Operating System - Processes

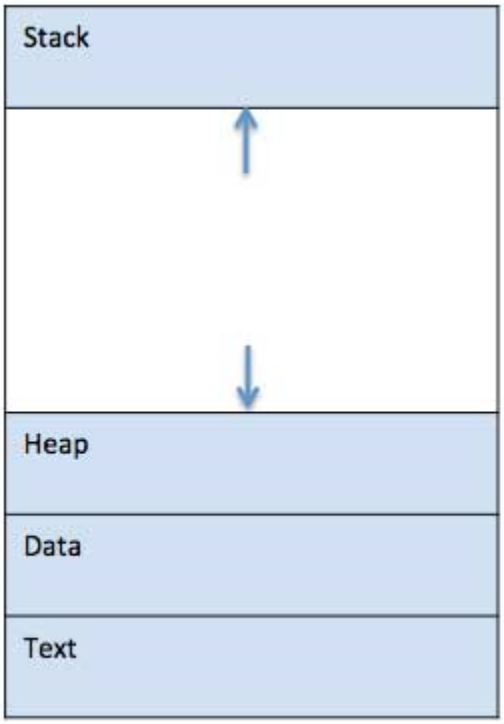
Process

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

A process is defined as an entity which represents the basic unit of work to be implemented in the system.

To put it 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.

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 −



S.N.	Component & Description
1	**Stack** The process Stack contains the temporary data such as method/function parameters, return address and local variables.
2	**Heap** This is dynamically allocated memory to a process during its run time.
3	**Text** This includes the current activity represented by the value of Program Counter and the contents of the processor's registers.
4	**Data** This section contains the global and static variables.

Program

A program is a piece of code which may be a single line or millions of lines. A computer program is usually written by a computer programmer in a programming language. For example, here is a simple program written in C programming language −

```
#include <stdio.h>

int main() {
   printf("Hello, World! \n");
   return 0;
}
```

A computer program is a collection of instructions that performs a specific task when executed by a computer. When we compare a program with a process, we can conclude that a process is a dynamic instance of a computer program.

A part of a computer program that performs a well-defined task is known as an **algorithm**. A collection of computer programs, libraries and related data are referred to as a **software**.

Difference between process and the program

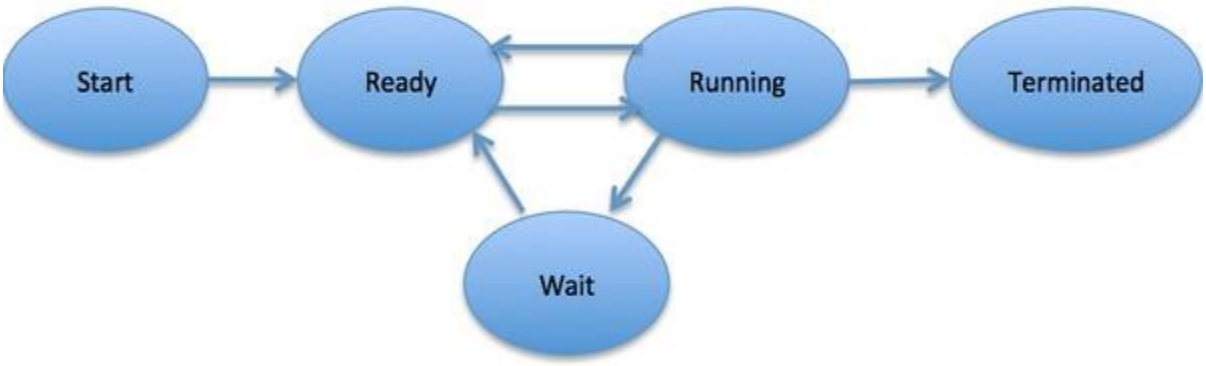
S. No	Process	Program
1	A process is actively running software or a computer code. Any procedure must be carried out in a precise order. An entity that helps in describing the fundamental work unit that must be implemented in any system is referred to as a process	Program is a set of instructions which are executed when the certain task is allowed to complete that certain task
2	Process is Dynamic in Nature	Program is Static in Nature
3	Process is an Active in Nature	Program is Passive in Nature
4	Process is created during the execution and it is loaded directly into the main memory	Program is already existed in the memory and it is present in the secondary memory.
5	Process has its own control system known as Process Control Block	Program does not have any control system. It is just called when specified and it executes the whole program when called
6	Process changes from time to time by itself	Program cannot be changed on its own. It must be changed by the programmer.
7	A process needs extra data in addition to the program data needed for management and execution.	Program is basically divided into two parts. One is Code part and the other part is data part.
8	Processes have significant resource demands; they require resources like Memory Addresses, Central Processing Unit, Input or Output until their presence or existence in the Operating System.	A program just needs memory space to store its instructions; no further resources are needed.

