

Naming the file: 20BCE1798\_EX-5\_PROCESSES

Reg No: **20BCE1798**

Name: **ANSH GOEL**

Course Code: CSE2005

Course Name: Operating Systems (Embedded Lab)

Slot: L27+L28

Ex No 5: PROCESSES

Date: 11-02-22

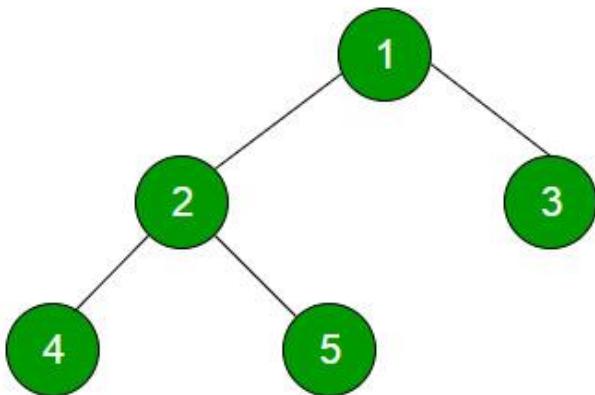
QUESTION:

Ex No 5

Date: 11-02-21

**Processes**

Create processes 1,2,3,4, and 5 as specified in the tree.



Print I am 1 from process 1, I am 2 from process 2..... Till process 5.

CODE FOR THE ABOVE QUESTION:

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/wait.h>
4 #include <unistd.h>
5 int
6 main }()
7 {
8     pid_t p1, p2, p3, p4;
9     p1 = fork ();
10    if (p1 < 0)
11        printf ("Fork Failure\n");
12    else if (p1 == 0)
13    {
14        printf ("I am 2 from Process 2\n");
15        p3 = fork ();
16        if (p3 < 0)
17            printf ("Fork Failure\n");
18        else if (p3 > 0)
19        {
20            wait (NULL);
21            //printf("I am Process 2\n");
22            p4 = fork ();
23            if (p4 < 0)
24                printf ("Fork Failure");
25            else if (p4 > 0)
26            {
27                wait (NULL);
28                //printf("I am Process 2\n");
29            }
30            else if (p4 == 0)
31                printf ("I am 5 from Process 5\n");
32        }
33        else if (p3 == 0)
34            printf ("I am 4 from Process 4\n");
35    }
36    else if (p1 > 0)
37    {
38        wait (NULL);
39        printf ("I am 1 from Process 1\n");
40        p2 = fork ();
41        if (p2 < 0)
42            printf ("Fork Failure\n");
43        else if (p2 > 0)
44        {
45            wait (NULL);
46            //printf("I am Process 1\n");
47        }
48        else if (p2 == 0)
49            printf ("I am 3 from Process 3\n");
50    }
51 }
```

## OUTPUT:

The screenshot shows a terminal window with the following text output:

```
I am 2 from Process 2
I am 4 from Process 4
I am 5 from Process 5
I am 1 from Process 1
I am 3 from Process 3

...Program finished with exit code 0
Press ENTER to exit console.
```

The terminal window is titled "input". To the right of the terminal, there is a vertical toolbar with several icons, including "Breakpoints and Watchpoints". The system tray at the bottom of the screen displays various application icons and system status information, including the date and time (22/02/2022, 20:27).

I took some reference form this code:

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/wait.h>
4 #include <unistd.h>
5 int
6 main ()
7 {
8     pid_t p1, p2, p3, p4;
9     p1 = fork ();
10    if (p1 < 0)
11        printf ("Fork Failure\n");
12    else if (p1 == 0)
13    {
14        printf ("Child Process %d Parent %d\n", getpid (), getppid ());
15        p3 = fork ();
16        if (p3 < 0)
17            printf ("Fork Failure\n");
18        else if (p3 > 0)
19        {
20            wait (NULL);
21            printf ("Parent Process %d\n", getpid ());
22            p4 = fork ();
23            if (p4 < 0)
24                printf ("Fork Failure");
25            else if (p4 > 0)
26            {
27                wait (NULL);
28                printf ("Parent Process %d\n", getpid ());
29            }
30            else if (p4 == 0)
31                printf ("Child Process %d Parent %d\n", getpid (), getppid ());
32        }
33        else if (p3 == 0)
34            printf ("Child Process %d Parent %d\n", getpid (), getppid ());
35    }
36    else if (p1 > 0)
37    {
38        wait (NULL);
39        printf ("Parent Process %d\n", getpid ());
40        p2 = fork ();
41        if (p2 < 0)
42            printf ("Fork Failure");
43        else if (p2 > 0)
44        {
45            wait (NULL);
46            printf ("Parent Process %d\n", getpid ());
47        }
48        else if (p2 == 0)
49            printf ("Child Process %d Parent %d\n", getpid (), getppid ());
50    }
51 }
```