

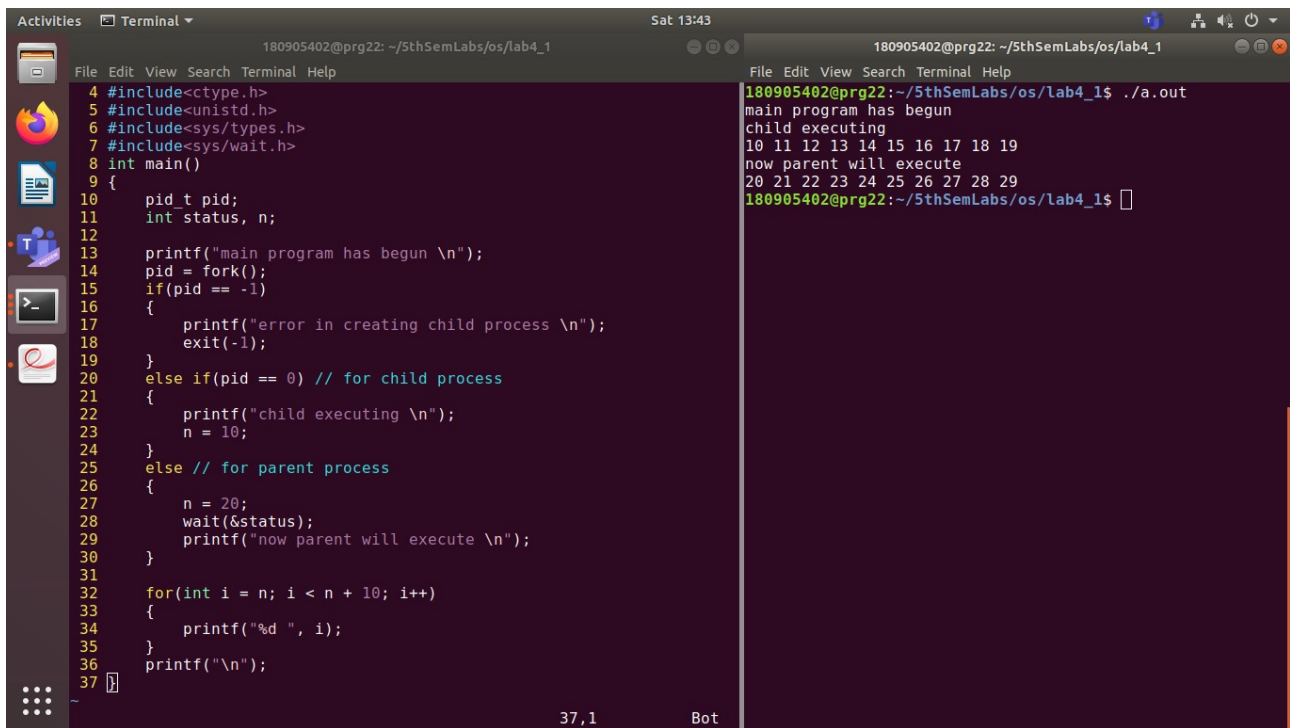
## LAB 3: Processes and Signals

Q1)

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
int main()
{
    pid_t pid;
    int status, n;

    printf("main program has begun \n");
    pid = fork();
    if(pid == -1)
    {
        printf("error in creating child process \n");
        exit(-1);
    }
    else if(pid == 0) // for child process
    {
        printf("child executing \n");
        n = 10;
    }
    else // for parent process
    {
        n = 20;
        wait(&status);
        printf("now parent will execute \n");
    }

    for(int i = n; i < n + 10; i++)
    {
        printf("%d ", i);
    }
    printf("\n");
}
```



```
180905402@prg22: ~/5thSemLabs/os/lab4_1
File Edit View Search Terminal Help
4 #include<ctype.h>
5 #include<unistd.h>
6 #include<sys/types.h>
7 #include<sys/wait.h>
8 int main()
9 {
10     pid_t pid;
11     int status, n;
12
13     printf("main program has begun \n");
14     pid = fork();
15     if(pid == -1)
16     {
17         printf("error in creating child process \n");
18         exit(-1);
19     }
20     else if(pid == 0) // for child process
21     {
22         printf("child executing \n");
23         n = 10;
24     }
25     else // for parent process
26     {
27         n = 20;
28         wait(&status);
29         printf("now parent will execute \n");
30     }
31
32     for(int i = n; i < n + 10; i++)
33     {
34         printf("%d ", i);
35     }
36     printf("\n");
37 }
```

