desktop$ touch bbak
preksha@ubuntu:~/Desktop$ ls
a      b      bbak      concatfile  f1  poem1      poem2.txt
abak  baba1.txt  bollywood  errorfile  f2  poem1.txt  twinkle.txt
abc   baba.txt   c      ex.txt     f3  poem2      user.txt
preksha@ubuntu:~/Desktop$ rm -r *bak
preksha@ubuntu:~/Desktop$ ls
a      b      baba.txt  c      errorfile  f1  f3  poem1.txt  poem2.txt  user.txt
abc  baba1.txt  bollywood  concatfile  ex.txt  f2  poem1  poem2  twinkle.txt
preksha@ubuntu:~/Desktop$
```

Display lines 10 to 14 of any file which contains 25 lines.

```
preksha@ubuntu:~/Desktop$ touch number
preksha@ubuntu:~/Desktop$ cat >number
1
2
3
4
5
6
7
8
9
10
11
1
12
13
14
15
16
17
18
19
20
21
22
23
24
^Z
[2]+  Stopped                  cat > number
preksha@ubuntu:~/Desktop$
```

```
preksha@ubuntu:~/Desktop$ head -n 14 number | tail +10
10
11
1
12
13
preksha@ubuntu:~/Desktop$
```

Count how many lines contain the word science in a word file science.txt.

```
preksha@ubuntu:~/Desktop$ touch science.txt
preksha@ubuntu:~/Desktop$ cat > science.txt
science
science
science
^Z
[3]+  Stopped                  cat > science.txt
preksha@ubuntu:~/Desktop$ grep -c science science.txt
3
```

List the statistics of the largest file (and only the largest file) in the current directory.

```
preksha@ubuntu:~/Desktop$ du
4      ./c
4      ./a
4      ./b
4      ./bollywood/movie/comedy
8      ./bollywood/movie
12     ./bollywood
4      ./ex.txt
20     ./abc
4      ./poem2.txt
4      ./poem1.txt
108    .
preksha@ubuntu:~/Desktop$ du -a | sort -n -r | head -n 1
108    .
preksha@ubuntu:~/Desktop$
```

LAB 8:

Kill any process with the help of the PID and run any process at the background.

```
preksha@ubuntu:~/Desktop$ sudo kill -SIGKILL 8877
[sudo] password for preksha:
```

```
preksha@ubuntu:~/Desktop$ jobs
[1]  Stopped                  cat > twowords.txt
[2]  Stopped                  cat > words.txt
[3]  Stopped                  cat > two.txt
[4]  Stopped                  cat > blank.txt
[5]  Stopped                  cat > p1.txt
[6]- Stopped                  cat > p2.txt
[7]+ Stopped                  cat > humpty.txt
preksha@ubuntu:~/Desktop$
```

Select a text file and double space the lines.

```
preksha@ubuntu:~/Desktop$ cd b
preksha@ubuntu:~/Desktop/b$ touch fruits.txt
preksha@ubuntu:~/Desktop/b$ cat > fruits.txt
apple
mango
orange
cherry
lichi
banana
^Z
[1]+  Stopped                  cat > fruits.txt
preksha@ubuntu:~/Desktop/b$ sed 's/$/\n\n/' fruits.txt
apple

mango

cherry

lichi

banana

preksha@ubuntu:~/Desktop/b$
```

List all the users from /etc/passwd in the alphabetically sorted order.

```
preksha@ubuntu:~/Desktop$ cut -d: -f1 /etc/passwd | sort
_apt
avahi
avahi-autoipd
backup
bin
colord
cups-pk-helper
daemon
dnsmasq
games
gdm
geoclue
gnats
gnome-initial-setup
hplip
irc
kernoops
list
lp
mail
man
messagebus
news
nm-openvpn
nobody
preksha
proxy
pulse
root
rtkit
saned
speech-dispatcher
sync
sys
syslog
systemd-coredump
systemd-network
systemd-resolve
systemd-timesync
tcpdump
tss
usbmux
uucp
uuid
whoopsie
www-data
preksha@ubuntu:~/Desktop$
```

Create a file with duplicate records and delete duplicate records for that file.

```
preksha@ubuntu:~/Desktop$ touch original.txt
preksha@ubuntu:~/Desktop$ cat > original.txt
january
february
march
april
^Z
[2]+  Stopped                  cat > original.txt
preksha@ubuntu:~/Desktop$ touch duplicate.txt
preksha@ubuntu:~/Desktop$ cat > duplicate.txt
january
february
march
april
^Z
[3]+  Stopped                  cat > duplicate.txt
preksha@ubuntu:~/Desktop$ sort duplicate.txt | uniq
april
february
january
march
preksha@ubuntu:~/Desktop$ sort -u duplicate.txt
april
february
january
march
preksha@ubuntu:~/Desktop$
```

Use the grep command to search the file example1 for occurrences of the string

"water".

```
preksha@ubuntu:~/Desktop$ chmod 777 egs.txt
preksha@ubuntu:~/Desktop$ grep water egs.txt
water is pure
water is natural
water is healthy
water is simple
water helps us all
preksha@ubuntu:~/Desktop$
```

Write grep commands to do the following activities:

To select the lines from a file that have exactly two characters.

```
preksha@ubuntu:~/Desktop$ touch two.txt
preksha@ubuntu:~/Desktop$ cat > two.txt
go
in
^Z
[3]+  Stopped                  cat > two.txt
preksha@ubuntu:~/Desktop$ grep -c -E "^.{2}$" two.txt
2
preksha@ubuntu:~/Desktop$
```

To select the lines from a file that start with the upper case letter.

```
preksha@ubuntu:~/Desktop$ cat twinkle.txt | grep '^[:upper:]][:lower:]'
Twinkle, twinkle, little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky
```

To select the lines from a file that end with a period.

```
preksha@ubuntu:~/Desktop$ cat baba1.txt
ba ba black sheep,
have u any wool?
yes sir yes sir
Three bags full.
preksha@ubuntu:~/Desktop$ grep '[.]\$' baba1.txt
Three bags full.
```

To select the lines in a file that has one or more blank spaces.

```
preksha@ubuntu:~/Desktop$ touch blank.txt
preksha@ubuntu:~/Desktop$ cat > blank.txt
humpty dumpty sat
on the wall.^Z
[4]+  Stopped                  cat > blank.txt
preksha@ubuntu:~/Desktop$ grep "\s" blank.txt
humpty dumpty sat
```

To select the lines in a file and direct them to another file which has digits as one of

the characters in that line.

