Operating Systems

CMPT 424

Lab 100

Goals

Simulating disk hardware and software, and implementing swapping
This active learning exercise will you help you make progress on the practical aspects of developing your operating system.

Instructions

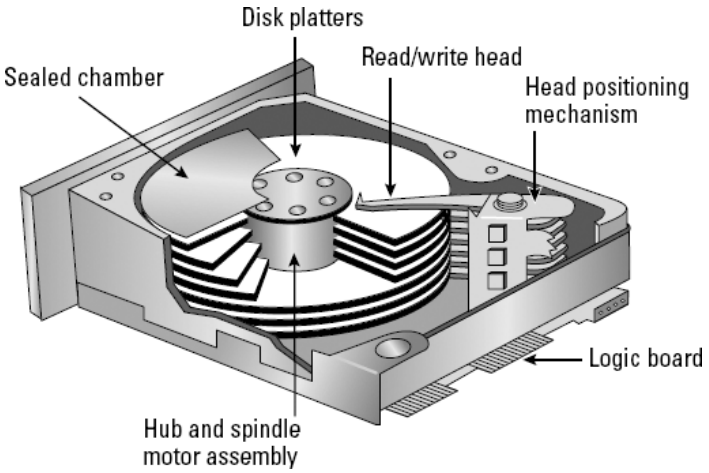
1. Add the *i*Project 4 / Final Project functional requirements as Issues (*enhancement*s, really) in GitHub as element of an "Final Project" milestone.
2. We need to simulate physical storage in the browser. We'll use HTML5 Session Storage for this. Read up on this in the resources listed below.
3. Once you're familiar with the resources below, begin your HTML5 storage implementation. Please use **Session Storage** (as opposed to Local Storage).
4. Once you have the "raw" storage, implement a file system API supporting the commands specified in your Issues and Final Project.
5. The test commands given in the resources section below give you a good idea of what to expect in terms of file system usage from the CLI.
6. Remember, your scheduler or context switcher will make use of the file system too, so be mindful of that in your design.
7. Finally, now that your raw disk and file system are working, develop a **swapper** that can *roll out* a program from memory and *roll in* to memory a program stored on disk. Be sure that you update the PBCs as necessary.
8. Expand your CPU scheduler to handle four (4) or more processes in execution at once, taking into account using the disk as a backing store for your swapper.
9. Read chapters 10, 11, and 12 in the 8th edition of our text. They're good.

Resources

- https://mislav.github.io/diveintohtml5/storage.html
- Chapter 8 in the Max Hailperin book.
- Chapter 37 in http://pages.cs.wisc.edu/%7Eremzi/OSTEP/
- Chapter 21 in http://pages.cs.wisc.edu/%7Eremzi/OSTEP/

Submitting

Really? You know this by now.



Operating Systems

CMPT 424

```
>format
Format successful
>create alan
File created: alan
>write alan "this is a test."
File updated: alan
>read alan
this is a test.
>write alan "this"
File updated: alan
>read alan
this
>ls
alan
>delete alan
File deleted: alan
>read alan
File does not exist: alan
>ls
No files exist
>
```

Some tests your file system

format
create alan
write alan "this is a test."
ls
read alan
write alan "this"
read alan
write alan "12345678901234567890123456789012345678901234567890123456789012345678901234567890"
read alan
delete alan
ls
read alan