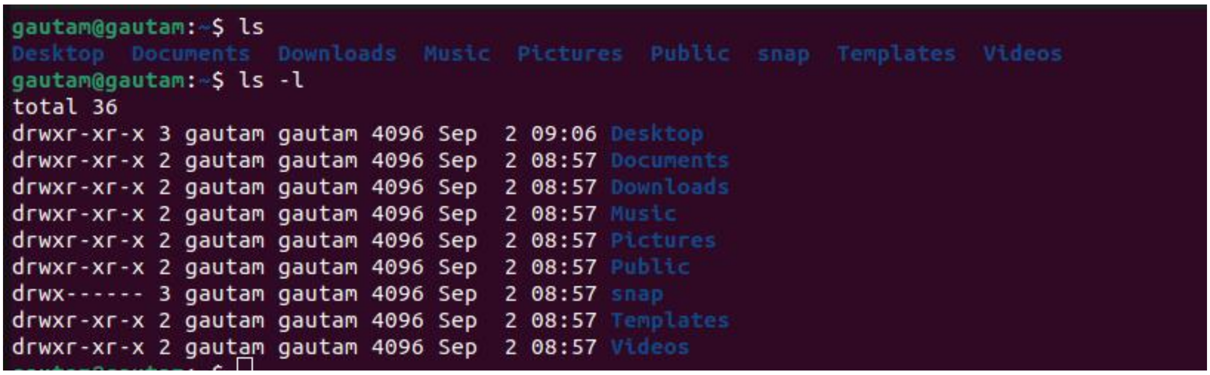
Atma Ram Sanatan Dharma College

University Of Delhi

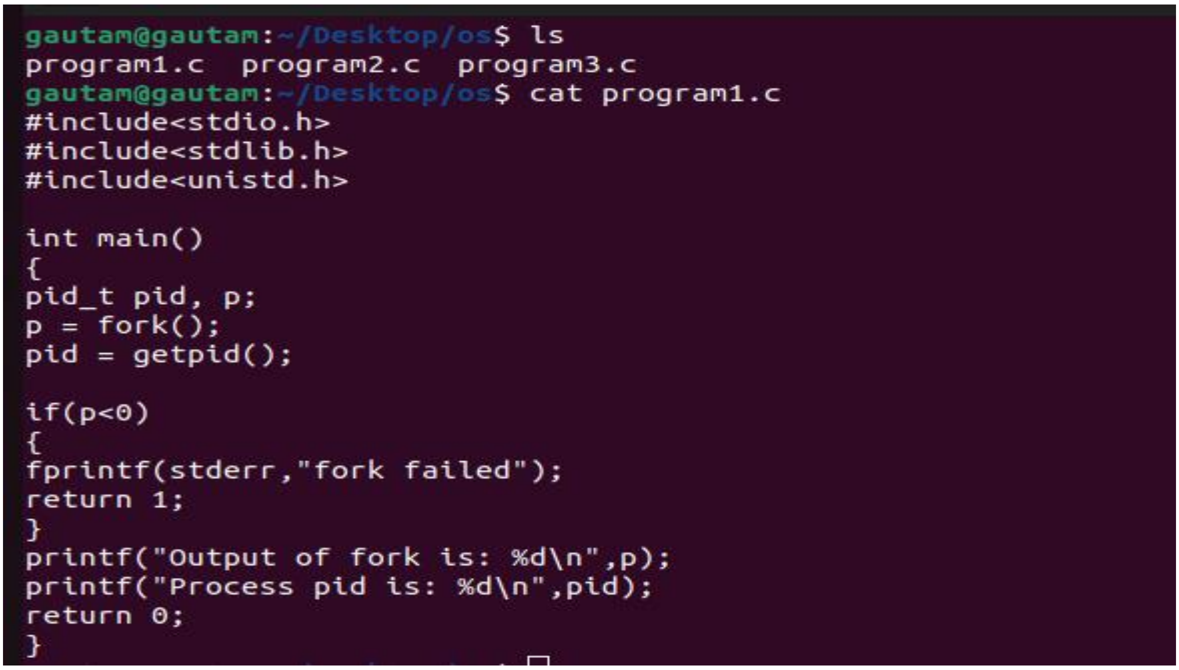
Name	Neeraj
Roll no.	18088
Course	B.sc.(H)computer science
Assignment	Practical Question 1
Submitted to	**Dr. Parul Jain**

