Input/Output

Instructors:
Alvin R. Lebeck

Some Slides provided by Randy Bryant and Dave O'Hallaron and G. Kedem

Administrivia

- HW #7 up, Due Monday April 25 (no late submissions)
- Work on Y86 simulator, Due April 27 (no late submissions)
- Final Exam: Friday May 6, 7pm-10pm
Overview

- I/O devices
  - device controller
- Device drivers
- Memory Mapped I/O
- Programmed I/O
- Direct Memory Access (DMA)
- Rotational media (disks)

Why I/O?

- Interactive Applications (keyboard, mouse, screen)
- Long term storage (files, data repository)
- Swap for VM
- Many different devices
  - character vs. block
  - Networks are everywhere!
- $10^6$ difference CPU ($10^{-9}$) & I/O ($10^{-3}$)
- Response Time vs. Throughput
  - Not always another process to execute
- OS hides (some) differences in devices
  - same (similar) interface to many devices
- Permits many apps to share one device
## I/O Device Examples

<table>
<thead>
<tr>
<th>Device</th>
<th>Behavior</th>
<th>Partner</th>
<th>Data Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>Input</td>
<td>Human</td>
<td>0.01 (KB/s)</td>
</tr>
<tr>
<td>Mouse</td>
<td>Input</td>
<td>Human</td>
<td>0.02 (KB/s)</td>
</tr>
<tr>
<td>Laser Printer</td>
<td>Output</td>
<td>Human</td>
<td>100.00 (KB/s)</td>
</tr>
<tr>
<td>Optical Disk</td>
<td>Storage</td>
<td>Machine</td>
<td>500.00 (KB/s)</td>
</tr>
<tr>
<td>Magnetic Disk</td>
<td>Storage</td>
<td>Machine</td>
<td>60.00 (MB/s)</td>
</tr>
<tr>
<td>USB 1.1</td>
<td>Input/Output</td>
<td>Machine</td>
<td>1.50 (MB/s)</td>
</tr>
<tr>
<td>USB 2.0</td>
<td>Input/Output</td>
<td>Machine</td>
<td>40.00 (MB/s)</td>
</tr>
<tr>
<td>Flash drive</td>
<td>I/O device</td>
<td>Machine</td>
<td>30.00 (MB/s)</td>
</tr>
<tr>
<td>Network-LAN</td>
<td>Input/Output</td>
<td>Machine</td>
<td>1.00 (Gb/s)</td>
</tr>
<tr>
<td>Graphics Display</td>
<td>Output</td>
<td>Human</td>
<td>8.00 (GB/s)</td>
</tr>
<tr>
<td>L2 cache bandwidth</td>
<td></td>
<td></td>
<td>~50.00 (GB/s)</td>
</tr>
</tbody>
</table>

## I/O Systems

\[
\text{Time(workload)} = \text{Time(CPU)} + \text{Time(I/O)} - \text{Time(Overlap)}
\]

[Diagram of I/O Systems]

- Processor
- Cache
- Memory Bus
- Core Chip Set
- I/O Bridge
- I/O Bus
- Main Memory
- Disk Controller
- Graphics Controller
- Network Interface
- Disk
- Graphics
- Network

---

Duke University

*Compsci 104*
Device Drivers

- Software that provides access to devices (mp3, camera, phone, FLASH drive)
- **top-half**
  - Application Programming Interface (API): e.g., open, close, read, write, ioctl
  - I/O Control: e.g., IOCTL, device specific arguments
- **bottom-half**
  - interrupt handler
  - communicates with device
  - resumes process
- Must have access to user address space and device control registers => runs in kernel mode.

Review: Handling an Interrupt/Exception

- Invoke specific **kernel** routine based on type of interrupt
  - interrupt/exception handler
- Must determine what caused interrupt
- Clear the interrupt
- Return from interrupt (iret)
Processor <-> Device Interface Issues

- Interconnections
  - Busses
- Processor interface
  - I/O Instructions
  - Memory mapped I/O
- I/O Control Structures
  - Device Controllers
  - Polling/Interrupts
- Data movement
  - Programmed I/O / DMA
- Capacity, Access Time, Bandwidth

---

Device Controllers

Controller deals with mundane control (e.g., position head, error detection/correction)
Processor communicates with Controller
I/O Instructions

Separate instructions (in, out)

Memory Mapped I/O

- Issue command through store instruction
- Check status with load instruction
  - Caches?

