

Design and Implementation of a Memory Management Simulator

By Niviti Sharma 23117093

Core Modules:

1. Memory Management (memory.hpp, memory.cpp)
 - Manages a contiguous memory region using a linked list of blocks
 - Supports allocation, deallocation, and coalescing of free blocks
 - Tracks internal/external fragmentation
2. Allocators (allocator.hpp, first_fit.cpp, best_fit.cpp, worst_fit.cpp)
 - First-fit: Allocates first suitable block
 - Best-fit: Allocates smallest suitable block
 - Worst-fit: Allocates largest suitable block
3. Cache Simulator (cache.hpp, cache.cpp)
 - Configurable cache hierarchy (L1, L2, L3)
 - Replacement policies: FIFO, LRU, LFU
 - Tracks hit/miss ratios and cache performance
4. Statistics (stats.hpp, stats.cpp)
 - Reports memory utilization, fragmentation, allocation success rates
 - Cache hit/miss analysis

Features:

Memory Management:

- Dynamic block splitting/merging
- Fragmentation analysis (internal/external)
- Memory utilization tracking
- Visual memory dump (dump memory)

Cache Simulation:

- Multi-level cache support (up to L3)
- Configurable size, block size, associativity
- Replacement policy comparison
- Hit/miss tracking per level

Technical Implementation

Data Structures:

- Memory: Linked list of Block nodes
- Cache: Set-associative arrays with per-set replacement lists
- Allocators: Strategy pattern with lambda-based block selection

Design Patterns:

- Strategy Pattern: Different allocation algorithms
- Factory Pattern: Allocator creation
- Observer Pattern: Statistics collection

Building:

Make clean && make