Lab 7 System Design Project

Goals	• Develop multi-threaded application example on top of your OS, • Demonstrate functionality of your OS, • Demonstrate performance of your OS.
Review	• Labs 1 through 6.
Starter files	• Your Lab 2, 3, 4, 5 or 6 OS.

Background

The goal of this final lab is to demonstrate both the functionality/capabilities and performance of your ultimate OS implementation that you developed throughout the previous labs in this class. In order to do so, you should design, develop and implement an example of a multi-threaded application of your choice running on top of your OS that will exercise and stress test the various aspects of your OS. Your application should at minimum use your Lab 2/3 OS, i.e. the capabilities and OS services we implemented for those labs. Use of any Lab 4, 5 or 6 features is optional (but will make for more interesting projects). The minimum application requirements are:

- Multi-threaded with at least 3 additional foreground threads beyond the idle task.
- Non-trivial use of semaphores for thread synchronization and/or communication.

As such, the expected minimum level of complexity is not more than what the *realmain()* from `Lab3.c` demonstrated. However, your application should be materially different from any of the existing *realmain* or *Testmain* in the labs.

You are not required to add any new features or capabilities to your OS for this lab. However, you are free to do so. We will give extra credit to any projects that do implement and test additional OS features, similar to the level of graduate students projects. You can look at slide 2 in Lecture 5 for inspiration and possible examples. If you already implemented OS features that went beyond the original requirements for earlier labs, you will be able to count those for extra credit in Lab 7. Details of extra credit points depend on the complexity of the extensions versus the size of the team, and should be discussed with the TAs during preparation. Graduate students who want to get extra credit for Lab 7 must implement what amounts to an additional set of features on top of their original graduate project proposal (where the amount of extra credit points depends on the complexity).

You will also need to measure and report on the performance of your final OS in the same way that you already did in Lab 3, i.e. running the *realmain()* from `Lab3.c` and reporting updated performance data on jitter and CPU utilization. We will use this data to rank student OSs and crown the best OS in this class. We will also award extra credit points to the top-ranked OSs in the class. Hence, one option for obtaining additional credit is to work on optimizing the performance of your OS.

Some ideas and suggestions for possible applications and projects:

- Network chat/instant messaging between two Launchpads.
- Extended web server that serves pages from files stored on disk.
- Networked game across two launchpads (e.g., wireless two-player pong).
- Standalone game (with multiple competing actors/AIs running as threads).
- Launchpad as (single-/multi-channel) oscilloscope and/or spectrum analyzer (FFT).
- Launchpad as Blynk or IFTTT IoT device (integrate with cloud service).
- Re-implement your 319K or 445L project using OS and threads.

Preparation

Present and discuss project proposal including any extra credit plans with TAs.

Procedure

Design, implement, test and debug your application example and/or OS extensions.

Checkout (show this to the TA)

Demonstrate your project to the TAs and instructor. Be prepared to also demonstrate your OS performance (running *realmain()* from `Lab3.c`) and to discuss any (functional and performance) limitations of your current OS.

Deliverables (exact components of the lab report and lab submission)

A) Objectives (1/2 page maximum)

B) Hardware Design (if any)

C) Software Design (documentation and code)

D) Measurement Data

1) Updated version of Table 3.1 data from Lab 3 for your final OS running *realmain()* from **`Lab3.c`**

E) Analysis and Discussion (none). Instead, include a demo video of your project:

1) Record a 2-3 minute video of your system and include a link in the lab report. Please mention your names (team and members) and send the link to the video (e.g. on YouTube) to the instructors. We will create a YouTube channel to showcase all the projects.