COMP3211 03S2 Lecture 13

I/O Systems

Adapted from

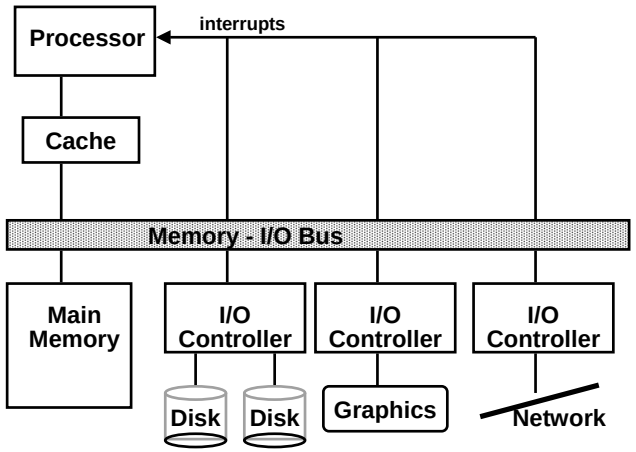
CS152: Computer Architecture and Engineering
Dave Patterson (www.cs.berkeley.edu/~pattrsn)

Key Points for This Lecture

- **I/O Performance Measures**
  - **Throughput & Response Time (Latency)**
- **Magnetic Disk Characteristics & Performance**
  - **Reliability & Availability**
- **I/O Access Schemes**
  - **Commands: instructions vs memory-mapped I/O**
  - **Status: polling vs interrupt**
  - **Delegating I/O responsibility: DMA & IOP**
- **OS Interaction**
  - **Shared resource: protection, scheduling**
  - **Provision of device drivers**
- **Summary**

I/O System Design Issues

- **Performance**
- **Expandability**
- **Resilience in the face of failure**



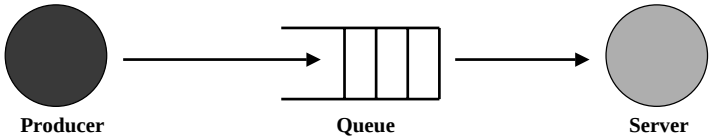
I/O Device Examples

Device	Behavior	Partner	Data Rate (KB/sec)
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Line Printer	Output	Human	1.00
Floppy disk	Storage	Machine	100.00
Laser Printer	Output	Human	200.00
Optical Disk	Storage	Machine	1,000.00
Magnetic Disk	Storage	Machine	10,000.00
Network-LAN	Input or Output	Machine	20 – 100,000.00
Graphics Display	Output	Human	100,000.00

I/O System Performance

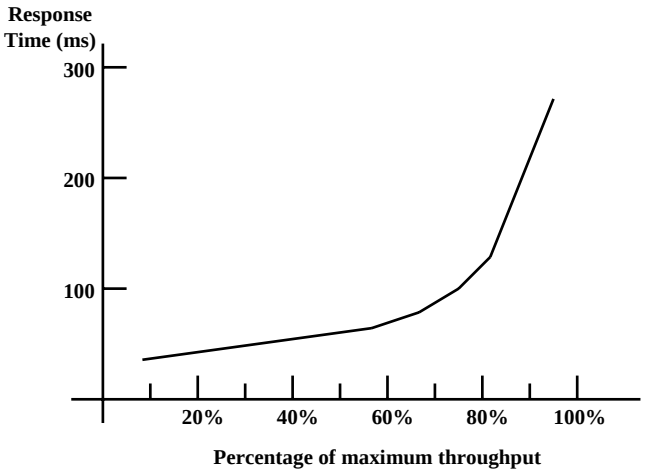
- I/O System performance depends on many aspects of the system ("limited by weakest link in the chain"):
  - The CPU
  - The memory system:
    - Internal and external caches
    - Main Memory
  - The underlying interconnection (buses)
  - I/O controllers
  - I/O devices
  - The speed of the I/O software (Operating System)
  - The efficiency of the software's use of the I/O devices
- Challenge to design & build a balanced system given current technology
  - Match differing speeds
  - Provide capacity to cope with load
  - Utilize the capacity
- Two common performance metrics:
  - Throughput: I/O bandwidth
  - Response time: Latency

