Systems and Tools

Daniel R. Cender

Grand Canyon University: CST-221

January 12, 2020

**Operating System Exploration**

In this section, I will explore five different features of the OSX operating system and attempt to explain their functionality from how I perceive them to work “under the hood.”

1. **Creating a new file**

I imagine that a few processes run in sequence when a new file is created. There is likely a low-level system call that is made, similar to calling “touch file.txt” from the command line to create a blank file. It checks the space remaining on the disk, whether the user has permissions to create files and at that location, and whether a file of that name and extension already exists in the current directory. With that, the process likely assigns a specific identification hash to the new file, so it is identifiable throughout the entire disk. A starting amount of storage space is also blocked off for use by this file, until the file threatens to spill over that allotted amount sometime later on. The process grabs the current time and assigns it to some meta-data fields like “created at” and “updated at.” When I create a file in Visual Studio Code’s directory (by selecting the + document symbol), I can then select “More Info” when I right-click the file in OSX’s Finder program. This signifies that the file contains “meta” information, and not just the plain text contents. Once all the previous is completed, an event is registered which will refresh any open Finder windows to show the new file.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. **Deleting an Item in Finder**

The action of deleting a file or directory has some underlying complexity to it. When I click “Remove to Trash” on a file in Finder, then decide to “Delete Immediately” in the Recycle Bin program, a process kicks off which will make checks against my permissions and the file’s role.

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

The process first checks to make sure I have permission to delete that file. If I do, then the process checks whether that file or directory is listed as a required dependency of any critical operating system functions or utilities. We wouldn’t like to be allowed to accidentally delete crucial pieces of the OS.

1. **Listing all Items in a Directory**

I believe this process is deceptively easy to perform as a user, but it requires more upkeep for the operating system behind the scenes. When I open a Terminal instance and type “ls,” I can see a list of all the files and subdirectories in my current directory.

A screenshot of a computer

Description automatically generated

Since a file system looks much like a graph, where a single node links to many child nodes, which both contain references to each other for easy traversal. I imagine that the file system on all operating systems contain a graph of information that tracks this information for every directory and file in it. The OS needs to know what directories hold what files, and which files symbolically link to other ones in different directories. When I run “ls” in the Terminal, it grabs my current directory and seeks for it in the file system graph. Once found, it logs the name of all my directory’s child “nodes” to my screen.

1. **Using the System Calculator**

The calculator program that comes with OSX is quite handy. When I type in a basic math expression “4 + 3”, I would expect the program to attempt to parse out my expression for actionable values, which it can check against a list of inappropriate expressions or values. Obviously, the operating system wouldn’t want to pass my request to run “0 / 0” to the CPU, since terrible errors might occur from dividing by zero. Instead, it checks that table and spits out a pre-defined result.

A picture containing electronics

Description automatically generated

Once the calculator has verified my requests, it will then run the operations using system calls or base CPU instructions, whichever is sufficient for the level of math operation needed.

1. **Plugging in My Headphones**

When I plug my headphones into my laptop, there are some checks that the OS likely performs against my accessories. Although it’s presumed the headphone jack is to be used exclusively for headphones, that’s not to say someone could have adapted a USB stick to send malicious instructions to the hardware via a headphone jack adapter. When I plug them in, the OS inspects the software running in the accessory and if any safe I/O drivers map to that form of input. If so, then the driver is loaded and utilized as a safe instruction “passer” between the OS and the accessory device. If not, the system may throw an error or trigger an alert to the user that their device is invalid. Opening the Sound menu in OSX’s System Preferences shows me all currently configured, valid sound inputs and outputs.

A screenshot of a cell phone

Description automatically generated

**UNIX Availability**

On Mac devices installed with OSX, underlying UNIX utilities are accessible from the command line, called Terminal.

A screenshot of a computer screen

Description automatically generated

**Output of the HelloWorld C Program**

Per the assignment instructions, I wrote, compiled, and ran a basic “Hello World” C program. The output is shown below:

A screenshot of a computer screen

Description automatically generated

The code for the basic HelloWorld program is accessible from my GitHub repository for this course: <https://github.com/DanielCender/CST-221/blob/master/Assignment1/HelloWorld.c>.

The README file for my repository is available here: <https://github.com/DanielCender/CST-221>

References