

## **Operation on Processes**

The process in the system can execute concurrently, and they must be created and deleted dynamically.

### ▪ **Process Creation**

A process may create several new processes during the course of execution. The creating process is called a parent process, whereas the new processes are called the children.

When a process is created it obtains various resources and initialization values that may be passed along from the parent process to the child process.

### ▪ **Process Termination**

A process terminates when it finishes executing its final statement and asks the operating system to delete it. At that point the process may return data to its parent process and the OS deallocate all the physical and logical resources that are previously allocated to that process.

## **Cooperating Processes**

The concurrent process executing in the operating system may be either independent processes that does not share any data or cooperating that affects each others.

We may provide an environment that allows process cooperation for several reasons:

- Information sharing
- Computation speedup
- Modularity
- Convenience

## **Inter process Communication**

The cooperating processes can communicate in a shared memory environment. The scheme requires that these processes share a common buffer pool. Another way to achieve the same effect for the operating system is provided via an interprocess communication (IPC).

IPC provides a mechanism to allow processes to communicate and synchronize their actions without sharing the same address space. This technique is useful for distributed systems. IPC is provided by a message passing system.

### **Inter Process Communication (IPC)**

IPC is some times necessary but it presents two main problems:

1. Address violation problem :IPC means sharing some data (access common locations in memory). The shared data will be outside the address space of at least one process which, in turn, creates address violation problem. This problem may be solved by using "System Calls" for shared variables.
2. Write Access Problem : If the shared variable is of type Read/Write then another problem has to be solved in order to keep data integrity. This topic will be discussed later when studying "Asynchronous Concurrent Execution".