Zombie and Orphan Process

1. What is a Zombie Process ✓
2. what is an Orphan Process ✓
3. How to kill a Zombie Process ✓
4. Side effects of Zombie Process ✓
5. Examples using code in C ✓

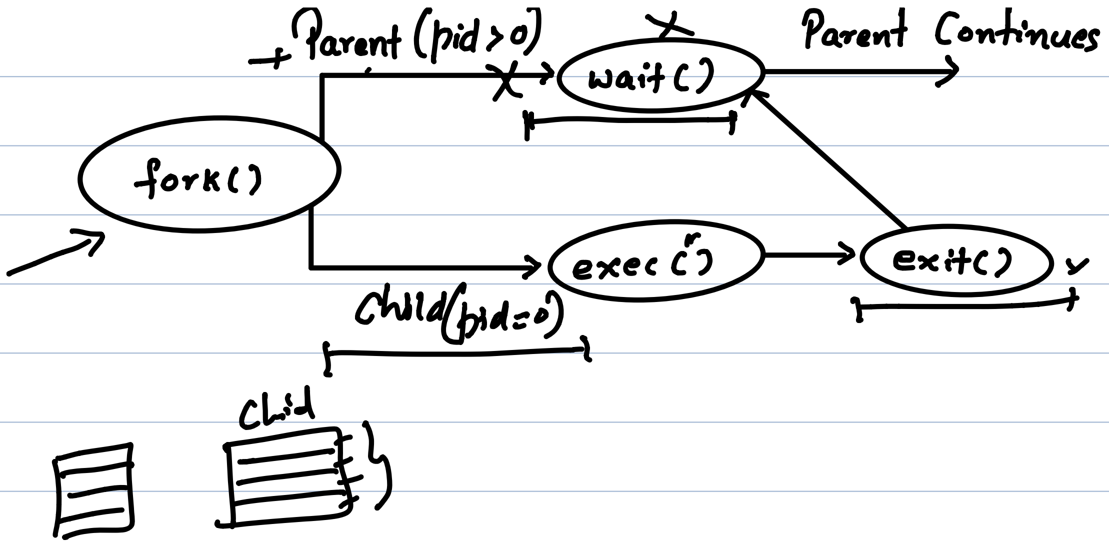
fork() → Creates a new process

exec() → Replaces Current process image with a new process image

wait() → used by Parent Process to wait for child's Completion

exit() → used to terminate a process

Process Creation typical flow :-



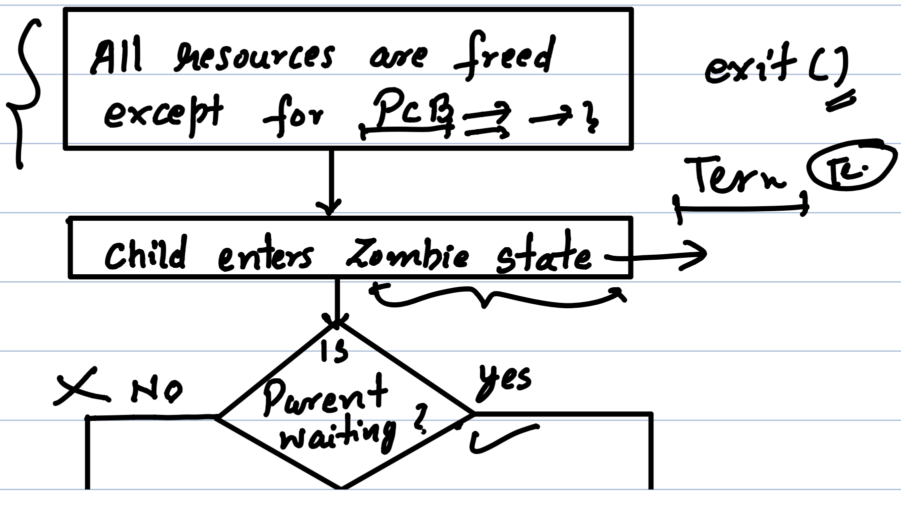
Let's consider three cases for the parent process:

