

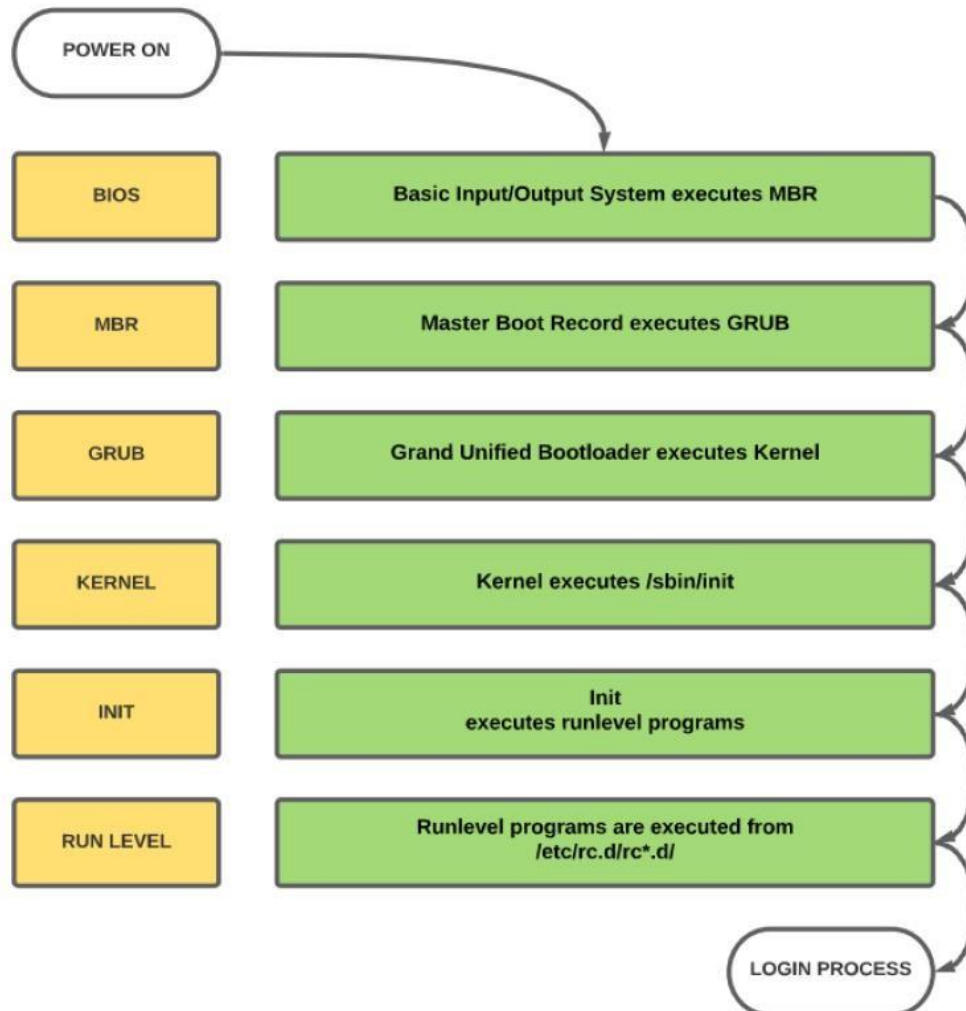
# INDEX

S NO.	DATE	NAME OF THE EXPERIMENT	PAGE NO.
1		Understanding the booting process of Linux	
2		Understanding the Linux file system	
3		Understanding the various Phases of Compilation of a 'C' Program	
4		System admin commands – Basics	
5		System admin commands – Simple task automations	
6		Linux commands	
7		Shell Programs – Basic level	
8		Process Creation	
9		Overlay concept	
10		IPC using Pipes	
11		IPC using shared memory and Message queues	
12		Process synchronization	
13		Study of OS161	
14		Understanding the OS161 file system and working with test programs	

## OS Exp.1: UNDERSTANDING THE BOOTING PROCESS OF LINUX

### • BOOTING PROCESS:

Press the power button on your system, and after few moments you see the Linux login prompt. From the time you press the power button until the Linux login prompt appears, the following sequence occurs. The following are the 6 high level stages of a typical Linux boot process.



#### Step 1. BIOS:

- BIOS stands for Basic Input/Output System
- Performs some system integrity checks
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, CD-ROMs, or hard drive.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

#### Step 2. MBR:

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.

- It contains information about GRUB (or LILO in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.

### **Step 3. GRUB**

- GRUB stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has the knowledge of the filesystem (the older Linux loader LILO didn't understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5PAE)
root(hd0,0)
kernel/boot/vmlinuz-2.6.18-194.el5PAE ro
root=LABEL=/
initrd /boot/initrd-2.6.18-194.el5PAE.img
```

- As you notice from the above info, it contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

### **Step 4. Kernel**

- Mounts the root file system as specified in the "root=" in grub.conf
- Kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a 'ps -ef | grep init' and check the pid.
- initrd stands for Initial RAM Disk.
- initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

### **Step 5. Init**

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels
  - ← 0 – halt
  - ← 1 – Single user mode
  - ← 2 – Multiuser, without NFS
  - ← 3 – Full multiuser mode
  - ← 4 – unused
  - ← 5 – X11
  - ← 6 – reboot
- Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.

- Execute 'grep initdefault /etc/inittab' on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.
- 

### **Step 6. Runlevel programs**

- When the Linux system is booting up, you might see various services getting started.
- For example, it might say "starting sendmail .... OK". Those are the runlevel programs, executed from the run level directory as defined by your run level.
- Depending on your default init level setting, the system will execute the programs from one of the following directories.
  - o Run level 0 – /etc/rc.d/rc0.d/
  - o Run level 1 – /etc/rc.d/rc1.d/
  - o Run level 2 – /etc/rc.d/rc2.d/
  - o Run level 3 – /etc/rc.d/rc3.d/
  - o Run level 4 – /etc/rc.d/rc4.d/
  - o Run level 5 – /etc/rc.d/rc5.d/
  - o Run level 6 – /etc/rc.d/rc6.d/
- Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
- Under the /etc/rc.d/rc\*.d/ directories, you would see programs that start with S and K.
- Programs starts with S are used during startup. S for startup.
- Programs starts with K are used during shutdown. K for kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
- For example, S12syslog is to start the syslog daemon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

### **Login Process**

1. Users enter their username and password
2. The operating system confirms your name and password.
3. A "shell" is created for you based on your entry in the "/etc/passwd" file
4. You are "placed" in your "home" directory.
5. Start-up information is read from the file named "/etc/profile". This file is known as the system login file. When every user logs in, they read the information in this file.
6. Additional information is read from the file named ".profile" that is located in your "home" directory. This file is known as your personal login file.

## OS Exp. 2: UNDERSTANDING THE LINUX FILE SYSTEM

- **Linux File System**

Linux File System or any file system generally is a layer which is under the operating system that handles the positioning of your data on the storage, without it; the system cannot know which file starts from where and ends where.

Linux offers many file systems types like:

- ← **Ext**: an old one and no longer used due to limitations.
- ← **Ext2**: first Linux file system that allows 2 terabytes of data allowed.
- ← **Ext3**: came from Ext2, but with upgrades and backward compatibility.
- ← **Ext4**: faster and allow large files with significant speed. (**Best Linux File System**)

It is a very good option for SSD disks and you notice when you try to install any Linux distro that this one is the default file system that Linux suggests.

- ← **JFS**: old file system made by IBM. It works very well with small and big files, but it failed and files corrupted after long time use, reports say.
- ← **XFS**: old file system and works slowly with small files.
- ← **Btrfs**: made by Oracle. It is not stable as Ext in some distros, but you can say that it is a replacement for it if you have to. It has a good performance.

### File System Structure

The following table provides a short overview of the most important higher-level directories you find on a Linux system.

Directory	Contents
/	Root directory—the starting point of the directory tree.
/bin	Essential binary files. Binary Executable files
/boot	Static files of the boot loader.
/dev	Files needed to access host-specific devices.
/etc	Host-specific system configuration files.
/lib	Essential shared libraries and kernel modules.
/media	Mount points for removable media.
/mnt	Mount point for temporarily mounting a file system.
/opt	Add-on application software packages.
/root	Home directory for the superuser <code>root</code> .
/sbin	Essential system binaries.
/srv	Data for services provided by the system.
/tmp	Temporary files.
/usr	Secondary hierarchy with read-only data.
/var	Variable data such as log files

## **OS Exp.3: UNDERSTANDING THE VARIOUS PHASES OF COMPILATION OF A 'C' PROGRAM**

### **Compilation of C Program**

Step 1 : Open the terminal and edit your program using vi editor/gedit editor and save with extension ".c"

Ex. vi test.c

(or)

gedit test.c

Step 2 : Compile your program using gcc compiler

Ex. gcc test.c ⑦ Output file will be "a.out"

(or)

gcc -o test test.c ⑦ Output file will be "test"

Step 3 : Correct the errors if any and run the program

Ex. ./a.out

or

. /test

Optional Step : In order to avoid ./ prefix each time a program is to be executed, insert the following as the last line in the file **.profile**

export PATH=.:\$PATH

This Step needs only to be done once.

### **Debug C Programs using gdb debugger**

Step 1 : Compile C program with debugging option -g

Ex. gcc -g test.c

Step 2 : Launch gdb. You will get gdb prompt

Ex. gdb a.out

Step 3 : Step break points inside C program

Ex. (gdb) b 10

Break points set up at line number 10. We can have any number of break points

Step 4 : Run the program inside gdb

Ex. (gdb) r

Step 5 : Print variable to get the intermediate values of the variables at break point

Ex. (gdb) p i ⑦ Prints the value of the variable 'i'

Step 6 : Continue or stepping over the program using the following gdb commands

c ⑦ continue till the next break

n ⑦ Execute the next line. Treats function as single statement

s ⑦ Similar to 'n' but executes function statements line by line

l ⑦ List the program statements

Step 7 : Quit the debugger


(gdb) q

## OS Exp.4:

## SYSTEM ADMIN COMMANDS – BASICS

Q1. Update the package repositories.

Screenshot:

 nikhil@LAPTOP-LJDBM66Q: ~

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
```

Q2. Install the package “simplescreenrecorder”

Screenshot:

