ECE598: Advanced Operating Systems – Final Project

Spring 2016

Due: Friday, 6 May 2016 (Last day of Classes)

Overview:

- Design a project that extends our custom operating system.

Guidelines:

- You may work either alone or in groups of two. If you work in a group your end project will have higher expectations.
- You probably will want to use one of the homeworks as a starting place.

Part 1: Topic Selection (due 29 March 2016) (5pts)

Each group should send a brief e-mail describing your project topic and listing group members.

Part 2: Progress Report (due 19 April 2016) (10pts)

A brief status update detailing progress your group has made. This is primarily to make sure your project is on track to be finished in time; if things are not going well the topic can be adjusted.

Send this report by e-mail. Only one submission is needed per group.

1. State in one sentence a summary of your project.
2. Are things working? Do you plan to finish on time?
3. You can submit the status update by e-mail.

Part 3: In-Class Presentation 3 and 5 May 2016 (40pts)

- You will have 15 minutes to present. Plan for 10 minutes of presenting plus 5 minutes for questions.
- You may present slides using the projector if you want, but that's not strictly necessary.
- You should cover the following things (but feel free to include more):
  - An overview of exactly what you did.
  - Any challenges you encountered.
  - Related work: list any similar projects, and how your project differs.
  - Future work. If you had more time, what else would you do.

Part 4: Project Writeup, Officially due 6 May 2016 (45pts)

This will be a short paper (at least 4 pages, but you can include pictures, diagrams, etc.) that must contain all of the following:

1. Introduction: What your project is and what the goal was.
2. Related work. List and properly cite at least one related project that did a similar project. Something like the Linux kernel might be a good example (list the source file that accesses the same hardware).
3. Interface
   (a) Describe the software/hardware interface you used. Be sure to list where in the data sheet (or online) you got info for how the interface worked.
4. Software Implementation
   (a) Briefly describe your code.
   (b) Describe the operating system interface needed to use the code (system call, etc).
5. Conclusion
   (a) If you worked in a group: List who worked on what part.
   (b) Summarize what you did.
   (c) List and difficulties you encountered.
   (d) Future Work: List any improvements you might make if you had more time and resources to work on the project.
6. Appendix
   (a) I would like a copy of any relevant source code if possible. (this can be submitted as a separate file, does not have to be included in the report).

(b) I may want to post a list of projects, including project reports, to the course website. If you do not want your project posted, please indicate this in the final writeup.

You can e-mail your final report to me. pdf or word document is fine, the code should be attached too.

Potential Project Ideas:

These are just suggestions, feel free to come up with anything else that might be relevant.
Note, the difficulty level is my best guess, things might be way harder or easier than I estimate.

- Easy
  - Some sort of video game
  - Read from the internal temperature sensor, do something useful with the results
  - Read from the internal random number generator, do something useful with the results
- Medium
  - Drive an i2c display
  - Drive some SPI hardware
  - Drive some 1-wire hardware
  - Get the OS running on a Pi-2 or Pi-3
  - Optimize the memcpy() routine to be as fast as possible (in assembly language?)
  - Read values from a PS/2 mouse
  - Create an interface for the hardware performance counters
- Hard
  - Play sound output through the PWM/sound interface
  - Write a FAT filesystem driver
  - Write a driver for some other filesystem
  - Write a driver for the SD card interface
- Really Hard
  - Implement full virtual memory
  - Utilize multiple cores on the Pi2
  - Get the ethernet card working
  - Get USB working
  - Read from the Pi-cam or other webcam
  - Generate 3D graphcis using the GPU
  - Reverse engineer the on-board DSP
  - Reverse-engineer the Video Core firmware