|  |  |
| --- | --- |
|  | **DEPARTMENT OF COMPUTER ENGINEERING** |

**Experiment No. 2**

|  |  |
| --- | --- |
| Semester | S.E-Semester IV – Computer Engineering |
| Subject | Operating System |
| Subject Professor In-charge | Prof. Pankaj Vanwari |
| Assisting Teachers | Prof. Pankaj Vanwari |

|  |
| --- |
| Student Name – Deep Salunkhe |
| Roll Number – 21102A0014 |
| Division and Batch – Division A, Batch 1 |
| Date of Implementation – |
| Experiment Title: FCFS |
| **Theory:** |
|  |

|  |
| --- |
| #include<iostream>  #include<list>  #include<vector>  #include<queue>  #include<algorithm>  using namespace std;  struct ProcessControlBlock  {      int pid;      int BT;      int AT;      int CT;      int waitingtime;      int TAT;    };    vector<ProcessControlBlock>PCB;  vector<ProcessControlBlock>Arrived;  queue<ProcessControlBlock>running;  vector<ProcessControlBlock>ans;  int sortByarrival(ProcessControlBlock a , ProcessControlBlock b)  {      if(a.AT < b.AT)          return 1;      else          return 0;  }  int sortByburst(ProcessControlBlock a , ProcessControlBlock b)  {      if(a.BT < b.BT)          return 1;      else          return 0;  }    int main()  {      int clock=0,pcbindex=0;        int noOfProcess;      cout<<"ENTER THE NO. PROCESS =>"<<"\t";      cin>>noOfProcess;      cout<<endl;      int n=noOfProcess;        for(int i=0;i<n;i++)      {          ProcessControlBlock dummy;          cout<<"Enter the pid          "<<i+1<<"\t";          cin>>dummy.pid;          cout<<"Enter the Burst Time   "<<i+1<<"\t";          cin>>dummy.BT;          cout<<"Enter the Arrival Time "<<i+1<<"\t";          cin>>dummy.AT;          cout<<endl;          PCB.push\_back(dummy);      }      *//----------------------------------------Logic for SJF----------------------------------------------------------//*      sort(PCB.begin(),PCB.end(),sortByarrival);      int envruntime;      int inpcb=0;      int time=0;      int currBT=0;      int index=0;      cout<<"For how much time the env must be active";      cin>>envruntime;      while(envruntime--)      {          while(time==PCB[inpcb].AT)          {              Arrived.push\_back(PCB[inpcb]);              inpcb++;              sort(Arrived.begin(),Arrived.end(),sortByburst);          }          if(running.empty())          {              running.push(Arrived[0]);              currBT=running.front().BT;                Arrived.erase(Arrived.begin()+0);            }          if(currBT==0)          {              ans.push\_back(running.front());                ans[index].waitingtime=time-ans[index].BT-ans[index].AT;              ans[index].CT=time;              ans[index].TAT=time-ans[index].AT;              cout<<time<<endl;              index=index+1;              running.pop();              time--;          }            time++;              currBT--;      }         cout<<"\t\tSJF"<<endl;      cout<<"PID"<<"\t"<<"BT"<<"\t"<<"AT"<<"\t"<<"WT"<<endl;        for(int i=0;i<n;i++)      {             cout<<ans[i].pid<<" \t"<<ans[i].BT<<"  \t"<<ans[i].AT<<"  \t"<<ans[i].waitingtime<<"\t"<<endl;        }                } |
| Output: |