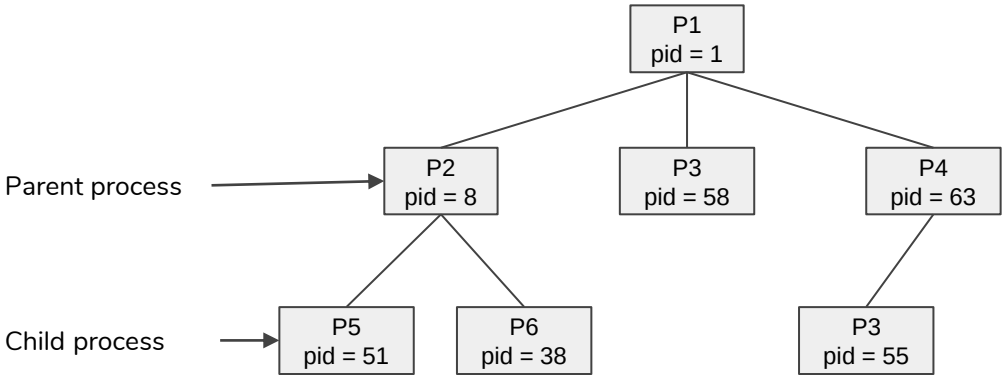
Operations on Process

Process Creation

A process is identified by a unique PID (Process Identifier) in the OS.

A process may create new processes.



- Child process obtain resources from OS or are restricted to Parent's resources
- Parent process may pass initializing data to child process

Process Creation

When a process creates new process -

The parent continues to execute concurrently with its children
Or,
The parent waits until some or all of its children have terminated

Two address-space possibilities for the new process -

The child process is a duplicate of the parent process
Or
The child process has a new program loaded into it.

Process creation in UNIX

System Call: offers the services of the operating system to the user programs.

***fork()*:** create a new process, which becomes the child process of the caller

***exec()*:** runs an executable file , replacing the previous executable

***wait()*:** suspends execution of the current process until one of its children terminates.

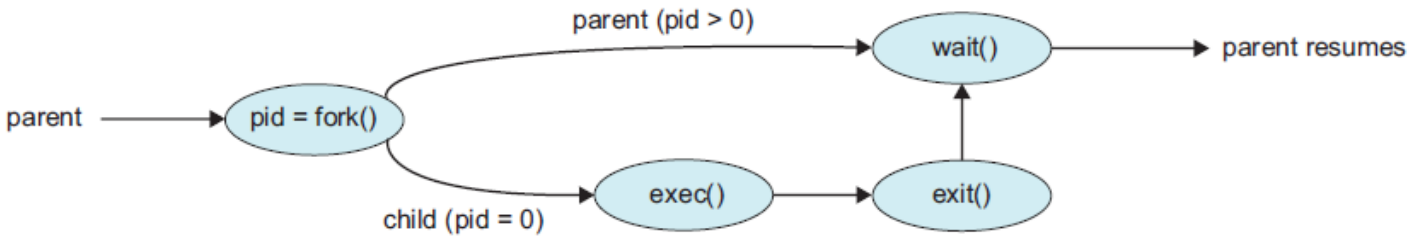


Fig: Process creation using fork() system call

Process Termination

A process is terminated when -

It executes its last statement
Or
Termination cause by another process

When a process is terminated, the resources are deallocated.

A parent may terminate its child if -

1. Child has exceeded the usage of resources
2. Task assigned to child is no longer needed
3. Parent is exiting (cascading termination)

Zombie Process in UNIX