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo apt-get install simplescreenrecorder
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme at-spi2-core fontconfig gtk-update-icon-cache hicolor-icon-theme
  humanity-icon-theme i965-va-driver intel-media-va-driver libaacs0 libaom0 libatk-bridge2.0-0
  libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3
  libavcodec58 libavformat58 libavutil56 libbdplus0 libbluray2 libcairo-gobject2 libcairo2
  libchromaprint1 libcodec2-0.9 libcolor2 libcupst2 libdat1 libdouble-conversion3 libegl-mesa0
  libegl1 libepoxy0 libevdev2 libgbm1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common
  libgl1 libgl1-mesa-dri libglapi-mesa libglu1-mesa libglvnd0 libglx-mesa0 libglx0 libgme0
  libgraphite2-3 libgsm1 libgtk-3-0 libgtk-3-bin libgtk-3-common libharfbuzz0b libigdgmm11
  libinput-bin libinput10 libjack-jackd2-0 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm11
  libmp3lame0 libmpeg123-0 libmtdev1 libopenjp2-7 libopenmpt0 libopus0 libpango-1.0-0
  libpangocairo-1.0-0 libpangoft2-1.0-0 libpcre2-16-0 libpixmap-1-0 libqt5core5a libqt5dbus5
  libqt5gui5 libqt5network5 libqt5svg5 libqt5widgets5 libqt5x11extras5 librest-0.7-0 librsvg2-2
  librsvg2-common libsample1 libshine3 libsnappy1v5 libsoup-gnome2.4-1 libsoxr0 libspeex1
  libssh-gcrypt-4 libswresample3 libswscale5 libthai-data libthai0 libtheora0 libtiff5 libtwolame0
  libva-drm2 libva-x11-2 libva2 libvdpau1 libvpx6 libwacom-bin libwacom-common libwacom2 libwavpack1
  libwayland-cursor0 libwayland-egl1 libwayland-server0 libwebp6 libwebpmux3 libx264-155 libx265-179
  libxcb-icccm4 libxcb-image0 libxcb-keysyms1 libxcb-render-util0 libxcb-render0 libxcb-shm0
```

Q3. Remove the package “simplescreenrecorder”

Screenshot:

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo apt remove simplescreenrecorder
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  adwaita-icon-theme at-spi2-core fontconfig gtk-update-icon-cache hicolor-icon-theme
  humanity-icon-theme i965-va-driver intel-media-va-driver libaacs0 libaom0 libatk-bridge2.0-0
  libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3
  libavcodec58 libavformat58 libavutil56 libbdplus0 libbluray2 libcairo-gobject2 libcairo2
  libchromaprint1 libcodec2-0.9 libcolor2 libcupst2 libdat1 libdouble-conversion3 libegl-mesa0
  libegl1 libepoxy0 libevdev2 libgbm1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common
  libgl1 libgl1-mesa-dri libglapi-mesa libglu1-mesa libglvnd0 libglx-mesa0 libglx0 libgme0
  libgraphite2-3 libgsm1 libgtk-3-0 libgtk-3-bin libgtk-3-common libharfbuzz0b libigdgmm11
  libinput-bin libinput10 libjack-jackd2-0 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2
  libmp3lame0 libmpeg123-0 libmtdev1 libopenjp2-7 libopenmpt0 libopus0 libpango-1.0-0
  libpangocairo-1.0-0 libpangoft2-1.0-0 libpcre2-16-0 libpixmap-1-0 libqt5core5a libqt5dbus5
  libqt5gui5 libqt5network5 libqt5svg5 libqt5widgets5 libqt5x11extras5 librest-0.7-0 librsvg2-2
  librsvg2-common libsample1 libshine3 libsnappy1v5 libsoup-gnome2.4-1 libsoxr0 libspeex1
  libssh-gcrypt-4 libswresample3 libswscale5 libthai-data libthai0 libtheora0 libtiff5 libtwolame0
  libva-drm2 libva-x11-2 libva2 libvdpau1 libvpx6 libwacom-bin libwacom-common libwacom2 libwavpack1
  libwayland-cursor0 libwayland-egl1 libwayland-server0 libwebp6 libwebpmux3 libx264-155 libx265-179
```



**Q4.** Create a user 'elias'. Login to the newly created user and exit.

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo adduser elias
Adding user `elias' ...
Adding new group `elias' (1001) ...
Adding new user `elias' (1001) with group `elias' ...
Creating home directory `/home/elias' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for elias
Enter the new value, or press ENTER for the default
  Full Name []: Nikhil Anirudh
  Room Number []: 634
  Work Phone []: Nik
  Home Phone []: 998514....
  Other []: nil
Is the information correct? [Y/n] Y
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q5.** Disable the user 'elias', try to login and enable again.

**Screenshot:**

```
eliazer@eliazer-4250s:~$ sudo passwd -l elias
passwd: password expiry information changed.
eliazer@eliazer-4250s:~$ su - elias
Password:
su: Authentication failure
eliazer@eliazer-4250s:~$ sudo passwd -u elias
passwd: password expiry information changed.
eliazer@eliazer-4250s:~$ su - elias
Password:
elias@eliazer-4250s:~$ exit
```

**Q6.** Create a group 'cse' and add the user 'elias' in that group.

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo addgroup cse
[sudo] password for nikhil:
Adding group `cse' (GID 1002) ...
Done.
nikhil@LAPTOP-LJDBM66Q:~$ sudo adduser elias cse
Adding user `elias' to group `cse' ...
Adding user elias to group cse
Done.
nikhil@LAPTOP-LJDBM66Q:~$
```



**Q7.** List the account expiry information of the user 'elias'

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo chage -l elias
Last password change           : Mar 04, 2021
Password expires                : never
Password inactive               : never
Account expires                 : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q8.** Change the 'Number of days warning before password expires' as 5 for the user 'elias'

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo chage -l elias
Last password change           : Mar 04, 2021
Password expires                : never
Password inactive               : never
Account expires                 : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 5
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q9.** Delete the user 'elias' and then delete the group 'cse'

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q:~$ sudo deluser elias
Removing user `elias' ...
Warning: group `elias' has no more members.
Done.
nikhil@LAPTOP-LJDBM66Q:~$ sudo delgroup cse
Removing group `cse' ...
Done.
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q10.** List the partitions available in your system.

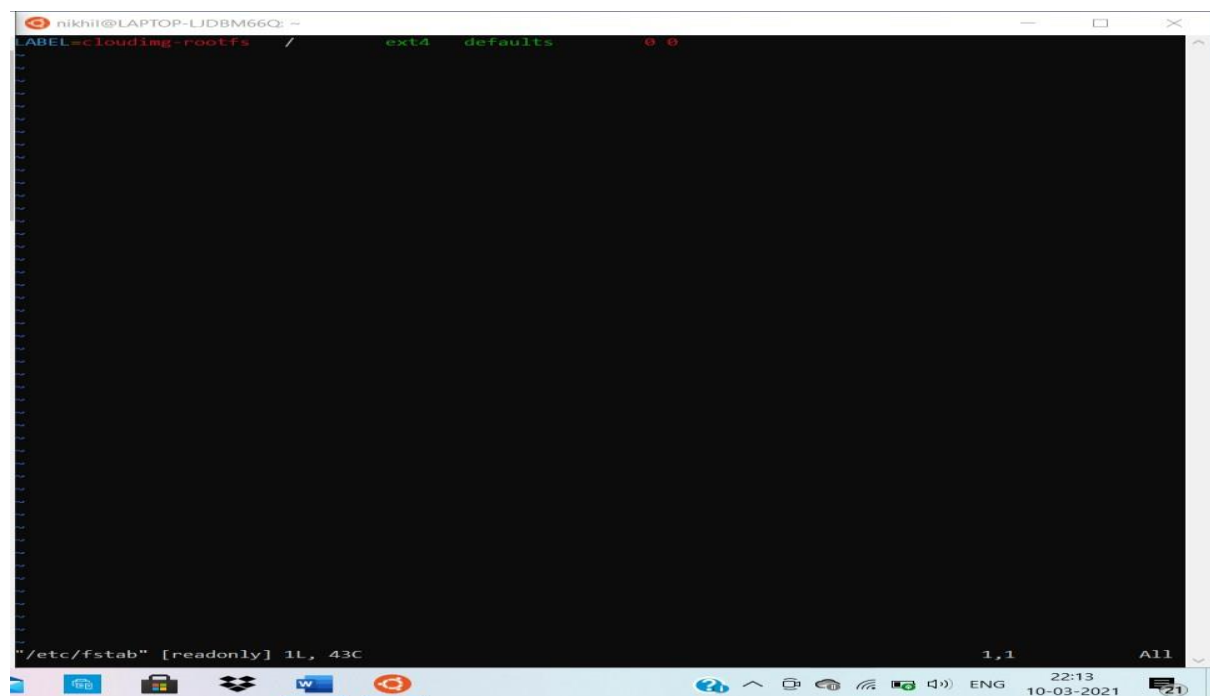
**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ sudo fdisk -l  
fdisk: cannot open /proc/partitions: No such file or directory
```

**Q11.** What are the file systems used in your system

**Screenshot:**

```
fdisk: cannot open /proc/partitions: No such file or directory  
nikhil@LAPTOP-LJDBM66Q:~$ vi /etc/fstab  
nikhil@LAPTOP-LJDBM66Q:~$
```



**Q12.** Stop the networking service and then start the service.

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ sudo /etc/init.d/networking stop  
sudo: /etc/init.d/networking: command not found  
nikhil@LAPTOP-LJDBM66Q:~$ sudo /etc/init.d/networking start  
sudo: /etc/init.d/networking: command not found  
nikhil@LAPTOP-LJDBM66Q:~$ sudo /etc/init.d/networking restart  
sudo: /etc/init.d/networking: command not found  
nikhil@LAPTOP-LJDBM66Q:~$
```

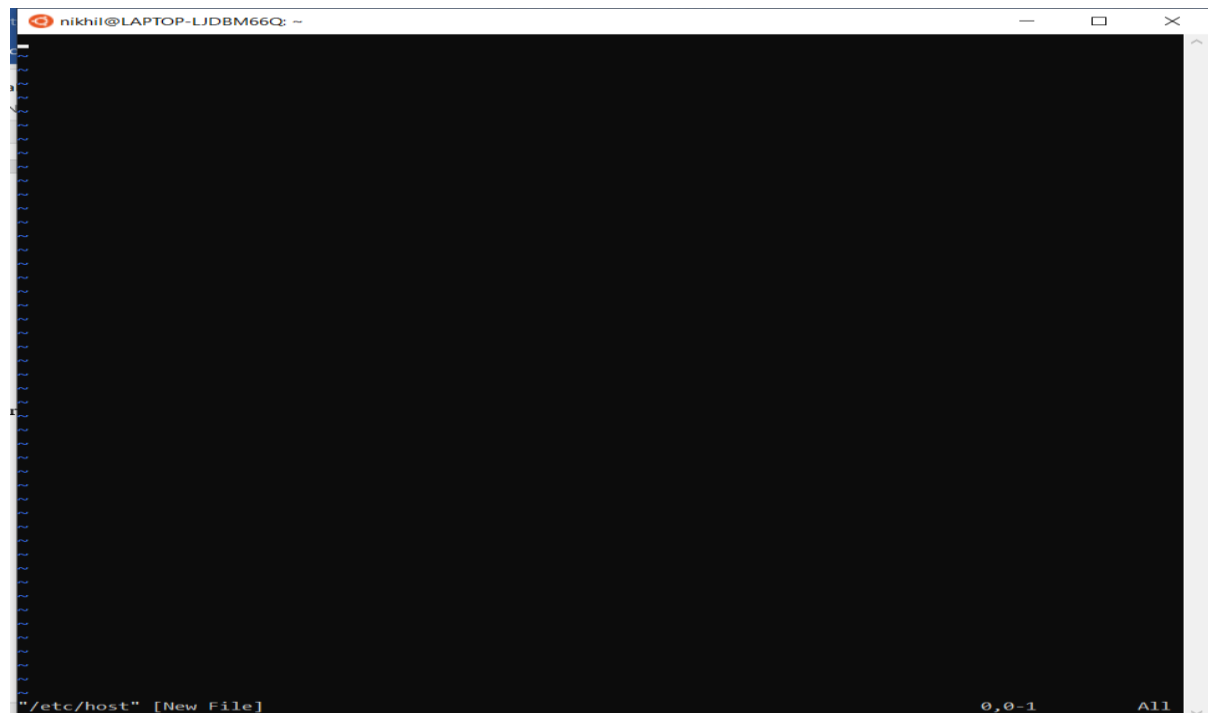
**Screenshot:**