```
preksha@ubuntu:~/Desktop$ touch p1.txt
preksha@ubuntu:~/Desktop$ cat > p1.txt
ba ba black sheep,
have you any wool?
yes sir yes sir,
3 bags full.
^Z
[5]+  Stopped                  cat > p1.txt
preksha@ubuntu:~/Desktop$ touch p2.txt
preksha@ubuntu:~/Desktop$ cat > p2.txt
1 2 3 4
lets all say,
hello!!
^Z
[6]+  Stopped                  cat > p2.txt
preksha@ubuntu:~/Desktop$ grep "3" p2.txt > p1.txt
preksha@ubuntu:~/Desktop$ cat p1.txt
1 2 3 4
preksha@ubuntu:~/Desktop$
```

Make a sorted wordlist from the file.

```
preksha@ubuntu:~/Desktop$ touch humpty.txt
preksha@ubuntu:~/Desktop$ cat > humpty.txt
Humpty Dumpty sat on the wall,
Humpty Dumpty had a great fall,
All the kings horses and all the kingsmen,
Couln't put Humpty 2gether again.
^Z
[7]+  Stopped                  cat > humpty.txt
preksha@ubuntu:~/Desktop$ sort humpty.txt
All the kings horses and all the kingsmen,
Couln't put Humpty 2gether again.
Humpty Dumpty had a great fall,
Humpty Dumpty sat on the wall,
preksha@ubuntu:~/Desktop$
```

LAB 9,10,11 SHELL SCRIPTS:

1)Write a shell script that delete all lines containing a specified word

```
1 #!/bin/bash
2 if [ $# -eq 0 ]
3 then
4 echo "Please enter one or more filename as argument "
5 exit
6 fi
7 echo "Enter the word to be searched in files "
8 read word
9 for file in $*
10 do
11 sed "/$word/d" $file | tee tmp
12 mv tmp $file
13 done
14
```

```
preksha@ubuntu:~/Desktop$ cat baba1.txt
ba ba black sheep,
have u any wool?
yes sir yes sir
Three bags full.
preksha@ubuntu:~/Desktop$ cat humpty.txt
Humpty Dumpty sat on the wall,
Humpty Dumpty had a great fall,
All the kings horses and all the kingsmen,
Couln't put Humpty 2gether again.
preksha@ubuntu:~/Desktop$ bash one.sh baba1.txt humpty.txt
Enter the word to be searched in files
horses
one.sh: line 11: tmp: command not found
mv: cannot stat 'tmp': No such file or directory
one.sh: line 11: tmp: command not found
mv: cannot stat 'tmp': No such file or directory
preksha@ubuntu:~/Desktop$ bash one.sh baba1.txt humpty.txt
Enter the word to be searched in files
horses
ba ba black sheep,
have u any wool?
yes sir yes sir
Three bags full.
Humpty Dumpty sat on the wall,
Humpty Dumpty had a great fall,
Couln't put Humpty 2gether again.
preksha@ubuntu:~/Desktop$
```

2) Write a shell script that displays a list of all the files in the current directory

```
1 #!/bin/bash
2 echo "The list of file names in the current directory: "
3 for file in *
4 do
5 if [ -f $file ]
6 then
7 ls -l $file
8 fi
9 done
```

```

preksha@ubuntu:~/Desktop$ bash two.sh
The list of file names in the current directory:
-rw-rw-r-- 1 preksha preksha 69 Jun 14 06:51 baba1.txt
-rw-rw-r-- 1 preksha preksha 1836 Jun 12 10:45 baba.txt
-rw-rw-r-- 1 preksha preksha 22 Jun 14 03:00 blank.txt
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatefile
-rw-rw-r-- 1 preksha preksha 29 Jun 14 02:30 duplicate.txt
-rwxrwxrwx 1 preksha preksha 108 Jun 14 02:47 egs.txt
-rw-rw-r-- 1 preksha preksha 28 Jun 12 11:24 errorfile
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 97 Jun 14 06:51 humpty.txt
-rw-rw-r-- 1 preksha preksha 65 Jun 12 11:30 number
-rwxrwxr-x 1 preksha preksha 222 Jun 14 06:51 one.sh
-rw-rw-r-- 1 preksha preksha 30 Jun 14 02:29 original.txt
-rw-rw-r-- 1 preksha preksha 8 Jun 14 02:29 ori.txt
-rw-rw-r-- 1 preksha preksha 9 Jun 14 03:06 p1.txt
-rw-rw-r-- 1 preksha preksha 31 Jun 14 03:06 p2.txt
-rw-rw-r-- 1 preksha preksha 3 Jun 12 10:37 poem1
-rw-rw-r-- 1 preksha preksha 87 Jun 12 10:36 poem2
-rw-rw-r-- 1 preksha preksha 24 Jun 12 11:32 science.txt
-rw-rw-r-- 1 preksha preksha 0 Jun 14 03:01 twinkle.txt
-rwxrwxr-x 1 preksha preksha 127 Jun 14 06:58 two.sh
-rw-rw-r-- 1 preksha preksha 6 Jun 14 02:54 two.txt
-rw-rw-r-- 1 preksha preksha 8 Jun 14 02:49 twowords.txt
-rwxrwxrwx 1 preksha preksha 180 Jun 11 23:04 user.txt
-rw-rw-r-- 1 preksha preksha 8 Jun 14 02:51 words.txt
preksha@ubuntu:~/Desktop$ 

```

3)Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file or directory.

```

1#!/bin/bash
2echo -n "enter file or dir name :"
3read name
4
5if [ -d $name ]
6then
7    echo "given name is directory"
8elif [ -f $name ]
9then
10    echo "file name is : $name"
11
12fi

```

```

preksha@ubuntu:~/Desktop$ bash three.sh
enter file or dir name :p1.txt
file name is : p1.txt
preksha@ubuntu:~/Desktop$ bash three.sh
enter file or dir name :ex.txt
given name is directory
preksha@ubuntu:~/Desktop$ 

```

4)Write a shell script to find the factorial of given integer

```
preksha@ubuntu:~/Desktop$ bash four.sh
enter a number
4
24
```

```
1#!/bin/bash
2echo "enter a number"
3read num
4
5fact=1
6
7while [ $num -gt 1 ]
8do
9    fact=$((fact *num))
10   num=$((num - 1 ))
11done
12
13echo $fact|
```

5) write a shell script that list the all files in a directory.

