xCS 3423 Operating Systems

Fall Semester 2021

Prof. Pai H. Chou

Weekly Review 13

Scope: Chapter 12, I/O Systems and
(part of) Chapter 13, File System Interface

## 1. Definitions and Short Answers

1. In terms of I/O systems,
	1. what is a **port** of a device?
	2. what is a **bus**?
	3. what is a **daisy chain**?
	4. what does a **controller** do?
2. What are four kinds of **device registers** that need to be accessed by the host for the following operations?
	1. How does a host receive data?
	2. How does a host send data?
	3. How does a host find the status or error of a device?
	4. How does a host change settings on a device?
3. If a processor supports **I/O instructions**, what kind of I/O is it called?
4. If a processor does **not** support I/O instructions, can it perform I/O? If so, what is it called, and what kind of instruction does it use to perform I/O? Or if not, why not?
5. Does the processor in a personal computer "see" the PCI or PCIe bus directly? If not, what does it have to go through?
6. How is a hard drive with a SATA interface connected to a PC with a PCIe bus?
7. How is a USB keyboard connected to a PC with a PCIe bus?
8. Is **polling** more suitable for slow or fast devices? Why?
9. Can polling be made more efficient by *context switching* to another process between two status checks? What potential problems may happen?
10. What is the meaning of **interrupt chaining**? What is a good reason for it?
11. What are examples of **exceptions** that an OS handles? What kind of mechanism is used for an OS to handle an exception?
12. What is the meaning of **split interrupt management**? What is the reason for it?
13. When a DMA controller and a processor both have to access the main memory, what happens? Which one gets priority? Or can both access memory simultaneously? Does it slow down the CPU? Does the DMA controller update the processor's data cache?
14. Which of the following Unix calls are for which types of devices?

| API | block device or character device | synchronous, asynchronous, or nonblocking |
| --- | --- | --- |
| read() | block | nonblocking |
| write() |  |  |
| seek() |  |  |
| get() |  |  |
| put() |  |  |
| select() |  | synchronous (blocking) |

1. What is the purpose of ioctl()?
2. What is the meaning of **vectored I/O**, and why is it a good idea?
3. Does every I/O system call cause a device driver to be invoked? Why or why not?

--- Chapter 13 ---
4. Where are the metadata such as file name of a file stored and where is it kept?
5. What is the **file pointer** and what are different ways it can be moved?
6. What does truncate() do and why is it a better idea than delete() and (re-)create the file?
7. Is it a good idea for an OS to define the formats for most types of files? Why or why not?

## 2. Programming Exercise

There is no programming exercise this week, but be sure you are caught up with your project checkpoint.