**OSL Assignment 2A**

**Aditya Kangune**

**K 11**

**33323**

**Problem Statement:**

Assignment No. 2:

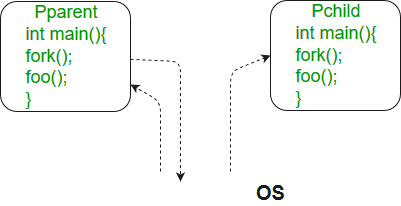
Process control system calls: The demonstration of FORK, EXECVE and WAIT system calls along with zombie and orphan states.

Implement the C program in which main program accepts the integers to be sorted. Main program uses the FORK system call to create a new process called a child process. Parent process sorts the integers using sorting algorithm and waits for child process using WAIT system call to sort the integers using any sorting algorithm. Also demonstrate zombie and orphan states.

1. **Fork system call in C:**

Fork system call is used for creating a new process, which is called **child process**, which runs concurrently with the process that makes the fork() call (parent process). After a new child process is created, both processes will execute the next instruction following the fork() system call. A child process uses the same pc(program counter), same CPU registers, same open files which use in the parent process.

It takes no parameters and returns an integer value. Below are different values returned by fork().



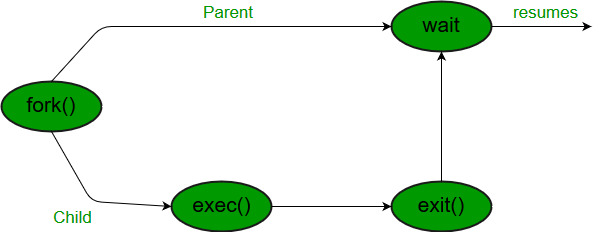
1. **Wait system call in C:**

A call to wait() blocks the calling process until one of its child processes exits or a signal is received. After child process terminates, parent ***continues*** its execution after wait system call instruction.   
Child process may terminate due to any of these:

It calls exit();

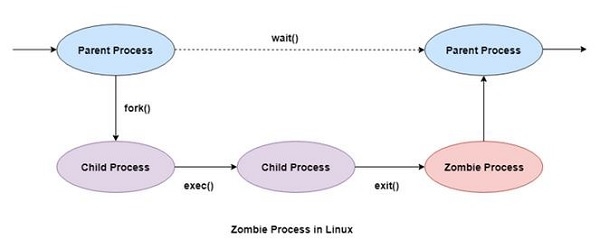
It returns (an int) from main

It receives a signal (from the OS or another process) whose default action is to terminate.



1. **Zombie state in C:**

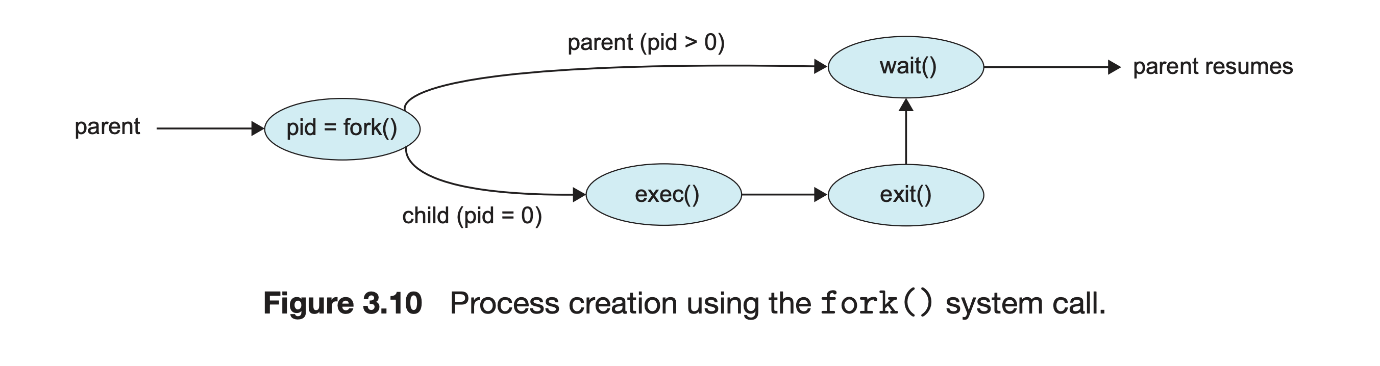
A process which has finished the execution but still has entry in the process table to report to its parent process is known as a zombie process. A child process always first becomes a zombie before being removed from the process table. The parent process reads the exit status of the child process which reaps off the child process entry from the process table.



