CSCI315 – Operating Systems Design

Department of Computer Science
Bucknell University

Operating Systems Services and I/O

**Ch 2.1-2.4,
12.1-12.3**

*This set of notes is based on notes from the textbook authors, as well as L. Felipe Perrone and other instructors.
Xiannong Meng, Fall 2021.*

Services Provided by OS to Users (1 of 2)

- Provide users with direct services
  - User interfaces: command line, GUI
  - Program execution: running programs
  - I/O: printing, viewing screen, issuing commands
  - File operations: creating and maintaining files on disks

Services Provided by OS to Users (2 of 2)

- Provide users with direct services (cont.)
  - Process communication: user programs can talk to each other
  - Error detecting and handling: notifying users with program errors, not bringing down the system

Services Provided by OS to Systems

- Services needed by the system
  - Resource management: allocating CPU time, memory, storage for all programs, including user programs
  - Logging: keeping track of the resource usage and error status
  - Protection and security: making sure the programs access only the authorized resources

INPUT AND OUTPUT

Computing System Input and Output

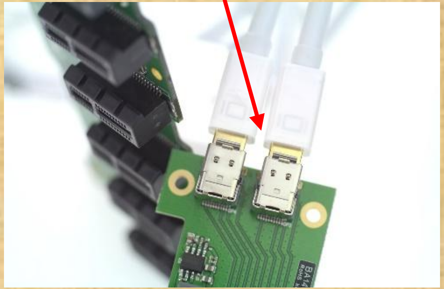
Assumptions:

Wired network card controller

- I/O devices and the CPU can execute **concurrently**.
- Each *device controller* is in charge of a particular device type.
- Each device controller has a *local buffer* – memory on board.
- There must be some mechanism to move data between main memory and local buffers on the controller.
- I/O operations move data between the device and a controller's local buffer.
- There must be some mechanism for the CPU to learn that an I/O operation has completed.

Commonly Used Terms

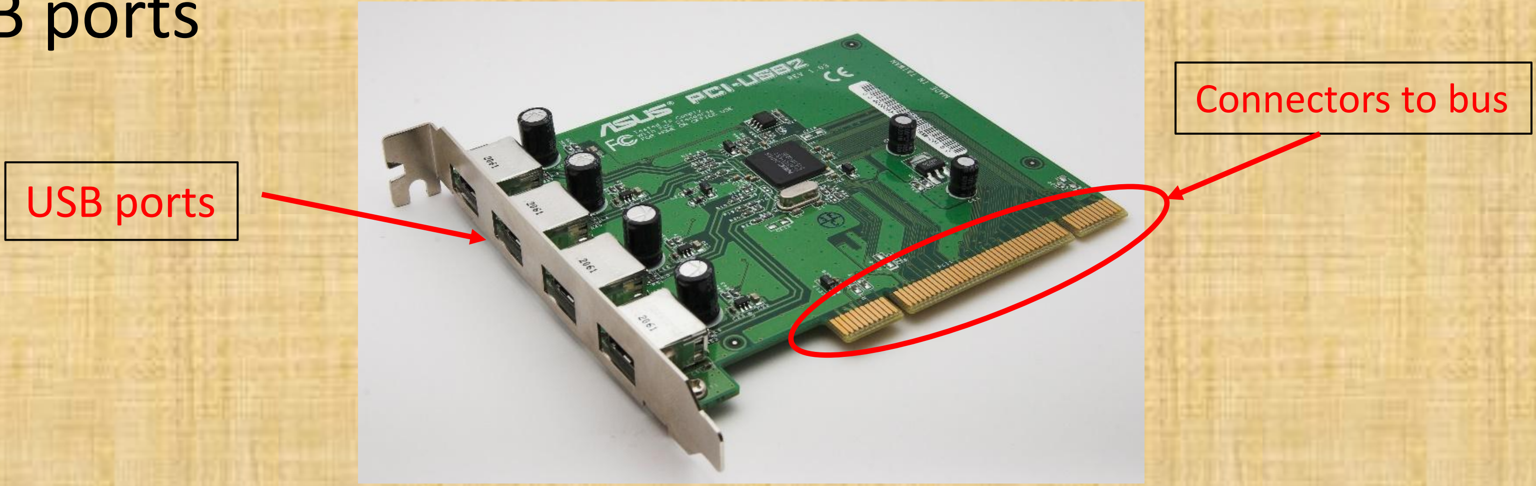
- **Controller**: A piece of hardware along with the software that works between the hardware device and the CPU
- **Device driver**: The program (software) that operates or controls a device through a controller for the computer system
- **Port:** The connection point for device
- **Bus**: Wires that the data and commands travel through
- **Polling**: The operating system periodically checks the status of an I/O device
- **Interrupt**: The I/O device in operation sends signal to the operating system when I/O is ready or needed



