Program:
A program is a bunch of instructions written in a computer language that tells the computer what to do. These instructions are designed to be carried out by the computer's brain, known as the central processing unit (CPU).

Process:
When we want the computer to do something with a program, a special thing happens:
1.If the program isn't already in the computer's memory, the computer makes a copy of the program(instance).
2.This copy is made on the hard disk, a storage place inside the computer.
3.Then, this copied version is brought into the computer's memory, which is called RAM. This helps the computer do things faster.
Note: The copy of the program is made right on the hard disk.

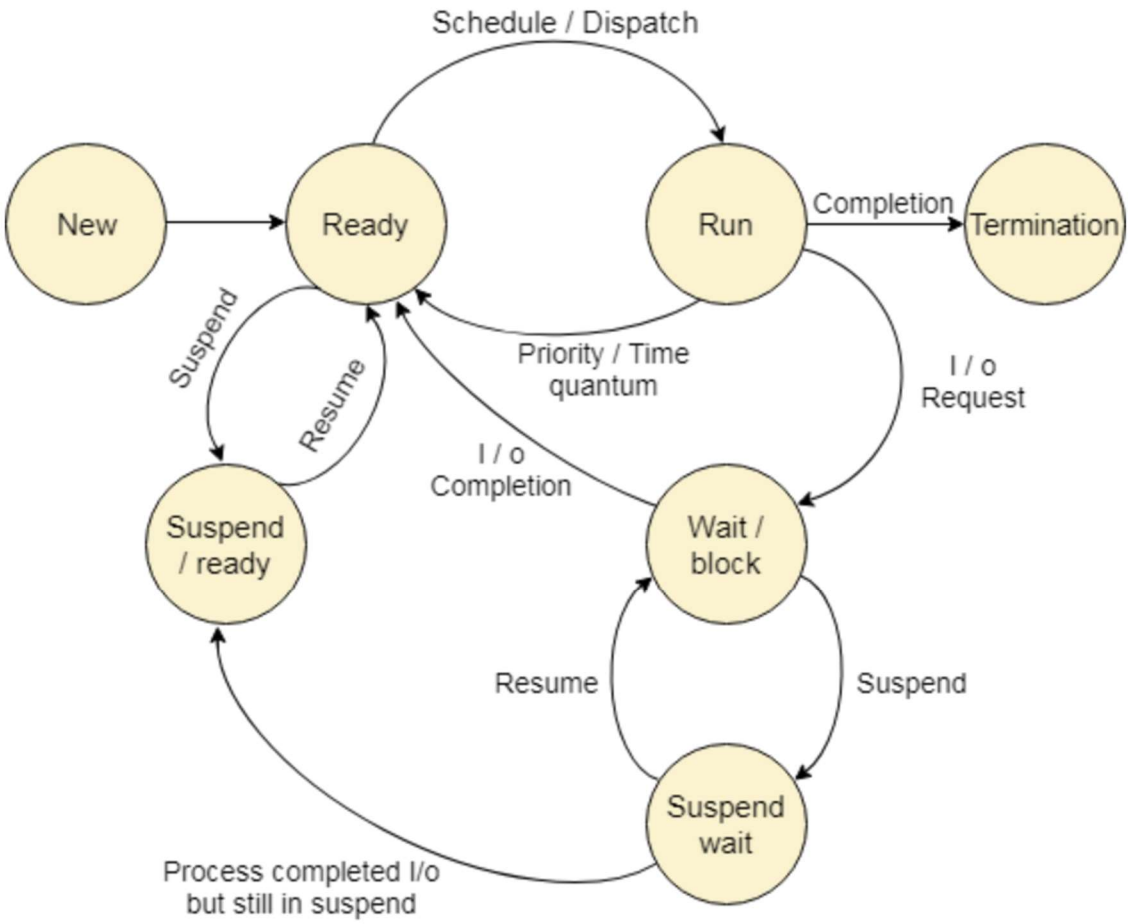
Example:
Think about Google Chrome:
1.When we download Chrome, its special code (the Program) gets saved in the computer's hard disk.
2.When we click to open Chrome, a fresh copy of the program is created on the hard disk. If that program in not there in RAM
3.Then, this new copy is moved into the computer's memory (RAM), and this allows Chrome to start up quickly.
4.If we open more tabs in Chrome, the computer creates new copies for each tab, each time on the hard disk.
Note: Each of these copies is called a "process."

