# *Operating System Design MP1 Report*

Pin-Xuan Lee (plee18)
Yu-Lin Chien (ychien8)
Shuo-Yang Wang (swang234)
Wei-Tze Tsai (wtsai10)

***Contain files:***

Makefile mp1.c mp1_given.h userapp.c userapp.h reinstall_and_test_module.sh

***How to run:***

make
Sudo insmod mp1.ko
./userapp 10 &
./userapp 15 &
cat /proc/mp1/status

***Run in another way:***

./reinstall_and_test_module.sh

***Implementation and design decisions:***

Init:

Create /proc/mp1 directory and create /proc/mp1/status file.

Initialize linked list, timer, workqueue struct, and work struct.

register process (write /proc):

Use system() to echo pid into /proc/mp1/status in userapp.c, then in mp1.c, the pid information will be stored into the linked list.

read /proc:

Use sscanf() to read pid and cpu_time for each process.

timer & interrupt:

Timer is initialized by mp1_init(). For each 5 seconds, the timer callback function (top half) will be intrigued, and the work in workqueue will be called.

workqueue:

The work in workqueue (buttom half) will traverse the linked list, and use get_cpu_time() to update the cpu_time for each process.

lock:

Use spin_lock_irqsave() and spin_lock_irqrestore() to lock and unlock critical sections which will access linked list.

exit:

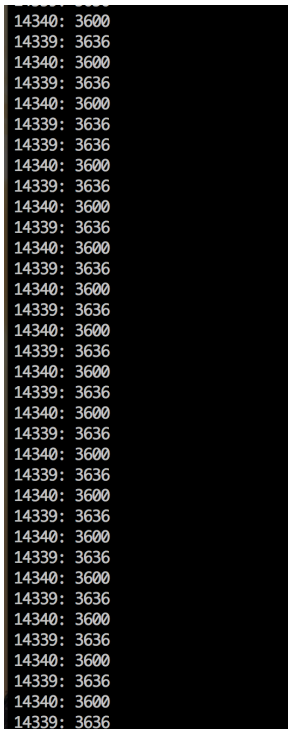
1. remove /proc/mp1/status
2. remove /proc/mp1
3. free memory of timer
4. free memory of linked list with list_for_each_entry_safe() macro
5. free memory of workqueue

userapp.c

Determine if the current time surpasses the start time plus the specified time period. If so, break out from the while loop.

***Screen shot:***

2 processes




1 process