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?

connection point for device

* 1. what is a **bus**?

shared (group of) wires for connecting ports

* 1. what is a **daisy chain**?
  2. what does a **controller** do?

operates on port, bus, device. Sometimes separate circuit board. Some talk to per-device

controller with bus controller, microcode, memory

1. 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?

data-in

* 1. How does a host send data?

data-out

* 1. How does a host find the status or error of a device?

status

* 1. How does a host change settings on a device?

control

1. If a processor supports **I/O instructions**, what kind of I/O is it called?

Direct I/O instructions

1. 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?
2. Does the processor in a personal computer "see" the PCI or PCIe bus directly? If not, what does it have to go through?
3. How is a hard drive with a SATA interface connected to a PC with a PCIe bus?
4. How is a USB keyboard connected to a PC with a PCIe bus?
5. Is **polling** more suitable for slow or fast devices? Why?

wasting time

1. Can polling be made more efficient by *context switching* to another process between two status checks? What potential problems may happen?

yes

But if miss a cycle data overwritten / lost

1. What is the meaning of **interrupt chaining**? What is a good reason for it?

Multiple devices share an IRQ => share ISR

Once invoked, the ISR must query each of shared device

1. What are examples of **exceptions** that an OS handles? What kind of mechanism is used for an OS to handle an exception?

Divide by zero, memory access violation, insufficient privilege, page fault

ISR, Trap

1. What is the meaning of **split interrupt management**? What is the reason for it?

first-level interrupt handler (FLIH) - actual ISR to do I/O

second-level interrupt handler (SLIH) - separately scheduled routine to process the data

(without I/O) for the OS

1. 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?

DMA controller grabs bus from CPU

DMA

no

yes

…

1. Which of the following Unix calls are for which types of devices?  
   p27,28,30

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

1. What is the purpose of ioctl()?

For direct manipulation of I/O device specific characteristics, usually an escape / back door:

to send arbitrary bits to a device control register and data to device data register

1. What is the meaning of **vectored I/O**, and why is it a good idea?

Allows one system call to perform multiple I/O operations

better than multiple individual I/O calls, Decreases context switching and system call overhead

1. Does every I/O system call cause a device driver to be invoked? Why or why not?  
     
   --- Chapter 13 ---
2. Where are the metadata such as file name of a file stored and where is it kept?

In the directory structure, maintained on disk

Could also be kept in a registry or metadata file

1. What is the **file pointer** and what are different ways it can be moved?

pointer to last read/write location in file

…

1. What does truncate() do and why is it a better idea than delete() and (re-)create the file?

write over file & update (instead of recreate) attributes

more efficient

1. Is it a good idea for an OS to define the formats for most types of files? Why or why not?

no

OS get more complicated

## 2. Programming Exercise

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