```
1#!/bin/bash
2echo "The list of file names in the current directory: "
3for file in *
4do
5if [ -f $file ]
6then
7ls -l $file
8fi
9done
```

```
preksha@ubuntu:~/Desktop$ bash two.sh
The list of file names in the current directory:
-rw-rw-r-- 1 preksha preksha 69 Jun 14 06:51 baba1.txt
-rw-rw-r-- 1 preksha preksha 1836 Jun 12 10:45 baba.txt
-rw-rw-r-- 1 preksha preksha 22 Jun 14 03:00 blank.txt
-rw-rw-r-- 1 preksha preksha 59 Apr 27 00:25 concatfile
-rw-rw-r-- 1 preksha preksha 29 Jun 14 02:30 duplicate.txt
-rwxrwxrwx 1 preksha preksha 108 Jun 14 02:47 egs.txt
-rw-rw-r-- 1 preksha preksha 28 Jun 12 11:24 errorfile
-rw-rw-r-- 1 preksha preksha 28 Apr 27 00:25 f1
-rw-rw-r-- 1 preksha preksha 32 Apr 27 00:25 f2
-rw-rw-r-- 1 preksha preksha 27 Apr 27 00:25 f3
-rw-rw-r-- 1 preksha preksha 97 Jun 14 06:51 humpty.txt
-rw-rw-r-- 1 preksha preksha 65 Jun 12 11:30 number
-rwxrwxr-x 1 preksha preksha 222 Jun 14 06:51 one.sh
-rw-rw-r-- 1 preksha preksha 30 Jun 14 02:29 original.txt
-rw-rw-r-- 1 preksha preksha 8 Jun 14 02:29 ori.txt
-rw-rw-r-- 1 preksha preksha 9 Jun 14 03:06 p1.txt
-rw-rw-r-- 1 preksha preksha 31 Jun 14 03:06 p2.txt
-rw-rw-r-- 1 preksha preksha 3 Jun 12 10:37 poem1
-rw-rw-r-- 1 preksha preksha 87 Jun 12 10:36 poem2
-rw-rw-r-- 1 preksha preksha 24 Jun 12 11:32 science.txt
-rw-rw-r-- 1 preksha preksha 0 Jun 14 03:01 twinkle.txt
-rwxrwxr-x 1 preksha preksha 127 Jun 14 06:58 two.sh
-rw-rw-r-- 1 preksha preksha 6 Jun 14 02:54 two.txt
-rw-rw-r-- 1 preksha preksha 8 Jun 14 02:49 twowords.txt
-rwxrwxrwx 1 preksha preksha 180 Jun 11 23:04 user.txt
-rw-rw-r-- 1 preksha preksha 8 Jun 14 02:51 words.txt
preksha@ubuntu:~/Desktop$
```

6) Write a shell script that accepts a string from the terminal and echo a suitable message if it doesn't have at least 5 characters including the other symbols.

```
1 #!/bin/bash
2 echo "enter string"
3 read str
4
5 len=`echo $str | wc -c`
6 len=`echo $len -1 |bc`
7 if [ $len -lt 5 ]
8 then
9     echo "less than 5 characters"
10 fi
11
```

```
preksha@ubuntu:~/Desktop$ bash five.sh
enter string
hihello
preksha@ubuntu:~/Desktop$ bash five.sh
enter string
hi
less than 5 characters
preksha@ubuntu:~/Desktop$
```

7) Write a shell script to echo the string length of the given string as argument.

```
preksha@ubuntu:~/Desktop$ var="preksha"
preksha@ubuntu:~/Desktop$ expr length "$var"
7
preksha@ubuntu:~/Desktop$ var="twinkle twinkle little star"
preksha@ubuntu:~/Desktop$ expr length "$var"
27
preksha@ubuntu:~/Desktop$
```

8) Write a shell script that accepts two directory names as arguments and deletes those files in the first directory which are similarly named in the second directly. Note: Contents should also match inside the files.

```

1 #!/bin/bash
2 SRC="$1"
3 DST="$2"
4 if [ $# -ne 2 ]
5     echo "$(basename $0) dir1 dir2"
6     exit 1
7 fi
8
9 if [ ! -d $SRC ]
10 then
11     echo "directory $SRC does not exists!"
12     exit 2
13 fi
14
15 if [ ! -d $DST ]
16 then
17     echo "directory $DST does not exists!"
18     exit 2
19 fi
20
21 for f in $DST/*
22 do
23 #echo Processing $f
24     if [ -f $f ]
25     then
26         tFile="$SRC/$(basename $f)"
27         if [ -f $tFile ]
28             then
29                 echo -n "Deleting $tFile..."
30                 /bin/rm $tFile
31                 [ $? -eq 0 ] && echo "done" || echo "failed"
32             fi
33
34     fi
35
36 done

```

EXTRA PROGRAMS:

1. Write a shell script that displays the last modification time of any file.

```

1 #!/bin/bash
2 echo -n "enter a filename to see last modification time :"
3 read fn
4
5 # make sure file exists
6 if [ ! -f $fn ]
7 then
8     echo "$fn not a file"
9     exit 1
10 fi
11
12 # use stat command to display
13 echo "$fn was last modified on $(stat -c %x $fn)"
```

```

preksha@ubuntu:~/Desktop$ cat atwords
hat
bat
sat
fat

preksha@ubuntu:~/Desktop$ bash six.sh
enter a filename to see last modification time :atwords
atwords was last modified on 2021-06-14 07:54:55.842036324 -0700
preksha@ubuntu:~/Desktop$
```

2. Write a shell script that changes the name of the files passed as arguments to lowercase.

```
1 file="$1"
2 if [ $# -eq 0 ]
3 then
4     echo "$(basename $0) file"
5     exit 1
6 fi
7
8 if [ ! $file ]
9 then
10    echo "$file not a file"
11    exit 2
12 fi
13
14 lowercase=$(echo $file | tr '[A-Z]' '[a-z]')
15
16 if [ -f $lowercase ]
17 then
18    echo "Error - File already exists!"
19    exit 3
20 fi
21
22 # change file name
23 /bin/mv $file $lowercase
24
```

3. Write a shell script to combine any three text files into a single file (append them in the order as they appear in the arguments) and display the word count.