```
180905402@prg22: ~/5thSemLabs/os/lab4_1$ ./a.out
main program has begun
child executing
10 11 12 13 14 15 16 17 18 19
now parent will execute
20 21 22 23 24 25 26 27 28 29
180905402@prg22: ~/5thSemLabs/os/lab4_1$
```

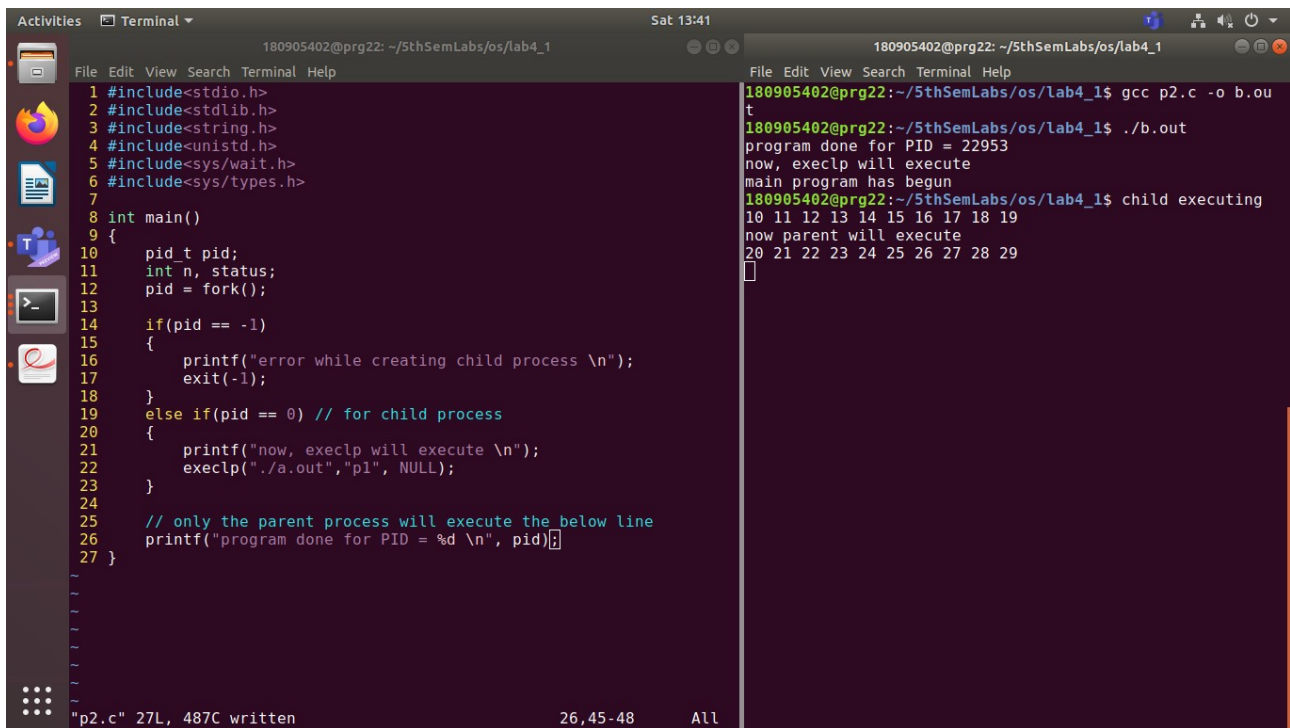
Q2)

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

```
int main()
{
    pid_t pid;
    int n, status;
    pid = fork();

    if(pid == -1)
    {
        printf("error while creating child process \n");
        exit(-1);
    }
    else if(pid == 0) // for child process
    {
        printf("now, execlp will execute \n");
        execlp("./a.out", "p1", NULL);
    }

    // only the parent process will execute the below line
    printf("program done for PID = %d \n", pid);
}
```



```
180905402@prg22: ~/5thSemLabs/os/lab4_1
File Edit View Search Terminal Help
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 #include<unistd.h>
5 #include<sys/wait.h>
6 #include<sys/types.h>
7
8 int main()
9 {
10     pid_t pid;
11     int n, status;
12     pid = fork();
13
14     if(pid == -1)
15     {
16         printf("error while creating child process \n");
17         exit(-1);
18     }
19     else if(pid == 0) // for child process
20     {
21         printf("now, execlp will execute \n");
22         execlp("./a.out", "p1", NULL);
23     }
24
25     // only the parent process will execute the below line
26     printf("program done for PID = %d \n", pid);
27 }
```

```
180905402@prg22: ~/5thSemLabs/os/lab4_1$ gcc p2.c -o b.out
180905402@prg22: ~/5thSemLabs/os/lab4_1$ ./b.out
program done for PID = 22953
now, execlp will execute
main program has begun
180905402@prg22: ~/5thSemLabs/os/lab4_1$ child executing
10 11 12 13 14 15 16 17 18 19
now parent will execute
20 21 22 23 24 25 26 27 28 29
```

"p2.c" 27L, 487C written 26,45-48 All

Q3)

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

