بسم الله الرحمن الرحيم

FCFS Scheduling

Lab 08

fcfs.cpp

اللهم علمنا ما ينفعنا،،، وانفعنا بما علمتنا،،، وزدنا علماً

Lab Objective

- To practice the FCFS scheduling.

Quick Refresh

- Turnaround time:
  - the time of submission to the time of completion.
- Waiting time:☆
  - amount of time a process has been waiting in the ready queue.
- Response time:
  - amount of time it takes from when a request was submitted until the first response is produces.

FCFS Scheduling

- Assigns the CPU based on the order of requests
  - *Nonpreemptive*: A process keeps running on the CPU until it's blocked or terminated.

ايجابيات + Simple

سلبيات - Short jobs can get stuck behind long jobs (convoy effect)

Procedure

- Write a C++ program that simulate the FCFS CPU scheduling policy.
- Assume that you have only three processes.
- The inputs to the program are the arrival time and burst time of each process.
- The output of the program are the response time, waiting time, and turnaround time for each of the three process.

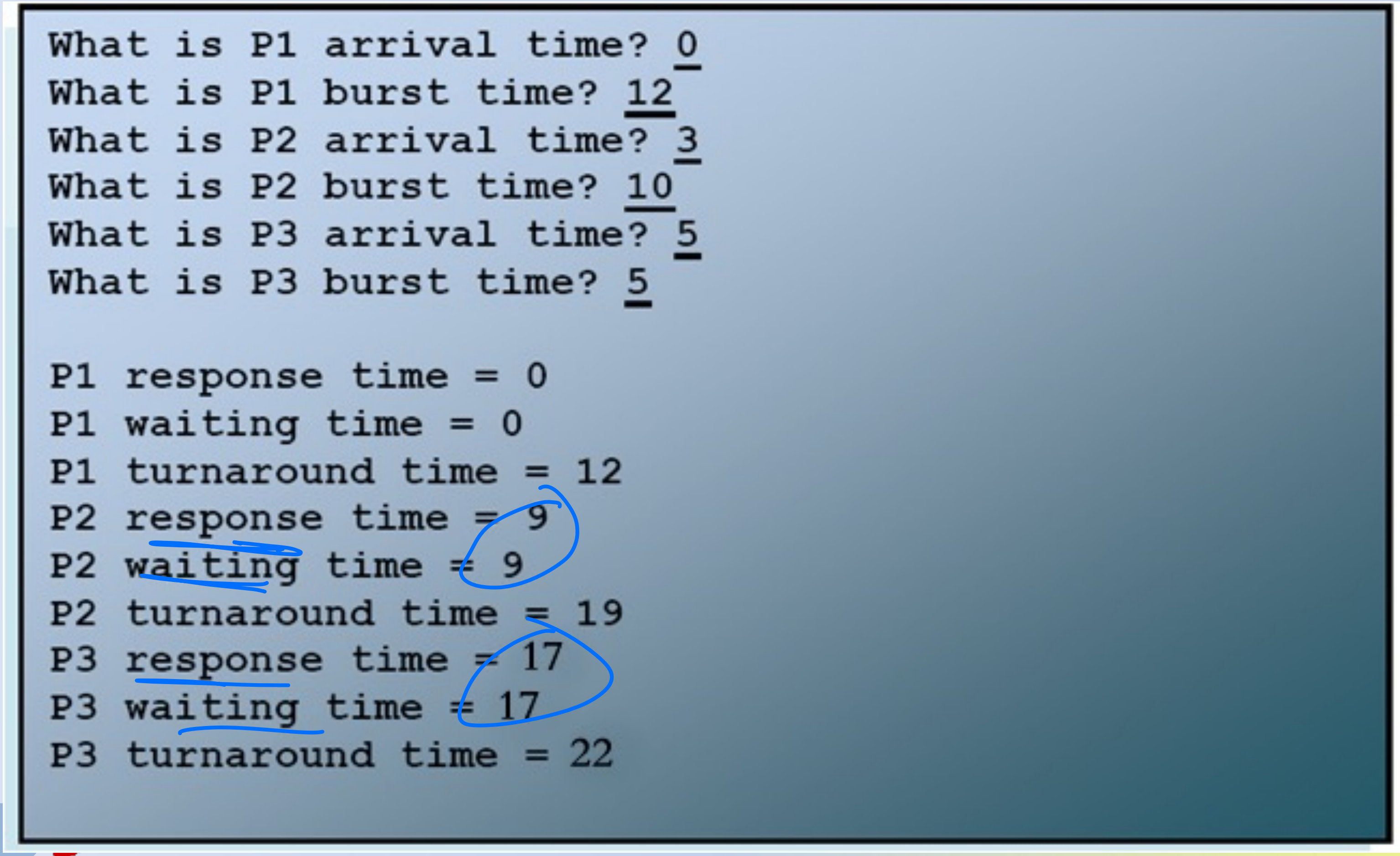
- Extra: Calculate the average waiting time

