**CO2**

**PROCESS MANAGEMENT**

**PROCESS CONCEPT**

Program in execution is called process. A process will need certain resources—such as CPU time, memory, files, and I/O devices — to accomplish its task. These resources are allocated to the process either when it is created or while it is executing. A system has a collection of processes — user processes as well as system processes. operating-system processes execute system code, and user processes execute user code. All these processes may +execute concurrently. The process is not as same as program code but a lot more than it. A program is a passive entity while a process is an active entity.

A program becomes a process when an executable file is loaded into memory. Two common techniques for loading executable files are double-clicking an icon representing the executable file and entering the name of the executable file on the command line.

**1)Representation of Process in memory:**

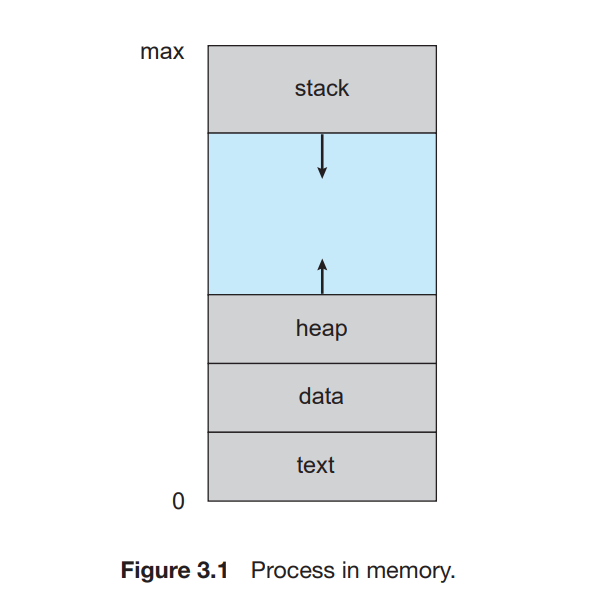
Process in memory is divided into the following four sections:

1)Text

2)Data

3)Heap

4)Stack

****

**1)Text:** The Text section is made up of the compiled program code. This includes the current activity represented by the value of Program Counter and the contents of the processor's registers.

**2)Data:** This section contains the global and static variables.

**3)Heap:** This is dynamically allocated memory to a process during its run time.

**4)Stack:** The process Stack contains the temporary data such as method/function parameters, return adder ss and local variables.

**2)Process States:**

During the execution of a process, a process keeps on changing its states. The state of a process refers to by the current activity of that process. A process can be in one of the following states during its lifetime:

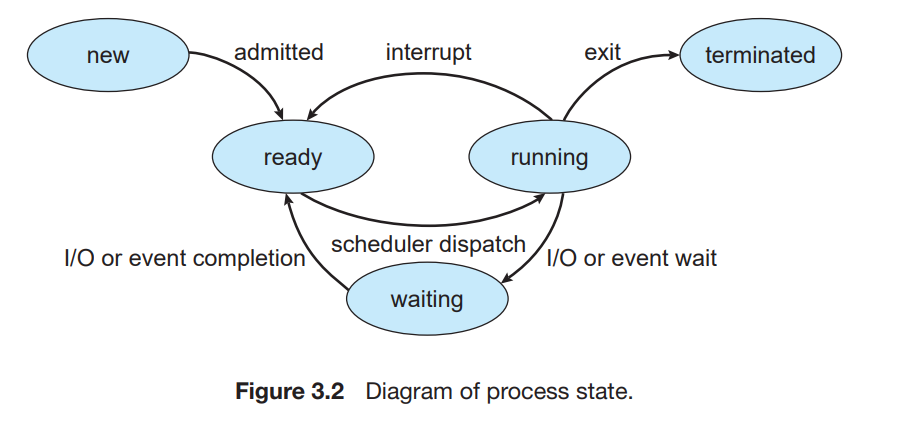
**1)New:** The process is being created.

**2)Ready:** The process is waiting to be assigned to a processor.

**3)Running:** Instructions are being executed.

**4)Waiting:** The process is waiting for some event to occur (such as an I/O completion or reception of a signal).

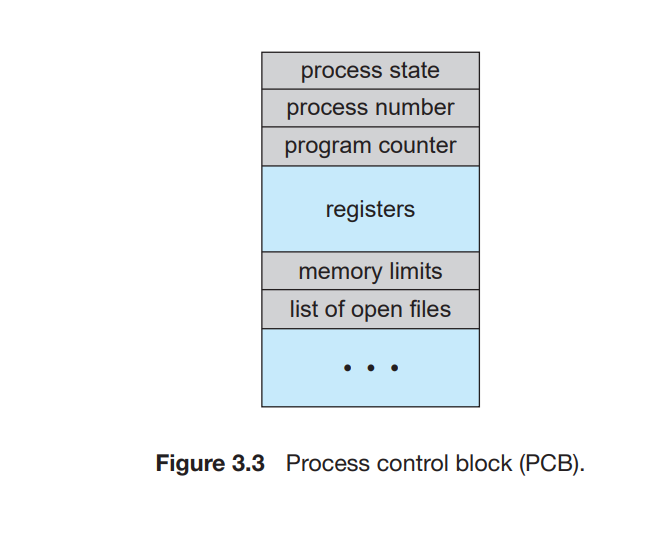
**5)Terminated:** The process has finished execution.

