UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 537 — A. Arpaci-Dusseau
Intro to Operating Systems — Spring 2000

# I/O Devices

## Questions answered in these notes

- What are the components of the I/O system?
- How do users interact with I/O devices?
- What can the kernel do to speed up I/O requests?
- How do device drivers interact with device controllers?
- What are the characteristics of modern disks?

## Readings for this topic: Chapter 12

# Course Overview

- Process Management
  - Synchronization
  - Scheduling
- Memory Management
  - Storage Allocation
  - Virtual Memory
- Disk Management
  - **I/O Devices**
  - File Systems
- Communication
- Security and Protection



# Motivation

## Two primary aspects of computer system

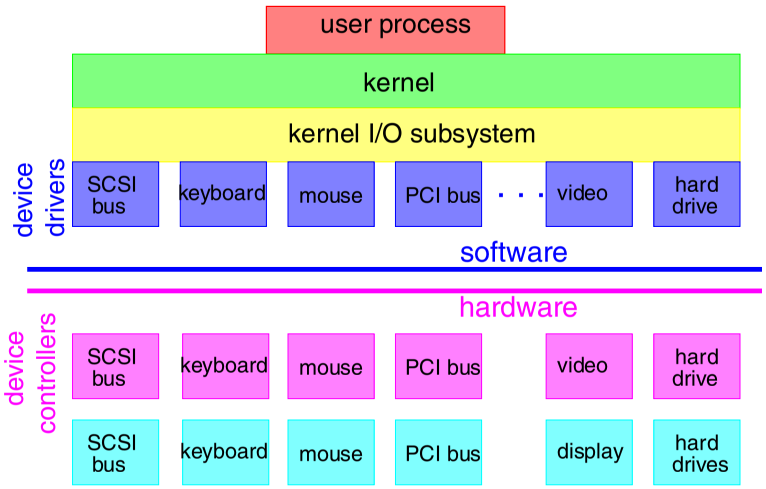
- Processing
- Input/Output

## Role of Operating System

- Provide a consistent interface
  - Simplify access to hardware devices
  - Implement mechanisms for interacting with devices
- Allocate and manage resources
  - Protection
  - Fairness
- Obtain efficient performance
  - Understand performance characteristics of device
  - Develop policies



# I/O Subsystem



# User View of I/O

## User processes cannot have direct access to devices

- Manage resources fairly
- Protects data from access-control violations
- Protect system from crashing

## Operating system exports higher-level functions

- User process performs system calls (e.g., read() and write())

## Blocking vs. Nonblocking I/O

- Blocking: Suspend execution of process until I/O complete
  - Not acceptable if have multiple interactive components
  - Structure application with multiple threads
- Nonblocking: Return from system call immediately
  - Notify when I/O completes in future



# User View: Types of Devices

## Character-stream

- Transfer one byte at a time (produce data “spontaneously”)
- Interface: get() or put()
- Example: keyboard, mice, modems

## Block

- Transfer a block of bytes as a unit
  - Variable or fixed-size transfers
- Interface: read() and write()
  - Random access: seek() specifies which byte to transfer next
- Example: disks and tapes



# Kernel I/O Subsystem

## I/O Scheduling from pool of requests

- Rearrange disk requests for efficiency
- Perform transfers that are near each other on disk

## Buffering

- Deal with different transfer rates
  - Double buffer: Producer and consumer switch between two buffers
- Adjust transfer sizes
  - Fragmentation and reassembly
- Copy semantics
  - Can calling process can reuse buffer immediately?

