Assignment 4: Familiarization with Process Management in Linux environment.
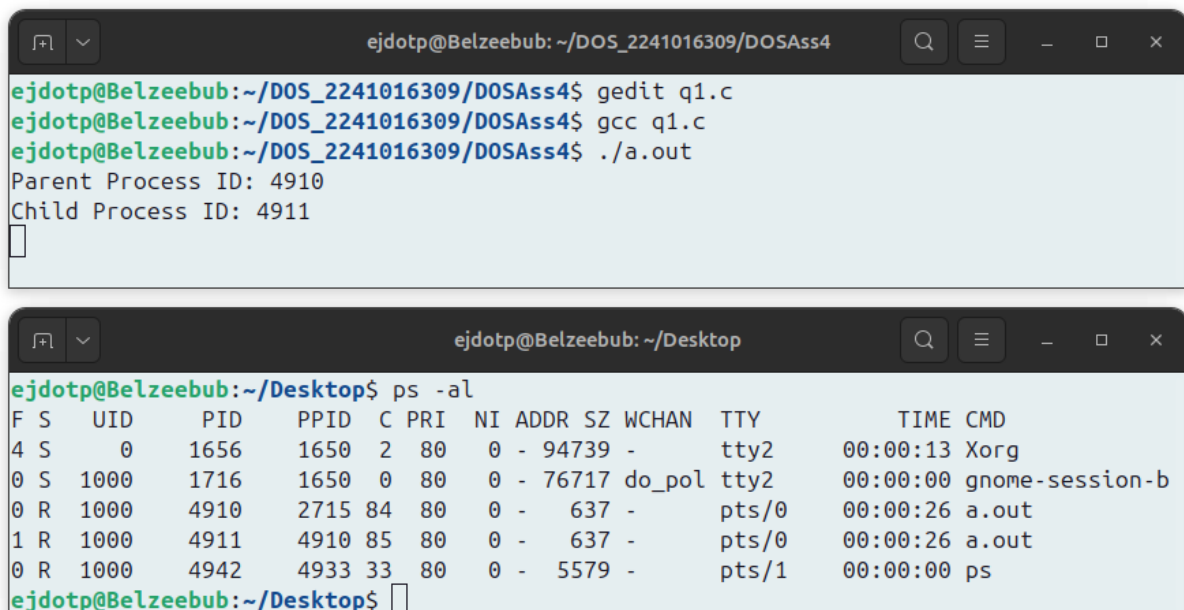
Objective of this Assignment:

- To trace the different states of a process during its execution.
- To learn the use of different system calls such as (fork(),vfork(),wait(),execl())
  for process handling in Unix/Linux environment.

Q1:

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

int main()
{
    pid_t c1 = fork();
    if(c1 == 0){
        fprintf(stderr, "Child Process ID: %d\n", getpid());
        while(1);
    }
    else{
        fprintf(stderr, "Parent Process ID: %d\n", getpid());
        while(1);
    }

    return 0;
}
```

(a):

```
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gedit q1.c
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gcc q1.c
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ ./a.out
Parent Process ID: 4910
Child Process ID: 4911
```

```
ejdotp@Belzeebub:~/Desktop$ ps -al
F S   UID    PID   PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0   1656   1650  2  80   0 - 94739 -      tty2     00:00:13 Xorg
0 S  1000   1716   1650  0  80   0 - 76717 do_pol tty2     00:00:00 gnome-session-b
0 R  1000   4910   2715 84  80   0 -   637 -      pts/0    00:00:26 a.out
1 R  1000   4911   4910 85  80   0 -   637 -      pts/0    00:00:26 a.out
0 R  1000   4942   4933 33  80   0 -  5579 -      pts/1    00:00:00 ps
ejdotp@Belzeebub:~/Desktop$
```

(b):

```
0 R  1000     4942     4933 33  80   0 -  5579 -        pts/1    00:00:00 ps
ejdotp@Belzeebub:~/Desktop$ kill -9 4911
ejdotp@Belzeebub:~/Desktop$ ps -al
F S   UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0     1656     1650  2  80   0 - 94859 -        tty2     00:00:14 Xorg
0 S  1000     1716     1650  0  80   0 - 76717 do_pol  tty2     00:00:00 gnome-session-b
0 R  1000     4910     2715 92  80   0 -   637 -        pts/0    00:01:58 a.out
1 Z  1000     4911     4910 88  80   0 -     0 -        pts/0    00:01:53 a.out
0 R  1000     5280     4933 99  80   0 -  5579 -        pts/1    00:00:00 ps
ejdotp@Belzeebub:~/Desktop$
```

(c):

```
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gcc q1.c
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ ./a.out
Parent Process ID: 5554
Child Process ID: 5555
Killed
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$
```

```
ejdotp@Belzeebub:~/Desktop$ ps -al
F S   UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0     1656     1650  3  80   0 - 95011 -        tty2     00:00:28 Xorg
0 S  1000     1716     1650  0  80   0 - 76717 do_pol  tty2     00:00:00 gnome-session-b
0 R  1000     5554     2715 96  80   0 -   637 -        pts/0    00:00:05 a.out
1 R  1000     5555     5554 94  80   0 -   637 -        pts/0    00:00:05 a.out
0 R  1000     5556     4933 99  80   0 -  5579 -        pts/1    00:00:00 ps
ejdotp@Belzeebub:~/Desktop$ kill -9 5554
ejdotp@Belzeebub:~/Desktop$ ps -al
F S   UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0     1656     1650  3  80   0 - 95011 -        tty2     00:00:28 Xorg
0 S  1000     1716     1650  0  80   0 - 76717 do_pol  tty2     00:00:00 gnome-session-b
1 R  1000     5555     1582 98  80   0 -   637 -        pts/0    00:00:22 a.out
0 R  1000     5557     4933  0  80   0 -  5579 -        pts/1    00:00:00 ps
ejdotp@Belzeebub:~/Desktop$
```

(d):

