capstone project

Dynamic storage management

CSA0488-OPERATING SYSTEM

FACULTY NAME: DR. R. HEMAVATHI

GROUP MEMBERS:

Keerthi .C
(192211419)

POOJITHA.J
(192211829)

Navyasree
(192211)

1. PROBLEM STATEMENT

- Develop a dynamic memory allocation system that optimizes memory usage in a computer program, ensuring efficient utilization and preventing memory leaks or fragmentation.
- Implement a secure and reliable memory management mechanism to protect against unauthorized access and ensure compliance with industry security standards.
- Integrate real-time memory usage updates to provide developers with accurate information, reducing the risk of memory-related issues and improving overall program stability.
- Enhance developer communication by implementing features such as memory allocation confirmations, warnings about high memory usage, and the ability to receive real-time updates on critical memory events via logs or notifications.
- Create a visually appealing and interactive memory allocation visualization, allowing developers to view and manage memory allocations, considering factors like allocation size, deallocation, and overall program performance.
- Implement a waitlist feature to efficiently manage memory requests and notify developers when memory becomes available, ensuring a smooth execution of the program and minimizing resource contention.
- Provide developers with a user-friendly backend system to monitor memory usage, track allocation patterns, and streamline the overall memory management experience for both developers and system administrators.

2. PROPOSED DESIGN WORK

Identifying the Key Components

- Develop an interface for developers to allocate memory dynamically, specifying size, type, and any special requirements.
- Implement algorithms for efficient memory allocation, considering factors like allocation size, deallocation patterns, and time complexity.
- Display available memory blocks with filters like size, type, and usage. Allow developers to deallocate memory within a specified timeframe and handle errors gracefully.

Functionality

- Intuitive interface for easy memory allocation and deallocation.
- Interactive memory visualization for developers to understand memory usage patterns.
- Secure memory allocation options for safe and controlled transactions.
- Convenient tools for developers to view, modify, or deallocate memory blocks.
- Accessibility features and multilingual support for a diverse developer community.
-

Architectural Design

- Define the overall structure of memory allocation, including heap, stack, and dynamic memory areas.
- Design the memory layout to comply with security standards, providing features such as memory protection, access control, and encryption.
- Balance efficiency with security, ensuring optimal memory usage while maintaining program integrity.
-

3. UI DESIGN

Layout Design

- The dashboard serves as the entry point for developers and provides an overview of the memory allocation system.
- After entering their memory allocation preferences, developers are directed to the visualization page displaying memory usage.
- Provide an allocation form or interface where developers can specify memory size, type, and any special requirements.
- After allocating memory, developers can proceed to the confirmation and monitoring stage.

Feasible Elements Used

- Implementing a user authentication system for secure access to memory allocation features.

- Integrating a system that updates memory usage in real-time based on allocations made.
- Providing developer support channels such as logs, notifications, or a helpdesk system to assist with memory-related inquiries.

Elements Positioning

- Place prominently at the top of the dashboard or visualization page.
- Display memory blocks in a clear and interactive format, allowing developers to easily understand allocation patterns.
- Include comprehensive information about memory blocks, including size, type, and usage.
- Position memory logs and notifications adjacent to the visualization for easy access.

Elements Function

- Allows developers to input memory allocation preferences such as size, type, and special requirements.
- Enables developers to visualize and monitor memory usage based on criteria such as size, type, and deallocation patterns.
- Provides options for developers to receive alerts or notifications about critical memory events.
- Each element plays a crucial role in facilitating the memory management process, providing developers with a seamless and efficient experience while optimizing memory usage.

CONCLUSION

In summary, the implementation of an efficient dynamic memory allocation system holds significant potential for enhancing program performance and stability. This system promises to streamline memory management processes, significantly improving developer satisfaction and program efficiency within the software development industry. By incorporating innovative features such as real-time memory visualization, secure allocation options, seamless communication, and data analytics, the dynamic memory allocation

system not only caters to the evolving needs of developers but also provides valuable insights for program optimization. Embracing such technology not only modernizes memory management but also contributes to creating a more dynamic and performance-centric software development environment.