**25 – I/O and File Systems**

**Activities**

COMP256 – Computing Abstractions

Dickinson College

Spring 2021

Prof. Grant Braught

**Name:**

**Introduction:**

Today’s class discussed how some of the operating systems abstractions that we have seen previously (e.g. libraries, system calls and device interrupts) are used to perform input/output operations on devices. It also introduced device drivers as another layer of abstraction that allows the operating system to interact with devices that have different physical implementations using a uniform interface (e.g. HDD and SDD). The activities below will first explore those topics and will then, through another resource, provide insight into how the operating systems produces the abstraction of directories and files.

**The Device Stack:**

1. What are the 6 levels of abstraction in the device stack?

2. Which levels of the device stack operate in user mode? Which operate in kernel mode? Think about when the OS switches from user mode to kernel mode and which operations would need to be performed in kernel mode. Briefly justify your answers.

3. Early operating systems where all of the devices were produced by the same manufacturer as the computer did not use device drivers. However, device drivers now provide an essential layer of abstraction in the device stack.

a. As an abstraction, what information do device drivers allow the operating system to ignore?

b. Why is being able to ignore the details you identified in a be important in an environment where devices might be produced by many different manufacturers?

c. Does it make more sense for the code in the device driver to be written by the device manufacturer or by the operating system developer? Briefly explain your answer.

4. As an abstraction, what information does the use of a Compiler/Interpreter library allow the high-level language programmer to ignore?

5. When a high-level language program wants to interact with an I/O device it must do so through the operating system. Briefly explain in a paragraph of your own words how control passes from a high-level language program to the physical device. Include the following terms in your explanation: high level language program, library function, interrupt (IRQ) instruction, interrupt vector, interrupt service routine, system call vector, system call handler, device driver.

**Your Metaphor:**

6. In class we used the campus metaphor to illustrate how the abstractions of the device stack bridge the gap between the user experience and the physical hardware of a device. This question asks you to do the same using your metaphor.

a. In a sentence, reintroduce your metaphor. You can copy this from a previous assignment. I ask for it to help me remember what it was as I read the rest of your answer.

b. Identify the elements of your metaphor that you will be using to play the roles of the hardware (i.e. device(s)), the processes and the operating system (i.e. the system call and device interrupt handlers).

c. Use the elements of your metaphor from part b to explain the how the OS uses the device stack to process an I/O request. Be sure to clearly connect your metaphor elements to the elements of the device stack as you explain.

**File Systems:**

As described at the end of today’s class, a file system is an operating system construct that allows users to interact with data in the sectors of a disk abstractly as directories and files. In class we did not cover the internals of how the OS creates these abstractions. Instead, watch the video *Files & File Systems* with Carrie Anne from the Crash Course Computer Science series and then answer the questions below:

* <https://www.youtube.com/watch?v=KN8YgJnShPM> (12:02)

7. What are file formats?

8. Formatted files often contain metadata.

a. What is metadata?

b. Give a few examples of metadata that appears in sound or image files.

c. Think of another type of file that you have on your computer. What type of metadata might it contain?

9. What is the purpose of a directory file? Where is it stored?

10. The directory file is used by the operating system to maintain metadata about each file in the directory.

a. What are the most important pieces of metadata that are maintained about each file?

b. Give two examples of how the operating system would use the metadata.

11. What is the difference between a flat file system and a hierarchical file system? Which type is used by the operating system on your computer? How can you tell?

12. As files and directories are changed the metadata information in the directory files must be maintained and updated. Will this metadata be changed by: The HLL code? The library functions? The System Call handler? Or the Device Driver? To be clear, this question is asking about changes to the metadata, not the actual writing of the meta data to the physical disk. Briefly justify your answer. Note: The answer to this is not in the Crash Course video, rather it requires you to think about the device stack and what happens or can happen at each level of abstraction.

13. At one-point in the video Carrie Anne suggests that the allocation of blocks to files is conceptually similar to what is done in virtual memory. In what way is it similar?

Optional: To help me improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.