So, a program is like a set of instructions, and the copies of these instructions that the computer makes are called processes.

Program	Process
A program is a set of sequential ordered operations to accomplish a programming goal.	Executing part of the program is known as a process.
Program is generally static.	Process is generally dynamic.
The program needs memory (primary/secondary) for storage.	The Resource requirement is high for the process.
There is no duplication needed in the case of the program.	A new sub/child process requires replication/duplication of their parent process.
A program is stored on disk (secondary memory) or in it contains by some file for eg. batch script and does not require any other resources.	Process captures resources like CPU, memory(primary), address, disk, I/O, etc and sometimes it produces deadlock situations.
A program needs memory space on the disk to store all instructions.	A process occupies many resources like memory, disk, mouse, keyboard, and other external devices.
The nature of the program is passive, does not do anything until it gets executed by allocating the resource.	The process is an instance of the program executed.

Program has no computation time and cost.	**Process** has considerable single fact access and computation time.
Program can be optimized to reduce the execution time and process.	Process is a singleton entity and the execution depends on the processor.
Deadlock creation and prevention depend on the program.	Deadlock condition process stucks and waits for the resource consumed by the process which is waiting for the resource holding by the previous process.
Program can control the memory resources and data needs to be used by memory.	Process and its information can be controlled by process control block (PCB).
Program cannot be decomposed or divided further and there is no such threshing in the program.	The process can be further decomposed into the thread which is a part or instance of the single process.
Program can contain the instruction to use cache for its data.	Process can use cache to store and retrieve the data as it uses operating system paging scheme and cache replacement policy such as FCFS, LIFO, LRU, RR.

State Diagram



The process, from its creation to completion, passes through various states. The minimum number of states is five.

The names of the states are not standardized although the process may be in one of the following states during execution.

1. New

A program which is going to be picked up by the OS into the main memory is called a new process.

2. Ready

Whenever a process is created, it directly enters in the ready state, in which, it waits for the CPU to be assigned. The OS picks the new processes from the secondary memory and put all of them in the main memory.

The processes which are ready for the execution and reside in the main memory are called ready state processes. There can be many processes present in the ready state.

3. Running

One of the processes from the ready state will be chosen by the OS depending upon the scheduling algorithm. Hence, if we have only one CPU in our system, the number of running processes for a particular time will always be one. If we have n processors in the system then we can have n processes running simultaneously.

4. Block or wait

From the Running state, a process can make the transition to the block or wait state depending upon the scheduling algorithm or the intrinsic behavior of the process.

When a process waits for a certain resource to be assigned or for the input from the user then the OS move this process to the block or wait state and assigns the CPU to the other processes.

5. Completion or termination

When a process finishes its execution, it comes in the termination state. All the context of the process (Process Control Block) will also be deleted the process will be terminated by the Operating system.

6. Suspend ready

A process in the ready state, which is moved to secondary memory from the main memory due to lack of the resources (mainly primary memory) is called in the suspend ready state.

If the main memory is full and a higher priority process comes for the execution then the OS have to make the room for the process in the main memory by throwing the lower priority process out into the secondary memory. The suspend ready processes remain in the secondary memory until the main memory gets available.

7. Suspend wait

Instead of removing the process from the ready queue, it's better to remove the blocked process which is waiting for some resources in the main memory. Since it is already waiting for some resource to get available hence it is better if it waits in the secondary memory and make room for the higher priority process. These processes complete their execution once the main memory gets available and their wait is finished.

Operations on the Process

1. Creation

Once the process is created, it will be ready and come into the ready queue (main memory) and will be ready for the execution.

2. Scheduling

Out of the many processes present in the ready queue, the Operating system chooses one process and start executing it. Selecting the process which is to be executed next, is known as scheduling.

3. Execution

Once the process is scheduled for the execution, the processor starts executing it. Process may come to the blocked or wait state during the execution then in that case the processor starts executing the other processes.

4. Deletion/killing

Once the purpose of the process gets over then the OS will kill the process. The Context of the process (PCB) will be deleted and the process gets terminated by the Operating system.