This Week

- Tutorials Thurs: Lab 4 File Systems
- Homework:

  Chap 10: problems 1, 7, 10

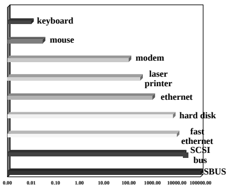
  Chap 11: problem 6

I/O and Network Structures

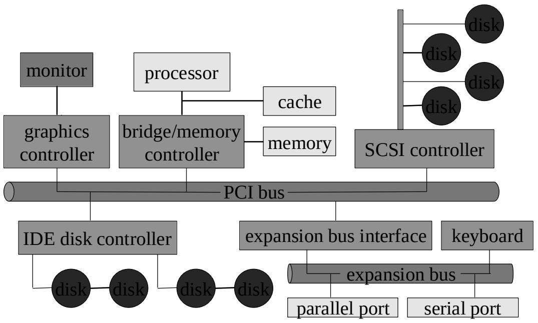
Comp 305 Lecture 10

The Problem

- Extensive use of IO devices, network
- Devices are idiosyncratic
- Devices require kernel
- Speed differences



IO Hardware



monitor

processor

cache

graphics controller

bridge/memory controller

memory

SCSI controller

disk

disk

disk

disk

PCI bus

IDE disk controller

expansion bus interface

keyboard

expansion bus

disk

disk

disk

disk

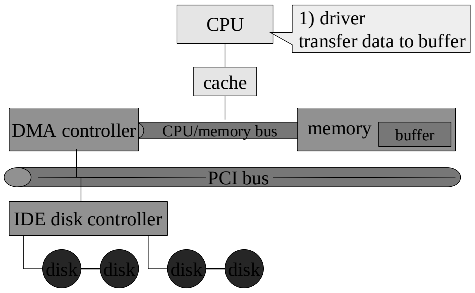
parallel port

serial port

Controller or Device Interface

- IO Instructions
- Memory mapped
- Polling
- Interrupts
- DMA

Direct Memory Access



CPU

1) driver transfer data to buffer

cache

DMA controller

CPU/memory bus

memory

buffer

PCI bus

IDE disk controller

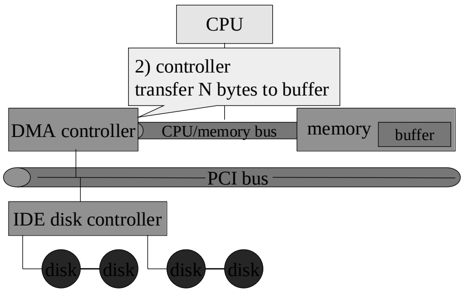
disk

disk

disk

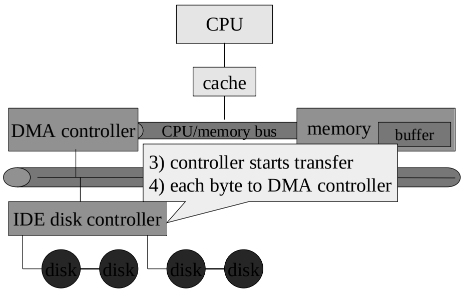
disk

Direct Memory Access



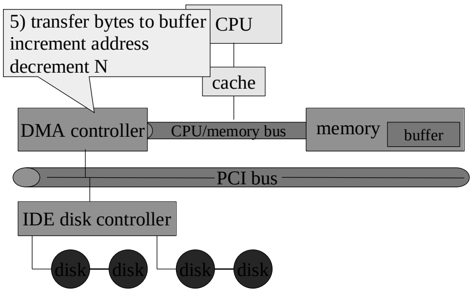
2) controller transfer N bytes to buffer

Direct Memory Access



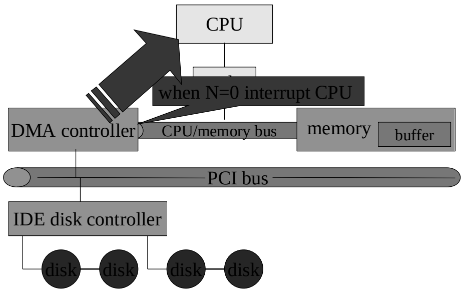
3) controller starts transfer
4) each byte to DMA controller

