Outline

• Overview
• IDE Disk Driver
• Physical Disks
• Disk Scheduling
• Flash Memory
PC Architecture

- Processor
  - CPU Core
  - CPU Core
- North Bridge
- South Bridge
  - CPU Core
- RAM
  - PCIeExpress
  - APIC bus
  - SATA
  - USB
  - Audio
  - ISA
What is memory?

- **SRAM – Static RAM**
  - Two NOT gates in a circular loop
  - 4 or 6 MOSFETs per bit
  - Fast used for caches

- **DRAM – Dynamic RAM**
  - 1 Capacitor + 1 MOSFET per bit
  - High density used for main memory
  - Requires periodic refresh to recharge

- **VRAM – Video RAM**
  - Multiported: Multiple readers/writers
What is an IO bus?

• Interconnect between devices
• Many kinds of busses: PCI, USB, SMBUS, etc.
• Bridges or Bus Controllers convert one to another
• Single Master (e.g. ISA)
  • North bridge can issue reads/writes/DMA to devices
• Multi-master (e.g. PCI)
  • Any device can issue reads/writes to main memory
Device Communication

• Port IO (in/out instructions in x86)
• Memory Mapped IO (PCI, PCIExpress)
• Interrupt Controller
• Direct Memory Access
Port IO

• Using `in/out` instructions
  • `uint8_t inb(uint16_t port);` // Read Byte
  • `void outb(uint16_t port, uint8_t data);` // Write Byte
  • Can read or write 1, 2, or 4 bytes

• Bulk transfer with `rep` prefix
  • `void insw(uint16_t port, void *buf, int len);`

• Older mechanism used for legacy devices
Memory Mapped IO

- Devices can map memory in for control and/or data
- 0xF00000-0xFFFFFFFF: ISA Memory Hole
- 0xC0000000-0xFFFFFFFF: PCI Memory Hole
- PCI Express creates a memory hole above 4 GiB

- Reads/Writes to this memory must be uncached
Interrupts

• Advanced Programmable Interrupt Controller
• Local APIC: Per-CPU
  • Deliver interrupts
  • APIC Timer provide timer interrupts
  • Starting up extra cores
  • Delivery of Inter-processor Interrupts (IPIs)
• IO APIC: Per-Machine (typically)
  • Routing interrupts to Local APIC
  • Global interrupt masking
  • Usually 24 interrupts
Direct Memory Access (DMA)

- ISA Device
  - Programmed by the OS
  - Limited to 4 or 8 DMA channels

- PCI Devices
  - Every device can DMA no central control
  - Control structures often read from main memory

- IOAT (Intel’s IO Acceleration Technology)
  - New dedicated DMA engine for bulk data copy
  - Used by the OS often for network acceleration
  - Other vendors have supported extra features
Device Driver Architecture

- Driver provides several entry points to kernel
  - Reset, ioctl, read/write/strategy, interrupt, ...
- OS provides callback to entry points
  - Device File System
  - Interrupt Handlers
  - Storage IO Stack
Polling vs Interrupts

- **Polling**
  - Periodically check device for data
  - Uses CPU when device is idle or not doing much
  - Higher latency depending on poll interval

- **Interrupts**
  - A small handler is called on a CPU interrupt
  - Lower latency as CPU notified instantly
  - Uses less CPU for idle or low throughput devices

- **Downside: Interrupts can block the entire OS**
  - Receive Livelock
  - Interrupt Coalescing helps mitigate this
  - Dynamic switching between polling/interrupt mode
IDE Disk Driver
IDE Driver

- IDE or PATA is the older parallel driver interface
- IDE Controllers usually accessed through port IO
- Data transfer through DMA or PIO
- Used with or without interrupts
- Two disks per-controller
PIO IDE Driver

IDE_ReadSector(int disk, int off, void *buf) {
    outb(0x1F6, disk == 0 ? 0xE0 : 0xF0); // Select Drive
    IDEWait();
    outb(0x1F2, 512); // Read length (512 B)
    outb(0x1F3, off); // LBA Low
    outb(0x1F4, off >> 8); // LBA Mid
    outb(0x1F5, off >> 16); // LBA High
    outb(0x1F7, 0x20); // Read Command
    insw(0x1F0, buf, 256); // Read 256 Words
}
PIO IDE Driver Con’t

void IDEWait() {
    // Discard status 4 times
    inb(0x1F7); inb(0x1F7);
    inb(0x1F7); inb(0x1F7);
    // Wait for status BUSY flag to clear
    while (((inb(0x1F7) & 0x80) != 0)
    {
    
    
    
}
ATA Commands

• ATA Commands
  • PIO READ (0x20), PIO WRITE (0x30)
  • DMA READ/WRITE
  • FLUSH: Flush disk cache
  • IDENTIFY: Return device specification/features
  • SMART: Drive Diagnostics/Statistics
  • SLEEP/STANDBY: Power Management

• ATAPI Commands:
  • PACKET, IDENTIFY PACKET: For other devices (CDs)
Serial ATA (SATA)

- Serial ATA is the modern disk interface
- Disks compatible with Serial Attached SCSI (SAS)
- Introduces new useful features
  - NCQ, Hotplug, Port Multipliers (PM)
- One device per-port, or multiplexed with PM
- AHCI (Most common controller standard)
  - PCI Device
  - Uses memory mapped IO for issuing IOs
  - Up to 32 ports (no current device has more than 8)
Physical Disks
Anatomy of Disk [Ruemmler]

• Magnetic Platters
  • Multiple magnetic platters
  • Rotate together between 3600-15000 RPM
  • Drive speed not constant, position not certain