ls:- list item **ls-l**:- list items in detail



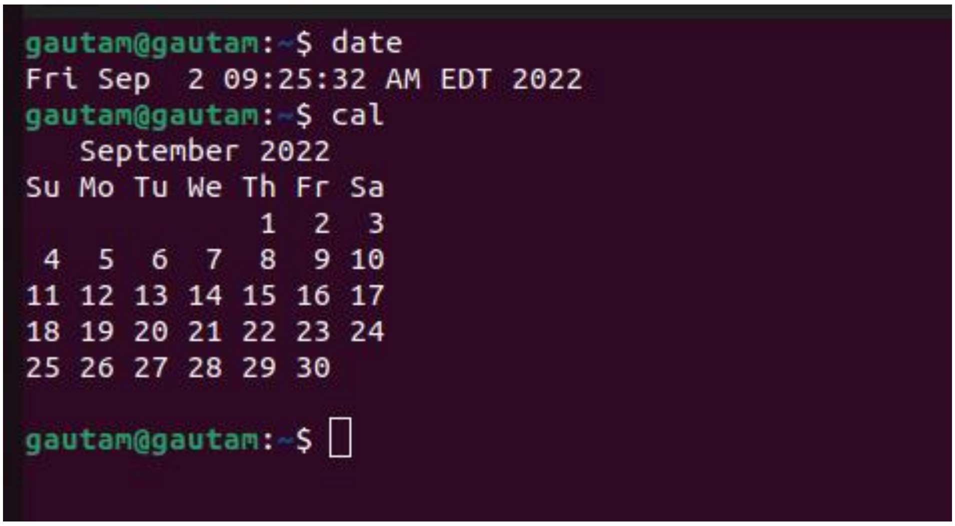
```
gautam@gautam:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
gautam@gautam:~$ ls -l
total 36
drwxr-xr-x 3 gautam gautam 4096 Sep  2 09:06 Desktop
drwxr-xr-x 2 gautam gautam 4096 Sep  2 08:57 Documents
drwxr-xr-x 2 gautam gautam 4096 Sep  2 08:57 Downloads
drwxr-xr-x 2 gautam gautam 4096 Sep  2 08:57 Music
drwxr-xr-x 2 gautam gautam 4096 Sep  2 08:57 Pictures
drwxr-xr-x 2 gautam gautam 4096 Sep  2 08:57 Public
drwx------ 3 gautam gautam 4096 Sep  2 08:57 snap
drwxr-xr-x 2 gautam gautam 4096 Sep  2 08:57 Templates
drwxr-xr-x 2 gautam gautam 4096 Sep  2 08:57 Videos
```

cat:- this command is used for display content of file.



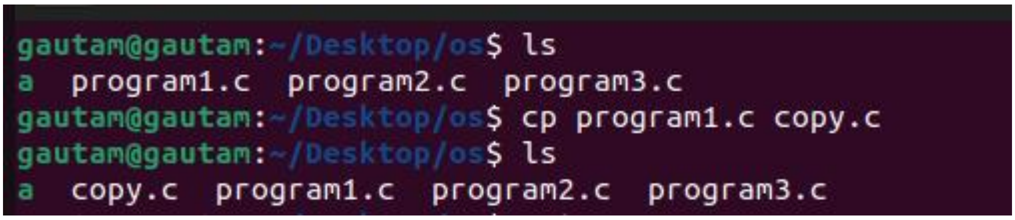
```
gautam@gautam:~/Desktop/os$ ls
program1.c  program2.c  program3.c
gautam@gautam:~/Desktop/os$ cat program1.c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main()
{
pid_t pid, p;
p = fork();
pid = getpid();

if(p<0)
{
fprintf(stderr,"fork failed");
return 1;
}
printf("Output of fork is: %d\n",p);
printf("Process pid is: %d\n",pid);
return 0;
}
```

date and **cal**:- these command show the date and calender.



```
gautam@gautam:~$ date
Fri Sep  2 09:25:32 AM EDT 2022
gautam@gautam:~$ cal
   September 2022
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

gautam@gautam:~$
```

cp:- copy the content of one file to another



```
gautam@gautam:~/Desktop/os$ ls
a  program1.c  program2.c  program3.c
gautam@gautam:~/Desktop/os$ cp program1.c copy.c
gautam@gautam:~/Desktop/os$ ls
a  copy.c  program1.c  program2.c  program3.c
```

Q1. Write a program (using fork() and/or exec() commands) where parent and child execute:

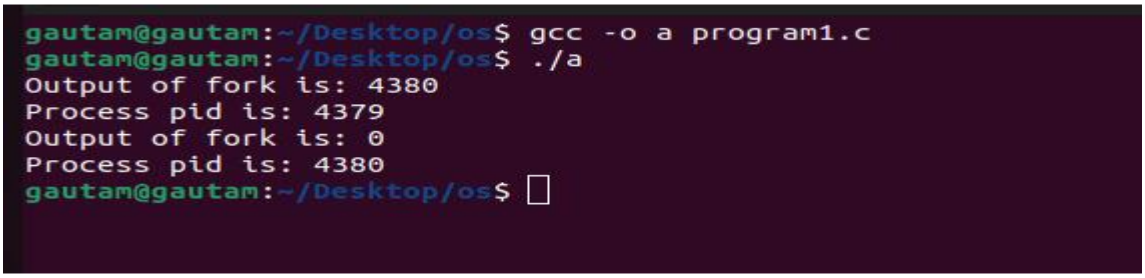
a) Same program, same code

Answer

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

int main()
{
 pid_t pid, p; //creating variable of type pid to store the pid of the process

     p = fork(); //fork command is used to create the child process

     pid = getpid(); //getpid command is used to get the pid of the process

     if(p<0)//pid value of a child process is 0(zero)
     {
     fprintf(stderr,"fork faild");
     return 1;
```

