

**PROJECT REPORT**

**Project Title**

Scheduling algorithms

**COURSE: Operating System (OS)**

**CID: 110646**

|  |  |  |
| --- | --- | --- |
| **GROUP MEMBERS** | | |
| **S.NO** | **Name** | **Student ID** |
| 1 | Safiullah Rehmani | 12413 |
| 2  3 | Shaheer Shakeel  Shahmeer Khan | 12452  12113 |

**INSTRUCTED BY**

**Dr. Naveed Sheikh**

# TABLE OF CONTENTS

**TABLE OF CONTENTS**

**LIST OF FIGURES**

**Chapter 1 PROJECT OVERVIEW**

1.1 Project Objective

1.2 Project Description

**Chapter 2 Tools and Technologies**

2.1 Visual Studio (Windows Form) C#

**Chapter 3 FEATURES AND WORKING OF PROJECT**

3.1 Scheduling Algorithms

**Appendix A SNAPSHOTS**

# Chapter 1

# Project Overview

In this scheduling algorithm we can add range to enter the number of process after that we will select the algorithm from combo box to ask user to input

necessary information to execute, according to the selected algorithm and then perform some calculation and print the result.

##### Project Description

**CPU Scheduling** is a process of determining which process will own CPU for execution while another process is on hold. The main task of CPU scheduling is to make sure that whenever the CPU remains idle, the OS at least select one of the processes available in the ready queue for execution. The selection process will be carried out by the CPU scheduler. It selects one of the processes in memory that are ready for execution.

## Types of CPU Scheduling

### Preemptive Scheduling

In Preemptive Scheduling, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

### Non-Preemptive Scheduling

In this type of scheduling method, the CPU has been allocated to a specific process. The process that keeps the CPU busy will release the CPU either by switching context or terminating. It is the only method that can be used for various hardware platforms. That’s because it doesn’t need special hardware (for example, a timer) like preemptive scheduling.

### When scheduling is Preemptive or Non-Preemptive?

To determine if scheduling is preemptive or non-preemptive, consider these four parameters:

1. A process switches from the running to the waiting state.
2. Specific process switches from the running state to the ready state.
3. Specific process switches from the waiting state to the ready state.
4. Process finished its execution and terminated.

**Only conditions 1 and 4 apply, the scheduling is called non- preemptive.**

**All other scheduling are preemptive.**

## Types of CPU scheduling Algorithm

There are mainly five types of process scheduling algorithms

1. First Come First Serve (FCFS)
2. Shortest-Job-First (SJF) Scheduling
3. Priority Scheduling
4. Round Robin Scheduling

## First Come First Serve

First Come First Serve is the full form of FCFS. It is the easiest and most simple CPU scheduling algorithm. In this type of algorithm, the process which requests the CPU gets the CPU allocation first. This scheduling method can be managed with a FIFO queue.

As the process enters the ready queue, its PCB (Process Control Block) is linked with the tail of the queue. So, when CPU becomes free, it should be assigned to the process at the beginning of the queue.

### Characteristics of FCFS method:

* It offers non-preemptive and pre-emptive scheduling algorithm.
* Jobs are always executed on a first-come, first-serve basis
* It is easy to implement and use.
* However, this method is poor in performance, and the general wait time is quite high.

## Priority Based Scheduling

Priority scheduling is a method of scheduling processes based on priority. In this method, the scheduler selects the tasks to work as per the priority.

Priority scheduling also helps OS to involve priority assignments. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority can be decided based on memory requirements, time requirements, etc.

## Round-Robin Scheduling

Round robin is the oldest, simplest scheduling algorithm. The name of this algorithm comes from the round-robin principle, where each person gets an equal share of something in turn. It is mostly used for scheduling algorithms in multitasking. This algorithm method helps for starvation free execution of processes.

### Characteristics of Round-Robin Scheduling

* Round robin is a hybrid model which is clock-driven
* Time slice should be minimum, which is assigned for a specific task to be processed. However, it may vary for different processes.
* It is a real time system which responds to the event within a specific time limit.

