What is a process?

A **process** is essentially a program in action. When a program is executed, it becomes a process. It is not just a static set of instructions but includes additional elements like the program's current state, allocated resources, and its interaction with the operating system.

Key Components of a Process:

1. **Program Code**: The set of instructions to be executed.
2. **Program Counter**: Keeps track of the next instruction to execute.
3. **Registers**: Temporary storage within the CPU used during execution.
4. **Stack**: Contains temporary data like function parameters, return addresses, and local variables.
5. **Heap**: Provides memory for dynamic variables.
6. **Data Section**: Contains global variables used by the program.

Process Lifecycle:

A process goes through the following stages:

1. **New**: The process is created.
2. **Ready**: The process is waiting in the queue for CPU time.
3. **Running**: The process is actively being executed by the CPU.
4. **Waiting**: The process is paused, waiting for an event or resource.
5. **Terminated**: The process has completed execution or was stopped.

Processes can be of different types:

- **Foreground Processes**: Interact directly with the user (e.g., opening applications like a browser).
- **Background Processes**: Run behind the scenes, without user interaction (e.g., system updates, antivirus scans).