**Program:**

#include<sys/types.h>

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

int main()

{

pid\_t pid;

pid=fork();

if(pid<0)

{

fprintf(stderr,"Fork Failed");

return 1;

}

else if(pid==0){

execlp("/bin/s","ls",NULL);

printf("Child's pid number is%d",getpid());

sleep(10); //ORPHAN STATE

}

else{

printf("Child Complete");

printf("Parent's pid number is%d",getppid());

sleep(10);//ZOMBIE STATE

}

return 0;

}

**OUTPUT:**

**//Orphan State**

**mml@mml-Vostro-3470:~/DSS71$ gcc -o first.out first.c**

**mml@mml-Vostro-3470:~/DSS71$ ./first.out**

**Child CompleteParent's pid number is3377mml@mml-Vostro-3470:~/DSS71$ Child's pid number is5458**

**//Zombie State**

**mml@mml-Vostro-3470:~/DSS71$ gcc -o first.out first.c**

**mml@mml-Vostro-3470:~/DSS71$ ./first.out**

**Child's pid number is5527^Z**

**[11]+ Stopped ./first.out**

**mml@mml-Vostro-3470:~/DSS71$ ps -l**

**F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD**

**0 S 1000 3377 3369 0 80 0 - 2686 do\_wai pts/0 00:00:00 bash**

**0 T 1000 5132 3377 0 80 0 - 624 do\_sig pts/0 00:00:00 first.out**

**1 Z 1000 5133 5132 0 80 0 - 0 - pts/0 00:00:00 first.out <defunct>**

**0 T 1000 5218 3377 0 80 0 - 624 do\_sig pts/0 00:00:00 first.out**

**1 Z 1000 5219 5218 0 80 0 - 0 - pts/0 00:00:00 first.out <defunct>**

**0 T 1000 5526 3377 0 80 0 - 624 do\_sig pts/0 00:00:00 first.out**

**1 Z 1000 5527 5526 0 80 0 - 0 - pts/0 00:00:00 first.out <defunct>**

**4 R 1000 5534 3377 0 80 0 - 2854 - pts/0 00:00:00 ps**