Assignment-1

Memory Allocation++

Partners : **Bhavinkumar Patel (bp412) (Section-02)**

Aksharkumar Patel (ahp75) (Section-04)

- This Assignment is all about dynamic memory allocation. In this assignment implemented own malloc() and free() functions. The malloc() is giving a size of memory block with a given size. Here used an array of size 5000 to allocate a memory. The free() is checking that the memory is allocated in memory before free() is trying to free memory. This program is also preventing fragmentation issues.
- **my_malloc()** function has a struct pointer and pointer is pointing to the head of 5000 bytes of memory block. And after check the size and allocated if the allocation requires are fulfill and if requirement is not fulfill then will return null and not going to allocate a memory. If memory is more than the requirement then it will return an error message like a memory is exceeds the final limit.
- **my_free()** function is checking the required requirements to free the allocated memory and freeing nicely. If there is any kind of mistakes then will show an error message for example if memory is not allocated and trying to free it will print an error message like a cannot free the pointer that was not allocated. If trying to free allocation memory twice then shows an error message accordingly that.
- **memgrind.c –** In this created six different workloads to examine and exercise memory allocation. Each workloads are executed 100 times. Every time the execution time will be different for each workloads and all workloads time will be different.
  - **workload A ,** the execution average time for A is 630 microseconds.
  - **workload B,** the execution average time for B is 17 microscends.
  - **workload C,** have to keep track of total mallocs and total frees. Total mallocs and total free is 100000.
    - The execution time was 189 microseconds.
  - **workload D,** have to keep track of total mallocs and total frees and random allocation size between 1 and 64 bytes. By choosing 1 byte, total mallocs is 10000, successful free is 20000, and total frees is 20000.
    - The execution time is 684 microseconds.
  - **workload E**, trying to free pointer twice. We got an error saying cannot free pointer twice.
    - The execution time is 4 microseconds.(Note: can take any example for workload E)
  - **workload F,** trying to free pointer without allocation it. We got an error saying Cannot free pointer that was not allocated.(Note: can take any example for workload F)
    - The execution time is 4 microseconds.