ASSIGNMENT

For

Three students (a, b, c) are arriving in the mess at the same time. The id numbers of these students are 2132, 2102, 2453 and the food taken time from the mess table is 2, 4 and 8 minutes. If the two students have same remaining time so it is broken by giving priority to the students with the lowest id number. Consider the longest remaining time first (LRTF) scheduling algorithm and calculate the average turnaround time and waiting time.

Table of Contents

1.Introduction

1.1 Purpose

This document describes the Assignment and specification for a LRTF (Longest remaining time first) scheduling algorithm and calculate the average turnaround time and waiting time.

We have taken such a case that,

If the two students have same remaining time so it is broken by giving priority to the students with the lowest id number.

1.2 Minimum Requirements and Reading Suggestions already stated

->A DESCENT LAPTOP OR PC

The given Assignment project will be developed in Dev C++ IDE using C language.

2. Overall Description

2.1 Here are the working conditions:

- User should be able to give the input (sitting at the computer).
- The board should be printed out every time given no of students with their IDs are proceeded.
- User should be able to get average turnaround time and waiting time.

Algorithm –

- **Step-1:** Create a structure of process containing all necessary fields like AT (Arrival Time), BT (Burst Time), CT (Completion Time), TAT (Turn Around Time), WT (Waiting Time).
- **Step-2:** Sort according to the AT;
- **Step-3:** Find the process having Largest Burst Time and execute for each single unit. Increase the total time by 1 and reduce the Burst Time of that process with 1.
- **Step-4:** When any process has 0 BT left, then update the CT (Completion Time of that process CT will be Total Time at that time).
- **Step-5:** After calculating the CT for each process, find TAT and WT.

3. CPU Scheduling | Longest Remaining Time First (LRTF) algorithm

3.1 (Algorithms Used):

This is a pre-emptive version of Longest Job First (LJF) scheduling algorithm. In this scheduling algorithm, we find the process with maximum remaining time and then process it. We check for the maximum remaining time after some interval of time (say 1 unit each) to check if another process having more Burst Time arrived up to that time.

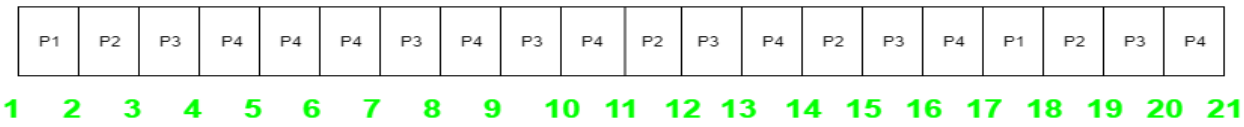
3.2 Procedure:

- **Step-1:** First, sort the processes in increasing order of their Arrival Time.
- **Step-2:** Choose the process having least arrival time but with most Burst Time. Then process it for 1 unit. Check if any other process arrives up to that time of execution or not.
- **Step-3:** Repeat the above both steps until execute all the processes.

Example: Consider the following table of arrival time and burst time for four processes P1, P2, P3 and P4.

Process	Arrival time	Burst Time
P1	1 ms	2 ms
P2	2 ms	4 ms
P3	3 ms	6 ms
p4	4 ms	8 ms

3.3 Gantt chart will be as following below,



Since, completion time (CT) can be directly determined by Gantt chart, and

Therefore,

Output:

Total Turn Around Time = 68 ms
So, Average Turn Around Time = 68/4 = 17.00 ms

And, Total Waiting Time = 48 ms
So, Average Waiting Time = 12.00 ms

4. Methodology

4.1 Common terms for users:

Arrival Time: Time at which the process arrives in the ready queue.
Completion Time: Time at which process completes its execution.
Burst Time: Time required by a process for CPU execution.
Turn Around Time: Time Difference between completion time and arrival time.

Turn Around Time = Completion Time - Arrival Time

Waiting Time (W.T): Time Difference between turnaround time and burst time.
Waiting Time = Turn Around Time - Burst Time

4.2 Formulas Used:

Turn Around Time (TAT)
= (Completion Time) - (Arrival Time)

Also, Waiting Time (WT)
= (Turn Around Time) - (Burst Time)

5.SNAPSHOTS OF LRTF WORKING CODE:

```c
//project done by RK17DMB56
//OS PROJECT
#include <stdio.h>
struct student
{
    int StudentId;
    int FoodTakenTime;
    int WaitingTime;
    int TurnAroundTime;
};

void accept(struct student list[], int s);
void display(struct student list[], int s);
void scheduling(struct student list[], int s);
void waitingTime(struct student list[], int n);
void turnAroundTime(struct student list[], int n);

int main()
{
    struct student data[20];
    int n,i;
    char c='n';
    do
    {
    printf("Please enter the No. of Students wants to eat in mess? : ");
    scanf("%d", &n);
    accept(data, n);
    scheduling(data, n);
    waitingTime(data,n);
    turnAroundTime(data,n);
    display(data, n);
    printf("Want to continue? press 'y' : ");
    scanf("%s",&c);
    }while(c=='y');
```

```
Please enter the No. of Students wants to eat in mess? : 3


Enter data for Student #1
Enter Student id : 2132
Enter time taken for food (minuts): 2


Enter data for Student #2
Enter Student id : 2102
Enter time taken for food (minuts): 4


Enter data for Student #3
Enter Student id : 2453
Enter time taken for food (minuts): 8


                        Output according to LRTF

|============================================================|
|Student id     FoodTakenTime   WaitingTime     TurnAroundTime   |
|============================================================|
|2453           8               0               8                |
|------------------------------------------------------------|
|2102           4               8               12               |
|------------------------------------------------------------|
|2132           2               12              14               |
|------------------------------------------------------------|

                        Total Waiting Time is: = 20
                        Total Turn around Time is: = 34


                        Average Waiting Time is: = 6
                        Average Turn around Time is: = 11

Want to continue? press 'y' :
```