```
preksha@ubuntu:~/Desktop$ cat f1
hi,how r u?
ok bye!see you!
preksha@ubuntu:~/Desktop$ cat f2
apple
mango
banana
berry
cherry
preksha@ubuntu:~/Desktop$ cat f3
carrot
cabbage
cauliflower
preksha@ubuntu:~/Desktop$ bash eight.sh f1 f2 f3
14 words written to output.
preksha@ubuntu:~/Desktop$ cat output
cat: output: No such file or directory
preksha@ubuntu:~/Desktop$ cat output.
hi,how r u?
ok bye!see you!
apple
mango
banana
berry
cherry
carrot
cabbage
cauliflower
preksha@ubuntu:~/Desktop$
```

```
1#!/bin/bash
2 f1=$1
3 f2=$2
4 f3=$3
5 op="output.$s"
6 count=0
7 if [ $# -ne 3 ]
8 then
9     echo "$(basename $0) f1 f2 f3"
10    exit 1
11 fi
12 if [ ! -f $f1 ]
13 then
14     echo "$f1 not a file!"
15     exit 2
16 fi
17 if [ ! -f $f2 ]
18 then
19     echo "$f2 not a file!"
20     exit 3
21 fi
22 if [ ! -f $f3 ]
23 then
24     echo "$f3 not a file!"
25     exit 4
26 fi
27 cat $f1 $f2 $f3 >> $op
28 count=$(cat $op | wc -w)
29 echo "$count words written to $op"
30
```

4. Write a script called hello which outputs the following:

- your username
- the time and date
- who is logged on

```
1 #!/bin/bash
2 function line(){
3     echo "~~~~~"
4 }
5
6 echo "username: $(echo $USER)"
7 line # call function
8
9 echo "current date and time: $(date)"
10 line
11
12 echo "currently logged on users:"
13 who
14 line
```

```
preksha@ubuntu:~/Desktop$ bash nine.sh
username: preksha
~~~~~
current date and time: Mon 14 Jun 2021 09:01:25 AM PDT
~~~~~
currently logged on users:
preksha :0          2021-06-14 01:53 (:0)
~~~~~
preksha@ubuntu:~/Desktop$
```

FILE API ASSIGNMENT

Program to retrieve the permission of a file

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
int main(int argc, char **argv)
{
    int result;
    char *filename = (char *)malloc(512);
    if (argc < 2)
    {
```

```
        strcpy(filename, "/usr/bin/adb");

    }

    else

    {

        strcpy(filename, argv[1]);

    }

    result = access (filename, R_OK);

    if ( result == 0 )

    {

        printf("%s has read permissions\n",filename);

    }

    else

    {

        printf("%s has no read permissions\n",filename);

    }

    result = access (filename, W_OK);

    if ( result == 0 )

    {

        printf("%s has write permissions\n",filename);

    }

    else

    {

        printf("%s has no write permission\n",filename);

    }

    result = access (filename, X_OK);
```

```

if ( result == 0 )

{

printf("%s is executable\n",filename);

}

else

{

printf("%s is not executable\n",filename);

}

free(filename);

return 0;

}

```

```

rachita@rachita-VirtualBox:~/USPlab$ gcc files_permission.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out 1.txt
1.txt has read permissions
1.txt has no write permission
1.txt is not executable
rachita@rachita-VirtualBox:~/USPlab$ chmod 777 1.txt
rachita@rachita-VirtualBox:~/USPlab$ ./a.out 1.txt
1.txt has read permissions
1.txt has write permissions
1.txt is executable

```

Program to find the number of links

```

#include <stdio.h>

#include <sys/stat.h>

int main ( int argc, char ** argv )

{
    int    i;

```

```
struct stat st;

if(argc<=1)

{

    printf("enter file name in the command line\n");

}

else

{

    for (i = 1; i < argc; ++i)

    {

        if (stat(argv[i], &st) == -1)

        {

            perror(argv[i]);

        }

        else

        {

            printf("%s has %ld link(s)\n", argv[i], st.st_nlink);

        }

    }

}

return 0;

}
```

```

rachita@rachita-VirtualBox:~/USPlab$ gcc files_no_of_links.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
enter file name in the command line
rachita@rachita-VirtualBox:~/USPlab$ ./a.out 1.txt
1.txt has 1 link(s)
rachita@rachita-VirtualBox:~/USPlab$ ./a.out testsource.txt
testsource.txt has 2 link(s)
rachita@rachita-VirtualBox:~/USPlab$ ln testsource.txt link2
rachita@rachita-VirtualBox:~/USPlab$ ./a.out testsource.txt
testsource.txt has 3 link(s)
rachita@rachita-VirtualBox:~/USPlab$ ls
1.txt           file3           locks1.c
2.txt           filenew         locks.c
3.txt           filenew.txt     modif.sh
a.out          files_all.c   name_change.sh
a.txt           files.c        one1.txt
combine.sh      files_fifo.cpp open.c
compile.txt     file.sh        outp.txt
del_lines.sh    files_link.c  output
details.sh      files_link.cpp outpu.txt
devide_api.c   files_lseek.c  science.txt
devide_api.cpp  files_lseek.cpp string_length.sh
directory       files_no_of_links.c string.sh
directory_list.sh files_permission.c testsource.txt
displaylines.txt files_read.c   test.txt
duplicate.txt   files_unlink.cpp three.txt
errorFile       files_write.c  two3.txt
factorial.sh   fool.txt      two.txt
fill            foo.txt       usp.txt
file1           ISE.txt      USP.txt
file1.txt       link2        xx.txt
file2.txt       linkdest

```

Program to find the user and group name

```

#include <pwd.h>

#include <grp.h>

#include <sys/stat.h>

#include<stdio.h>

int main(int argc, char ** argv)

{
    struct stat info;

    if(argc<=1)

    {

        printf("enter file name in the command line\n");

    }

    else

```

```

    {
        for (int i = 1; i < argc; ++i)

        {

            stat(argv[i], &info); // Error check omitted

            struct passwd *pw = getpwuid(info.st_uid);

            struct group *gr = getgrgid(info.st_gid);

            printf("For file %s :\n", argv[i]);

            printf("\t%s is the username\n", pw->pw_name);

            printf("\t%s is the groupname\n", gr->gr_name);

        }

    }

    return 0;
}

```

```

group_rachita -groupname
rachita@rachita-VirtualBox:~/USPlab$ gcc files_user_and_group.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
enter file name in the command line
rachita@rachita-VirtualBox:~/USPlab$ ./a.out a
For file a :
    root is the username
    root is the groupname
rachita@rachita-VirtualBox:~/USPlab$ ./a.out 1.txt testsource.txt
For file 1.txt :
    rachita is the username
    group_rachita is the groupname
For file testsource.txt :
    rachita is the username
    group_rachita is the groupname
rachita@rachita-VirtualBox:~/USPlab$ █

```

Program to find the type of the file. The input will be given as a command line argument.

