Practical Name : FCFS Algorithm Implementation in C Practical No. 2

AIM: To write a C program to implement the FCFS scheduling algorithm.

Theory : Given n processes with their burst times, the task is to find average waiting time and average turnaround time using FCFS scheduling algorithm.

First in, first out (FIFO), also known as first come, first served (FCFS), is the simplest scheduling algorithm. FIFO simply queues processes in the order that they arrive in the ready queue.

In this, the process that comes first will be executed first and the next process starts only after the previous gets fully executed.

Here we are considering that arrival time for all processes is 0.

What is Waiting Time and Turnaround Time?

1. Turnaround Time is the time interval between the submission of a process and its completion.

Turnaround Time = completion of a process – submission of a process

2. Waiting Time is the difference between turnaround time and burst time

Waiting Time = turnaround time – burst time

we have assumed arrival times as 0, so turn around and completion times are same

FCFS (Example)

Process	Duration	Oder	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

Gantt Chart :

P1 waiting time : 0
P2 waiting time : 24
P3 waiting time : 27

The Average waiting time :

(0+24+27)/3 = 17

Code:-

```
#include<iostream>
using namespace std;
int main(){
        int i,n;
        cout<<"Enter total Number of Processes: ";
        cin>>n;
        int bt[n],wt[n],ct[n];
        for(i=0;i<n;i++){
        cout<<"Enter Burst Time for Process P"<<i<<": ";
        cin>>bt[i]; }
        wt[0]=0;
        for(i=1;i<=n;i++){
        wt[i]=wt[i-1]+bt[i-1]; }
        for(i=0;i<n;i++){
        ct[i]=wt[i]+bt[i];
        }
        cout<<"Process\t Burst Time \t Waiting Time \t Completion Time \n";
        for(i=0;i<n;i++){
        cout<<"P"<<i<<"\t";
        cout<<bt[i]<<"\t\t"<<wt[i]<<"\t\t"<<ct[i]<<"\t";
        cout<<"\n"; }
}
```

Output:-

```
Enter total Number of Processes: 3
Enter Burst Time for Process P0: 23
Enter Burst Time for Process P1: 3
Enter Burst Time for Process P2: 4
Process  Burst Time      Waiting Time    Completion Time
P0      23              0               23
P1      3               23              26
P2      4               26              30
```

Code - (Different Arrival Time)

```
#include<iostream>
using namespace std;
int main(){
        int i,n,j;
        cout<<"Enter total Number of Processes: ";
        cin>>n;
        int bt[n],wt[n],ct[n],art[n];
        for(i=0;i<n;i++){
        cout<<"Enter Burst Time for Process P"<<i<<": ";
        cin>>bt[i];
        cout<<"Enter Arrival Time for Process P"<<i<<": ";
        cin>>art[i]; }
        wt[0]=0;
        for(i=1;i<=n;i++){
        int sum=0;
        for(j=0;j<i;j++){
        sum+=bt[j]; }
        wt[i]=sum-art[i]; }
        for(i=0;i<n;i++){
        ct[i]=wt[i]+bt[i]; }
        cout<<"Process\t Burst Time \t Arrival Time \t Waiting Time \t Execution Time \n";
        for(i=0;i<n;i++){
        cout<<"P"<<i<<"\t";
        cout<<bt[i]<<"\t\t"<<art[i]<<"\t\t"<<wt[i]<<"\t\t"<<ct[i]<<"\t";
        cout<<"\n"; } }
```

Output :-

```
Enter total Number of Processes: 3
Enter Burst Time for Process P0: 23
Enter Arrival Time for Process P0: 0
Enter Burst Time for Process P1: 3
Enter Arrival Time for Process P1: 2
Enter Burst Time for Process P2: 4
Enter Arrival Time for Process P2: 3
Process  Burst Time      Arrival Time    Waiting Time    Execution Time
P0      23              0               0               23
P1      3               2               21              24
P2      4               3               23              27
```

Aim - Study of hardware & Software requirements of various Operating Systems

Theory -

What is an Operating System?

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it.

Parameters	UNIX	LINUX	Windows XP	Windows 7	Windows 10
RAM	256MB, minimum.	1 GB or greater	64 MB of RAM	1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)	1 gigabyte (GB) for 32-bit or 2 GB for 64-bit.
Processor	HP 9000/800	64-bit Opteron, EM64T	233 MHz processor	1 GHz or faster 32-bit or 64-bit processor	2 gigahertz (GHz) or faster processor or SoC.
Disk Space	Minimum of 300MB	500 MB free space	1.5 gb of free hard drive space.	16 GB available hard disk space (32-bit) or 20 GB (64-bit)	16 GB for 32-bit OS or 20 GB for 64-bit OS.
Drivers/Hardware	ODBC Driver	None	SVGA-capable video card.	DirectX 9 graphics device with WDDM 1.0 or higher driver	DirectX 9 or later with WDDM 1.0 driver

Result - Studied and Understood Hardware and Software requirements of various operating systems.