Qs.No	Questions
1	**a)Write about BIOS and MBR stages of a typical Linux boot process. (exp1)** Linux Boot Process Diagram and brief explanation. **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. You can press a key (typically F12 or F2, but it depends on your system) during the BIOS startup to change the boot sequence. □ 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. **MBR** □ MBR stands for Master Boot Record. □ It islocated 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. **b)Write a program to send a message (pass through command line arguments) into a message queue. Send few messages with unique message numbers. (exp12b)** **Message Queues:** A message queue is a linked list of messages stored within the kernel. It is identified by a message queue identifier. This method offers communication between single or multiple processes with full-duplex capacity. Reader and Writer Process Program (with user input from cmd) https://www.geeksforgeeks.org/ipc-using-message-queues/ **For questions with:** 1) send a message – Reader Process Program 2) receive a message – Writer Process Program In this question, only Reader Process Program. (send a message)
2	**a)Write about File permissions in Basic Linux commands (exp4)** https://linuxize.com/post/understanding-linux-file-permissions/ **File Permission** -- chmod command is used to change the access permission of a file. Method-1

	Syntax : chmod [ugo] [+/-] [rwxa] filename u : user, g : group, o : others + : Add permission - : Remove the permission r : read, w : write, x : execute, a : all permissions ex. chmod ug+rw f1 adding ‗read & write' permissions of file f1 to both user and group members. Method-2 Syntax : chmod octnum file1 The 3 digit octal number represents as follows □ first digit -- file permissions for the user □ second digit -- file permissions for the group □ third digit -- file permissions for others Each digit is specified as the sum of following 4 – read permission, 2 – write permission, 1 – execute permission ex. chmod 754 f1 it change the file permission for the file as follows □ read, write & execute permissions for the user ie; 4+2+1 = 7 □ read, & execute permissions for the group members ie; 4+0+1 = 5 □ only read permission for others ie; 4+0+0 = 4 **b)Write a program to receive a particular message from the message queue. Use message number to receive the particular message** (Refer Q1) In this question, only Writer Process Program. (receive a message)
3	**a)Write about Linux File system. (exp2)** **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 knows 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, butit 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.
 Nfs: The network file system used to access disks located on remote computers.
 Ntfs: replaces Microsoft Window's FAT file systems (VFAT, FAT32). It has reliability, performance, and space- utilization.
 Umsdos: It is an extended DOS file system used by Linux.

File System Structure

A file system is a logical collection of files on a partition or disk. A partition is a container for information and can span an entire hard drive if desired. UNIX uses a hierarchical file system structure, much like an upside-down tree, with root (/) at the base of the file system and all other directories spreading from there.
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 super user root.
/sbin Essential system binaries.
/srv Data for services provided by the system.
/proc Contains all processes marked as a file by process number or other information that is dynamic to the system

/tmp Temporary files.
/usr Secondary hierarchy with read-only data.
/var Variable data such as log files
/kernal Contains kernel files

b)Write a command (exp2)

i. to sort the file os and write the output into the file f22. Also eliminate duplicate lines.
ii. to display the unique lines of the sorted file f21. Also display the number of occurrences of each line.

Answer

Concept : Filters in Linux : Sort, Uniq

Aim
To sort content of files and then write the sorted output to a new file f22

Theory
Sort https://www.geeksforgeeks.org/sort-command-linuxunix-examples/
 Used to sort the file in order
Syntax : sort filename

Sorts the data as text by default
Sorts by the first filed by default
-r option sorts the file in descending order
-u eliminates duplicate lines
-o filename writes sorted data into the file fname
-tdchar sorts the file in which fileds are separated by dchar
-n sorts the data as number
+1n skip first filed and sort the file by second filed numerically

Uniq https://www.geeksforgeeks.org/uniq-command-in-linux-with-examples/
 Displays unique lines of a sorted file
Syntax : uniq filename
-d option displays only the duplicate lines
-c displays unique lines with no. of occurrences.

