

CS 311 Operating Systems I Final Exam
Comparison of POSIX and Windows APIs
Due 09:30 Monday, 09 December 2013

Background

Throughout this class, we have spent a lot of time discussing the POSIX API. While this is becoming more truly “portable” with the rise of POSIX compliant mobile OSes such as iOS and Android, the fact remains that Windows is the dominant operating system. Unfortunately, the Windows API and the POSIX API are quite different, both in terms of the functionality provided as well as the calls themselves.

The Exam

I want you to look back at the concepts we have covered in this class. For instance, file I/O, pipes, mutual exclusion, etc. Pick any three (for extra credit, do a fourth) topics which we discussed, and compare and contrast the Windows and POSIX APIs. Remember, a “topic” is an overarching idea, not a specific piece of functionality. If there isn’t at least a chapter covering it, it’s too small to be an individual topic.

You are to write your paper in L^AT_EX and submit it via TEACH. Feel free to include snippets of code which can illustrate any points you might be making. And by “feel free”, I mean make sure to do so.

Why the heck am I doing this?

Knowing and being able to make use of multiple systems APIs makes you fundamentally more marketable. As you enter the real world and start your careers, there is no telling what you will be doing in 5 or 10 years. While you might start out in a nice comfy open source POSIX compliant programming position, you might transition to a Windows centric position in the near future. Further, while systems programming is often OS specific, much of what a low level tool does is OS agnostic. Once a file is open and read from, the data is the data. As such, being able to use something like preprocessor macros or abstraction layers to select which systems API you are using will allow you to write truly portable code.