****

The process is in the new state when it is being created. Then the process is moved to the ready state, where it waits till it is taken for execution. There can be many such processes in the ready state. One of these processes will be selected and will be given the processor, and the selected process moves to the running state. A process, while running, may have to wait for I/O or wait for any other event to take place. That process is now moved to the waiting state. After the event for which the process was waiting gets completed, the process is moved back to the ready state. Similarly, if the time-slice of a process ends while still running, the process is moved back to the ready state. Once the process completes execution, it moves to the terminated state.

**3)Process Control Block (PCB):**

Each process is represented in the operating system by a process control block (PCB)—also called a task control block. A PCB is shown in below Figure

****

The information present in the PCB includes the following:

•Process state

•Process number

• Program counter

• CPU registers

• CPU scheduling information

• Memory-management information

• Accounting information

• I/O status information

**1)Process State:** Current state of the process. The state may be new, ready, running, waiting, terminated and so on.

**2)Process number:** Each process in the OS has its own unique identifier.

**3)Program counter:** The program counter indicates the address of the next instruction to be executed for this process.

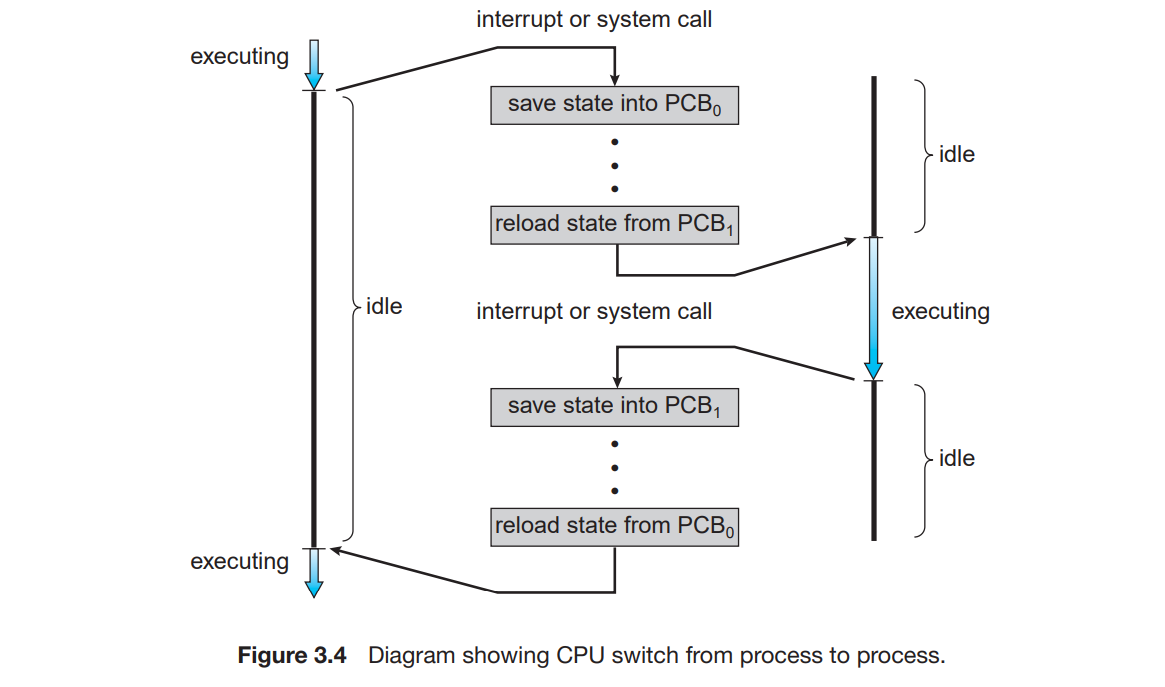
**4)CPU registers:** The registers vary in number and type, depending on the computer architecture. They include accumulators, index registers, stack pointers, and general-purpose registers, plus any condition-code information. Along with the program counter, this state information must be saved when an interrupt occurs, to allow the process to be continued correctly afterward.

**5)CPU-scheduling information:** This information includes a process priority, pointers to scheduling queues, and any other scheduling parameters.

**6)Memory-management information:** This information may include such information as the value of the base and limit registers, the page tables, or the segment tables, depending on the memory system used by the operating system .

**7)Accounting information:** This information includes the amount of CPU and real time used, time limits, account numbers, job or process numbers, and so on.

**8) I/O status information:** This information includes the list of I/O devices allocated to the process, a list of open files, and so on.



**4)Threads:**

The process model discussed so far has implied that a process is a program that performs a single thread of execution. For example, when a process is running a word-processor program, a single thread of instructions is being executed. This single thread of control allows the process to perform only one task at one time. The user cannot simultaneously type in characters and run the spell checker within the same process, for example. Most modern operating systems have extended the process concept to allow a process to have multiple threads of execution and thus to perform more than one task at a time.