```
    }

    printf("Output of fork id: %d\n",p);
    printf("Process id is: %d\n",pid);
    return 0;
}
```

Output:-



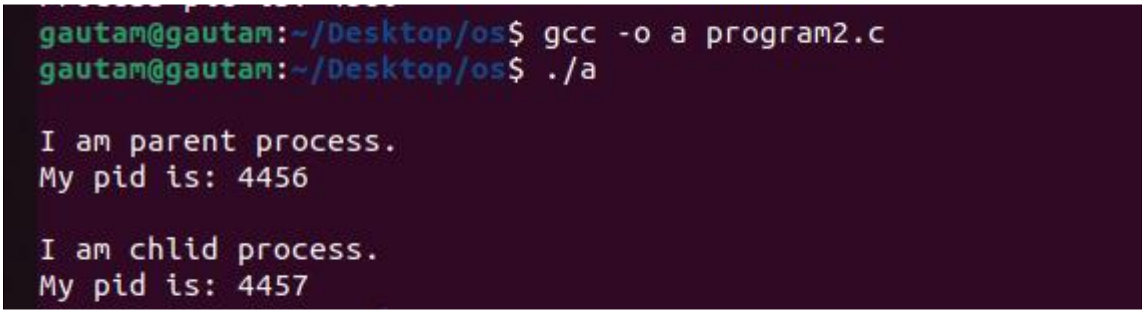
```
gautam@gautam:~/Desktop/os$ gcc -o a program1.c
gautam@gautam:~/Desktop/os$ ./a
Output of fork is: 4380
Process pid is: 4379
Output of fork is: 0
Process pid is: 4380
gautam@gautam:~/Desktop/os$ 
```

b) Same program, different code

Answer

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

int main()
{
pid_t  p; //creating variable of type pid to store the pid of the process

p = fork(); //fork command is used to create the child process


    if(p<0)
    {
    fprintf(stderr,"fork faild");
    exit(1);
    }
    else if(p == 0)//pid value of a child process is 0(zero)
    {
    printf("\nI am child process.");
    printf("\nMy pid is: %d\n",getpid());//getpid command is used to get the
pid of the process
    exit(0);
    }
    else
    {
    printf("\nI am parent process.");
    printf("\nMy pid is: %d\n",getpid());
    exit(0);
    }
return 0;
}
```

Output:-



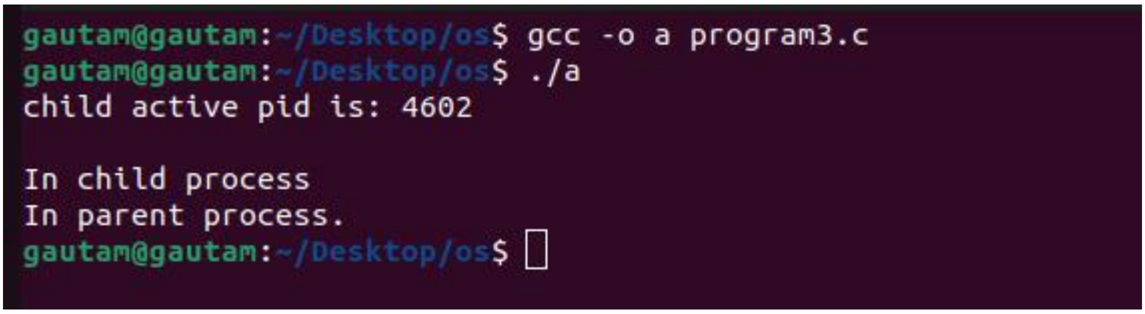
```
gautam@gautam:~/Desktop/os$ gcc -o a program2.c
gautam@gautam:~/Desktop/os$ ./a

I am parent process.
My pid is: 4456

I am chlid process.
My pid is: 4457
```

c) Before terminating, the parent waits for the child to finish its task.

Answer

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/wait.h>

int main()
{
pid_t  p; //creating variable of type pid to store the pid of the process

p = fork(); //fork command is used to create the child process


    if(p<0)
    {
    fprintf(stderr,"fork faild");
    exit(1);
    }
    else if(p == 0)//pid value of a child process is 0(zero)
    {
    printf("\nIn child process.");
    exit(0);
    }
    else
    {
    wait(NULL);
    printf("\nchild active pid is: %d\n",getpid());//getpid command is used to
get the pid of the process
    printf("\nIn parent process.\n");
    exit(0);
    }
return 0;
}
```

Output:-



```
gautam@gautam:~/Desktop/os$ gcc -o a program3.c
gautam@gautam:~/Desktop/os$ ./a
child active pid is: 4602

In child process
In parent process.
gautam@gautam:~/Desktop/os$ 
```