**1. Introduction**

This objective of this project is to simulate the process creation of an operating system using CPU scheduling, main and virtual management algorithms using C# Windows Forms Application.

1.1 Task Allocation

|  |  |
| --- | --- |
| **Task** | **Action by** |
| GUI Design & Integration | Chuan Jun |
| CPU Scheduling | Chee Xian |
| Main Memory Management | Jing Jun |
| Virtual Memory Management & Testing/Debugging | Marcus Chan |

**2. Theoretical Background**

2.1 CPU Scheduling

We have implemented the following CPU Scheduling algorithms:

* First Come First Serve
* Priority

Using the process’ known arrival time, burst time and priority, we implemented the algorithms to schedule the processes and calculate the turnaround time and waiting time results.

2.2 Main Memory Management

We have implemented the follow Main Memory Management algorithms:

* First-Fit
* Best-Fit
* Worst-Fit

Using the known sizes of the processes and partitioned memory blocks, we implemented the algorithms to allocate memory and calculate the internal and external fragmentation results.

2.3 Virtual Memory Management

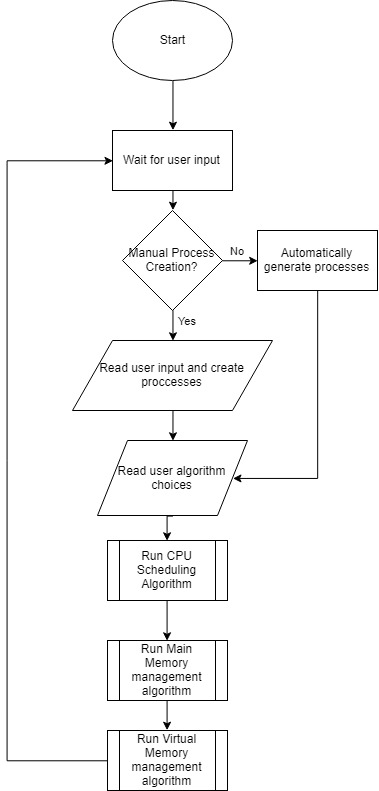
We have implemented the following Main Memory Management algorithms:

* First-in-First-out
* Least Recently Used

Processes that could not be allocated into main memory are separated into pages, and these pages will be queued and placed into memory frames using the Virtual Memory Management algorithms.

