MULTI-TYPE SCHEDULER

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Perumalla Dharan – AP21110010201

Tarun Teja Kudeti – AP21110010205

NVSS Pavan Sastry - AP21110010209

Vatala Phalgun – AP21110010223

Grandhi Dinesh – AP21110010240

Under the Guidance of

Manikandan V M

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

Certificate

Date: xx-xx-xxxx

This is to certify that the work present in this Project entitled "**MULTI-TYPE SCHEDULER**" has been carried out by **Perumalla Dharan (AP21110010201) , Kudeti Tarun Teja (AP21110010205) , NVSS Pavan Sastry (AP21110010209) , Vatala Phalgun (AP21110010223) , , Grandhi Dinesh (AP21110010240)** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. / Dr. **Manikandan VM**

Designation,

Affiliation.

Co-supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

The Objective of the Project:

Multi Type Scheduler which runs the algorithms that were being taught in our course operating systems CSE 301 such as CPU scheduling algorithms, deadlock handling, page replacement algorithms and semaphores etc. Our program provides an explanation along with an example and also provides the user to test their knowledge about the algorithm by running the code.

Tools and Technologies Used:

The Technologies we used in the multi-type scheduler are many.
They are:

- ➔ Our Algorithms in C++ language implemented in visual studio code and their version controlled in GitHub.

Project Code (MAIN):

```
// Operating System Project
//
// Group Members:
// 1. Perumalla Dharan        |      AP21110010201
// 2. Tarun Teja Kudeti       |      AP21110010205
// 3. Pavan Sastry NVSS       |      AP21110010209
// 4. Vatala Pahlgun          |      AP21110010223
// 5. Grandhi Dinesh          |      AP21110010240

#include <iostream>
#include "Header_Files\CPU_header.h"
#include "Header_Files\Page_header.h"
#include "Header_Files\Disc_header.h"
#include "Header_Files\Deadlocks.h"

using namespace std;

int main()
{
    int n;
    while (true)
    {
        cout << "-------------------------------------------------------------------
-----------------------------------------" << endl;
        cout << "----------------------------------------Operating System
Project----------------------------------------" << endl;
        cout << "-------------------------------------------------------------------
-----------------------------------------" << endl;
        cout << endl;
        cout << "\t\t\t\t\t\tWelcome" << endl;
        cout << "\t\t\t\t\tChoose any one of the following" << endl;
        cout << "\t\t\t\t\t1. CPU Scheduling Algorithms" << endl;
        cout << "\t\t\t\t\t2. Page Replacement Algorithms" << endl;
        cout << "\t\t\t\t\t3. Disk Scheduling Algorithms" << endl;
        cout << "\t\t\t\t\t4. Semaphores and Deadlocks" << endl;
        cout << "\t\t\t\t\t5. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> n;
        system("cls");
        switch (n)
        {
        case 1:
            cpu_menu();
            break;
        case 2:
            // page_menu();
            break;
```

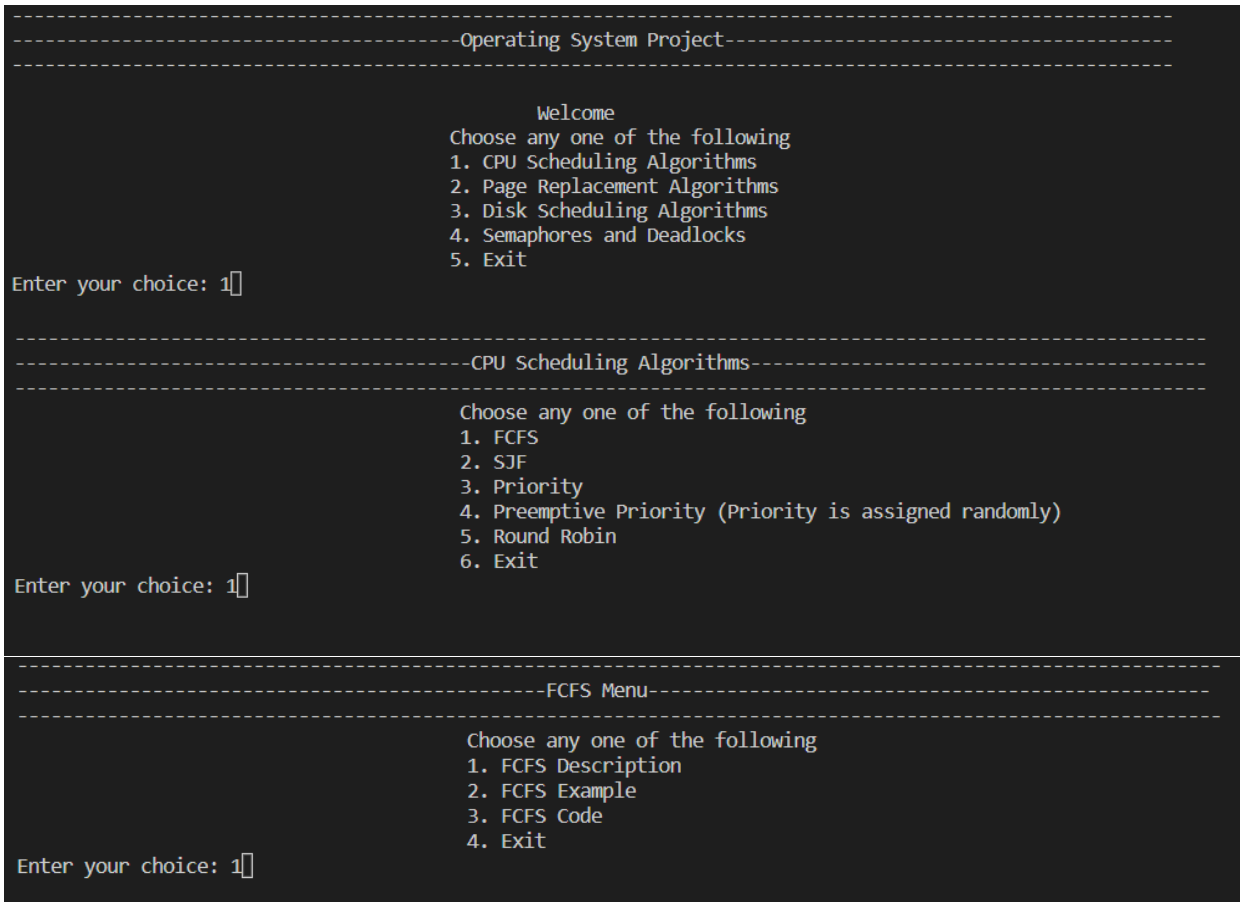
```
        case 3:
            // disc_menu();
            break;
        case 4:
            // deadlocks_menu();
            break;
        default:
            break;
        }
    }
}
```