Two USB ports

An Example of Controller

- We’ve seen the Fast Ethernet controller in last lecture. Here is another one: a controller for USB ports

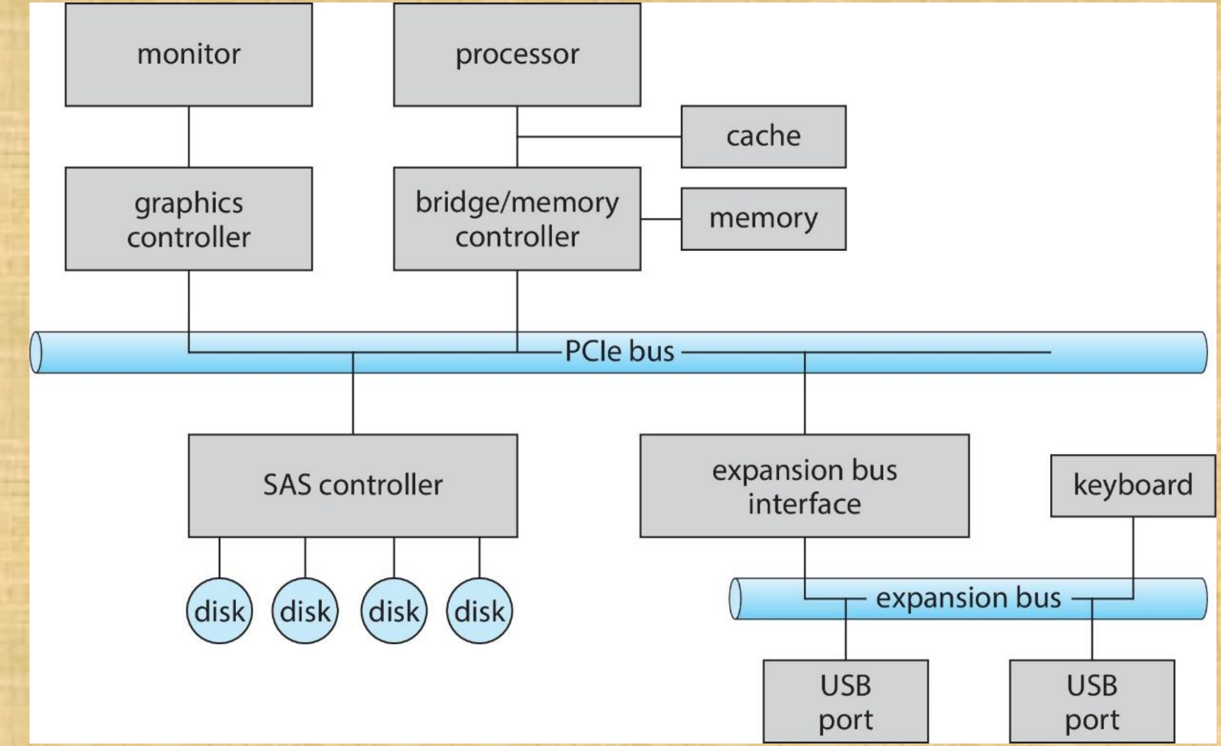


USB ports

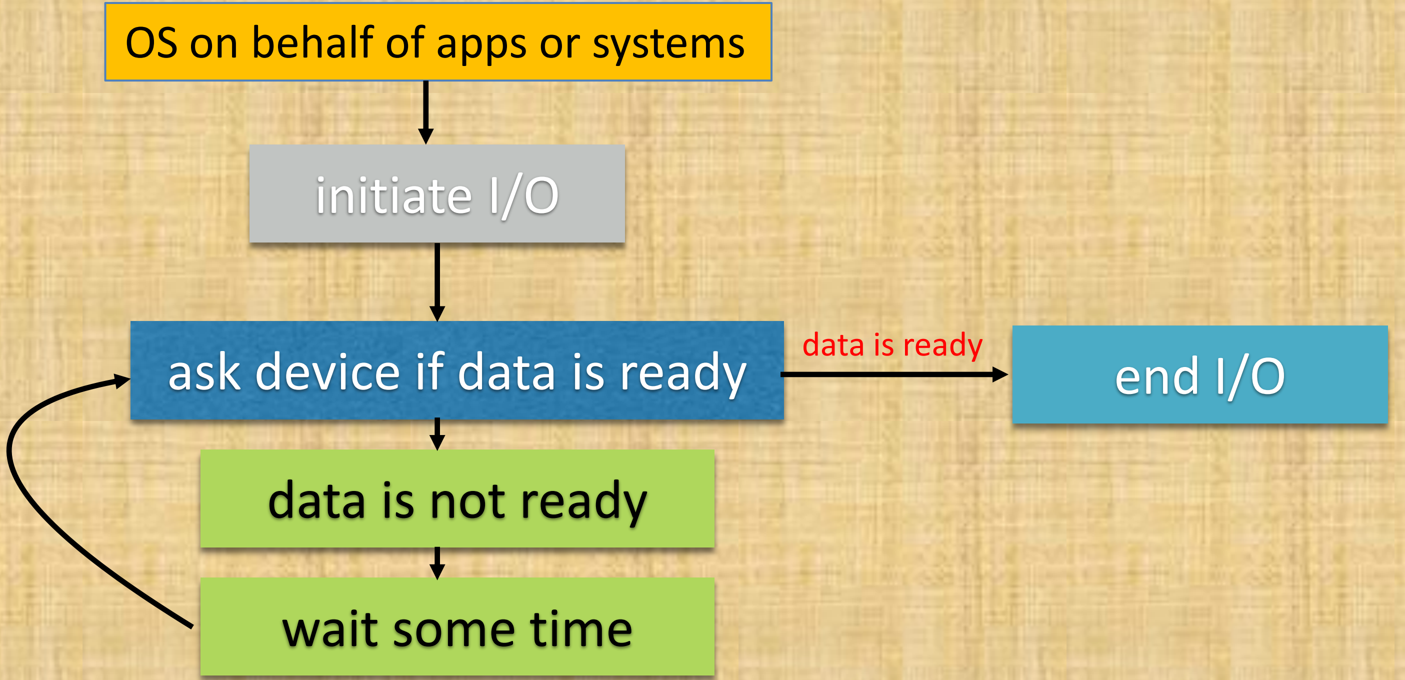
Connectors to bus

https://commons.wikimedia.org/wiki/File:USB2-PCI_Card.jpg

A Typical PC Bus Structure



