Content Page

CAOS Project Report

1. INTRODUCTION

The project aimed is to apply CPU Scheduling, Main and Virtual Memory allocation methods in real coding. For the past few weeks, the team has put in a lot of efforts and developed a C# Window Form Application in Microsoft Visual Studio 2015 with the following objectives:

- Create and stored processes with their arrival time, burst time and their priority.
- Implemented CPU Scheduling allocation with the stored processes via First Come First Served and Non-Preemptive Priority algorithm and display the result in Gantt Chart and calculate their Average Turnaround Time and Average Waiting Time or even Response Time and CPU Utilization.
- Implement Main Memory for best-fit to display the user's input memory and processing blocks. It can also calculate any internal and external fragmentation that occurs and display unloaded processes.
- Virtual Memory is implemented to check there is how many page fault in a page reference string and calculate page occurrence in it.

2. THEORETICAL BACKGROUND

CPU Scheduling FCFS:

First come first serve which means that the process that has the lowest arrival time will be executed first within the queue.

CPU Scheduling - Non Preemptive Priority:

Non-Preemptive Priority means that the Process that has the shortest Arrival Time will go first. If a process has not arrived it will remain as Idle. A running task is executed till completion. It cannot be interrupted as well. Those Processes that arrived shall be in the Ready Queue. After a running task has completed its execution, the Process (in the Ready Queue) that has the highest Priority will go next. It will then repeat until it finishes completed all the processes.

Main Memory - Best-Fit Allocation:

The best-fit allocation uses the smallest possible memory block that can fit the process. It makes the best use of memory space but it is slower in making allocation. It is able to minimize any external fragmentation due to its efficient memory allocation.

Virtual Memory:

Page replacement algorithms are needed for computer system to decide which memory page to page out when a page memory is required to be allocated to the memory. A page fault is occurred when the page is not in the frame.

3. SOFTWARE DESIGN

3.1. Screen Captures

Main Page

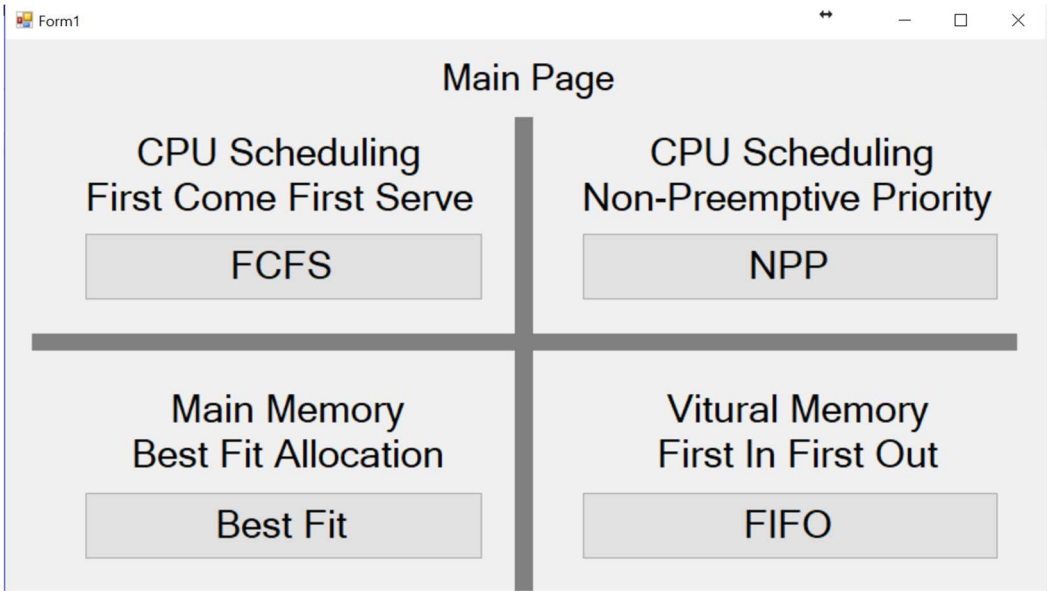


Figure 1: Main Page of Application

CPU Scheduling - FCFS Algorithm

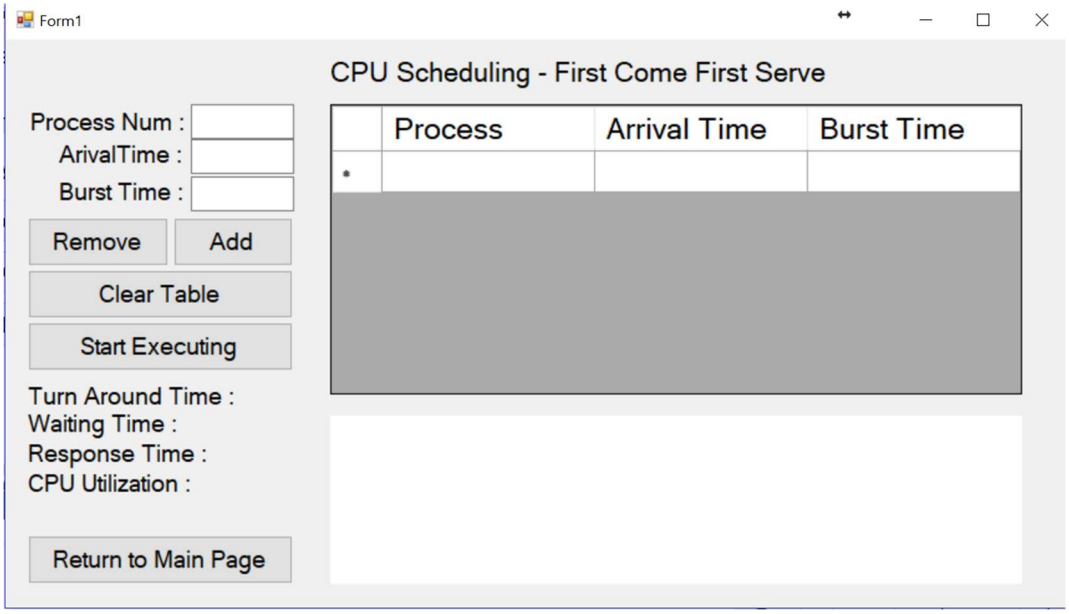


Figure 2: First Come First Serve GUI

User first keys in Process Number, Arrival Time and Burst time. User then click the add button to add in the Process data. Once done adding all the Process data user click start executing and the data will be calculated and shown in the label field below along with a Gantt Chart.

