2.

```
id=`ps -e | grep gedit | cut -c1-5`
echo $id
kill -9 $id
id=`ps -e | grep firefox | cut -c1-5`
echo $id
kill -9 $id
ls -l | grep ^-
ls -l | grep ^- | wc -l
ls -l | grep ^d
ls -l | grep ^d | wc -l
ls -l | grep -e "May" | grep ^-
ls -l | grep -e "May" | grep ^- | wc -l
```

3.

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<unistd.h>
#include<string.h>
int main(){
int pid;
char str1[10];
printf("Enter the string:-\n");
scanf("%s",str1);
pid=fork();
if(pid==0){
int i, len, temp;
len = strlen(str1);
for (i = 0; i < len/2; i++)
{
temp = str1[i];
str1[i] = str1[len - i - 1];
str1[len - i - 1] = temp;
}
printf("The reverse of a string:-\n%s",str1);
printf("\nChild process completed");
}
else{
wait(NULL);printf("\nParent process completed\n");
}
return 0;
}
```

4.

server

```
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/shm.h>
struct shmseg{
char data[100];
int complete;
```

```
};
void main()
{
struct shmseg *shm;
char a[100];
int shmid;
key_t key=1122;
shmid=shmget(key,1024,0666|IPC_CREAT);
shm=shmat(shmid,NULL,0);
printf("Enter a string : ");
scanf("%s",a);
strcpy(shm->data,a);
printf("You wrote : %s\n",shm->data);
shm->complete=0;
while(shm->complete!=1);
if(strcmp(a,shm->data)==0)
printf("Entered string is palindrome\n");
else
printf("Entered string is not palindrome\n");
}
client
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#include<string.h>
#include <stdio.h>
struct shmseg{
char data[100];
