HW 2

Patrick Wu, Alison Wong

All semaphores were implemented in proc.c. We also made system calls in usys.S, syscall.c, syscall.h and sysproc.c. We define the user functions in defs.h and user.h.

We define our semaphore table with semtableinit() in proc.c, which loops through an array of semaphores and calls sem_init() on each of them. sem_init() sets each semaphore to active and each value to sem.value. sem_wait() first acquires a spinlock, and checks with the semaphore value is below. If so, it will block the process with sleep() until value is above zero. Processes are arbitrarily unblocked in sem_signal() with wakeup().