Syntax1 Used (Sort)
sort -o outputfile.txt inputfile.txt
or
sort inputfile.txt > outputfile.txt

#sorts data of input file to output file and saves it in output file, no change in input file

Program1
#create input file
nano f21.txt #have names as repetitions for next program
abhishek
chitransh
divyam
harsh
harsh
naveen
naveen
rajan
rajan
rajan
satish
(ctrl + O, enter, ctrl + X)
cat f21.txt

#create output file
touch f22.txt

	cat f22.txt #sort and save in output file f22 Sort -o f22.txt f21.txt cat f22.txt #sorted cat f21.txt #unsorted **Syntax1 Used (Uniq)** unique filename **Note**: File must be sorted in order for uniq command to work **Program2** #display content of f21 file cat f21.txt #unsorted sort f21.txt #sorted but not saved sort -o f21.txt f21.txt #sorted and saved uniq f21.txt #displays only unique lines and no duplicates uniq -c f21.txt #displays all lines ones and with number of occurences
4	**a)Write about Grub and Kernel stages of a typical Linux boot process(exp1)** Linux Boot Process Diagram and brief explanation. **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). □ 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. **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. **b)Write a command (exp4)** **i. to sort the file /etc/passwd in descending order** **ii. to sort the file /etc/passwd by user-id numerically.** **Answer** **Theory**

	Sort https://www.geeksforgeeks.org/sort-command-linuxunix-examples/ □ Used to sort the file in order Syntax : sort filename or sort filepath Sorts the data as text by default Sorts by the first filed by default -r option sorts the file in descending order -u eliminates duplicate lines -o filename writes sorted data into the file fname -tdchar sorts the file in which fileds are separated by dchar -n sorts the data as number +1n skip first filed and sort the file by second filed numerically **Syntax Used (Sort)** sort -r filepath #sorts in descending order sort -n filepath #sorts in numerical order https://www.spiceworks.com/tech/question/how-to-sort-the-etc-passwd-by-uid-in-hp-ux-060110/ cat /etc/passwd |awk -F ":" '{print $3}' |sort -rnk1 |more #displays user-id number only cat /etc/passwd | sort -n -t: + #sorts the /etc/passwd sorted by useri-id number numberically **Program** sort /etc/passwd #sorts sort -r /etc/passwd #sorts in reverse sort -n /etc/passwd #sorts numerically cat /etc/passwd |awk -F ":" '{print $3}' |sort -rnk1 |more #displays user-id number only cat /etc/passwd | sort -n -t: + #sorts the /etc/passwd sorted by useri-id number numberically
5	**a) Write about init and run level stages of a typical Linux boot process.(exp1)** Linux Boot Process Diagram and brief explanation. **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. **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 forstartup.
 Programs starts with K are used during shutdown. K forkill.
 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 deamon, 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.

b) Write a command (exp 2&4)
i. to cut 5 to 8 characters of the file f1.
ii. to display user-id of all the users in your system.

Answer

Theory (filter&command)
cut
 Used to cut characters or fileds from a file/input
Syntax : cut -cchars filename
-ffieldnos filename

 By default, tab is the filed separator(delimiter). If the fileds of the files are separated by any other character, we need to specify explicitly by –d option

cut -ddelimitchar -ffileds filename

user-id https://www.cyberciti.biz/faq/linux-list-users-command/
Get a List of All Users using the /etc/passwd File
Local user information is stored in the /etc/passwd file. Each line in this file represents login information for one user. To open the file you can use cat and display all local users.
cat /etc/passwd
If you want to display only the username you can use cut commands to print only the first field containing the username.
cut -d: -f1 /etc/passwd
Displays only user-id number
id