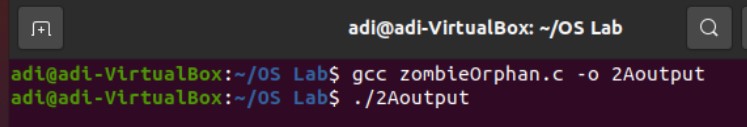
1. **Orphan state in C:**

Orphan processes are those processes that are still running even though their parent process has terminated or finished. A process can be orphaned intentionally or unintentionally.

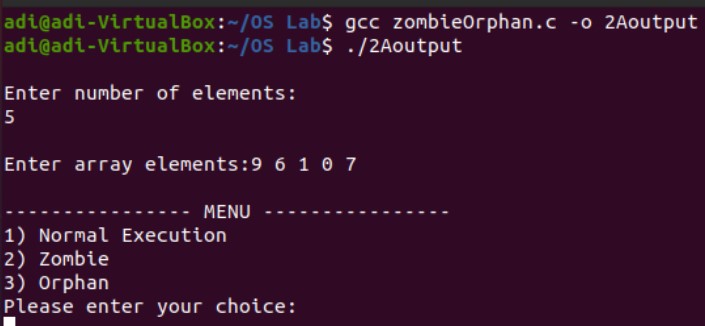
An intentionally orphaned process runs in the background without any manual support. This is usually done to start an indefinitely running service or to complete a long-running job without user attention.



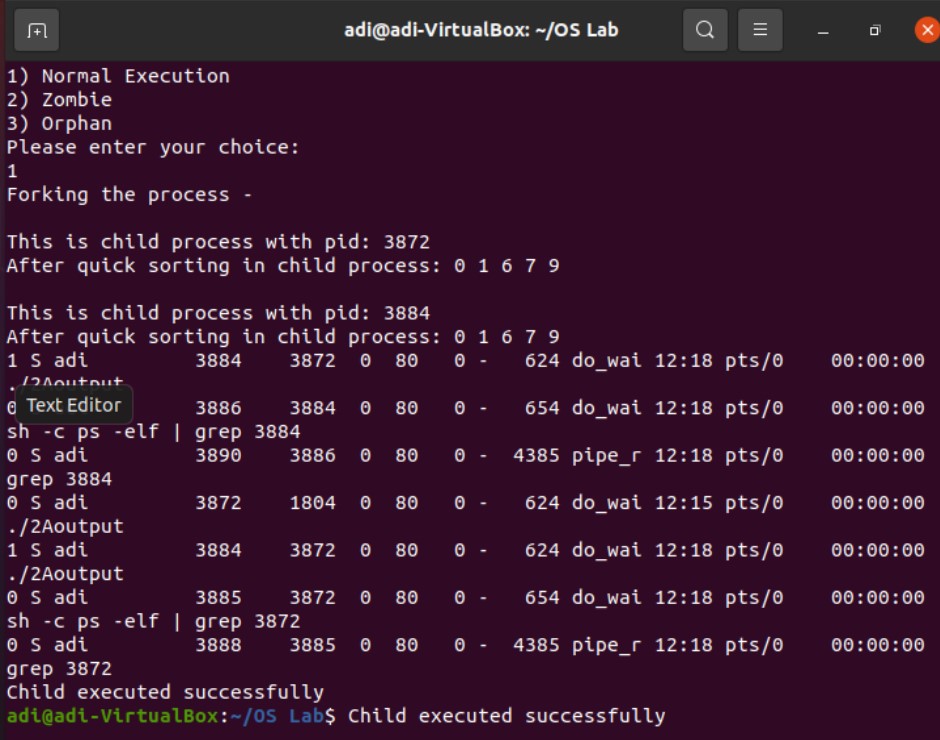
**Output screenshots:**

Compile and run: 