Physical Address

<table>
<thead>
<tr>
<th>ROM</th>
<th>RAM</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Communicating with the processor

- **Polling**
  - can waste time waiting for slow I/O device
  - busy wait
  - can interleave with useful work

- **Interrupts**
  - interrupt overhead
  - interrupt could happen anytime - asynchronous
  - no busy wait

Data Movement

- **Programmed I/O**
  - processor has to touch all the data
  - too much processor overhead
    - for high bandwidth devices (disk, network)

- **DMA**
  - processor sets up transfer(s)
  - DMA controller transfers data
  - complicates memory system
Programmed I/O & Polling

- **Advantage:** CPU totally in control

- **Disadvantage:** Overhead of polling
  - Program must perform check of device, thus can’t do useful work

Is the data ready?

- yes
  - load data
  - store data

- done?
  - no
  - Is the data ready?
  - yes

Programmed I/O & Interrupt Driven Data Transfer

- User program progress halted only during actual transfer
- Interrupt overhead can dominate transfer time
- Processor must touch all data...too slow for some devices

1. I/O interrupt
2. save PC
3. interrupt service addr
4. rti
Direct Memory Access (DMA)

CPU sends a starting address, direction, and length count to DMAC. Then issues "start".

- CPU delegates responsibility for data transfer to a special controller

DMAC provides handshake signals for device controller, and memory addresses and handshake signals for memory.

I/O and Virtual Caches

I/O is accomplished with physical addresses DMA
- flush pages from cache
- need pa->va reverse translation
- coherent DMA
Other I/O Issues

- Multiple devices want to use the same bus (wires)?
- Arbitration
  - Daisy chain
  - Centralized
  - Distributed

Obtaining Access to the Bus

- One of the most important issues in bus design:
  - How is the bus reserved by a devices that wishes to use it?
- Chaos is avoided by a master-slave arrangement:
  - Only the bus master can control access to the bus:
    - It initiates and controls all bus requests
    - A slave responds to read and write requests
- The simplest system:
  - Processor is the only bus master
  - All bus requests must be controlled by the processor
  - Major drawback: the processor is involved in every transaction
Multiple Potential Bus Masters: the Need for Arbitration

- **Bus arbitration scheme:**
  - A bus master wanting to use the bus asserts the bus request
  - A bus master cannot use the bus until its request is granted
  - A bus master must signal to the arbiter after finish using the bus

- **Bus arbitration schemes usually try to balance two factors:**
  - **Bus priority:** the highest priority device should be serviced first
  - **Fairness:** Even the lowest priority device should never be completely locked out from the bus

- **Bus arbitration schemes can be divided into four broad classes:**
  - **Distributed arbitration by self-selection:** each device wanting the bus places a code indicating its identity on the bus.
  - **Distributed arbitration by collision detection:** Ethernet uses this.
  - **Daisy chain arbitration:** single device with all request lines.
  - **Centralized, parallel arbitration:** see next-next slide

---

**Centralized Bus Arbitration**

![Centralized Bus Arbitration Diagram]

Arbitration Circuit

- **Request-C**
- **Grant-C**
- **Request-B**
- **Grant-B**
- **Request-A**
- **Grant-A**
- **Release**

Master A

Master B

Master C
The Daisy Chain Bus Arbitration Scheme

- **Advantage:** simple
- **Disadvantages:**
  - Cannot assure fairness:
    - A low-priority device may be locked out indefinitely
  - The use of the daisy chain grant signal also limits the bus speed

Types of Storage Devices

- Magnetic Disks
- FLASH Drive (USB Stick, SSD)
- Magnetic Tapes
- CD/DVD
- Juke Box (automated tape library, robots)
What’s Inside A Disk Drive?

Image courtesy of Seagate Technology

Disk Geometry

- Disks consist of platters, each with two surfaces.
- Each surface consists of concentric rings called tracks.
- Each track consists of sectors separated by gaps.
Disk Geometry (Muliple-Platter View)

- Aligned tracks form a cylinder.

Disk Capacity

- **Capacity**: maximum number of bits that can be stored.
  - Vendors express capacity in units of gigabytes (GB), where 1 GB = $10^9$ Bytes (< $2^{20}$ Bytes, be careful)

- **Capacity is determined by these technology factors:**
  - Recording density (bits/in): number of bits that can be squeezed into a 1 inch segment of a track.
  - Track density (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment.
  - Areal density (bits/in$^2$): product of recording and track density.