```

nikhil@LAPTOP-LJDBM66Q: ~
nikhil@LAPTOP-LJDBM66Q:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=128 time=0.573 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=128 time=0.526 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=128 time=1.16 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=128 time=0.695 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=128 time=0.700 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=128 time=0.683 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=128 time=0.688 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=128 time=0.902 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=128 time=0.694 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=128 time=0.729 ms
64 bytes from 127.0.0.1: icmp_seq=11 ttl=128 time=0.699 ms
64 bytes from 127.0.0.1: icmp_seq=12 ttl=128 time=0.738 ms
64 bytes from 127.0.0.1: icmp_seq=13 ttl=128 time=0.854 ms
64 bytes from 127.0.0.1: icmp_seq=14 ttl=128 time=0.690 ms
64 bytes from 127.0.0.1: icmp_seq=15 ttl=128 time=0.721 ms
64 bytes from 127.0.0.1: icmp_seq=16 ttl=128 time=0.758 ms
^C
--- 127.0.0.1 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15014ms
rtt min/avg/max/mdev = 0.526/0.738/1.161/0.138 ms
nikhil@LAPTOP-LJDBM66Q:~$

```

**Screenshot:**



**Q15.** Find the IP address of the DNS Server (name server)

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following entry to /etc/wsl.conf:  
# [network]  
# generateResolvConf = false  
nameserver 192.168.0.1  
~  
~  
~
```

**Q16.** Install mysql server

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ sudo apt install mysql-server  
[sudo] password for nikhil:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
adwaita-icon-theme at-spi2-core fontconfig gtk-update-icon-cache hicolor-icon-theme  
humanity-icon-theme i965-va-driver intel-media-va-driver libaacs0 libaom0 libatk-bridge2.0-0  
libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3  
libavcodec58 libavformat58 libavutil56 libbdplus0 libbluray2 libcairo-gobject2 libcairo2  
libchromaprint1 libcodec2-0.9 libcolord2 libcups2 libdatrie1 libdouble-conversion3 libegl-mesa0  
libegl1 libepoxy0 libevdev2 libgbm1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common  
libgl1-mesa libgme0 libgraphite2-3 libgsm1 libgtk-3-0 libgtk-3-bin libgtk-3-common libharfbuzz0b  
libigdmm11 libinput-bin libinput10 libjack-jackd2-0 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2  
libmp3lame0 libmpeg123-0 libmtdev1 libopenjp2-7 libopenmpt0 libopus0 libpango-1.0-0  
libpangocairo-1.0-0 libpangoft2-1.0-0 libpcre2-16-0 libpixman-1-0 libqt5core5a libqt5dbus5  
libqt5gui5 libqt5network5 libqt5svg5 libqt5widgets5 libqt5x11extras5 librest-0.7-0 librsvg2-2  
librsvg2-common libsamplerate0 libshine3 libsnappy1v5 libsoup-gnome2.4-1 libsoxr0 libspeex1  
libssh-gcrypt-4 libswresample3 libswscale5 libthai-data libthai0 libtheora0 libtiff5 libtwolame0  
libva-drm2 libva-x11-2 libva2 libvdpau1 libvpx6 libwacom-bin libwacom-common libwacom2 libwavpack1
```

**Q17.** Restart mysql server

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ sudo /etc/init.d/mysql restart  
* Stopping MySQL database server mysqld [ OK ]  
* Starting MySQL database server mysqld  
su: warning: cannot change directory to /nonexistent: No such file or directory [ OK ]  
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q18.** Check the configuration file for mysql server

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
#  
# The MySQL database server configuration file.  
#  
# You can copy this to one of:  
# - "/etc/mysql/my.cnf" to set global options,  
# - "~/.my.cnf" to set user-specific options.  
#  
# One can use all long options that the program supports.  
# Run program with --help to get a list of available options and with  
# --print-defaults to see which it would actually understand and use.  
#  
# For explanations see  
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html  
#  
# * IMPORTANT: Additional settings that can override those from this file!  
#   The files must end with '.cnf', otherwise they'll be ignored.  
#  
!includedir /etc/mysql/conf.d/  
!includedir /etc/mysql/mysql.conf.d/  
~  
~  
~
```

**Q19.** Log on as root into mysql server

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ mysql -u root -p  
Enter password:  
ERROR 1698 (28000): Access denied for user 'root'@'localhost'  
nikhil@LAPTOP-LJDBM66Q:~$ mysql -u root -p  
Enter password:  
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
```

**Q20.** Create a new database for mysql server

**Screenshot:**

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ create database nikhil  
Command 'create' not found, did you mean:  
  command 'pcreate' from deb pbuilder-scripts (22)  
Try: sudo apt install <deb name>  
nikhil@LAPTOP-LJDBM66Q:~$
```

## OS Exp.5:

## SYSTEM ADMIN COMMANDS – SIMPLE TASK AUTOMATIONS

Q1. Schedule a task to display the following message on the monitor for every 2 minutes.

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ tty  
/dev/tty1  
nikhil@LAPTOP-LJDBM66Q:~$ crontab -e  
no crontab for nikhil - using an empty one  
crontab: installing new crontab  
nikhil@LAPTOP-LJDBM66Q:~$ systemctl start cron
```

Q2. Schedule a task to take backup of your important file (say file f1) for every 30 minutes

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow command  
*/2 * * * * echo welcome >> /dev/tty1  
nikhil@LAPTOP-LJDBM66Q:~$
```

Q3. Schedule a task to take backup of login information everyday 9:30am

```
nikhil@LAPTOP-LJDBM66Q: ~  
GNU nano 4.8 /tmp/crontab.5BtswH/crontab  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow command  
30 9 * * * cp /etc/passwd passwdback
```

- **Basic Linux Commands:**

 nikhil@LAPTOP-LJDBM66Q: ~

To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo\_root" for details.

nikhil@LAPTOP-LJDBM66Q:~\$ echo SRM University  
SRM University.

nikhil@LAPTOP-LJDBM66Q:~\$ date

Thu Feb 4 22:46:15 IST 2021

nikhil@LAPTOP-LJDBM66Q:~\$ cal 2021

```

                                2021
    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           1 2 3 4 5 6           1 2 3 4 5 6
 3 4 5 6 7 8 9       7 8 9 10 11 12 13       7 8 9 10 11 12 13
10 11 12 13 14 15 16 14 15 16 17 18 19 20 14 15 16 17 18 19 20
17 18 19 20 21 22 23 21 22 23 24 25 26 27 21 22 23 24 25 26 27
24 25 26 27 28 29 30 28                        28 29 30 31
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 2 3           1                        1 2 3 4 5
 4 5 6 7 8 9 10       2 3 4 5 6 7 8       6 7 8 9 10 11 12
11 12 13 14 15 16 17  9 10 11 12 13 14 15 13 14 15 16 17 18 19
18 19 20 21 22 23 24 16 17 18 19 20 21 22 20 21 22 23 24 25 26
25 26 27 28 29 30     23 24 25 26 27 28 29 27 28 29 30
                        30 31
```

nikhil@LAPTOP-LJDBM66Q:~\$ passwd

Changing password for nikhil.

Current password:

New password:

Retype new password:

passwd: password updated successfully



- **Working with Files:**

```

nikhil@LAPTOP-LJDBM66Q:~$ ls
f3 nikhil
nikhil@LAPTOP-LJDBM66Q:~$ ls -l
total 0
-rw-r--r-- 1 nikhil nikhil 60 Feb  4 11:50 f3
drwxr-xr-x 1 nikhil nikhil 4096 Feb  4 11:55 nikhil
nikhil@LAPTOP-LJDBM66Q:~$ ls -a
. . . .bash_history .bash_logout .bashrc .landscape .motd_shown .profile f3 nikhil
nikhil@LAPTOP-LJDBM66Q:~$ cat f3
Hello World!
This is Nikhil
I'm studying in SRM University.
nikhil@LAPTOP-LJDBM66Q:~$ cat > nick
Aa Bb Cc Dd Ee Ff
Gg Hh Ii Jj Kk Ll
Jj Kk Ll Mm Nn Oo
Pp Qq Rr Ss Tt Uu
Vv Ww Xx Yy Zz
1 2 3 4 5 6 7
8 9 10

```

```

nikhil@LAPTOP-LJDBM66Q: ~
Aa Bb Cc Dd Ee Ff
Gg Hh Ii Jj Kk Ll
Jj Kk Ll Mm Nn Oo
Pp Qq Rr Ss Tt Uu
Vv Ww Xx Yy Zz
1 2 3 4 5 6 7
8 9 10
nikhil@LAPTOP-LJDBM66Q:~$ ls
f3 nick nikhil
nikhil@LAPTOP-LJDBM66Q:~$ wc nick
 7 39 108 nick
nikhil@LAPTOP-LJDBM66Q:~$ wc -l nick
7 nick
nikhil@LAPTOP-LJDBM66Q:~$ wc -w nick
39 nick
nikhil@LAPTOP-LJDBM66Q:~$ wc -c nick
108 nick
nikhil@LAPTOP-LJDBM66Q:~$ cp nick J1
nikhil@LAPTOP-LJDBM66Q:~$ mv J1 anirudh
nikhil@LAPTOP-LJDBM66Q:~$ rm f3
nikhil@LAPTOP-LJDBM66Q:~$ ls
anirudh nick nikhil
nikhil@LAPTOP-LJDBM66Q:~$ head -2 nick
Aa Bb Cc Dd Ee Ff
Gg Hh Ii Jj Kk Ll
nikhil@LAPTOP-LJDBM66Q:~$ tail -3 nick
Vv Ww Xx Yy Zz
1 2 3 4 5 6 7
8 9 10

```

- **Working with directories**

```

nikhil@LAPTOP-LJDBM66Q:~$ mkdir nikhil anirudh
mkdir: cannot create directory 'nikhil': File exists
mkdir: cannot create directory 'anirudh': File exists
nikhil@LAPTOP-LJDBM66Q:~$ mkdir SRM University
nikhil@LAPTOP-LJDBM66Q:~$ ls
SRM University anirudh nick nikhil
nikhil@LAPTOP-LJDBM66Q:~$ cd anirudh
-bash: cd: anirudh: Not a directory
nikhil@LAPTOP-LJDBM66Q:~$ cd SRM University
-bash: cd: too many arguments
nikhil@LAPTOP-LJDBM66Q:~$ cd
nikhil@LAPTOP-LJDBM66Q:~$ cd ..
nikhil@LAPTOP-LJDBM66Q:/home$ cd -
/home/nikhil
nikhil@LAPTOP-LJDBM66Q:~$ pwd
/home/nikhil
nikhil@LAPTOP-LJDBM66Q:~$ cd/
-bash: cd/: No such file or directory

