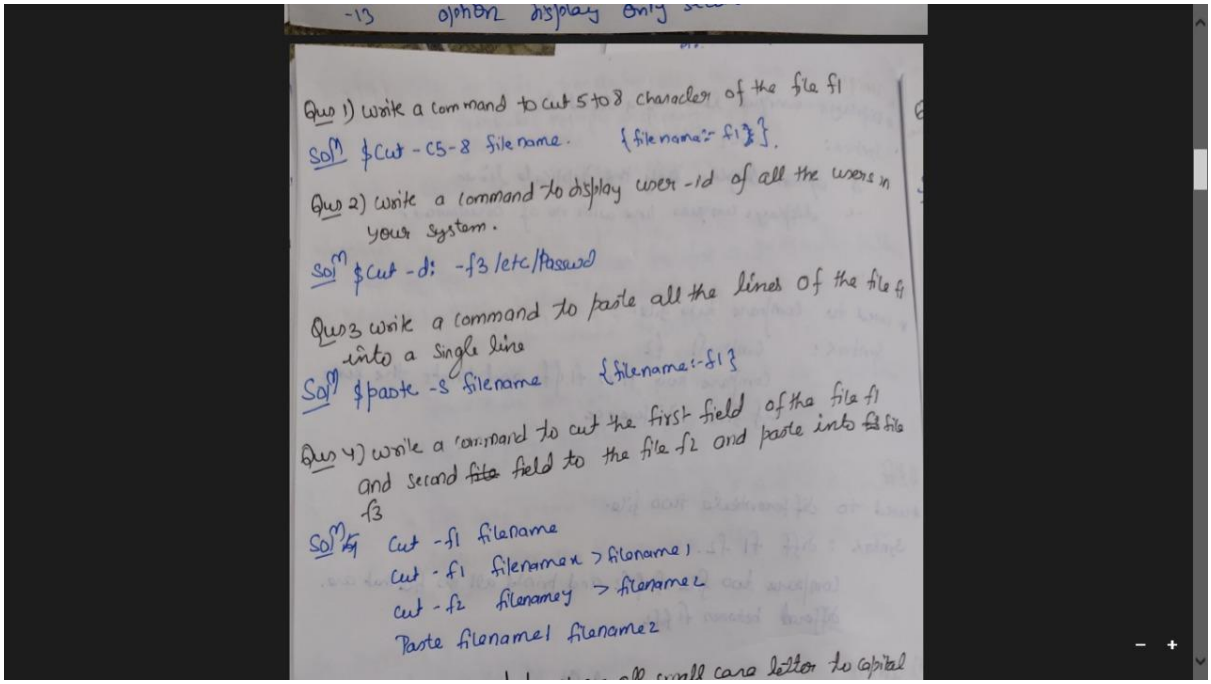
exp 4



-13 option display only

Qus 1) write a command to cut 5 to 8 character of the file f1

Sol^{n} $cut -c5-8 filename. {filename:- f1}.

Qus 2) write a command to display user-id of all the users in your system.