Simple Producer-Server Model

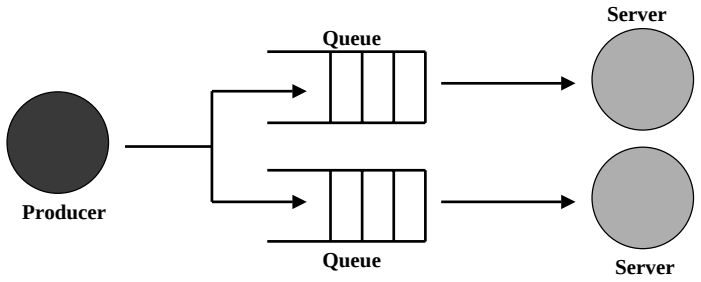


- Throughput:
  - The number of tasks completed by the server in unit time
  - In order to get the highest possible throughput:
    - The server should never be idle
    - The queue should never be empty
- Response time:
  - Begins when a task is placed in the queue
  - Ends when it is completed by the server
  - In order to minimize the response time:
    - The queue should be empty
    - The server should be idle

Throughput versus Respond Time



Throughput Enhancement

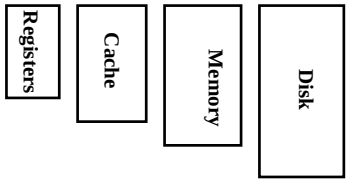


- In general throughput can be improved by:
  - Throwing more hardware at the problem
  - reduces load-related latency
- Response time is much harder to reduce:
  - Ultimately it is limited by the speed of light (but we're far from it)
  - Need to optimise architecture/improve implementation technology

I/O Benchmarks for Magnetic Disks

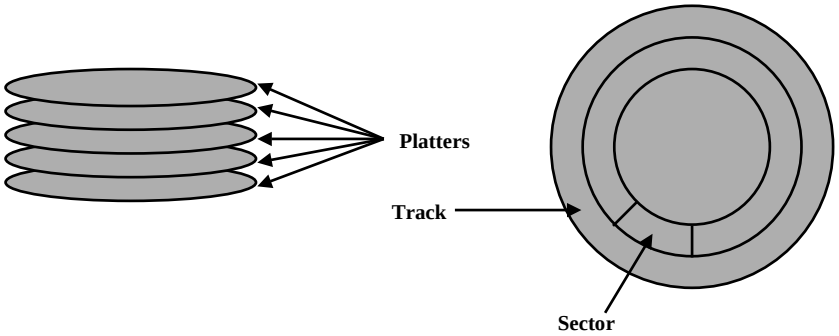
- Supercomputer application:
  - Large-scale scientific problems => large files
  - One large read and many small writes to snapshot computation
  - Data Rate: MB/second between memory and disk
- Transaction processing:
  - Examples: Airline reservations systems and bank ATMs
  - Small changes to large shared software
  - I/O Rate: No. disk accesses / second given upper limit for latency
- File system:
  - Measurements of UNIX file systems in an engineering environment:
    - 80% of accesses are to files less than 10 KB
    - 90% of all file accesses are to data with sequential addresses on the disk
    - 67% of the accesses are reads, 27% writes, 6% read-write
  - I/O Rate & Latency: No. disk accesses /second and response time

Magnetic Disk



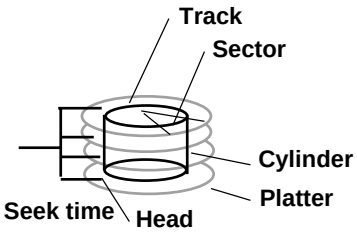
- Purpose:
  - Long term, nonvolatile storage
  - Large, inexpensive, and slow
  - Lowest level in the memory hierarchy
- Two major types:
  - Floppy disk
  - Hard disk
- Both types of disks:
  - Rely on a rotating platter coated with a magnetic surface
  - Use a moveable read/write head to access the disk
- Advantages of hard disks over floppy disks:
  - Platters are more rigid (metal or glass) so they can be larger
  - Higher density because it can be controlled more precisely
  - Higher data rate because it spins faster
  - Can incorporate more than one platter

Organization of a Hard Magnetic Disk



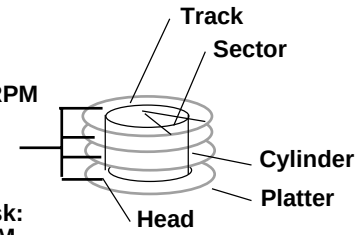
- Typical numbers (depending on the disk size):
  - 5000 to 30,000 tracks per surface
  - 100 to 500 sectors per track
    - A sector is the smallest unit that can be read or written
- Traditionally all tracks have the same number of sectors:
  - Constant bit density: record more sectors on the outer tracks

