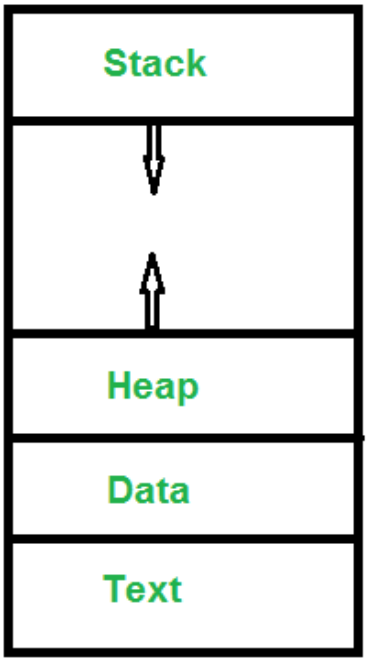
Program vs Process

A process is a program in execution. For example, 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, as opposed to a program, which is considered to be a 'passive' entity. A single program can create many processes when run multiple times; for example, when we open a .exe or binary file multiple times, multiple instances begin (multiple processes are created).

What does a process look like in memory?



Text Section:A Process, sometimes known as the Text Section, also includes the current activity represented by the value of the ***Program Counter***.
Stack: The Stack contains the temporary data, such as function parameters, returns addresses, and local variables.
Data Section: Contains the global variable.
Heap Section: Dynamically allocated memory to process during its run time.
Refer this for more details on sections.

Attributes or Characteristics of a Process

A process has following attributes.

1. **Process Id:** A unique identifier assigned by the operating system
2. **Process State:** Can be ready, running, etc.
3. **CPU registers:** Like the Program Counter (CPU registers must be saved and restored when a process is swapped in and out of CPU)
5. **Accounts information:**
6. **I/O status information:** For example, devices allocated to the process, open files, etc
8. **CPU scheduling information:** For example, Priority (Different processes may have different priorities, for example a short process may be assigned a low priority in the shortest job first scheduling)

All of the above attributes of a process are also known as the***context of the process***.
Every process has its own program control block(PCB), i.e each process will have a unique PCB. All of the above attributes are part of the PCB.

States of Process:

A process is in one of the following states:

1. **New:** Newly Created Process (or) being-created process.

2. **Ready:** After creation process moves to Ready state, i.e. the process is ready for execution.

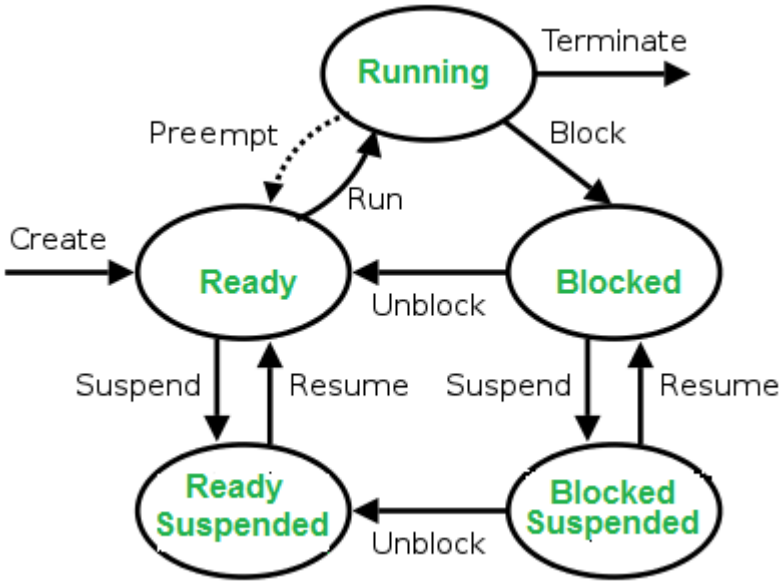
3. **Run:** Currently running process in CPU (only one process at a time can be under execution in a single processor).

4. **Wait (or Block):** When a process requests I/O access.

5. **Complete (or Terminated):** The process completed its execution.

6. **Suspended Ready:** When the ready queue becomes full, some processes are moved to suspended ready state

7. **Suspended Block:** When waiting queue becomes full.



Context Switching

The process of saving the context of one process and loading the context of another process is known as Context Switching. In simple terms, it is like loading and unloading the process from running state to ready state.

When does context switching happen?

1. When a high-priority process comes to ready state (i.e. with higher priority than the running process)
2. An Interrupt occurs
3. User and kernel mode switch (It is not necessary though)
4. Preemptive CPU scheduling used.

Context Switch vs Mode Switch

A mode switch occurs when CPU privilege level is changed, for example when a system call is made or a fault occurs. The kernel works in more a privileged mode than a standard user task. If a user process wants to access things which are only accessible to the kernel, a mode switch must occur. The currently executing process need not be changed during a mode switch.
A mode switch typically occurs for a process context switch to occur. Only the kernel can cause a context switch.

