Title: Interprocess Communication

Objectives

- To learn about IPC
- To learn about concepts and uses of forks, pipe

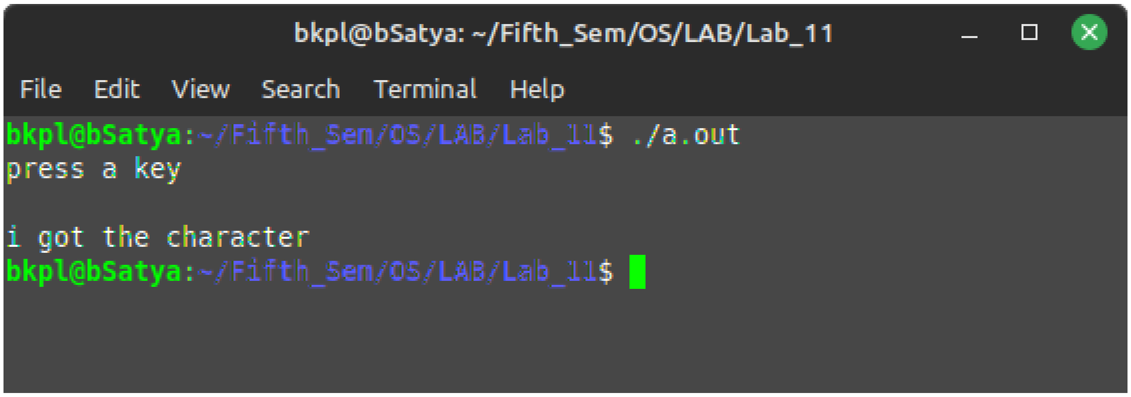
Background

Interprocess Communication:
IPC stands for inter-process communication. Well, inter-process communication is very important in UNIX. It might not look as trivial but let's start with the following problem.
Suppose we have two processes running in memory, a child process, and a parent process. The requirement is such that the parent has to wait till a key is pressed from within the child and then the parent has to exit.

Lab Activities

1.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
void main()
{
 char c;
int exflag=0;
printf("press a key\n");
scanf("%c",&c);
if(!fork()){
exflag=1;
exit(0);
}
else{
while(!exflag){
printf("i got the character");
exit(0);
}}
}
```



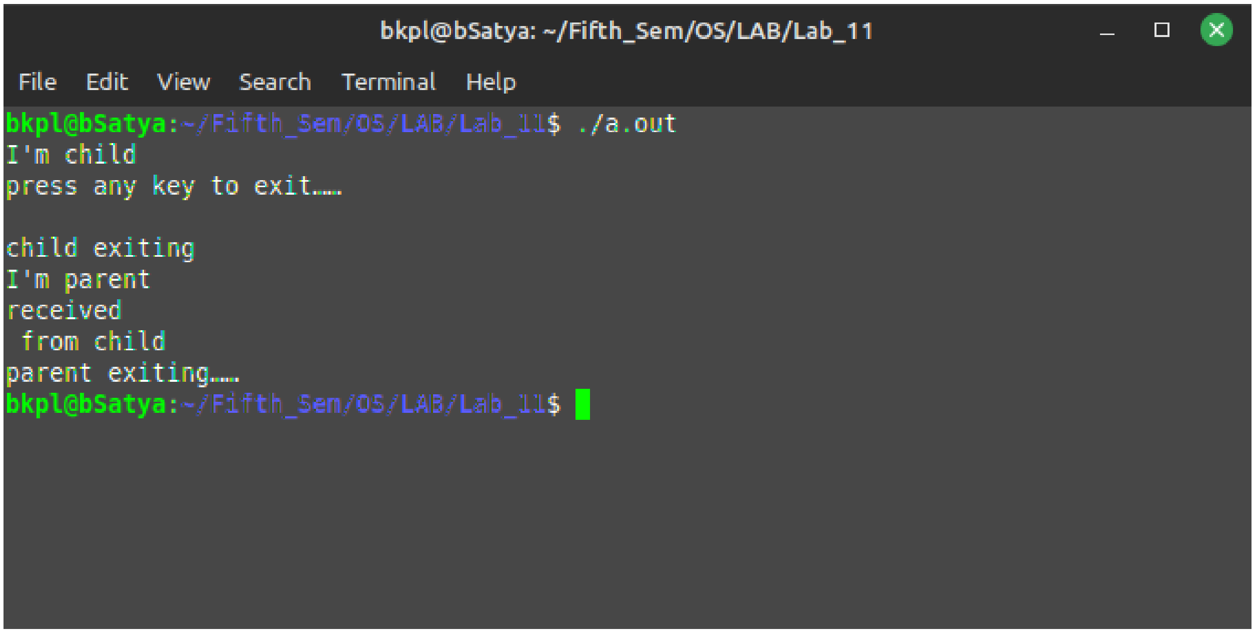
```
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$ ./a.out
press a key

i got the character
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$ 
```

2.

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

void main()
{
 int pfd[2];
if(pipe(pfd)<0)
printf("error");
if(!fork()){
char data;
printf("I'm child");
printf("press any key to exit……");
scanf("%c",&data);
write(pfd[1],&data,1);
printf("child exiting");
}
else{
char data;
read(pfd[0],&data,1);
printf("I'm parent");
printf("received %c from child",data);
printf("parent exiting……\n");
exit(0);
}
}
```