	Program1(Cut) nano f1.txt WelcomeHome (ctrl + O, enter, ctrl + X) cat f1.txt cut -c 5-8 f1.txt #displays omeH **Program2(user-id)** cat /etc/passwd #displays all local users with columns cut -d: -f1 /etc/passwd #displays only usernames id #displays user-id numbers with format https://www.spiceworks.com/tech/question/how-to-sort-the-etc-passwd-by-uid-in-hp-ux-060110/ cat /etc/passwd |awk -F ":" '{print $3}' |sort -rnk1 |more #displays the user-id number only
6	**a) Write about Vi Editors.(exp 2)** **VI EDITOR** https://www.javatpoint.com/vi-editor □ vi fname □ to open the file fname □ There are two types of mode in vi editor Escape mode – used to give commands – to switch to escape mode, press <Esc> key Command mode – used to edit the text – to switch to command mode, press any one the following inserting text command **a) Inserting Text** i □ insert text before the cursor a □ append text after the cursor I □ insert text at the beginning of the line A □ append text to the end of the line r □ replace character under the cursor with the next character typed R □ Overwrite characters until the end of the line o □ (small o) open new line after the current line to type text O □ (capital O) open new line before the current line to type text **b) Cursor movements** h □ left j □ down k □up l □ right (The arrow keys usually work also) ^F □forward one screen ^B □back one screen

	^D □down half screen ^U □up half screen (^ indicates control key; case does not matter) 0 □ (zero) beginning of line $ □ end of line **c) Deleting text** Note : (n) indicates a number, and is optional dd □ deletes current line (n) dd □ deletes(n) line(s) ex. 5dd □ deletes 5 lines (n)dw □ deletes (n) word(s) D □ deletes from cursor to end of line x □ deletes current character (n)x □ deletes (n) character(s) X □ deletes previous character **d) Saving files** :w □ to save & resume editing (write & resume) :wq □ to save & exit (write & quit) :q! □ quit without save e) Cut, Copy and Paste yy □ copies current line (n) yy □ copies (n) lines from the current line. ex. 4yy copies 4 lines. p □ paste deleted or yanked (copied) lines after the cursor **b)Write a program to perform process synchronization in producer-consumer problem (exp13)** https://www.geeksforgeeks.org/producer-consumer-problem-in-c/
7	**a) Write about OS161 file system. (exp14&15)** Draw two diagrams, pre installation, installation steps (mention step heading and few easy commands) https://cs-uob.github.io/COMS20012/labs/LAB5.html To run simulation, in cmd: cd $HOME/cs350-os161/root cp $HOME/sys161/sys161.conf sys161.conf sys161 kernel-ASST0 **b) Write a Program to demonstrate the concept of process creation. (exp 8)** Syntax for process creation int fork(); Returns 0 in child process and child process ID in parent process. Other Related Functions int getpid() □ returns the current process ID int getppid() □ returns the parent process ID wait() □ makes a process wait for other process to complete

	Virtual fork vfork() function is similar to fork but both processes shares the same address space. Child & parent process creation https://www.tutorialspoint.com/how-to-create-a-process-in-linux#:~:text=A%20new%20process%20can%20be,newly%20is%20called%20child%20process. Extra concepts https://www.tutorialspoint.com/inter_process_communication/inter_process_communication_process_creation_termination.htm

8	**a)Write the OS161 Installation steps**
	(refer Q7)
	
	b)Write a program to find the factorial of a given number. (Exp3)
	
	nano fact.c
	
	`#include<stdio.h>`
	`int main()`
	`{`
	`int i,fact=1,number;`
	`printf("Enter a number: ");`
	`scanf("%d",&number);`
	`for(i=1;i<=number;i++){`
	`fact=fact*i;`
	`}`
	`printf("Factorial of %d is: %d",number,fact);`
	`return 0;`
	`}`
	gcc fact.c #creates a.out file
	or
	gcc -o fact fact.c #creates output file named fact
	ls #a.out file or fact file created
	./a.out #displays compiled output
	or
	./fact #displays compiled output

9	**a)Understanding the OS161 file system and working with test programs** (refer Q7) **b)Write a program using exelp(). Rewrite the same using execlp() and execv() functions. (exp 9&10)** **Exec() System Call** The exec() system call replaces (overwrites) the current process with the new process image. The PID of the new process remains the same however code, data, heap and stack of the process are replaced

by the new program. There are 6 system calls in the family of exec(). All of these functions mentioned
below are layered on top of execve(), and they differ from one another and from execve() only in the
way in which the program name, argument list, and environment of the new program are specified