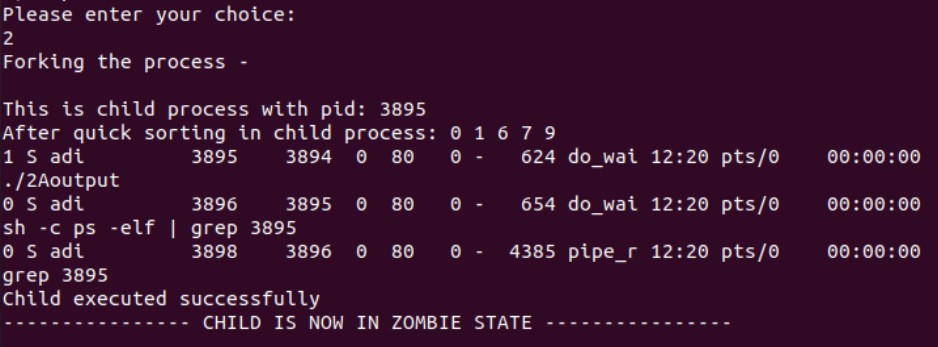
Entering array as input:

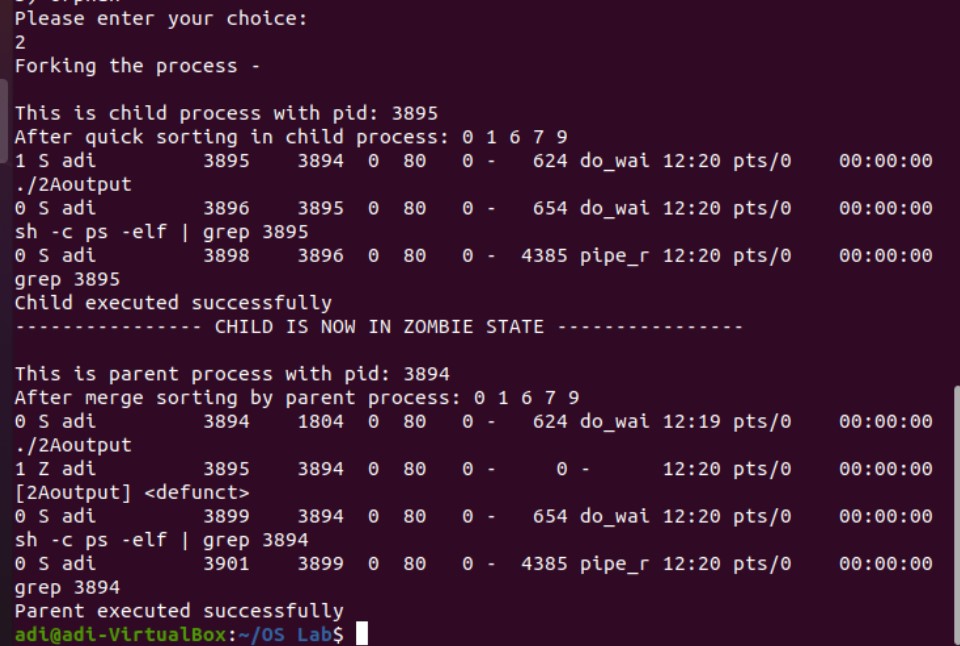


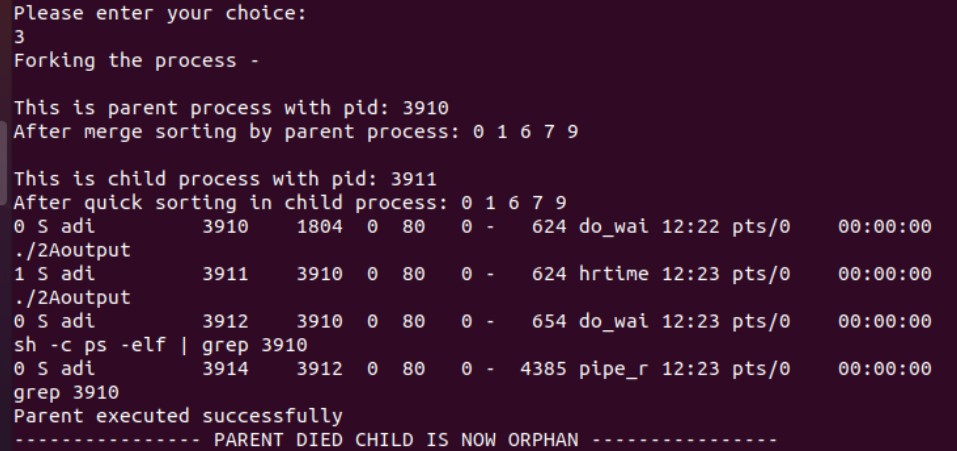
Normal execution:



Zombie State:



*After 10 seconds:* 

Orphan State: 

*After 10 seconds:* 