Operating Systems!

Input/Output (I/O):
I/O & Disk Scheduling

Prof. Travis Peters

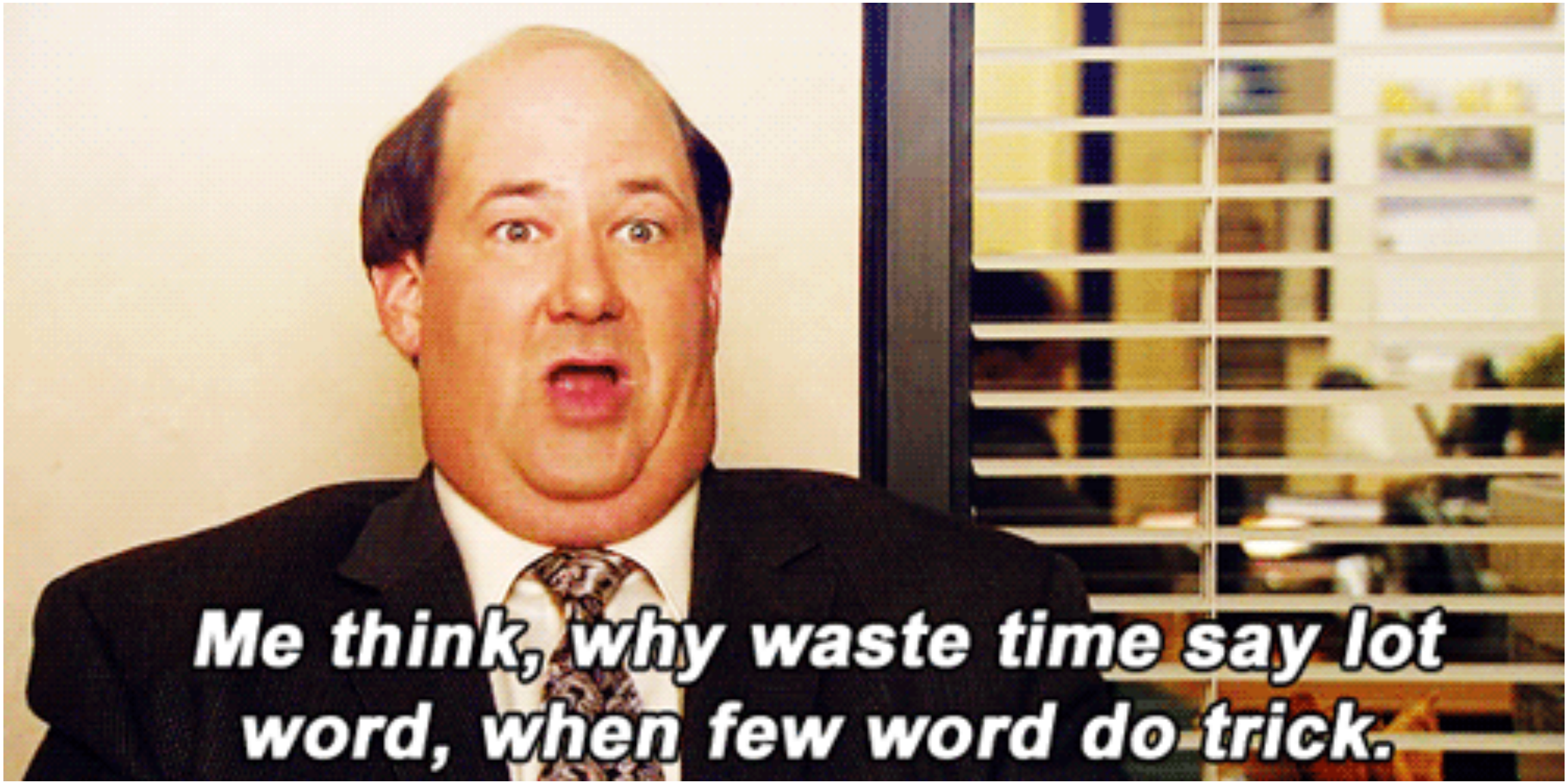
Montana State University

CS 460 - Operating Systems

Fall 2020

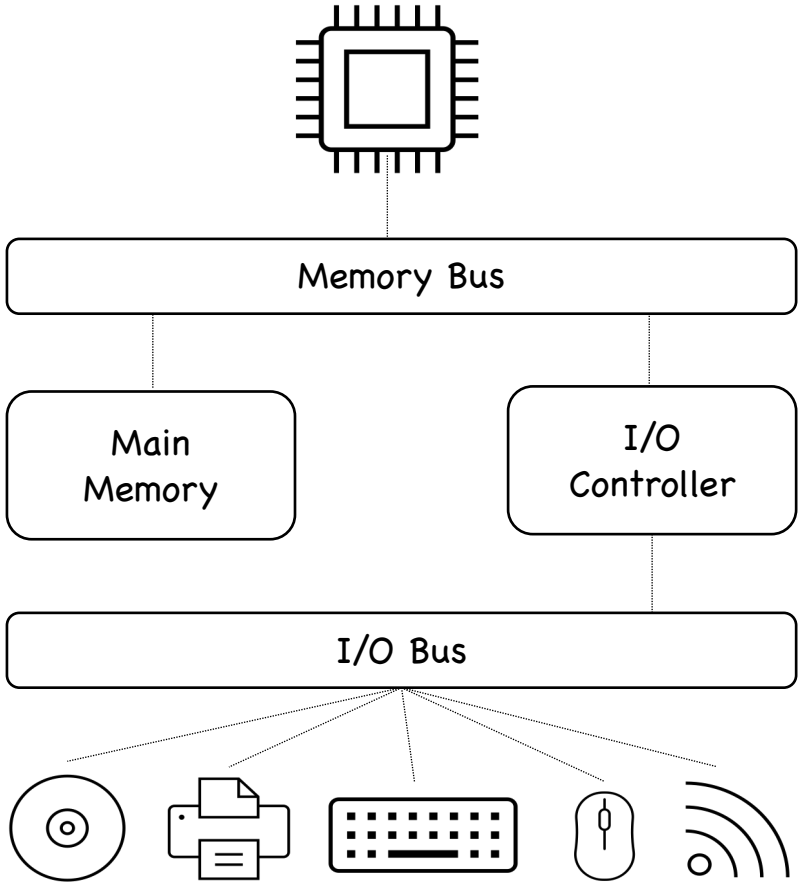
https://www.cs.montana.edu/cs460

Some diagrams and notes used in this slide deck have been adapted from Sean Smith's OS courses @ Dartmouth. Thanks, Sean!


Me think, why waste time say lot
word, when few word do trick.

Today

- Agenda
  - Key concepts behind I/O
    - *What is I/O? I/O Categories. Memory access.*
    - *What is a file? What is a file system? etc.*
  - Secondary Storage
    - Emphasis on *disks* and *disk scheduling*
  - Files
    - Operations
    - Implementations
  - File System
    - Virtual File Systems
    - Implementations
  - Security



I/O: How?