Total 6 types with syntax:
execl, execlp, execle, execv, execvp, execvpe

https://linuxhint.com/exec_linux_system_call_c/

error: there is no command called exelp, either its execl or execle or execvp

Program1.1 (execl())
nano f1.c

```
//int execl(const char *path, const char *arg, ..., NULL);

#include <unistd.h>
int main(void) {
  char *binaryPath = "/bin/ls";
  char *arg1 = "-lh";
  char *arg2 = "/home";

  execl(binaryPath, binaryPath, arg1, arg2, NULL);

  return 0;
}
```

gcc f1.c
./a.out #delete previous a.out file just to avoid ambiguity

Output1.1
total 8.0K
drwxr-xr-x 8 namz namz 4.0K Jun 28 16:52 namz
drwxr-xr-x 2 username username 4.0K Apr 29 10:44 username

Program1.2 (execle())
nano f1.c

```
//int execle(const char *path, const char *arg, ..., NULL, char * const envp[]
);

#include <unistd.h>
int main(void) {
  char *binaryPath = "/bin/bash";
  char *arg1 = "-c";
  char *arg2 = "echo \"Visit $HOSTNAME:$PORT from your browser.\"";
  char *const env[] = {"HOSTNAME=www.linuxhint.com", "PORT=8080", NULL};
```

```
  execle(binaryPath, binaryPath,arg1, arg2, NULL, env);

  return 0;
}
```

gcc f1.c
./a.out #delete previous a.out file just to avoid ambiguity

Output1.2
Visit www.linuxhint.com:8080 from your browser.

Program1.3 (execvp())
nano f1.c

```
//int execvp(const char *file, char *const argv[]);

#include <unistd.h>
int main(void) {
  char *programName = "ls";
  char *args[] = {programName, "-lh", "/home", NULL};

  execvp(programName, args);

  return 0;
}
```

gcc f1.c
./a.out #delete previous a.out file just to avoid ambiguity

Output1.3
total 8.0K
drwxr-xr-x 8 namz namz 4.0K Jun 28 16:52 namz
drwxr-xr-x 2 username username 4.0K Apr 29 10:44 username

Program2 (execlp())
nano f2.c

```
//int execlp(const char *file, const char *arg, …, NULL );

#include <unistd.h>
int main(void) {
  char *binaryPath = "/bin/ls";
  char *arg1 = "-lh";
  char *arg2 = "/home";

  execlp(binaryPath, binaryPath, arg1, arg2, NULL);
```

```
  return 0;
}
```

gcc f2.c
./a.out #delete previous a.out file just to avoid ambiguity

Output2
total 8.0K
drwxr-xr-x 8 namz namz 4.0K Jun 28 16:52 namz
drwxr-xr-x 2 username username 4.0K Apr 29 10:44 username

Program3 (execv)
nano f3.c

```
//int execv(const char *path, char *const argv[]);

#include <unistd.h>
int main(void) {
  char *binaryPath = "/bin/ls";
  char *args[] = {binaryPath, "-lh", "/home", NULL};

  execv(binaryPath, args);

  return 0;
}
```

gcc f3.c
./a.out #delete previous a.out file just to avoid ambiguity

Output3 3
total 8.0K
drwxr-xr-x 8 namz namz 4.0K Jun 28 16:52 namz
drwxr-xr-x 2 username username 4.0K Apr 29 10:44 username

10

a)Write about OS161.
(refer Q7)

b)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. (exp3)

grep

 Used to search one or more files for a particular pattern.