Magnetic Disk Characteristic



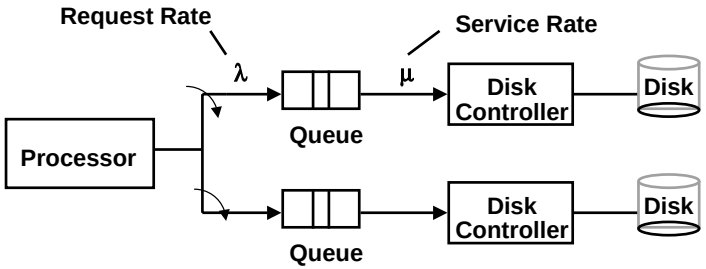
- Cylinder: all the tacks under the head at a given point on all surface
- Read/write data is a three-stage process:
  1. Position the arm over the proper track: Seek time
  2. Wait for the desired sector to rotate under the read/write head: Rotational latency
  3. Transfer a block of bits (sector) under the read-write head: Transfer time
- Average seek time as reported by the industry:
  - Typically in the range of 5 ms to 12 ms
  - (Sum of the time for all possible seek) / (total # of possible seeks)
- Due to locality of disk reference, actual average seek time may:
  - Only be 25% to 33% of the advertised number

Typical Numbers of a Magnetic Disk



- Rotational Latency:
  - Most disks rotate at 3,600 to 15,000 RPM
  - Approximately 16 ms to 0.4 ms per revolution, respectively
  - An average latency to the desired information is halfway around the disk: 8 ms at 3600 RPM, 2 ms at 15,000 RPM
- Transfer Time is a function of :
  - Transfer size (usually a sector): 1 KB / sector
  - Rotation speed: 3600 RPM to 15000 RPM
  - Recording density: bits per inch on a track
  - Diameter typical diameter ranges from 1.0 to 3.5 in
  - Typical transfer rates: 3 to 65 MB per second
  - Typical transfer times: 0.01 to 0.33 ms per sector

Disk I/O Performance



- Disk Access Time = Seek time + Rotational Latency + Transfer time + Controller Time + Queuing Delay
- Estimating Queue Length:
  - Utilization = U = Request Rate / Service Rate
  - Mean Queue Length = $U^2 / (1 - U)$ [see H&P, p.726]
  - As Request Rate $\rightarrow$ Service Rate
    - Mean Queue Length $\rightarrow$ Infinity

Example

- 512 byte sector, rotate at 5400 RPM, advertised seeks is 12 ms, transfer rate is 4 MB/sec, controller overhead is 1 ms, queue idle so no service time
- Disk Access Time = Seek time + Rotational Latency + Transfer time + Controller Time + Queuing Delay
- Disk Access Time = 12 ms + 0.5 / 5400 RPM + 0.5 KB / 4 MB/s + 1 ms + 0
- Disk Access Time = 12 ms + 0.5 / 90 RPS + 0.125 / 1024 s + 1 ms + 0
- Disk Access Time = 12 ms + 5.5 ms + 0.1 ms + 1 ms + 0 ms
- Disk Access Time = 18.6 ms
- If real seeks are 1/3 advertised seeks, then its 10.6 ms, with rotation delay contributing to 50% of the time!

Magnetic Disk Futures?

- Disk industry concentrates on increasing *areal density* measured in bits per square inch
- Past improvements in areal density

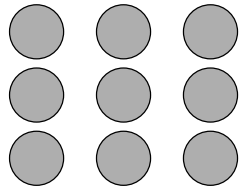
Year	Improve/yr	Doubling time
< 1988	29%	3 yrs
< 1996	60%	1.5 yrs
< 2001	100%	1 yr

- Cost per gigabyte has dropped at least as fast as areal density has increased – cost has decreased logarithmically from approximately $100,000/GB in 1984 to $10/GB in 2004
- Because it is more efficient to spin smaller mass, smaller diameter disks save power as well as volume
  - Large drive speeds have fallen from 3600 RPM in the 1980's to 7200 RPM in the 90's and 15,000 RPM around 2000
  - Combined with increased densities, this has lead to a 40% improvement in transfer rates per year; seek times drop <10%/year

Reliability and Availability

- Two terms that are often confused:
  - Reliability: Is anything broken?
  - Availability: Is the system still available to the user?
- Availability can be improved by adding hardware:
  - Example: adding ECC on memory
- Reliability can only be improved by:
  - Bettering environmental conditions
  - Building more reliable components
  - Building with fewer components
    - Improving availability may thus come at the cost of lower reliability