```
                                            q1.c
 Open    ∨    ⊞              ~/DOS_2241016309/DOSAss4          Save    ≡    _   □   ×

 1 #include <stdio.h>
 2 #include <unistd.h>
 3 #include <sys/wait.h>
 4
 5 int main()
 6 {
 7     pid_t c1 = fork();
 8     if(c1 == 0){
 9         fprintf(stderr, "Child Process ID: %d\n", getpid());
10         while(1);
11     }
12     else{
13         fprintf(stderr, "Parent Process ID: %d\n", getpid());
14         wait(NULL);
15         while(1);
16     }
17
18     return 0;
19 }

 Saving file "~/DOS_2241016309/DOSAss4/q1.c"...          C ∨   Tab Width: 4 ∨    Ln 14, Col 20    INS
```

```
 ⊞  ∨           ejdotp@Belzeebub: ~/DOS_2241016309/DOSAss4      🔍  ≡   _   □   ×

 ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gedit q1.c
 ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gcc q1.c
 ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ ./a.out
 Parent Process ID: 5666
 Child Process ID: 5667
```

```
 ⊞  ∨                  ejdotp@Belzeebub: ~/Desktop              🔍  ≡   _   □   ×

 ejdotp@Belzeebub:~/Desktop$ ps -all
 F S   UID    PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY         TIME CMD
 4 S     0   1656    1650  3  80   0 - 95071 -      tty2     00:00:37 Xorg
 0 S  1000   1716    1650  0  80   0 - 76717 do_pol tty2     00:00:00 gnome-session-b
 0 S  1000   5666    2715  0  80   0 -   637 do_wai pts/0    00:00:00 a.out
 1 R  1000   5667    5666 99  80   0 -   637 -      pts/0    00:00:09 a.out
 0 R  1000   5668    4933  0  80   0 -  5579 -      pts/1    00:00:00 ps
 ejdotp@Belzeebub:~/Desktop$
```

(e):

```
 ⊞  ∨                  ejdotp@Belzeebub: ~/Desktop              🔍  ≡   _   □   ×

 0 R  1000   5668    4933  0  80   0 -  5579 -      pts/1    00:00:00 ps
 ejdotp@Belzeebub:~/Desktop$ kill -9 5667
 ejdotp@Belzeebub:~/Desktop$ ps -all
 F S   UID    PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY         TIME CMD
 4 S     0   1656    1650  2  80   0 - 94951 -      tty2     00:00:37 Xorg
 0 S  1000   1716    1650  0  80   0 - 76717 do_pol tty2     00:00:00 gnome-session-b
 0 R  1000   5666    2715  2  80   0 -   637 -      pts/0    00:00:03 a.out
 0 R  1000   5696    4933  0  80   0 -  5579 -      pts/1    00:00:00 ps
 ejdotp@Belzeebub:~/Desktop$
```

Q2:

Q3:

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

int main()
{
    if (vfork() == 0) {
        fprintf(stderr, "\nChild Process ID: %d\nParent Process ID: %d\n", getpid(), getppid());
        execl("/bin/cp", "cp", "file1", "file2", NULL);
    }
    sleep(1);

    if (vfork() == 0) {
        fprintf(stderr, "\nChild Process ID: %d\nParent Process ID: %d\n", getpid(), getppid());
        execl("/bin/cat", "cat", "file2", NULL);
    }
    sleep(1);

    if (vfork() == 0) {
        fprintf(stderr, "\nChild Process ID: %d\nParent Process ID: %d\n", getpid(), getppid());
        execl("/usr/bin/sort", "sort", "-r", "file2", NULL);
    }
    sleep(1);

    while (wait(NULL) > 0);
    fprintf(stderr, "\nParent Process ID: %d\n", getpid());

    return 0;
}
```

```
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ cat > file1

2
4
3
1
5
^C
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ cat > file2

^C
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gedit q3.c

ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gcc q3.c

ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ ./a.out


Child Process ID: 3595
Parent Process ID: 3594

Child Process ID: 3596
Parent Process ID: 3594
2
4
3
1
5


Child Process ID: 3597
Parent Process ID: 3594
5
4
3
2
1


Parent Process ID: 3594
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$
```

Q4:

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

int main()
{
    int len;
    printf("Enter the Length: ");
    scanf("%d", &len);
    int fib[len];
    if(vfork() == 0){
        fib[0] = 0;
        fib[1] = 1;
        for(int i = 2; i < len; i++)
            fib[i] = fib[i-1] + fib[i-2];
        _exit(0);
    }
    printf("\nFibonacci Series: ");
    for(int i = 0; i < len; i++)
        printf("%d ", fib[i]);
    printf("\nPrimes in the series:-\n");
    for(int i = 3; i < len; i++){
        int isPrime = 1;
        for(int j = 2; j*j <= fib[i]; j++){
            if(fib[i]%j == 0)
                isPrime = 0;
        }
        if(isPrime)
            printf("%d at index %d\n", fib[i], i);
    }

    return 0;
}
```

```
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gedit q4.c
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ gcc q4.c
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$ ./a.out
Enter the Length: 10

Fibonacci Series: 0 1 1 2 3 5 8 13 21 34
Primes in the series:-
2 at index 3
3 at index 4
5 at index 5
13 at index 7
ejdotp@Belzeebub:~/DOS_2241016309/DOSAss4$
```