# Assignment - 04
# TOPIC: Process Scheduling- PART2

1. **Write a C program to simulate a multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.**

**->Ans:**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_PROCESSES 10

typedef struct {
    int pid;
    char type[10]; // "system" or "user"
    int arrival_time;
    int burst_time;
    int waiting_time;
    int turnaround_time;
} Process;

void sort_by_arrival_time(Process *queue, int n) {
    Process temp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (queue[j].arrival_time > queue[j + 1].arrival_time) {
                temp = queue[j];
                queue[j] = queue[j + 1];
                queue[j + 1] = temp;
            }
        }
    }
}
void calculate_waiting_and_turnaround_time(Process *queue, int n) {
    queue[0].waiting_time = 0;
    queue[0].turnaround_time = queue[0].burst_time;
    for (int i = 1; i < n; i++) {
        queue[i].waiting_time = queue[i - 1].waiting_time + queue[i -
1].burst_time;
        queue[i].turnaround_time = queue[i].waiting_time +
queue[i].burst_time;
    }
}
void print_queue(Process *queue, int n) {
    printf("PID\tType\tArrival Time\tBurst Time\tWaiting Time\tTurnaround
Time\n");
    for (int i = 0; i < n; i++) {
```

```
        printf("%d\t%s\t%d\t\t%d\t\t%d\t\t%d\n",
               queue[i].pid, queue[i].type, queue[i].arrival_time,
               queue[i].burst_time, queue[i].waiting_time,
queue[i].turnaround_time);
    }
}

int main() {
    Process processes[MAX_PROCESSES];
    int n;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        printf("Enter details for process %d\n", i + 1);
        printf("PID: ");
        scanf("%d", &processes[i].pid);
        printf("Type (system/user): ");
        scanf("%s", processes[i].type);
        printf("Arrival Time: ");
        scanf("%d", &processes[i].arrival_time);
        printf("Burst Time: ");
        scanf("%d", &processes[i].burst_time);
    }
    // Separate system and user processes
    Process system_queue[MAX_PROCESSES], user_queue[MAX_PROCESSES];
    int system_count = 0, user_count = 0;
    for (int i = 0; i < n; i++) {
        if (strcmp(processes[i].type, "system") == 0) {
            system_queue[system_count++] = processes[i];
        } else if (strcmp(processes[i].type, "user") == 0) {
            user_queue[user_count++] = processes[i];
        }
    }
    // Sort both queues by arrival time
    sort_by_arrival_time(system_queue, system_count);
    sort_by_arrival_time(user_queue, user_count);
    // Calculate waiting time and turnaround time for system processes
    printf("\nSystem Processes Queue:\n");
    calculate_waiting_and_turnaround_time(system_queue, system_count);
    print_queue(system_queue, system_count);
    // Calculate waiting time and turnaround time for user processes
    printf("\nUser Processes Queue:\n");
    calculate_waiting_and_turnaround_time(user_queue, user_count);
    print_queue(user_queue, user_count);
    return 0;
}
```

**Output:**

```
abhignya@hplaptop:~/MCA2023/Abhignya_B_16/Assignment1$ vim MultilevelQueueScheduling.c
abhignya@hplaptop:~/MCA2023/Abhignya_B_16/Assignment1$ gcc MultilevelQueueScheduling.c -o MultilevelQueueScheduling
abhignya@hplaptop:~/MCA2023/Abhignya_B_16/Assignment1$ ./MultilevelQueueScheduling
Enter the number of processes: 4
Enter details for process 1
PID: 1
Type (system/user): system
Arrival Time: 0
Burst Time: 4
Enter details for process 2
PID: 2
Type (system/user): user
Arrival Time: 1
Burst Time: 3
Enter details for process 3
PID: 3
Type (system/user): user
Arrival Time: 2
Burst Time: 2
Enter details for process 4
PID: 4
Type (system/user): system
Arrival Time: 3
Burst Time: 1

System Processes Queue:
PID     Type    Arrival Time    Burst Time      Waiting Time    Turnaround Time
1       system  0               4               0               4
4       system  3               1               4               5

User Processes Queue:
PID     Type    Arrival Time    Burst Time      Waiting Time    Turnaround Time
2       user    1               3               0               3
3       user    2               2               3               5
```