Disk Arrays



- A new organization of disk storage:
  - Arrays of small and inexpensive disks
  - Increase potential throughput by having many disk drives:
    - Data is spread over multiple disks
    - Multiple accesses are made to several disks
  - Latency not necessarily improved
- Reliability is lower than a single disk:
  - But availability can be improved by adding redundant disks (RAID): Lost information can be reconstructed from redundant information
  - MTTR: mean time to repair is in the order of hours
  - MTTF: mean time to failure of disks is tens of years

RAID Levels [Patterson, Gibson, and Katz, 1987]

RAID	Level	Minimum number of disk faults survived	Example data disks	Corresponding check disks	Usage
0	Nonredundant striped	0	8	0	Widely used
1	Mirrored	1	8	8	EMC, Compaq (Tandem), IBM
2	Memory-style ECC	1	8	4	
3	Bit-interleaved parity	1	8	1	Storage Concepts
4	Block-interleaved parity	1	8	1	Network Appliance
5	Block-interleaved distributed parity	1	8	1	Widely used
6	P + Q redundancy	2	8	2	

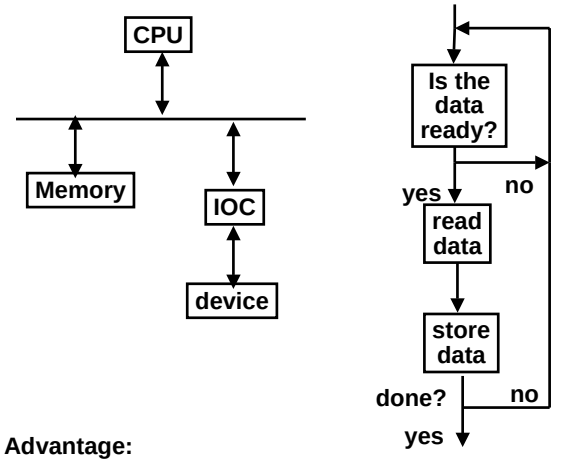
Giving Commands to I/O Devices

- Two methods are used to address a device:
  - Special I/O instructions
  - Memory-mapped I/O
- Special I/O instructions specify:
  - Both the device number and the command word
    - Device number: the processor communicates this via a set of control lines normally included as part of the I/O bus
    - Command word: this is usually send on the bus's data lines
- Memory-mapped I/O:
  - More commonly used
  - Portions of the address space are assigned to I/O device
  - Reads and writes to those addresses are interpreted as commands to the I/O devices
  - Can use memory protection mechanisms to prevent user from accessing I/O device directly

I/O Device Notifying the OS

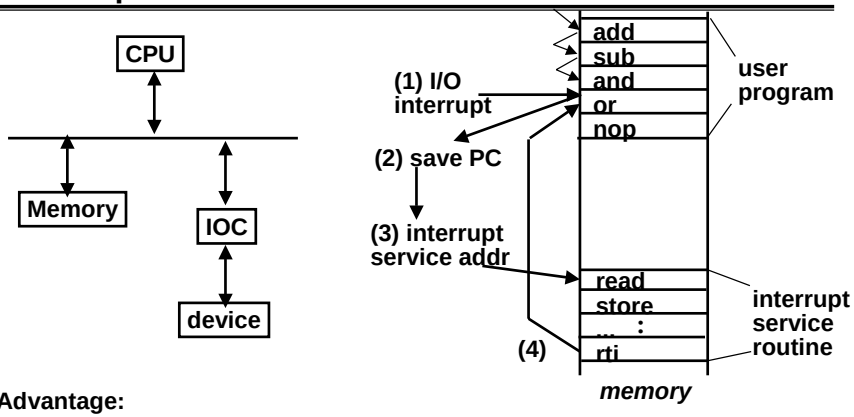
- The OS needs to know when:
  - The I/O device wants to initiate an operation
  - The I/O device has completed an operation
  - The I/O operation has encountered an error
- This can be accomplished in two different ways:
  - Polling:
    - The I/O device puts information in a status register
    - The OS periodically checks the status register
  - I/O Interrupt:
    - Whenever an I/O device needs attention from the processor, it interrupts the processor from what it is currently doing.