Recall: Techniques for I/O & Memory Access

- Programmed I/O

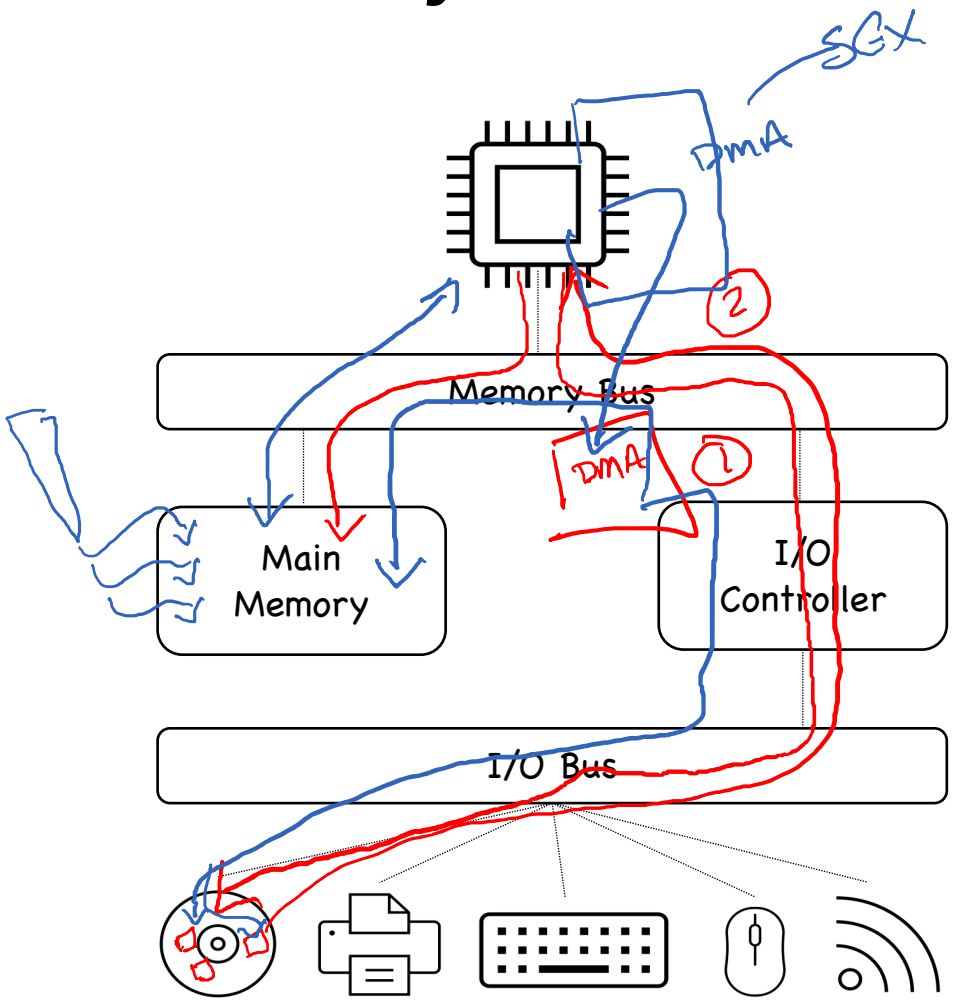
  (*active*; processor polls I/O device)

- Interrupt Driven I/O

  (*assist*; I/O device does work; interrupt processor for help)

- DMA

  (*delegate*; DMA module give OP, DEV, ADDR, #WORDS; send interrupt when done)