## Shortest Job First

SJF is a full form of (Shortest job first) is a scheduling algorithm in which the process with the shortest execution time should be selected for execution next. This scheduling method can be preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution.

### Characteristics of SJF Scheduling

* It is associated with each job as a unit of time to complete.
* In this method, when the CPU is available, the next process or job with the shortest completion time will be executed first.
* It is implemented with non-preemptive policy.
* This algorithm method is useful for batch-type processing, where waiting for jobs to complete is not critical.
* It improves job output by offering shorter jobs, which should be executed first, which mostly have a shorter turnaround time.

##### Project Purpose

## The Purpose of a Scheduling algorithm

Here are the reasons for using a scheduling algorithm:

* The CPU uses scheduling to improve its efficiency.
* It helps you to allocate resources among competing processes.
* The maximum utilization of CPU can be obtained with multi-programming.
* The processes which are to be executed are in ready queue.

# Chapter 2

# Visual Studio:

Visual Studio is an **Integrated Development Environment (IDE)** developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

# Chapter 3

## Features and working of project

**Scheduling Algorithm:**

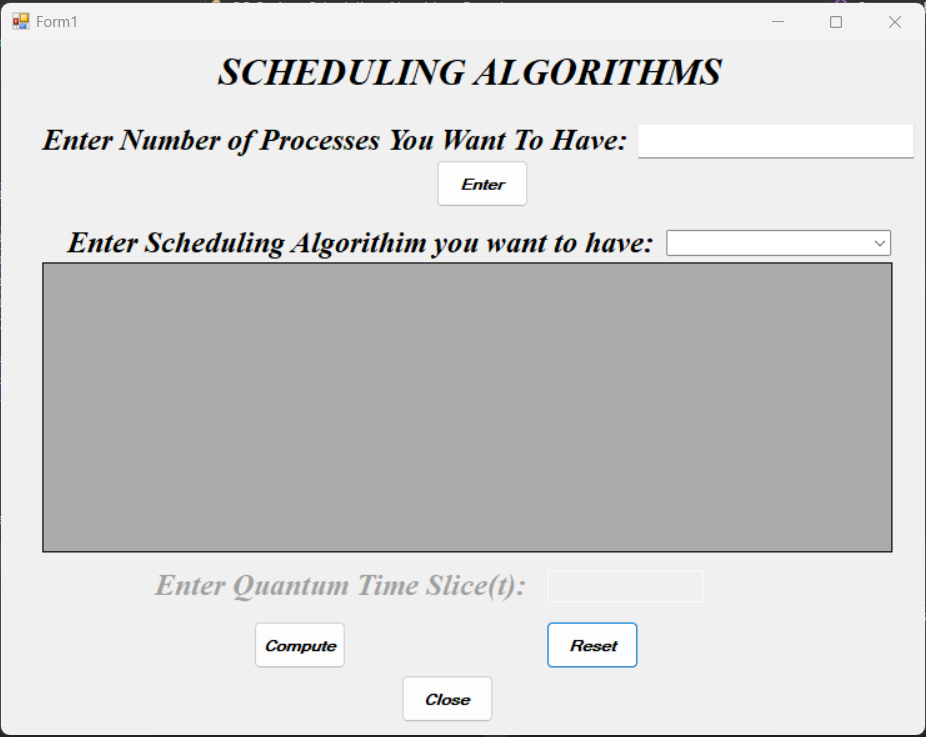
In this Scheluding Algorithm we have made five (5) forms, first form (1) form is main which enter the range for number of processes and then select which algorithm do you want to perform after selecting the algorithm user click on enter button and then insert values in given Text Box like arrival Time, Burst Time, Priority, Time Slice on the basis of selected algorithms after that it save the values in an array and perform calculation for Waiting Time, Turnaround Time and swap the values for completing the conditions of Algorithm after that it print all values on the forms.