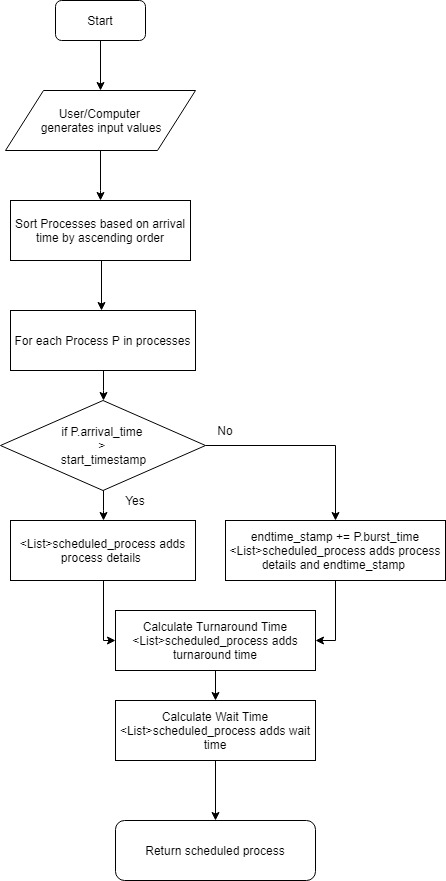
**3. Software Design**

3.1 Flow Chart

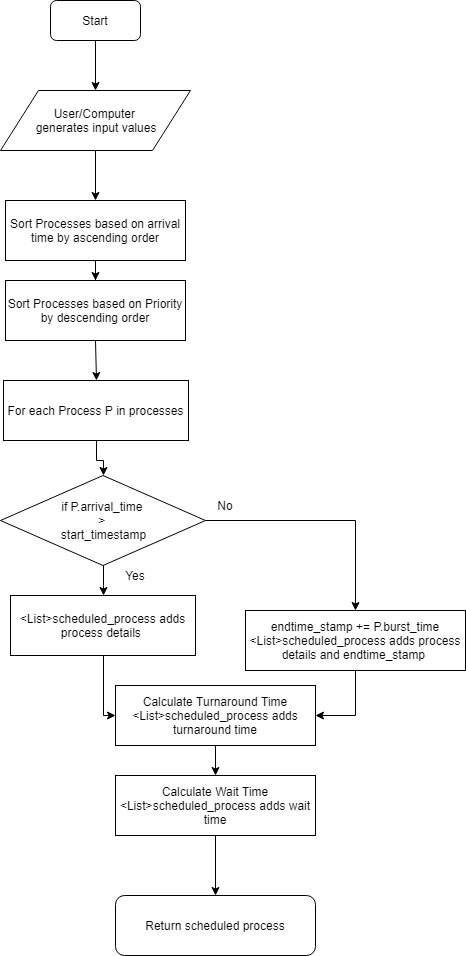


Flowchart 1: General Flow Chart

3.1.1 CPU Scheduling

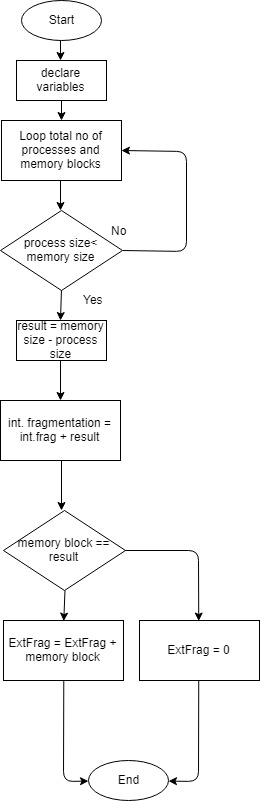


Flowchart 2: First-Come-First-Serve CPU Scheduling Flow Chart

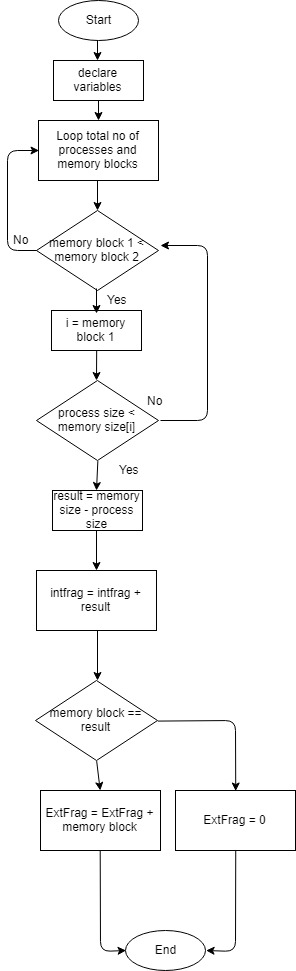


Flowchart 3: Non Pre-emptive Priority CPU Scheduling Flow Chart

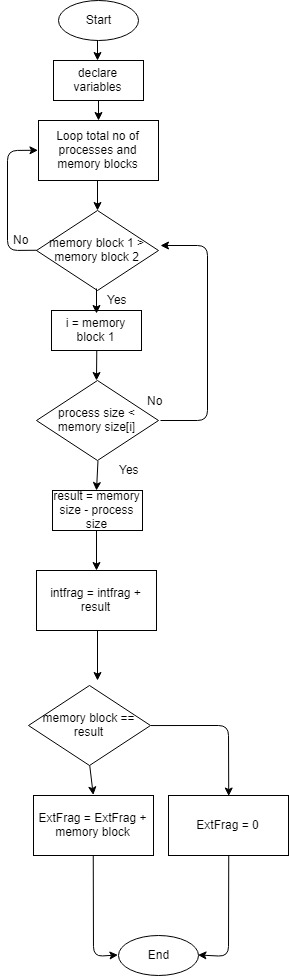
3.1.2 Main Memory Management