Process Life Cycle

When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.

In general, a process can have one of the following five states at a time.

S.N.	State & Description
1	**Start** This is the initial state when a process is first started/created.
2	**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.
3	**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.
4	**Waiting** Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.
5	**Terminated or Exit** 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.



Process Control Block (PCB)

A Process Control Block is a data structure maintained by the Operating System for every process. The PCB is identified by an integer process ID

(PID). A PCB keeps all the information needed to keep track of a process as listed below in the table –

S.N.	Information & Description
1	**Process State** The current state of the process i.e., whether it is ready, running, waiting, or whatever.
2	**Process privileges** This is required to allow/disallow access to system resources.
3	**Process ID** Unique identification for each of the process in the operating system.
4	**Pointer** A pointer to parent process.
5	**Program Counter** Program Counter is a pointer to the address of the next instruction to be executed for this process.

6 **CPU registers**
Various CPU registers where process need to be stored for execution for running state.

7 **CPU Scheduling Information**
Process priority and other scheduling information which is required to schedule the process.

8 **Memory management information**
This includes the information of page table, memory limits, Segment table depending on memory used by the operating system.

9 **Accounting information**
This includes the amount of CPU used for process execution, time limits, execution ID etc.

10 **IO status information**
This includes a list of I/O devices allocated to the process.

The architecture of a PCB is completely dependent on Operating System and may contain different information in different operating systems. Here is a simplified diagram of a PCB –

The PCB is maintained for a process throughout its lifetime, and is deleted once the process terminates.

Process Control Block (Repeat)

An Operating System helps in process creation, scheduling, and termination with the help of Process Control Block. The Process Control Block (PCB), which is part of the Operating System, aids in managing how processes operate. Every OS process has a Process Control Block related to it. By keeping data on different things including their state, I/O status, and CPU Scheduling, a PCB maintains track of processes.

Now, let us understand the Process Control Block with the help of the components present in the Process Control Block.

A Process Control Block consists of :

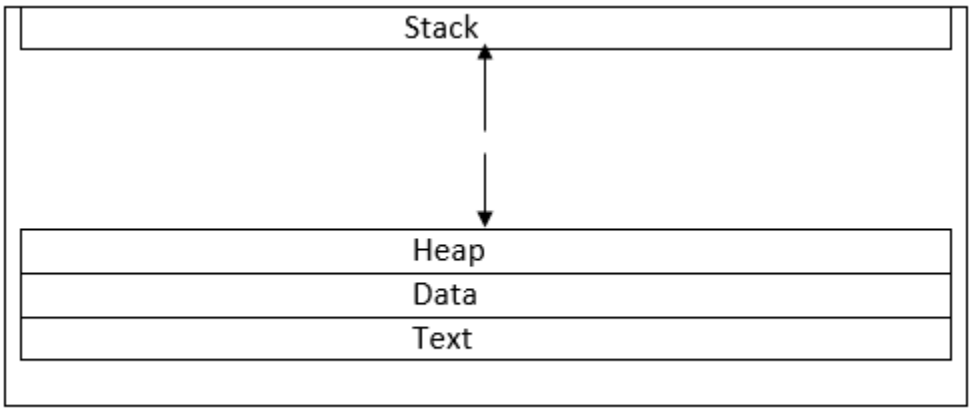
1. Process ID
2. Process State
3. Program Counter
4. CPU Registers
5. CPU Scheduling Information
6. Accounting and Business Information
7. Memory Management Information
8. Input Output Status Information

Now, let us understand about each and every component in detail now.

1) Process ID

It is a Identification mark which is present for the Process. This is very useful for finding the process. It is also very useful for identifying the process also.

2) Process State



Process Diagram

Process states in detail (explain about each and every state).

i) New State

A Program which is going to be taken up by the Operating System directly into the Main Memory is known as a New Process State

ii) Ready State

The ready state, when a process waits for the CPU to be assigned, is the first state it enters after being formed. The operating system pulls new processes from secondary memory and places them all in main memory.

The term "ready state processes" refers to processes that are in the main memory and are prepared for execution. Numerous processes could be active at the moment.

iii) Running State

The Operating System will select one of the processes from the ready state based on the scheduling mechanism. As a result, if our system only has one CPU, there will only ever be one process operating at any given moment. We can execute n processes concurrently in the system if there are n processors.

iv) Waiting or Blocking State

Depending on the scheduling mechanism or the inherent behavior of the process, a process can go from the Running state to the Block or Wait states.