**For Example:**

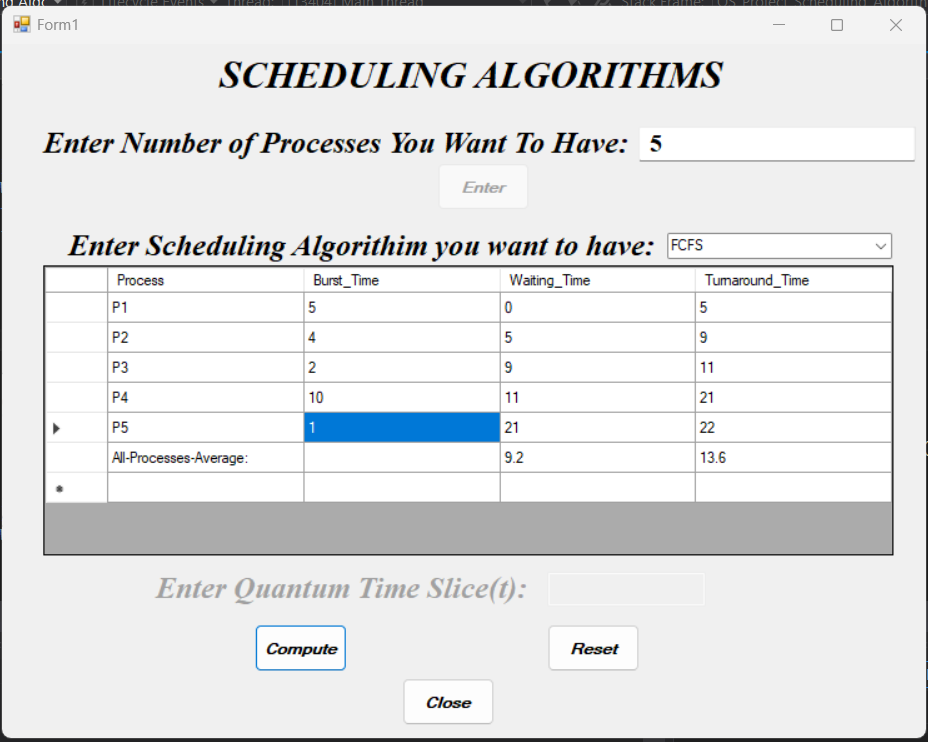
If the user select “FCFS” Algorithm then it will print the value of number of process, Arrival Time, Burst Time, Waiting Time, Turnaround Time in the third form (3) and if the user click on exit button and it will return to Form (1) and again ask user to enter and select all the information until the user exit form main form.