Flowchart 4: First Fit Main Memory Management Flow Chart

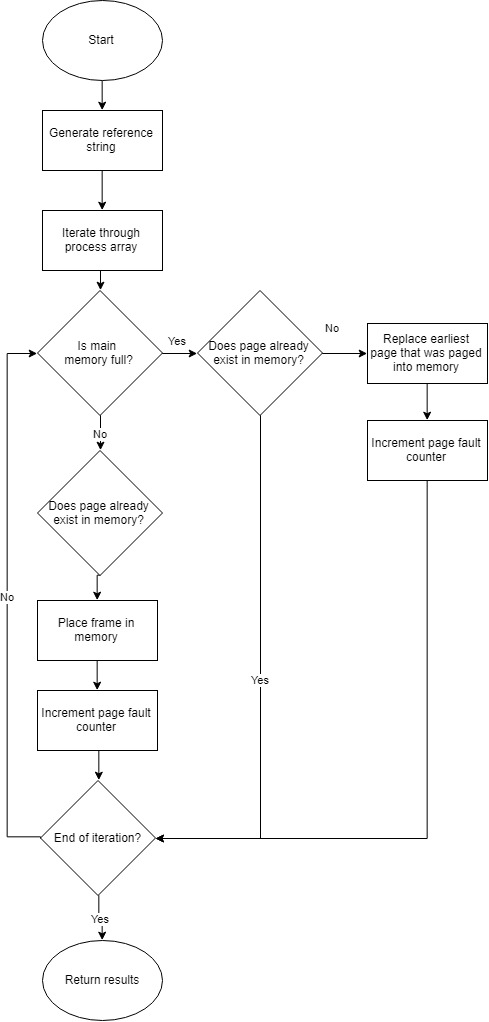


Flowchart 5: Best Fit Main Memory Management Flow Chart

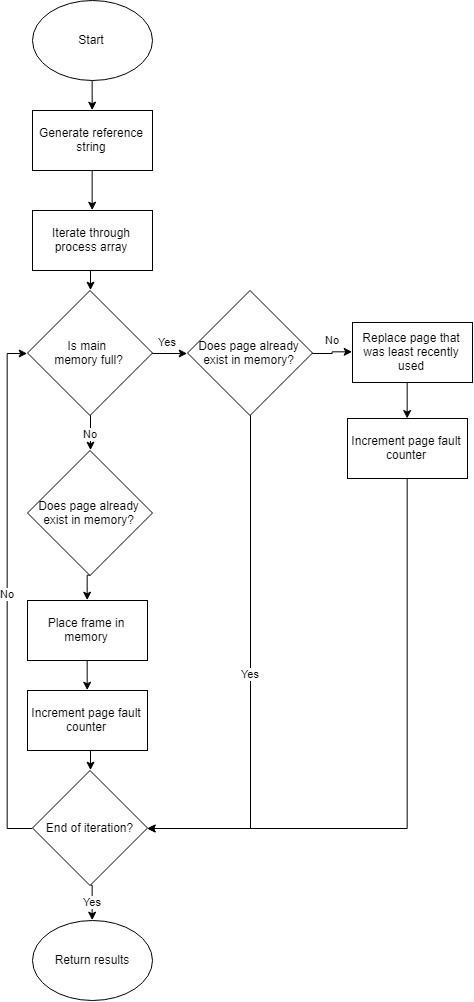


Flowchart 6: Worst Fit Main Memory Management Flow Chart

3.1.3 Main Memory Management

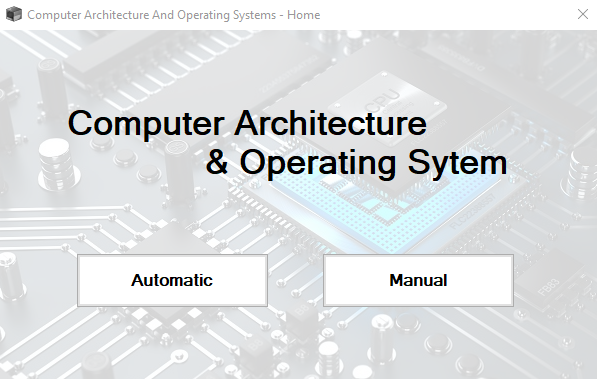


Flowchart 7: First-in-First-out Virtual Memory Management Flow Chart



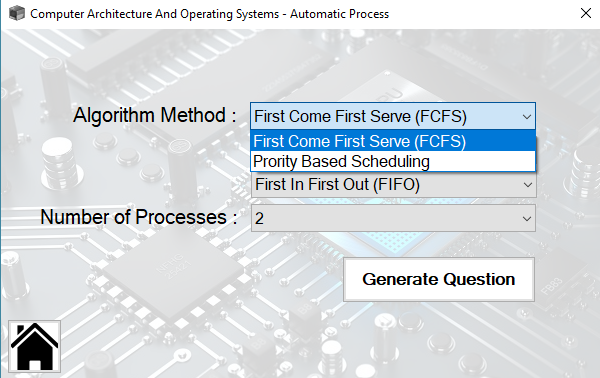
Flowchart 8: Least Recently Used Virtual Memory Management Flow Chart