```

- **File name substitution:**

```

nikhil@LAPTOP-LJDBM66Q:~$ cat > J2
qwert
yuiop
asdfg
hjkl
zxcvb
mnbvcv
nikhil@LAPTOP-LJDBM66Q:~$ lf
lf: command not found
nikhil@LAPTOP-LJDBM66Q:~$ ls
J2 SRM University anirudh nick nikhil
nikhil@LAPTOP-LJDBM66Q:~$ ls f?
ls: cannot access 'f?': No such file or directory
nikhil@LAPTOP-LJDBM66Q:~$ cat > f1
kjnnkjsdndjnzn
lkjxdjgkj^[s^[j^[g^[j
nikhil@LAPTOP-LJDBM66Q:~$ cat > f2
kjkjnkjh
568645
jknkjin58485
nikhil@LAPTOP-LJDBM66Q:~$ ls
J2 SRM University anirudh f1 f2 nick nikhil
nikhil@LAPTOP-LJDBM66Q:~$ ls f?
f1 f2
nikhil@LAPTOP-LJDBM66Q:~$ ls [sua]run
ls: cannot access '[sua]run': No such file or directory
nikhil@LAPTOP-LJDBM66Q:~$ ls [s]
ls: cannot access '[s]': No such file or directory
nikhil@LAPTOP-LJDBM66Q:~$ wc < nick
 7 39 108
nikhil@LAPTOP-LJDBM66Q:~$ who > anirudh
nikhil@LAPTOP-LJDBM66Q:~$ ls anirudh
anirudh
nikhil@LAPTOP-LJDBM66Q:~$ ls
J2 SRM University anirudh f1 f2 nick nikhil
nikhil@LAPTOP-LJDBM66Q:~$ cat >> f2

```

- I/O Redirection:

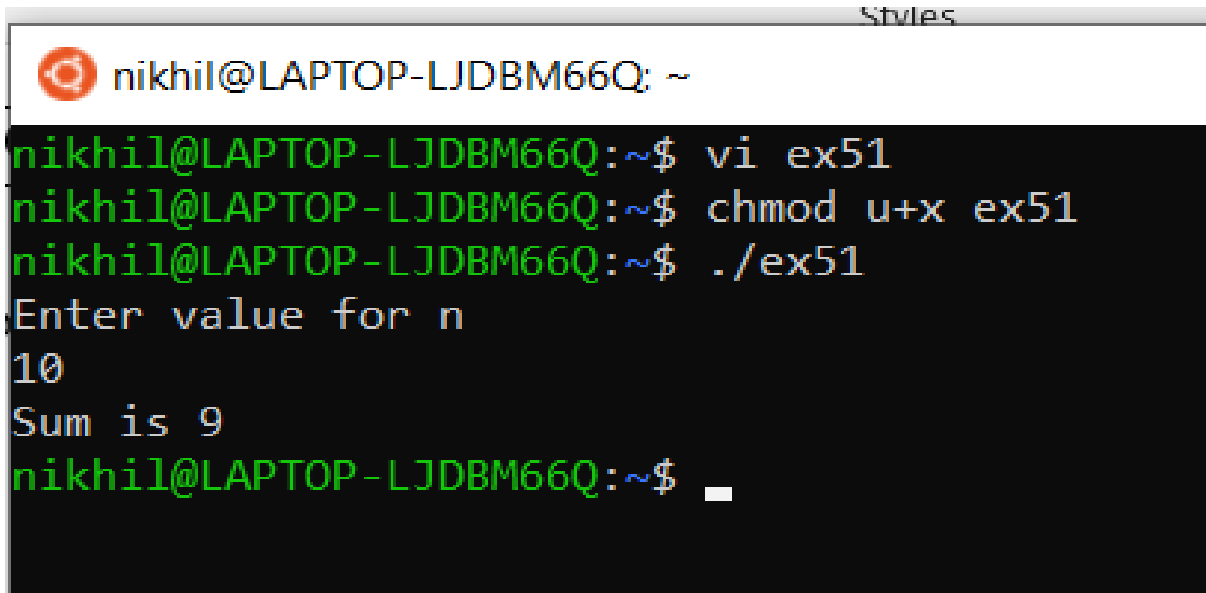
```
nikhil@LAPTOP-LJDBM66Q:~$ cat >> f2
hello
nikhil@LAPTOP-LJDBM66Q:~$ head -3 nikhil
head: error reading 'nikhil': Is a directory
nikhil@LAPTOP-LJDBM66Q:~$ head -3 anirudh
nikhil@LAPTOP-LJDBM66Q:~$ head -3 J2
qwert
yuiop
asdfg
nikhil@LAPTOP-LJDBM66Q:~$ head -3 J2|tail -2
yuiop
asdfg
nikhil@LAPTOP-LJDBM66Q:~$
```

## OS Exp.7: SHELL PROGRAMS – BASIC LEVEL

**Q1.** Find the output of the following shell scripts

```
echo Enter value for n
read n
sum=0
i=1
while [ $i -le $n ]
do
    sum=$((sum+i))
    i=$((i+2))
done
echo Sum is $sum
```

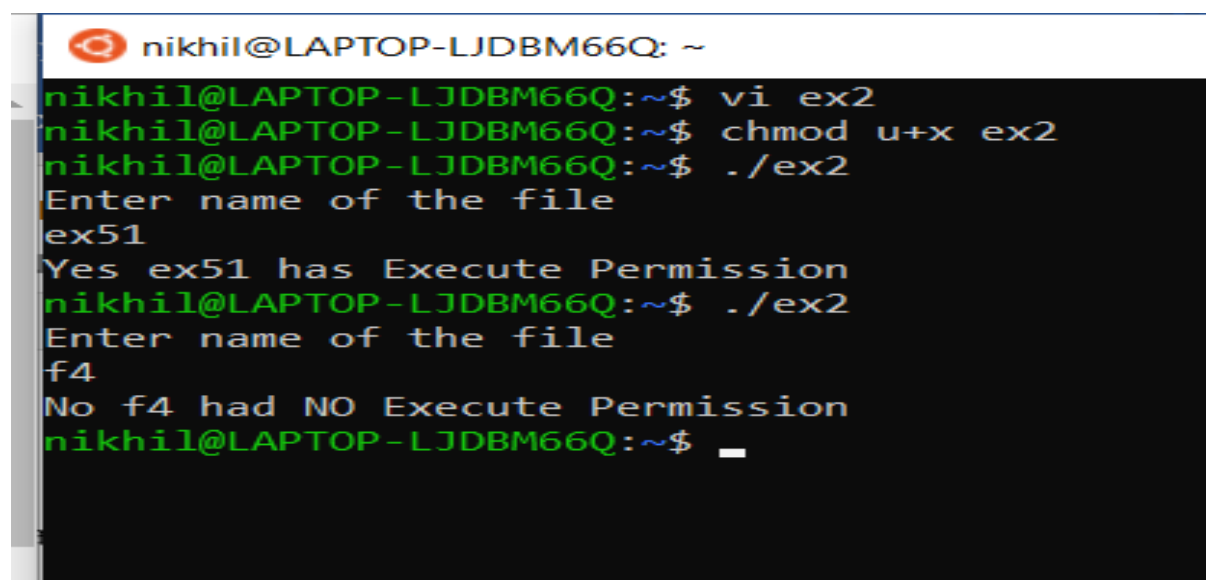
**Output :**



```
nikhil@LAPTOP-LJDBM66Q: ~
nikhil@LAPTOP-LJDBM66Q:~$ vi ex51
nikhil@LAPTOP-LJDBM66Q:~$ chmod u+x ex51
nikhil@LAPTOP-LJDBM66Q:~$ ./ex51
Enter value for n
10
Sum is 9
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q2.** Write a program to check whether the file has execute permission or not. If not, add the permission.

**Output:**



```
nikhil@LAPTOP-LJDBM66Q: ~
nikhil@LAPTOP-LJDBM66Q:~$ vi ex2
nikhil@LAPTOP-LJDBM66Q:~$ chmod u+x ex2
nikhil@LAPTOP-LJDBM66Q:~$ ./ex2
Enter name of the file
ex51
Yes ex51 has Execute Permission
nikhil@LAPTOP-LJDBM66Q:~$ ./ex2
Enter name of the file
f4
No f4 had NO Execute Permission
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q3.** Write a shell script to print a greeting as specified below.

If hour is greater than or equal to 0 (midnight) and less than or equal to 11 (up to 11:59:59), "Good morning" is displayed.

If hour is greater than or equal to 12 (noon) and less than or equal to 17 (up to 5:59:59 p.m.), "Good afternoon" is displayed.

If neither of the preceding two conditions is satisfied, "Good evening" is displayed.

```
hour=$(date | cut -c12-13)
```

```
if [ "$hour" -ge 0 -a "$hour" -le 11 ]
```

```
then
```

← complete the program

```
nikhil@LAPTOP-LJDBM66Q:~$ vi ex3
nikhil@LAPTOP-LJDBM66Q:~$ chmod u+x ex3
nikhil@LAPTOP-LJDBM66Q:~$ ./ex3
cut: 13: No such file or directory
./ex3: line 2: [: : integer expression expected
./ex3: line 5: [: -le: unary operator expected
Good Evening
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q4.** Write a shell script to list only the name of sub directories in the present working directory.

```
nikhil@LAPTOP-LJDBM66Q: ~
nikhil@LAPTOP-LJDBM66Q:~$ vi ex4
nikhil@LAPTOP-LJDBM66Q:~$ chmod u+x ex4
nikhil@LAPTOP-LJDBM66Q:~$ ./ex4
SRM
University
nikhil
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q5.** Write a program to check all the files in the present working directory for a pattern (passed through command line) and display the name of the file followed by a message stating that the pattern is available or not available.