Syntax : grep pattern filename(s)

Lines that contain the pattern in the file(s) get displayed

pattern can be any regular expressions

More than one files can be searched for a pattern

	-v option displays the lines that do not contain the pattern -l list only name of the files that contain the pattern -n displays also the line number along with the lines that matches the pattern **Grep** is an essential Linux and Unix command. It is used to search text and strings in a given file. In other words, grep command searches the given file for lines containing a match to the given strings or words. It is one of the most useful commands on Linux and Unix-like system for developers. The grep utilities are a family that includes grep, egrep, and fgrep for searching files. https://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/ **SYNTAX** Look for all files in the current directory and in all of its subdirectories in Linux for the word 'httpd': **grep -R 'httpd' .** To interpret patterns **grep -F 'pattern' filename *# same as frgrep*** CMD commands pwd #present working directory ls #list files nano pattern.txt #create new file or open existing file using nano text editor # # # # # # # # # # # # # # # (ctrl + O, enter, ctrl + X) cat pattern.txt #to display pattern grep -R '# # # # #' . #searches for the string specified in all the existing files in present directory grep -F '#' pattern.txt #displays the line with contains the specified character
11	**a)Write about basic linux commands** (refer Q2) **b) Given the following values num=10, x=*, y=`date` a="Hello, 'he said'".Execute and write the output of the following commands** **echo num ,echo $num,echo $x,echo $(date). (exp 7)**

	https://www.javatpoint.com/steps-to-write-and-execute-a-shell-script nano f1.sh num=10, x=*, y=`date` a="Hello, 'he said'" echo num #num echo $num #10 echo $x #+ echo '$x' #$x echo "$x" #* echo $y #date echo $(date) #Tue 02 Mar 2021 12:21:26 AM EST echo $a #Hello, 'he said' echo \$num #$num echo \$$num #$10 (ctrl + O, enter, ctrl + X) chmod +x f1.sh ./f1.sh
12	**a)Write about Linux File System** (refer Q3) **b) Write a program to send a message (pass through command line arguments) into a message queue. Send few messages with unique message numbers.** (refer Q1)
13	**a)State the File System structure of Linux** (refer Q3) **b) Write a program to receive a particular message from the message queue. Use message number to receive the particular message** (refer Q1)

14	**a) Write about Vi Editors** (refer Q6) **b) Write a program to do the following:** **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) In the shared memory, the writing process, creates a shared memory of size 1K (and flags) and attaches the shared memory.The write process writes the data read from the standard input into the shared memory. Last byte signifies the end of buffer Read process would read from the shared memory and write to the standard (exp 12a)** https://www.geeksforgeeks.org/ipc-shared-memory/ Extra : https://dextutor.com/program-for-ipc-using-shared-memory/
15	**a) Write a command** **i. to display the names of nologin users.** **ii. to sort the file /etc/passwd in descending order** (refer Q4) **b) Write about Installation steps of OS161** (refer Q7)
16	**a) State the Steps to Build the Software from Source file (exp5)** File installation, package update To update the package repositories sudo apt-get update To update installed software sudo apt-get upgrade To install a package/software sudo apt-get install <package-name> https://www.linux.com/news/how-install-packages-source-linux/ Ways to build from source : apt-get, git, dpkg-buildpackage, apt-get build-dep **b) 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.** **#version1 (simple)** nano f1.sh TIME=$(date +"%H")

```
echo $TIME

if [ $TIME -lt 12 ]; then
   echo "Good morning"
elif [ $TIME -lt 18 ]; then
   echo "Good afternoon"
else
   echo "Good evening"
fi

chmod +x f1.sh
./f1.sh

#version2
nano f1.sh

hour=$(date | cut -c12-13)
if [ "$hour" -ge 0 -a "$hour" -le 11 ]
then
 echo "Good morning"
elif [ "$hour" -ge 12 -a "$hour" -le 17 ]
then
 echo "Good afternoon"
else
 echo "Good evening"
fi

chmod +x f1.sh
./f1.sh
```

17

a) Write about OS161

(refer Q7)

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

https://linuxize.com/post/chmod-command-in-linux/
https://www.guru99.com/file-permissions.html

note : x- eXecuting a directory means Being allowed to “enter” a dir and gain possible access to sub-dirs

```
nano hello.sh
num=10
echo num
(ctrl+O, enter, ctrl+x)
./hello.sh #-bash: ./hello.sh: Permission denied
ls -l hello.sh #to check permissions
#-rw-r--r-- 1 namz namz 16 Jun 28 19:00 hello.sh ,
chmod +x hello.sh
#-rwxr-xr-x 1 namz namz 16 Jun 28 19:00 hello.sh
./hello.sh #now file has permission to run
```