Steps

1. Get values from the user. Arrive time
   Burst time
2. Sort the processes based on the arrival time.
3. Calculate the start and end time for each process.
4. Calculate response, waiting, turnaround times for each process.
5. Display the results. Responee = waiting

Procedure (Cont.)

- The following is a sample run of the program (the underlined numbers are entered by the user who runs the program):



```
What is P1 arrival time? 0
What is P1 burst time? 12
What is P2 arrival time? 3
What is P2 burst time? 10
What is P3 arrival time? 5
What is P3 burst time? 5

P1 response time = 0
P1 waiting time = 0
P1 turnaround time = 12
P2 response time = 9
P2 waiting time = 9
P2 turnaround time = 19
P3 response time = 17
P3 waiting time = 17
P3 turnaround time = 22
```

```
#include <iostream>
using namespace std;
int main()
{
   float n,tempb,tempa,tempp,tw,average,gap,arrive[3],burst[3],
process[3],start[3],finish[3],waiting[3],response[3],
turnaround[3];
```

رقم البروسس

```
   int i,j;
```

loop counters

```
////////////// Get values from User///////////
   for(i=0;i<3;i++)
   { n=i+1;
```

خطوه اضافيه عشان أول بروسس تكون 1 مو 0

```
     process[i]=n;
     cout<<"what is p"<<n<<" arrival time\t";
     cin>>arrive[i];
     cout<<" what is p"<<n<<" burst time\t";
     cin>>burst[i];
   }//end for
```

```
//////////Sort process based on arrival time//////////////

  for(i=0;i<2;i++)
     for( j=i+1;j<3;j++)
     {
       if(arrive[j]<arrive[i])
       {
          tempa=arrive[i];
          arrive[i]=arrive[j];
          arrive[j]=tempa;
          tempb=burst[i];
          burst[i]=burst[j];
          burst[j]=tempb;
          tempp=process[i];
          process[i]=process[j];
          process[j]=tempp;
       }//end if
      }//end for
```

arrive

burst

Process

الي
بتكون أقل

```
///////////////calculate start and finish time ////////////////
  start[0]=arrive[0];
  finish[0]=arrive[0]+burst[0];
  for(i=1;i<3;i++)
  {
    gap=0;
    if(arrive[i]>finish[i-1])
    {
      gap=arrive[i]-finish[i-1];
      start[i]=finish[i-1]+gap;
    }//end if
    else
      start[i]=finish[i-1];

    finish[i]=start[i]+burst[i];
  }//end for
```

حق البروسس السابقه

→ Another Scenario

→ Scenario الثاني

```
///calculate response, waiting, turnaround times for each process///
    tw=0;  total waiting
    for(i=0;i<3;i++)
    { response[i]=......................;start[i]-arrive[i];
      waiting[i]=.........//.........;   -  //
      turnaround[i]=....finish[i]..........; -arrive[i];
      tw+=waiting[i];
    }//end for

    average=tw/3............ . .;

//////////////////Display results////////////////////
    for(i=0;i<3;i++)
    {
      cout<<"process Number"<<process[i]<<'\n'<<"arrive at 
"<<arrive[i]<<'\n'<<"waiting Time = "<<waiting[i]<<'\n'<<"response 
Time= "<<response[i]<<'\n'<<"Turnaround Time = 
"<<turnaround[i]<<'\n’;
    }
    cout<<"Total waiting time = "<<tw;
    cout<<"\n \n Average waiting time = "<<average;
    cout<<"\n\n\t\t\t--------*******FCFS ********----------\n";
    return(0);
}//end main
```

Question ?