Experiment 10

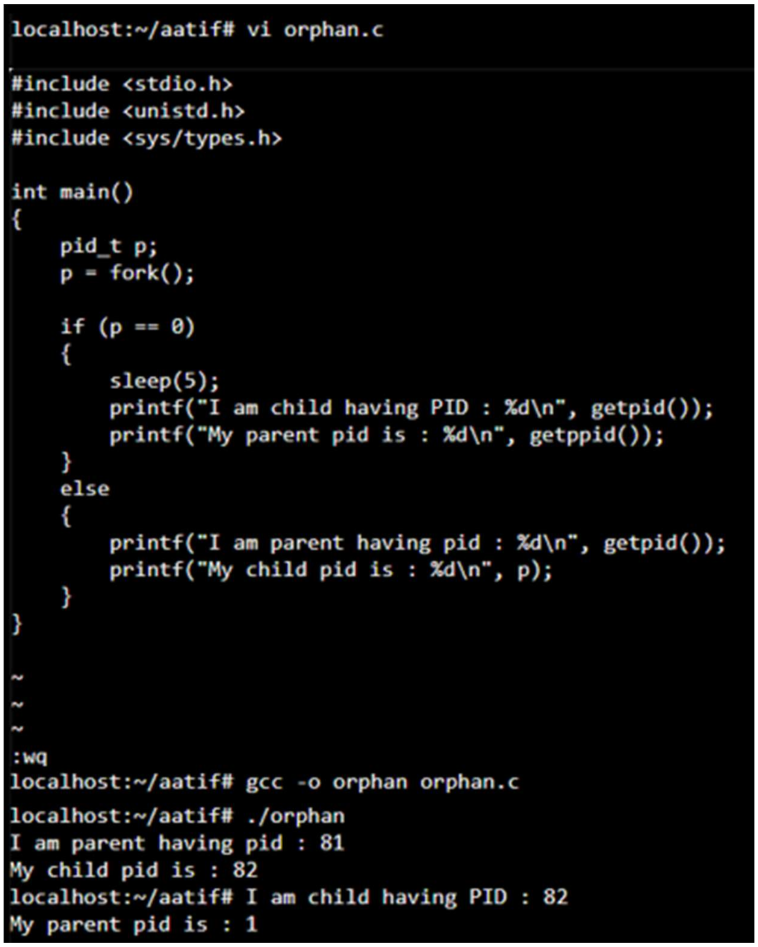
Aim: Program to create orphan process and zombie process.

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

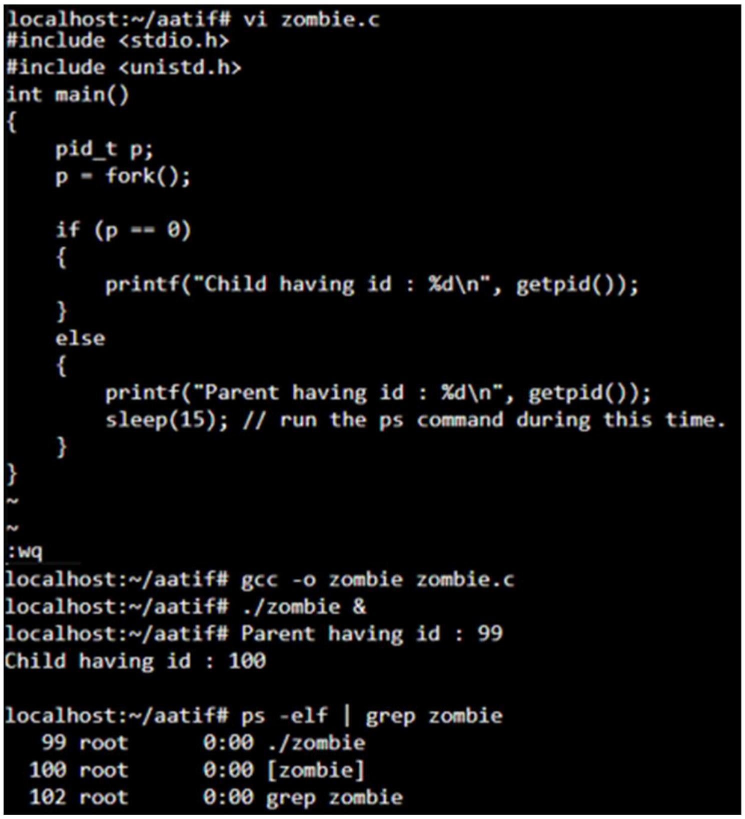
```
vi orphan.c

#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);
   }
}

:wq

gcc -o orphan orphan.c

./orphan
```



```
localhost:~/aatif# 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);
    }
}

~
~
~
:wq
localhost:~/aatif# gcc -o orphan orphan.c
localhost:~/aatif# ./orphan
I am parent having pid : 81
My child pid is : 82
localhost:~/aatif# 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.}
}

:wq

gcc -o zombie zombie.c

./zombie &

ps -elf | grep defunct
```



```
localhost:~/aatif# 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.
    }
}
~
~
:wq
localhost:~/aatif# gcc -o zombie zombie.c
localhost:~/aatif# ./zombie &
localhost:~/aatif# Parent having id : 99
Child having id : 100

localhost:~/aatif# ps -elf | grep zombie
   99 root      0:00 ./zombie
  100 root      0:00 [zombie]
  102 root      0:00 grep zombie
```