Direct Memory Access

Role of DMA: transfer data to/from memory over system bus

< op, dev, addr, n >

-> most efficient to have all I/O done via DMA on dedicated I/O bus

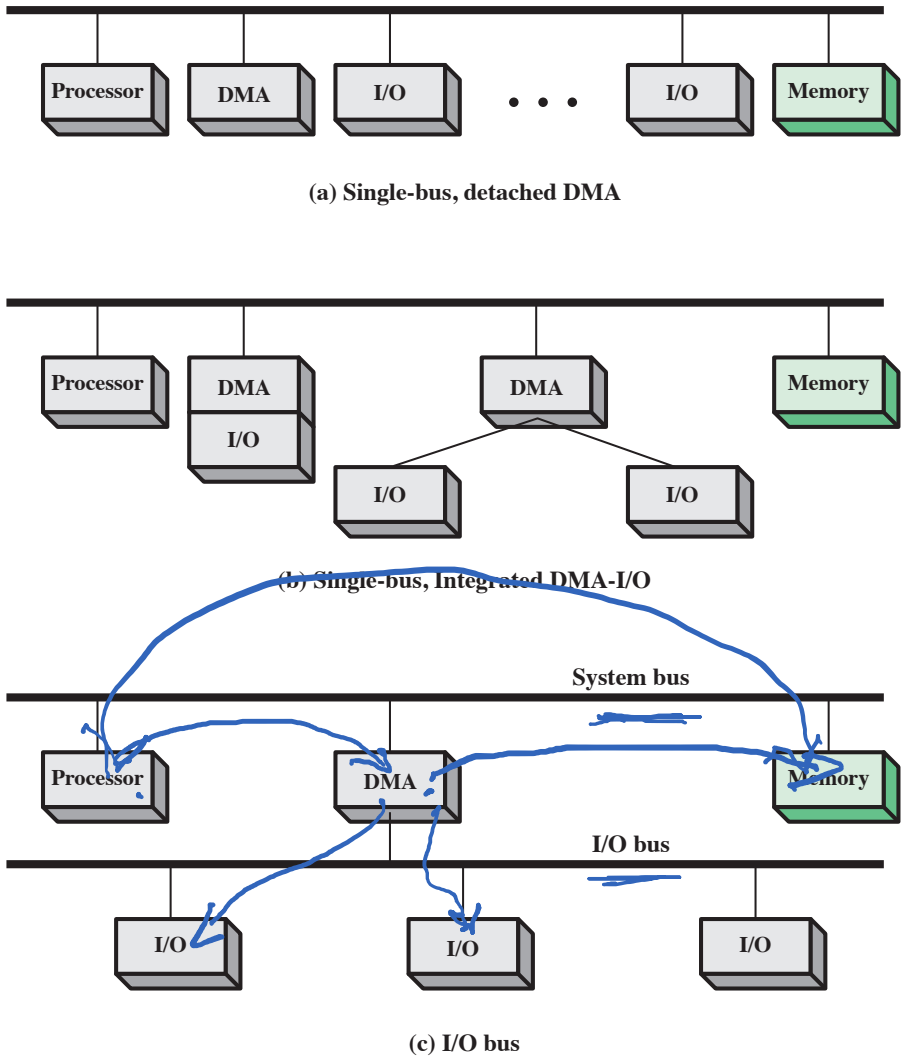
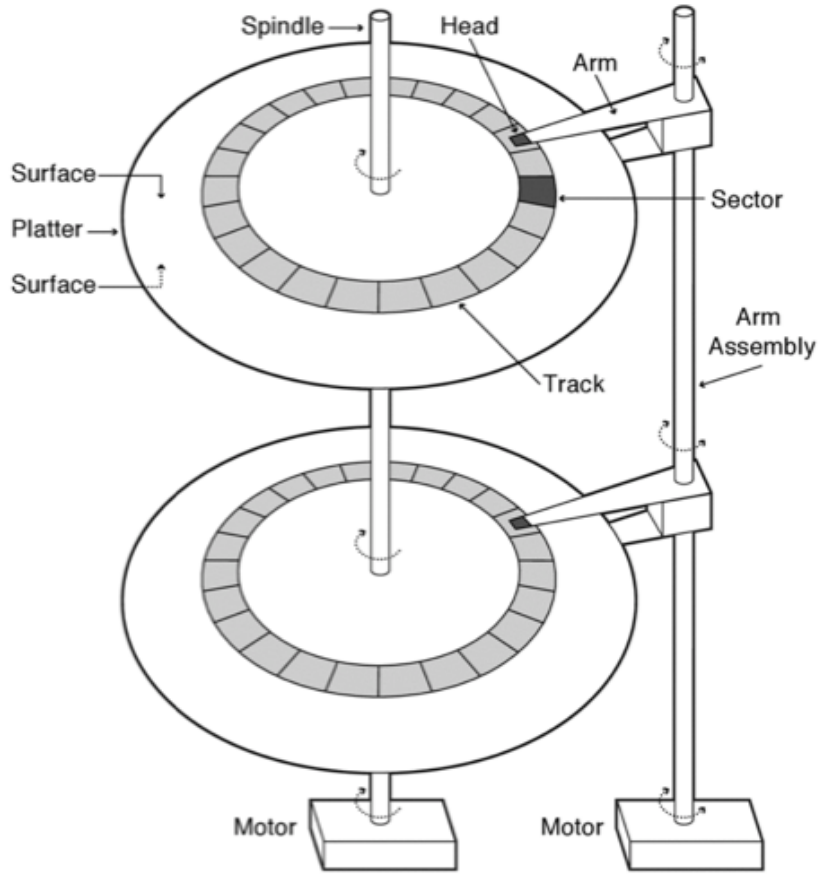