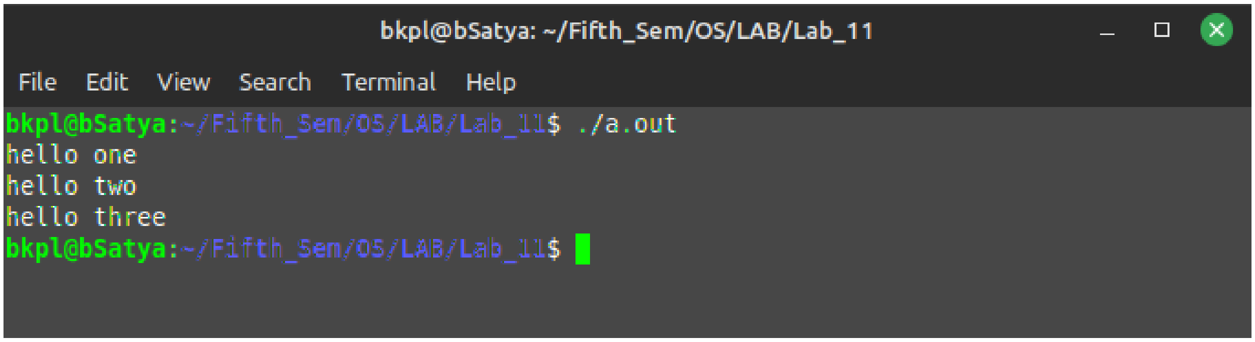


```
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$ ./a.out
I'm child
press any key to exit.....

child exiting
I'm parent
received
 from child
parent exiting.....
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$ 
```

3.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#define msgsz 16
void main()
{
char *msg1= ("hello one");
char *msg2= ("hello two");
char *msg3= ("hello three");
char inbuf[msgsz];
int p[2],j;
pipe(p);
write(p[1],msg1,msgsz);
write(p[1],msg2,msgsz);
write(p[1],msg3,msgsz);
for(j=0;j<3;j++){
read(p[0],inbuf,msgsz);
printf("%s\n",inbuf);
}
exit(0);
}
```

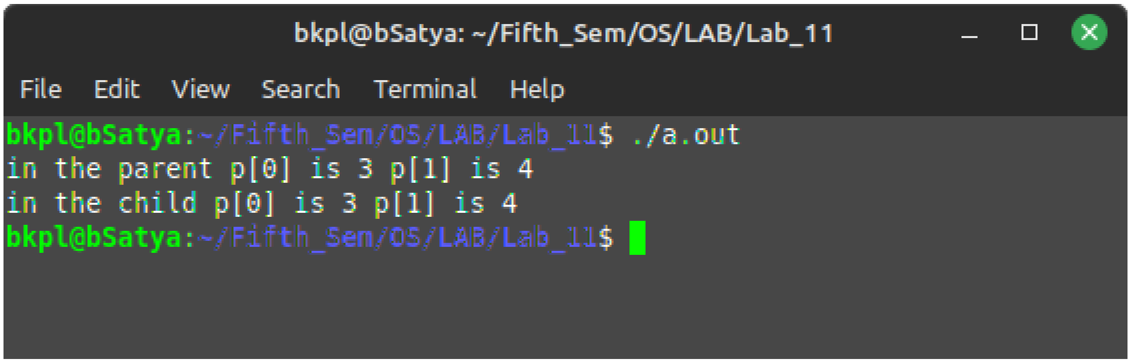

bkpl@bSatya: ~/Fifth_Sem/OS/LAB/Lab_11
File Edit View Search Terminal Help

```
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$ ./a.out
hello one
hello two
hello three
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$
```

4.

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

void main()
{
int p[2],pid;
pipe(p);
pid=fork();
if(pid==0)
printf("in the child p[0] is %d p[1] is %d\n",p[0],p[1]);
else
printf("in the parent p[0] is %d p[1] is %d\n",p[0],p[1]);
}
```

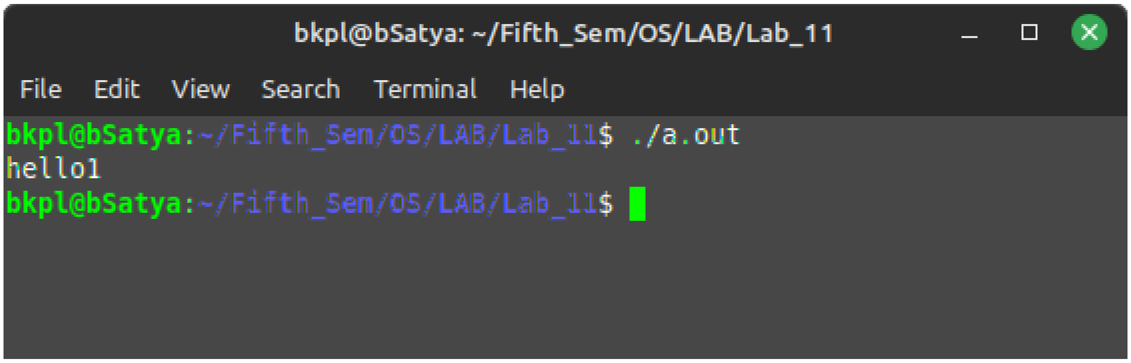

bkpl@bSatya: ~/Fifth_Sem/OS/LAB/Lab_11
File Edit View Search Terminal Help

```
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$ ./a.out
in the parent p[0] is 3 p[1] is 4
in the child p[0] is 3 p[1] is 4
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$
```

5.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#define MSGSZ 16
 main() {
```

```
char *msg="hello1";
char inbuf[MSGSZ];
int p[2],pid,j;
pipe(p);
pid=fork();
 if(pid>0) {
close(p[0]);
write(p[1],msg,MSGSZ);
}
 if(pid==0) {
 close(p[1]);
read(p[0],inbuf,MSGSZ);
printf("%s\n",inbuf);
 }
 exit(0);
 }
```



```
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$ ./a.out
hello1
bkpl@bSatya:~/Fifth_Sem/OS/LAB/Lab_11$
```

Conclusion:

- Learned about IPC
- Learned about the uses of fork and piping.