CPS3250 W03 – Group#666

Computer Operation System by Dr. Rashid Sangi

1. **Group Member:**

**Kaiyu Liu 1195058** liukai@kean.edu

**Fan Yu 1195235** yufa@kean.edu

**Yiyang Hu 1194116** huyiya@kean.edu

**Lei Xia 1195180 xial@kean.edu**

1. **Project Title:**

**Virtual Memory Explorer:**

**An Interactive Educational and Performance Analysis Platform**

1. **Short description:**

**The "Virtual Memory Explorer" is an innovative educational platform designed to demystify the intricacies of virtual memory management for students and enthusiasts. This interactive tool will elucidate key concepts such as memory allocation, paging, and swapping, through dynamic visualizations that bridge the gap between theoretical understanding and practical application.**

**Our platform will delve into the mechanics of how virtual memory leverages both Random Access Memory (RAM) and disk storage, orchestrating them to optimize system resources. By simulating the inner workings of paging and various page replacement algorithms, users will gain hands-on experience with the decision-making process of the memory management system.**

**We are dedicated to crafting a user-centric experience, offering a dual-interface system where users can navigate through a command-line interface (CLI) for precise control or engage with a graphical user interface (GUI) for a more intuitive learning journey. The GUI is designed to provide real-time, visual feedback on virtual memory operations, fostering an environment where users can experiment with and observe the consequences of different memory management strategies.**

1. **Which program language:**

**The project will primarily use Java for implementation, and if needed, certain sections may incorporate C.**

**For the front-end graphical user interface (GUI), we plan to utilize VUE as the main framework, potentially incorporating JavaScript as well.**

1. **How about interface:**

**The following is a rough draft of our GUI, intending to visualize what’s happening in the virtual memory management process.**

图片包含 文本

描述已自动生成

**The following is an example input and output of the CLI:**

**Virtual Memory Explorer CLI**

**1. Load Page**

**2. Show Page Table**

**3. Exit**

**Input: Select an option: 1**

**Input: Enter virtual page number to load: 5**

**Return:**

**Page 5 is not in RAM.**

**Loading Page 5 into RAM... Done.**

**Page Table:**

|  |  |
| --- | --- |
| **Virtual Page** | **Physical RAM** |
| **5** | **Frame 2** |
|  |  |
|  |  |

**Input: Select an option: 3**

**Return:**

**Exiting Virtual Memory Explorer CLI. Thank you!**

1. **Overall Structure**

**Focusing on a conceptual approach of this Virtual Memory Management project, the following is the overall structure that we plan to implement:**

**Simulation Environment Setup:**

* + **Physical Memory Representation: Simulate RAM with a predefined size.**
  + **Virtual Memory Space: Create a larger virtual address space.**
  + **Page Table Structure: Implement a basic page table for address mapping.**
  + **Swap Space: Simulate a disk space for paging.**

**Memory Management Algorithms:**

* **FIFO (First-In-First-Out)**
* **LRU (Least Recently Used)**
* **Optimal Page Replacement**
* **Custom or less common algorithms for comparison.**

1. **Project Goal and Objectives:**

* **Goal 1: Enhance Educational Understanding of Virtual Memory Management**
  1. **Objective 1.1: Develop a comprehensive educational tool that demystifies the complexities of virtual memory management, including memory allocation, paging, and swapping processes.**
  2. **Objective 1.2: Provide interactive learning experiences through a user-friendly interface that caters to different learning styles, incorporating both Command Line Interface (CLI) and Graphical User Interface (GUI) options.**
* **Goal 2: Simulate Real-world Virtual Memory Operations**
  1. **Objective 2.1: Implement a simulation environment that visualizes the dynamic operations of paging and page replacement algorithms in real-time.**
  2. **Objective 2.2: Allow users to interact with the simulation, adjusting parameters to see the impact on virtual memory management and system performance.**
* **Goal 3: Offer In-depth Performance Analysis**

1. **Objective 3.1: Create performance analysis tools within the platform to demonstrate the efficiency of various page replacement algorithms.**
2. **Objective 3.2: Equip the platform with logging and statistical display features to help users analyze and understand the behavior of the virtual memory system under different workloads.**

**First Follow-up:**

**1. Simulation Environment Setup: Standardized Data Formats**

**RAM Representation: Define a standard data structure for RAM. For example, an array or list where each element represents a memory frame.**

**Virtual Memory Space: Agree on a virtual address format, possibly as integers or a specific string format.**

**Page Table Structure: Decide on the structure of the page table entries, including fields like virtual page number, frame number, and status flags (valid/invalid, modified, etc.).**

**Swap Space: Establish a format for representing swap space, similar to RAM, possibly with additional metadata for swapped-out pages.**

**2. Memory Management Algorithms: Interface Specifications**

**Input Parameters: Define the inputs required for each algorithm, like the current state of the page table, the sequence of page requests, etc.**

**Output Format: Standardize how the algorithms will output their results, such as changes to the page table, page faults, and swap operations.**

**Function Signatures: Ensure consistency in the way functions are declared and called across different algorithms for ease of integration.**

**3. GUI Development: Consistent Data Handling and Display**

**Data Input: Ensure the GUI accepts data in the standardized formats set for the RAM, virtual memory, and page tables.**

**User Interaction: Define how user actions in the GUI translate to inputs for the simulation environment and algorithms. For instance, a button press in the GUI could generate a specific page request.**

**Data Visualization: Standardize the way data is visualized in the GUI, such as color codes for different memory states, consistent table formats, etc.**

**4. CLI Development: Command and Response Standards**

**Command Format: Establish a clear and consistent format for commands, including the syntax and valid options.**

**Feedback and Output: Standardize the CLI feedback for different operations, ensuring it aligns with the GUI's feedback mechanism.**

**Error Handling: Define a common format for reporting errors or invalid inputs in the CLI.**