	(from q 2 https://linuxize.com/post/understanding-linux-file-permissions/ **File Permission** -- chmod command is used to change the access permission of a file. Method-1 Syntax : chmod [ugo] [+/-] [rwxa] filename u : user, g : group, o : others + : Add permission - : Remove the permission r : read, w : write, x : execute, a : all permissions ex. chmod ug+rw f1 adding ‗read & write' permissions of file f1 to both user and group members. Method-2 Syntax : chmod octnum file1 The 3 digit octal number represents as follows □ first digit -- file permissions for the user □ second digit -- file permissions for the group □ third digit -- file permissions for others Each digit is specified as the sum of following 4 – read permission, 2 – write permission, 1 – execute permission ex. chmod 754 f1 it change the file permission for the file as follows □ read, write & execute permissions for the user ie; 4+2+1 = 7 □ read, & execute permissions for the group members ie; 4+0+1 = 5 □ only read permission for others ie; 4+0+0 = 4)
18	**a) Write about Building Software framework of OS161** (refer Q7) **b) Schedule the following tasks (exp6)** **i. to display the following message on the monitor for every 2 minutes.** **ii. to take backup of your important file (say file f1) for every 30 minutes** **iii. to take backup of login information everyday 9:30am** **Crontab** Linux Cron utility is an effective way to schedule a routine background job at a specific

	time and/or day on an on-going basis. User can use this to schedule activities, either as one-time events or as recurring tasks. **Format:** https://www.youtube.com/watch?v=QZJ1drMQz1A minute(s) hour(s) day(s) month(s) weekday(s) command(s) "Argument1" "Argument2" Eg: 1*3*45 command_to_execute * -no parameter Or (to understand) # ┌──────────── minute (0 - 59) # │ ┌──────────── hour (0 - 23) # │ │ ┌──────────── day of month (1 - 31) # │ │ │ ┌──────────── month (1 - 12) # │ │ │ │ ┌──────────── day of week (0 - 6) (Sunday to Saturday; # │ │ │ │ │ 7 is also Sunday on some systems) # │ │ │ │ │ # │ │ │ │ │ # * * * * * command_to_execute **Program** windows subsystem crontab doesn't work touch f1.txt crontab -e #opens crontab editor in nano #to display the following message on the monitor for every 2 minutes 2 * * * * echo 'Hello' >> /home/namz/f1.txt #to take backup of your important file (say file f1) for every 30 minutes 30 * * * * /f1.txt #backup of login information everyday 9:30pm 30 9 * * * tar -zcf /var/backups/home.tgz /home (ctrl+O, enter, ctrl+x) # must show….crontab: installing new crontab crontab -l #lists crontab code cat f1.txt #check after 3 min
19	**a) Write about Prerequisites and Pre Installation involved in OS161** (refer Q7) **b) Write the commands for following** **i. to sort the file /etc/passwd by user-id numerically** **ii. to paste all the lines of the file f1 into single line** (refer Q4) nano f1.txt

	Hello There User echo $(cat f1.txt) #displays in single line

20	**a) Write about Installation steps of OS161**
	(refer Q7)
	
	b) Write the commands for following:
	i. Stop the networking service and then start the service
	ii. Check the connectivity of the host with IP address 127.0.0.1
	iii. Find the IP address of the localhost
	iv. Find the IP address of the DNS Server
	
	i)
	https://www.layerstack.com/resources/tutorials/How-to-restart-Network-Interface-or-Network-Adapter-on-Linux-and-Windows-Cloud-Servers
	