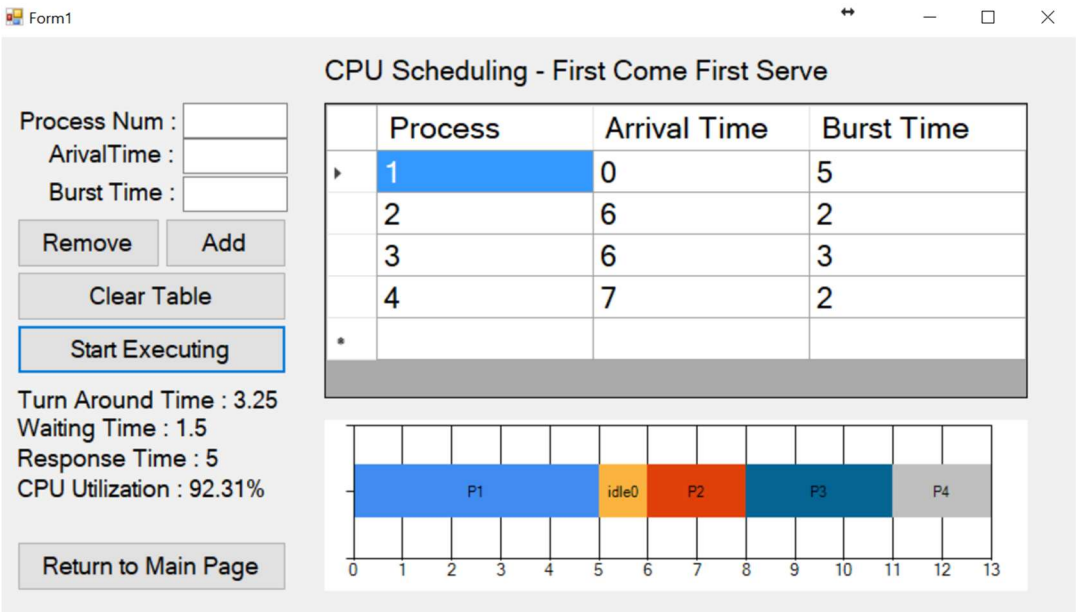


Figure 3: First Come First Serve Result

Above in figure 3 shown the calculations and Gantt Chart.

CPU Scheduling - Non-Preemptive Priority Algorithm

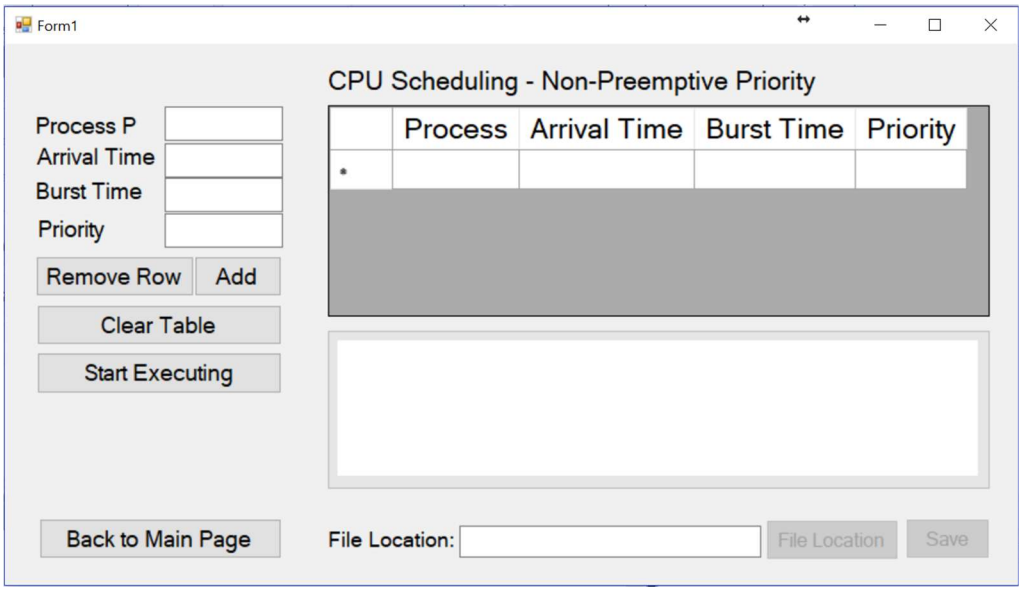


Figure 4: Non-Preemptive Priority GUI

Firstly in Figure 4, the user will have to input their selected Process, Arrival Time, Burst Time and Priority values into the TextBox and select the Add Button. Then the Table will be updated depending on the selection made by the user. If the user made a mistake they are able to delete the whole row (Select row and click on Remove Row) or clear the table.

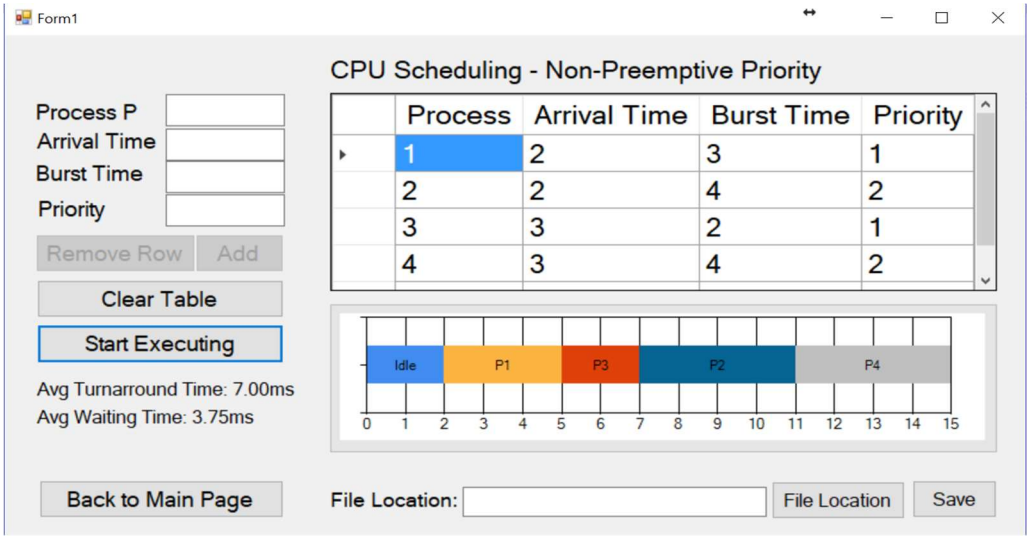


Figure 5: Non-Preemptive Priority Result

After filling up the table the user can Start Executing and the result will be calculated and display it as a Gantt Chart (Figure 5). The Average Turnaround Time and the Average Waiting Time will be calculated in code behind page and also display it to the user.