Sol^{n} $cut -d: -f3 /etc/passwd

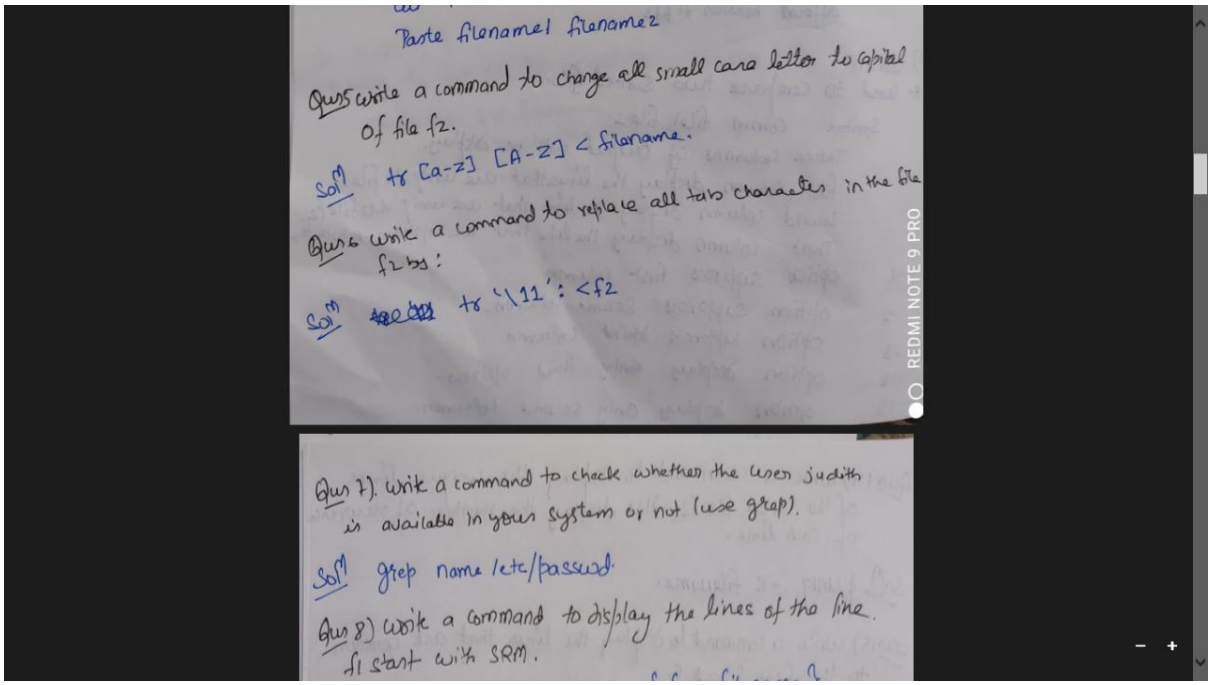
Qus 3 write a command to paste all the lines of the file f1 into a single line

Sol^{n} $paste -s filename {filename:-f1}

Qus 4) write a command to cut the first field of the file f1 and second ~~fifth~~ field to the file f2 and paste into ~~f3~~ file f3

Sol^{n} by Cut -f1 filename
Cut -f1 filename1 > filename1
Cut -f2 filename2 > filename2
Paste filename1 filename2

all small case letter to capital



Paste filename1 filename2

Qus 5 write a command to change all small case letter to capital of file f2.

Sol^{n} tr [a-z] [A-Z] < filename.

Qus 6 write a command to replace all tab character in the file f2 by :

Sol^{n} ~~tr~~ tr '\11' : <f2

REDMI NOTE 9 PRO

Qus 7). write a command to check whether the user judith is available in your system or not (use grep).

Sol^{n} grep name /etc/passwd.

Qus 8) write a command to display the lines of the line. f1 start with SRM.

Qus 8) Write a command to display the ... of f1 start with SRM.

Sol) $ sed -n '/^srm/p' f2 {f2 :- filename}

Qus 9). Write a command to display the name of the files in the directory /etc/init.d that contains the pattern grep.

Sol) $ grep grep /etc/init.d

Qus 10) Write a command to display the names of nologin user. (Hint:- the command nologin is specified in the last filed of the file /etc/passwd for nologin users)

Sol) $ grep nologin /etc/passwd | cut -d: -f1.

Qus 11 write a command to sort the file /etc/passwd in descending order.

Sol) $ sort -r /etc/passwd

Qus 12) Write a command to sort the file /etc/passwd by user-id numerically (Hint: user-id is in 3rd field)

Sol) $ cut -d: -f3 /etc/passwd | sort -n.

Qus 13) Write a command to sort the file f2 and write the output into the file f22. Also eliminate duplicate lines.

Sol) $ sort filename x > filename1
$ sort -u filename x > filename1.

Qus 14) Write a command to display the unique lines of the sorted file f21. Also display the number of occurrences of each line.

Sol) $ uniq -c filename.

Qus 15). Write a command to display the lines that are common to the files f1 and f2

Sol) comm -12 filename x filename y

Experiment no :-5

Q1. Update the package repositories

Sol:- sudo apt-get update

Q2. Install the package “simplescreenrecorder”

Sol:- sudo apt-get install simplescreenrecorder

Q3. Remove the package “simplescreenrecorder”

sol:-sudo apt-get remove simplescreenrecorder

Q4. Create a user ‘elias’. Login to the newly created user and exit

Sol:- sudo adduser elias
vi /etc/passwd

Q5. Disable the user ‘elias’, try to login and enable again.

