

Assignment 3: Virtual Memory Management

Professor: Dalu Jacob

TA: Anshika Prajapati, Priyal Jain, Saket Kandoi

Deadline : 8th November 11:59 PM

Problem Statement

- You are required to implement your own dynamic memory allocation functions: `malloc_*` and `my_free` with the same calling conventions and signatures as the standard `malloc` and `free`. The implementation has to be done in C.
- Each call to a `malloc_*` should return a pointer to the **head of the allocated memory region**. This pointer will be passed as input to `my_free`.
- You must implement an internal **Free Space Manager (FSM)** responsible for handling allocation and deallocation requests.
- Implement a data structure for **free list** to manage available memory blocks.
- Implement the following allocation strategies (each strategy is of 2 marks):
 - (a) **First Fit**
 - (b) **Next Fit**
 - (c) **Best Fit**
 - (d) **Worst Fit**
 - (e) **Buddy Allocation**
- Parts (a) and (b) must be implemented so that any new memory allocation should take time in the order of \log in the number of available free segments. Part (c) and (d) can take linear time.
- Parts (a)-(d) will be called any number of times in any order in a single testcase. Part (e) will be tested separately.
- Use system calls such as `mmap`, `brk`, and `sbrk` for memory allocation. Refer to their respective `man` pages for details.
- Complete the functions provided in the template code and ensure that your implementation behaves identically to the standard memory management library when linked to a user program.
- Each strategy should clearly show how memory blocks are allocated and freed, along with corresponding free list updates.

Submission

- Submit a zip file containing a single header file named `EntryNommu.h` (e.g., `2021MT60986mmu.h`).
- Ensure that your code is well-commented, modular, and adheres to best coding practices for clarity and maintainability.
- All submissions will undergo plagiarism checks against past submissions and AI-generated content. Any form of plagiarism will result in a zero score.

Good luck, and happy coding!