```
#include <sys/stat.h>

#include <time.h>
```

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    struct stat sb;

    if(argc<=1)
    {
        printf("enter file name in the command line\n");
    }
    else
    {
        for (int i = 1; i < argc; ++i)
        {
            if (stat(argv[i], &sb) == -1)
            {
                perror(argv[i]);
                return 0;
            }

            printf("\nFor %s \n\ttype:", argv[i]);

            switch (sb.st_mode & S_IFMT)
            {

```



```

rachita@rachita-VirtualBox:~/USPlab$ gcc files_type.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
enter file name in the command line
rachita@rachita-VirtualBox:~/USPlab$ ./a.out a
a: No such file or directory
rachita@rachita-VirtualBox:~/USPlab$ ./a.out test_directory 1.txt

For test_directory
    type:directory

For 1.txt
    type:regular file
rachita@rachita-VirtualBox:~/USPlab$ ./a.out testsource
testsource: No such file or directory
rachita@rachita-VirtualBox:~/USPlab$ ./a.out linkdest

For linkdest
    type:regular file
rachita@rachita-VirtualBox:~/USPlab$ █

```

Program to create a file f1.txt. Write to the file “We are quarantined since one month” and the move the file pointer to 50 bytes ahead and then write “we are not sure when lock down will be released”

Filename used-output1.txt

```

#include<stdio.h>

#include<unistd.h>

#include<sys/types.h>

#include<iostream>

#include<fcntl.h>

#include<cstring>

using namespace std;

int main()

{

    int fd;

    char c1[]="We are quarantined since one month";

    char c2[]="We are not sure when lock down will be released";

    fd=creat("output1.txt",0777);

    write(fd,c1,strlen(c1));

```

```

    lseek(fd,50,SEEK_CUR);

    write(fd,c2,strlen(c2));

    return 0;

}

```

offset-50,SEEK_CUR

```

rachita@rachita-VirtualBox:~/USPlab$ g++ files_write_pos.cpp
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
rachita@rachita-VirtualBox:~/USPlab$ cat output1.txt
We are quarantined since one monthWe are not sure when lock down will b
e releasedrachita@rachita-VirtualBox:~/USPlab$ g++ files_write_pos.cpp

```

offset-10,SEEK_CUR

```

e releasedrachita@rachita-VirtualBox:~/USPlab$ g++ files_write_pos.cpp
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
rachita@rachita-VirtualBox:~/USPlab$ cat output1.txt
We are quarantined since one monthWe are not sure when lock down will b
e releasedrachita@rachita-VirtualBox:~/USPlab$ g++ files_write_pos.cpp

```

offset-10,SEEK_SET

```

e releasedrachita@rachita-VirtualBox:~/USPlab$ g++ files_write_pos.cpp
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
rachita@rachita-VirtualBox:~/USPlab$ cat output1.txt
We are quaWe are not sure when lock down will be releasedrachita@rachit
a-VirtualBox:~/USPlab$ █

```

Program to read the first 10 bytes of data from the file f1.txt and again read the file contents from 20 th byte.

```

#include<unistd.h>

#include<fcntl.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<stdio.h>

int main()

```

```

{
    int n,f;
    char buff[20],buff2[20];
    f=open("file1",O_RDWR);
    read(f,buff,10);
    printf("%s\n",buff);
    lseek(f,10,SEEK_CUR);
    read(f,buff2,20);
    printf("%s\n",buff2);
    return 0;
}

```

```

rachita@rachita-VirtualBox:~/USPlab$ cat file1
hello123kes about 75% of our body
all living beings require water for survival
save water
thank you
hi
hello
Have a nice evening
hello
rachita@rachita-VirtualBox:~/USPlab$ gcc files_readlseek.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
hello123keoXU
% of our body
all lio
rachita@rachita-VirtualBox:~/USPlab$ 

```

Problem based assignment on API's

1. Write a C program that takes one or more file/directory names as command line input and reports the following information on the file:

- o File type
- o Number of links
- o Time of last access

- o Read, write, and execute permission

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

char * error_msg[] = {

    "\nUsage: ./12filestat <file|dir> [<file|dir>]\n\n",
    "\nFile does not exist !!\n\n",
    "\nError doing 'stat' on file\n\n"

};

void print_error(int msg_num, int exit_code, int exit_flag);

int main(int argc, char *argv[])
{
    int i;
    mode_t file_perm;
    struct stat file_details;
```

```
char success_msg[] = "\nCommand executed successfully\n\n";  
if ( argc < 2 )    print_error(0,2,1);  
  
for ( i = 1; i < argc; i++ )  
  
{  
  
printf("\n%s\n%s\n%s\n",-----,argv[i],-----);  
  
if ( access(argv[i],F_OK) == -1 )  
  
{  
  
print_error(1,3,0);  
  
continue; /* Check the next file */  
  
}  
  
if ( lstat(argv[i],&file_details) < 0 )  
  
{  
  
print_error(2,4,0);  
  
continue; /* Check the next file */
```

```
}

if ( S_ISREG(file_details.st_mode) )

    printf("File type : Regular\n");

else if ( S_ISDIR(file_details.st_mode) )

    printf("File type : Directory\n");

else if ( S_ISLNK(file_details.st_mode) )

    printf("File type : Symbolic link\n");

else

    printf("File type : Other");

printf("Number of links : %d\n", (int)file_details.st_nlink);

/* Get the time of last access of the file */

printf("Time of last access : %s", ctime(&file_details.st_atime));

/* Get the file permissions */

printf("File Permissions:\n");
```

```
file_perm = file_details.st_mode & ~S_IFMT;

printf("\tUser : ");

if ( file_perm & S_IRUSR ) printf("Readable, ");

else printf("Not readable, ");

if ( file_perm & S_IWUSR ) printf("Writable, ");

else printf("Not writable, ");

if ( file_perm & S_IXUSR ) printf("Executable\n");

else printf("Not executable\n");

printf("\tGroup : ");

if ( file_perm & S_IRGRP ) printf("Readable, ");

else printf("Not readable, ");

if ( file_perm & S_IWGRP ) printf("Writable, ");

else printf("Not writable, ");
```

```
    if ( file_perm & S_IXGRP ) printf("Executable\n");

    else printf("Not executable\n");

    printf("\tOthers : ");

    if ( file_perm & S_IROTH ) printf("Readable, ");

    else printf("Not readable, ");

    if ( file_perm & S_IWOTH ) printf("Writable, ");

    else printf("Not writable, ");

    if ( file_perm & S_IXOTH ) printf("Executable\n");

    else printf("Not executable\n");