I/O Option 1: Polling


OS on behalf of apps or systems

initiate I/O

ask device if data is ready

data is ready

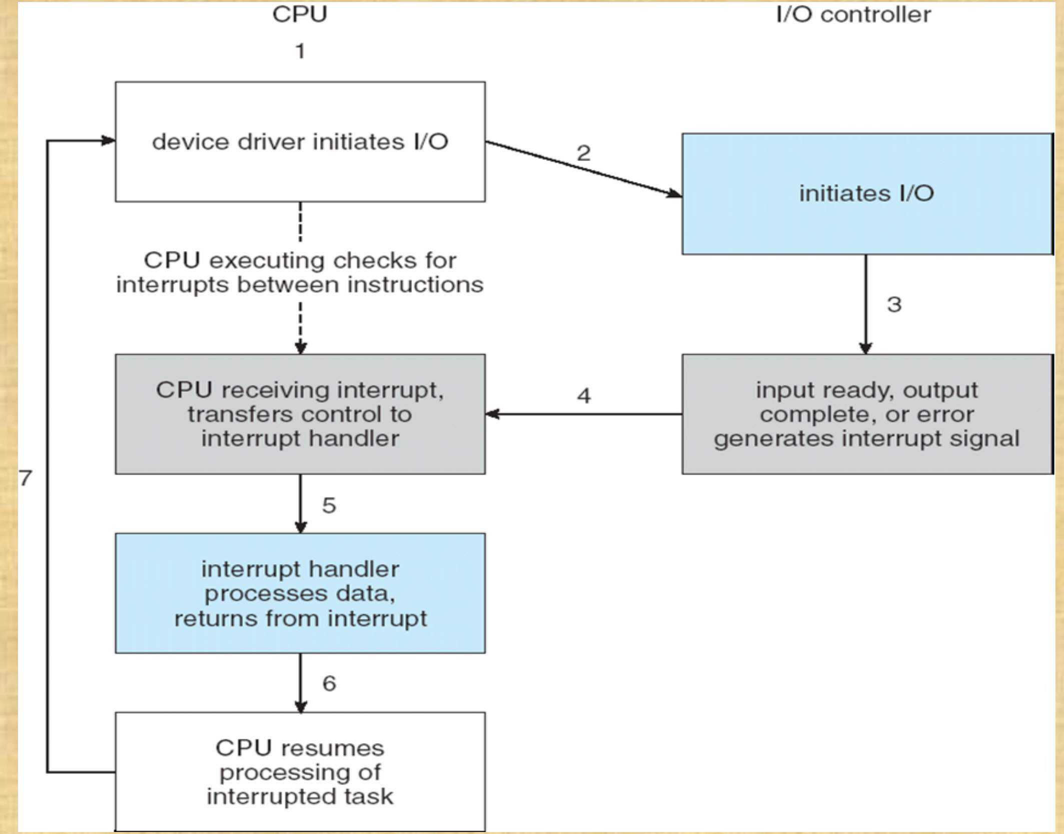
end I/O

data is not ready

wait some time

The Simpsons: https://www.youtube.com/watch?v=18AzodTPG5U

I/O Option 2: Interrupt


CPU
I/O controller
1
device driver initiates I/O
2
initiates I/O