3.2 Flow Chart



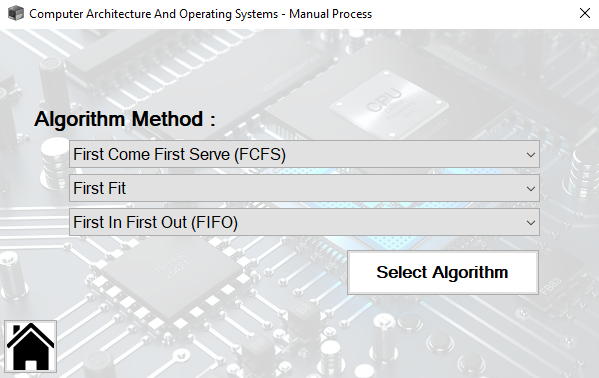
**Homepage –**

User will be able to select either Manual Operation or Automatic Operation.



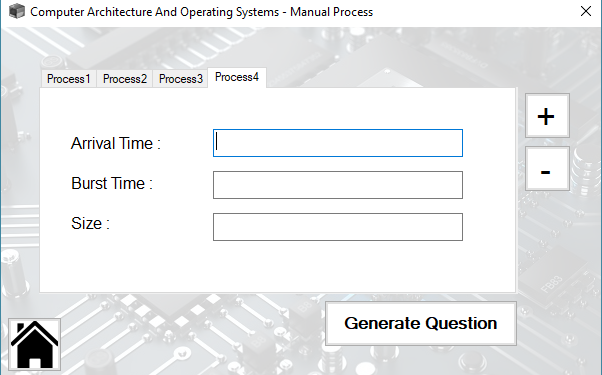
**Automatic Process –**

User will be able to select their Algorithm Methods for CPU Scheduling, Main Memory and Virtual Memory.



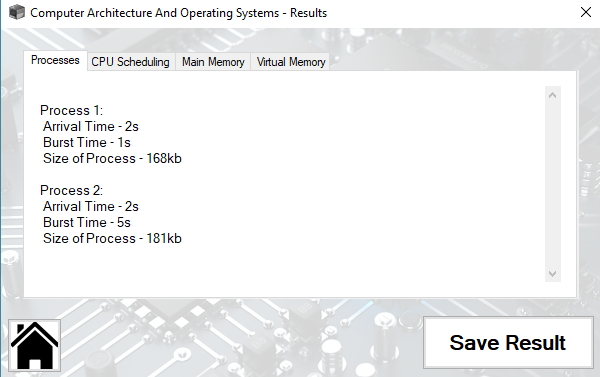
**Manual Process Part 1 –**

Before inputting the Question Parameters, Users will be required to select the algorithms.