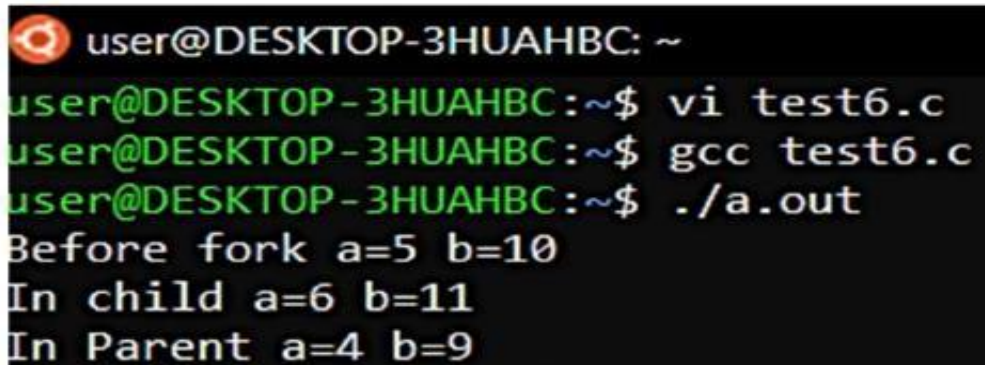
```
./ex55: line 2: 10
nikhil@LAPTOP-LJDBM66Q:~$ vi ex55
nikhil@LAPTOP-LJDBM66Q:~$ chmod u+x ex55
nikhil@LAPTOP-LJDBM66Q:~$ ./ex55 SRM
J2 not found
a.out found
anirudh not found
ex2 not found
ex3 not found
ex4 not found
ex51 not found
ex52 not found
ex55 not found
exp3-ques1 not found
exp3-ques1.c not found
exp3-ques3 found
exp3-ques3.c found
f1 not found
f2 not found
nick not found
test5.c not found
nikhil@LAPTOP-LJDBM66Q:~$
```



## OS Exp.8: PROCESS CREATION

**Q1. Find the output of the following program**

```
#include <stdio.h>
#include<unistd.h>
int main()
{
    int a=5,b=10,pid;
    printf("Before fork a=%d b=%d \n",a,b);
    pid=fork();
    if(pid==0)
    {
        a=a+1;
        b=b+1;
        printf("In child a=%d b=%d \n",a,b);
    }
    else
    {
        sleep(1);
        a=a-1;
        b=b-1;
        printf("In Parent a=%d b=%d \n",a,b);
    }
    return 0;
}
```



```
user@DESKTOP-3HUAHBC: ~
user@DESKTOP-3HUAHBC:~$ vi test6.c
user@DESKTOP-3HUAHBC:~$ gcc test6.c
user@DESKTOP-3HUAHBC:~$ ./a.out
Before fork a=5 b=10
In child a=6 b=11
In Parent a=4 b=9
```

**Q2. Rewrite the program in Q1 using vfork() and write the output**

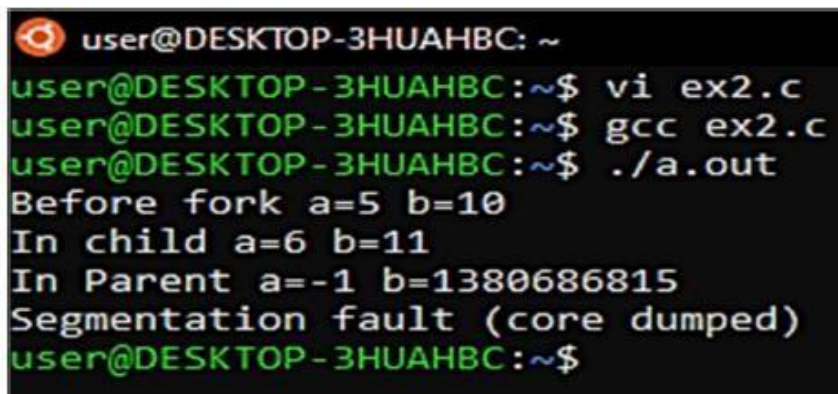
```
#include<stdio.h>
#include<unistd.h>
int main()
{
    int a=5,b=10,pid;
    printf("Before fork a=%d b=%d \n",a,b);
    pid=vfork();
    if(pid==0)
```



```

{
a=a+1;
b=b+1;
printf("In child a=%d b=%d \n",a,b);
}
else
{
sleep(1);
a=a-1;
b=b-1;
printf("In Parent a=%d b=%d \n",a,b);
}
return 0;
}

```



```

user@DESKTOP-3HUAHBC: ~
user@DESKTOP-3HUAHBC:~$ vi ex2.c
user@DESKTOP-3HUAHBC:~$ gcc ex2.c
user@DESKTOP-3HUAHBC:~$ ./a.out
Before fork a=5 b=10
In child a=6 b=11
In Parent a=-1 b=1380686815
Segmentation fault (core dumped)
user@DESKTOP-3HUAHBC:~$

```

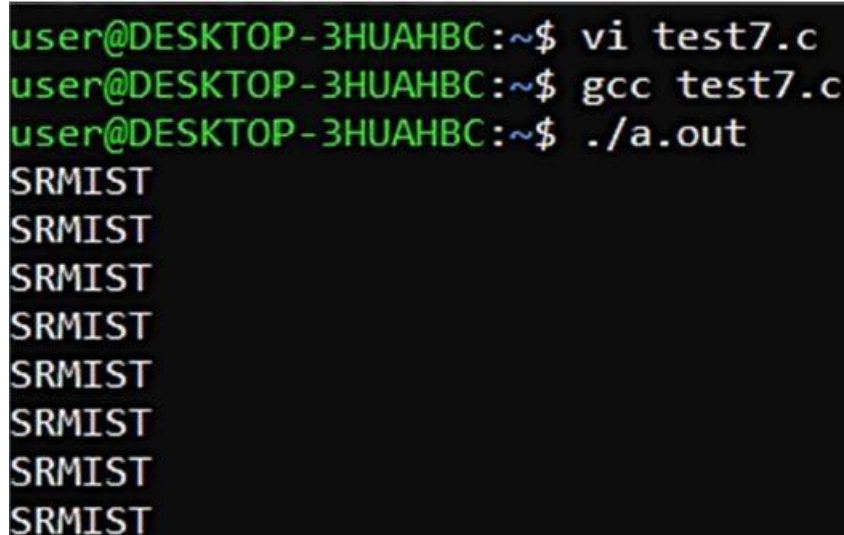
**Q3. Calculate the number of times the text "SRMIST" is printed.**

```

#include <stdio.h>
#include<unistd.h>
int main()
{
}

```

Output :



```

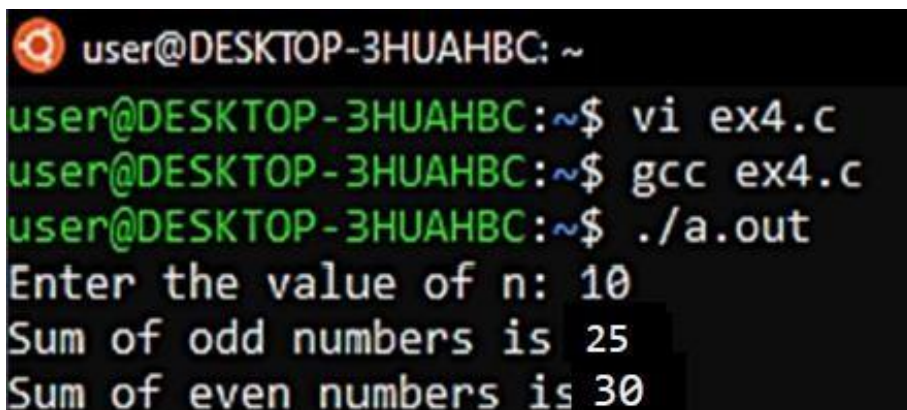
user@DESKTOP-3HUAHBC:~$ vi test7.c
user@DESKTOP-3HUAHBC:~$ gcc test7.c
user@DESKTOP-3HUAHBC:~$ ./a.out
SRMIST
SRMIST
SRMIST
SRMIST
SRMIST
SRMIST
SRMIST
SRMIST

```

**Q4. Complete the following program as described below :**

The child process calculates the sum of odd numbers and the parent process calculate the sum of even numbers up to the number 'n'. Ensure the Parent process waits for the child process to finish.

```
#include<stdio.h>
#include<unistd.h>
int main()
{
    int oddsum=0,evensum=0,i,n,pid;
    printf("Enter the value of n : ");
    scanf("%d",&n);
    pid=fork();
    if(pid==0)
    {
        for(i=1;i<=n;i=i+2)
            oddsum+=i;
        printf("Sum of odd numbers is %d\n",oddsum);
    }
    else
    {
        sleep(1);
        for(i=0;i<=n;i=i+2)
            evensum+=1;
        printf("Sum of even numbers is %d\n",evensum);
    }
    return 0;
}
```



```
user@DESKTOP-3HUAHBC: ~
user@DESKTOP-3HUAHBC:~$ vi ex4.c
user@DESKTOP-3HUAHBC:~$ gcc ex4.c
user@DESKTOP-3HUAHBC:~$ ./a.out
Enter the value of n: 10
Sum of odd numbers is 25
Sum of even numbers is 30
```

**Q5. How many child processes are created for the following code?**

**Hint : Check with small values of 'n'.**

```
for (i=0; i<n; i++)
    fork();
```

Output :

No. of processes = n

No. of times fork() is called = n times

No. of processes= 2n

No. of child processes = 2n-1

**Q6. Write a program to print the Child process ID and Parent process ID in both Child and Parent processes**

```
#include <stdio.h>
#include<unistd.h>
int main()
{
    int pid;
    pid=fork();
    if(pid==0)
    {
        printf("In Child.....\nChild ID is %d Parent ID is %d\n",getpid(),getppid());
    }
    else
    {
        sleep(1);
        printf("In Parent.....\nChild ID is %d Parent ID is %d\n",pid,getpid());
    }
    return 0;
}
```

```
user@DESKTOP-3HUAHBC: ~  
user@DESKTOP-3HUAHBC:~$ vi ex6.c  
user@DESKTOP-3HUAHBC:~$ gcc ex6.c  
user@DESKTOP-3HUAHBC:~$ ./a.out  
In Child....  
Child ID is 128 Parent ID is 127  
In Parent....  
Child ID is 128 Parent ID is 127  
user@DESKTOP-3HUAHBC:~$ _
```

**Q7. How many child processes are created for the following code?**

```
#include<stdio.h>
#include<unistd.h>
int main()
{
}
Output :
Verified by
fork();
fork()&&fork() || fork();
fork();
printf("Yes ");
return 0;
}
```

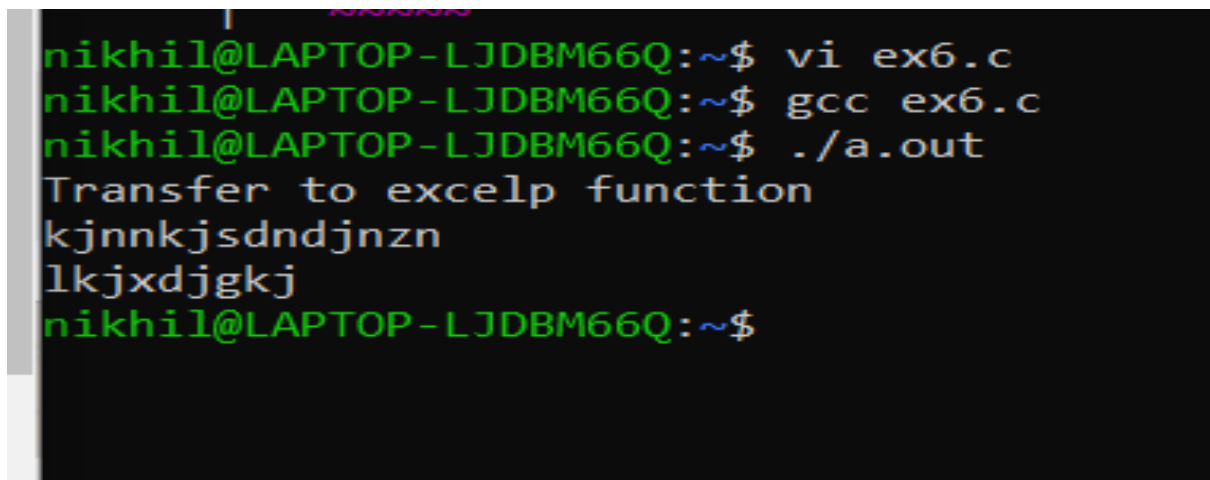
```
user@DESKTOP-3HUAHBC: ~  
user@DESKTOP-3HUAHBC:~$ vi ex.c  
user@DESKTOP-3HUAHBC:~$ gcc ex.c  
user@DESKTOP-3HUAHBC:~$ ./a.out  
YesYesYesYesYesYesYesYesYesYesYesYesYesYesYesYesYesYesYesYes
```