**SCREENSHOTS:**

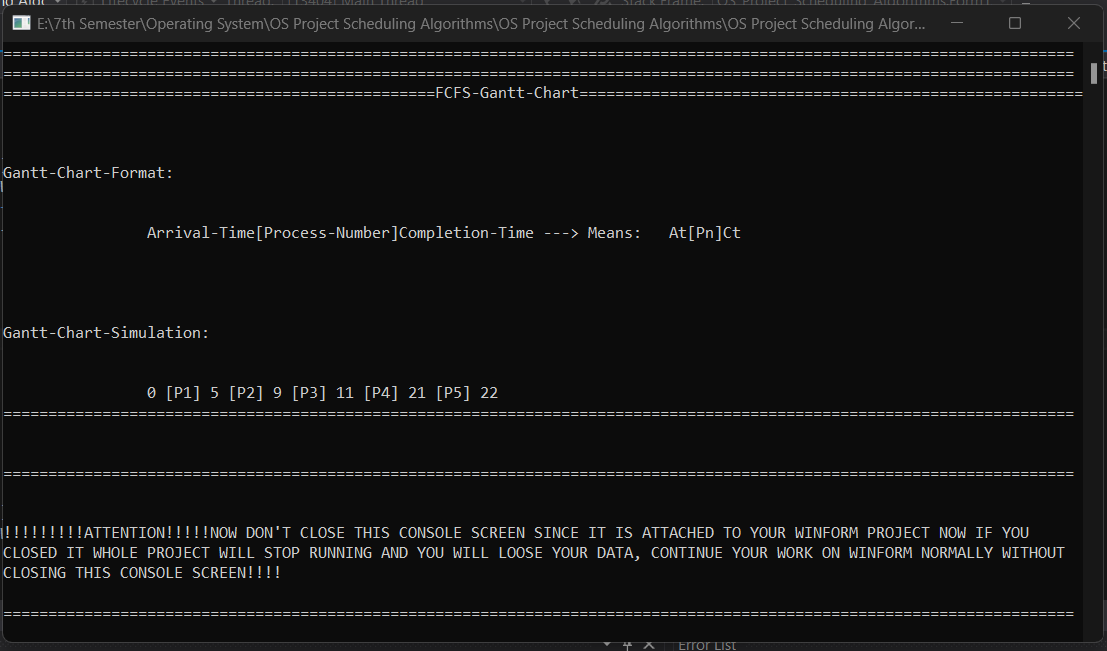
**Main form:**

****

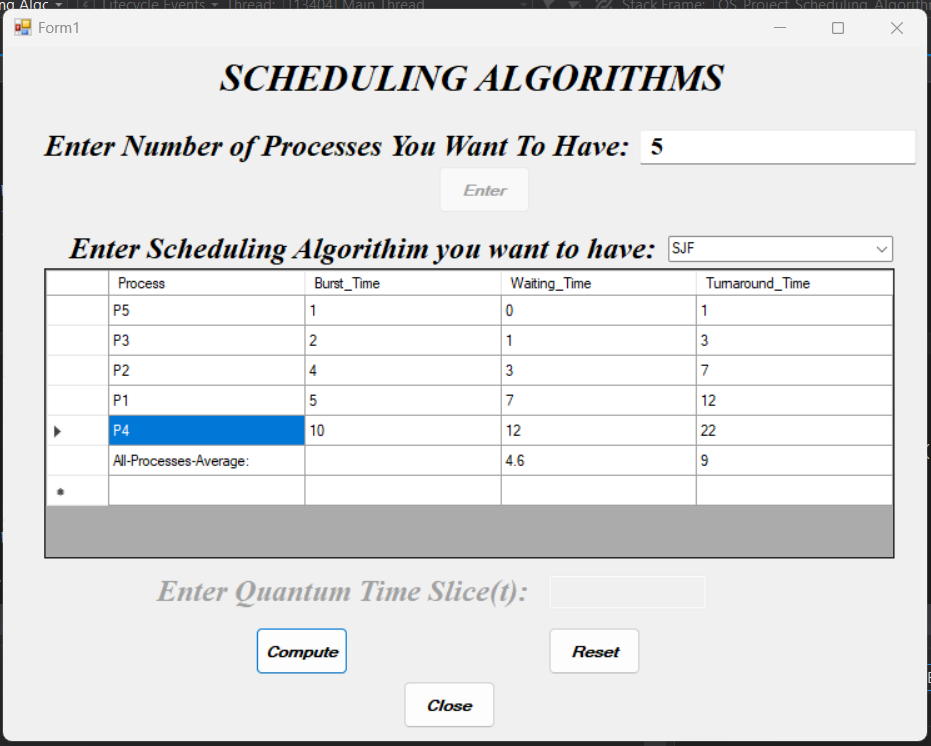
**fcfs(First Come First