## **Practical No. 7: Priority-Based CPU Scheduling Algorithm (Preemptive & Non-Preemptive)**

### **Aim:**

To write a C program to implement **priority-based CPU scheduling**, supporting both **preemptive** and **non-preemptive** modes.

### **Concepts Involved:**

* **Priority Scheduling:** CPU is allocated to the process with the highest priority (lower numerical value = higher priority).
* **Preemptive Mode:** Higher priority process can interrupt the current process.
* **Non-Preemptive Mode:** Once a process starts, it runs till completion even if a higher priority process arrives.

### **Sample Execution 1:**

**Inputs:**

makefile

CopyEdit

Processes: 2

P1: Arrival=0, Burst=11, Priority=2

P2: Arrival=5, Burst=28, Priority=0

Mode: 1 (Non-Preemptive)

**Output:**

nginx

CopyEdit

ID AT BT Priority CT TAT WT

P1 0 11 2 11 11 0

P2 5 28 0 39 34 6

### **Sample Execution 2:**

**Inputs:**

makefile

CopyEdit

Processes: 5

P1: AT=0, BT=11, Priority=2

P2: AT=5, BT=28, Priority=0

P3: AT=2, BT=3, Priority=1

P4: AT=10, BT=1, Priority=4

P5: AT=9, BT=16, Priority=4

Mode: 1 (Non-Preemptive)

**Output:**

nginx

CopyEdit

ID AT BT Priority CT TAT WT

P1 0 11 2 49 49 38

P2 5 28 0 39 34 6

P3 2 3 1 14 12 9

P4 10 1 4 40 30 29

P5 9 16 4 32 23 7

### **Conclusion:**

The program accurately simulates **Priority Scheduling** by processing tasks based on their priority levels. It provides flexibility to choose between preemptive and non-preemptive scheduling, and calculates key metrics: **Completion Time, Turnaround Time,** and **Waiting Time**.