Direct Memory Access



5) transfer bytes to buffer
increment address
decrement N

Direct Memory Access

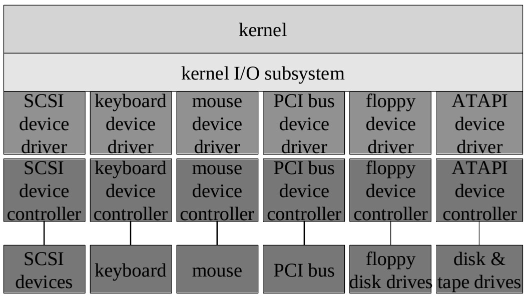


when N=0 interrupt CPU

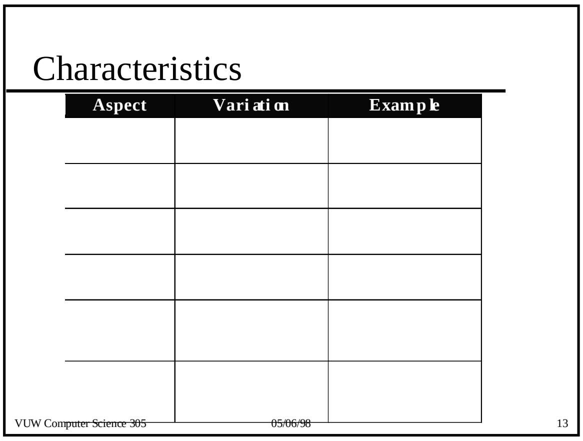
IO Software

- Device drivers
  - Block device interface
  - Character device interface
- Two level design
  - Strategy
  - Detail

Application Interface

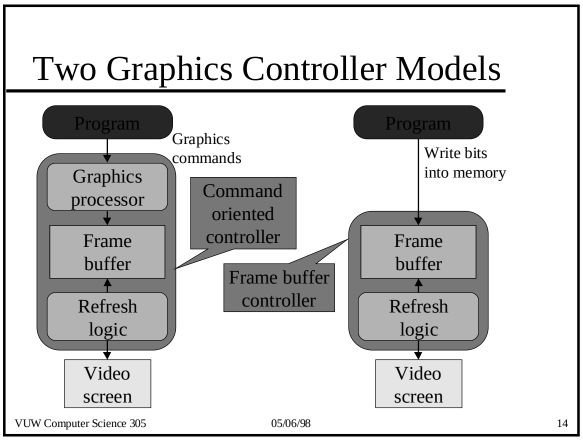


kernel					
kernel I/O subsystem					
SCSI device driver	keyboard device driver	mouse device driver	PCI bus device driver	floppy device driver	ATAPI device driver
SCSI device controller	keyboard device controller	mouse device controller	PCI bus device controller	floppy device controller	ATAPI device controller
SCSI devices	keyboard	mouse	PCI bus	floppy disk drives	disk & tape drives