Serve) form:**

****

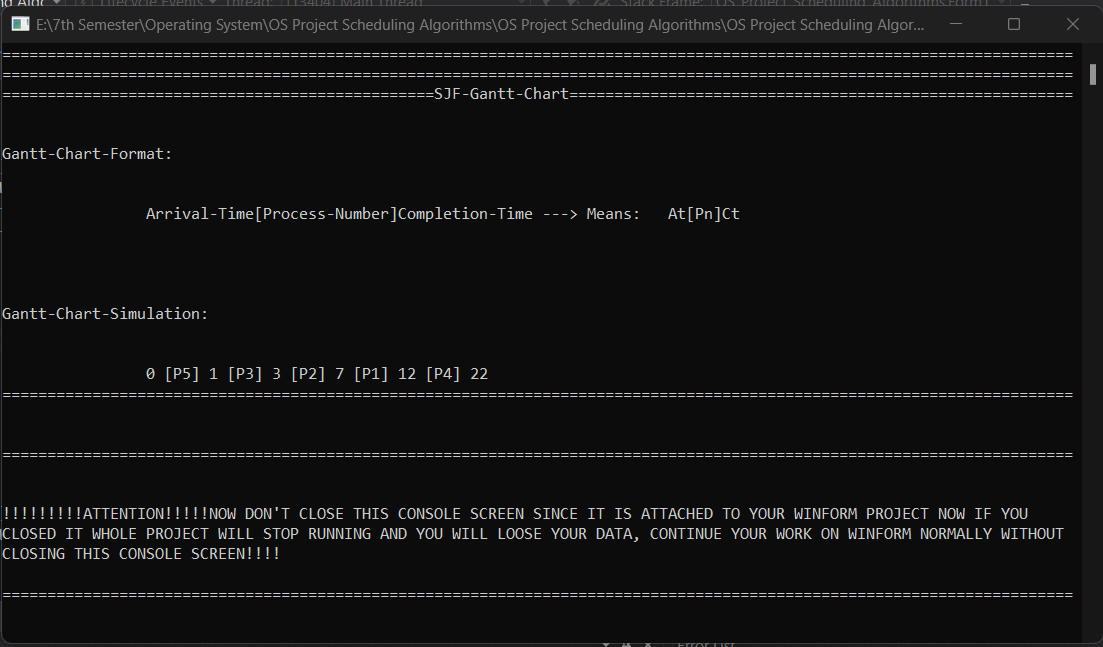
**Gannt Chart Simulation:**

****

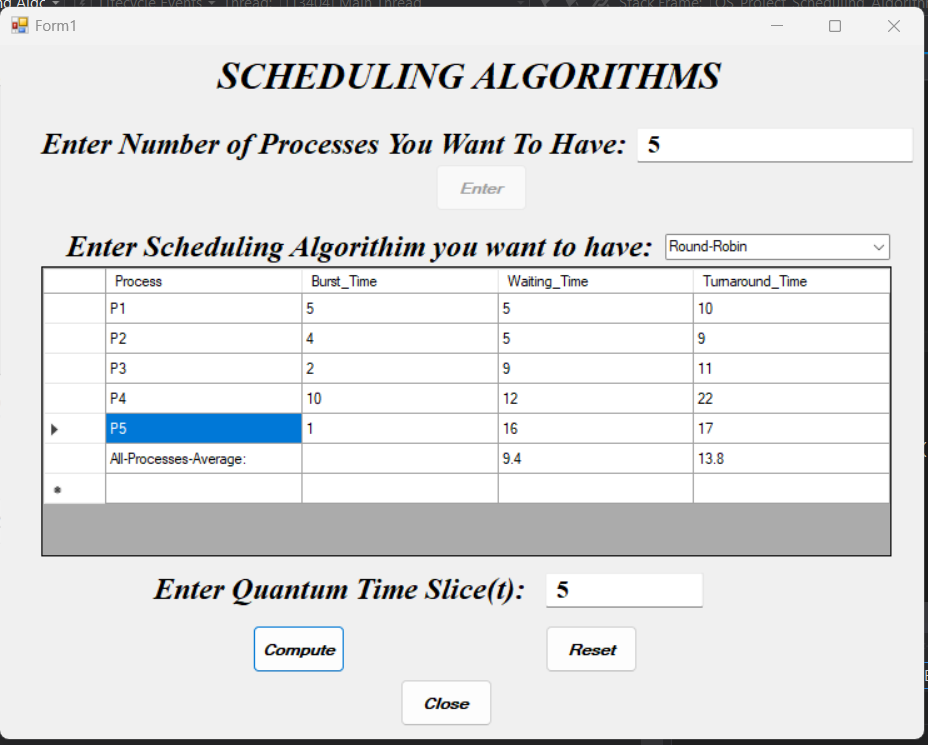