Figure 11.3 Alternative DMA Configurations

Disks & Disk Scheduling

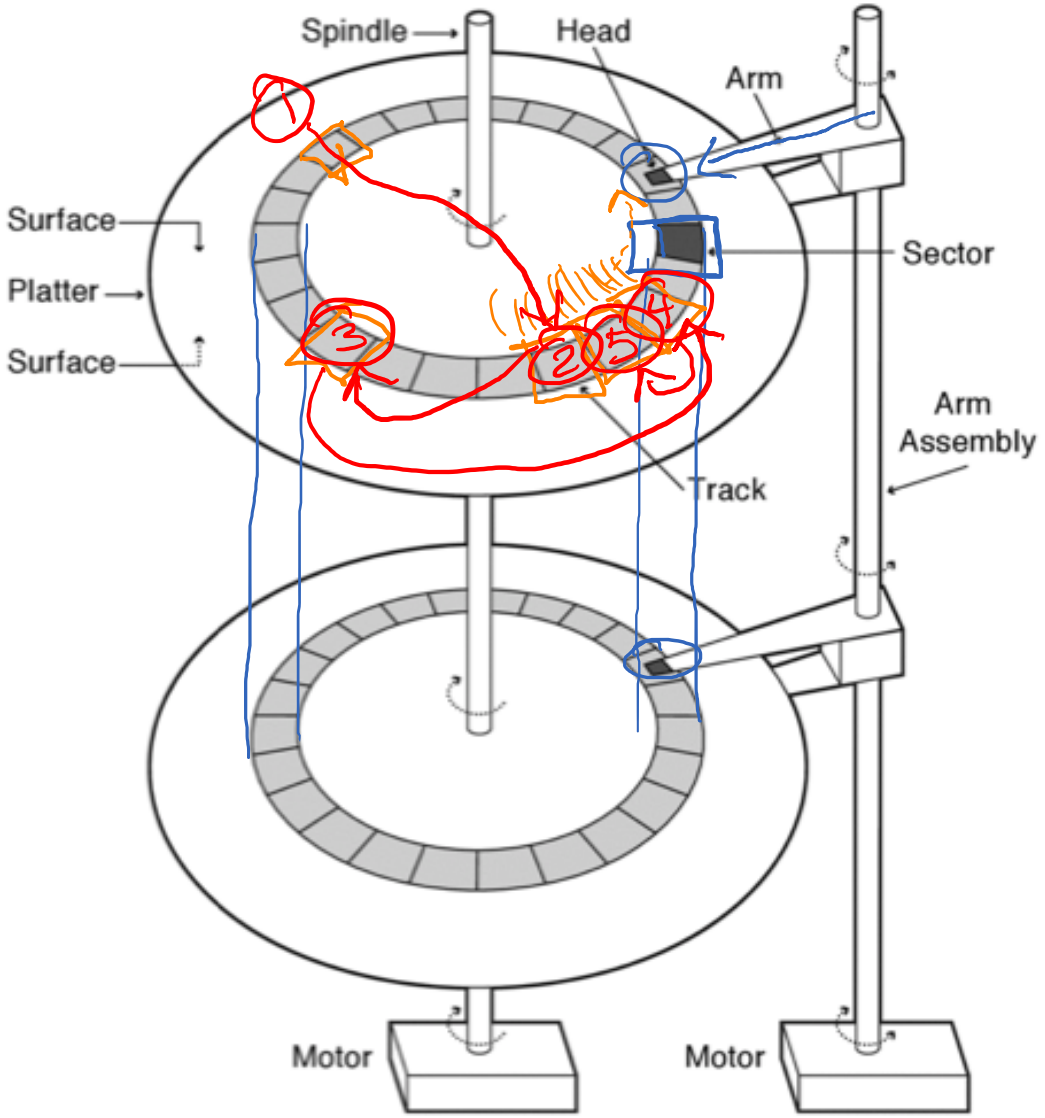
How can we use secondary storage (e.g., disks) efficiently?



