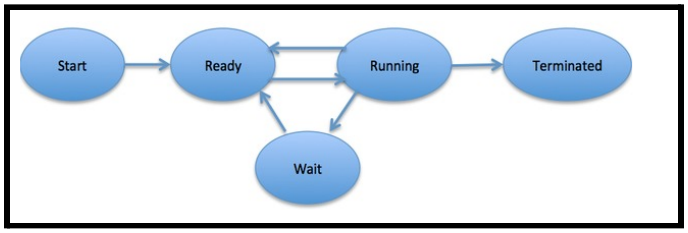
Processes:

- A **process** is the execution of a program that allows you to perform the appropriate actions specified in a program.
- It can be defined as an execution unit where a program runs. The OS helps you to create, schedule, and terminate the processes which are used by the CPU.
- The processes created by the main process are called **child processes**.
- A process's operations can be easily controlled with the help of **PCB (Process Control Block)**.
- A PCB is a data structure maintained by the OS for every process. The PCB is identified by a process ID (PID) and contains the following information:
    1. **Process State:** The current state of the process. (I.e. Whether it is ready, running, waiting, etc.)
    2. **Process Privileges:** This is required to allow/disallow access to system resources.
    3. **Process ID (PID):** A PID is a unique identification for each of the processes in the os.
    4. **Pointer:** A pointer to the parent process.
    5. **Program Counter (PC):** The PC is a pointer to the address of the next instruction to be executed for this process.
    6. **CPU registers:**
    7. **CPU Scheduling Information:** Used to process priority and other scheduling information which are required to schedule the processes.
    8. **Accounting Information:** This includes the amount of CPU used for process execution, time limits, execution ID etc.
    9. **IO Status Information:** A list of I/O devices allocated to the process.
- A process will be in 1 of the following 5 stages at a given time:
    1. Start
    2. Ready
    3. Running
    4. Blocked/Waiting
    5. Terminated/Exited/Finished

  Here is a picture of a process's flow:



- The OS has 3 queues to keep track of processes:
    1. **Job Queue:** This queue keeps all the processes in the system.
    2. **Ready Queue:** This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.
    3. **Waiting Queue:** The processes which are blocked due to unavailability of an I/O device are in this queue.
- A process can have multiple threads.

Threads:

- A **thread** is a basic unit of CPU utilization, consisting of a program counter, a stack, and a set of registers, and thread ID.
- A thread is also called a lightweight process.
- Traditional/heavyweight processes have a single thread of control, but multi-threaded applications have multiple threads within a single process, each having their own program counter, stack and set of registers, but sharing common code, data, and certain structures such as open files.
- Each thread belongs to exactly one process and no thread can exist outside a process.

Context Switching:

- A **context switch** is the process of storing the state of a process or thread, so that it can be restored and resume execution at a later point. This allows multiple processes to share a single CPU, and is an essential feature of a multitasking operating system.

Processes vs Threads:

Parameter	**Processes**	**Threads**
Definition	A program in execution	A segment of a process
Termination time	Take more time to terminate	Take less time to terminate
Creation time	Take more time for creation	Take less time for creation
Communication	Communication between processes needs more time	Communication between threads requires less time
Context switching time	Take more time	Take less time
Resource	Consume more resources	Consume fewer resources
Sharing	Do not share data	Share data with each other