

Name :- Kumar D

Roll :- 33143

Page No.	
Date	

## Assignment 2A

### Problem Statement

Implement C program where main program uses a FORK system call to create a new child process. The parent process sorts an array of integers in ascending order while the child process in descending order. Parent waits for the child using WAIT system call to sort integers. Also demonstrate Zombie & Orphan process

### Theory

#### Linux System call

A system call is a procedure that provides the interface between a process & the Operating System. It is the method by which a computer program requests a service from the kernel of OS. These system calls deal with processes such as process creation, process termination etc. There are many different system calls such as wait(), execv(), fork(), exit(), sleep(), kill() etc.

#### fork()

Processes use the fork() system call to create processes that are a copy of themselves. The process which makes the fork() call is called the parent process. After the fork() call, a new process called the child process gets created that executes the next instruction along with the parent process. A child process uses the same program counter, CPU registers & open files as parent process. fork() takes no parameters and returns an integer value  
-ve value  $\Rightarrow$  error in creating child process  
zero  $\Rightarrow$  Returned to the newly created child process  
+ve value  $\Rightarrow$  Returned to the parent process i.e. the caller



## wait()

A `wait()` system call makes the calling parent process suspend till the child process gets terminated. After the child process terminates, the parent process continues its execution after `wait` system call instruction.

If no child of the parent process is terminated, the parent stays in wait state. If any one of the child process is terminated, then `wait()` returns process ID of the terminated child.

## Conclusion

~~We demonstrated Zombie & Orphan~~

### Zombie Process

A process which has finished its execution but still has its entry in the process table is known as a Zombie process. Until the child process gets removed from the process table, it stays in Zombie state.

The parent process reads the exit status of child which removes the entry from the process table.

### Orphan Process

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

## Conclusion

We demonstrated Zombie & Orphan process states using system calls.