LOTTERY SCHEDULING SYSTEM

A
Mini Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

By

SVN. RAMAKANTH

1602-19-733-118

Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2021

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Computer Science & Engineering

DECLARATION BY THE CANDIDATE

I, **SVN. RAMAKANTH,** bearing hall ticket number, **1602-19-733-118**, hereby declare that the project report entitled **LOTTERY SCHEDULING SYSTEM,** Department of Computer Science & Engineering, VCE, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering**.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

SVN. RAMAKANTH,

1602-19-733-118.

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Computer Science & Engineering

BONAFIDE CERTIFICATE

This is to certify that the project entitled **LOTTERY SCHEDULING SYSTEM** being submitted by **SVN. RAMAKANTH,** bearing **1602-19-733-118,** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering is a record of bonafide work carried out by him/her under my guidance.

Dr. K. Srinivas,

Assoc Professor

Dept. of CSE,

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them. We would like to express my gratitude towards K. Srinivas sir for his kind co-operation and encouragement which help me in completion of this project. Our thanks and appreciations also go to my classmate in developing the project and people who have willingly helped me out with their abilities.

TABLE OF CONTENTS

ABSTRACT

This project is about drawing lottery tickets and scheduling them. Lottery scheduling is a probabilistic scheduling algorithm for processes in an operating system. We consider priority and burst times of each process and allocate tickets to them as required. Processes are each assigned some number of lottery tickets, and the scheduler (CPU) draws a random ticket to select a process and this random withdrawal continues till all the processes burst time is exhausted. This project can be implemented in areas where we rely on randomization and probabilistic scheduling where the output can't be predicted but can be estimated based on the input parameters.

INTRODUCTION

This code helps the user to schedule processes and choose a process randomly by implementing lottery system. This can be used at every lottery system implementation. Input parameters considered here are priority and burst time of that process.

DESCRIPTION:

Problem: In the problem we have to assign one or more lottery tickets to each of the process. A lottery ticket is chosen at random, and the process holding that ticket gets the CPU. The operating system implements lottery scheduling by holding a lottery 50 times each second, with each lottery winner getting 20 milliseconds of CPU time (20 milliseconds × 50 = 1 second). Hence the quantum time is 1sec.

Solution: According to the problem, first we allocate Burst time and Priority for each process. One or more lottery ticket is assigned to each of the process. Assign more tickets to the higher priority process. The probability of completion of process with more number of tickets increases. Random ticket is generated and process having the ticket gets the CPU for the specified quantum time. After the quantum time, the running process is pre-empted and another random ticket is generated. Now the process with the ticket gets the CPU. This process will continue until all the processes are completed.

Main Concept: Ticket allocation takes place considering 2 major factors, burst time and allocated priority. Relatively more tickets will be allocated to higher priority process so that ultimately the process with more priority has more chances of being chosen in lottery

SYSTEM DESIGN

This project involves use of a processor which sits at the core of a computer. To get to the core, a programming language must be low-level, which means it must provide little or no abstraction of programming concepts and be very close to writing actual machine instructions. My language of choice was C as it is a low-level language and supports multi-threaded programming with POSIX threads library.

Approach:

According to the problem, first we allocate Burst time and Priority for each process. One or more lottery ticket is assigned to each of the process. Assign more tickets to the higher priority process. Random ticket is generated and process having the ticket gets the CPU for the specified quantum time. After the quantum time, the running process is preempted and another random ticket is generated. Now the process with the ticket gets the CPU. This process will continue until all the processes are completed.

Objective:

In the problem we have to assign one or more lottery tickets to each of the process. A lottery ticket is chosen at random, and the process holding that ticket gets the CPU. The operating system implements lottery scheduling by taking quantum time as 1second.

Working:

Inputs Taken:

- No. of processes
- Priority and burst times of each process

Based on these inputs tickets will be allocated to each process and a ticket will be randomly chosen from all the tickets until burst times of all the tickets become zero.

Output shown:

- Currently running process
- Total time passed and remaining burst time of that process

If the ticket chosen is from the process which is completed, it'll show that the process is completed

IMPLEMENTATION:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
void main()
{
	int n,i,j,k,temp=65,flag=0;
	char process[20];
	int brust[20],priority[20],pos;
	int time=0,quantom=1,tbt=0;
	int z=0,lottery[20],ticket[20][20],q=0;
	printf("\t\t*Welcome to lottery scheduling system*\n\n");
	printf("Enter Number Of Process: ");
	scanf("%d",&n);
	if(n<=0)
	{
		printf("\n\n::::  Invalid Value Of Number Of Process  ::::");
		exit(0);
	}
	for(i=0;i<n;i++)
	{
		process[i] = temp;
		temp+=1;
	}
	for(i=0;i<n;i++)
```