	A	B	C	D	E
1	Date:	3/2/2018			
2	Time	22 Hrs :52 Min :52 Sec			
3					
4	Process	Arrival Time	Burst Time	Priority	
5	1	2	3	1	
6	2	2	4	2	
7	3	3	2	1	
8	4	3	4	2	
9					
10	Result				
11	Idle	2			
12	P1	5			
13	P3	7			
14	P2	11			
15	P4	15			
16					
17	Avg TAT:	7ms			
18	Avg WT:	3.75ms			
19					

Figure 6: Non-Preemptive Priority Result in Excel File

After Executing and displaying the result, the user will have an option on whether they want the result to be save in Excel or Text File. By selecting a file location to Save. Figure # shows the result that will be recorded into the Excel File. The Table Selection made by the user will be recorded into the Excel File along with the result of Gantt Chart and the calculated Average Turnaround and Waiting Time.

Main Memory - Best Fit

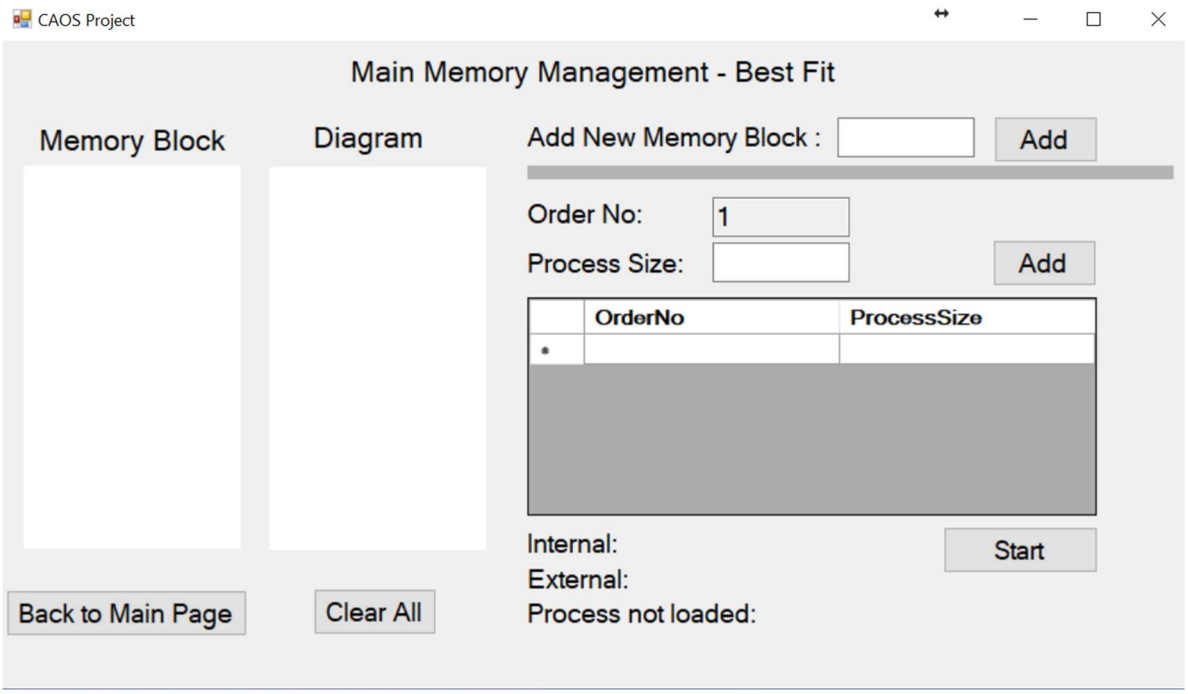


Figure 7: Main Memory Management - Best Fit GUI

User have to input memory blocks according to the size that they want and the number of memory blocks they want to add. They can also input the process size based on the order that they want the process size to be inserted in the memory block. The added memory blocks, process sizes and their orders number will be shown on a chart as shown in Figure 8.

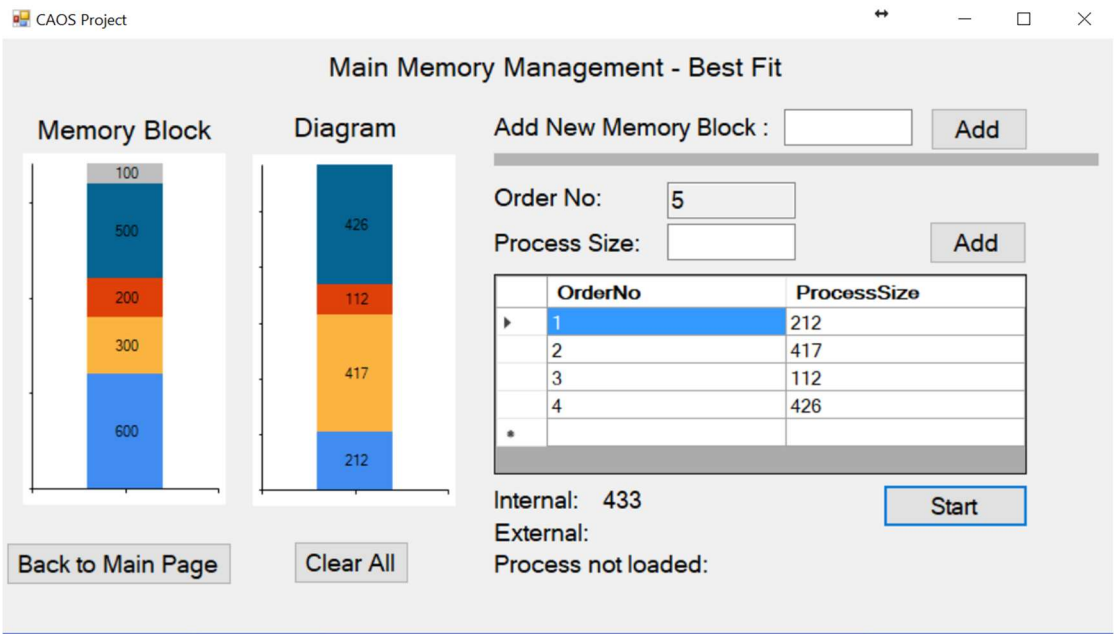


Figure 8: Main Memory Management - Best Fit Result GUI

The user can click the start button to begin the best fit memory allocation. It will display the internal fragmentation. It will also display any external fragmentation and processes that are not loaded as shown in Figure 9.