## OS Exp.9: OVERLAY CONCEPT

**Q1.** Execute the Following Program and write the

```
#include <stdio.h>
#include<unistd.h>
int main()
{
    printf("Transfer to execlp function
    \n"); execlp("head", "head", "-
    2", "f1", NULL); printf("This line
    will not execute \n"); return 0;
}
```

**Output :**

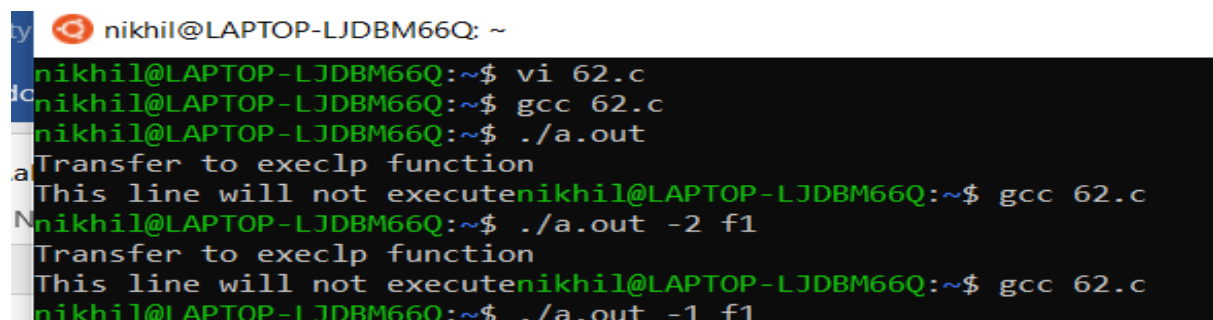
A terminal window with a black background and green text. The user 'nikhil' is at a laptop named 'LJDBM66Q'. They run 'vi ex6.c', then 'gcc ex6.c', and finally './a.out'. The output of the program is 'Transfer to execlp function' followed by two lines of random characters 'kjnnkjsdndjzn' and 'lkjxdjgkj', and then the prompt returns.

```
nikhil@LAPTOP-LJDBM66Q:~$ vi ex6.c
nikhil@LAPTOP-LJDBM66Q:~$ gcc ex6.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out
Transfer to execlp function
kjnnkjsdndjzn
lkjxdjgkj
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q2.** Rewrite question Q1 with execl() function. Pass the 3<sup>rd</sup> and 4<sup>th</sup> argument of the function execl() through command line arguments.

**Input :** ./a.out -3 f1

**Output :**

A terminal window showing the user 'nikhil' at 'LJDBM66Q'. They run 'vi 62.c', 'gcc 62.c', and './a.out'. The output is 'Transfer to execlp function' followed by 'This line will not execute'. Then they run './a.out -2 f1' and './a.out -1 f1', both of which produce the same output.

```
nikhil@LAPTOP-LJDBM66Q: ~
nikhil@LAPTOP-LJDBM66Q:~$ vi 62.c
nikhil@LAPTOP-LJDBM66Q:~$ gcc 62.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out
Transfer to execlp function
This line will not execute
nikhil@LAPTOP-LJDBM66Q:~$ gcc 62.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out -2 f1
Transfer to execlp function
This line will not execute
nikhil@LAPTOP-LJDBM66Q:~$ gcc 62.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out -1 f1
Transfer to execlp function
This line will not execute
```

Q3. Rewrite question Q1 with `execv()` function.

\$vi ex63.c

```
nikhil@LAPTOP-LJDBM66Q: ~  
#include<stdio.h>  
#include<unistd.h>  
int main()  
{  
    char *args[]={"head", "-2", "f1", NULL};  
    printf("Transfer to execlp function\n");  
    execv("/usr/bin/head",args);  
    printf("This line will not execute\n");  
    return 0;  
}
```

Output :

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ vi ex3.c  
nikhil@LAPTOP-LJDBM66Q:~$ gcc ex3.c  
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out  
Transfer to execlp function  
kjnnkjsdndjnzn  
lkjxdjgkj  
nikhil@LAPTOP-LJDBM66Q:~$
```

Q4. Rewrite question Q1 with `execvp()` function.

Output :

```
nikhil@LAPTOP-LJDBM66Q: ~  
#include<stdio.h>  
#include<unistd.h>  
int main()  
{  
    char *args[]={"head", "-2", "f1", NULL};  
    printf("Transfer to execlp function\n");  
    execvp("head", args);  
    printf("This line will not execute\n");  
    return 0;  
}
```

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ vi ex4.c  
nikhil@LAPTOP-LJDBM66Q:~$ vi ex4.c  
nikhil@LAPTOP-LJDBM66Q:~$ gcc ex4.c  
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out  
Transfer to execlp function  
kjnnkjsdndjnzn  
lkjxdjgkj  
nikhil@LAPTOP-LJDBM66Q:~$
```

Q5. a..

b. Write a program (ex652.c) to find the sum of numbers from 1 to n.

c. Write a program (ex653.c) to execute the program ex651.c in child process and the program ex652.c in parent process

*Note : Pass the values through command line arguments for the above programs*

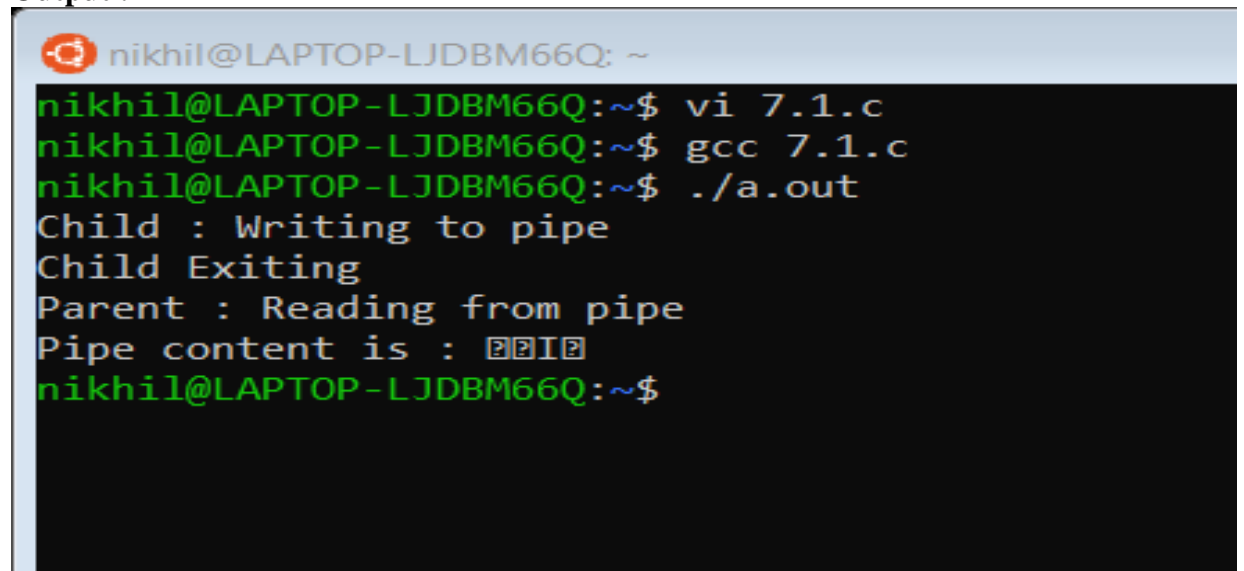
```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ vi ex651.c  
nikhil@LAPTOP-LJDBM66Q:~$ gcc ex651.c -o ex651  
nikhil@LAPTOP-LJDBM66Q:~$ ex651 5  
ex651: command not found  
nikhil@LAPTOP-LJDBM66Q:~$
```

## OS Exp.10: IPC USING PIPES

**Q1.** Write the output of the following program

```
#include <stdio.h>
#include<unistd.h>
#include<sys/wait.h>
int main()
{
    int p[2];
    char buff[25];
    if(fork()==0)
    {
        printf("Child : Writing to pipe
        \n"); write(p[1],"Welcome",8);
        printf("Child Exiting\n");
    }
    else
    {
        wait(NULL);
        printf("Parent : Reading from pipe
        \n"); read(p[1],buff,8);
        printf("Pipe content is : %s \n",buff);
    }
    return 0;
}
```

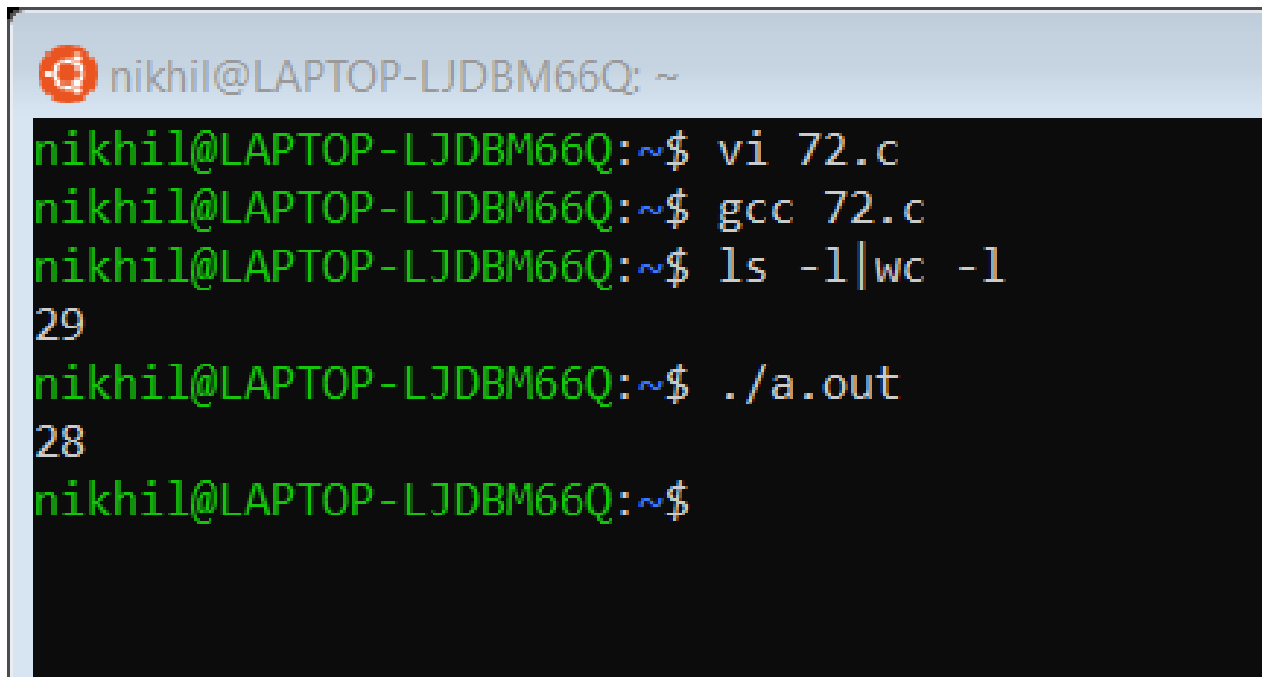
**Output :**



```
nikhil@LAPTOP-LJDBM66Q: ~
nikhil@LAPTOP-LJDBM66Q:~$ vi 7.1.c
nikhil@LAPTOP-LJDBM66Q:~$ gcc 7.1.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out
Child : Writing to pipe
Child Exiting
Parent : Reading from pipe
Pipe content is : Welcome
nikhil@LAPTOP-LJDBM66Q:~$
```



**Q2.** Write a program to implement the following command line pipe using pipe() and dup()

A terminal window with a light blue header bar showing the prompt 'nikhil@LAPTOP-LJDBM66Q: ~'. The terminal has a black background with green text. The user enters 'vi 72.c', then 'gcc 72.c', then 'ls -l|wc -l'. The output '29' is shown. Then the user enters './a.out', and the output '28' is shown. The prompt returns.

