**NAME:** YASH UNIYAL

**REGISTERATION NUMBER:** 11602157

**EMAIL:** [YASH381998@GMAIL.COM](mailto:YASH381998@GMAIL.COM)

**GIT HUB LINK:**

**QUESTION CODE:** 15

**PROBLEM ASSIGNED:**

A uniprocessor system has n number of CPU intensive processes, each process has its own requirement of CPU burst. The process with lowest CPU burst is given the highest priority. A late-arriving higher priority process can preempt a currently running process with lower priority. Simulate a scheduler that is scheduling the processes in such a way that higher priority process is never starved due to the execution of lower priority process. What should be its average waiting time and average turnaround time if no two processes are arriving are arriving at same time.

In this question, I have to make a program in which user will be asked to enter the number of processes, their arrival time and burst time. Now, according to the burst times to the processes they will be given the priority. The process with least arrival time will run first and after 1 sec. it will check for the new processes arriving at that time, if a new process arrive at any time, both processes’ burst time will be compared. If the burst time of the later arriving process is less than the currently running process then, later arriving process will preempt the currently running process and will execute first and the first process will wait. This process will continue till all the process are arrived and executed. Processes with the least execution time will be selected first. CPU is assigned to process with less CPU burst time.

For calculating average waiting time and average turnaround time, first I have to calculate waiting and turnaround times of all the processes and then adding them and dividing the sum with the number of total number of processes.

**Waiting Time** - It is the time a process spends in ready queue.

**Waiting time = Turnaround time – Burst time**

**Turnaround Time –** The time interval from submitting a process and its time of completion.

**Turnaround time = Completion time – Arrival Time**

**ALGORITHM USED:**

*SHORTESH JOB FIRST (PREMPTIVE):*

* Processes with least execution time are selected first.
* CPU is assigned to process with less CPU burst time.
* Pre-Emption: When a new process enters the queue, scheduler checks its execution time and compare with the already running process.
* If Execution time of running process is more, CPU is taken from it and given to new process.

**CODE:**

#include<stdio.h>

int main()

{

int at[10], bt[10], temp[10];

int i=0, min;

       int counter = 0, time, n,ct;

float wt = 0, tat = 0;

float average\_wt, average\_tat;

printf("\nEnter the Total Number of Processes:\t");

scanf("%d", &n);

while (i < n)

{

            printf("Process P%d\n",i);

printf("Enter Arrival Time:\t");

scanf("%d", &at[i]);

printf("Enter Burst Time:\t");

scanf("%d", &bt[i]);

temp[i] = bt[i];

i++;

}

bt[9] = 9999;

printf("\nSequence in which processes will execute :\n");

for(time = 0; counter != n; time++)

{

min = 9;

for(i = 0; i < n; i++)

{

if(at[i] <= time && bt[i] < bt[min] && bt[i] > 0)

{

min = i;

}

}

printf("P%d , ",min);

bt[min]--;

if(bt[min] == 0)

{

counter++;

ct = time + 1;

wt = wt + ct - at[min] - temp[min];

tat = tat + ct - at[min];

}

}

printf("end\n");

average\_wt = wt / n;

average\_tat = tat / n;

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\nAverage Waiting Time:%f\n", average\_wt);

printf("Average Turnaround Time:%f\n", average\_tat);

return 0;

}

**BOUNDARY CONDITIONS:**

Number of processes cannot be negative.

Burst time and arrival time cannot be negative.

Burst time and arrival time only accepts integer values.

**TEST CASES:**

1.

Number of processes entered are 3.

Process Arrival Time Burst Time

P0 3 5

P1 0 4

P2 2 1

Average Waiting time – 1.000

Average Turnaround Time – 4.333

2.

Number of processes: 3

Process Arrival Time Burst Time

P0 0 5

P1 0 3

P2 3 5

Average Waiting time – 3.000

Average Turnaround Time – 7.333

**GITHUB LINK:**