**sjf(Shortest Job First) form:**

****

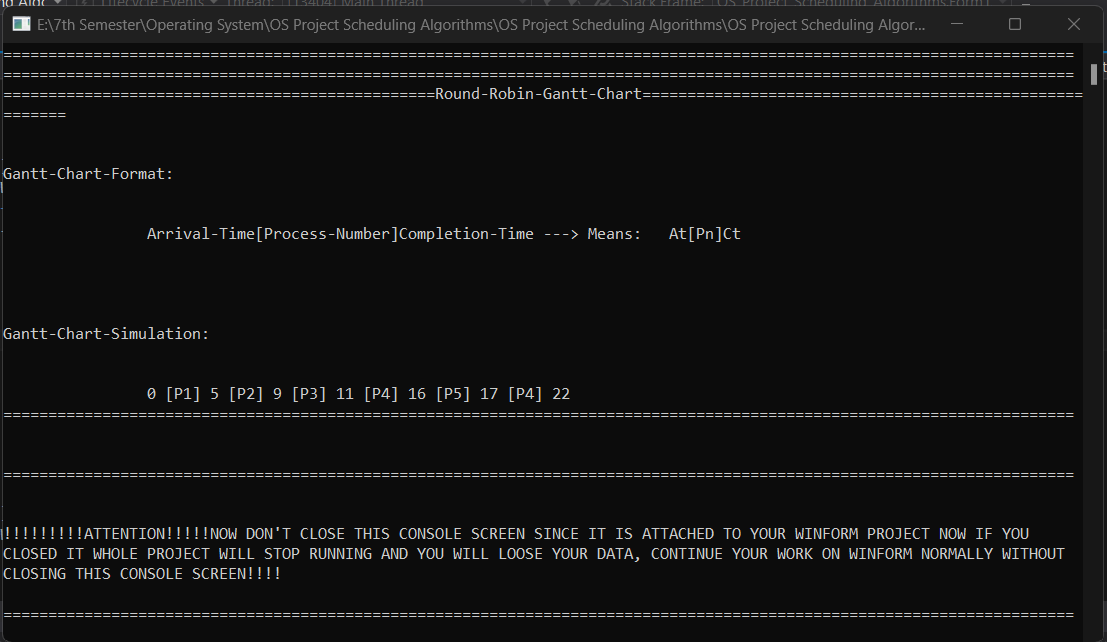
**Gannt Chart Simulation:**

****

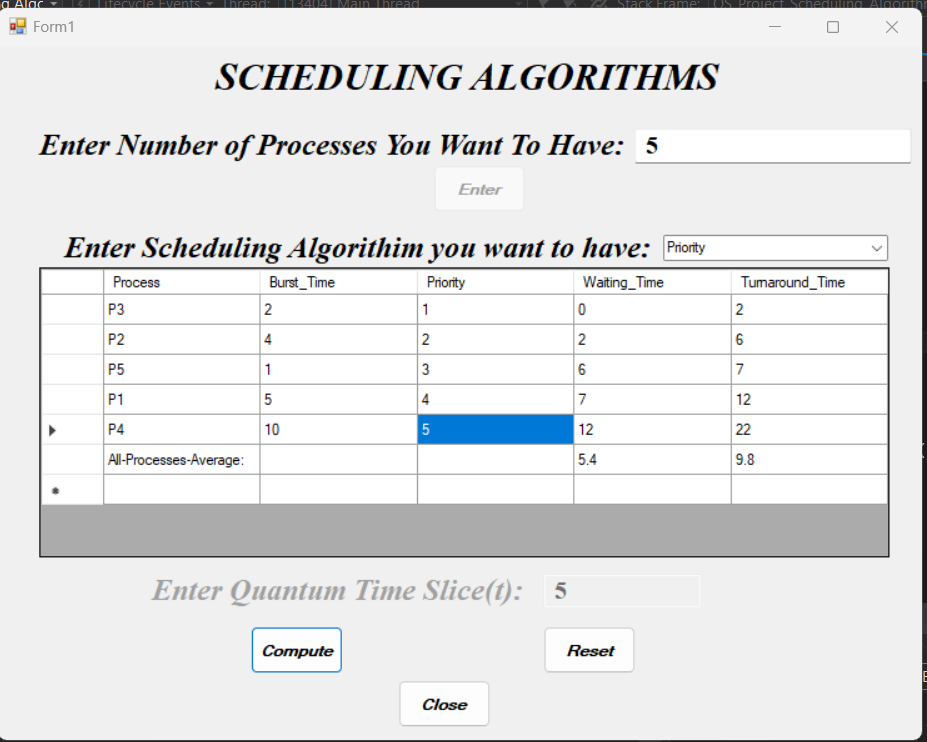