CPU-Bound vs I/O-Bound Processes:

A CPU-bound process requires more CPU time or spends more time in the running state.
An I/O-bound process requires more I/O time and less CPU time. An I/O-bound process spends more time in the waiting state.

Process Schedulers in Operating System

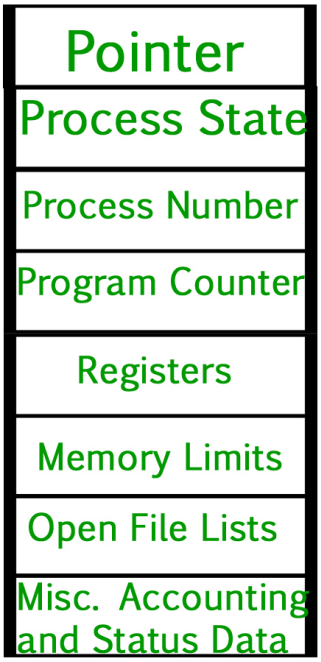
There are three types of process scheduler.

1. **Long Term or job scheduler** It brings the new process to the ‘Ready State’. It controls ***Degree of Multi-programming***, i.e., number of process present in ready state at any point of time.It is important that the long-term scheduler make a careful selection of both IO and CPU bound process.

2. **Short term or CPU scheduler:** It is responsible for selecting one process from ready state for scheduling it on the running state. Note: Short-term scheduler only selects the process to schedule it doesn’t load the process on running.
   Dispatcher is responsible for loading the process selected by Short-term scheduler on the CPU (Ready to Running State) Context switching is done by dispatcher only. A dispatcher does the following:
   1. Switching context.
   2. Switching to user mode.
   3. Jumping to the proper location in the newly loaded program.

3. **Medium-term scheduler** It is responsible for suspending and resuming the process. It mainly does swapping (moving processes from main memory to disk and vice versa). Swapping may be necessary to improve the process mix or because a change in memory requirements has overcommitted available memory, requiring memory to be freed up.

A process control block (PCB) contains information about the process, i.e. registers, quantum, priority, etc. The process table is an array of PCB's, that means logically contains a PCB for all of the current processes in the system.



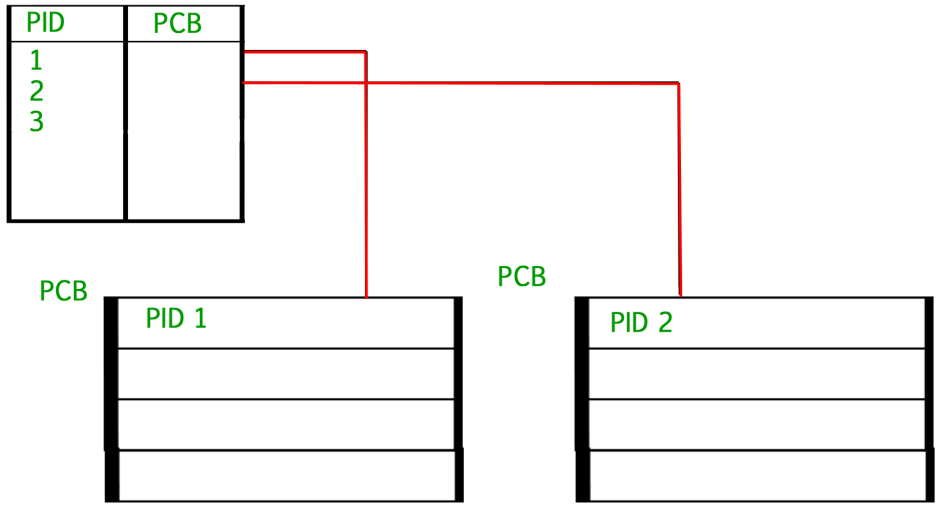
Process Control Block

- **Pointer –** It is a stack pointer which is required to be saved when the process is switched from one state to another to retain the current position of the process.
- **Process state –** It stores the respective state of the process.

- **Process number –** Every process is assigned with a unique id known as process ID or PID which stores the process identifier.
- **Program counter –** It stores the counter which contains the address of the next instruction that is to be executed for the process.
- **Register –** These are the CPU registers which includes: accumulator, base, registers and general purpose registers.
- **Memory limits –** This field contains the information about memory management system used by operating system. This may include the page tables, segment tables etc.
- **Open files list –** This information includes the list of files opened for a process.

Miscellaneous accounting and status data – This field includes information about the amount of CPU used, time constraints, jobs or process number, etc.

The process control block stores the register content also known as execution content of the processor when it was blocked from running. This execution content architecture enables the operating system to restore a process's execution context when the process returns to the running state. When the process made transitions from one state to another, the operating system update its information in the process's PCB. The operating system maintains pointers to each process's PCB in a process table so that it can access the PCB quickly.



Process table and process control block