Disks

Basics:

- rotating disk
- read/write head
- head must move (seek) to correct position (track / sector)



A = Track / B = Sector / C = Sector of Track / D = File

cyl.

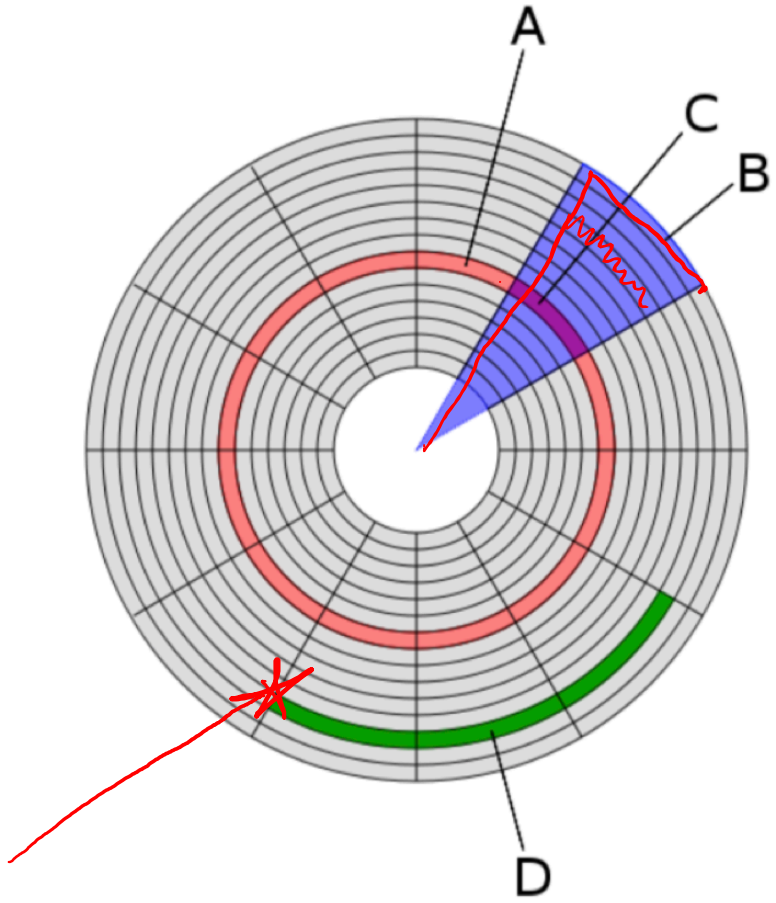
Disk Scheduling

Basics:

- rotating disk
- read/write head
- head must move (seek) to correct position (track / sector)

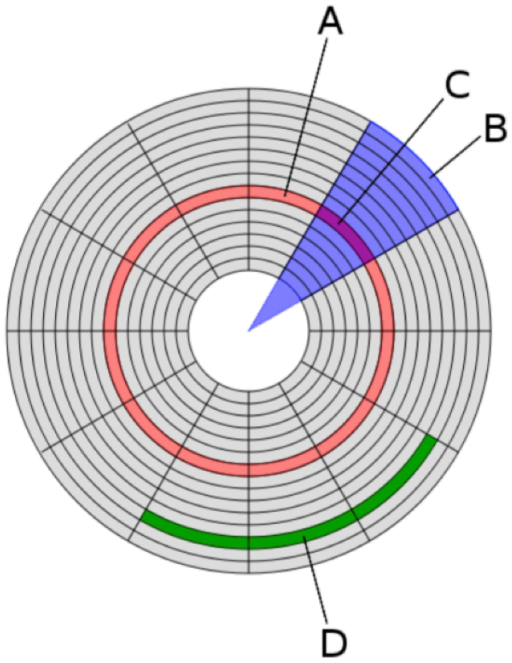
Disk Scheduling:

Given a series of access requests, on which track should the disk arm be placed next? (Criteria?!)