Polling: Programmed I/O



- Advantage:
  - Simple: the processor is totally in control and does all the work
- Disadvantage:
  - Polling overhead can consume a lot of CPU time

Interrupt Driven Data Transfer



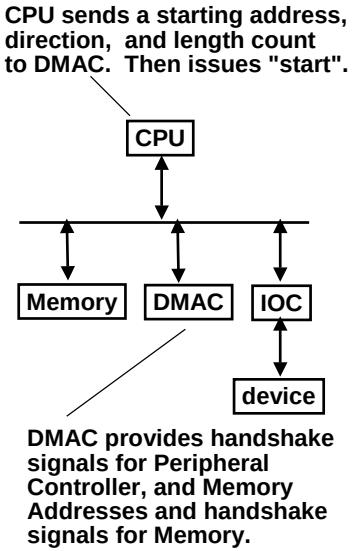
- Advantage:
  - User program progress is only halted during actual transfer
- Disadvantage, special hardware is needed to:
  - Cause an interrupt (I/O device)
  - Detect an interrupt (processor)
  - Save the proper states to resume after the interrupt (processor)

I/O Interrupt

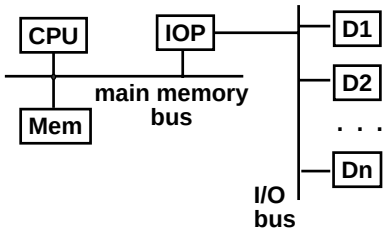
- An I/O interrupt is just like an exception except:
  - An I/O interrupt is asynchronous
  - Further information needs to be conveyed
- An I/O interrupt is asynchronous with respect to instruction execution:
  - I/O interrupt is not associated with any instruction
  - I/O interrupt does not prevent any instruction from completion
    - You can pick your own convenient point to take an interrupt
- I/O interrupt is more complicated than exception:
  - Needs to convey the identity of the device generating the interrupt
  - Interrupt requests can have different urgencies:
    - Interrupt request needs to be prioritized

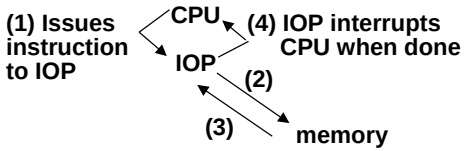
Delegating I/O Responsibility from the CPU: DMA

- Direct Memory Access (DMA):
  - External to the CPU
  - Act as a maser on the bus
  - Transfer blocks of data to or from memory without CPU intervention



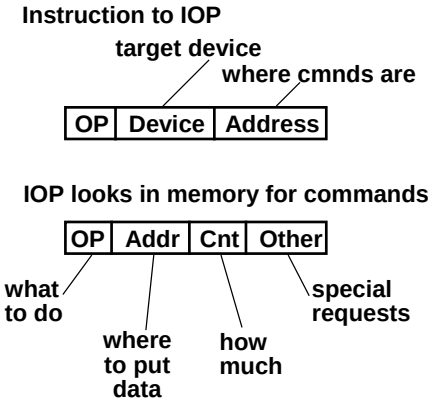
Delegating I/O Responsibility from the CPU: IOP





Device to/from memory transfers are controlled by the IOP directly.

IOP steals memory cycles.



Responsibilities of the Operating System

- The operating system acts as the interface between:
  - The I/O hardware and the programs that request I/O
- Three characteristics of I/O systems determine OS responsibilities:
  - The I/O system is shared by multiple programs using the processor
  - I/O systems often use interrupts (externally generated exceptions) to communicate information about I/O operations.
    - Interrupts must be handled by the OS because they cause a transfer to supervisor mode
  - The low-level control of an I/O device is complex:
    - Managing a set of concurrent events
    - The requirements for correct device control are very detailed

Operating System Requirements

- Handle the interrupts generated by I/O devices
- Provide protection to shared I/O resources
  - Guarantees that a user's program can only access the portions of an I/O device to which the user has rights
- Schedule accesses in order to enhance system throughput
- Provide fair access to the shared I/O resources
- Provide abstraction for accessing devices:
  - Supply routines that handle low-level device operation

I/O System Summary

- I/O performance is limited by weakest link in chain between OS and device
- Disk I/O Benchmarks: I/O rate vs. Data rate vs. latency
- Three Components of Disk Access Time:
  - Seek Time: advertised to be 5 to 12 ms. May be lower in real life.
  - Rotational Latency: 2 ms at 15000 RPM and 8.3 ms at 3600 RPM
  - Transfer Time: 0.01 – 0.33 ms per sector
- I/O device notifying the operating system:
  - Polling: it can waste a lot of processor time
  - I/O interrupt: similar to exception except it is asynchronous
- Delegating I/O responsibility from the CPU: DMA, or even IOP
- I/O imposes responsibilities on the OS