# Zombie and Orphan Processes in C

Difficulty Level : Medium • Last Updated : 29 May, 2017

Prerequisite: fork() in C

**Zombie Process:**

A process which has finished the execution but still has entry in the process table to report to its parent process is known as a zombie process. A child process always first becomes a zombie before being removed from the process table. The parent process reads the exit status of the child process which reaps off the child process entry from the process table.
In the following code, the child finishes its execution using exit() system call while the parent sleeps for 50 seconds, hence doesn't call wait() and the child process's entry still exists in the process table.

```
// A C program to demonstrate Zombie Process.
// Child becomes Zombie as parent is sleeping
// when child process exits.
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    // Fork returns process id
    // in parent process
    pid_t child_pid = fork();

    // Parent process
    if (child_pid > 0)
        sleep(50);

    // Child process
    else
        exit(0);

    return 0;
}
```

Note that the above code may not work with online compiler as fork() is disabled.

**Orphan Process:**

A process whose parent process no more exists i.e. either finished or terminated without waiting for its child process to terminate is called an orphan process.

In the following code, parent finishes execution and exits while the child process is still executing and is called an orphan process now.

However, the orphan process is soon adopted by init process, once its parent process dies.

```
// A C program to demonstrate Orphan Process.
// Parent process finishes execution while the
// child process is running. The child process
// becomes orphan.
#include<stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    // Create a child process
    int pid = fork();

    if (pid > 0)
        printf("in parent process");

    // Note that pid is 0 in child process
    // and negative if fork() fails
    else if (pid == 0)
    {
        sleep(30);
        printf("in child process");
    }

    return 0;
}
```

Note that the above code may not work with online compilers as fork() is disabled.

**Related :**
Any idea What are Zombies in Operating System?
Zombie Processes and their Prevention

This article is contributed by **Pranjal Mathur**.If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Want to learn from the best curated videos and practice problems, check out the C++ Foundation Course for Basic to Advanced C++ and C++ STL Course for foundation plus STL. To complete your preparation from learning a language to DS Algo and many more, please refer **Complete Interview Preparation Course**.

Like 17

Previous
Zombie Processes and their Prevention

Next
Mutex vs Semaphore



## RECOMMENDED ARTICLES

Page : 1 2 3

01 How to execute zombie and orphan process in a single program?
04, Sep 18

02 Zombie Processes and their Prevention
03, Apr 17

03 Chain processes vs Fan of processes using fork() function in C
05, Jul 21

04 Double forking to prevent Zombie process
30, May 17

05 Chat application between two processes using signals and shared memory
06, May 19

06 fork() and memory shared b/w processes created using it
23, Apr 17

07 fork() to execute processes from bottom to up using wait()
11, Jan 18

08 What are the differences between bitwise and logical AND operators in C/C++?
08, Mar 15

**Article Contributed By :**
GeeksforGeeks

**Vote for difficulty**
Current difficulty : Medium
Easy | Normal | Medium | Hard | Expert

**Article Tags :** CPP-Library, system-programming, C Language, C++
**Practice Tags :** CPP

Improve Article | Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.
Load Comments

## MOST VISITED IN C++

Vector in C++ STL
The C++ Standard Template Library (STL)
Object Oriented Programming in C++
Map in C++ Standard Template Library (STL)
C++ Classes and Objects