CS 3423 Operating Systems

Fall Semester 2019

Prof. Pai H. Chou

Assignment 13

Due Date: Sunday, December 15, 2019, 11:59pm

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
   2. what is a **bus**? shared (group of) wires for connecting ports
   3. what is a **daisy chain**? 一種bus的連接方式
   4. what does a **controller** do?operates on port, bus, device，Contains own "processor", microcode, memory, bus controller
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?data-in register is read by the host
   2. How does a host send data? data-out register is written by the host
   3. How does a host find the status or error of a device? status register contains bits that can be read by the host
   4. How does a host change settings on a device? control register can be written by the host
3. If a processor supports **I/O instructions**, what kind of I/O is it called?Direct I/O instructions
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?

yes, Memory-mapped I/O, Load/store instructions like regular memory, but effect is to access device data and command registers

1. Is **polling** more suitable for slow or fast devices? fast Why?if the wait may be long,

the host should probably switch to another task

1. Can polling be made more efficient by *context switching* to another process between two status checks? What potential problems may happen?may miss a cycle data overwritten / lost
2. What is the meaning of **interrupt chaining**? Multiple devices share an IRQ => share ISR What is a good reason for it? often there are many more devices than IRQ lines
3. What are examples of **exceptions** that an OS handles? Divide by zero, memory access violation, insufficient privilege, page fault
4. What kind of mechanism is used for an OS to handle an exception?ISR for OS
5. 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?

DMA: By passes CPU to transfer data directly between I/O device and memory

1. 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() | block |  |
| seek() | block |  |
| get() | character |  |
| put() | character |  |
| select() |  | synchronous (blocking) |

1. What is the purpose of ioctl()? "backdoor" to pass command & pointer to driver
2. What is the meaning of **vectored I/O**,allows one system call to perform multiple I/O operations and why is it a good idea? Decreases context switching and system call overhead
3. Does every I/O system call cause a device driver to be invoked?NO Why or why not? If already satisfied I/O request, then device driver not invoked (ch12 p51)  
     
   --- Chapter 13 ---
4. Where are the metadata such as file name of a file stored and where is it kept?registry(登錄黨)or metadata file
5. What is the **file pointer** and what are different ways it can be moved? pointer to last read/write location in file
6. 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
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.