}

printf("%s", success_msg);

return 1;

}

void print_error(int error_index, int exit_code, int exit_flag)
```

```

{
    fprintf(stderr, "%s\n", error_msg[error_index]);

    if (exit_flag) exit(exit_code);

}

```

```
rachita@rachita-VirtualBox:~/USPLab$ ./a.out two.txt files_permission.c directory

-----
two.txt
-----
File type : Regular
Number of links : 1
Time of last access : Tue Jun 22 11:38:31 2021
File Permissions:
    User : Readable, Writable, Executable
    Group : Readable, Not writable, Not executable
    Others : Readable, Not writable, Not executable

-----
files_permission.c
-----
File type : Regular
Number of links : 1
Time of last access : Fri Jul  9 19:16:08 2021
File Permissions:
    User : Readable, Writable, Not executable
    Group : Readable, Writable, Not executable
    Others : Readable, Not writable, Not executable

-----
directory
-----
File type : Directory
Number of links : 2
Time of last access : Sat Jun 12 21:18:34 2021
File Permissions:
    User : Readable, Writable, Executable
    Group : Readable, Writable, Executable
    Others : Readable, Not writable, Executable

Command executed successfully
```

2. Write a program in C that illustrates how to execute two commands concurrently with a command pipe.

```
#include<stdio.h>
#include<fcntl.h>
#include<stdlib.h>
#include<unistd.h>

void main()
{
    int pfd[2],p;
    pipe(pfd);
    p=fork();
    if(p==0)
    {
        close(pfd[0]);
        close(1);
        dup(pfd[1]);
        execlp("ls","ls","-l",(char*)0);
    }
    else
    {
        close(pfd[1]);
        close(0);
        dup(pfd[0]);
        execlp("wc","wc",(char*)0);
    }
}
```

```
rachita@rachita-VirtualBox:~/USPLab$ gcc commands_pipe.c
rachita@rachita-VirtualBox:~/USPLab$ ./a.out
86      769      5576
```

3. Write a C Program that makes a copy of a file using standard I/O and system calls

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int f1, f2;
    char buff[50];
    long int n;

    if(((f1 = open(argv[1], O_RDONLY)) == -1 || ((f2=open(argv[2], O_CREAT | O_WRONLY | O_TRUNC,
0700))== 1)))
    {
        perror("problem in file");
        exit(1);
    }

    while((n=read(f1, buff, 50))>0)
        if(write(f2, buff, n)!=n)
    {
        perror("problem in writing");
        exit(3);
    }
}
```

```

    }

    if(n===-1)

    {

        perror("problem in reading");

        exit(2);

    }

    close(f2);

    exit(0);

}

```

```

rachita@rachita-VirtualBox:~/USPlab$ gcc files_copy.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out files_copy.c s.txt
rachita@rachita-VirtualBox:~/USPlab$ cat s.txt
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    int f1, f2;
    char buff[50];
    long int n;
    if(((f1 = open(argv[1], O_RDONLY)) == -1 || ((f2=open(argv[2], O_CREAT | O_WRONLY | O_TRUNC,
0700))== 1)))
    {
        perror("problem in file");
        exit(1);
    }
    while((n=read(f1, buff, 50))>0)
    if(write(f2, buff, n)!=n)
    {
        perror("problem in writing");
        exit(3);
    }
    if(n===-1)
    {
        perror("problem in reading");
        exit(2);
    }
    close(f2);
    exit(0);
}

```

4. Implement in C the following Unix commands using system calls

A)Cat

```
#include<sys/types.h>
#include<sys/stat.h>
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>

void main( int argc,char *argv[3] )

{
    int fd,i;
    char buf[2];
    fd=open(argv[1],O_RDONLY,0777);
    if(fd==argc)
    {
        printf("file open error");
    }
    else
    {
        while((i=read(fd,buf,1))>0)
        {
            printf("%c",buf[0]);
        }
        close(fd);
    }
}
```

```
rachita@rachita-VirtualBox:~/USPLab$ gcc files_cat.c
rachita@rachita-VirtualBox:~/USPLab$ ./a.out file1
hello123kes about 75% of our body
all living beings require water for survival
save water
thank you
hi
hello
Have a nice evening
    hello
```

B)mv

```
#include<sys/types.h>
#include<sys/stat.h>
#include<stdio.h>
#include<fcntl.h>
#include<unistd.h>
void main( int argc,char *argv[] )
{
    int i,fd1,fd2;
    char *file1,*file2,buf[2];
    file1=argv[1];
    file2=argv[2];
    printf("file1=%s file2=%s",file1,file2);
    fd1=open(file1,O_RDONLY,0777);
    fd2=creat(file2,0777);
    while(i=read(fd1,buf,1)>0)
        write(fd2,buf,1);
    remove(file1);
```

```

        close(fd1);

        close(fd2);

    }

```

```

rachita@rachita-VirtualBox:~/USPlab$ gcc files_mv.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out file1 directory/filenew
file1=file1 file2=directory/filenewrachita@rachita-VirtualBox:~/USPlab$ cd direct
bry
rachita@rachita-VirtualBox:~/USPlab/directory$ ls
filenew
rachita@rachita-VirtualBox:~/USPlab/directory$
```

c)cp

```

#include<sys/types.h>

#include<sys/stat.h>

#include<stdio.h>

#include<fcntl.h>

#include<unistd.h>

void main( int argc,char *argv[] )

{

    int i,fd1,fd2;

    char *file1,*file2,buf[2];

    file1=argv[1];

    file2=argv[2];

    printf("file1=%s file2=%s",file1,file2);

    fd1=open(file1,O_RDONLY,0777);

    fd2=creat(file2,0777);

    while(i=read(fd1,buf,1)>0)

        write(fd2,buf,1);