```
        {
                printf("\nEnter The Brust Time For Process %c: ",process[i]);
                scanf("%d",&brust[i]);
                printf("Enter The Priority For Process %c(b/w 1 to %d): ",process[i],n);
                scanf("%d",&priority[i]);
        }

        //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(priority[j]<priority[pos])
            pos=j;
      }

      temp=process[i];
      process[i]=process[pos];
      process[pos]=temp;

      temp=brust[i];
      brust[i]=brust[pos];
      brust[pos]=temp;

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

```
        priority[pos]=temp;

        if(brust[i]<0)
                {
                        flag = 1;
                }
    }

    if(flag==1)
    {
        printf("\n\n:::  Invalid Time Entered ::: \n");
        exit(0);
        }

    printf("\n |Priority |  Process   |  Burst |");

        for(i=0;i<n;i++)
        {
                printf("\n |   %d \t   |    %c    \t|   %d
|",priority[i],process[i],brust[i]);
                tbt = tbt + brust[i];
        }

        printf("\n\n:::::::::   Quantom time is 1 Sec   ::::::::: \n\n");

        //assign one or more lottery numbers to each process
        int p=1,m_ticket=0;
        printf("\n\n\nPriority  process  Burst  Lottery  Tickets");
        for(i=0;i<n;i++)
```

```
        {
                lottery[i] = (brust[i]/quantom) + (n-priority[i]);
                for (z=0;z<lottery[i];z++)
                {
          ticket[i][z] = p++;
          m_ticket = p;
       }

                printf("\n  %d\t    %c\t    %d \t
%d\t",priority[i],process[i],brust[i],lottery[i]);
                for(z=0;z<lottery[i];z++)
                {
                        if(ticket[i][z]<10)
                                printf("  ::%d:: ",ticket[i][z]);
                        else
                                printf("  ::%d::",ticket[i][z]);
        }
        }

        while(time!=tbt)
        {

        int winner = (rand()%m_ticket-1)+ 1;
    for(i =0;i<n;i++)
       for(z=0;z<lottery[i];z++)
          if(ticket[i][z]==winner)
             q=i;

    printf("\n\n-----------------------------------");
```

```
   printf("<<<<   Winner: %d   >>>>",winner);
   printf("-----------------------------------\n");

        if ((brust[q]>0))
        {
      brust[q]-=quantom;

      if (brust[q]>0)
                {
         time+=quantom;
      }
                else
                {
         time+=(brust[q]+quantom);
      }

                if(brust[q]<0)
                {
            brust[q]=0;
      }
           printf("\n\t\t\t\t  Process That Are Running Is: %c",process[q]);
           printf("\n\t   (Total Time << Remaining Burst Time Of This Process << process ):  ( %d  <<  %d  <<  %c )\n",time,brust[q],process[q]);

        }
   else
   {
        printf("\n\t\t     >>>>>>Related Process With This Ticket Has Been Completed<<<<<<\n");}}}
```

OUTPUT SCREENSHOTS AND TEST CASES

Starting Interface:

Inputs such as number of processes, their burst times and priorities are taken.

```
                *Welcome to lottery scheduling system*

Enter Number Of Process: 4

Enter The Brust Time For Process A: 5
Enter The Priority For Process A(b/w 1 to 4): 3

Enter The Brust Time For Process B: 4
Enter The Priority For Process B(b/w 1 to 4): 1

Enter The Brust Time For Process C: 3
Enter The Priority For Process C(b/w 1 to 4): 4

Enter The Brust Time For Process D: 2
Enter The Priority For Process D(b/w 1 to 4): 2
```

Sorting processes as per priority:

```
 |Priority |  Process   |  Burst |
 |   1     |    B       |   4    |
 |   2     |    D       |   2    |
 |   3     |    A       |   5    |
 |   4     |    C       |   3    |

:::::::::   Quantom time is 1 Sec   :::::::::
```

Ticket allocation:

Tickets are allocated to each process considering their priority and burst times by the formula:

lottery[i] = (brust[i]/quantom) + (n-priority[i]);

Here,

i = a particular process instant

burst[i] = burst time of that process

quantum = quantum time of CPU (1sec)

n= number of processes

priority[i] = Priority of that particular process

```
Priority  process  Burst  Lottery  Tickets
  1         B       4       7     ::1::   ::2::   ::3::   ::4::   ::5::   ::6::   ::7::
  2         D       2       4     ::8::   ::9::   ::10::  ::11::
  3         A       5       6     ::12::  ::13::  ::14::  ::15::  ::16::  ::17::
  4         C       3       3     ::18::  ::19::  ::20::
```

Sample draw:

```
-----------------------------------<<<<   Winner: 15   >>>>-----------------------------------

                              Process That Are Running Is: A
          (Total Time << Remaining Burst Time Of This Process << process ):  ( 11  <<  3  <<  A )
```