The OS switches a process to the block or wait state and allots the CPU to the other processes while it waits for a specific resource to be allocated or for user input.

v) Terminated State

A process enters the termination state once it has completed its execution. The operating system will end the process and erase the whole context of the process (Process Control Block).

3) Program Counter

The address of the following instruction to be executed from memory is stored in a CPU register called a program counter (PC) in the computer processor. It is a digital counter required for both task execution speed and for monitoring the present stage of execution.

An instruction counter, instruction pointer, instruction addresses register, or sequence control register are other names for a program counter.

4) CPU Registers

When the process is in a running state, here is where the contents of the processor registers are kept. Accumulators, index and general-purpose registers, instruction registers, and condition code registers are the many categories of CPU registers.

5) CPU Scheduling Information

It is necessary to arrange a procedure for execution. This schedule determines when it transitions from ready to running. Process priority, scheduling queue pointers (to indicate the order of execution), and several other scheduling parameters are all included in CPU scheduling information.

6) Accounting and Business Information

The State of Business Addressing and Information includes information such as CPU use, the amount of time a process uses in real time, the number of jobs or processes, etc.

7) Memory Management Information

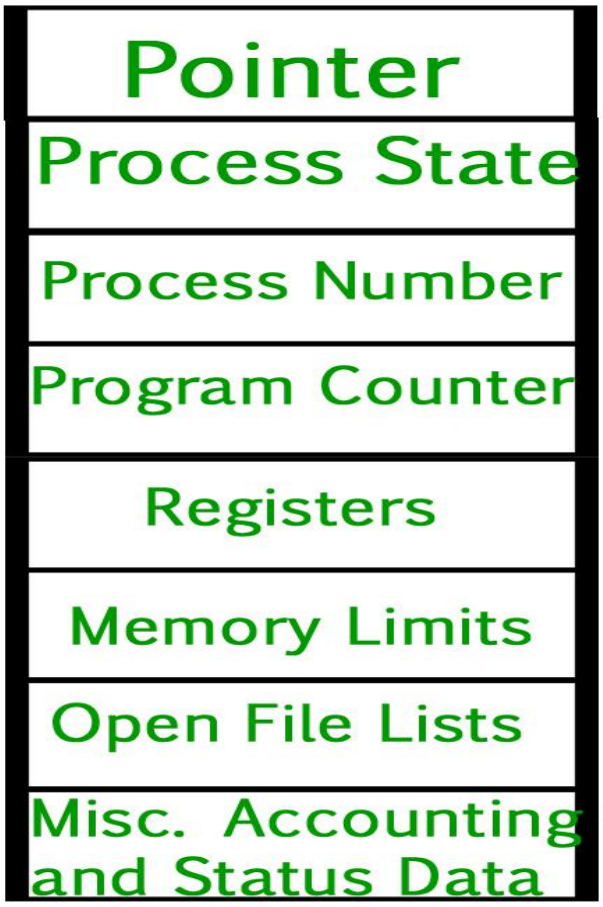
The Memory Management Information section contains information on the page, segment tables, and the value of the base and limit registers. It relies on the operating system's memory system.

8) Input Output Status Information

This Input Output Status Information section consists of Input and Output related information which includes about the process statuses, etc.

Process Table and Process Control Block (PCB)

While creating a process, the operating system performs several operations. To identify the processes, it assigns a process identification number (PID) to each process. As the operating system supports multi-programming, it needs to keep track of all the processes. For this task, the (PCB) is used to track the process's execution status. Each block of memory contains information about the process state, program counter, stack pointer, status of opened files, etc. All this information is required and must be saved when the process is switched from one state to another. When the process makes a transition from one state to another, the operating system must update information in the process's PCB. A process control block (PCB) contains information about the process, i.e. registers, quantum, priority, etc. The process table is an array of PCBs, that means logically contains for all of the current processes in the system.



Process Control Block

1. **Pointer:** It is a stack pointer that is required to be saved when the process is switched from one state to another to retain the current position of the process.
2. **Process state:** It stores the respective state of the process.
3. **Process number:** Every process is assigned a unique id known as process ID or PID which stores the process identifier.
4. **Program counter:** It stores the counter, which contains the address of the next instruction that is to be executed for the process.
5. **Register:** These are the CPU registers which include the accumulator, base, registers, and general-purpose registers.
6. **Memory limits:** This field contains the information about memory management system used by the operating system. This may include page tables, segment tables, etc.
7. **Open files list**: This information includes the list of files opened for a process.