```
int main()
{
    pid_t pid;

    pid = fork();
    if(pid < 0)
    {
        printf("error in creating a child process \n");
        exit(-1);
    }
    else if(pid == 0) // for child process
    {
        printf("inside the child process \n");
        printf("the child PID is: %d \n", getpid());
        printf("the parent PID is: %d \n", getppid());
    }
    else
    {
        printf("inside the parent process \n");
        printf("the child PID is: %d \n", getpid());
        printf("the parent PID is: %d \n", getppid());
    }
}
```

```
}
180905402@prg22: ~/5thSemLabs/os/lab4_1
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 #include<ctype.h>
5 #include<unistd.h>
6 #include<sys/types.h>
7 #include<sys/wait.h>
8
9 int main()
10 {
11     pid_t pid;
12
13     pid = fork();
14     if(pid < 0)
15     {
16         printf("error in creating a child process \n");
17         exit(-1);
18     }
19     else if(pid == 0) // for child process
20     {
21         printf("inside the child process \n");
22         printf("the child PID is: %d \n", getpid());
23         printf("the parent PID is: %d \n", getppid());
24     }
25     else
26     {
27         printf("inside the parent process \n");
28         printf("the child PID is: %d \n", getpid());
29         printf("the parent PID is: %d \n", getppid());
30     }
31 }
```

180905402@prg22: ~/5thSemLabs/os/lab4\_1\$ gcc p3.c -o c.out

180905402@prg22: ~/5thSemLabs/os/lab4\_1\$ ./c.out

inside the parent process  
the child PID is: 23901  
the parent PID is: 22907  
inside the child process  
the child PID is: 23902  
the parent PID is: 23901

180905402@prg22: ~/5thSemLabs/os/lab4\_1\$

Q4)

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<ctype.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<unistd.h>
int main()
{
    pid_t pid;
    pid = fork();
    if(pid < 0)
    {
        printf("error in creating child process \n");
    }
    else if(pid == 0)
    {
        printf("now in child process pid = %d \n", pid);
        exit(-1);
    }
    else
    {
        sleep(10);
    }
}
```

The screenshot shows a Linux terminal window with a dark background. The terminal displays a list of system processes, including gvfsd-http, sh, bash, kworker, nautilus, vim, and firefox. The user has executed a command to create a child process, and the output shows the child process ID as 24825. The terminal also shows the user's prompt and the current directory path.

```

180905402@prg22: ~/5thSemLabs/os/lab4_1
File Edit View Search Terminal Help
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4 #include<ctype.h>
...
180905402@prg22: ~/5thSemLabs/os/lab4_1
File Edit View Search Terminal Help
180905402@prg22:~/5thSemLabs/os/lab4_1$ ./d.out &
[1] 24825
180905402@prg22:~/5thSemLabs/os/lab4_1$ now in child pro
cess pid = 0