```
nikhil@LAPTOP-LJDBM66Q: ~  
nikhil@LAPTOP-LJDBM66Q:~$ vi 72.c  
nikhil@LAPTOP-LJDBM66Q:~$ gcc 72.c  
nikhil@LAPTOP-LJDBM66Q:~$ ls -l|wc -l  
29  
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out  
28  
nikhil@LAPTOP-LJDBM66Q:~$
```

**Q3.** Write the output of the following program

```
#include<fcntl.h>  
#include<sys/types.h>  
#include<sys/stat.h>  
#include<unistd.h>  
  
int main()  
{  
    char buff[25];  
    int rfd,wfd;  
  
    mkfifo("fif1",O_CREAT|0644);  
  
    if (fork()==0)  
    {  
        printf("Child writing into  
        FIFO\n");  
        wfd=open("fif1",O_WRONLY);  
        write(wfd,"Hello",6);  
    }  
    else  
    {  
        rfd=open("fif1",O_RDONLY);  
        read(rfd,buff,6);  
        printf("Parent reads from FIFO : %s\n",buff);  
    }  
}
```

```
        return 0;
    }
```

## Output :

```
nikhil@LAPTOP-LJDBM66Q: ~
nikhil@LAPTOP-LJDBM66Q:~$ gcc 73.c
73.c:1:9: fatal error: fcntl.h: No such file or directory
   1 | #include<fcntl.h>
     |           ^~~~~~
compilation terminated.
nikhil@LAPTOP-LJDBM66Q:~$ vi 73.c
nikhil@LAPTOP-LJDBM66Q:~$ gcc 73.c
73.c: In function 'main':
73.c:10:16: error: 'O_CREAT' undeclared (first use in this function); did you mean 'O_CREAT'?
   10 |     mkfifo("fif1", O_CREAT|0644);
       |                   ^~~~~~
       |                   O_CREAT
73.c:10:16: note: each undeclared identifier is reported only once for each function it appears in
73.c:13:3: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
   13 |     printf("Child : Writing into FIFO\n");
       |     ^~~~~~
73.c:13:3: warning: incompatible implicit declaration of built-in function 'printf'
73.c:5:1: note: include '<stdio.h>' or provide a declaration of 'printf'
   4 | #include<unistd.h>
+++ |+#include <stdio.h>
   5 | int main()
73.c:21:3: warning: incompatible implicit declaration of built-in function 'printf'
   21 |     printf("Parent reads from FIFO: %s\n", buff);
       |     ^~~~~~
73.c:21:3: note: include '<stdio.h>' or provide a declaration of 'printf'
nikhil@LAPTOP-LJDBM66Q:~$ vi 73.c
nikhil@LAPTOP-LJDBM66Q:~$ gcc 73.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out
Child : Writing into FIFO
```

## OS Exp.11: IPC Using Shared Memory And Message Queues

**Q1.** Write a program to send a message (pass through command line arguments) into a message queue. Send few messages with unique message numbers

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<sys/ipc.h>
#include<sys/msg.h>
int main(int argc,char*argv[])
{
    int len,mid,i=1;
    struct buffer
    {
        long mtype;
        char buf[50];
    } x;
    mid=msgget((key_t))6,IPC_CREAT|0666);
    x.mtype=atoi(argv[1]);
    strcpy(x.buf,argv[2]);
    len=strlen(x.buf);
    msgsnd(mid,&x,len,0);
    printf("Message of size %d sent successfully \n",len);
    return 0;
}
```

```
~$ vi ex81.c
~$ gcc ex81.c
~$ a.out 1 welcome
d
~$ ./a.out 1 welcome
successfully
~$ ./a.out 2 SRM University
successfully
~$
```

**Q2.** Write a program to receive a particular message from the message queue. Use message number to receive the particular message

```
nikhil@LAPTOP-LJDBM66Q:~$ vi 82.c
nikhil@LAPTOP-LJDBM66Q:~$ gcc 82.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out 2 10
The message is : 10
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out 2 12
The message is : ge12
nikhil@LAPTOP-LJDBM66Q:~$
```

```

#include<stdio.h>
#include<sys/ipc.h>
#include<sys/shm.h>
int main()
{
    int shmid;
    char *str;

    shmid=shmget((key_t)7,1024,IPC_CREAT|0666);
    str=(char *)shmat(shmid,(char *)0,0);
    printf("Enter data: ");
    fgets(str,sizeof(str),stdin);
    printf("Data successfully written into memory \n");
    shmdt(str);
    return 0;
}

msgrcv(mid,&x,len,x.mtype,0);
printf("The message is : %s\n", x.buf);
return 0;
}

```

Q3. Write a program to do the following:

- Σ Create two processes, one is for writing into the shared memory (shm\_write.c) and another is for reading from the shared memory (shm\_read.c)
- Σ In the shared memory, the writing process, creates a shared memory of size 1K (and flags) and attaches the shared memory
- Σ The write process writes the data read from the standard input into the shared memory. Last byte signifies the end of buffer
- Σ Read process would read from the shared memory and write to the standard

```

#include<stdio.h>
#include<sys/ipc.h>
#include<sys/shm.h>
int main()
{
    int shmid;
    char *str;

    shmid=shmget((key_t)7,1024,IPC_CREAT|0666);
    str=(char *)shmat(shmid,(char *)0,0);
    printf("Data read from memory : %s \n", str);
    shmdt(str);
    shmctl(shmid,IPC_RMID,NULL);
    return 0;
}

```

```
nikhil@LAPTOP-LJDBM66Q:~$ gcc shm_write.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out
Enter data: Welcome
Data successfully written into memory
nikhil@LAPTOP-LJDBM66Q:~$ gcc shm_read.c
nikhil@LAPTOP-LJDBM66Q:~$ ./a.out
Data read from memory : Welcome
nikhil@LAPTOP-LJDBM66Q:~$
```

## OS Exp.12: PROCESS SYNCHRONIZATION

Q1. Execute and write the output of the following program for mutual exclusion.

Output :

```
user@DESKTOP-3HUAHBC: ~  
#include<sys/ipc.h>  
#include<sys/sem.h>  
#include<sys/types.h>  
#include<unistd.h>  
#include<stdio.h>  
int main()  
{  
    int pid,semid,val;  
    struct sembuf sop;  
    semid=semget((key_t)6,1,IPC_CREAT|0666);  
    pid=fork();  
    sop.sem_num=0;  
    sop.sem_op=0;  
    sop.sem_flg=0;  
    if(pid!=0)  
    {  
        sleep(1);  
        printf("The Parent waits for WAIT signal\n");  
        semop(semid,&sop,1);  
        printf("The Parent WAKED UP & doing her job\n");  
        sleep(10);  
        printf("Parent Over\n");  
    }  
    else  
    {  
        printf("The Child sets WAIT signal & doing her job\n");  
        semctl(semid,0,SETVAL,1);  
        sleep(10);  
        printf("The Child sets WAKE signal & finished her job\n");  
        semctl(semid,0,SETVAL,0);  
        printf("Child Over\n");  
    }  
    return 0;  
}
```

```
user@DESKTOP-3HUAHBC: ~  
user@DESKTOP-3HUAHBC:~$ vi Exp9.c  
user@DESKTOP-3HUAHBC:~$ gcc Exp9.c  
user@DESKTOP-3HUAHBC:~$ ./a.out  
The Child sets WAIT signal & doing her job  
The Parent waits for WAIT signal  
The Child sets WAKE signal & finished her job  
Child Over  
The Parent WAKED UP & doing her job  
Parent Over
```

```
user@DESKTOP-3HUAHBC: ~  
#include<stdio.h>  
#include<stdlib.h>  
int mutex=1,full=0,empty=3,x=0;  
int main()  
{  
    int n;  
    void producer();  
    void consumer();  
    int wait(int);  
    int signal(int);  
    printf("\n1.Producer\n2.Consumer\n3.Exit");  
    while(1)  
    {  
        printf("Enter your Choice: ");  
        scanf("%d",&n);  
        switch(n)  
        {  
            case 1:  
                if((mutex==1)&&(empty!=0))  
                    producer();  
                else  
                    printf("Buffer is full!!\n");  
                break;  
            case 2:  
                if((mutex==1)&&(full!=0))  
                    consumer();  
                else  
                    printf("Buffer is empty!!\n");  
                break;  
            case 3:  
                exit(0);  
                break;  
        }  
    }  
    return 0;  
}  
  
int wait(int s)  
{  
    return(--s);  
}  
  
int signal(int s)  
{  
    return(++s);  
}
```



```
user@DESKTOP-3HUAHBC: ~  
user@DESKTOP-3HUAHBC:~$ vi Exp92.c  
user@DESKTOP-3HUAHBC:~$ gcc Exp92.c  
user@DESKTOP-3HUAHBC:~$ ./a.out
```

```
1.Producer  
2.Consumer  
3.ExitEnter your Choice: 1
```

```
Producer produces the item 1  
Enter your Choice: 2
```

```
Consumer consumes item 1  
Enter your Choice: 3
```

```
user@DESKTOP-3HUAHBC:~$ ./a.out
```