- **Modern disks partition tracks into disjoint subsets called recording zones**
  - Each track in a zone has the same number of sectors, determined by the circumference of innermost track.
  - Each zone has a different number of sectors/track
Computing Disk Capacity

Capacity = (# bytes/sector) x (avg. # sectors/track) x
(# tracks/surface) x (# surfaces/platter) x
(# platters/disk)

Example:
- 512 bytes/sector
- 300 sectors/track (on average)
- 20,000 tracks/surface
- 2 surfaces/platter
- 5 platters/disk

Capacity = 512 x 300 x 20000 x 2 x 5
= 30,720,000,000
= 30.72 GB

Disk Operation (Single-Platter View)

The disk surface spins at a fixed rotational rate

The read/write head is attached to the end of the arm and flies over the disk surface on a thin cushion of air.

By moving radially, the arm can position the read/write head over any track.
Disk Operation (Multi-Platter View)

Read/write heads move in unison from cylinder to cylinder

Disk Structure - top view of single platter

Surface organized into tracks
Tracks divided into sectors
Disk Access

Head in position above a track

Disk Access

Rotation is counter-clockwise
Disk Access – Read

About to read blue sector

After BLUE read

After reading blue sector
Disk Access – Read

After BLUE read

Red request scheduled next

Disk Access – Seek

After BLUE read

Seek for RED

Seek to red's track
Disk Access – Rotational Latency

After **BLUE** read  
Seek for **RED**  
Rotational latency

Wait for red sector to rotate around

Disk Access – Read

After **BLUE** read  
Seek for **RED**  
Rotational latency  
After **RED** read

Complete read of red
Disk Access – Service Time Components

- After **BLUE** read
- Seek for **RED**
- Rotational latency
- After **RED** read

Data transfer
Seek
Rotational latency
Data transfer

Disk Access Time

- **Average time to access some target sector approximated by** :
  - \( T_{\text{access}} = T_{\text{avg seek}} + T_{\text{avg rotation}} + T_{\text{avg transfer}} \)
- **Seek time** (**Tavg seek**) :
  - Time to position heads over cylinder containing target sector.
  - Typical \( T_{\text{avg seek}} \) is 3—9 ms
- **Rotational latency** (**Tavg rotation**) :
  - Time waiting for first bit of target sector to pass under r/w head.
  - \( T_{\text{avg rotation}} = \frac{1}{2} \times \frac{1}{\text{RPMs}} \times 60 \text{ sec/1 min} \)
  - Typical \( T_{\text{avg rotation}} = 7200 \text{ RPMs} \)
- **Transfer time** (**Tavg transfer**) :
  - Time to read the bits in the target sector.
  - \( T_{\text{avg transfer}} = \frac{1}{\text{RPM}} \times \frac{1}{\text{avg # sectors/track}} \times 60 \text{ secs/1 min} \)
### Disk Access Time Example

- **Given:**
  - Rotational rate = 7,200 RPM
  - Average seek time = 9 ms.
  - Avg # sectors/track = 400.

- **Derived:**
  - \( \text{Tavg rotation} = \frac{1}{2} \times (60 \text{ secs}/7200 \text{ RPM}) \times 1000 \text{ ms/sec} = 4 \text{ ms.} \)
  - \( \text{Tavg transfer} = \frac{60}{7200} \text{ RPM} \times \frac{1}{400} \text{ secs/track} \times 1000 \text{ ms/sec} = 0.02 \text{ ms} \)
  - \( \text{Taccess} = 9 \text{ ms} + 4 \text{ ms} + 0.02 \text{ ms} \)

- **Important points:**
  - Access time dominated by seek time and rotational latency.
  - First bit in a sector is the most expensive, the rest are free.
  - SRAM access time is about 4 ns/doubleword, DRAM about 60 ns
    - Disk is about 40,000 times slower than SRAM,
    - 2,500 times slower then DRAM.

### Logical Disk Blocks

- **Modern disks present a simpler abstract view of the complex sector geometry:**
  - The set of available sectors is modeled as a sequence of b-sized logical blocks (0, 1, 2, ...)

- **Mapping between logical blocks and actual (physical) sectors**
  - Maintained by hardware/firmware device called disk controller.
  - Converts requests for logical blocks into (surface,track,sector) triples.

- **Allows controller to set aside spare cylinders for each zone.**
  - Accounts for the difference in “formatted capacity” and “maximum capacity”.

**I/O Bus**

CPU initiates a disk read by writing a command, logical block number, and destination memory address to a port (address) associated with disk controller.

