AMRITSAR GROUP OF COLLEGES

Autonomous status conferred by UGC under UGC act-1956, (2f), NAAC-A Grade,
(Formerly Known as Amritsar College of Engineering & Technology| Amritsar Pharmacy College)

PROJECT REPORT

On

"To simulate the Priority CPU Scheduling algorithms"

Submitted in the Partial fulfilment of the requirement for the Award of Degree of

Bachelor of Technology

in

COMPUTER SCIENCE & ENGINEERING Batch (2020-2024)

Submitted by:

Sanjay Kumar Sah (2000193)
Shahil Kumar (2000196)
Siddharth Kumar Sonu (2000206)
Sumit Kumar Giri (2000213)

Submitted to:

Er. Bhuvnesh Kumar

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Amritsar Group of Colleges, Amritsar

ACKNOWLEDGEMENT

This is a humble effort to express our sincere gratitude towards those who have guided and helped us to complete this project.

A project is major milestone during the study period of a student. As such this project was a challenge to us and was an opportunity to prove our caliber. We are highly grateful and obliged to each and every one making us help out of problems being faced by team.

It would not have been possible to see through the undertaken project without the guidance of **Er**. **Bhuvnesh Kumar (Assistant Professor**) It was purely on the basis of their experience and knowledge that we are able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least we are very thankful to our Head of Department and all Members of Computer Science Department who gave us an opportunity to face real time problems while fulfilling need of an organization by making projects for them.

DECLARATION

We hereby declare that the project work entitled **"To simulate the Priority CPU scheduling algorithms"** has been completely prepared by our team as a part of my core subject during this semester. This report is the outcome of our efforts and has been submitter in the department of computer science and engineering. The contents of this report are fully verified as per our knowledge.

(Signature of students)

Sanjay Kumar Sah (2000193)

Shahil Kumar (2000196)

Siddharth Kumar Sonu(2000206)

Sumit Kumar Giri (2000213)

Certified that the above statement made by the students is correct to the best of our knowledge and belief.

Faculty Coordinator

Er. Bhuvnesh Kumar(Assistant Professor)

INDEX PAGE

Introduction to Priority CPU Scheduling Algorithm

Priority scheduling is one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with the highest priority is to be executed first and so on.

Processes with the same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Due to its simplistic nature, an Priority algorithm is predictable, regardless of the type of tasks or requests it has to process. Like a grocery store checkout scheme, Priority algorithms mimic real-life customer service situations where patrons who arrive first get served first regardless of the size and complexity of their interaction.

Priority is one of the most efficient and autonomous types of scheduling algorithm because it requires little-to-no human or **Artificial Intelligence** (AI) intervention and does not waste time prioritizing tasks and requests by their urgency or level of complexity. Additionally, the party responsible for the scheduling is the CPU itself instead of software or an alternate, more complex job scheduling algorithm.

Use of the **Priority** algorithm risks the possibility that a series of simple requests will get stuck in a central processing units queue for an unreasonably long wait time behind a single complex task, just because the complex task arrived first

How a Priority Scheduling Algorithm works

Here is how an Priority scheduling algorithm works. To begin, suppose there are three requests to process in the CPU's queue: P1, P2, and P3. Assume P1 is a complex process that requires approximately 25 seconds, P2 a much simpler request that requires only 10 seconds of processing, and P3 a moderately simple request that requires 15 seconds.

When P1 is first put in the queue, the wait time is zero and the CPU starts the processing immediately. P2, on the other hand, would have a wait time of 25 seconds. And P3, having arrived last, would have to wait 35 seconds. As a total, an Priority scheduling algorithm would need 50 seconds to complete all three requests and empty the queue, which would be the same as other sequential processing, mono CPU systems.

Because Priority does not evaluate requests before starting, it has fewer complete tasks per set period of time when compared to an intelligent scheduling algorithm. In this scenario, the Priority scheduling algorithm would complete a single task in the first half of its run time of 25 seconds. Other algorithms—that start from the simplest of requests, for example—would have finished two requests.

Process	Arrival Time	Burst Time	Priority
P1	0	11	2
P2	5	28	0
P3	12	2	3
P4	2	10	1
P5	9	16	4

Characteristics of Priority method

1. Its supports non-preemptive and preemptive scheduling algorithm.
2. Jobs are always executed on a **Priority** basis.
3. It is easy to implement and use
4. This method is poor in performance, and the general wait time is quite high.