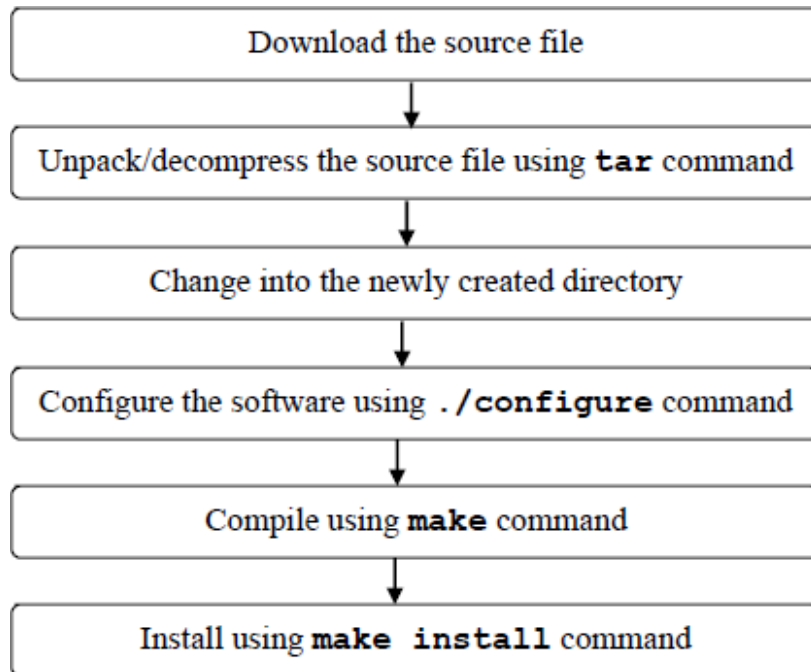
```
1.Producer  
2.Consumer  
3.ExitEnter your Choice: 2
```

```
Buffer is empty!!  
Enter your Choice: 2  
Buffer is empty!!  
Enter your Choice: 1
```

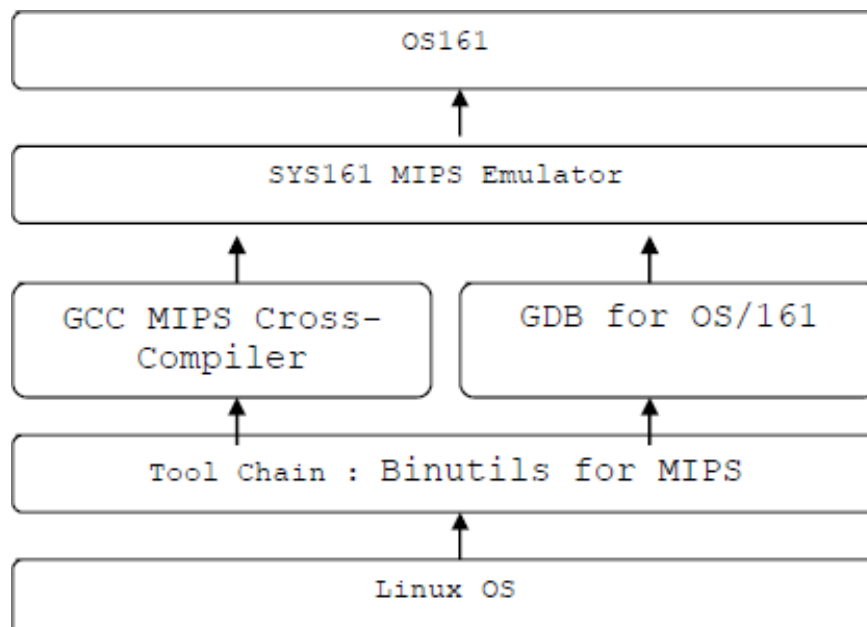
```
Producer produces the item 1  
Enter your Choice: 3
```

## OS Exp.13: STUDY OF OS161

### STEPS TO BUILD SOFTWARE FROM SOURCE FILE:



### BUILD SOFTWARE FRAMEWORK FOR OS/161



## **OS Exp.14: UNDERSTANDING THE OS161 FILE SYSTEM AND WORKING WITH TEST PROGRAMS**

### **Prerequisites**

- Linux desktop with UBUNTU Version 12.04 or later.
- Internet connections to download and install packages

### **Pre Installation Steps**

1. Install the packages gettext (to translate native language statements into English) , texinfo (to translate source code into other formats) and libncurses5-dev ( allows users to write text-base GUI)

```
sudo apt-get install gettext
sudo apt-get install texinfo
sudo apt-get install libncurses5-dev
```

2. Include the paths \$HOME/sys161/bin and \$HOME/sys161/tools/bin into PATH environment variable. Add the following line at the end of the file `.bashrc`

```
export PATH=$HOME/sys161/bin:$HOME/sys161/tools/bin:$PATH
```

- Now logout and login to get the PATH updated. You can check the current setting of the PATH environment variable using the command

```
printenv PATH
```

### **Installation Steps**

**STEP 1:** Download the following source codes one by one

- **Binutils for MIPS**  
[http://www.student.cs.uwaterloo.ca/~cs350/os161\\_repository/os161-binutils.tar.gz](http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-binutils.tar.gz)
- **GCC MIPS Cross-Compiler**  
[http://www.student.cs.uwaterloo.ca/~cs350/os161\\_repository/os161-gcc.tar.gz](http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-gcc.tar.gz)
- **GDB for Use with OS/161**  
[http://www.student.cs.uwaterloo.ca/~cs350/os161\\_repository/os161-gdb.tar.gz](http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-gdb.tar.gz)
- **bmake for use with OS/161**  
[http://www.student.cs.uwaterloo.ca/~cs350/os161\\_repository/os161-bmake.tar.gz](http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-bmake.tar.gz)
- **mk for use with OS/161**  
[http://www.student.cs.uwaterloo.ca/~cs350/os161\\_repository/os161-mk.tar.gz](http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-mk.tar.gz)
- **sys/161**  
[http://www.student.cs.uwaterloo.ca/~cs350/os161\\_repository/sys161.tar.gz](http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/sys161.tar.gz)
- **OS/161**  
[http://www.student.cs.uwaterloo.ca/~cs350/os161\\_repository/os161.tar.gz](http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161.tar.gz)
- Note : bmake and mk utilities are BSD make utilities used for OS161

**STEP 2: Build and Install the Binary Utilities (Binutils)**

- Unpack the binutils archive:  

```
tar -xzf os161-binutils.tar.gz
```
- Move into the newly-created directory:

```
cd binutils-2.17+os161-2.0.1
```

- Configure binutils:  

```
./configure --nfp --disable-werror  
target=mipsharvard-os161 --prefix=$HOME/sys161/tools
```
- Make binutils:  

```
make
```
- Finally, once **make** has succeeded, install the binutils into their final location:  

```
make install
```
- This will create the directory **\$HOME/sys161/tools/** and populate it.

### STEP 3: Install the GCC MIPS Cross-Compiler

- Unpack the gcc archive:  

```
tar -xzf os161-gcc.tar.gz
```
- Move into the newly-created directory:  

```
cd gcc-4.1.2+os161-2.0
```
- Configure gcc  

```
./configure -nfp --disable-shared --disable-threads  
--disable-libmudflap --disable-libssp  
target=mipsharvard-os161 --prefix=$HOME/sys161/tools
```
- Make it and install it:  

```
make  
make install
```

### STEP 4: Install GDB

- Unpack the gdb archive:  

```
tar -xzf os161-gdb.tar.gz
```
- Move into the newly-created directory:  

```
cd gdb-6.6+os161-2.0
```
- Configure gdb  

```
./configure --target=mips-harvard-os161  
prefix=$HOME/sys161/tools --disable-werror
```
- Make it and install it:  

```
make  
make install
```

### Step 5: Install bmake

- Unpack the bmake archive:  

```
tar -xzf os161-bmake.tar.gz
```
- Move into the newly-created directory:  

```
cd bmake
```
- Unpack mk within the bmake directory:  

```
tar -xzf ../os161-mk.tar.gz
```
- Run the bmake bootstrap script  

```
./boot-strap --prefix=$HOME/sys161/tools
```
- As the **boot-strap** script finishes, it should print a list of commands that you can run to install bmake under **\$HOME/sys161/tools**. The list should look something like this:  

```
mkdir -p /home/kmsalem/sys161/tools/bin
```

```

cp /home/kmsalem/bmake/Linux/bmake
/home/kmsalem/sys161/tools/bin/bmake-20101215
rm -f /home/kmsalem/sys161/tools/bin/bmake
ln -s bmake-20101215 /home/kmsalem/sys161/tools/bin/bmake
mkdir -p /home/kmsalem/sys161/tools/share/man/cat1
cp /home/kmsalem/bmake/bmake.cat1
/home/kmsalem/sys161/tools/share/man/cat1/bmake.1
sh /home/kmsalem/bmake/mk/install-mk
/home/kmsalem/sys161/tools/share/mk

```

- Run the commands printed by boot-strap in the order in which they are listed in your terminal screen.

## STEP 6: Set Up Links for Toolchain Binaries

```

mkdir $HOME/sys161/bin
cd $HOME/sys161/tools/bin
sh -c 'for i in mips-*; do ln -s $HOME/sys161/tools/bin/$i
$HOME/sys161/bin/cs350-`echo $i | cut -d- -f4-`; done'
ln -s $HOME/sys161/tools/bin/bmake $HOME/sys161/bin/bmake

```

- When you are finished with these steps, a listing of the directory **\$HOME/sys161/bin** should look similar to this:

```

bmake@          cs350-gcc@          cs350-ld@          cs350-run@
cs350-addr2line@ cs350-gcc-4.1.2@ cs350-nm@          cs350-size@
cs350-ar@        cs350-gccbug@     cs350-objcopy@     cs350-strings@
cs350-as@        cs350-gcov@       cs350-objdump@     cs350-strip@
cs350-c++filt@   cs350-gdb@        cs350-ranlib@
cs350-cpp@       cs350-gdbtui@     cs350-readelf@

```

## Step 7: Build and Install the sys161 Simulator

- Unpack the sys161 archive:  
**tar -xzf sys161.tar.gz**
- Move into the newly-created directory:  
**cd sys161-1.99.06**
- Next, configure sys161:  
**./configure --prefix=\$HOME/sys161 mipseb**
- Build sys161 and install it:  
**make**  
**make install**
- Finally, set up a link to a sample sys161 configuration file  
**cd \$HOME/sys161 ln-**  
**share/examples/sys161/sys161.conf.sample**  
**sys161.conf**

## Step 8: Install OS/161

- First, create a directory to hold the OS/161 source code, your compiled OS/161 kernels, and related test programs.  
**cd \$HOME**  
**mkdir cs350-os161**

- Next, move the OS/161 archive into your new directory and unpack it:  
`mv os161.tar.gz cs350-os161`  
`cd cs350-os161`  
`tar -xzf os161.tar.gz`
- This will create a directory called **os161-1.99** (under **cs350-os161**) containing the OS/161 source code. You should now be able build, install, and run an OS/161 kernel and related application and test programs by following steps.

### **Step 9: Configure OS/161 and Build the OS/161 Kernel**

The next step is to configure OS/161 and compile the kernel. From the **cs350-os161** directory, do the following:

```
cd os161-1.99
./configure --ostree=$HOME/cs350-os161/root --
toolprefix=cs350-
cd kern/conf
./config ASST0
cd ../compile/ASST0
bmake depend
bmake
bmake install
```