int complete;
};
void main()
{
struct shmseg *shm;
char b[100];
int shmid,l,i,j=0;key_t key=1122;
shmid=shmget(key,1024,0666);
shm=shmat(shmid,NULL,0);
printf("Data read from shared memory is %s\n",shm->data);
l=strlen(shm->data);
for(i=l-1;i>=0;i--)
{
b[j]=shm->data[i];
j++;
}
b[j]='\0';
printf("Reverse of the string : %s\n",b);
strcpy(shm->data,b);
shm->complete=1;
}
5.
#include<stdio.h>
typedef struct process{
```

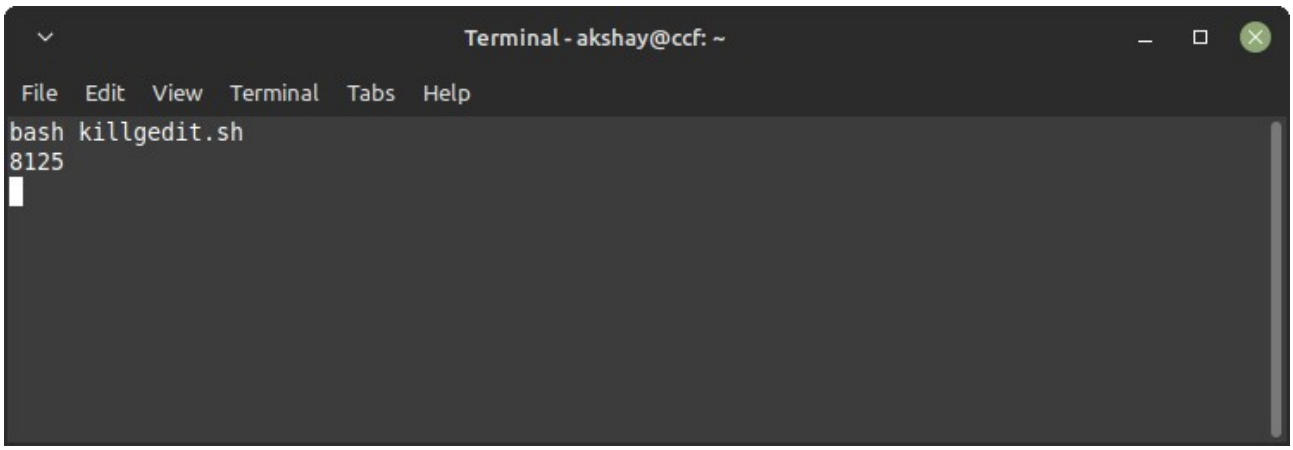
```
int pid;
int bt;
int tat;
int wt;
}proc;
void main(){
int n;
float ttat=0,atat=0;
float twt=0,awt=0;
proc p[10];
printf("Enter the total number of process : \n");
scanf("%d",&n);
printf("Enter the process id,burst time of each process:\n");
printf("\nPid\tBt\n");
printf("---------------\n");
for(int i=1;i<=n;i++){
scanf("%d%d",&p[i].pid,&p[i].bt);
}
//calculating turnaround time
p[0].tat=0;
for(int i=1;i<=n;i++){
p[i].tat = p[i-1].tat + p[i].bt;
}
//calculating waitting time
for(int i=1;i<=n;i++){
p[i].wt = p[i-1].tat;
}
//calculating average turnaround time
for(int i=1;i<=n;i++){
ttat = ttat + p[i].tat;
}
atat = (float)ttat / n;//calculating average waiting time
for(int i=1;i<=n;i++){
twt = twt + p[i].wt;
}
awt = (float)twt / n;
// print the table
printf("\nPid\tBt\tWt\tTat\n");
printf("-------------------------------\n");
for(int i=1;i<=n;i++){
printf("%d\t%d\t%d\t%d\n",p[i].pid,p[i].bt,p[i].wt,p[i].tat);
}
printf("\n");
printf("Average turnaround time= %.2fmsec",atat);
