CS 354
Lab 2
Manmohit Sehgal

The lab wanted us to implement basic locking and priority inheritence .
Steps taken :
– Implemented the following files:
    – lock.h
    – linit.c
    – lcreate.c
    – ldelete.c
    – lock.c
    – releaseall.c

– Modified the file wait.c, process.h, initialize.c

Implementing lock.c

– the file contains a structure which holds the values needed to implement the locking system for xinu.
– the file is also initialized in xinu.h

Implementing linit.c

– the file contains the initializing of the values in the structure made in lock.c.

Implementing lcreate.c

– the file contains two methods. 1) lcreate 2) generateNewLock. The generateNewLock function makes a new lock. It goes through all the locks and see which lock is in the LFREE(lock free) state and converts it into LUSED (lock used) state.

Implementing ldelete.c

– the file goes through the lock and deletes them if needed. It also checks for invalid locks

Implementing lock.c

– the file creates a lock checking weather the lock is a READ lock or a WRITE lock.
– it also contains the function increasePrio which increase the priority.

Implementing releaseall.c

– releases all the locks.

OUTPUT

```
15 spring CS354 Lab
Running test 0
 TEST0          DONE
Running test 1
 Reader1: Lock ..
 Reader2: Lock ..
 Reader3: Lock ..
 Reader1: Releasing ..
 Reader2: Releasing ..
 Reader3: Releasing ..
 TEST1: DONE
Running test 2
          TEST2

 Writer1: Lock ..
 Writer1: Releasing ..
 Writer3: Lock ..
 Writer3: Releasing ..
 Writer2: Lock ..
 Writer2: Releasing ..
 a49153
          TEST2DONE
Running test 3
 lock(2,READ) without lcreate() : -1
 lock(-2,READ)  : -1
 lock(lck,INVALID) : 1
 ldelete(lck) without lcreate() : -1
 ldelete(-1)  : -1
Running test 9
 Reader1: Lock ..
 Reader2: Lock ..
 Reader3: Lock ..
pid of medium priority process = 11
 Reader1: Releasing ..
 Reader2: Releasing ..
 Reader3: Releasing ..
 Writer1: Lock ..
 Writer1: Releasing ..
Test 9 done..


All user processes have completed.
```