Additional Points to Consider for Process Control Block (PCB)

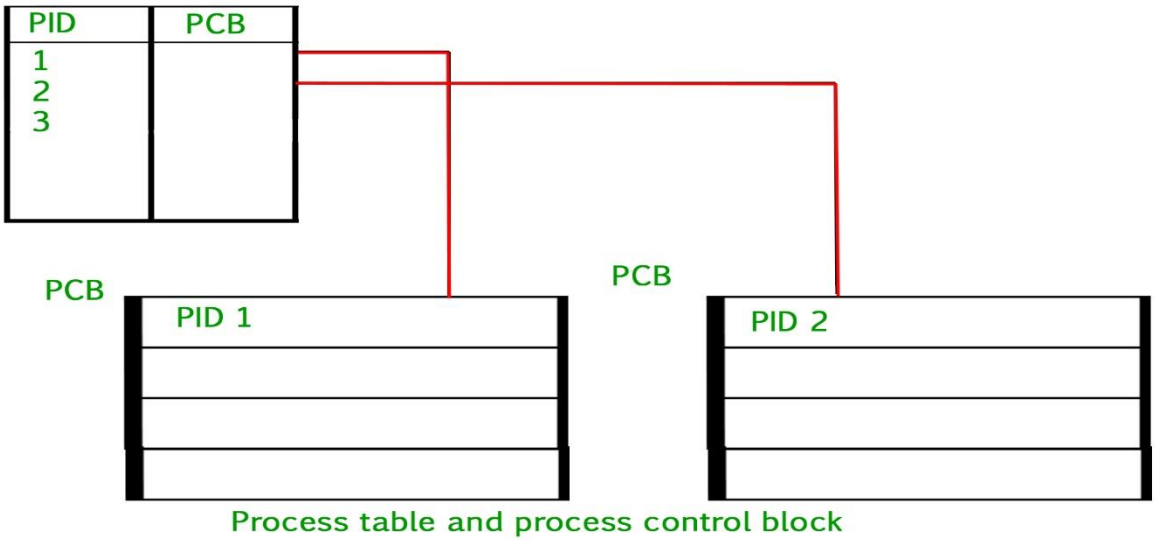
- **Interrupt handling:** The PCB also contains information about the interrupts that a process may have generated and how they were handled by the operating system.
- **Context switching:** The process of switching from one process to another is called context switching. The PCB plays a crucial role in context switching by saving the state of the current process and restoring the state of the next process.
- **Real-time systems:** Real-time operating systems may require additional information in the PCB, such as deadlines and priorities, to ensure that time-critical processes are executed in a timely manner.
- **Virtual memory management:** The PCB may contain information about a process's management, such as page tables and page fault handling.
- **Inter-process communication:** The PCB can be used to facilitate inter-process communication by storing information about shared resources and communication channels between processes.
- **Fault tolerance:** Some operating systems may use multiple copies of the PCB to provide fault tolerance in case of hardware failures or software errors.

Advantages

1. **Efficient process management:** The process table and PCB provide an efficient way to manage processes in an operating system. The process table contains all the information about each process, while the PCB contains the current state of the process, such as the program counter and CPU registers.
2. **Resource management:** The process table and PCB allow the operating system to manage system resources, such as memory and CPU time, efficiently. By keeping track of each process's resource usage, the operating system can ensure that all processes have access to the resources they need.
3. **Process synchronization:** The process table and PCB can be used to synchronize processes in an operating system. The PCB contains information about each process's synchronization state, such as its waiting status and the resources it is waiting for.
4. **Process scheduling:** The process table and PCB can be used to schedule processes for execution. By keeping track of each process's state and resource usage, the operating system can determine which processes should be executed next.

Disadvantages

1. **Overhead:** The process table and PCB can introduce overhead and reduce system performance. The operating system must maintain the process table and PCB for each process, which can consume system resources.
2. **Complexity:** The process table and PCB can increase system complexity and make it more challenging to develop and maintain operating systems. The need to manage and synchronize multiple processes can make it more difficult to design and implement system features and ensure system stability.
3. **Scalability:** The process table and PCB may not scale well for large-scale systems with many processes. As the number of processes increases, the process table and PCB can become larger and more difficult to manage efficiently.
4. **Security:** The process table and PCB can introduce security risks if they are not implemented correctly. Malicious programs can potentially access or modify the process table and PCB to gain unauthorized access to system resources or cause system instability.
5. **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 makes a transition from one state to another, the operating system updates 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

Context Switching

Context switching is the process of saving the current state of a running process and loading the state of another process so that the CPU can switch its execution from one process to another. The process control block (PCB) plays a key role in context switching because it contains all relevant information about the process. When the operating system decides to switch to another process, it stores the current process in the circuit's memory, including CPU registers and program counters. It then loads the chip to start the next process, resets its state and resumes execution from where it left off. This seamless switching between processes allows the operating system to create the illusion of simultaneous execution, even though the processor can only run one process at a time.