printf("\n");
printf("Average waiting time= %.2fmsec\n",awt);
}
#include <stdio.h>
# define MAXSIZE 10
int n;
struct process{
int pid;
```

```
int bt;
int wt;
int tt;
}P[MAXSIZE],temp;
void sort(){
for(int i=1;i<=n;i++){
for(int j=0;j<n-i;j++){
if(P[j].bt>P[j+1].bt){
temp=P[j];
P[j]=P[j+1];
P[j+1]=temp;
}
}
}
}
void waitingTime(){
P[0].wt=0;
for(int i=1;i<n;i++){
P[i].wt=P[i-1].wt+P[i-1].bt;
}
}
void turnAroundTime(){
P[0].tt=P[0].bt;
for(int i=1;i<n;i++){
P[i].tt=P[i-1].tt+P[i].bt;
}
}
float avg_wt(){
float total=0;for(int i=0;i<n;i++){
total=total+P[i].wt;
}
return total/n;
}
float avg_tt(){
float total=0;
for(int i=0;i<n;i++){
total=total+P[i].tt;
}
return total/n;
}
void main()
{
printf("Enter no.of processes:");
scanf("%d",&n);
for(int i=0;i<n;i++){
printf("Enter pid of process %d:",i+1);
scanf("%d",&P[i].pid);
printf("Enter burst time of process %d:",i+1);
scanf("%d",&P[i].bt);
}
sort();
waitingTime();
```

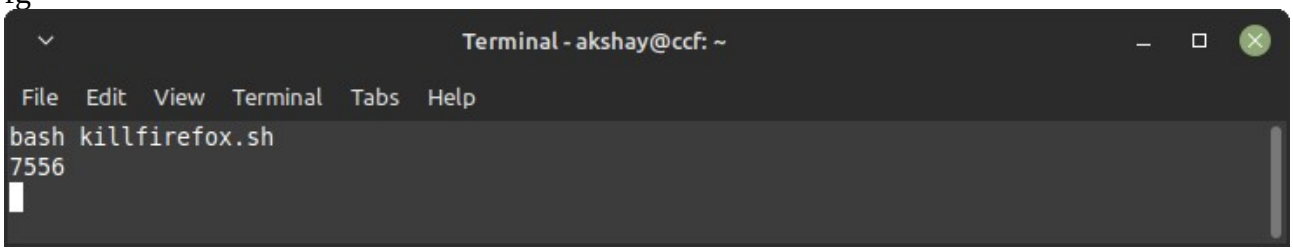
```
turnAroundTime();
printf("pid bt wt tt\n");
for(int i=0;i<n;i++){
printf(" %d %d %d %d\n",P[i].pid,P[i].bt,P[i].wt,P[i].tt);
}
float avg_waitingtime,avg_turnaroundtime;
avg_waitingtime=avg_wt();
avg_turnaroundtime=avg_tt();
printf("Avg waiting time=%f",avg_waitingtime);
printf("\nAvg turnaroundtime time=%f\n",avg_turnaroundtime);
}
#include <stdio.h>
# define MAXSIZE 10
int n,tq;
struct process{
int pid;
int bt;
int wt;
int tt;
int at;
}P[MAXSIZE];
float avg_wt(){
float total=0;
