A

Report On

**“Implementation of Round-Robin Scheduling Algorithm using c”**

**Under the subject**

**‘Operating System [22516]’**

**Submitted by**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Enrollment No** | **Name of the student** |
| 1 | 1700100271 | Miss. Shinde Mohini Anandrao |
| 2 | 1700100272 | Miss. Awate Amruta Narayan |
| 3 | 1700100370 | Mr. Patil Pratik Kumar |
| 4 | 1800100002 | Mr. Kare Abhilash Sudam |

**Under the guidance of**

Mrs. Amrutkar P. R.

Department of Computer Engineering

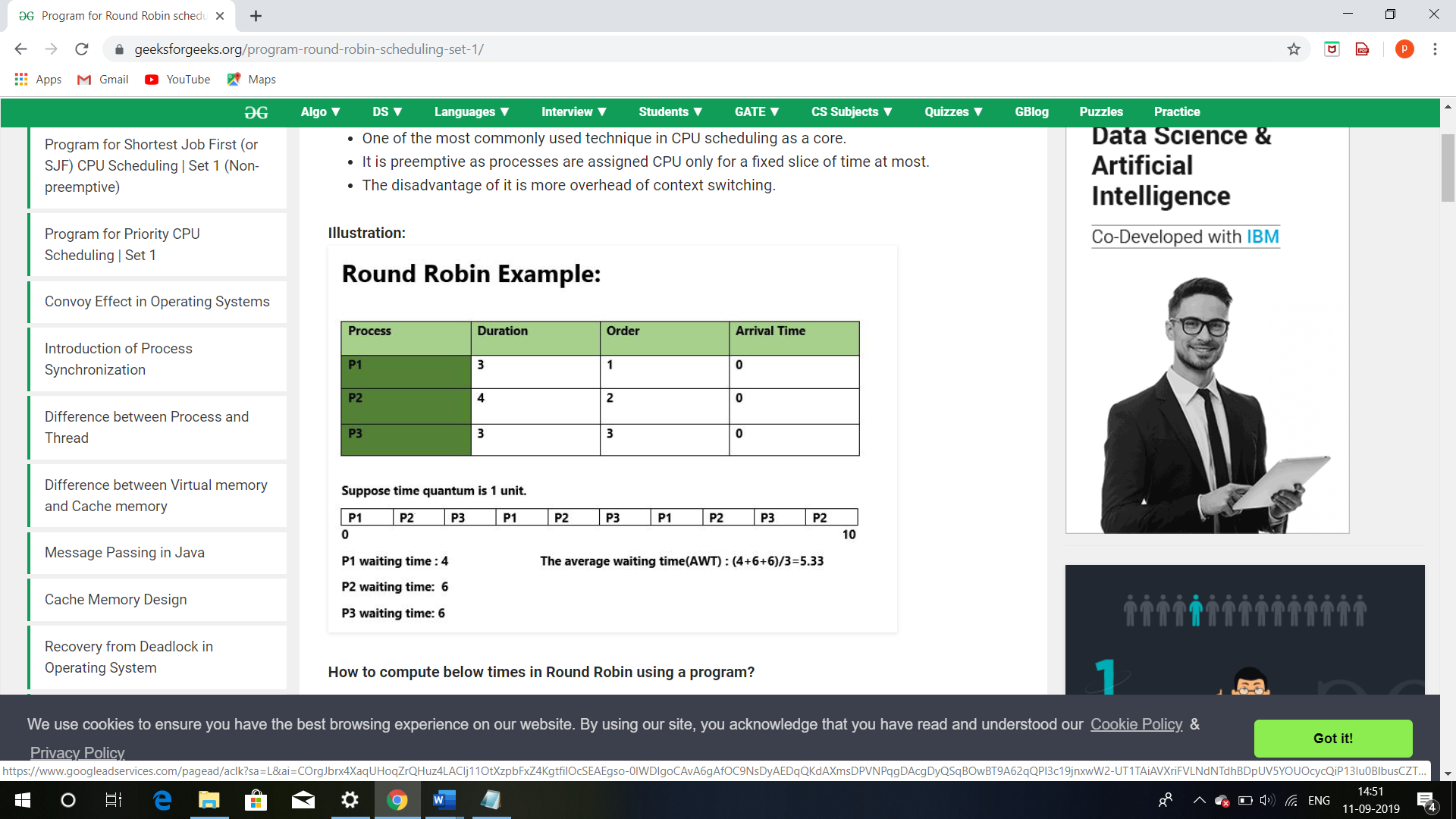
Government Polytechnic, Karad

**1.0 Rationale**

Multiprogramming operating system consists of various processes. On multiprogramming OS the processes run simultaneously for specific time. In the same way, process scheduler consists of various processes which will run for given time and when they are exited they are inserted on Queue and removed when the process is started. The algorithm used for this is named as Round Robin Algorithm. The program consists of implementation of this round robin algorithm.