	Wont work in linux subsystem
	
	`# sudo /etc/init.d/networking restart`
	`or`
	`# sudo /etc/init.d/networking stop`
	`# sudo /etc/init.d/networking start`
	`else`
	`# sudo systemctl restart networking`
	
	ii)
	ping localhost
	
	ping 127.0.0.1
	
	(ctrl+z – to exit)
	Localhost is the default name of the computer you are working on. The term is a pseudo name for 127.0. 0.1, **the IP address of the local computer**. This IP address allows the machine to connect to and communicate with itself
	
	iii) hostname -I # IP address of the localhost
	
	iv) cat /etc/resolv.conf # IP address of the DNS Server

21	**a)Write about BIOS and MBR stages of a typical Linux boot process** (Refer Q1) **b)Write a program to receive a particular message from the message queue. Use message number to receive the particular message**

	(refer Q1)
22	**a) Write about Installation steps of OS161** (refer Q7) **b) Write the commands for following** **i. to sort the file /etc/passwd by user-id numerically** **ii. to paste all the lines of the file f1 into single line** (refer Q 4& 19)
23	**a) State the Steps to Build the Software from Source file** (refer Q16) **b) Write a program to check whether the file has execute permission or not. If not, add the permission.** (refer Q17)
24	**a) Write about OS161** (refer Q7) **b) 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 display** (refer q16)
25	**a) State the Steps to Build the Software from Source file** (refer Q16) **b)Write a program to do the following: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) In the shared memory, the writing process, creates a shared memory of size 1K (and flags) and attaches the shared memory.The write process writes the data read from the standard input into the shared memory. Last byte signifies the end of buffer Read process would read from the shared memory and write to the standard** (refer Q14)
26	**a)Write about Linux File System** (refer Q3) **b) Write a program to receive a particular message from the message queue. Use message number to receive the particular message.** (refer Q1)
27	**a)State the File System structure of Linux** (refer Q3) **b) Write a program to send a message (pass through command line arguments) into a message queue. Send few messages with unique message numbers.**

	(refer Q1)
28	**a)Write about OS161.** (refer Q7) **b) Given the following values num=10, x=*, y=`date` a="Hello, 'he said'".Execute and write the output of the following commands** **echo num ,echo $num,echo $x,echo $(date).** (refer Q11)
29	**a)Write about basic linux commands** (refer Q3) **b)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.** (Refer Q10)
30	**a)Write the OS161 Installation steps** (refer Q7) **b)Write a program using exelp(). Rewrite the same using execlp() and execv() functions.** (referQ9)
31	**a)Understanding the OS161 file system and working with test programs** (refer Q7) **b)Write a program to send a message (pass through command line arguments) into a message queue. Send few messages with unique message numbers** (refer Q1)
32	**a)Write about BIOS and MBR stages of a typical Linux boot process.** (Refer Q1) **b)Write a program to find the factorial of a given number.** (refer q8)
33	**a)Write about File permissions in Basic Linux commands** (refer Q2) **b)Write a command** **i. to sort the file os and write the output into the file f22. Also eliminate duplicate lines.** **ii. to display the unique lines of the sorted file f21. Also display the number of occurrences of each line.** (refer q3)

34	**a) State the Steps to Build the Software from Source file**
	(refer Q16)
	
	b) Write a program to demonstrate the concept of mutual exclusion (exp13)
	
	https://www.geeksforgeeks.org/mutual-exclusion-in-synchronization/
	
	#include<sys/ipc.h>
	#include<sys/sem.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;
	}

35	**a)Write about Grub and Kernel stages of a typical Linux boot process** (refer Q4) **b)Write a command** **i. to cut 5 to 8 characters of the file f1.** **ii. to display user-id of all the users in your system.** (refer q5)
36	**a)Write about init and run level stages of a typical Linux boot process.** (refer Q5) **b)Write a command**