for(int i=0;i<n;i++){
total=total+P[i].wt;
}
return total/n;}
float avg_tt(){
float total=0;
for(int i=0;i<n;i++){
total=total+P[i].tt;
}
return total/n;
}
void main(){
int i,total,x,counter,temp[10];
printf("Enter no.of processes:");
scanf("%d",&n);
x=n;
for(int i=0;i<n;i++){
printf("Enter pid of process %d:",i+1);
scanf("%d",&P[i].pid);
printf("Enter arrival time of process %d:",i+1);
scanf("%d",&P[i].at);
printf("Enter burst time of process %d:",i+1);
scanf("%d",&P[i].bt);
temp[i]=P[i].bt;
}
printf("Enter time quantum:");
scanf("%d",&tq);
for(total=0,i=0;x!=0;){
if(temp[i]<=tq&&temp[i]>0){
```

```
total=total+temp[i];
temp[i]=0;
counter=1;
}
else if(temp[i]>0){
total=total+tq;
temp[i]=temp[i]-tq;
}
if(temp[i]==0&&counter==1){
x--;
P[i].tt=total-P[i].at;
P[i].wt=total-P[i].at-P[i].bt;
counter=0;
}
else if(i==n-1){
i=0;
}
else if(P[i+1].at<=total){
i++;
}
else{
i=0;
}
}
printf("pid bt wt tt\n");
for(int i=0;i<n;i++){printf(" %d %d %d %d\n",P[i].pid,P[i].bt,P[i].wt,P[i].tt);
}
float avg_waitingtime,avg_turnaroundtime;
avg_waitingtime=avg_wt();
avg_turnaroundtime=avg_tt();
printf("Avg waiting time=%f",avg_waitingtime);
printf("\nAvg turnaroundtime time=%f\n",avg_turnaroundtime);
}
#include <stdio.h>
# define MAXSIZE 10
int n;
struct process{
int pid;
int bt;
int wt;
int tt;
int priority;
}P[MAXSIZE],temp;
void sort(){
for(int i=1;i<=n;i++){
for(int j=0;j<n-i;j++){
if(P[j].priority>P[j+1].priority){
temp=P[j];
P[j]=P[j+1];
P[j+1]=temp;