## Caching: Keep disk data in memory

- If locality, then next access to data is fast



# Device Drivers

## Encapsulate details of device

- Wide variety of I/O devices (different manufacturers)
- Kernel I/O subsystem not aware of device details

## Load at boot time or after running

## UNIX system call: ioctl (I/O control)

- Alternative to adding new system call
- Interface between user processes and device driver
  - file descriptor
  - integer: selects command implemented by device
  - pointer: an arbitrary data structure in memory

# Device Driver: Accessing Controller

## Special instructions

- Valid only in kernel mode
- No longer popular

## Memory-mapped

- Read and write to special memory addresses
- Protect by placing in kernel address space
- Map part of device to user address space for fast access

  Example: Network Interfaces



# Device Driver: Accessing Controller

## How do you start an I/O operation?

## 1. Programmed I/O (PIO)

- Initiate and watch for every byte of I/O operation

## 2. Direct Memory Access (DMA)

- Offload work to special-purpose processor responsible for large xfers
- CPU initiates DMA transfer
- Write DMA command block into main memory

  Pointer to source and destination addresses

  Size of transfer
- Inform DMA controller of address of command block
- DMA controller handles transfer with I/O device controller
- DMA controller interrupts CPU when entire transfer complete
- DMA can use physical or virtual addresses (DVMA)



# Interacting with Device Controllers

## How to know when event is complete?

## 1. Polling

- **Handshake** by setting and clearing flags
- Disadvantage: Busy-waiting

  Waste cycles when I/O device is slow

  Must be attentive when device becomes ready (lose data)

## 2. Interrupts

- Goal: Handle asynchronous events
- Assert **interrupt request line** when device is ready
- CPU jumps to appropriate **interrupt service routine** (ISR)

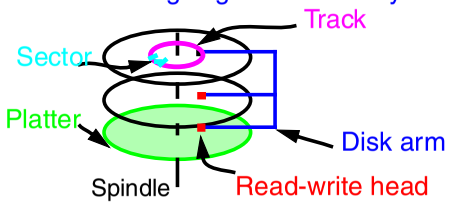
  Interrupt vector: Table of ISR addresses

  Index by interrupt number
- Lower priority interrupts postponed until higher priority finished
- Disadvantage: Costly to be interrupted for every event (use DMA!)



# Disk Characteristics

## Important to understand for designing efficient file systems




Sector: Minimum unit of transfer (512 bytes or 1K)

Two disk arms per platter (except skip outside surfaces)

Cylinder: Corresponding track on each platter (surface)

## To read or write disk block

- **Seek:** Position heads over destination cylinder (worst-case: 5 ms)
- **Rotational delay:** Wait for sector to rotate underneath head (8 ms)
- **Transfer:** Transfer bytes at bandwidth (5 MB/s)

## Average performance?

# Disk Sectors

How to pick the size of disk sector?

Goal: Increase density of sectors per track

- Increase capacity of disk
- Increase bandwidth (given fixed RPM)

Limitations

- Need space between sectors
  - ECC bits
  - Prevent overwriting next sector
- Internal fragmentation
  - Entire sector may not be filled with useful information

Tracks at outside of disk are longer

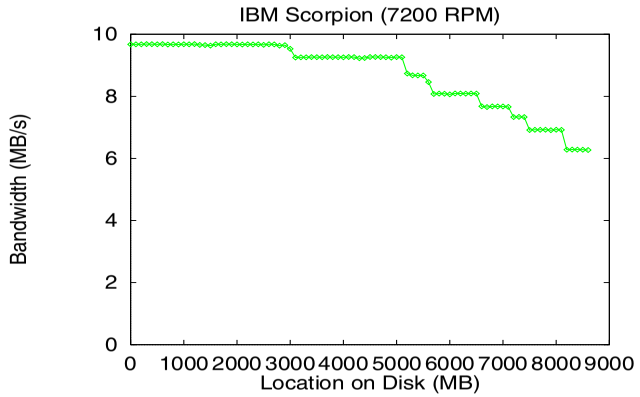
- Constant number of sectors per track
  - More space between sectors on outside track
- Constant bit density (multi-zone disks)



# Multi-Zone Disks

Outside tracks have more sectors

- Deliver more bandwidth at outside tracks




Implication: Empty disk delivers better bandwidth



# Disk Technology Trends

Data --> More dense

- More bits per square inch
- Disk head closer to surface
- Create smaller disks with same capacity

Disk diameter --> smaller

- Spin faster --> Increase bandwidth and shorter rotational delay
- Less distance for head to travel across --> faster seeks
- Lighter weight --> Good for laptops

Disk price --> Cheaper

- Decrease by factor of 2 per year (since 1991)

Density improving more than speed (mechanical limitations)