```

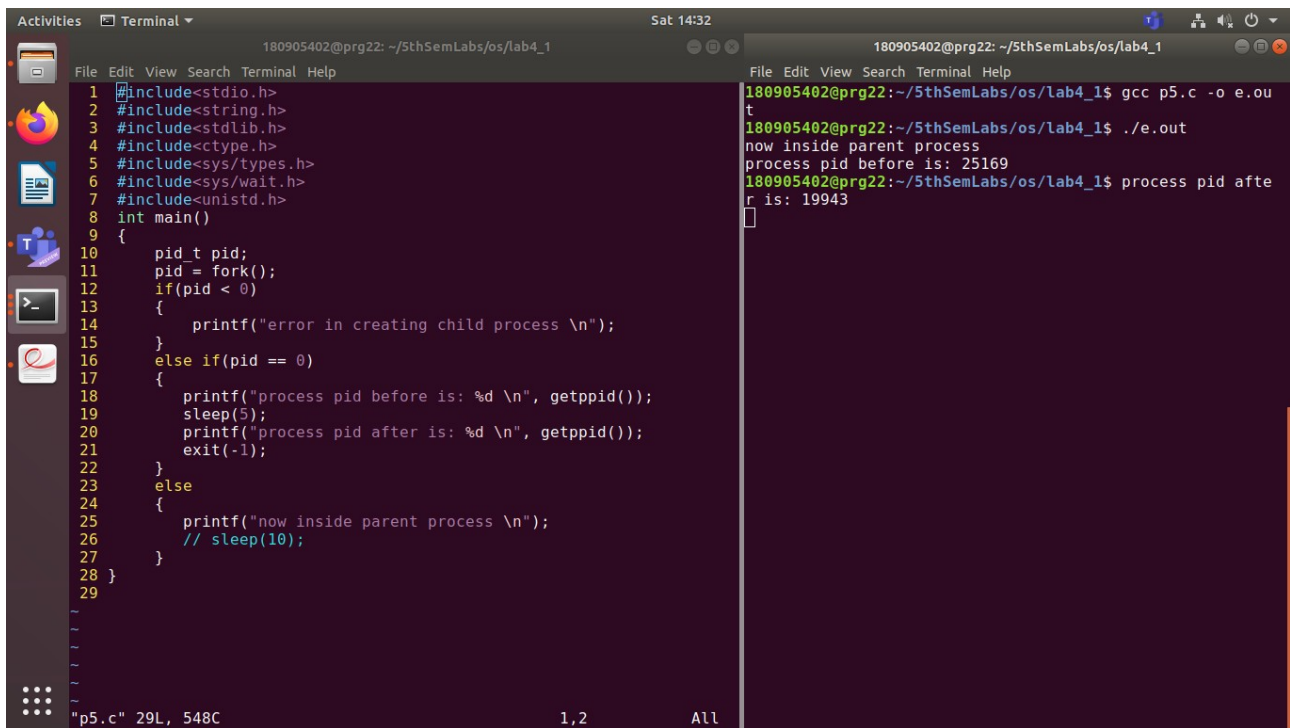
process id 24825

Q5)

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<ctype.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<unistd.h>
int main()
{
    pid_t pid;
    pid = fork();
    if(pid < 0)
    {
        printf("error in creating child process \n");
    }
    else if(pid == 0)
    {
        printf("process pid before is: %d \n", getpid());
        sleep(5);
        printf("process pid after is: %d \n", getpid());
        exit(-1);
    }
    else
    {
        printf("now inside parent process \n");
        // sleep(10);
    }
}

```



```
180905402@prg22: ~/5thSemLabs/os/lab4_1
File Edit View Search Terminal Help
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4 #include<ctype.h>
5 #include<sys/types.h>
6 #include<sys/wait.h>
7 #include<unistd.h>
8 int main()
9 {
10     pid_t pid;
11     pid = fork();
12     if(pid < 0)
13     {
14         printf("error in creating child process \n");
15     }
16     else if(pid == 0)
17     {
18         printf("process pid before is: %d \n", getpid());
19         sleep(5);
20         printf("process pid after is: %d \n", getpid());
21         exit(-1);
22     }
23     else
24     {
25         printf("now inside parent process \n");
26         // sleep(10);
27     }
28 }
29
"p5.c" 29L, 548C 1,2 All

180905402@prg22: ~/5thSemLabs/os/lab4_1$ gcc p5.c -o e.out
180905402@prg22: ~/5thSemLabs/os/lab4_1$ ./e.out
now inside parent process
process pid before is: 25169
180905402@prg22: ~/5thSemLabs/os/lab4_1$ process pid after is: 19943
```

Q6)

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<ctype.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<unistd.h>
int main()
{
    pid_t pid;
    pid = fork();
    int status;
    if(pid < 0)
    {
        printf("error in creating child process \n");
    }
    else if(pid == 0)
    {
        sleep(5);
        exit(-1);
    }
    else
    {
        printf("now inside parent process \n");
        wait(&status);
        printf("status code is %d \n", status);
    }
}
```

Activities Terminal Sat 14:41

180905402@prg22: ~/5thSemLabs/os/lab4\_1

```
File Edit View Search Terminal Help
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4 #include<ctype.h>
5 #include<sys/types.h>
6 #include<sys/wait.h>
7 #include<unistd.h>
8 int main()
9 {
10     pid_t pid;
11     pid = fork();
12     int status;
13     if(pid < 0)
14     {
15         printf("error in creating child process\n");
16     }
17     else if(pid == 0)
18     {
19         sleep(5);
20         exit(-1);
21     }
22     else
23     {
24         printf("now inside parent process \n");
25         wait(&status);
26         printf("status code is %d \n", status);
27     }
28 }
29
```

180905402@prg22: ~/5thSemLabs/os/lab4\_1

```
File Edit View Search Terminal Help
180905402@prg22:~/5thSemLabs/os/lab4_1$ gcc p6.c -o f.out
180905402@prg22:~/5thSemLabs/os/lab4_1$ ./f.out
now inside parent process
status code is 65280
180905402@prg22:~/5thSemLabs/os/lab4_1$
```

"p6.c" 29L, 497C 1,2 All