Sol:- sudo passwd -l elias
sudo passwd -u elias

Q6. Create a group ‘cse’ and add the user ‘elias’ in that group

Sol:- sudo addgroup cse
sudo useradd elias cse

Q7. List the account expiry information of the user ‘elias’

Sol:- sudo chage -l elias

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

Sol:- sudo chage elias
Sudo change -l elias

Q9. Delete the user ‘elias’ and then delete the group ‘cse’

Sol:- sudo deluser elias
sudo delgroup cse

Q10. List the partitions available in your system

Sol:- sudo fdisk -l

Q11. What are the file systems used in your system

Sol:- vi /etc/fstab

Q12. Stop the networking service and then start the service

sol:- sudo /etc/init.d/networking stop
sudo /etc/init.d/networking restart

Q13. Check the connectivity of the host with IP address 127.0.0.1

sol:- Ping 127.0.0.1

Q14. Find the IP address of the localhost

Sol:- vi /etc/hosts

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

Sol:- vi /etc/resolv.conf

Q16. Install mysql server

Sol:- sudo apt-get install mariadb-server
I used mariadb server because i faced a problem while using mysql server

Or

sudo apt install my-sql-server

Q17. Restart mysql server

Sol:- sudo /etc/init.d/mysql start
sudo /etc/init.d/mysql restart

Q18. Check the configuration file for mysql server

Sol:- vi /etc/mysql/my.cnf

Q19. Log on as root into mysql server

Sol:- mysql -u root -p

Q20. Create a new database for mysql server

Sol:- create database myDB

Exp-6-Answers

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

a) tty (to see the name of monitor)
*/2 * * * * echo Hi User>> (your monitor name)

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

a) */30 * * * * cp f1 f1backup

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

a) 30 9 * * * cp /etc/passwd backuppass

Exp-7-Answers

1. Given the following values
   num=10, x=*, y=`date` a="Hello, 'he said'"

Execute and write the output of the following commands

Command Output	**Output**
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,

2. Find the output of the following shell scripts

$ vi ex51

```
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 :

```
Enter value for n
20
Sum is 100
```

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

```
$ vi ex52
        echo Enter name of the file
        read name
        if [ -x $name ]
        then
                echo Yes $name has Execute Permission
        else
                echo No $name has NO Execute Permission
        fi
```

4. 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 5:59:59p.m.), "Good afternoon" is displayed.
   If neither of the preceding two conditions is satisfied, "Good evening" is displayed.

```
$ vi ex53
        hour=$(date | cut -c12-13)
        if [ $hour -ge 0 -a $hour -le 11 ]
        then
                echo Good Morning
        elif [ $hour -le 17 ]
        then
```

```
        echo Good Afternoon
    else
        echo Good Evening
    fi
```

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

```
$ vi ex54
    for i in *
    do
        if [ -d $i ]
        then
            echo $i
        fi
    done
```

6. 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.

```
$ vi ex55
    for i in *
    do
        if [ -f $i ]
```

```
        then
                grep $1 $i > /dev/null
                if [ $? -eq 0 ]
                then
                        echo $i found
                else
                        echo $i not found
                fi
        fi
done
```

OS LAB (8 experiment answer)

Q1:-

```
#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;
 }
```

Q02:-

```
     #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; }
```

Q 03:-

```
 #include<stdio.h>
 #include<unistd.h>
int main()
{
        fork();
        fork();
        fork();
        printf("SRMIST\n");
        return 0;
}
```

8 TIMES SRMIST will be printed 2^3

Q04:-

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

```
}
else
{
sleep(1);
for(i=0;i<=n;i=i+2)
evensum+=i;
printf("Sum of even number is %d\n",evensum);
}
return 0;
}
```

Q 05:-

2^n-1

Q 06:-

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

Q 07:-

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

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

20 times yes will be printed as output

OS LAB – 9&10

Q1. Execute the Following Program and write the output

```
$vi ex61.c
#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:

Transfer to excelp function

Rohit

Sky

(note:file f1 contains Rohit and Sky as the first two lines)

Why second printf statement is not executing?

When the OS executing the line “ **execlp(“head”, “head”,”-2”,”f1”,NULL);** ” the control transferred to execlp() function and will not return back to the calling place (unless there is an error) and therefore the second printf line is not executing.

Q2. Rewrite question Q1 with execl() function. Pass the 3rd and 4th argument of the
functionexecl() through command line arguments.

Input: ./a.out -3 f1

```
$vi ex62.c
#include <stdio.h>
#include<unistd.h>
int main(intargc, char *argv[])
{
printf("Transfer to execute function \n");
execl("/user/bin/head", "head",argv[1],argv[2],NULL);
printf("This line will not execute \n");
return 0;
}
```

Output:

Transfer to execute function

Rohit

Sky

Kishan

(note:file f1 contains Rohit Sky and Kishan as the first two lines)

O.S. EXPERIMENT –11

1. Program 1:

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

OUTPUT:

Child : Writing to pipe
Child Exiting

Parent : Reading from pipe

Pipe content is : Welcome
Parent Exiting

OS LAB EXP 12a:Shared memory

Program:1- Shared memory implementation using readers writers problem.
Writer process:
Algorithm:

· **Step 1** - Create a shared memory using (shmget()) function.