Advantages of Priority

Here, are pros/benefits of using Priority CPU scheduling algorithm:

1. The simplest form of a CPU scheduling algorithm
2. Easy to program
3. First come First served

Disadvantages of Priority

Here, are cons/drawback of using Priority scheduling algorithm:

1. It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing.
2. The Average Waiting Time is high.
3. Short processes that are at the back of the queue have to wait for the long process at the front to finish.
4. Not an ideal technique for time-sharing systems.
5. Because of its simplicity, Priority is not very efficient

Example of Priority Scheduling Algorithm

Let's take an example of The Priority scheduling algorithm. In the Following schedule, there are 5 processes with process ID **P1, P2, P3, P4 and P5**. P1 arrives at time 1, P2 at time 2, P3 at time 3, P3 at time 3,P4 arrives at time 4, and Process P5 arrives at time 5 in the ready queue. The processes and their respective Arrival and Burst time are given in the following table.

The Turnaround time and the waiting time are calculated by using the following formula.

1. Turn Around Time = Completion Time - Arrival Time
2. Waiting Time = Turnaround time - Burst Time

The average waiting Time is determined by summing the respective waiting time of all the processes and divided the sum by the total number of processes.

Project code

```
#include<iostream>

using namespace std;

int main()

{

   int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;

   cout<<"Enter Total Number of Process:";

   cin>>n;
```

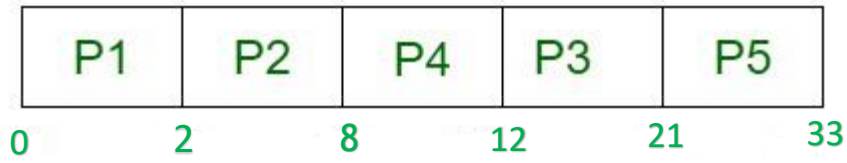
```
cout<<"\nEnter Burst Time and Priority\n";
for(i=0;i<n;i++)
{
   cout<<"\nP["<<i+1<<"]\n";
   cout<<"Burst Time:";
   cin>>bt[i];
   cout<<"Priority:";
   cin>>pr[i];
   p[i]=i+1;          //contains process number
}

//sorting burst time, priority and process number in ascending order using selection sort
for(i=0;i<n;i++)
{
   pos=i;
   for(j=i+1;j<n;j++)
   {
      if(pr[j]<pr[pos])
         pos=j;
   }
```

```
    temp=pr[i];
    pr[i]=pr[pos];
    pr[pos]=temp;

    temp=bt[i];
    bt[i]=bt[pos];
    bt[pos]=temp;

    temp=p[i];
    p[i]=p[pos];
    p[pos]=temp;
}

wt[0]=0;            //waiting time for first process is zero

//calculate waiting time
for(i=1;i<n;i++)
{
    wt[i]=0;
    for(j=0;j<i;j++)
        wt[i]+=bt[j];
```

```
    total+=wt[i];
  }

  avg_wt=total/n;      //average waiting time
  total=0;

  cout<<"\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time";
  for(i=0;i<n;i++)
  {
    tat[i]=bt[i]+wt[i];     //calculate turnaround time
    total+=tat[i];
    cout<<"\nP["<<p[i]<<"]\t\t  "<<bt[i]<<"\t\t    "<<wt[i]<<"\t\t\t"<<tat[i];
  }

  avg_tat=total/n;     //average turnaround time
  cout<<"\n\nAverage Waiting Time="<<avg_wt;
  cout<<"\nAverage Turnaround Time="<<avg_tat;

  return 0;
}
```

```
Enter Total Number of Process:5

Enter Burst Time and Priority

P[1]
Burst Time:2
Priority:0

P[2]
Burst Time:6
Priority:1

P[3]
Burst Time:4
Priority:2

P[4]
Burst Time:9
Priority:3

P[5]
Burst Time:12
Priority:6

Process     Burst Time          Waiting Time    Turnaround Time
P[1]             2                   0                  2
P[2]             6                   2                  8
P[3]             4                   8                  12
P[4]             9                   12                 21
P[5]             12                  21                 33

Average Waiting Time=8
Average Turnaround Time=15
```

GANTT CHART PREPARATION



TASK ALLOCATION

Sanjay Kumar Sah (2000193) : Testing connection between VS code

Shahil Kumar (2000196): coding C ++

Siddharth Kumar Sonu (2000206) : Providing security aids to project

Sumit Kumar Giri (2000213): Testing and fixing errors in whole project

We are working together for this project to do our best.We use to simulate the Priority CPU Scheduling algorithms