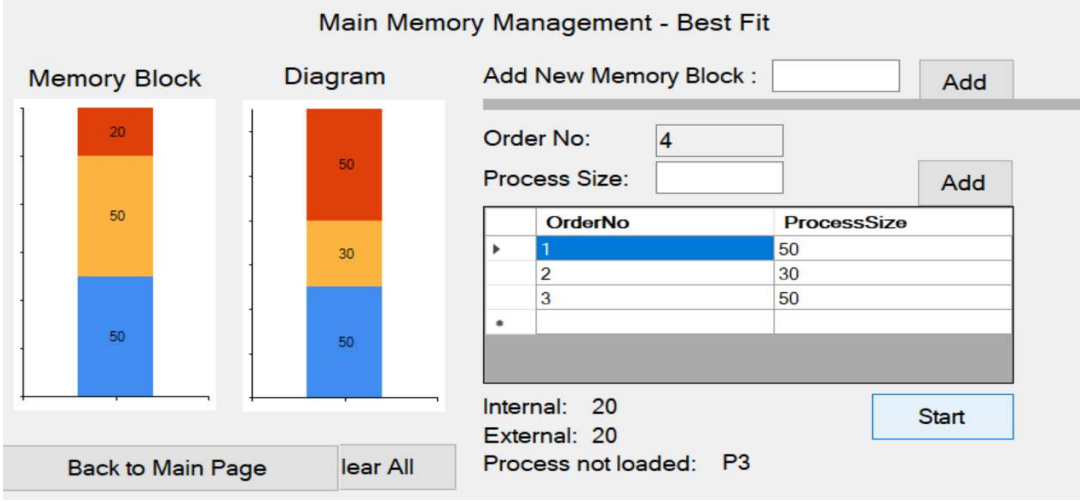


Figure 9: Main Memory Management -
Best Fit Result with external fragmentation and process not loaded

Virtual Memory - FIFO and LRU

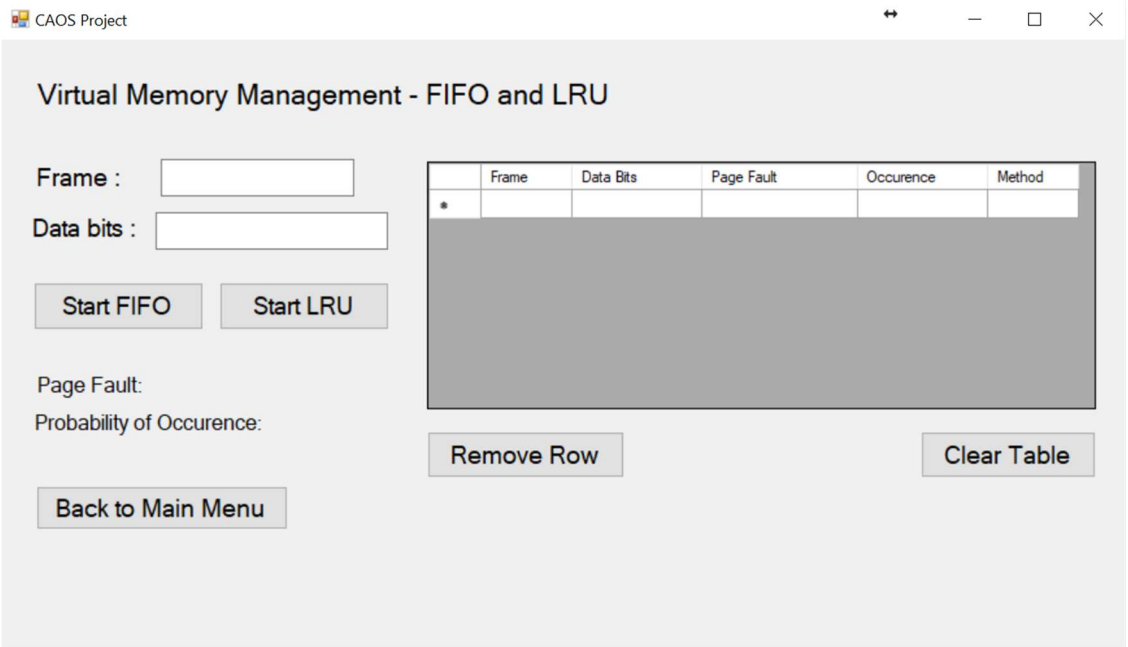


Figure 10: Virtual Memory Management - FIFO & LRU GUI

The user need to enter the frame and key in the page reference string. The user pressed FIFO button to do FIFO. The user pressed LRU button to do LRU.

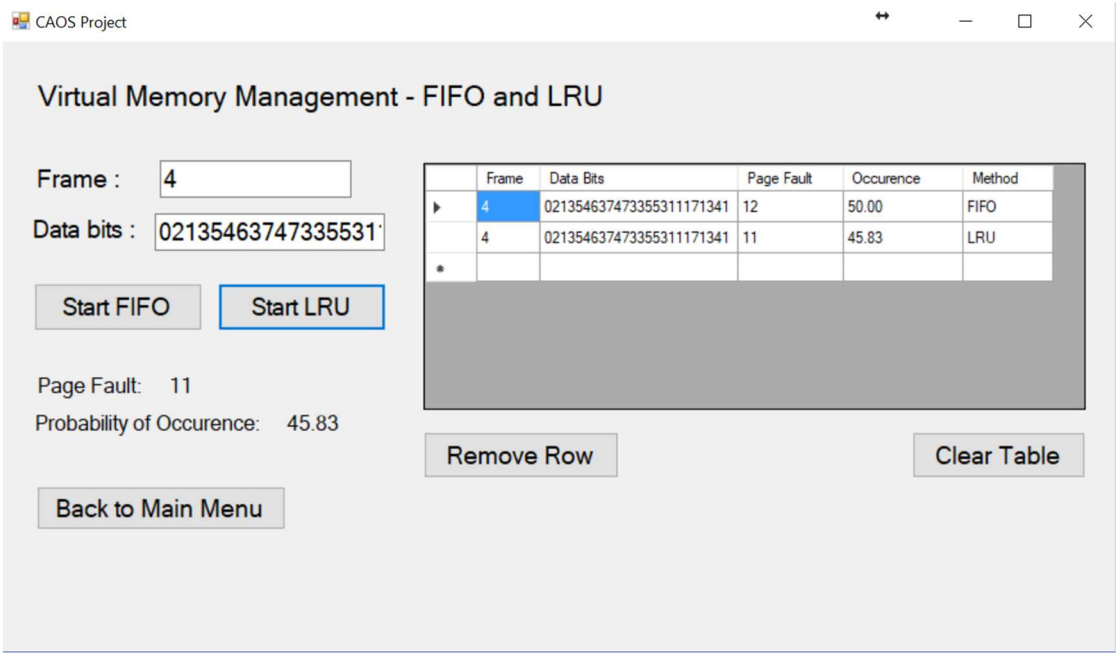
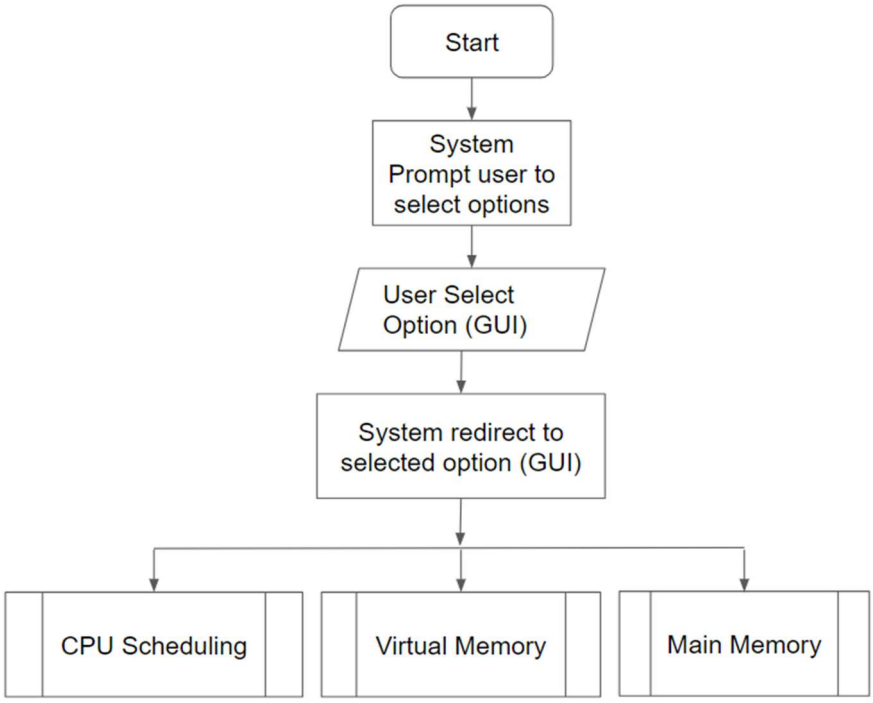


