**Micro Project Report**

**On**

**Memory Management Simulation**

**Diploma Computer Engineering**

**Semester 4**

**(Advance Object-Oriented Programming – 4340701)**

|  |  |  |
| --- | --- | --- |
| **Group Members** | | |
| **Sr. No.** | **Enrollment No.** | **Student Name** |
| **1** | **226090307001** | **Aditya Pithva** |

**Guided By: -**

**Mr. S. R. Bhalgama**

**Lecturer, CE Department,**

**C. U. Shah Polytechnic, Surendranagar**

**Index**

|  |  |
| --- | --- |
| **Sr. No.** | **Topic Name** |
| 1 | Introduction to Project |
| 2 | Functional Requirement of Project |
| 3 | Applications |
| 4 | Code |
| 5 | Screenshots |
| 6 | References |

**Introduction to Project**

In the realm of computer science, efficient memory management plays a pivotal role in ensuring smooth system operation. The Memory Management Unit (MMU) acts as the central control unit, allocating and overseeing memory usage for various processes. This project delves into simulating an MMU, implementing key memory allocation algorithms, and exploring techniques to optimize memory utilization.

The core objective of this project is to provide a comprehensive simulation environment for understanding memory management concepts and evaluating different allocation strategies. It achieves this by offering functionalities to:

* **Simulation Memory Allocation**:- The project implements three widely used memory allocation algorithm:First Fit,Best Fit and Worst Fit. Users can choose the designed algorithm to observe its impact on memory allocation decisions.
* **Process Memory Management:-** The simulationallow users to define pre-running processes with specific memory requirements.Additionally, it enables users to introduce new processes dynamically, simulated real-world scenarios.
* **Memory Fragmentation Analysis:**- Fragmentation occurs when free memory compaction functionality. This process rearranges allocated memory blocks,consolidating free memory into contiguous chuncks, thereby improving allocation efficiency.
* **Process Termination and Memory Release:**- The simulation mimics process termination by allowing users to specify a process ID for memory deallocation. This function realistically portrays memory management tasks during process lifecycle.

This project is implemented using Java programming language and presents a user-friendly command-line interface for interaction. Users can configure RAM size, define pre-existing processes, and select the preferred memory allocation algorithm. The interface also provides options to execute memory compaction, release memory for terminated processes, and analyze memory fragmentation.

This project serves as a valuable learning tool for students, educators, and professionals interested in computer systems and memory management strategies. It allows for experimentation and analysis, fostering a comprehensive understanding of memory management concepts and their practical applications.

**Functional Requirement**

**Applications**

**Code**

|  |
| --- |
| *Put code here* |

**Screenshots of Working Project**

**References**