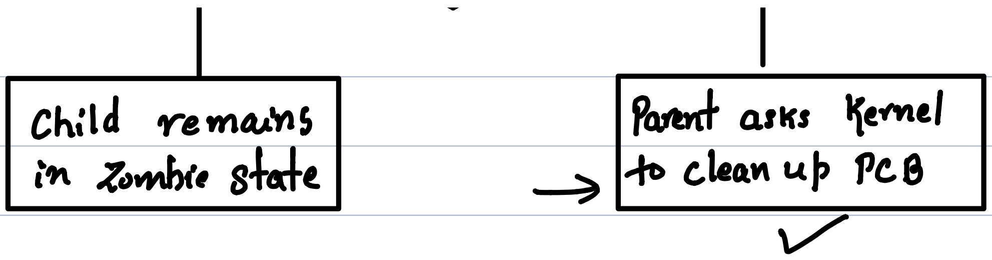
1. The Parent waits for the child

2. Parent gets killed before child terminates.

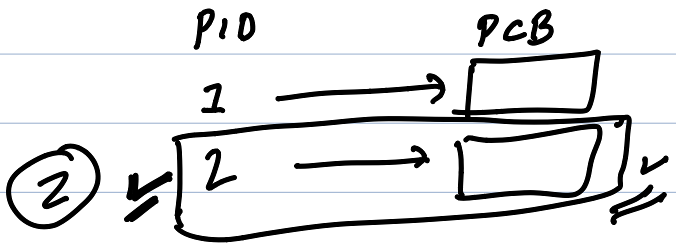
3. Parent doesn't wait for the child process

When a child terminates, the following event occurs:

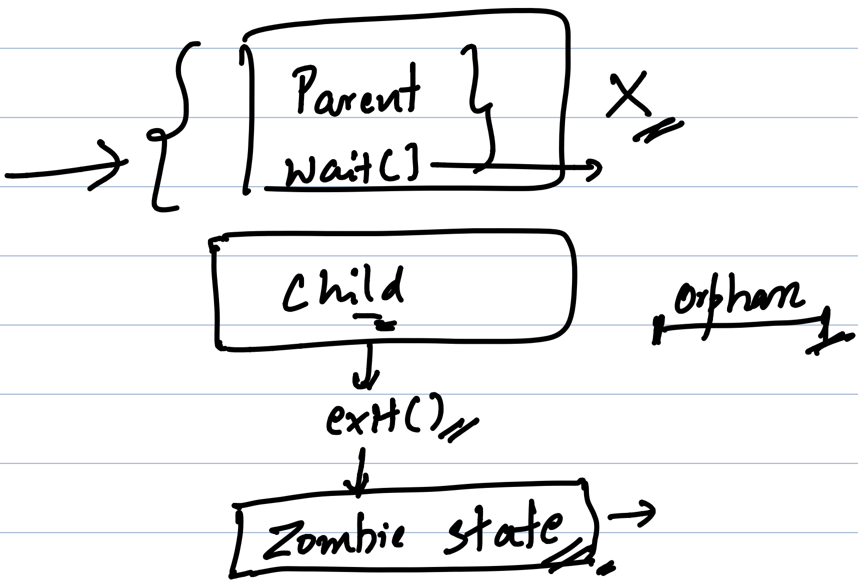

All resources are freed except for PCB
exit()
Term.
Child enters Zombie state
Is Parent waiting?
No
yes


Child remains in Zombie state
Parent asks Kernel to clean up PCB

Zombie Process : A process which has completed its execution but still has an entry in process Table.



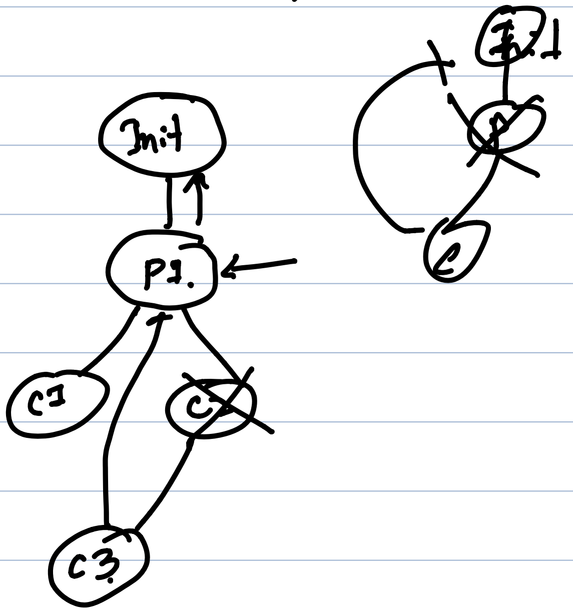
Reaping the child by Parent



Orphan Process : A process that doesn't have a parent.

New Parent → Init Process PID: 1

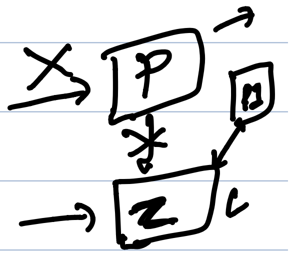
wait()



Reaping

Reparenting : Finding another parent for the process.

* All process enters the Zombie state after exit().
* Orphan process doesn't have a parent and it gets reparented.



How to remove a Zombie Process ?

1. Kill the Parent : Zombie process becomes orphan and gets reparented.

2. Send SIGCHLD Signal to parent and make the parent

handle it. ✓ signal_handler

Adverse effects of Zombie Process : ✓ PCB PID

If there are limited zombie process, no side effects.

1. Limited number of PID are available. - 32768 ✓ PCB → PID

2. Increase memory usage. ✓

PCB → Less