Program to create an orphan process

Leave a Comment / Programs, System calls / By Baljit Singh Saini

An orphan process is a process whose parent has finished. Suppose P1 and P2 are two process such that P1 is the parent process and P2 is the child process of P1. Now, if P1 finishes before P2 finishes, then P2 becomes an orphan process. The following programs we will see how to create an orphan process.

Program1: To create a normal child (duplicate) process (no orphan process in this case)

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

Output:

```
baljit@baljit:~/cse325$ ./a.out
I am parent having PID 130
I am child having PID 131
My child PID is 131
My parent PID is 130
```

Process duplication

```
#include<stdio.h
#include<unistd.h>
#include<sys/types.h>
int main()
{
pid_t p;
p=fork();
```


```
 if(p==0)
{
    sleep(5); //child goes to sleep and in the mean time parent terminates
    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);
}
}
```

Output:

```
baljit@baljit:~/cse325$ gcc orphan.c
baljit@baljit:~/cse325$ ./a.out
I am parent having PID 138
My child PID is 139
baljit@baljit:~/cse325$ I am child having PID 139
My parent PID is 1
```

Orphan Process

How it Works?

In this code, we add sleep(5) in the child section. This line of code makes the child process go to sleep for 5 seconds and the parent starts executing. Since, parent process has just two lines to print, which it does well within 5 seconds and it terminates. After 5 seconds when the child process wakes up, its parent has already terminated and hence the child becomes an orphan process. Hence, it prints the PID of its parent as 1 (1 means the init process has been made its parent now) and not 138.

Note: The process will not return to the command prompt. Hence, use Ctrl+C to come to the command prompt. What can be the possible reason?

Q1. What is an orphan process?

Q2. What is the importance of using sleep() function in the above code?

Q3. Why the program didn't return the command prompt even after termination?

Relevant Programs

- Program to duplicate a process
- Understanding the use of fork() system call
- Program to create a zombie process
- Understanding the use of system() function

← Previous Post

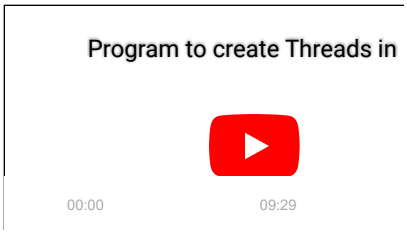
Next Post →

Leave a Comment

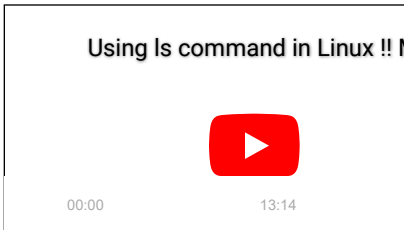
You must be logged in to post a comment.

Search ...

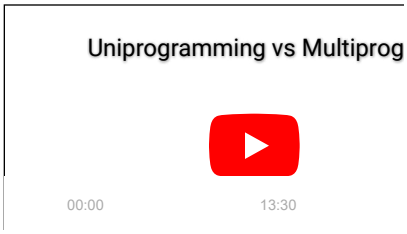
OS Lab



Linux Essentials Series



OS Theory



Categories

UGC-NET MCQ's for OS

Uncategorized