OUTPUT:

NOTE:

- The source code for algorithms used in main are provided in their respective header files.

- I.e., in their respective user defined sub folder.



```
First-Come First-Serve

First-Come, First-Served is a scheduling algorithm used in operating systems. It is a non-preemptive
scheduling algorithm, meaning that once a process or task is allocated the CPU , it will continue to
run until it completes or voluntarily gives up the CPU.In the FCFS algorithm, the processes are
executed in the order they arrive in the system. When a process finishes its execution, the next
process in the queue gets the CPU.

Advantage:-
The main advantage of the FCFS algorithm is its simplicity. It is easy to understand and implement.


Disadvantage:-
It can suffer from the  convoy effect  where a long process may hold up other shorter processes that
arrive later which increse in wait time


Example:-
      Process id            arrival time
      1                        3
      2                        2
      3                        1
      4                        0
      5                        6

solution:-

4->3->2->1->6
-------------------------------------------------------------------------------------------------------
----------------------------------------Operating System Project---------------------------------------
-------------------------------------------------------------------------------------------------------

                                          Welcome
                                    Choose any one of the following
                                    1. CPU Scheduling Algorithms
                                    2. Page Replacement Algorithms
                                    3. Disk Scheduling Algorithms
                                    4. Semaphores and Deadlocks
                                    5. Exit
Enter your choice: 1

-------------------------------------------------------------------------------------------------------
-------------------------------------------------FCFS Menu---------------------------------------------
-------------------------------------------------------------------------------------------------------
                                       Choose any one of the following
                                       1. FCFS Description
                                       2. FCFS Code
                                       3. Exit
Enter your choice: 2
```

```
Enter the number of processes: 5

Enter the process ID: 1
Enter the arrival time: 4
Enter the burst time: 9

Enter the process ID: 2
Enter the arrival time: 3
Enter the burst time: 5

Enter the process ID: 3
Enter the arrival time: 9
Enter the burst time: 10

Enter the process ID: 4
Enter the arrival time: 8
Enter the burst time: 12

Enter the process ID: 5
Enter the arrival time: 11
Enter the burst time: 14


Process ID      Arrival Time    Burst Time      Completion Time Turn Around Time        Wait Time
2               3               5               8               5               0
1               4               9               17              13              4
4               8               12              29              21              9
3               9               10              39              30              20
5               11              14              53              42              28

Average Wait Time: 12.2
Average Turn Around Time: 22.2
 ------------------------------------------------------------------------------------------------------------
 ----------------------------------------------FCFS Menu-----------------------------------------------------
 ------------------------------------------------------------------------------------------------------------
                                    Choose any one of the following
                                    1. FCFS Description
                                    2. FCFS Code
                                    3. Exit
 Enter your choice: 3
```

Future work possible:

In Future, we might consider turning this project into a website and can add few more algorithms. So that people from all over the world can learn and be more productive in operating system course.

CONCLUSION:

This project provides an idea and playground to execute the algorithms that are being taught in operating system CSE 301.