

Modern general purpose OS permits a user to create and destroy processes. A process may create several new processes during its time of execution. The creating process is called Parent Process, while the new processes are called Child Processes.

There are different possibilities concerning creating new processes:-

Execution: The parent process continues to execute concurrently with its children processes or it waits until all of its children processes have terminated (Sequential)

Sharing: Either the parent and children processes share all resources (like memory or files) or the children processes share only a subset of their parent's resources or the parent and children process share no resources in common.

A parent process can terminate the execution of one of its children for one of these reasons:-

- 1) The child process has exceeded its usage of the resources it has been allocated. For this a mechanism must be available to allow the parent process to inspect the state of its children process.
- 2) Task assigned to child process is no longer required.

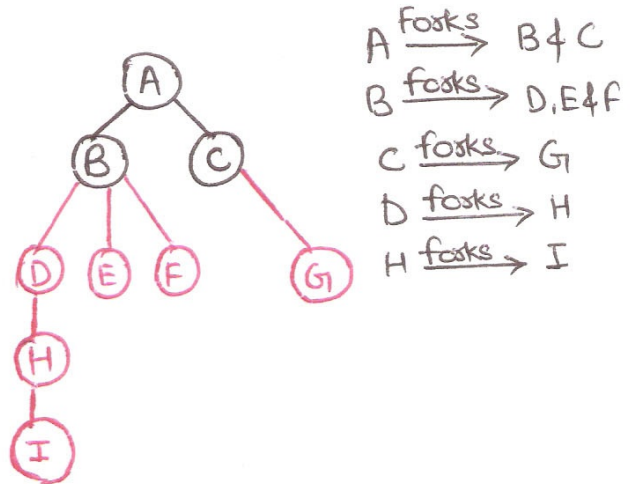
Let us discuss this with an example:-

In Unix this is done by the 'fork' System call, which creates a 'Child' process and the 'exit' System call, which terminates Current process.

The root of tree is a special process created by the OS during startup.

A process can choose to wait for children to terminate

Example C issued a wait () System call it would block until G finished.



Subscribe to our  
**You Tube Channel**