}
}
```

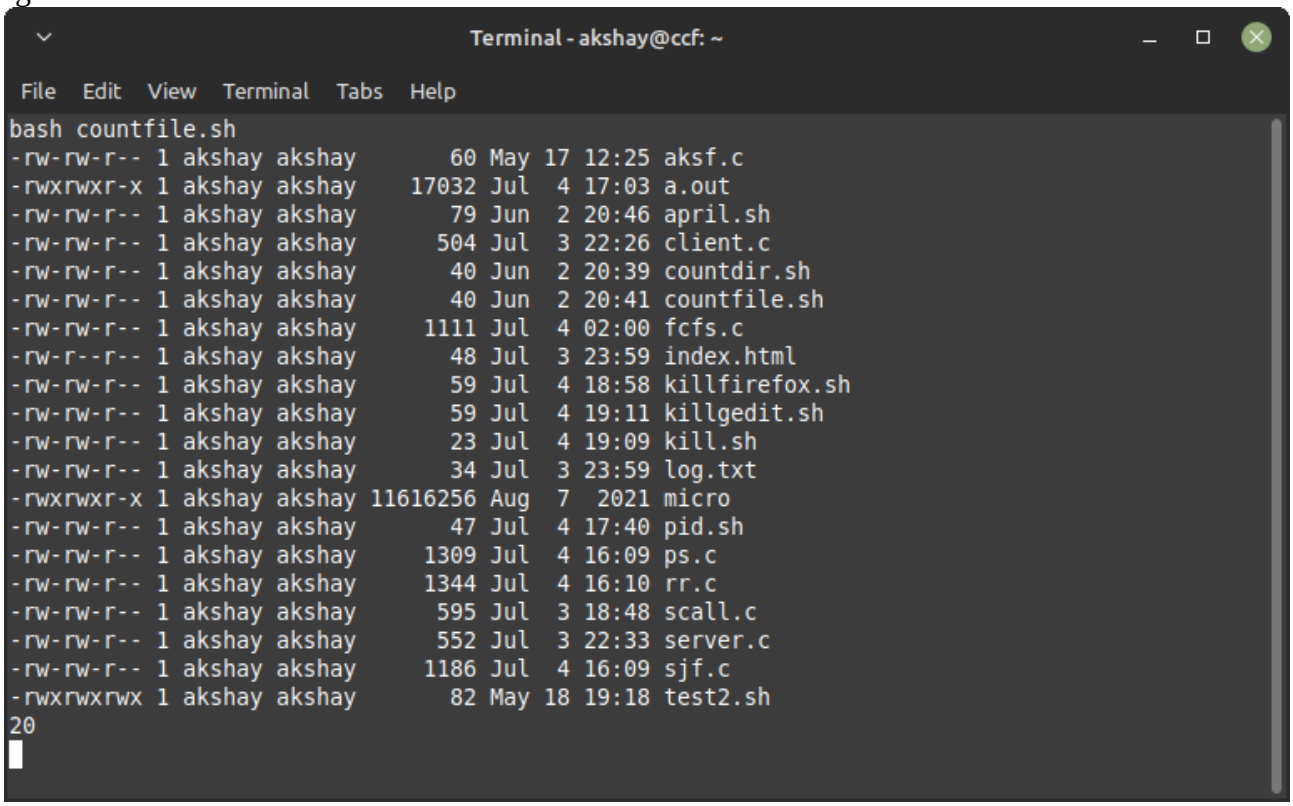
```
}
}
void waitingTime(){
P[0].wt=0;
for(int i=1;i<n;i++){
P[i].wt=P[i-1].wt+P[i-1].bt;
}
}
void turnAroundTime(){
P[0].tt=P[0].bt;
for(int i=1;i<n;i++){
P[i].tt=P[i-1].tt+P[i].bt;
}
}
float avg_wt(){
float total=0;
for(int i=0;i<n;i++){
total=total+P[i].wt;
}
return total/n;
}
float avg_tt(){
float total=0;
for(int i=0;i<n;i++){total=total+P[i].tt;
}
return total/n;
}
void main()
{
printf("Enter no.of processes:");
scanf("%d",&n);
for(int i=0;i<n;i++){
printf("Enter pid of process %d:",i+1);
scanf("%d",&P[i].pid);
printf("Enter burst time of process %d:",i+1);
scanf("%d",&P[i].bt);
printf("Enter priority of process %d:",i+1);
scanf("%d",&P[i].priority);
}
sort();
waitingTime();
turnAroundTime();
printf("pid priority bt wt tt\n");
for(int i=0;i<n;i++){
printf(" %d %d%d %d %d\n",P[i].pid,P[i].priority,P[i].bt,P[i].wt,P[i].tt);
}
float avg_waitingtime,avg_turnaroundtime;
avg_waitingtime=avg_wt();
avg_turnaroundtime=avg_tt();
printf("Avg waiting time=%f",avg_waitingtime);
printf("\nAvg turnaroundtime time=%f\n",avg_turnaroundtime);
}
```


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
bash killgedit.sh
8125

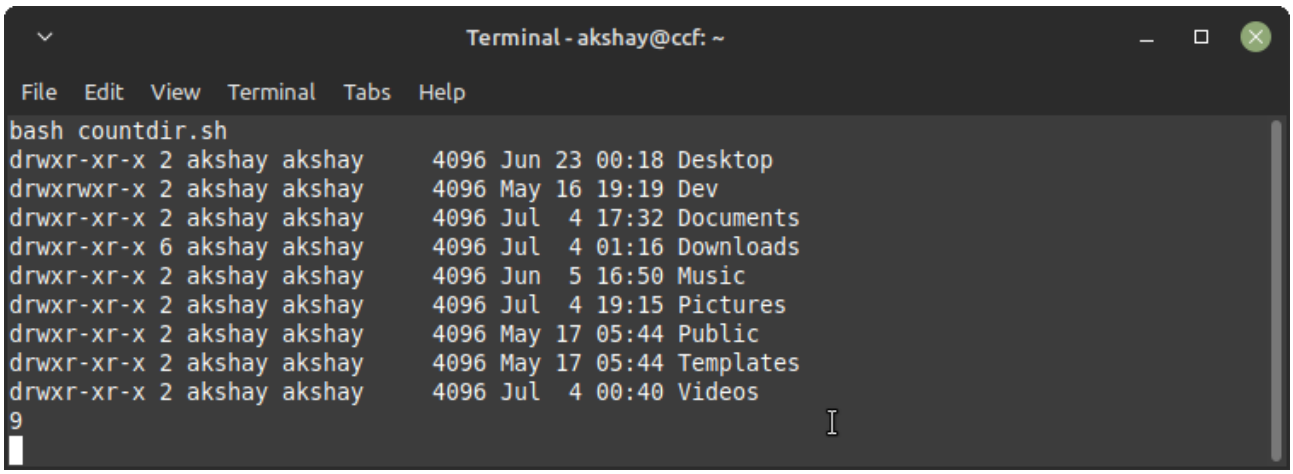
fg


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
bash killfirefox.sh
7556

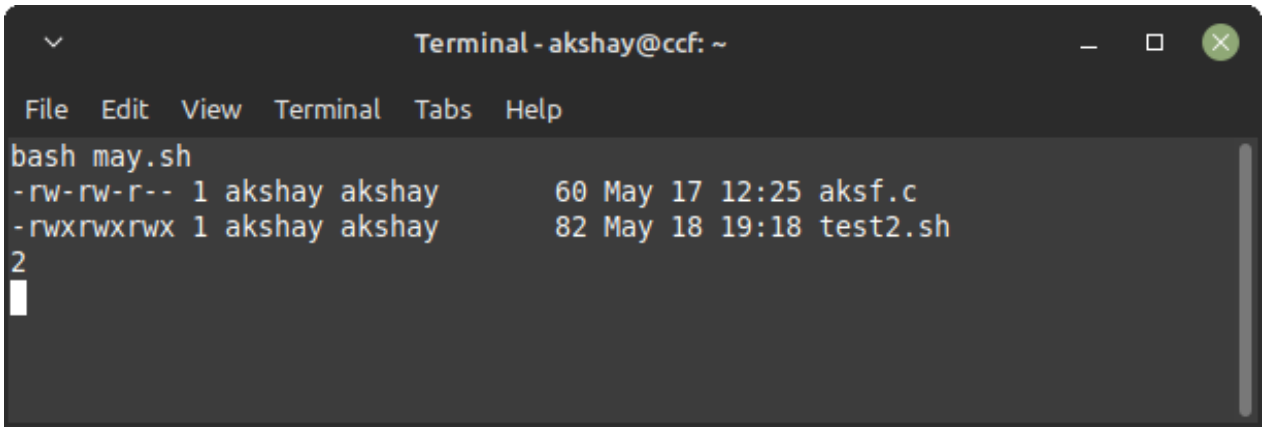
fg


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
bash countfile.sh
-rw-rw-r-- 1 akshay akshay 60 May 17 12:25 aksf.c