Figure 11: Virtual Memory Management - FIFO & LRU Result GUI

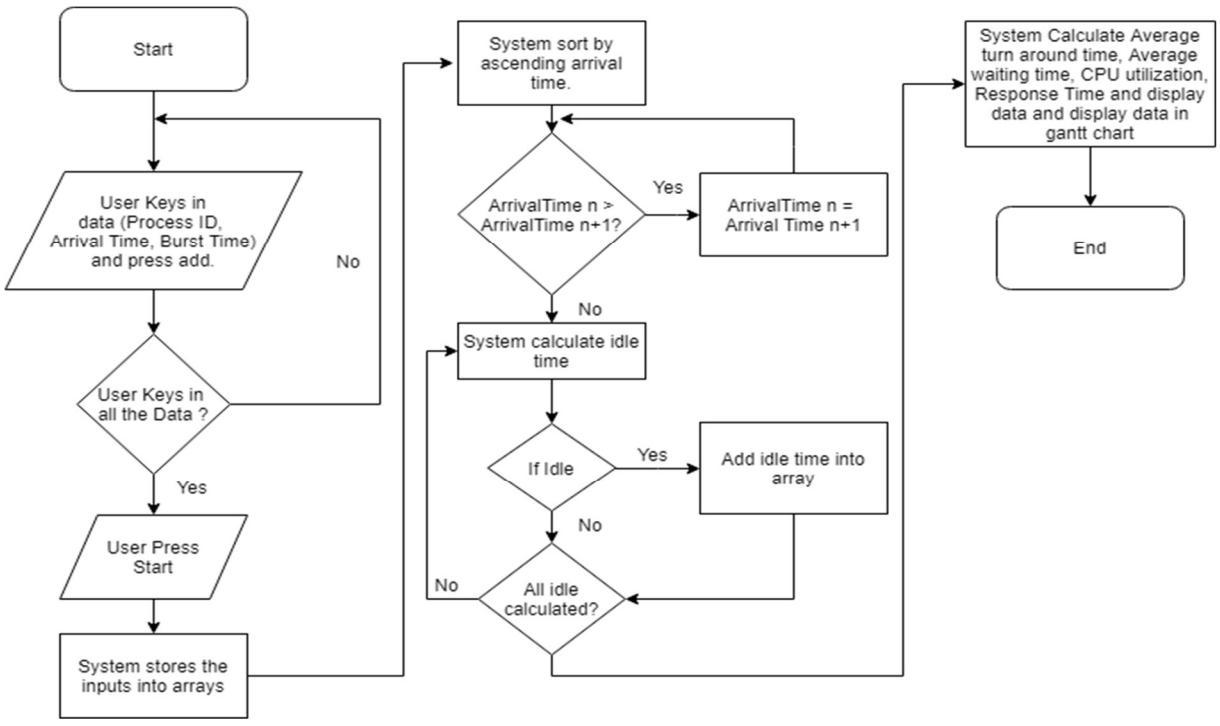
After the user had entered the value and pressed the button. It calculated the page fault and probability of occurrence and display on the GUI screen. The table will display all the user history.