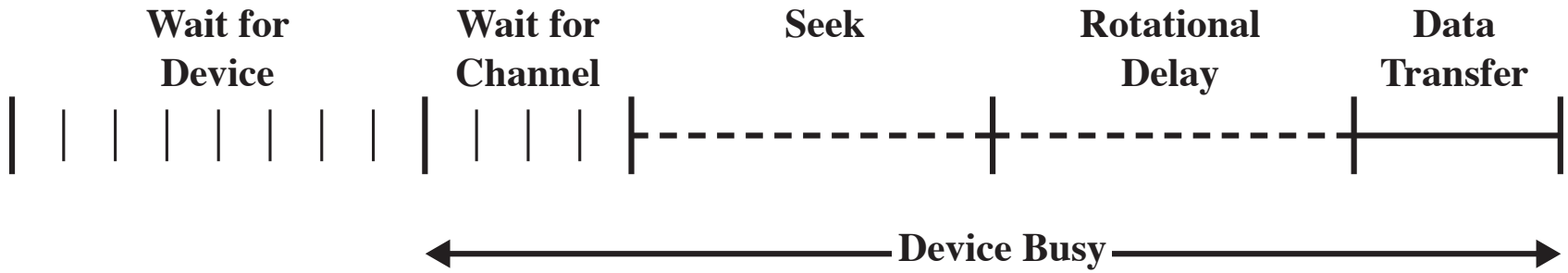
Disk Scheduling *(cont.)*

A = Track / B = Sector / C = Sector of Track / D = File

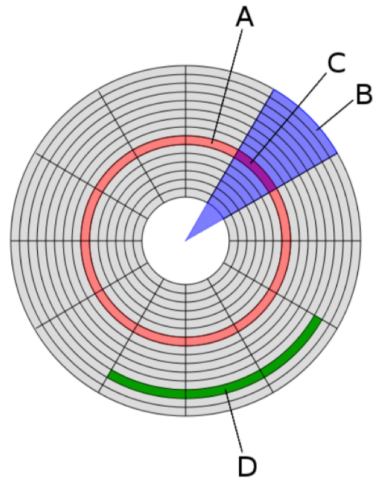


Basics:

- rotating disk
- read/write head
- head must move (seek) to correct position (track / sector)
  - wait…
  - seek time
  - rotational latency / rotational delay
  - access time
  - transfer time