CPU executing checks for interrupts between instructions
3
CPU receiving interrupt, transfers control to interrupt handler
4
input ready, output complete, or error generates interrupt signal
7
5
interrupt handler processes data, returns from interrupt
6
CPU resumes processing of interrupted task

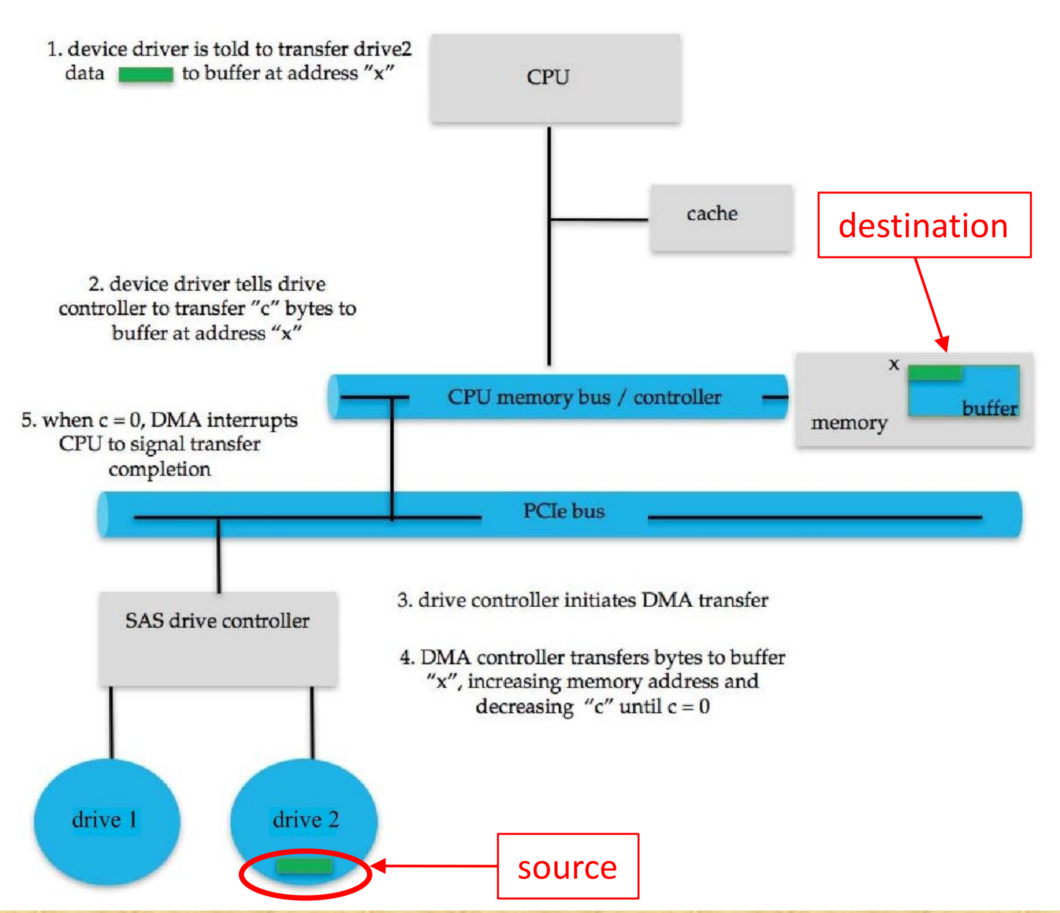
Direct Memory Access (1)