3.2. Flowchart

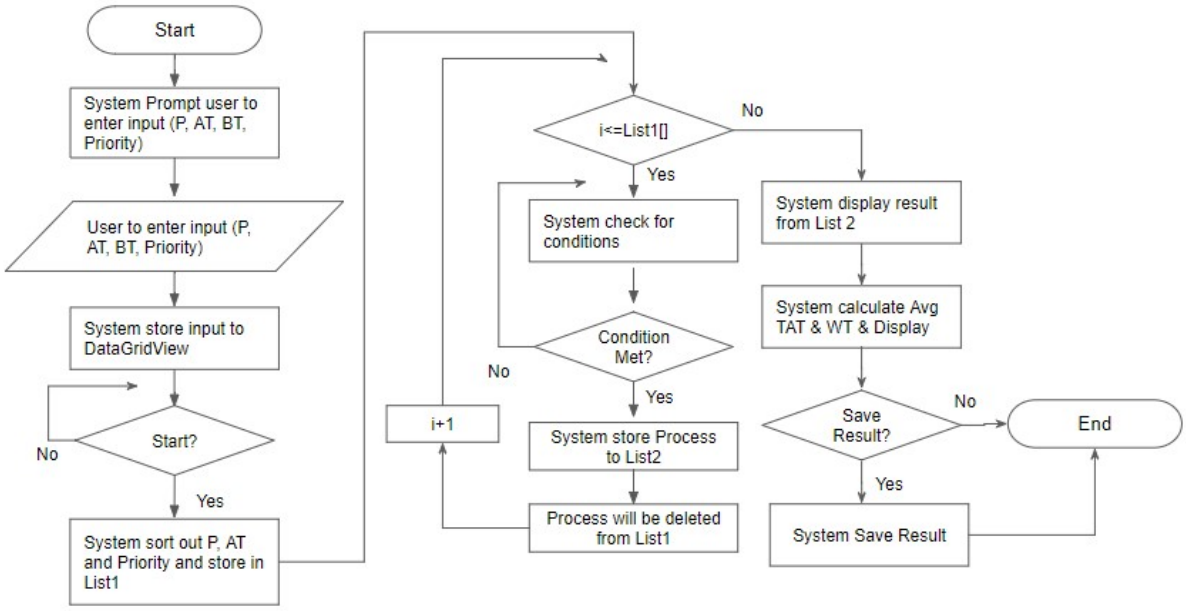
Main Page



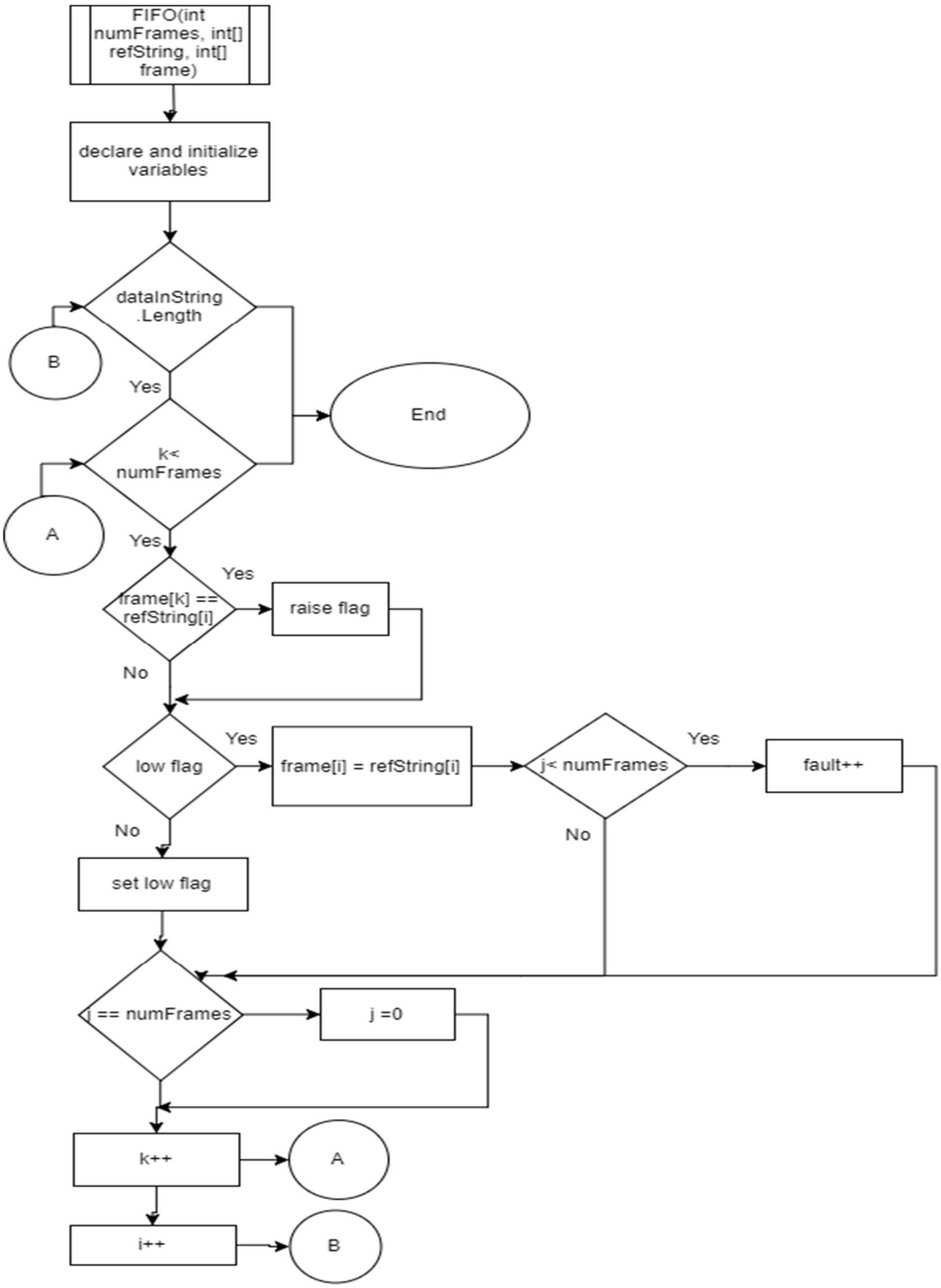
CPU Scheduling - FCFS Algorithm



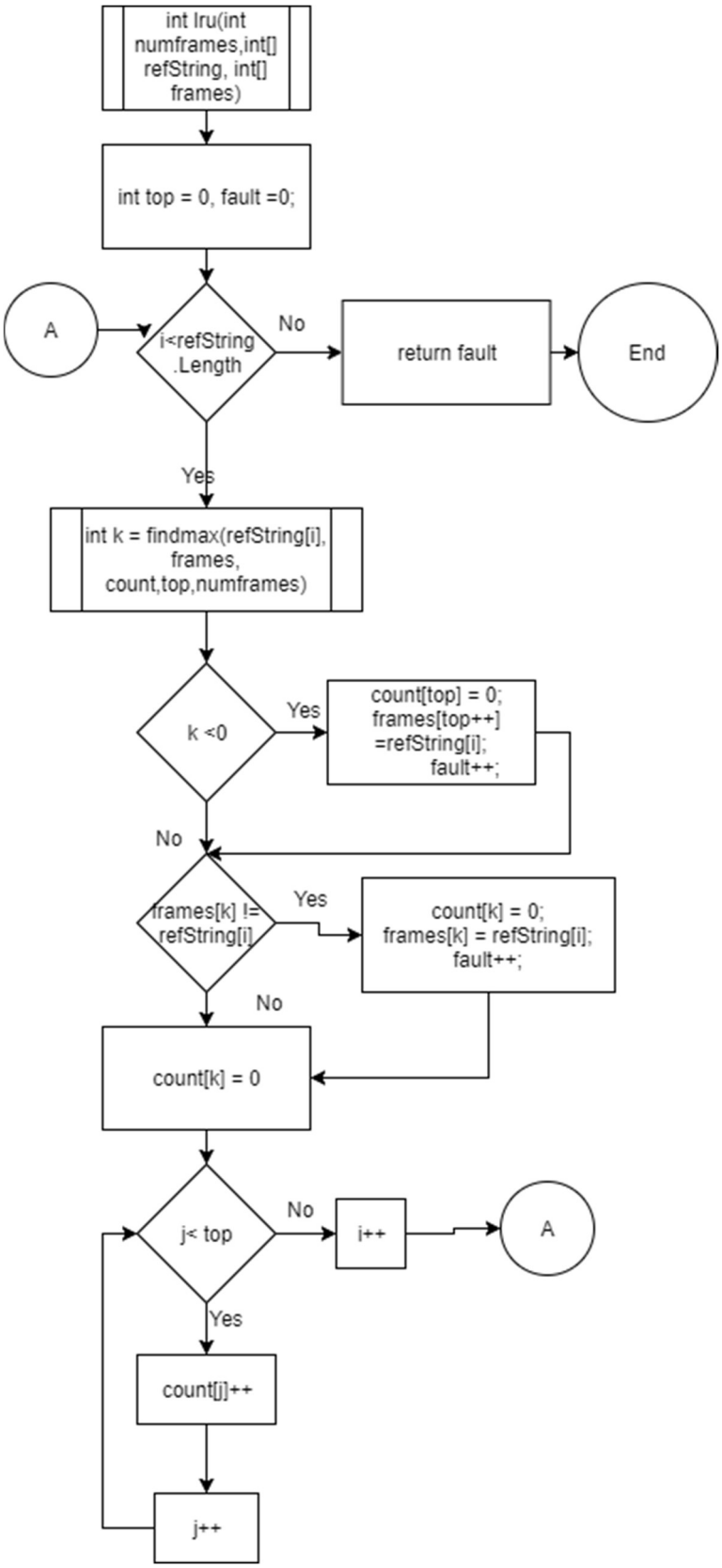
CPU Scheduling - Non-Preemptive Priority Algorithm