All process execution till burst time exhaustion:

```
------------------------------------<<<<   Winner: 1   >>>>------------------------------------

                                 Process That Are Running Is: B
          (Total Time << Remaining Burst Time Of This Process << process ):  ( 1  <<  3  <<  B )


------------------------------------<<<<   Winner: 4   >>>>------------------------------------

                                 Process That Are Running Is: B
          (Total Time << Remaining Burst Time Of This Process << process ):  ( 2  <<  2  <<  B )


------------------------------------<<<<   Winner: 9   >>>>------------------------------------

                                 Process That Are Running Is: D
          (Total Time << Remaining Burst Time Of This Process << process ):  ( 3  <<  1  <<  D )
```

```
------------------------------------<<<<   Winner: 8   >>>>------------------------------------

                                 Process That Are Running Is: D
          (Total Time << Remaining Burst Time Of This Process << process ):  ( 5  <<  0  <<  D )


------------------------------------<<<<   Winner: 10   >>>>-----------------------------------

                    >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


------------------------------------<<<<   Winner: 10   >>>>-----------------------------------

                    >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


------------------------------------<<<<   Winner: 9   >>>>------------------------------------

                    >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


------------------------------------<<<<   Winner: 15   >>>>-----------------------------------

                                 Process That Are Running Is: A
          (Total Time << Remaining Burst Time Of This Process << process ):  ( 6  <<  4  <<  A )
```

```
----------------------------------<<<<   Winner: 9   >>>>----------------------------------

                >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


----------------------------------<<<<   Winner: 15   >>>>----------------------------------

                            Process That Are Running Is: A
       (Total Time << Remaining Burst Time Of This Process << process ):  ( 6  <<  4  <<  A )


----------------------------------<<<<   Winner: 10   >>>>----------------------------------

                >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


----------------------------------<<<<   Winner: 2   >>>>----------------------------------

                            Process That Are Running Is: B
       (Total Time << Remaining Burst Time Of This Process << process ):  ( 7  <<  1  <<  B )


----------------------------------<<<<   Winner: 19   >>>>----------------------------------

                            Process That Are Running Is: C
       (Total Time << Remaining Burst Time Of This Process << process ):  ( 8  <<  1  <<  C )


----------------------------------<<<<   Winner: 20   >>>>----------------------------------
```

```
-------------------------------------<<<<   Winner: 19   >>>>-------------------------------------

                                Process That Are Running Is: C
         (Total Time << Remaining Burst Time Of This Process << process ):  ( 8  <<  1  <<  C )


-------------------------------------<<<<   Winner: 20   >>>>-------------------------------------

                                Process That Are Running Is: C
         (Total Time << Remaining Burst Time Of This Process << process ):  ( 9  <<  0  <<  C )


-------------------------------------<<<<   Winner: 4   >>>>-------------------------------------

                                Process That Are Running Is: B
         (Total Time << Remaining Burst Time Of This Process << process ):  ( 10  <<  0  <<  B )


-------------------------------------<<<<   Winner: 20   >>>>-------------------------------------

                  >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


-------------------------------------<<<<   Winner: 7   >>>>-------------------------------------

                  >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


-------------------------------------<<<<   Winner: 3   >>>>-------------------------------------
```

```
-----------------------------------<<<<   Winner: 3   >>>>-------------------------------------

                >>>>>>Related Process With This Ticket Has Been Completed<<<<<<


-----------------------------------<<<<   Winner: 15   >>>>-------------------------------------

                              Process That Are Running Is: A
       (Total Time << Remaining Burst Time Of This Process << process ):  ( 11  <<  3  <<  A )


-----------------------------------<<<<   Winner: 16   >>>>-------------------------------------

                              Process That Are Running Is: A
       (Total Time << Remaining Burst Time Of This Process << process ):  ( 12  <<  2  <<  A )


-----------------------------------<<<<   Winner: 16   >>>>-------------------------------------

                              Process That Are Running Is: A
       (Total Time << Remaining Burst Time Of This Process << process ):  ( 13  <<  1  <<  A )


-----------------------------------<<<<   Winner: 17   >>>>-------------------------------------

                              Process That Are Running Is: A
       (Total Time << Remaining Burst Time Of This Process << process ):  ( 14  <<  0  <<  A )
```

Now burst time of all the N processes has reached zero

CONCLUSION:

This project is implemented by randomization and probabilistic scheduling where the output can't be predicted but can be estimated based on the input parameters, i.e, process priority and burst time. Hence lottery scheduling system is implemented, considering burst time and priority of each process.

REFERENCES

- https://linuxize.com
- www.tutorialspoint.com
- www.geeksforgeeks.com