Unit-II
Lecture: 4
(Process Management)
(Part-I)

The Process Model:

What is a Process?
A process is a program in execution.
In simple terms, we write our computer programs in a text file and when we execute this program, it becomes a process which performs all the tasks mentioned in the program.

Program vs Process:
A process is a program in execution e.g. when we write a program in C or C++ and compile it, the compiler creates binary code. The original code and binary code are both programs When we actually run the binary code, it becomes a process.
A process is an active entity, instead of a program, which is a passive entity.
A single program can create many processes when run multiple times e.g. when we open a.exe or binary file multiple times, multiple instances begin (multiple processes are created).

All the runnable software on the computer, sometimes including the operating system is organized into a number of sequential processes (or processes for short).

- A process is just an instance of an executing program, including the current values of the program counter, registers and variables.
  - *A program counter is a register in a computer processor that contains the address (location) of the instruction being executed at the current time.*
- Conceptually, each process has its own virtual CPU. **In reality, the real CPU switches back and forth from process to process.**

For understanding, it's like a collection of processes running in parallel. This rapid switching back and forth is called multiprogramming.
Fig (a) below shows a computer a computer multiprogramming four programs in memory.
 In fig (b), we see four processes, each with its own flow of control (i.e. its own logical program counter) and each one running independently of the other ones. In reality, there is only one physical program counter, so when each

process runs, its logical program counter is loaded into the real program counter. When it is finished (for the time being) the physical program counter is saved in the process' stored logical program counter in memory.

In Fig (c), we can see that, viewed over a long enough time interval, all the processes have made progress, but at any given instant only one process is actually running.
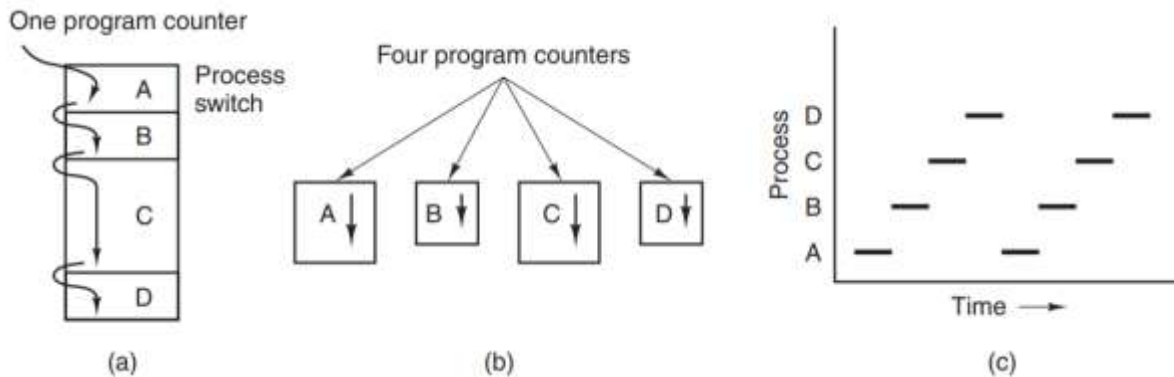

Fig: a) Multiprogramming four programs b) Conceptual model of four independent, sequential processes c) Only one program is active at once

So,
- Process is an activity of some kind. It has a program, input, output and a state.
- A single processor may be shared among several processes with some scheduling algorithm being accustomed to determine when to stop work on one process and service a different one.
- In contrast, a program is something that may be stored on disk, not doing anything.
- If a program is running twice, it counts as two processes.