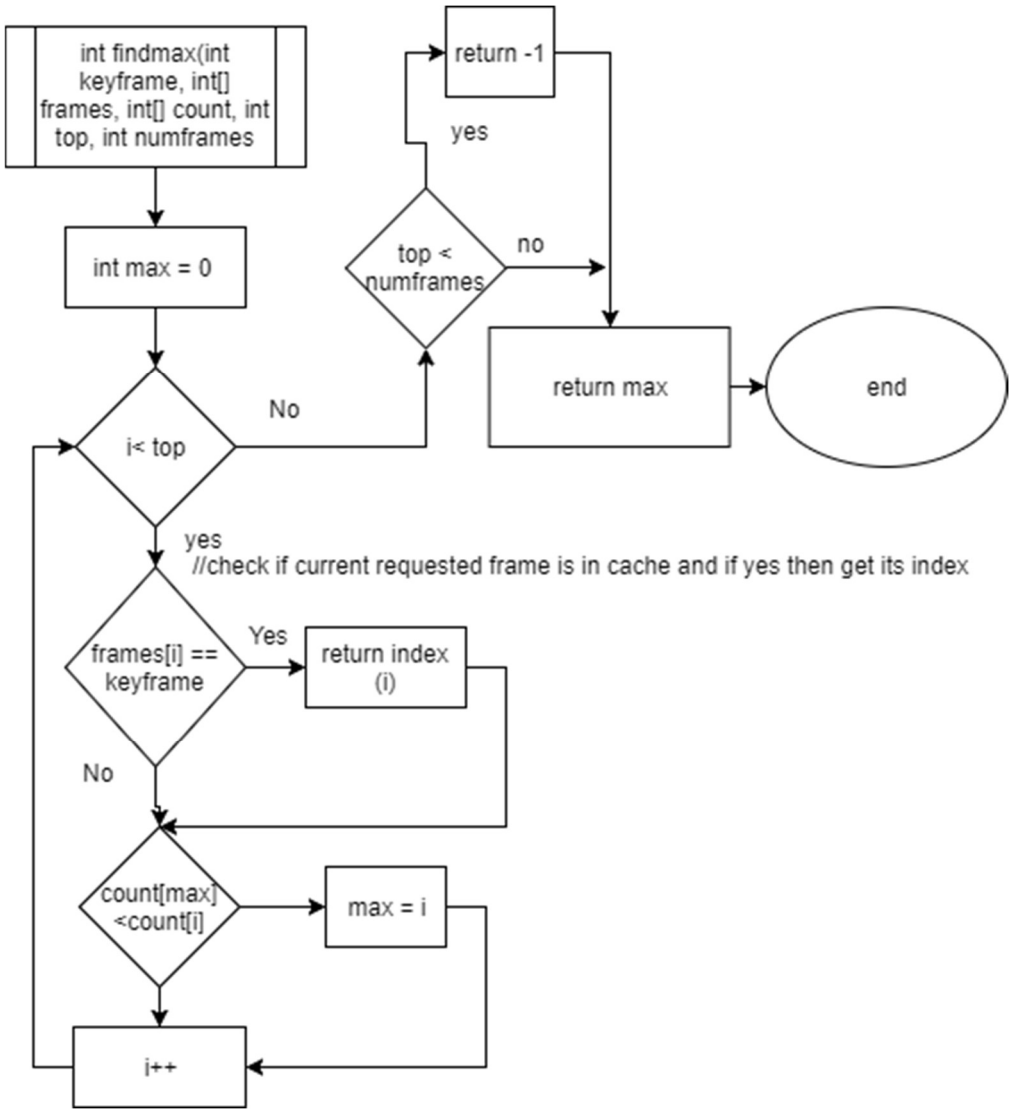
Virtual Memory - FIFO



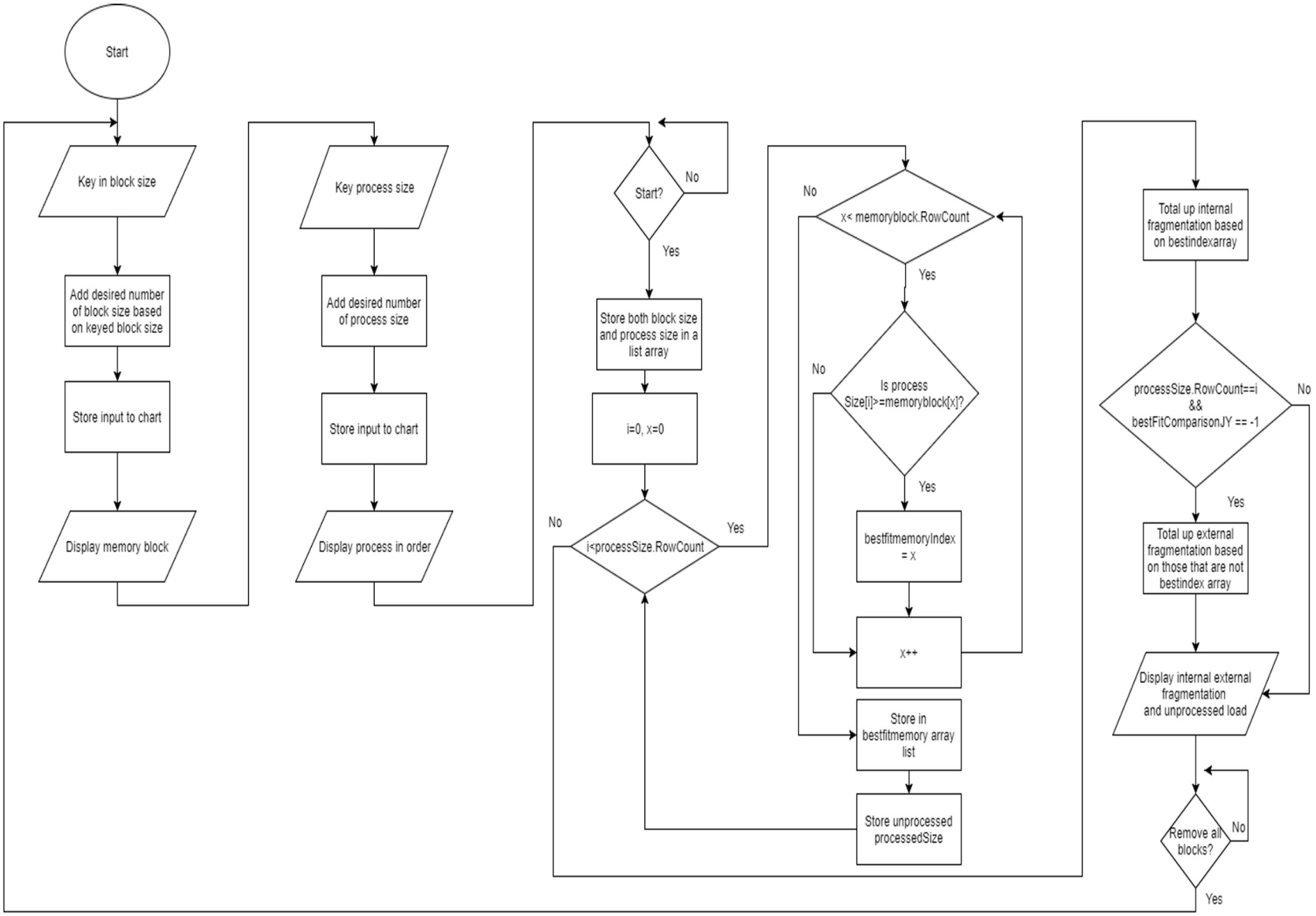
Virtual Memory – LRU



Virtual Memory – LRU findmax



Main Memory - Best Fit



4. Group Reflection Journal

Temasek Polytechnic
School of Engineering

PROBLEM-BASED LEARNING

Subject : CAOS
Problem Name : _____
Class : PE01
PBL Team : _____
Date : _____

GROUP REFLECTION
(part of group deliverable)

This is a journal which is to be completed by your team towards the end of your project development. It is to be submitted together with your project deliverable.

In this journal, your team is required to recall, discuss, consolidate and critically reflect on the team's performance, attitudes as well as your team's approach in developing your project deliverable.

A. TASK BREAKDOWN OF YOUR TEAM :

Member's Name	Task Completed
1. Lim Chong Yee	CPU Scheduling - FCFS Algorithm
2. Quek Kang Jie	CPU Scheduling - Non Preemptive Priority Algorithm
3. Tan Chin Siang	Virtual Memory - FIFO & LRU
4. Jay Yap Zong Hua	Main Memory - Best Fit

B. EVALUATION OF YOUR TEAM'S DELIVERABLE:

a) Has the deliverable met the minimum project specifications as stipulated by your facilitator? If no, which specification is not met?

Yes, we have met the minimum specification of CPU Scheduling First Come First Served (FCFS), Non-Preemptive Priority (NPP) and Main Memory Best Fit we also have add in Virtual Memory to calculate First In First Out FIFO) and Least Recently Used (LRU) Algorithm.

b) What factors could have contributed to your team's ability/inability to meet the minimum specifications of the project?

With the knowledge that was shared in class through peer teaching, online researching and programming knowledge for the past few years in Temasek Polytechnic has allowed us to easily meet the minimum specification of the project.

C. EVALUATION OF YOUR TEAM'S ABILITY IN SOLVING TECHNICAL DIFFICULTIES:

(a) What were some of the technical difficulties your team had when developing the project? Did your team manage to resolve these difficulties? If yes, proceed to (b).. If no, proceed to (c).