```

```

        close(fd1);

        close(fd2);

    }

```

```
rachita@rachita-VirtualBox:~/USPlab$ ./a.out two.txt  directory/two_new1.txt
file1=two.txt file2=directory/two_new1.txtrachita@rachita-VirtualBox:~/USPlab$
rachita@rachita-VirtualBox:~/USPlab$ ls
1.txt          file3           files_write.c      signals2.c
2.txt          filenew         files_write_pos.cpp signals2.cpp
3.txt          filenew.txt     fool.txt          signals3.cpp
a.out          files_all.c    foo.txt           signals4.cpp
a.txt          files_c         ISE.txt           signals5.cpp
combine.sh     files_cat.c    link2             signals6.c
commands_pipe.c files_copy.c   linkdest          signals6.cpp
compile.txt    files_cp.c     locks1.c         signals7.cpp
del_lines.sh   files_fifo.cpp locks.c           string_length.sh
details.sh     file.sh        modif.sh          string.sh
devide_api.c   files_link.c   name_change.sh  s.txt
devide_api.cpp files_link.cpp new1.c            test.c
directory      files_lseek.c  new.c              test_directory
directory_list.sh files_lseek.cpp newoutput1.txt testsource.txt
displaylines.txt files_mv.c    newoutput.txt   test.txt
duplicate.txt  files_no_of_links.c onel.txt        three.txt
errorFile      files_permission.c open.c          two3.txt
f5.txt         files_read.c   outp.txt          two.txt
factorial.sh   files_readlseek.c output           usp.txt
fill           file_Stat.c    output1.txt       USP.txt
file1.txt      files_type.c   outpu.txt         xx.txt
file2          files_unlink.cpp recordlocking.c
file2.txt      files_user_and_group.c science.txt
rachita@rachita-VirtualBox:~/USPlab$ cd directory
rachita@rachita-VirtualBox:~/USPlab/directory$ ls
a.out  filenew  two_new1.txt  two_new.txt
rachita@rachita-VirtualBox:~/USPlab/directory$ █
```

d)In

```

#include<stdio.h>

#include<unistd.h>

#include<iostream>

using namespace std;

```

```
int main(int argc,char*argv[])
{
    if(argc!=3)
    {
        cerr<<"usage:<<argv[0]<<<src_file><dest_file>\n";
        return 0;
    }
    if(link(argv[1],argv[2])==-1)
    {
        perror("link");
        return 1;
    }
    return 0;
}
```

```
rachita@rachita-VirtualBox:~/USPlab$ g++ files_link.cpp
rachita@rachita-VirtualBox:~/USPlab$ ./a.out two.txt two_link.txt
rachita@rachita-VirtualBox:~/USPlab$ ls
1.txt           file3           files_write.c      signals2.c
2.txt           filenew         files_write_pos.cpp signals2.cpp
3.txt           filenew.txt     fool.txt          signals3.cpp
a.out          files_all.c    foo.txt            signals4.cpp
a.txt           files.c          ISE.txt           signals5.cpp
combine.sh      files_cat.c    link2              signals6.c
commands_pipe.c files_copy.c   linkdest          signals6.cpp
compile.txt     files_cp.c     locks1.c          signals7.cpp
del_lines.sh    files_fifo.cpp locks.c            string_length.sh
details.sh      file.sh        modif.sh          string.sh
devide_api.c    files_link.c  name_change.sh    s.txt
devide_api.cpp  files_link.cpp newl.c             test.c
directory       files_lseek.c new.c               test_directory
directory_list.sh files_lseek.cpp newoutput1.txt  testsource.txt
displaylines.txt files_mv.c    newoutput.txt    test.txt
duplicate.txt   files_no_of_links.c one1.txt        three.txt
errorFile       files_permission.c open.c          two3.txt
f5.txt          files_read.c  outp.txt          two_link.txt
factorial.sh   files_readlseek.c output           two.txt
fill           file_stat.c   output1.txt      user.txt
```

5. Write a C program to list for every file in a directory, its inode number and file name.

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
void main(int argc, char *argv[])
{
    char d[50];
    if(argc==2)
    {
        bzero(d,sizeof(d));
        strcat(d,"ls ");
        strcat(d,"-i ");
        strcat(d,argv[1]);
```

```

        system(d);
    }
else
printf("\nInvalid No. of inputs");
}

```

```

rachita@rachita-VirtualBox:~/USPlab/directory$ cd ..
rachita@rachita-VirtualBox:~/USPlab$ gcc files_inode_name.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out directory
1703894 a.out 1703906 filenew 1702645 two_new1.txt 1703908 two_new.txt
rachita@rachita-VirtualBox:~/USPlab$ █

```

6. Write a C Program that demonstrates redirection of standard output to a file .EX:ls>f1.

```

#include<stdlib.h>
#include<stdio.h>
#include<string.h>
void main(int argc, char *argv[])
{
    char d[50];
    if(argc==2)
    {
        bzero(d,sizeof(d));
        strcat(d,"ls ");
        strcat(d,> " );
        strcat(d,argv[1]);
    }
}

```

```

        system(d);

    }

else

printf("\nInvalid No. of inputs");

}

```

```

rachita@rachita-VirtualBox:~/USPlab/directory$ gcc files_redirect.c
rachita@rachita-VirtualBox:~/USPlab/directory$ ./a.out list
rachita@rachita-VirtualBox:~/USPlab/directory$ ls
a.out  filenew  files_redirect.c  list  two_new1.txt  two_new.txt
rachita@rachita-VirtualBox:~/USPlab/directory$ cat list
a.out
filenew
files_redirect.c
list
two_new1.txt
two_new.txt

```

7. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen

```

#include<stdio.h>

#include<sys/wait.h>

#include<stdlib.h>

#include<unistd.h>

int main(void)

{

    int pid;

    int status;

    printf("Printing based on pid:\n");

    pid = fork();

    if(pid == -1)

```

```

    {
        perror("bad fork");
        exit(1);
    }

    if (pid == 0)
        printf("Child\n");
    else
    {
        wait(&status);
        printf("Parent\n");
    }
}

```

```

parent
rachita@rachita-VirtualBox:~/USPlab$ gcc parent_child.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
Printing based on pid:
Child
Parent

```

8. Write a C program to create a Zombie process.

```

#include <stdlib.h>

#include <sys/types.h>

#include<stdio.h>

#include <unistd.h>

int main ()
{
    int pid_t,child_pid;

```

```

child_pid = fork ();
if (child_pid > 0) {
    sleep (60);
}
else {
    exit (0);
}
return 0;
}

```



```

rachita@rachita-VirtualBox:~/USPlab$ gcc zombie_process.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
rachita@rachita-VirtualBox:~/USPlab$ █

```

9. Write a C program that illustrates how an orphan is created.