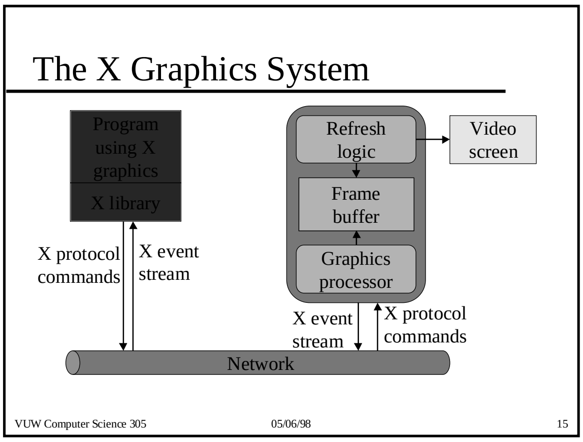


Characteristics

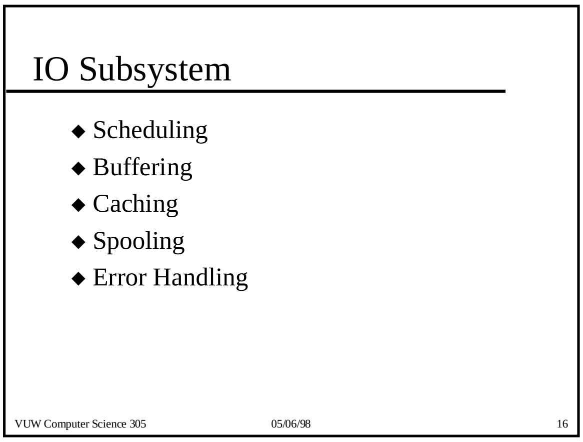
Aspect	Variation	Example



Two Graphics Controller Models

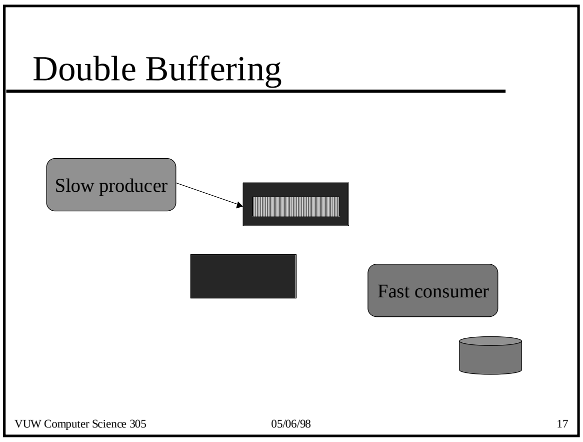


The X Graphics System

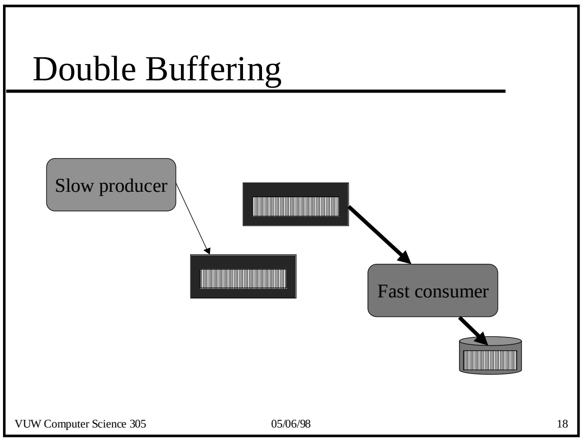


IO Subsystem

- Scheduling
- Buffering
- Caching
- Spooling
- Error Handling

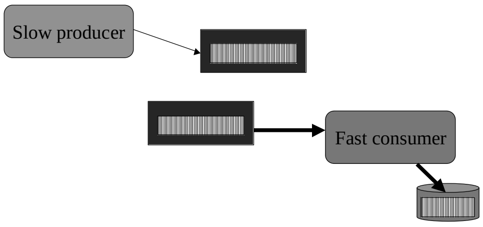


Double Buffering



Double Buffering

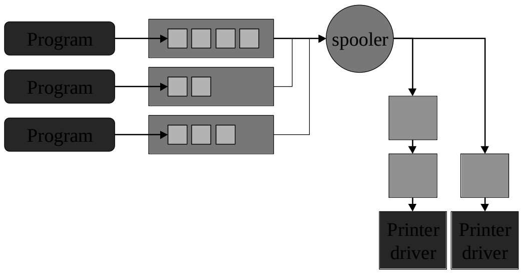
Double Buffering



IO Subsystem

- Scheduling
- Buffering
- Caching
- Spooling
- Error Handling

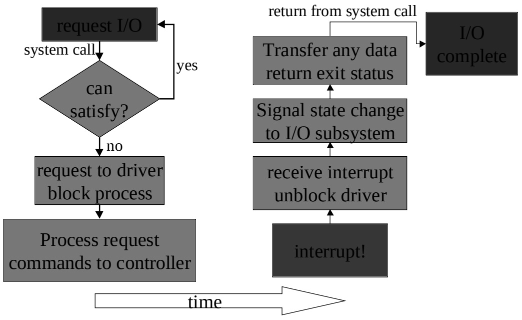
Spooling



IO Subsystem

- Scheduling
- Buffering
- Caching
- Spooling
- Error Handling

Making it Happen


request I/O
system call
can satisfy?
yes
no
request to driver block process
Process request commands to controller
interrupt!
receive interrupt unblock driver
Signal state change to I/O subsystem
Transfer any data return exit status
return from system call
I/O complete
time

Performance Issues

- Context switches
- Network IPC
- Interrupt "readies" a process
- Encourages big kernel

How to Get Efficient I/O

- Minimise context switches
  - Minimise interrupts (polling, large transfers)
- Minimise data copying
- Increase concurrency - smart controllers
- Move processing into IO hardware
- Balance CPU, memory, bus and IO use

Design Techniques

- Caching techniques
- Optimisation and Hints
- Names

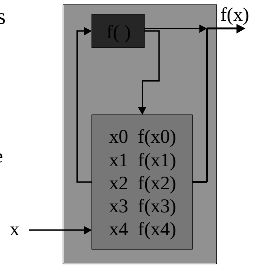
Example Problem: Network File Access

- Idempotent transactions
- Remote invocation

  <read, pathname, offset, length>
- Requires
  - Multiple directory lookups
  - Read inode or similar
  - Read data block

Caching

- Remember previous result of operation
  - Virtual memory
  - TLB
  - Path name look up
  - File descriptor table
  - Disk block caches



Issues With Caching

- Searching for cached results
- Cache invalidation
  - Hooks
- Cache replacement
- Locality of operation

Optimisation and Hints

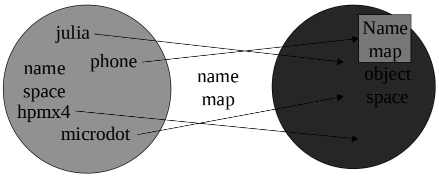
- Optimisation depends on predicting the future
- Approaches to optimisation:
  - Every instance
    - » Fixing a linear search by switching to a binary search (always faster)
  - Optimise some instances
    - » Caching results (sometimes faster)

Hints

- Hint: Answer that may be correct, but isn’t guaranteed
- Examples:
  - Name address binding
  - Resource location
- Difference from caching
  - Try hint to determine correctness

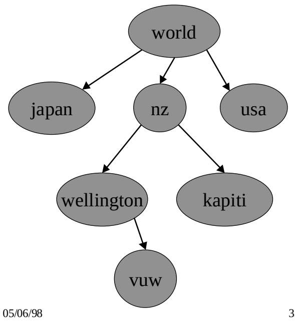
Hierarchical Names

- Name = identifier
- Object - to be identified
- Map: Names onto objects



Name Hierarchy

- Examples
  - Domain names
  - IP addresses
  - Files
  - X.500



Types of Names

- External
  - Character strings
  - User friendly
- Internal
  - Bit string
  - May contain hint to location
- Unique