contiguous_memory

Introduction

This is a simple program that demonstrates how contiguous memory is allocated in C++.

Usage

run the following command:

```
make
```

type `help` to see the available commands.

Implementation

Your implementation seems to be a memory allocation simulation program in C++. Let's flesh out the Implementation section by providing a brief overview of the key functions and data structures used in the program:

Implementation

Data Structures:

1. **Block Structure (`block_t`):**
   - Represents a memory block.
   - Contains fields for start address, available space, process ID, and a pointer to the next block.

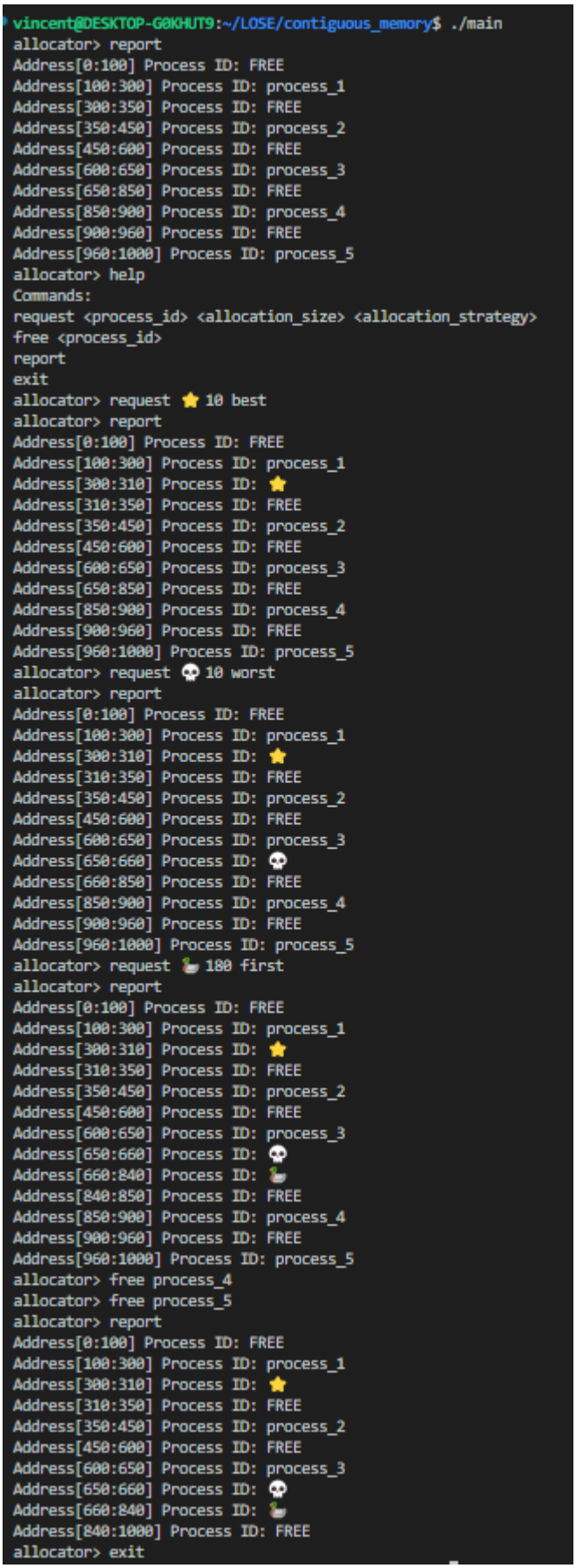
Functions:

1. `init_block`**:**
   - Initializes a memory block with the provided parameters.
2. **Memory Allocation Strategies:**
   - **First Fit (`find_first_fit`):**
     - Searches for the first available memory block that fits the allocation size.
   - **Best Fit (`find_best_fit`):**
     - Searches for the memory block that best fits the allocation size.
   - **Worst Fit (`find_worst_fit`):**
     - Searches for the memory block that worst fits the allocation size.
3. `allocate_memory`**:**
   - Allocates memory for a process in the specified block.

4. `request_block`:
    - Requests memory block allocation based on the specified allocation strategy.
5. **Memory Management:**
    - `merge_unused_adjacent_blocks`:
        - Merges adjacent free memory blocks into a single larger block.
    - `free_block`:
        - Frees the memory block associated with the given process ID.
6. `setup_memory_state`:
    - Sets up the initial state of the memory by dividing it into segments and initializing memory blocks.
7. `report_memory_usage`:
    - Reports the current memory usage, including allocated and free blocks.
8. `free_main_memory`:
    - Frees the allocated memory blocks.
9. `main`:
    - Entry point of the program.
    - Sets up the initial memory state, processes user commands, and manages memory allocation and deallocation.

This program provides a command-line interface for users to request memory allocation, free memory blocks, report memory usage, and exit the simulation. Additionally, it supports different allocation strategies such as First Fit, Best Fit, and Worst Fit.

Screenshots


vincent@DESKTOP-G0KHUT9:~/LOSE/contiguous_memory$ ./main
allocator> report
Address[0:100] Process ID: FREE
Address[100:300] Process ID: process_1
Address[300:350] Process ID: FREE
Address[350:450] Process ID: process_2
Address[450:600] Process ID: FREE
Address[600:650] Process ID: process_3
Address[650:850] Process ID: FREE
Address[850:900] Process ID: process_4
Address[900:960] Process ID: FREE
Address[960:1000] Process ID: process_5
allocator> help
Commands:
request <process_id> <allocation_size> <allocation_strategy>
free <process_id>
report
exit
allocator> request ⭐ 10 best
allocator> report
Address[0:100] Process ID: FREE
Address[100:300] Process ID: process_1
Address[300:310] Process ID: ⭐
Address[310:350] Process ID: FREE
Address[350:450] Process ID: process_2
Address[450:600] Process ID: FREE
Address[600:650] Process ID: process_3
Address[650:850] Process ID: FREE
Address[850:900] Process ID: process_4
Address[900:960] Process ID: FREE
Address[960:1000] Process ID: process_5
allocator> request 💀 10 worst
allocator> report
Address[0:100] Process ID: FREE
Address[100:300] Process ID: process_1
Address[300:310] Process ID: ⭐
Address[310:350] Process ID: FREE
Address[350:450] Process ID: process_2
Address[450:600] Process ID: FREE
Address[600:650] Process ID: process_3
Address[650:660] Process ID: 💀
Address[660:850] Process ID: FREE
Address[850:900] Process ID: process_4
Address[900:960] Process ID: FREE
Address[960:1000] Process ID: process_5
allocator> request 🦆 180 first
allocator> report
Address[0:100] Process ID: FREE
Address[100:300] Process ID: process_1
Address[300:310] Process ID: ⭐
Address[310:350] Process ID: FREE
Address[350:450] Process ID: process_2
Address[450:600] Process ID: FREE
Address[600:650] Process ID: process_3
Address[650:660] Process ID: 💀
Address[660:840] Process ID: 🦆
Address[840:850] Process ID: FREE
Address[850:900] Process ID: process_4
Address[900:960] Process ID: FREE
Address[960:1000] Process ID: process_5
allocator> free process_4
allocator> free process_5
allocator> report
Address[0:100] Process ID: FREE
Address[100:300] Process ID: process_1
Address[300:310] Process ID: ⭐
Address[310:350] Process ID: FREE
Address[350:450] Process ID: process_2
Address[450:600] Process ID: FREE
Address[600:650] Process ID: process_3
Address[650:660] Process ID: 💀
Address[660:840] Process ID: 🦆
Address[840:1000] Process ID: FREE
allocator> exit