• Disk arm
  • Rotates around a pivot together
  • One disk head per recording surface (2xPlatters)
  • Head flying height 3 nm over disk surface
    • Air creates a natural cushion
  • Sensitive to motion and vibration
Disks
Storage Addressing

- Platters divided into concentric *tracks*
- Tracks of a fixed radius form a *cylinder*
- Tracks contain many sectors
- Heads read/write data along cylinders
  - Head adjustment requires reading one platter at a time
  - Significant space for synchronization/error correction
  - Per-sector data previously 512 B, now usually 4 KB
- CHS Addressing: Cylinder-Head-Sector
- LBA Addressing: Logical Block Address
Cylinders, tracks, and sectors

- track $t$
- sector $s$
- cylinder $c$
- platter
- arm
- spindle
- arm assembly
- rotation
- read-write head
Disk Latency

• **Seek Time**
  • Time to move head into position
  • Depends on head mechanism

• **Rotational Latency**
  • Time it takes for disk platter to rotate into position
  • Depends on rotational speed

• **Transfer Time**
  • Time to read/write actual data
  • Depends on interface, rotational speed, etc

• **Average Latency:** 2 ms (server) – 15 ms (laptops)

• **Other Sources**
  • OS/Controller add additional latency
  • Usually small except for flash storage!
Noise and Drive Latency

Disk: I/O operations per second taking at least 521740 microseconds broken down by disk

Range average:

1 2029QTFO802GCK 21
1 2029QTFO802GCK 9
1 2029QTFO802GCK 5

Show hierarchy
58 ops per second

Disk: I/O bytes per second broken down by disk

Range average:

14.2G 2029QTFO802GCK 8
13.6G 2029QTFO802GCK 0
13.5G 2029QTFO802GCK 17

Show hierarchy
1.08G per second
Failures/Bad Sectors

- Drives detect bad sectors based on read issues
- Logical sector may be reallocated elsewhere
- If too many occur, drive errors visible to OS
- SMART allows you to see relocation events

Bad Reads:
- Data may decay (bit-rot)
- Disk will retry reading IO many times
- May take as long as 15 Minutes!
Disk Scheduling
Optimizing Disk Performance

- Disks involve physical movement and are slow
- Average seek time: 2 ms (server) – 15 ms (mobile)

- File Systems optimize layout (more next week)
  - Reading a file laid out linearly on disk
  - Placing fragments of files near each other

- Operating Systems often schedule IOs for speed
- Disks/Controllers reorder outstanding IOs
Disk Scheduling: FCFS

• First Come First Served (FCFS)
  • Process request in order received

• Advantages:
  • Easy to implement
  • Good fairness

• Disadvantages
  • Cannot exploit request locality
  • Increases average latency, decreasing throughput
FCFS Example

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
Disk Scheduling: SSTF

- Shortest Seek Time First (SSTF)
  - Always pick request with shortest seek time
  - Also called shortest positioning time first (SPTF)

- Advantages:
  - Exploits locality of disks
  - Higher throughput

- Disadvantages:
  - Poor fairness/Starvation
  - Don’t alwys know what request will be fastest
SSTF Example

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
Disk Scheduling: SCAN

• Elevator Scheduling (SCAN)
  • Pick request with shortest seek time
  • Switch direction at end of disk (or CSCAN loops linearly)

• Advantages:
  • Exploits locality of disks
  • Bounded waiting

• Disadvantages:
  • Might miss locality SPTF could exploit
CSCAN Example

queue  98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
SATA NCQ/SCSI TCQ

• Native/Tagged Command Queueing
• Allows 32 outstanding commands (queue length)
• Drive can reorder requests for performance
  • Using scheduling algorithms like we previously discussed
• NCQ is important for SSDs!
  • NCQ enables better hardware pipelining
• SCSI TCQ enables queue modes
  • Insert into Head of Queue, Enforce Order, Simple
Flash Memory
Flash Memory

• No moving parts
  • Stores data using charge
  • No mechanical seek times

• Limited writes
  • Blocks wear out after 10000 (MLC) – 100000 (SLC) erases
  • Requires flash translation layer (FTL) for wear leveling
  • FTL can impact performance and reliability!

• Limited durability
  • Charges wear out over time (10 – 100 years at SATP)
Types of Flash

• NAND flash
  • High Density
  • Faster erase/write

• NOR flash
  • Faster reads
  • Slower erases

• Single-level cell (SLC) vs. Multi-level cell (MLC)
  • MLC encodes multiple bits in voltages
  • MLC is slower to write
  • MLC has shorter durability (bits decay faster)
Flash Characteristics [Caulfield’09]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SLC</th>
<th>MLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density Per Die</td>
<td>4 GB</td>
<td>8 GB</td>
</tr>
<tr>
<td>Page Size</td>
<td>2048+32 Bytes</td>
<td>2048+64 Bytes</td>
</tr>
<tr>
<td>Block Size</td>
<td>64 Pages</td>
<td>128 Pages</td>
</tr>
<tr>
<td>Read Latency</td>
<td>25 us</td>
<td>25 us</td>
</tr>
<tr>
<td>Write Latency</td>
<td>200 us</td>
<td>800 us</td>
</tr>
<tr>
<td>Erase Latency</td>
<td>2000 us</td>
<td>2000 us</td>
</tr>
<tr>
<td>40 MHz read</td>
<td>75.8 MB/s</td>
<td>75.8 MB/s</td>
</tr>
<tr>
<td>40 MHz program</td>
<td>20.1 MB/s</td>
<td>5.0 MB/s</td>
</tr>
<tr>
<td>133 MHz read</td>
<td>126.4 MB/s</td>
<td>126.4 MB/s</td>
</tr>
<tr>
<td>133 MHz program</td>
<td>20.1</td>
<td>5.0 MB/s</td>
</tr>
</tbody>
</table>