ECS150 Discussion Section

Sophie Engle

(*week of 03 November 2003*)

Announcements

No discussion section

Tuesday, November 11th, 2003

due to Veteran's Day

(*bring questions to Monday's section*)

Upcoming Dates

Thu 16 Oct 2003	Homework 2 assigned
Tue 21 Oct 2003	Last chance to resubmit homework 1
Tue 11 Nov 2003	No discussion section
Thu 13 Nov 2003	**Midterm**
Tue 18 Nov 2003	Homework 2 due
	Homework 3 assigned
Fri 12 Dec 2003	**Final Exam**

Upcoming Discussions

- Week of October 20/21 (Sophie)
  - How/where to add system calls
- Week of October 27/28 (Eric)
  - How to modify schedule in kernel
- **Week of November 3/4 (Sophie)**
  - **Miscellaneous homework 2 questions**
- Week of November 10 (Eric)
  - Midterm questions

TA Website

http://wwwcsif.cs.ucdavis.edu/~cs150/

Informal Evaluation

- Would like informal evaluation
  - What works well?
  - What needs to be changed?
  - Comments? Suggestions?
- Turn in sheet of paper
  - Do <u>not</u> include name
  - Please fold in half and turn in at end of class

Homework 2 Clarification

System call `setLotteryTickets()` must access the process table

- ♦ Process table is located inside the kernel
- ♦ Must go between user-space to server-space (MM), and then server-space to kernel-space to access the process table

(see Eric's slides from last week)

Homework 2 Clarification

Adding MINIX System Call

Space	Name/Details

USER	`int setLotteryTickets( int _PID, int _tickets )`
	`/* calls do_setLotteryTickets() */`

SERVER
(MM)
	`int do_setLotteryTickets( void )`
	`/* calls do_syslottery() */`

KERNEL	`int do_syslottery( int _PID, int _tickets )`
	`/* main functionality implemented here */`

Homework 2 Clarification

- `setLotteryTickets( pid, tickets )` sets the total number of tickets for the process to `tickets`
- `setLotteryTickets( pid, tickets )` should return the number of tickets actually assigned to process `pid`

Homework 2 Hints & Tips

- Be careful naming parameters…
  - `pid` is defined in `/usr/src/mm/param.h`
  - `PID` is defined in `/usr/minix/com.h`

Homework 2 Hints & Tips

- Try to name parameters consistent to other system/tasks calls…
  - All parameters in `/usr/include/unistd.h` are prefixed with an underscore (i.e. `int _fd`)

Homework 2 Hints & Tips

- Recompile `ps` if you modify `struc proc` to prevent `ps` returning garbage
  - ♦ Go to folder `/usr/src/tools`
  - ♦ Type `make ps`
  - ♦ Type `make install`

Homework 2 Hints & Tips

- You may be able to avoid having to recompile the libraries for this assignment
  - Example: Instead of making a library function that makes a taskcall, include `minix/com.h` and use `_taskcall( … )` directly

Homework 2 Hints & Tips

- Define the constant MAX_TICKETS so that both the kernel and servers can access it
  - Where is NR_PROCS used?
  - Where is NR_PROCS defined?

Homework 2 Hints & Tips

- Try making a system/task call that returns the total number of tickets current assigned
  - Useful for debugging
  - May be good to implement before attempting `setLotteryTickets( … )`
  - May be helpful to determine if `fork()` should fail

Homework 2 Hints & Tips

- Understand how to use `rdy_head[USER_Q]` and the `p_nextready` attribute of `proc`
  - Useful for lottery scheduling
  - Useful for counting total tickets
  - Book has entire section on how processes are implemented in Minix

Homework 2 Hints & Tips

- If calculating the total number of tickets using `proc[]`, it may be helpful to:
  - Understand `p_priority` in `proc.h`
    - Should only count tickets from valid user processes
  - Understand where to start and end searching in the process table
    - See `/usr/src/kernel/main.c` line `06817` for how the kernel accesses `INIT`

Homework 2 Hints & Tips

- Determine if `fork()` should fail before extra memory is allocated for a child process
  - See the lecture 2 slides for more on `fork()`

Homework 2 Hints & Tips

- May be able to use `rand()` function in kernel if include `stdlib.h`
  - Useful for lottery scheduling algorithm

Testing setLotteryTickets

(look at sample test code)

(test programs will be provided on TA website)

Testing Lottery Scheduling

(test program will be provided on TA website)