-rwxrwxr-x 1 akshay akshay 17032 Jul 4 17:03 a.out
-rw-rw-r-- 1 akshay akshay 79 Jun 2 20:46 april.sh
-rw-rw-r-- 1 akshay akshay 504 Jul 3 22:26 client.c
-rw-rw-r-- 1 akshay akshay 40 Jun 2 20:39 countdir.sh
-rw-rw-r-- 1 akshay akshay 40 Jun 2 20:41 countfile.sh
-rw-rw-r-- 1 akshay akshay 1111 Jul 4 02:00 fcfs.c
-rw-r--r-- 1 akshay akshay 48 Jul 3 23:59 index.html
-rw-rw-r-- 1 akshay akshay 59 Jul 4 18:58 killfirefox.sh
-rw-rw-r-- 1 akshay akshay 59 Jul 4 19:11 killgedit.sh
-rw-rw-r-- 1 akshay akshay 23 Jul 4 19:09 kill.sh
-rw-rw-r-- 1 akshay akshay 34 Jul 3 23:59 log.txt
-rwxrwxr-x 1 akshay akshay 11616256 Aug 7 2021 micro
-rw-rw-r-- 1 akshay akshay 47 Jul 4 17:40 pid.sh
-rw-rw-r-- 1 akshay akshay 1309 Jul 4 16:09 ps.c
-rw-rw-r-- 1 akshay akshay 1344 Jul 4 16:10 rr.c
-rw-rw-r-- 1 akshay akshay 595 Jul 3 18:48 scall.c
-rw-rw-r-- 1 akshay akshay 552 Jul 3 22:33 server.c
-rw-rw-r-- 1 akshay akshay 1186 Jul 4 16:09 sjf.c
-rwxrwxrwx 1 akshay akshay 82 May 18 19:18 test2.sh
20

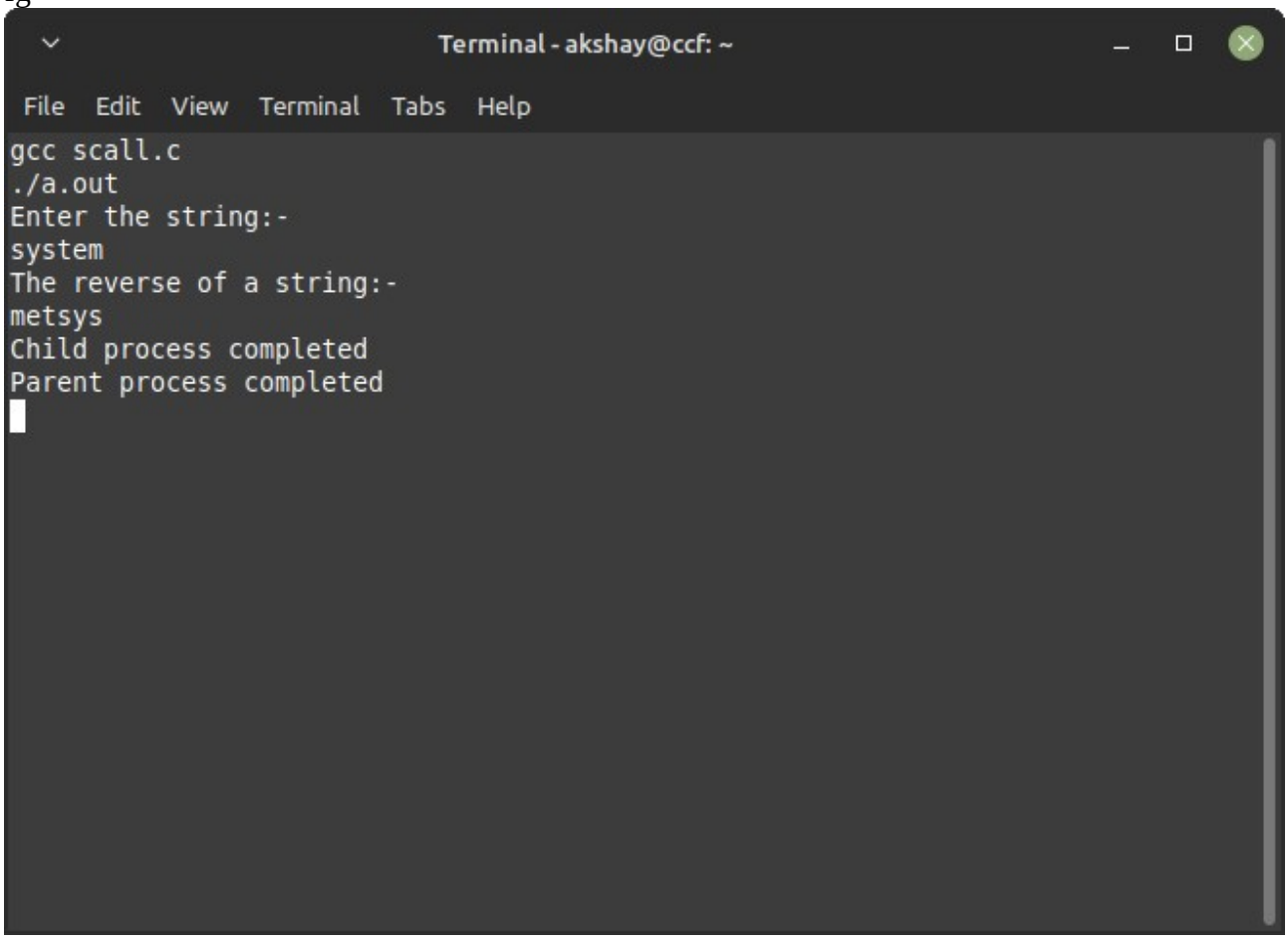
hg


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
bash countdir.sh
drwxr-xr-x 2 akshay akshay 4096 Jun 23 00:18 Desktop
drwxrwxr-x 2 akshay akshay 4096 May 16 19:19 Dev
drwxr-xr-x 2 akshay akshay 4096 Jul 4 17:32 Documents
drwxr-xr-x 6 akshay akshay 4096 Jul 4 01:16 Downloads
drwxr-xr-x 2 akshay akshay 4096 Jun 5 16:50 Music
drwxr-xr-x 2 akshay akshay 4096 Jul 4 19:15 Pictures
