CS 439 - Principles of Computer Systems
Project 4

Assigned on:  Oct 29 2015
Due by:  Dec 04 2015

This is the final project that will keep you busy until the end of the course. From the experience of past iterations of the class it is very important to start the project immediately as it will require a substantial amount of time that can hardly made up for in the week. In particular, you should use opportunities to discuss problems with TAs and professors and test your code.

We will offer two different options from which you can choose now. You are expected to make a decision by end of next Wednesday (11/04) by either posting to the Kernel group (Option B) thread on Piazza or signing up for a Threads group (Option A) on Canvas.

Both projects are based on CourseOS, an operating system that students developed in this 439 class in past semesters. You can find the source code for CourseOS on https://github.com/rellermeyer/course_os.

Option A: Implementing a Threads Package

We will create a new branch on the github repository in the next days and give you a starting point from which you are expected to implement a threads package including mechanisms for creating kernel-level threads (comparable to the pthread library) and extending the scheduler and dispatcher to deal with threads. With the code, you will also receive the proposed API, a detailed documentation of the steps required (on Piazza), and (gradually) a set of test cases that you need to pass in the end. It is advised to use the test cases regularly and track progress.

50 points will be awarded for correctness (determined by how many test cases of the final test case set your code passes), 20 points will be given based on the peer evaluation of your group members, and 10 points will be given for code style and documentation. You are allowed to work in groups of up to three students and are expected to sign up for your group on Canvas. After the initial round of group sign-up no further changes in group membership are allowed without permission of the instructors.

Towards the end of the project we will be posting an additional task together with separate test cases that you can implement for extra credit.

For this project, you are not required to sign up for a github account as you will only be cloning the sources from the repository but are not required to commit changes back. For the duration of the project you are asked to not share your code outside your group but you are allowed to discuss issues with peers outside your group.
Option B: Enhancing the Kernel

In this project, we will write an operating system almost from scratch starting from nothing but a toolchain, a system emulator, and a bare kernel that was developed by some of last year’s 439 students. Our target platform will be ARM and the goal is the be eventually able to run the OS not only on qemu but also on real Raspberry Pi devices.

The organization of the project will be similar to the one of an open source project. Everybody can contribute to the source code (residing on github) and we will track bugs, feature requests, enhancements, etc. through an issue tracker. People can grab a work items and implement them. However, there will be inter-dependencies so that communication is paramount.

It is important to understand that this is an open-ended project and everybody who wants to participate needs to be able to commit a significant amount of time to writing code and following what the other contributors are doing. Furthermore, people should be aware that this project requires to write a lot of code (compared to other projects) so decent programming skills are a prerequisite for succeeding in this project.

In the next days we will post a few suggestions for improvements on the kernel that could be beneficial through the bug tracker system on github. However, the majority of the project is self-organized and you can always propose interesting new problems to work on.

20 points will be given for the success of the entire project, the remaining 60 points are given for the individual contributions, e.g., determined through commit history, bug reports filed against work items, etc. You will be expected to submit documentation showing what you have worked on and to which extent this contributed to the overall success of the project so documenting the code accordingly and keeping track of your own work is advised.

In order to participate, you need to create an account on github and request to be added as a collaborator by posting to the corresponding topic on Piazza.