Programming Assignment (PA) -4
(Heap Management)
Report

CS307 – Operating Systems

15 December 2021
DEADLINE: 26 December 2021,
23:55

Rebah Özkoç
00029207

Synchronization Mechanisms:

In this assignment, I have used only one mutex for one heap object. Only myMalloc and myFree methods use this mutex. Once one of these methods are called by a thread the method grabs the mutex and unlocks at the end of the method. This behavior provided atomicity for myMalloc and myFree methods. Furthermore, it ensures mutual exclusion of these methods and prevents any race condition on the list because only one thread have access to the heap any time. If a thread is in myMalloc or myFree methods, other threads wait at the mutex lock line of the myMalloc or myFree methods and this avoids any race conditions. print and initHeap methods does not need mutex because only myMalloc and myFree methods call print method and initHeap is called only in the main thread.

Pseudo code for the locking management:

```
Heap class {

mutex = initialize the mutex

…

myMalloc (int ID, int size) {
      lock mutex
      if a free space exists:
            allocate the node
            print the resulting heap
            unlock the mutex
            return the method
      if not:
            print the resulting heap
            unlock the mutex
            return the method


myFree (int ID, int index) {
      lock mutex
      free the node
      merge the nodes if free nodes are next to each other
      print the result
      unlock the mutex
      return the method


…

}
```