~~How to Write a Design Document~~ "What I See in the Design Documents of Students Who Also Do Well in this Course"

CS161 Course Staff
cs161@eecs.harvard.edu

February 18, 2013

- *Tuesday in-class*: Peer design review. Please bring in a printed copy of your rough draft.
- *Thursday 9:OOPM*: Design documents due.
- *Friday, March 7th, 5:OOPM*: ASST2 due. Do not leave this until the last minute; it is more work than ASST1.

1 A Table of Contents

1. Data Structures
2. Bootstrapping/Initialization
3. File Syscalls
4. Fork/Exec
5. Whatever
6. Stuff

Break down your design into parts. Your breakdown should show that you understand the parts of the assignment. Do not write monolithic descriptions of your entire system, or you will end up writing a monolithic system.

2 Data Structures

"These are the fields we're adding to the Thread struct:"

```
struct Thread {
...
int foo // for fooing bars
}
```

"This is the format of a file handle:"

```
struct FileHandle {
lock_t baz // protects access to biff
...
}
```

You can put this at the beginning, or the end, or break them up on a section-by-section basis. Whatever you think is most clear. But don't allude to your data structures; make them explicit.

In particular, identify the synchronization structures in your design, and what they are used for/where they are used.

3 Real Pseudocode

This is not pseudocode for fork:

> "Fork creates a new child process of the caller (parent) process. The child is given a copy of its parent's address space and file handles. Fork returns twice; the parent returns the child's PID, and the child returns 0. Fork returns -1 on error."

This is just a description of fork!

I want to know how your system will implement fork.

1. This is how I create a new process. This is how I make it a child of the caller process.
2. This is how I copy its address space and file handles.
3. Fork is called once, but returns twice. So how do I return twice with only one trap frame? This is how.
4. Fork involves two threads. Do they share data? Does this require synchronization? (YES. And this is how...)
5. If any of the above steps fails, then I (clean up and return an error/kernel panic/give up and play starcraft)

I have probably forgotten some steps. You will probably not know how to do every step. But these are the kinds of questions you should try to ask and answer.

If you cannot figure out how to do something, then you'll know to ask for help! If you only miss a few important steps, then one of your reviewers is likely to point it out. But if you simply repeat back what the code is SUPPOSED to do in your own words, then all I can confirm is that you understand the assignment, which is good, but not as good as knowing how to do it.

3.1 Function Definitions

- It is good to identify the functions you need to modify for a given design goal.
- It is good to identify some helper functions you think will be worth implementing.
- It is bad to paste huge blocks of structured C code into your document.
- It is, in my humble opinion, surprisingly bad to rely on Javadoc style.

4 Effort Proportional to Difficulty

Fork is strictly harder than opening a file. So if you spend the same amount of time and space explaining them, then either you are underestimating fork or padding open file. You do not get bonus points for length; actually, it is much harder to give good feedback on an egregiously long design doc.

4.1 A Realistic Timeline

You are required to submit a timeline. Doubtless you will fall off this timeline, due to parties most excellent, or time wizards ("chronomancers"). The point of the timeline is to show us you have realistic expectations of how long each part of the assignment will take. So that's what you should try to achieve.

5 A Note on Partner Dynamics

This applies well beyond this course.

Teams are always assymetric, always. You should try to work together when you can, and divide work evenly when you can't. But in practice, some part of the assignment will take more time than expected, or will be harder than expected, or someone might be delayed due to other coursework, or whatever. There will be assymetries, so you need to be flexible, and if your partner needs help, then help.

If you feel there's a problem with your teamwork, then please talk to the staff. Do not work independently from your partner until the day before the project is due. And do not go off and implement the assignment on your own, because you feel (your partner is slacking/your code is better/whatever). I know, empirically, this will not work. If you are at the point where you would consider that, talk to the course staff.