

## Experiment – 10

**Aim: Program to create orphan process and zombie process.**

An orphan process is a process whose parent has terminated before it finishes its execution.

A zombie process is a process that has completed execution but still has an entry in the process table.

### 1) Orphan Process

vi orphan.c

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main()
{
    pid_t p;
    p = fork();
    if (p == 0)
    {
        sleep(5);
        printf("I am child having PID: %d\n",
getpid());
        printf("My parent pid is: %d\n",
getppid()); }
    else
    {
        printf("I am parent having pid:
%d\n", getpid());
        printf("My child pid is: %d\n", p);
    }
}
```

gcc -o orphan orphan.c  
./orphan

```
localhost:~# vi orphan.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
int main() {
    pid_t p;
    p = fork();
    if (p == 0) {
        // Child process
        sleep(5);
        printf("I am child having PID: %d\n", getpid());
        printf("My parent PID is: %d\n", getppid());
    } else {
        // Parent process
        printf("I am parent having PID: %d\n", getpid());
        printf("My child PID is: %d\n", p);
    }
    return 0;
}
localhost:~# gcc -o orphan orphan.c
localhost:~# ./orphan
I am parent having PID: 81
My child PID is: 82
localhost:~# I am child having PID: 82
My parent PID is: 1
```

## 2) Zombie Process

vi zombie.c

```
#include <stdio.h>
#include <unistd.h>
int main()
{
    pid_t p;
    p = fork();
    if (p == 0)
    {
        printf("Child having id : %d\n",
getpid());
    }
    else
    {
        printf("Parent having id : %d\n",
getpid());
        sleep(15); //run the ps command
during this time.
    }
}
```

gcc -o zombie zombie.c

./zombie &

ps -elf | grep defunct

```
localhost:~# vi zombie.c
#include <stdio.h>
#include <unistd.h>
int main() {
    pid_t p;
    p = fork();
    if (p == 0) {
        // Child process
        printf("Child having ID: %d\n", getpid());
    } else {
        // Parent process
        printf("Parent having ID: %d\n", getpid());
        sleep(15); // Run `ps` command during this time
    }
    return 0;
}
localhost:~# gcc -o zombie zombie.c
localhost:~# ./zombie &
localhost:~# Parent having id : 94
Child having id : 95

localhost:~# ps -elf | grep defunct
  97 root      0:00 grep defunct
[1]+  Done                  ./zombie
localhost:~#
```