**Bahria University, Lahore Campus**

Department of Computer Sciences

Lab Journal 9

**(Spring 2024)**

|  |  |  |
| --- | --- | --- |
| Course: | **Operating System Lab** | Date: 5/16/24 |
| Course Code: | CSL – 320 | Max Marks: 20 |
| Faculty’s Name: | ABDULLAH |  |

Name: \_**Muhammad Hammad** Enroll No: **03-134221-024**

Objective(s) :

To write a C program to implement CPU scheduling algorithm for Round Robin.

## Lab Tasks :

### Task 01: Calculate the Average Time using Round Robin. Draw the GANTT Chart.

**Task 02:** Write the output for Round Robin Scheduling Algorithm.

**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| 1. | 10 |  |  |
| 2. | 10 |  |  |
| **Total** | **20** |  | **Signature** |

**Note : Attempt all tasks and get them checked by your Lab. Instructor**.

# Lab 9: Round Robin Scheduling

**Objective(s):**

To write a C program to implement CPU scheduling algorithm for Round Robin.

**Tool(s) used:**

Ubuntu, VIM Editor

CPU scheduler will decide which process should be given the CPU for its execution. For this its use different algorithm to choose among the process. One algorithm among that is Round robin algorithm. In this algorithm we are assigning some time slice. The process is allocated according to the time slice, if the process service time is less than the time slice then process itself will release the CPU voluntarily. The scheduler will then proceed to the next process in the ready queue. If the CPU burst of the currently running process is longer than time quantum, the timer will go off and will cause an interrupt to the operating system. A context switch will be executed and the process will be put at the tail of the ready queue.

### Task 01: Calculate the Average Time using Round Robin. Draw the GANTT Chart. TQ=2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Process No | Arrival Time (AT) | Burst Time (BT) | Complete Time (CT) | Turnaround Time (CT-TAT) | Waiting Time (TAT-BT) | Response Time |
| P1 | 0 | 10 | 20 | 20 | 10 | 0 |
| P2 | 1 | 4 | 10 | 9 | 10 | 1 |
| P3 | 2 | 5 | 17 | 15 | 10 | 2 |
| P4 | 3 | 3 | 13 | 10 | 10 | 3 |

Ready Queue:

## Process P1:

## AT: 0, BT: 10

## Remaining BT: 10

## Process P2:

## AT: 1, BT: 4

## Remaining BT: 4

## Process P3:

## AT: 2, BT: 5

## Remaining BT: 5

## Process P4:

## AT: 3, BT: 3

## Remaining BT: 3

Gantt Chart:

| P1 | P1 | P2 | P2 | P3 | P3 | P4 | P4 | P1 | P3 | P1 |

## Average Turnaround Time: 13.5

## Average Waiting Time: 8

1. **Step 2: Calculate Turnaround Time (TAT), and Waiting Time (WT):**

Average Turnaround Time (TAT):

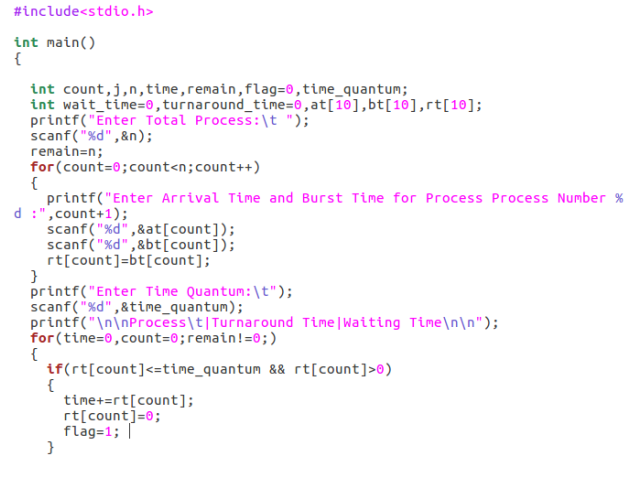
(TAT\_P1 + TAT\_P2 + TAT\_P3 + TAT\_P4) / 4

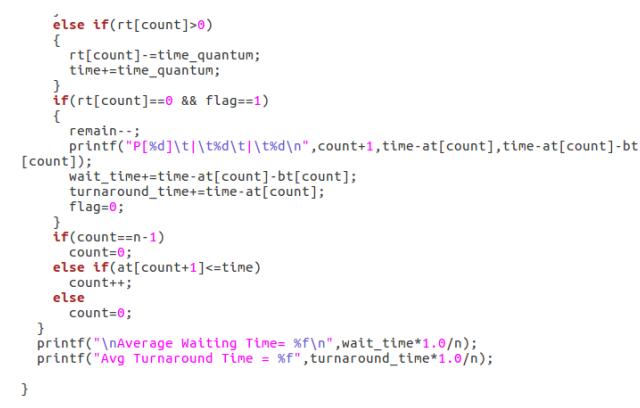
Average Waiting Time (WT):

(WT\_P1 + WT\_P2 + WT\_P3 + WT\_P4) / 4

### Task 02: Write the output of Round Robin Scheduling Algorithm. Attach the screenshot of dry run of code.

**Code**





**OUTPUT**

|  |
| --- |
| Code:  #include <iostream>  using namespace std;  int main() {  int count, j, n, time, remain, flag = 0, time\_quantum;  int wait\_time = 0, turnaround\_time = 0, at[10], bt[10], rt[10];  cout << "Enter Total Process: \t";  cin >> n;  remain = n;  for (count = 0; count < n; count++) {  cout << "Enter Arrival Time and Burst Time for Process Process Number " << count + 1 << ": ";  cin >> at[count] >> bt[count];  rt[count] = bt[count];  }  cout << "Enter Time Quantum: \t";  cin >> time\_quantum;  cout << "\n\nProcess\t | Turnaround Time | Waiting Time\n\n";    for (time = 0, count = 0; remain != 0;) {  if (rt[count] <= time\_quantum && rt[count] > 0) {  time += rt[count];  rt[count] = 0;  flag = 1;  }  else if (rt[count] > 0) {  rt[count] -= time\_quantum;  time += time\_quantum;  }  if (rt[count] == 0 && flag == 1) {  remain--;  cout << "P[" << count + 1 << "]\t\t" << time - at[count] << "\t\t" << time - at[count] - bt[count] << endl;  wait\_time += time - at[count] - bt[count];  turnaround\_time += time - at[count];  flag = 0;  }  count = (count + 1) % n; // Move to the next process in a circular manner  }  cout << "\nAverage Waiting Time= " << wait\_time \* 1.0 / n << endl;  cout << "Avg Turnaround Time = " << turnaround\_time \* 1.0 / n << endl;  return 0;  } |
| Screenshot: |