CS 162 Project 2

Deliverables / Design

Design

TABLE OF CONTENTS

Before you start writing any code for your project, you need to create an design plan for each feature and convince yourself that your design is correct. You must **submit a design document** and **attend a design review with your TA**. This will help you solidify your understanding of the project and have a solid attack plan before tackling a large codebase.

Disclaimer: Your design document must not exceed 15 pages in length. Please do not include large blocks of code in your design document. Exceeding this limit will result in a grade penalty.

Document

Like any techinical writing, your design document needs to be clean and well formatted. We've provided you with a template linked on the website that you must use. The template can be found on the website. We use Dropbox Paper which supports real-time collaboration like Google Docs with the added benefit of techincal writing support (e.g. code blocks, LaTeX). **Not using this template or failure to use code formatting will result in losing points.** The main goal of this is not to punish you for this but rather make it easy for TAs to read.

To get started, navigate to the template and click the "Create doc" button on the top right hand corner. You can share this doc with other group members to collaborate on it. **Make sure to click the blue "Share" button in *your* document, not the template.**

For each task except for Concept Check, you must explain the following aspects of your proposed design. We suggest you create a section for each task which has subsections for each of the following aspects.

Data Structures and Functions

List any `struct` definitions, global or static variables, `typedef`s, or enumerations that you will be adding or modifying (if it already exists). These should be **written with C not pseudocode**. Include a **brief explanation** (i.e. a one line comment) of the purpose of each modification. A more in depth explanation should be left for the following sections.

Algorithms

Tell us how you plan on writing the necessary code. Please explain the majority of your design in words, **if there is only code with no explanations for how you are going to implement your design, you will get no points for this section.** Your description should be at a level below the high level description of requirements given in the assignment. Do not repeat anything that is already stated on the spec.

On the other hand, your description should be at a level above the code itself. Don't give a line-by-line rundown of what code you plan to write. You may use *small snippets* of pseudocode or C in places you deem appropriate. Instead, you need to convince us that your design satisfies all the requirements, **especially any edge cases**. We expect you to have read through the Pintos source code when preparing your design document, and your design document should refer to the appropriate parts of the Pintos source code when necessary to clarify your implementation.

Synchronization

List all resources that are shared across threads and processes. For each resource, explain how the it is accessed (e.g. from an interrupt context) and describe your strategy to ensure it is shared and modified safely (i.e. no race conditions, deadlocks).

Similarly, if you believe that a section does not require synchronization, please also state the reasoning behind your decision. If there is no explanation given to your reasoning you won't be granted any points for this section.

In general, the best synchronization strategies are simple and easily verifiable. If your synchronization strategy is difficult to explain, this is a good indication that you should simplify your strategy. Discuss the time and memory costs of your synchronization approach, and whether your strategy will significantly limit the concurrency of the kernel and/or user processes/threads. When discussing the concurrency allowed by your approach, explain how frequently threads will contend on the shared resources, and any limits on the number of threads that can enter independent critical sections at a single time. You should aim to avoid locking strategies that are overly coarse.

Interrupt handlers cannot acquire locks. If you need to access a synchronized variable from an interrupt handler, consider disabling interrupts. Locks do not prevent a thread from being preempted. Threads can be interrupted during a critical section. Locks only guarantee that the critical section is only entered by one thread at a time.

Do not forget to consider memory deallocation as a synchronization issue. If you want to use pointers to `struct thread`, then you need to prove those threads can't exit and be deallocated while you're using them.

If you create new functions, you should consider whether the function could be called in 2 threads at the same time. If your function access any global or static variables, you need to show that there are no synchronization issues.

Rationale

Tell us why your design is better than the alternatives that you considered, or point out any shortcomings it may have. You should think about whether your design is easy to conceptualize, how much coding it will require, the time/space complexity of your algorithms, and how easy/difficult it would be to extend your design to accommodate additional features.

Review

After you submit your design doc, you will schedule a design review with your TA. A calendar signup link will be posted sometime before the design doc due date. During the design review, your TA will ask you questions about your design for the project. You should be prepared to defend your design and answer any clarifying questions your TA may have about your design document. The design review is also a good opportunity to get to know your TA for participation points.

Grading

The design document and design review will be graded together. Your score will reflect how convincing your design is, based on your explanation in your design document and your answers during the design review. If you cannot make a design review, please contact your TA to work out an arrangement. **An unexcused absence from a design review will result in a 0 for the design portion**.