THE OPEN UNIVERSITY OF SRI LANKA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
BACHELOR OF TECHNOLOGY HONOURS IN ENGINEERING
BACHELOR OF SOFTWARE ENGINEERING HONOURS

EEX5563/EEX5564 – COMPUTER ARCHITECTURE AND OPERATING SYSTEMS
Academic Year – 2023/2024

Mini Project Report should be uploaded to the given link in Moodle on or before the given due date. Hard copies will not be evaluated.

<div align="center">Due Date :14/12/2024</div>

Consider your registration number as the **dividend** and the number **6** as the **divisor**. Perform a division operation and calculate the **remainder**.

Refer to the table that lists different memory management techniques used for dynamic memory allocation. Based on the **remainder** obtained above, select the corresponding algorithm from the table.

Once you have identified the algorithm, design and implement a module to simulate its functionality. The module should accurately reflect how the algorithm allocates and manages memory dynamically. Include necessary documentation and comments in your implementation to ensure clarity and ease of understanding for future reference.

> For example, if your registration number is **120268730**, dividing it by **6** gives a remainder of **2**. In this case, the selected algorithm is the **Best Fit** algorithm. Your task is to write a program that simulates how the Best Fit algorithm works in dynamic memory allocation.

Remainder	Algorithm	Speed	Fragmentation	Suitability
0	**First Fit**	Fast	Moderate	Simple and quick allocation.
1	**Next Fit**	Moderate	High	Prevents repeated scans.
2	**Best Fit**	Slow	Low	Minimizes wasted space in partitions.
3	**Worst Fit**	Moderate	Moderate to High	Good for creating larger free blocks.
4	**Buddy System**	Moderate	Low	Ideal for systems with varying demands.
5	**Quick Fit**	Very Fast	Low to Moderate	Suitable for fixed-size frequent allocations.

You are expected to produce a report including the following sections;

1. Title Page
2. Table of Contents
3. List of Figures and Tables (if applicable)
4. Introduction
5. Requirements, Assumptions and justifications for the assumptions and/or Specifications
6. System Design for the Proposed Solution (Overview of the software architecture, design

patterns, data structures, and algorithms used in the project if any.)

7. Implementation (Description of the software development process, programming languages, frameworks, tools, and technologies employed.)
8. User Interface (UI) Design (if applicable)
9. Functionality and Features
10. Code Structure and Documentation
11. GitHub Repository(Include your project and allow the public to access it. The link should be provided here for the examiner to evaluate)
12. Testing Results
13. Deployment and Installation (if applicable)
14. Conclusion
15. Future Enhancements(Suggestions for future improvements or additional features that could be implemented)
16. References
17. Appendix(Include the link to self reflection video of the project implementation)