drwxr-xr-x 2 akshay akshay 4096 May 17 05:44 Public
drwxr-xr-x 2 akshay akshay 4096 May 17 05:44 Templates
drwxr-xr-x 2 akshay akshay 4096 Jul 4 00:40 Videos
9

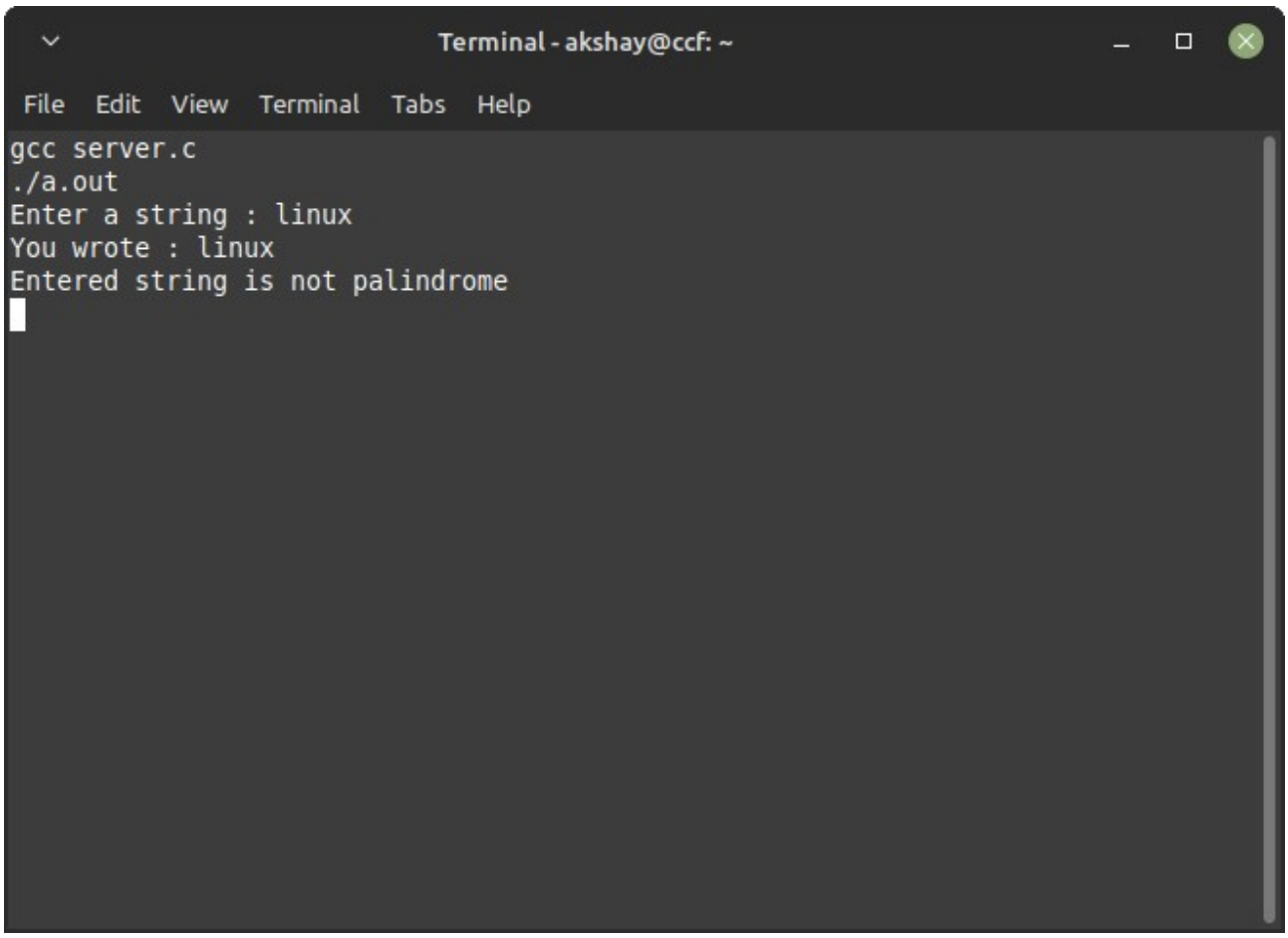
jh


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
bash may.sh
-rw-rw-r-- 1 akshay akshay 60 May 17 12:25 aksf.c
-rwxrwxrwx 1 akshay akshay 82 May 18 19:18 test2.sh
2

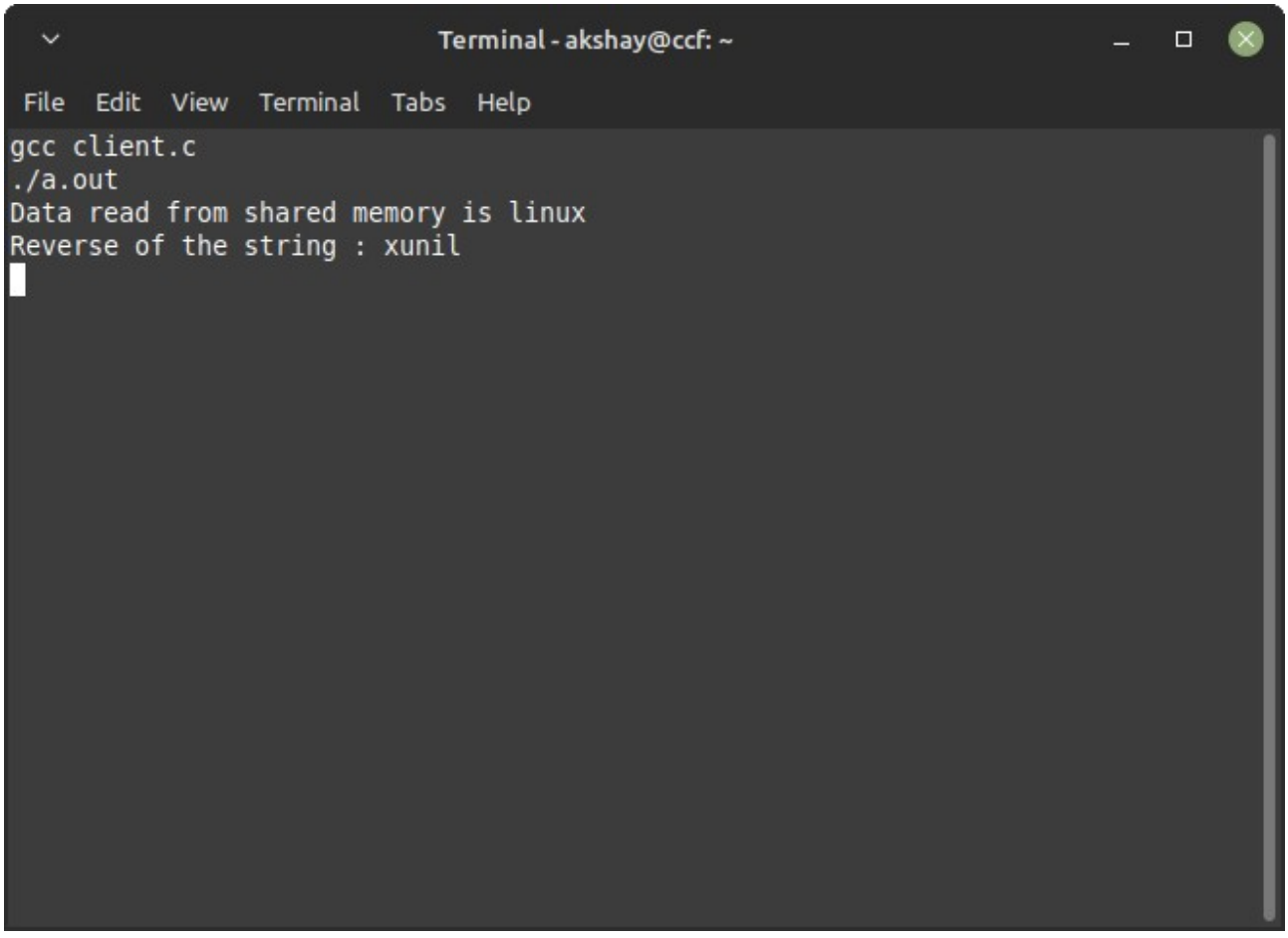
fg


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
gcc scall.c
./a.out
Enter the string:-
system
The reverse of a string:-
metsys
Child process completed
Parent process completed

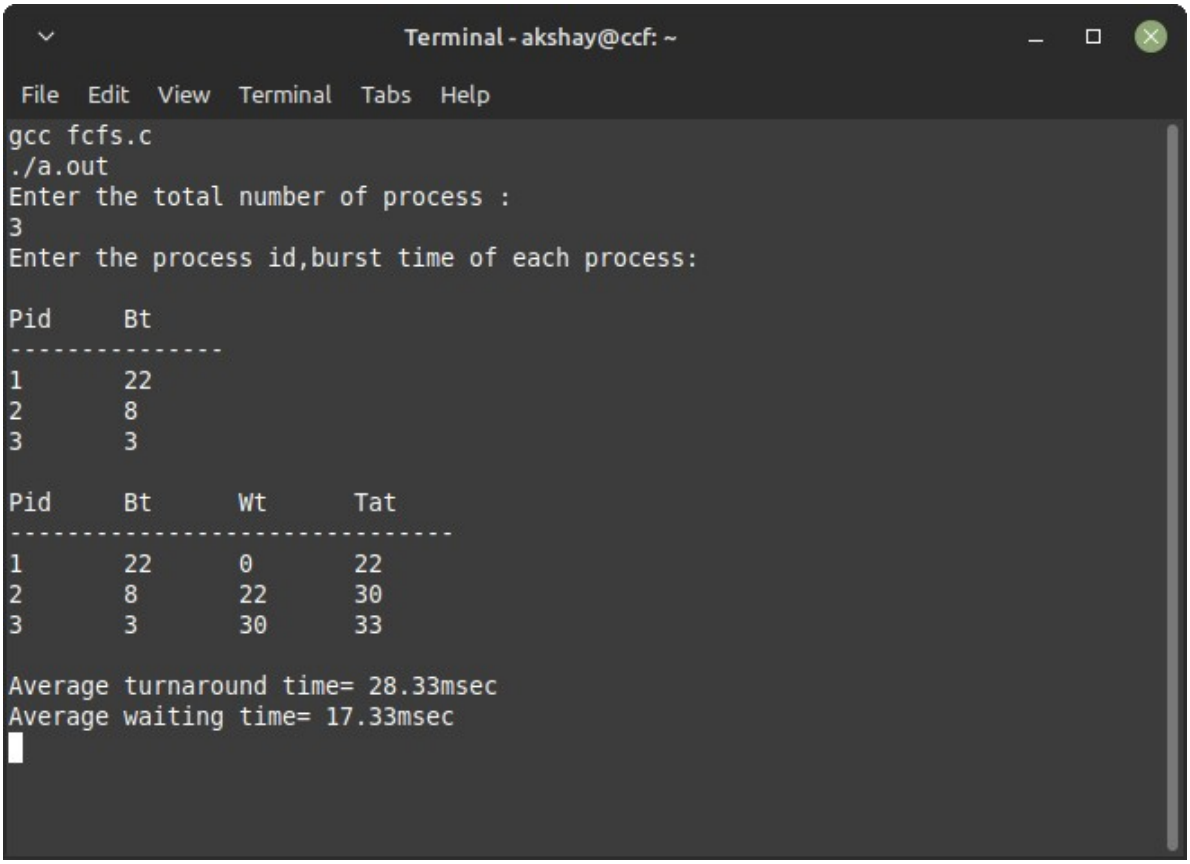
df


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
gcc server.c
./a.out
Enter a string : linux
You wrote : linux
Entered string is not palindrome

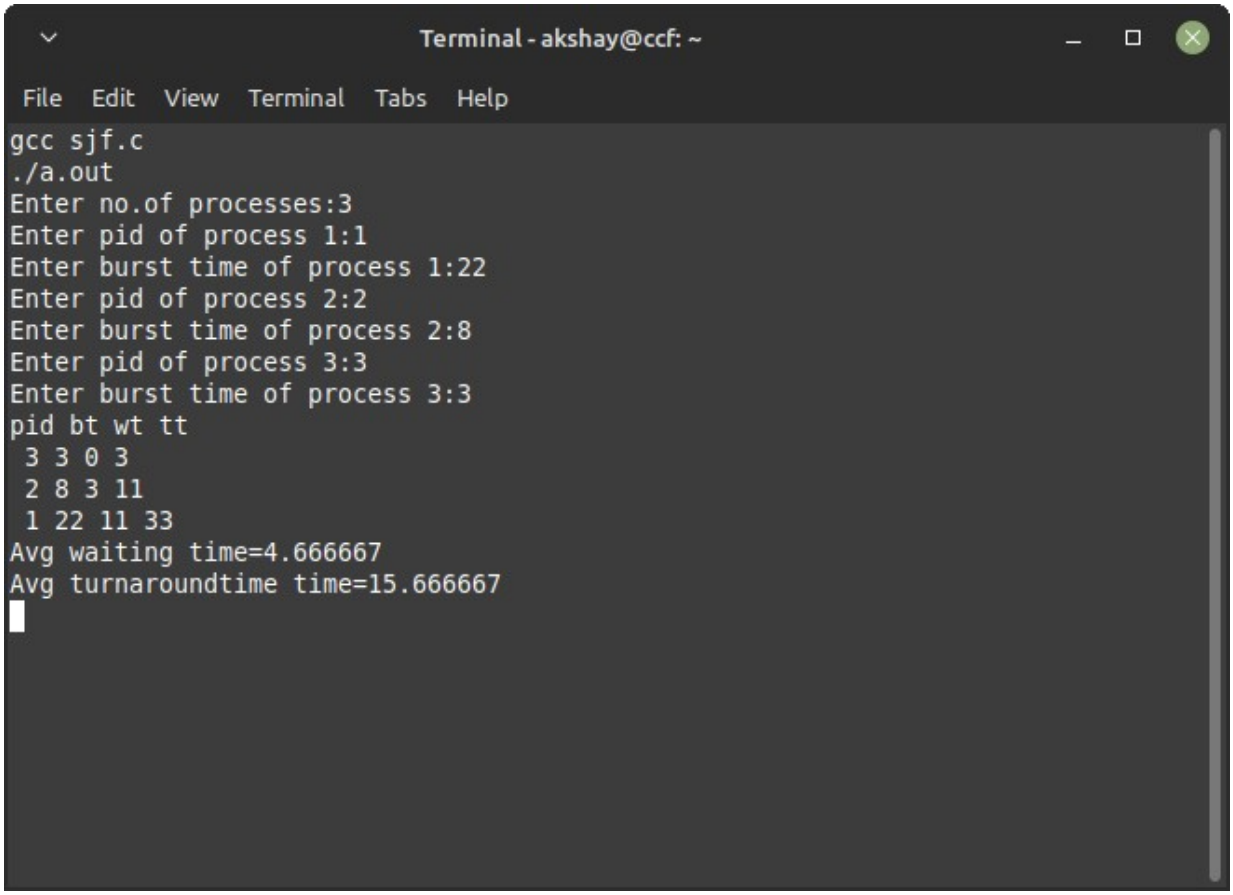
df


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
gcc client.c
./a.out
Data read from shared memory is linux
Reverse of the string : xunil

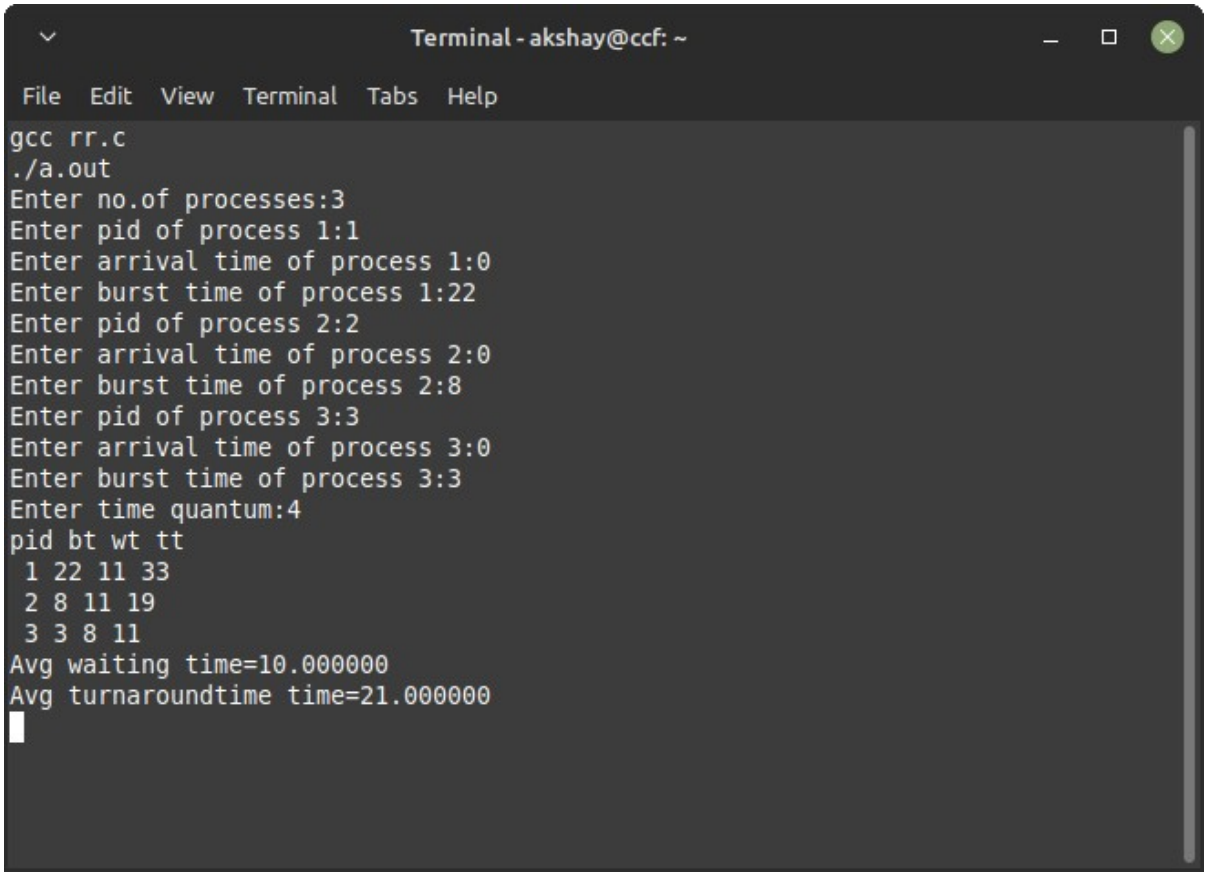
jh


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
gcc fcfs.c
./a.out
Enter the total number of process :
3
Enter the process id,burst time of each process:
Pid Bt
1 22
2 8
3 3
Pid Bt Wt Tat
1 22 0 22
2 8 22 30
3 3 30 33
Average turnaround time= 28.33msec
Average waiting time= 17.33msec

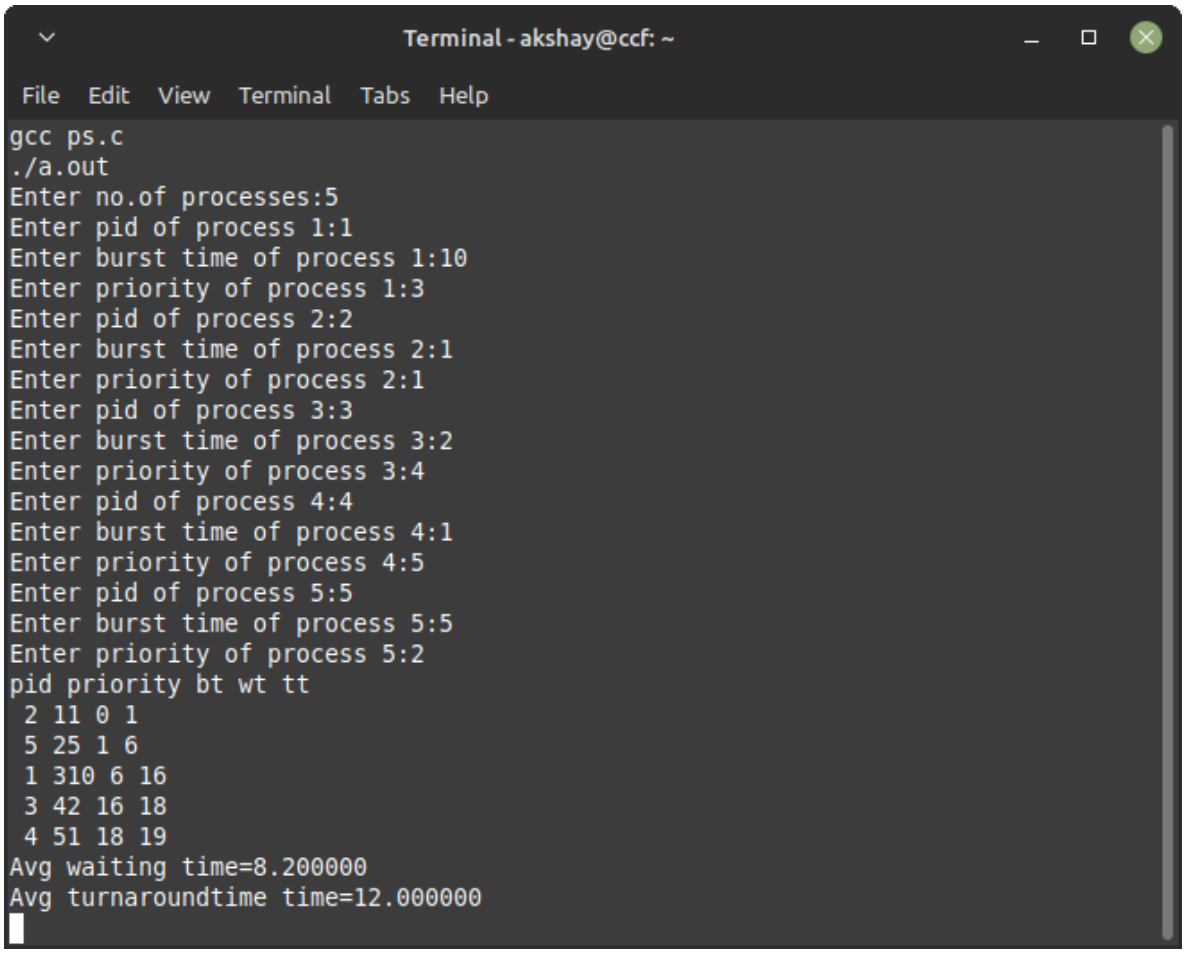
hj


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
gcc sjf.c
./a.out
Enter no.of processes:3
