



Lec-9: Introduction to Process

1. What is a program? Compiled code, that is ready to execute.
2. What is a process? Program under execution.
3. How OS creates a process? Converting program into a process.

STEPS:

- a. Load the program & static data into memory.
 - b. Allocate runtime stack.
 - c. Heap memory allocation.
 - d. IO tasks.
 - e. OS handoffs control to main ().
4. **Architecture of process:**

Stack	Local variables, function arguments & return values
Heap	Dynamically allocated variables
Data	Global & Static data
Text	Compiled code (Loaded from disk)

5. **Attributes of process:**

- a. Feature that allows identifying a process uniquely.
- b. Process table
 - i. All processes are being tracked by OS using a table like data structure.
 - ii. Each entry in that table is process control block (PCB).
- c. PCB: Stores info/attributes of a process.
 - i. Data structure used for each process, that stores information of a process such as process id, program counter, process state, priority etc.

6. **PCB structure:**

Process ID	Unique identifier
Program Counter (PC)	Next instruction address of the program
Process State	Stores process state
Priority	Based on priority a process gets CPU time
Registers	
List of open files	
List of open devices	

Registers in the PCB, it is a data structure. When a processes is running and it's time slice expires, the current value of process specific registers would be stored in the PCB and the process would be swapped out. When the process is scheduled to be run, the register values is read from the PCB and written to the CPU registers. This is the main purpose of the registers in the PCB.