Paper Name : Operating System PCC-CS502

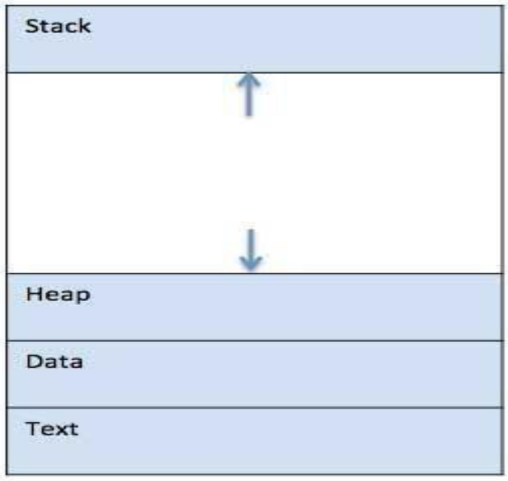
Process

A process is basically a program in execution. The execution of a process must progress in a sequential fashion.

Process Life Cycle

A process cycles through various execution states during it's execution lifetime. In general, a process can have one of the following five states at a time.

When a program is loaded into the memory and it becomes a process, it can be divided into four sections ─ stack, heap, text and data. The following image shows a simplified layout of a process inside main memory –



Stack

It is used for local variables. Space on the stack is reserved for local variables when they are declared.

Heap

This is dynamically allocated memory to a process during its run time.

Text

This includes the current activity represented by the value of Program Counter and the contents of the processor's registers.

Data

This section contains the global and static variables, allocated and initialized prior to executing the main.

When a process executes, it passes through different states. In general, a process can have one of the following five states at a time.

- **New**

In this step, the process is about to be created but not yet created, it is the program which is present in secondary memory that will be picked up by os to create the process.

- **Ready**

The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after start state or while running it by but interrupted by the scheduler to assign CPU to some other process.

- **Running**

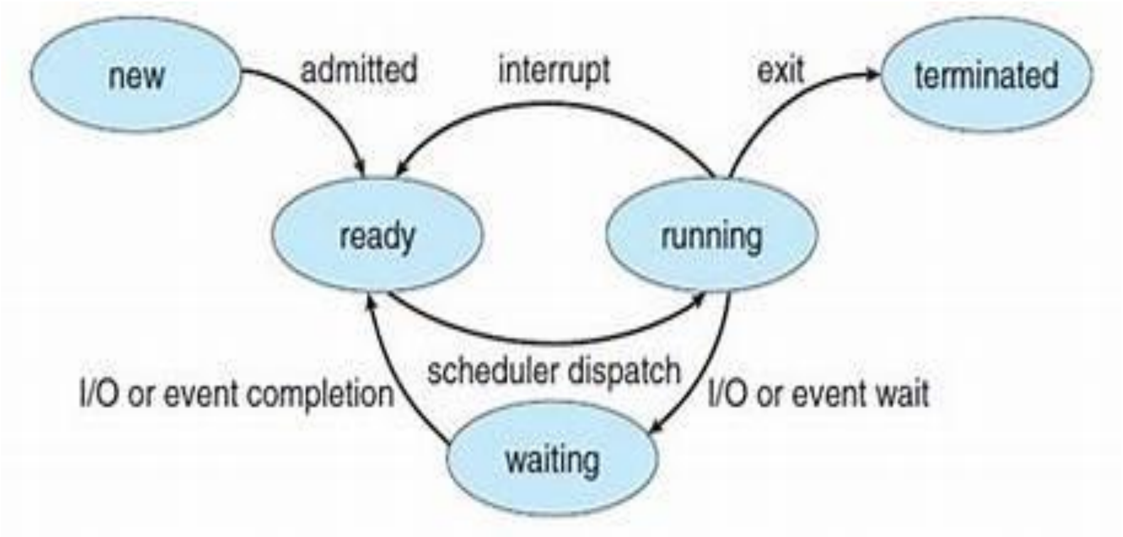
Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.

- **Waiting**

Process moves into the waiting state if it is waiting for some event to occur (such as an I/0 completion or reception of a signal). The process continues to wait in the main memory and does not require CPU. Once the I/O operation is completed the process goes to the ready state.

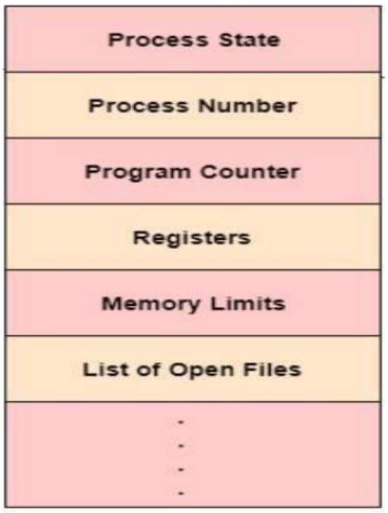
- **Terminated**

Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.



<u>Process Control Block (PCB)</u>

Whenever a process is created in the main memory , the operating system creates a data structure in the kernel space to store the various attributes of the process. This is known as process control block.


Process State

Process Number

Program Counter

Registers

Memory Limits

List of Open Files

-
-
-
-

Process Control Block (PCB)

The following are the data items −

Process State

This specifies the process state i.e. new, ready, running, waiting or terminated.

Process Number

This shows the number of the particular process.

Program Counter

This contains the address of the next instruction that needs to be executed in the process.

Registers

This specifies the registers that are used by the process. They may include accumulators, index registers, stack pointers, general purpose registers etc. 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.

CPU-scheduling information

This information includes a process priority, pointers to scheduling queues, and any other scheduling parameters.

Memory Management Information

The memory management information includes the page tables or the segment tables depending on the memory system used. It also contains the value of the base registers, limit registers etc.

I/O Status Information

This information includes the list of I/O devices allocated to the process, a list of open files etc.

Accounting information

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

In brief, the PCB simply serves as the repository for any information that may vary from process to process.

The operating system manages various types of queues for each of the process states. These are the following queues maintained by the operating system:

1. **Job Queue**
   In starting, all the processes get stored in the job queue. It is maintained in the secondary memory.

2. **Ready Queue**

It is a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.

3. **Device Queue**

A list of processes that are waiting for one or more I/O devices are lined up in a queue called device queue.