Enter pid of process 1:1
Enter burst time of process 1:22
Enter pid of process 2:2
Enter burst time of process 2:8
Enter pid of process 3:3
Enter burst time of process 3:3
pid bt wt tt
3 3 0 3
2 8 3 11
1 22 11 33
Avg waiting time=4.666667
Avg turnaroundtime time=15.666667

mj


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
gcc rr.c
./a.out
Enter no.of processes:3
Enter pid of process 1:1
Enter arrival time of process 1:0
Enter burst time of process 1:22
Enter pid of process 2:2
Enter arrival time of process 2:0
Enter burst time of process 2:8
Enter pid of process 3:3
Enter arrival time of process 3:0
Enter burst time of process 3:3
Enter time quantum:4
pid bt wt tt
1 22 11 33
2 8 11 19
3 3 8 11
Avg waiting time=10.000000
Avg turnaroundtime time=21.000000

hj


Terminal - akshay@ccf: ~
File Edit View Terminal Tabs Help
gcc ps.c
./a.out
Enter no.of processes:5
Enter pid of process 1:1
Enter burst time of process 1:10
Enter priority of process 1:3
Enter pid of process 2:2
Enter burst time of process 2:1
Enter priority of process 2:1
Enter pid of process 3:3
Enter burst time of process 3:2
Enter priority of process 3:4
Enter pid of process 4:4
Enter burst time of process 4:1
Enter priority of process 4:5
Enter pid of process 5:5
Enter burst time of process 5:5
Enter priority of process 5:2
pid priority bt wt tt
2 11 0 1
5 25 1 6
1 310 6 16
3 42 16 18
4 51 18 19
Avg waiting time=8.200000
Avg turnaroundtime time=12.000000