**round robin form:**

****

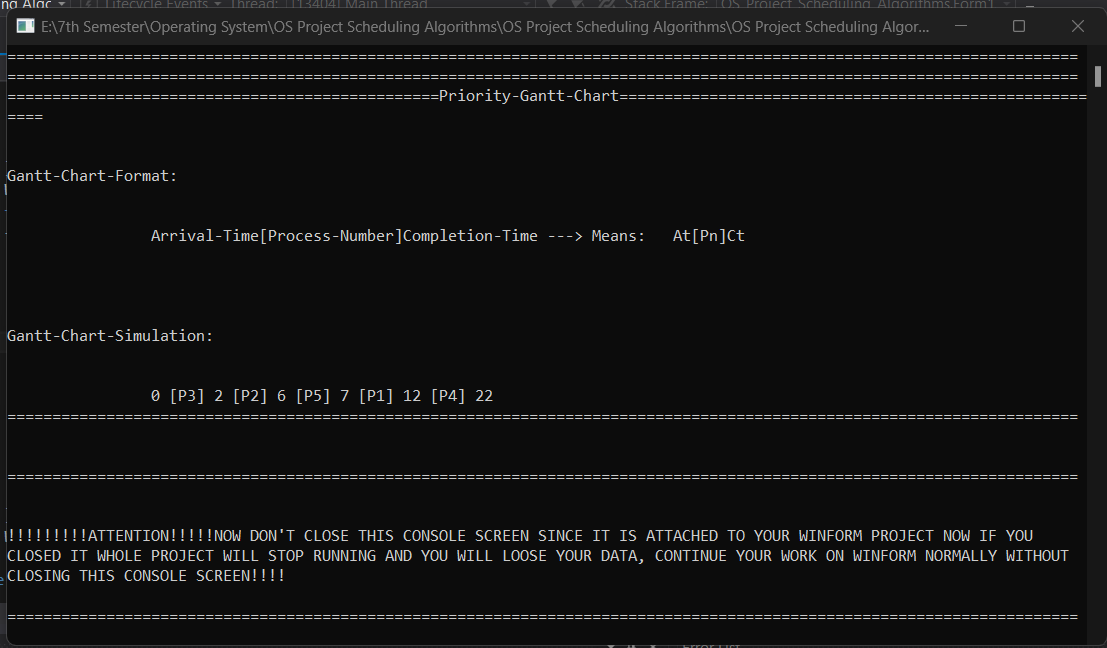
**Gannt Chart Simulation:**

****

**priority form:**

****

**Gannt Chart Simulation:**

****