- Used to avoid **programmed I/O** (one byte at a time) for large data movement
- Requires **DMA** controller
- Bypasses CPU to transfer data directly between I/O device and memory

Direct Memory Access (2)

- OS sends the request to DMA controller. The request (command block) includes
  - Source and destination addresses
  - Read or write mode
  - Count of bytes
  - Writes location of command block to DMA controller
- DMA controller gains control of the bus and starts data transfer
- When done, interrupts to signal completion

Five Steps in DMA Transfer



1. device driver is told to transfer drive2 data 🟩 to buffer at address ”x”

2. device driver tells drive controller to transfer ”c” bytes to buffer at address “x”

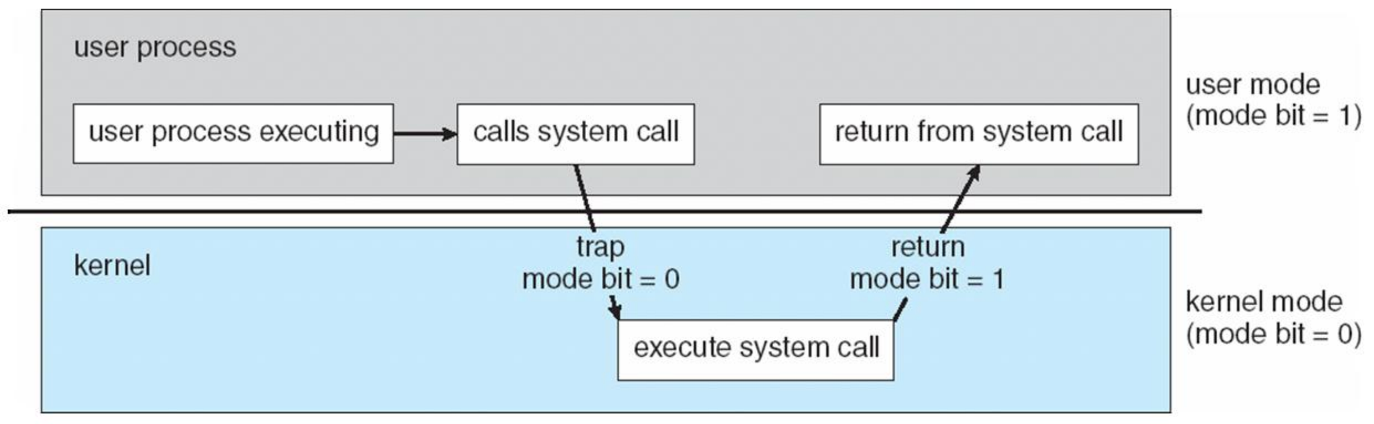
3. drive controller initiates DMA transfer

4. DMA controller transfers bytes to buffer “x”, increasing memory address and decreasing “c” until c = 0

5. when c = 0, DMA interrupts CPU to signal transfer completion

Hardware Support for the OS

- Two classes of instructions: one class for everyone to use, others with privileged use (for the OS kernel).
- Need to be able to switch between user mode and kernel mode.
- To switch to kernel mode, you need to trap to the kernel.


user process
user process executing
calls system call
return from system call
user mode
(mode bit = 1)
kernel
trap
mode bit = 0
return
mode bit = 1
execute system call
kernel mode
(mode bit = 0)