**Experiment No: 5**

|  |
| --- |
| **Student Name and Roll Number:** Namit Kumar |
| **Semester /Section:** V/FS-A1 |
| **Link to Code:** |
| **Date:** 31st August 2021 |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**  Write a program to implement CPU scheduling for first come first serve approach. |
| **Outcome:**  The students will understand the First-cum-first-serve algorithm |
| **Problem Statement:**  Implement the following CPU scheduling Algorithms.   1. FCFS with Arrival time 2. FCFS without Arrival time |
| **Background Study:**  **FCFS**   * The simplest CPU-scheduling algorithm is the first-come, first-served (FCFS) scheduling algorithm. With this algorithm, processes are assigned the CPU in the order they request it. * There is a single queue of ready processes. * The implementation of the FCFS policy is easily managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. * The average waiting time under the FCFS policy, however, is often quite long. |
| **Question Bank:**   1. Which module gives control of the CPU to the process selected by the short-term scheduler?    1. **dispatche**r    2. interrupt    3. scheduler    4. none of the mentioned 2. The processes that are residing in main memory and are ready and waiting to execute are kept on a list called    1. job queue    2. **ready queue**    3. execution queue    4. process queue 3. The interval from the time of submission of a process to the time of completion is termed as    1. waiting time    2. **turnaround time**    3. response time    4. throughput 4. Which scheduling algorithm allocates the CPU first to the process that requests the CPU first?    1. **first-come, first-served scheduling**    2. shortest job scheduling    3. priority scheduling    4. none of the mentioned 5. In priority scheduling algorithm    1. **CPU is allocated to the process with highest priority**    2. CPU is allocated to the process with lowest priority    3. equal priority processes cannot be scheduled    4. none of the mentioned |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

#include<stdio.h>

int main()

{

int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;

printf("Enter total number of processes(maximum 20):");

scanf("%d",&n);

printf("\nEnter Process Burst Time\n");

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

{

printf("P[%d]:",i+1);

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

}

wt[0]=0;

for(i=1;i<n;i++)

{

wt[i]=0;

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

wt[i]+=bt[j];

}

printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time");

//calculating turnaround time

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

{

tat[i]=bt[i]+wt[i];

avwt+=wt[i];

avtat+=tat[i];

printf("\nP[%d]\t\t%d\t\t%d\t\t%d",i+1,bt[i],wt[i],tat[i]);

}

avwt/=i;

avtat/=i;

printf("\n\nAverage Waiting Time:%d",avwt);

printf("\nAverage Turnaround Time:%d",avtat);

return 0;

}

A screenshot of a computer

Description automatically generated with medium confidence