	i)to sort the file /etc/passwd in descending order **ii)to sort the file /etc/passwd by user-id numerically** (refer Q4)
37	**a)Write about OS161 File system** (refer Q7) **b)Write a program to perform process synchronization in producer-consumer problem** (referQ6)
38	**a)Write about Vi Editors** (refer Q6) **b)Write a Program to demonstrate the concept of process creation.** (refer q7)
39	**a)Write about Linux File System** (refer Q3) **b) Write a program to do the following:** **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) In the shared memory, the writing process, creates a shared memory of size 1K (and flags) and attaches the shared memory.The write process writes the data read from the standard input into the shared memory. Last byte signifies the end of buffer Read process would read from the shared memory and write to the standard** (refer Q14)
40	**a) Write about Prerequisites and Pre Installation involved in OS161** (refer Q7) **b) Write the commands for following:** **i. Stop the networking service and then start the service** **ii. Check the connectivity of the host with IP address 127.0.0.1** **iii. Find the IP address of the localhost** **iv. Find the IP address of the DNS Server** (referQ20)
41	**a) Write about Vi Editors.** (refer Q6) **b) Write a program to perform process synchronization in producer-consumer problem** (referQ6)
42	**a) Write about init and run level stages of a typical Linux boot process.** (refer Q5) **b) Given the following values num=10, x=*, y=`date` a="Hello, 'he said'".Execute and write the output of the following commands** **echo $y,echo $(date),echo $a,echo \$num,echo \$$num**

	(refer Q11)
43	**a) Write about OS161 File system** (refer Q7) **b) Write a program to demonstrate the concept of mutual exclusion** (refer Q34)
44	**a)Write about Linux File system.** (refer Q3) **b)Write a program to receive a particular message from the message queue. Use message number to receive the particular message** (refer Q1)
45	**a) Write about Linux File System** (refer Q3) **b) Write a program to demonstrate the concept of mutual exclusion** (refer Q34)
46	**a) Write about File permissions in Basic Linux commands** (refer Q2) **b) Write a program to demonstrate the concept of mutual exclusion** (refer Q34)
47	
48	**a) Write about File permissions in Basic Linux commands** (refer Q2) **b) Given the following values num=10, x=*, y=`date` a="Hello, 'he said'".Execute and write the output of the following commands** **echo $y,echo $(date),echo $a,echo \$num,echo \$$num** (refer Q11)
49	**a) Write about basic linux commands .** (refer Q2) **b) Write a program to demonstrate the concept of mutual exclusion** (refer Q34)
50	**a) Write about Vi Editors.** (refer Q6) **b) Write a program to demonstrate the concept of mutual exclusion** (refer Q34)
51	**a) Write about basic linux commands .** (refer Q2) **b) Write a program to perform process synchronization in producer-consumer problem** (referQ6)
52	**a) Write about File permissions in Basic Linux commands** (refer Q2) **b) Write a program to perform process synchronization in producer-consumer problem**

	(refer Q6)
53	**a) State the Steps to Build the Software from Source file** (refer Q16) **b) Write a program to perform process synchronization in producer-consumer problem** (referQ6)
54	**a) State the Steps to Build the Software from Source file** (refer Q16) **b) Given the following values num=10, x=*, y=`date` a="Hello, 'he said'".Execute and write the output of the following commands** **echo $y,echo $(date),echo $a,echo \$num,echo \$$num** (refer Q11)
55	**a) Write about Grub and Kernel stages of a typical Linux boot process.** (Refer Q4) **b) Write a program to demonstrate the concept of mutual exclusion** (refer Q34)
56	**a) Write about Vi Editors.** (refer Q6) **b) Write a program to perform process synchronization in producer-consumer problem** (referQ6)
57	**a) Write about init and run level stages of a typical Linux boot process.** (refer Q5) **b) Write a program to demonstrate the concept of mutual exclusion** (refer Q34)
58	**a) Write about File permissions in Basic Linux commands** (refer Q2) **b) Given the following values num=10, x=*, y=`date` a="Hello, 'he said'".Execute and write the output of the following commands** **echo $y,echo $(date),echo $a,echo \$num,echo \$$num** (refer Q11)
59	**a) Write about init and run level stages of a typical Linux boot process.** (refer Q5) **b) Write a program to perform process synchronization in producer-consumer problem** (referQ6)
60	**a) Write about File permissions in Basic Linux commands** (refer Q2) **b) Given the following values num=10, x=*, y=`date` a="Hello, 'he said'".Execute and write the output of the following commands**

	echo $y,echo $(date),echo $a,echo \$num, echo \$$num (refer Q11)