Round Robin is a [CPU Scheduling Algorithm](http://quiz.geeksforgeeks.org/gate-notes-operating-system-process-scheduling/) where each process is assigned a fixed time slot in a cyclic way.

* It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.
* One of the most commonly used technique in CPU scheduling as a core.
* It is pre-emptive as processes are assigned CPU only for a fixed slice of time at most.
* The disadvantage of it is more overhead of context switching.



**Fig 1. Above is the example of round-robin scheduling algorithm**

**2.0 Aim of the project**

This micro-project aims were

1. To analyze the concepts of Multiprogramming Operating System.
2. To analyze the real-world applications of Operating System.
3. To implement the round robin algorithm.

We have achieved all the aims which were decided at the start of the project. We have completed the project with achieving all the aims at the completion.

**3.0 Course Outcomes Achieved**

1. Install Operating System and configure it.
2. Apply scheduling algorithms to calculate turnaround time and average waiting time.

**4.0 Literature Review**

|  |  |  |
| --- | --- | --- |
| **Author (Publication)** | **Abstract** | **Conclusion** |
| Neha Mittal (IJLTEMAS) | This journal explains working of round robin scheduling algorithm | We used this portal to analyze the working of round-robin scheduling algorithm and understand its working. |
| Pallabh Baneerje (IJARCSSE) | This paper explains about the c program for implementation of round- robin scheduling algorithm | We used this paper to analyze the c code and the concepts of c programming to be used in the implementation of c program. |
| Dr. Rajendra Kawale  (IOSR) | This paper explains about the concepts of the Multiprogramming Operating System. | We used this book to understand the working of the Multiprogramming Operating System. |
| Y. K. Singh (ResearchGate) | This paper explains about the concepts of the Operating System. | We used this paper to understand working of Multiprogramming Operating System. |

**Table 1- Literature Review**

**1.** <https://en.wikibooks.org/wiki/Operating_System_Design/SchedulingProcesses/RoundRobin>

We used this link to analyze various scheduling algorithms and study advantages and disadvantages of round-robin scheduling algorithm.

**2.**[https://https;//www.en.wikipedia.org/wiki/Operatingsystem](https://www.en.wikipedia.org/wiki/Operatingsystem)

We used this book to understand various concepts of operating system. We used this portal also for understanding different types of Operating System.

**5.0 Actual procedure followed**

Actual procedure followed during the project is mentioned in the table below with the planned start date and completed finish date.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Details of Activity** | **Planned Start Date** | **Completed Finish Date** | **Name of responsible Team Members** |
| 1. | Collecting the information of Process Scheduler | 25/07/2019 | 05/08/2019 | 1.Patil Pratik  2.Kare Abhilash |
| 2. | Collecting information of multi programming operating system | 07/08/2019 | 16/08/2019 | 1.Shinde Mohini  2.Awate Amruta |
| 3. | Collecting Information about round robin algorithm | 17/08/2019 | 28/08/2019 | 1.Shinde Mohini  2.Awate Amruta |
| 4. | Designing the logic of the round robin algorithm | 29/08/2019 | 05/09/2019 | 1.Kare Abhilash  2.Patil Pratik |
| 5. | Actual Coding of the project | 06/09/2019 | 15/09/2019 | 1.Patil Pratik  2.Kare Abhilash  3.Shinde Mohini  4.Awate Amruta |
| 6. | Observing Output | 16/09/2019 | 26/09/2019 | 1.Shinde Mohini  2.Awate Amruta  3.Patil Pratik  4.Kare Abhilash |
| 7. | Making Report | 27/09/2019 | 05/10/2019 | 1.Patil Pratik  2.Kare Abhilash  3.Shinde Mohini  4.Awate Amruta |

**Table 2- Actual procedure followed**

**C code for Round- Robin Scheduling Algorithm with Gantt chart:**

#include<stdio.h>

struct times

{

int p,art,but,wtt,tat,rnt;

};

void sortart(struct times a[],int pro)

{

int i,j;

struct times temp;

for(i=0;i<pro;i++)

{

for(j=i+1;j<pro;j++)

{

if(a[i].art > a[j].art)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

return;

}

int main()

{

int i,j,pro,time,remain,flag=0,ts;

struct times a[100];

float avgwt=0,avgtt=0;

clrscr();

printf("\t\t\t\*\*\*Round Robin Scheduling Algorithm\*\*\*\n\n\n");

printf("Enter Number Of Processes : ");

scanf("%d",&pro);

remain=pro;

for(i=0;i<pro;i++)

{

printf("Enter Arrival time for Process P[%d] : ",i);

scanf("%d",&a[i].art);

printf("Enter Burst time for Process P[%d] : ",i);

scanf("%d",&a[i].but);

a[i].p = i;

a[i].rnt = a[i].but;

}

sortart(a,pro);

printf("Enter Time Quantum : ");

scanf("%d",&ts);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("Gantt Chart\n\n");

printf("%d",a[0].art);

for(time=a[0].art,i=0;remain!=0;)

{

if(a[i].rnt<=ts && a[i].rnt>0)

