Yunus Delipınar 26617

PA4

In this assignment I have used simple mutex class because I thought that I didn't need advanced locking mechanisms such as semaphores. Even though mutex seems simple locking mechanism it is very efficient and useful and as I expected it worked great on my class. As a result of using mutex, I haven't had any errors and my program worked concurrently.

I used mutexes at the beginning of each critical member function of the class and I released the lock before the function returns. In below you can see the code psuedo code of my class with mutexes.

```
MyMalloc(int ID, int size)  //member function
Lock the mutex.
Find the first location to allocate some space
If location is not found
    Release the lock and Return
Else
    Create two nodes and adjust the list according to format given in the assignment
    Print the list
    Release the lock and return
```

```
Int myFree(int ID, int index)   //member function
Lock the mutex
Find the index
If given id and index does not exist
    Release the mutex and return
Else
    Make empty (change ID to -1)
    Adjust the list according to instructions (merge with neighbors)
    Print and unlock mutex
```