**Reading a Disk Sector (1)**

CPU initiates a disk read by writing a command, logical block number, and destination memory address to a port (address) associated with disk controller.
Reading a Disk Sector (2)

Disk controller reads the sector and performs a direct memory access (DMA) transfer into main memory.

Reading a Disk Sector (3)

When the DMA transfer completes, the disk controller notifies the CPU with an interrupt (i.e., asserts a special "interrupt" pin on the CPU).
Solid State Disks (SSDs)

- Pages: 512KB to 4KB, Blocks: 32 to 128 pages
- Data read/written in units of pages.
- Page can be written only after its block has been erased
- A block wears out after 100,000 repeated writes.

SSD Performance Characteristics

<table>
<thead>
<tr>
<th></th>
<th>Sequential read tput</th>
<th>Sequential write tput</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>250 MB/s</td>
<td>170 MB/s</td>
</tr>
<tr>
<td>Random read tput</td>
<td>140 MB/s</td>
<td></td>
</tr>
<tr>
<td>Random write tput</td>
<td>14 MB/s</td>
<td></td>
</tr>
<tr>
<td>Rand read access</td>
<td>30 us</td>
<td></td>
</tr>
<tr>
<td>Random write access</td>
<td>300 us</td>
<td></td>
</tr>
</tbody>
</table>

Why are random writes so slow?
- Erasing a block is slow (around 1 ms)
  - Write to a page triggers a copy of all useful pages in the block
    - Find an used block (new block) and erase it
    - Write the page into the new block
    - Copy other pages from old block to the new block
SSD Tradeoffs vs Rotating Disks

- **Advantages**
  - No moving parts → faster, less power, more rugged

- **Disadvantages**
  - Have the potential to wear out
    - Mitigated by “wear leveling logic” in flash translation layer
    - E.g. Intel X25 guarantees 1 petabyte (10^{15} bytes) of random writes before they wear out
    - In 2010, about 100 times more expensive per byte

- **Applications**
  - MP3 players, smart phones, laptops
  - Beginning to appear in desktops and servers

---

**Storage Trends**

**SRAM**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB$</td>
<td>19,200</td>
<td>2,900</td>
<td>320</td>
<td>256</td>
<td>100</td>
<td>75</td>
<td>60</td>
<td>320</td>
</tr>
<tr>
<td>access (ns)</td>
<td>300</td>
<td>150</td>
<td>35</td>
<td>15</td>
<td>3</td>
<td>2</td>
<td>1.5</td>
<td>200</td>
</tr>
</tbody>
</table>

**DRAM**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB$</td>
<td>8,000</td>
<td>880</td>
<td>100</td>
<td>30</td>
<td>1</td>
<td>0.1</td>
<td>0.06</td>
<td>130,000</td>
</tr>
<tr>
<td>access (ns)</td>
<td>375</td>
<td>200</td>
<td>100</td>
<td>70</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>9</td>
</tr>
<tr>
<td>typical size (MB)</td>
<td>0.064</td>
<td>0.256</td>
<td>4</td>
<td>16</td>
<td>64</td>
<td>2,000</td>
<td>8,000</td>
<td>125,000</td>
</tr>
</tbody>
</table>

**Disk**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB$</td>
<td>500</td>
<td>100</td>
<td>8</td>
<td>0.30</td>
<td>0.01</td>
<td>0.005</td>
<td>0.0003</td>
<td>1,600,000</td>
</tr>
<tr>
<td>access (ms)</td>
<td>87</td>
<td>75</td>
<td>28</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>typical size (MB)</td>
<td>1</td>
<td>10</td>
<td>160</td>
<td>1,000</td>
<td>20,000</td>
<td>160,000</td>
<td>1,500,000</td>
<td>1,500,000</td>
</tr>
</tbody>
</table>
### CPU Clock Rates

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>8080</td>
<td>386</td>
<td>Pentium</td>
<td>P-III</td>
<td>P-4</td>
<td>Core 2</td>
<td>Core i7</td>
<td>---</td>
</tr>
</tbody>
</table>

- **Clock rate (MHz)**: 1, 20, 150, 600, 3300, 2000, 2500, 2500
- **Cycle time (ns)**: 1000, 50, 6, 1.6, 0.3, 0.50, 0.4, 2500
- **Cores**: 1, 1, 1, 1, 1
- **Effective cycle time (ns)**: 1000, 50, 6, 1.6, 0.3, 0.25, 0.1, 10,000

Inflection point in computer history when processors (chips) hit the "Power Wall"

### The CPU-Memory Gap

The gap widens between DRAM, disk, and CPU speeds.