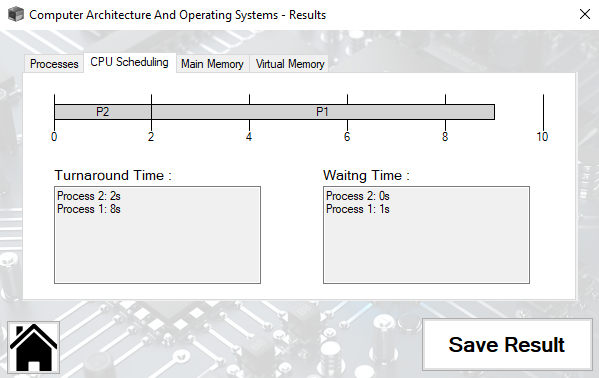
**Manual Process Part 2 –**

Here, User will be required to input the parameters for each Process, with a minimum of 2 Process to 8 Processes.



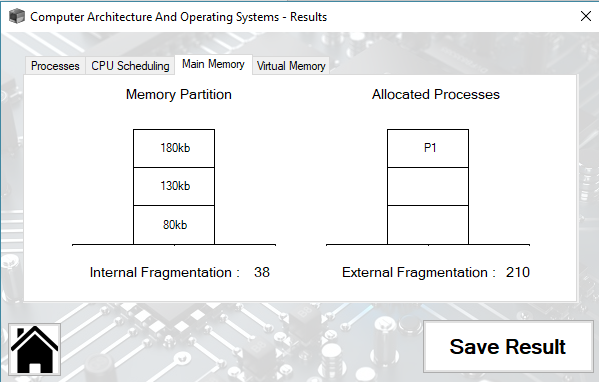
**Result Page [Process Parameters] –**

This page consist of all the parameters for each Process.



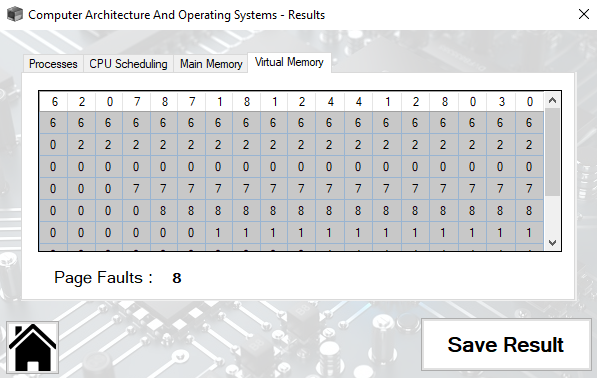
**Result Page [CPU Scheduling] –**

This page shows the generated results for CPU Scheduling, including the Gantt Chart.



**Result Page [Main Memory] –**

This page shows the Partition Results and the total Fragmentation caused.



**Result Page [Virtual Memory] –**

This page will only show a table when there is a Page Fault

4. Group Reflection Journal

|  |  |
| --- | --- |
| **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?  The minimum deliverable has been met.  b) What factors could have contributed to your team’s ability/inability to meet the minimum specifications of the project?  Mutual understanding between the team and cooperation. | |
| **C. EVALUATION OF YOUR TEAM’S ABILITY IN SOLVING TECHNICAL DIFFICULTIES :** |
| 1. 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).   Lack of programming structure and documentation in some groupmates work increases the difficulty during integration. Difficulties were resolved.   1. If yes, how did you all do it? How can your approach be further improved?   Early completion gave us ample time to approach issues if any. Clear set programming structure should be set before any of us starts to develop the algorithms. |
| **D. EVALUATION OF YOUR TEAM’S ABILITY IN SOLVING PROCESS DIFFICULTIES:** |
| 1. 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).   Miscommunication. Yes.   1. If yes, how did you all do it? How can your approach be further improved?   Miscommunication was due to a poor communication channel such as whatsapp. It was better when we communicate verbally face to face. So more face-to-face communication in the future. |

**5. Conclusion**

In conclusion, our team has successfully developed an interactive simulation of process creation in an operating system, using multiple management algorithms integrated wholly into a C# Windows Form Application program.

With the cooperation of all team members, our project is completed and functional.