```

#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main()
{
    int pid;
    pid=fork();
    if(pid==0)
    {
        printf("IAM CHILD PROCESS,MY PROCESS ID IS: %d",getpid());
    }
}

```

```

printf("\n THE CHILDS'S PARENT PROCESSS ID IS : %d\n",getppid());

sleep(10);

printf("\n *****AFTER 10 SECUNDS*****");

printf("\n IAM CHILD PROCESS,MY PROCESS ID IS: %d",getpid());

printf("\n THE CHILDS'S PARENT PROCESS ID IS :%d\n",getppid());

}

else

{

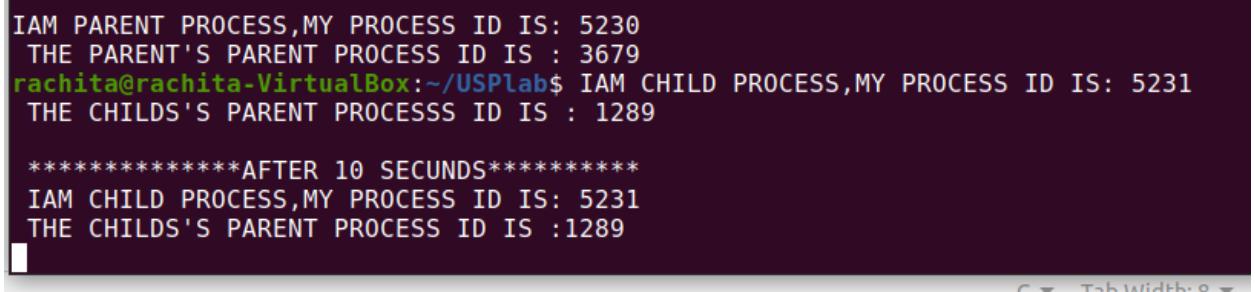
printf("\nIAM PARENT PROCESS,MY PROCESS ID IS: %d",getpid());

printf("\n THE PARENT'S PARENT PROCESS ID IS : %d\n",getppid());

}

}

```



IAM PARENT PROCESS,MY PROCESS ID IS: 5230
THE PARENT'S PARENT PROCESS ID IS : 3679
rachita@rachita-VirtualBox:~/USP\$ IAM CHILD PROCESS,MY PROCESS ID IS: 5231
THE CHILDS'S PARENT PROCESSS ID IS : 1289
*****AFTER 10 SECUNDS*****
IAM CHILD PROCESS,MY PROCESS ID IS: 5231
THE CHILDS'S PARENT PROCESS ID IS :1289

10. Write a program that illustrates how to execute two commands concurrently with a command

```

#include<stdio.h>

#include<fcntl.h>

#include<stdlib.h>

#include<unistd.h>

```

```

void main()
{
    int pfd[2],p;
    pipe(pfd);
    p=fork();
    if(p==0)
    {
        close(pfd[0]);
        close(1);
        dup(pfd[1]);
        execlp("ls","ls","-l",(char*)0);
    }
    else
    {
        close(pfd[1]);
        close(0);
        dup(pfd[0]);
        execlp("wc","wc",(char*)0);
    }
}

```

```

rachita@rachita-VirtualBox:~/USPLab$ gcc commands_pipe.c
rachita@rachita-VirtualBox:~/USPLab$ ./a.out
      86      769      5576

```

11. Write a C programs that illustrate communication between two unrelated processes using named pipe.

```
#include<stdio.h>
```

```

#include<stdlib.h>
#include<errno.h>
#include<unistd.h>
int main()
{
    int pfds[2];
    char buf[30];
    if(pipe(pfds)==-1)
    {
        perror("pipe");
        exit(1);
    }
    printf("writing to file descriptor #%d\n", pfds[1]);
    write(pfds[1],"test",5);
    printf("reading from file descriptor #%d\n ", pfds[0]);
    read(pfds[0],buf,5);
    printf("read\"%s\"\n",buf);
}

```

```
rachita@rachita-VirtualBox:~/USPLab$ ./a.out
writing to file descriptor #4
reading from file descriptor #3
read"test"
```

12. Write a C program that illustrates suspending and resuming processes using signals.

```

#include<stdio.h>
#include<signal.h>
```

```

#include<stdlib.h>
#include<unistd.h>
void alarms(int signo)
{
    printf("alarm function. signo = %d\n",signo);
}
int main()
{
    if(signal(SIGALRM,alarms)==SIG_ERR)
        printf("Signal error");
    alarm(5);
    for(int i=0;i<10;i++)
        printf(" i=%d \n",i);
    printf("main program terminated\n");
    return 0;
}

```

```

rachita@rachita-VirtualBox:~/USPlab$ gcc resume_suspend.c
rachita@rachita-VirtualBox:~/USPlab$ ./a.out
i=0
i=1
i=2
i=3
i=4
i=5
i=6
i=7
i=8
i=9
main program terminated

```

13. Write a separate program using signal system call to catch the following signals.

- o SIGSEGV
- o SIGINT
- o SIGFPE
- o SIGALRM (use alarm system call)
- o SIGALRM (use setitimer system call)
- o SIGVTALRM (use setitimer system call)
- o SIGPROF (use setitimer system call)

```
#include <iostream>
#include <signal.h>
using namespace std;
int main()
{
    sigset_t sigset;
    sigemptyset(&sigset);
    sigaddset(&sigset, SIGINT);
    sigaddset(&sigset, SIGUSR1);
    sigaddset(&sigset, SIGQUIT);
    sigaddset(&sigset, SIGALRM);
    sigaddset(&sigset, SIGFPE);
    sigaddset(&sigset, SIGSEGV);
    if (sigprocmask(SIG_SETMASK, &sigset, 0) == -1)
```

```
perror("sigprocmask");

/*initialize set*/

if (sigprocmask(0, NULL, &sigset) < 0)
perror("sigprocmask error");

if (sigismember(&sigset, SIGINT)) cout<<"SIGINT"=<<endl;
if (sigismember(&sigset, SIGFPE)) cout<<"SIGFPE "=<<endl;
if (sigismember(&sigset, SIGQUIT)) cout<<"SIGQUIT "=<<endl;
if (sigismember(&sigset, SIGUSR1)) cout<<"SIGUSR1 "=<<endl;
if (sigismember(&sigset, SIGUSR1)) cout<<"SIGUSR1 "=<<endl;
if (sigismember(&sigset, SIGALRM)) cout<<"SIGALRM "=<<endl;
if (sigismember(&sigset, SIGSEGV)) cout<<"SIGSEGV "=<<endl;

}
```

```
rachita@rachita-VirtualBox:~/USP lab$ ./a.out
SIGINT
SIGFPE
SIGQUIT
SIGUSR1
SIGUSR1
SIGALRM
SIGSEGV
```