Mawlana Bhashani Science and Technology University

Lab-Report

Report No:07

Report Name:Implementation of FCFS Scheduling algorithm

Course code:ICT-3110

Course title:Operating System Lab

Date of Performance:16-09-2020

Date of Submission:19-09-2020

Submitted by

Name:Sabikun Nahar Piya

ID:IT-18020

3rd year 1st semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No:07

Experiment Name:Implementation of FCFS scheduling algorithm

Theory:

The FCFS is a non-preemptive scheduling algorithm in which a process is automatically queued and processing occurs according to an incoming request or process order implementation of the FCFS policy is managed with a FIFO queue.When a process enters the ready queue,its PCB is linked on the tail of the queue.When the CPU is free,it is allocated to the process at the head of the queue.

Implementation:

Step 1: Input the processes along with their burst time(bt).

Step 2: Find waiting time (wt) for all processes.

Step 3: At first process that comes need not to wait so waiting time for process 1 will be 0 i.e.wt[0]=0.

Step 4: Find waiting time for all other processes i.e. for process I will be wt[i]= wt[i-1]+bt[i-1].

Step 5: Find turnaround time= waiting time + burst time for all processes.

Step 6: Find average waiting time = total waiting time/ no of process.

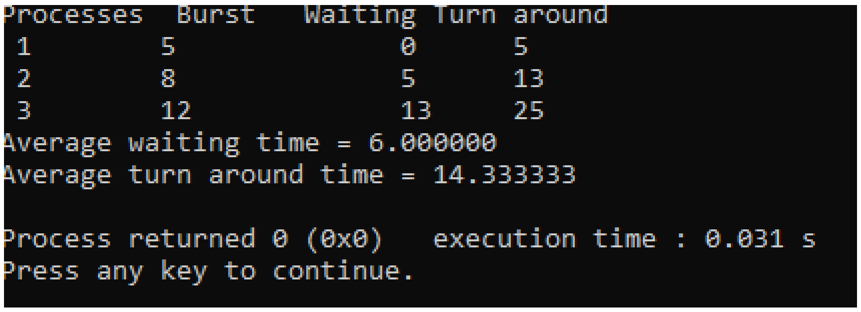
Step 7: Similarly, find average turnaround time = total turnaround time/no of process.

Working process:

Code for FCFS scheduling algorithm:

```
#include <stdio.h>
int waitingtime(int proc[], int n,
int burst_time[], int wait_time[]){
   wait_time[0] = 0;
   for (int i = 1; i < n ; i++ )
   wait_time[i] = burst_time[i-1] + wait_time[i-1] ;
   return 0;
}
int turnaroundtime( int proc[], int n,
int burst_time[], int wait_time[], int tat[]) {
   int i;
   for ( i = 0; i < n ; i++)
   tat[i] = burst_time[i] + wait_time[i];
   return 0;
}
int avgtime( int proc[], int n, int burst_time[]) {
   int wait_time[n], tat[n], total_wt = 0, total_tat = 0;
   int i;
   waitingtime(proc, n, burst_time, wait_time);
   turnaroundtime(proc, n, burst_time, wait_time, tat);
   printf("Processes  Burst   Waiting Turn around \n");
   for ( i=0; i<n; i++) {
      total_wt = total_wt + wait_time[i];
      total_tat = total_tat + tat[i];
      printf(" %d\t  %d\t\t %d \t%d\n", i+1, burst_time[i], wait_time[i],
tat[i]);
   }
   printf("Average waiting time = %f\n", (float)total_wt / (float)n);
   printf("Average turn around time = %f\n", (float)total_tat / (float)n);
   return 0;
}
int main() {
   int proc[] = { 1, 2, 3};
   int n = sizeof proc / sizeof proc[0];
   int burst_time[] = {5, 8, 12};
   avgtime(proc, n, burst_time);
   return 0;
}
```

Output:



```
Processes  Burst   Waiting Turn around
 1        5               0       5
 2        8               5       13
 3        12              13      25
Average waiting time = 6.000000
Average turn around time = 14.333333

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

Discussion:

FCFS is the best approach to minimize waiting time.It is easy to implement in batch systems where required CPU time is known in advance.It is impossible to implement in interacting systems where required cpu time is not known.The processor should know in advance how much time process will take.