· **Step 2** - attach the current process in to created shared memory be calling shmat() function.

· **Step 3** - Write into shared memory after attaching in to it.

· **Step 4-** After completing write operation detach the process from shared memory area.

Reader process:
Algorithm:

· **Step 1** - Create a shared memory using (shmget()) function.

· **Step 2** - attach the current process in to created shared memory be calling shmat() function.

· **Step 3** - read the data which is already written by the reader process from shared memory after
attaching in to it.

· **Step 4-** Print the string and detach the process from shared memory area.

Writer Program:

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

 shmid=shmget((key_t)9,1024,IPC_CREAT|0666);
 str=(char *)shmat(shmid,(char *)0,0);
 printf("Write data:");
 fgets(str,20,stdin);
 printf("Data written in memory : %s \n",str);
```

```
 shmdt(str);
 return 0;
}
```

Reader Program :

```
#include<stdio.h>
#include<sys/ipc.h>
#include<sys/shm.h>
int main()
{
	int shmid;
	char *str;
	shmid=shmget((key_t)6,1024,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;
}
```

Output:

Writer.c

Write Data : Operating System Data

Written in memory: Operating System

Reader.c

Data read from memory: Operating System

OS LAB EXP 12b:Message Queue

Program :To perform communication using message queues, following are the steps -
Writer Process:

· **Step 1** - Create a message queue or connect to an already existing message queue (msgget())

· **Step 2** – specify the message type as 1.

· **Step 3-** Write into message queue (msgsnd())

· **Step 4**- terminate the process

Reader Process:

· **Step 1** - Create a message queue or connect to an already existing message queue (msgget())

· **Step 2** – specify the message type as 1.

· **Step 3** – Read from the message queue (msgrev())

· **Step 4** - Perform control operations on the message queue (msgctl())

· **Step 5** – terminate the reader process

Writer program:

```
#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;
}
```

OUTPUT:

$./a.out 1 welcome (note: 1 is messageId and welcome is message)

Message of size 7 sent successfully

Reader program:

```
#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)6,0666);
  x.mtype=atoi(argv[1]);
  len=atoi(argv[2]);

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

OUTPUT:

$ ./a.out 1 7 (note: 1 is messageId and 7 is size of the message)

The message is: welcome

OS LAB EXP 13

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

Output:

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

Q2. Write a program to perform process synchronization in producer-consumer problem

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

int mutex = 1;
int full = 0;
int empty = 10, x = 0;
void producer()
{
        --mutex;

        // Increase the number of full slots by 1
        ++full;

        // Decrease the number of empty slots by 1
        --empty;

        // Item produced
        x++;
        printf("\nProducer produces item %d",x);

        // Increase mutex value by 1
        ++mutex;
}
void consumer()
```

```
{
	// Decrease mutex value by 1
	--mutex;

	// Decrease the number of full slots by 1
	--full;

	// Increase the number of empty slots by 1
	++empty;
	printf("\nConsumer consumes item %d",x);
	x--;

	// Increase mutex value by 1
	++mutex;
}
int main()
{
	int n, i;
	printf("\n1. Press 1 for Producer"
		"\n2. Press 2 for Consumer"
		"\n3. Press 3 for Exit");

	for (i = 1; i > 0; i++) {

		printf("\nEnter your choice: ");
		scanf("%d", &n);

		switch (n) {
		case 1:

			if ((mutex == 1)
				&& (empty != 0)) {
				producer();
			}

			else {
				printf("Buffer is full!");
```

```
                }
                break;

        case 2:

                if ((mutex == 1)
                        && (full != 0)) {
                        consumer();
                }

                else {
                        printf("Buffer is empty!");
                }
                break;

        case 3:
                exit(0);
                break;
        }
    }
}
```

OUTPUT:

1. Press 1 for Producer
2. Press 2 for Consumer
3. Press 3 for Exit

Enter your choice: 1

Producer produces item 1

Enter your choice: 1

Producer produces item 2

Enter your choice: 2

Consumer consumes item 2

Enter your choice: 2

Consumer consumes item 1
Enter your choice: 2
Buffer is empty!
Enter your choice: 3