

Submission Guidelines

Name of the File

:20BCE1798_CSE2005_Ex8_InterCommunicationProcess

Required Contents: Virtual Box

Reg No:20BCE1798

Name: ANSH GOEL

Course: CSE2005-Operating Systems (Embedded Lab)

Slot: L27+L28

Ex No 2: Shell Programming

Date: 6th April,2022

1. Write a program which creates a child process and continues to run along with its child (choose any small task of your own). Once the child completes its task, it should send a signal to parent which in turn terminates the parent. (Expected output: output of the task carried out by the child process, termination of parent)
2. Write two c programs: One displaying the PID infinitely and the other program sending a signal to terminate the first program.(Note: Execute the programs in separate terminals)

Q1)

CODE:

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Signals_Ex.c". The code in the terminal is as follows:

```
1 #include <stdio.h>
2 #include <signal.h>
3 #include<unistd.h>
4 #include <stdlib.h>
5 void sigt()
6 {
7     exit(0);
8 }
9 int main()
10 {
11     int pid,parentId;
12     parentId=getpid();
13     pid=fork();
14     if (pid==0)
15     {
16         for (int i=0; i<5; i++)
17         {
18             printf("%d\n", i);
19             sleep (1);
20         }
21         kill(parentId, SIGINT);
22         exit(0);
23     }
24     else
25     {
26         // printf("\n");
27         for(int i=6; i<90; i++)
28         {
29             printf("%c\n", (char) i);
30             sleep (1);
31             signal (SIGINT, sigt);
32     }
33 }
34 }
35 }
```

The terminal window shows the output of the program, which prints the numbers 0 through 5 followed by the characters A through F.

OUTPUT:

The screenshot shows a terminal window with the title "ansh@ansh: ~". The terminal output is as follows:

```
ansh@ansh:~$ gcc Signals_Ex.c
ansh@ansh:~$ ./a.out
A
0
B
1
C
2
D
E
F
```

Q2)

CODE:

The image shows two terminal windows side-by-side. The left window is titled 'Ender.c' and contains the following C code:

```
1 #include <stdio.h>
2 #include <signal.h>
3 #include <unistd.h>
4 int main()
5 {
6     int pid;
7     printf("Enter the process id to kill \n");
8     scanf("%d",&pid);
9     kill (pid, SIGQUIT);
10}
11
```

The right window is titled 'Printer.c' and contains the following C code:

```
1 #include <stdio.h>
2 #include <signal.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 void sigit()
6 {
7     exit(0);
8 }
9 int main()
10{
11    int pid,parentId;
12    parentId=getpid();
13    signal (SIGQUIT, sigit);
14    while (1)
15    {
16        printf("process id is : %d\n", parentId);
17        sleep (1);
18    }
19}
```

OUTPUT:

The image shows a single terminal window with two sessions. The left session shows the execution of 'Printer.c':

```
ansh@ansh:~$ gcc Printer.c
ansh@ansh:~$ ./a.out
process id is : 4099
ansh@ansh:~$
```

The right session shows the execution of 'Ender.c':

```
ansh@ansh:~$ gcc Ender.c
ansh@ansh:~$ ./a.out
Enter the process id to kill
4099
ansh@ansh:~$
```