CS307 HW3 MEMORY MANAGEMENT REPORT

Begüm Altunbaş 26824

We have 3 additional functions besides init() and dump_memory().

In my_malloc(): we create a node struct and give thread_id number and random size and lastly push it to our thread queue.

In server_function(): I check if there are enough space for threads request. In this loop if the queue is not empty, I pop a thread and check if I have available memory space for its request size, if I grant space for that thread, I write its starting index to a thread_message array. Starting index initially is 0 and it is incremented in thread function by adding the size of granted thread. If I decline the thread's request, we push -1 to thread message array. I also unblock the semaphore of appropriate thread id which is previously blocked in thread_function.

In thread_function(): First I lock the mutex so that each time only one thread enters this region. After that, I create a random memory size for a thread and call my_malloc to allocate its memory, id and push thread to shared queue. Then I block this thread because I need to get answer from server function in order to proceed to next step in threadfunction. After the server_fucntion unblocks the thread it continues from thread_function and then according to answer from server function it arranges the memory array. Meaning; if we grant memory for the thread, we change memory array's indexes to thread's id number for length of that thread's size. If we declined its request, then we print message saying there are not enough memory. Lastly, I unblock the mutex since I am done with this thread.

Init and dump memory functions are given to us. Init, initializes the mutexes semaphores, mutexes etc. Dumpy memory prints the final output of memory array.

In main (): I create a thread array and integer array (later to pass as parameter to threads). I use pthread_create function to create my threads using thread array, thread function and threadnums array as parameters. Before terminating and printing the memory and thread message I join the threads so that main function waits for threads to execute and then it can finish.