{

time = time + a[i].rnt;

printf(" -> [P%d] <- %d",a[i].p,time);

a[i].rnt=0;

flag=1;

}

else if(a[i].rnt > 0)

{

a[i].rnt = a[i].rnt - ts;

time = time + ts;

printf(" -> [P%d] <- %d",a[i].p,time);

}

if(a[i].rnt==0 && flag==1)

{

remain--;

a[i].tat = time-a[i].art;

a[i].wtt = time-a[i].art-a[i].but;

avgwt = avgwt + time-a[i].art-a[i].but;

avgtt = avgtt + time-a[i].art;

flag=0;

}

if(i==pro-1)

i=0;

else if(a[i+1].art <= time)

i++;

else

i=0;

}

printf("\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("| Process | Arrival Time | Burst Time | Turnaround Time | Waiting Time |\n");

for(i=0;i<pro;i++)

{

printf("| P[%d]\t | %d\t\t | %d\t | %d\t\t| %d\t |\n",a[i].p,a[i].art,a[i].but,a[i].tat,a[i].wtt);

}

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("Results\n");

avgwt = avgwt/pro;

avgtt = avgtt/pro;

printf("Average Waiting Time : %.2f\n",avgwt);

printf("Average Turnaround Time : %.2f\n",avgtt);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

getch();

return 0;

}

**Advantages of Round- Robin Scheduling Algorithm**

* Each process is served by the CPU for a fixed time quantum, so all processes are given the same priority.
* Starvation doesn't occur because for each round robin cycle, every process is given a fixed time to execute. No process is left behind.

**Disadvantages ofRound- Robin Scheduling Algorithm**

* The throughput in RR largely depends on the choice of the length of the time quantum. If time quantum is longer than needed, it tends to exhibit the same behaviour as FCFS.
* If time quantum is shorter than needed, the number of times that CPU switches from one process to another process, increases. This leads to decrease in CPU efficiency.

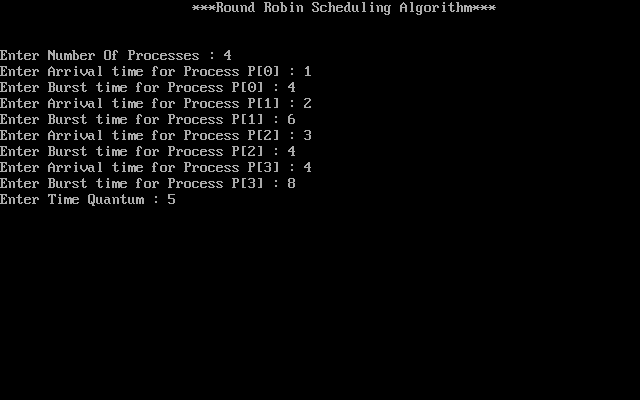
**6.0 Actual Resources Required**

The resources used during the completion of project are mentioned in the below table:

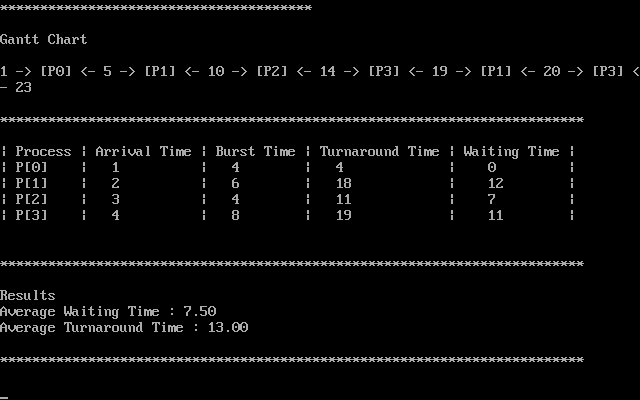
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Name of resources material** | **Specifications** | **Quantity** | **Remarks** |
| 1. | YouTube | MP4-file format, 640 x 360 pixels | 1 |  |
| 2. | Microsoft Word | 2007 version | 1 |  |
| 3. | Turbo C7 | TURBO C7 compatible with Windows 7,8 and vista | 1 |  |

**Table 3- Resources Required**

**7.0 Outputs of the Micro-project**



**Fig 2. Screenshot of the c program- input values**



**Fig 3. Screenshot of the c program- ouput of program**

**8.0 Skill Developed/ learning out of this Micro-Project**

We learnt,

1. Developing a c program for implementation of Round- Robin Algorithm.
2. Importance of Operating System and `CPU scheduling.
3. Importance of Multiprogramming Operating System.
4. Efficient communication skills.
5. Working as a team member for developing c program.
6. Developing leadership qualities.

**9.0 Applications of the Project**

1. This project can be used in the implementation of Round- Robin Scheduling Algorithm.
2. The project can be also used to understand importance of CPU scheduling.
3. The project can be used in Multiprogramming Operating System to calculate turnaround time and waiting time.

**Subject Teacher**

**Mrs. Amrutkar B R.**