We met many difficulties when creating the application like attempting to display data into more presentable form such as Gantt Chart, unsure of how to start the project and many more. But we are able to resolve these difficulties.

(b) If yes, how did you all do it? How can you approach be further improved?

With the help of online research such as referring to the .NET Library or used stackoverflow for examples. Therefore, we are able to put together a working Gantt Chart and understand how the process is being executed.

(c) If not, why was your team unable to resolve them? How can your team prevent this problem from happening in future?

-

D. EVALUATION OF YOUR TEAM'S ABILITY IN SOLVING PROCESS DIFFICULTIES:
(a) What were some of the process-related difficulties your team had when developing the project? Did your team manage to resolve these difficulties? If yes, proceed to (b).. If no, proceed to (c). Due to the volatility of user input the program must be able to detect wrong inputs from the user. This is to prevent the program from crashing due to the wrong format or wrong input being entered. However, we are able to resolve the difficulties. (b) If yes, how did you all do it? How can you approach be further improved? By having code to check errors before allowing the data to be calculated. This prevents the user from proceeding when a wrong input is detected. (c) If not, why was your team unable to resolve them? How can your team prevent this problem from happening in future? -

5. Conclusion

Overall the project was done with not much difficulty as we are equipped with the skills of problem based learning from the past few years in Temasek Polytechnic and SIP. As this is our third year, having a problem based learning project everyone in the group knows what was expected and required when doing the project.

Therefore, our team has successfully met the requirement given to us in the Project Description and are able to come up with the application using C# Window Form Application in Microsoft Visual Studio 2015 and execute the application as intended. The project and application will be demonstrated to Mr Teo Kok Keong at 12 February 2018 during CAOS Lesson.