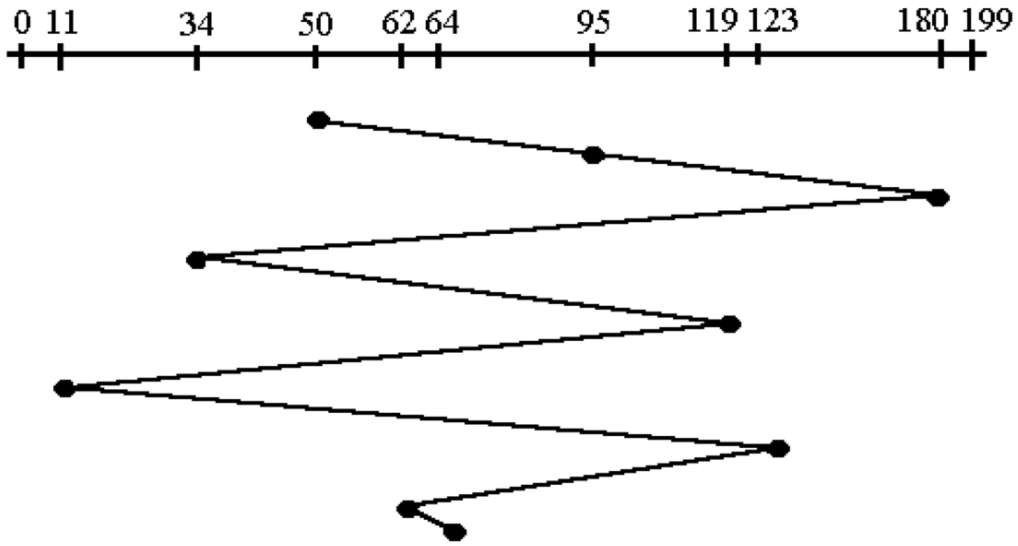
Disk Scheduling Policies



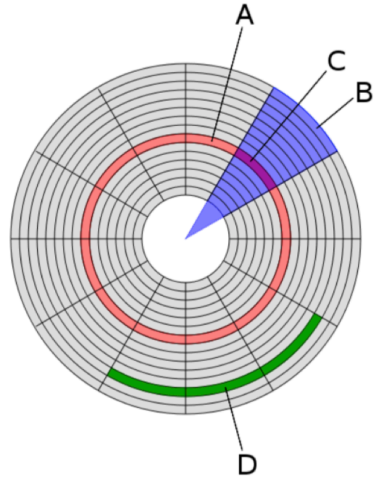
Consider multiple I/O requests queued up to access an I/O device (e.g., a disk)…

Q: How to schedule I/O operations?!

- Random?
- FIFO/FCFS?
- Priority-based?
- LIFO?
- *…can we do better…?*



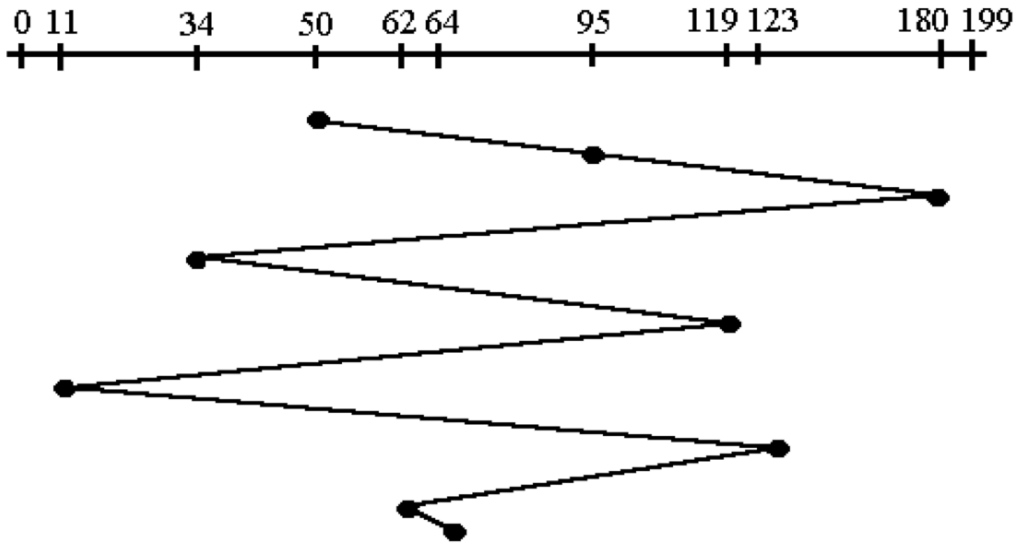
Disk Scheduling Policies *(cont.)*



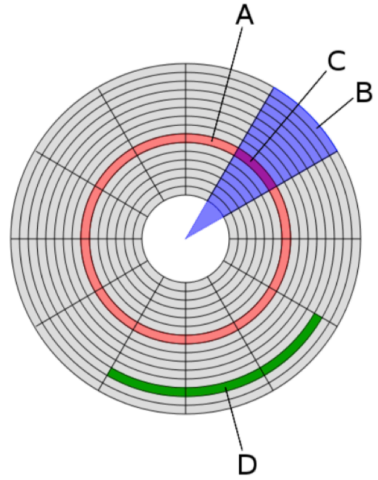
Shortest-Service-Time-First (SSTF)

- select I/O op with least amount of movement from disk arm's current position
- choose next op that incurs min. seek time

- **Pros?**
- **Cons?**



Disk Scheduling Policies *(cont.)*

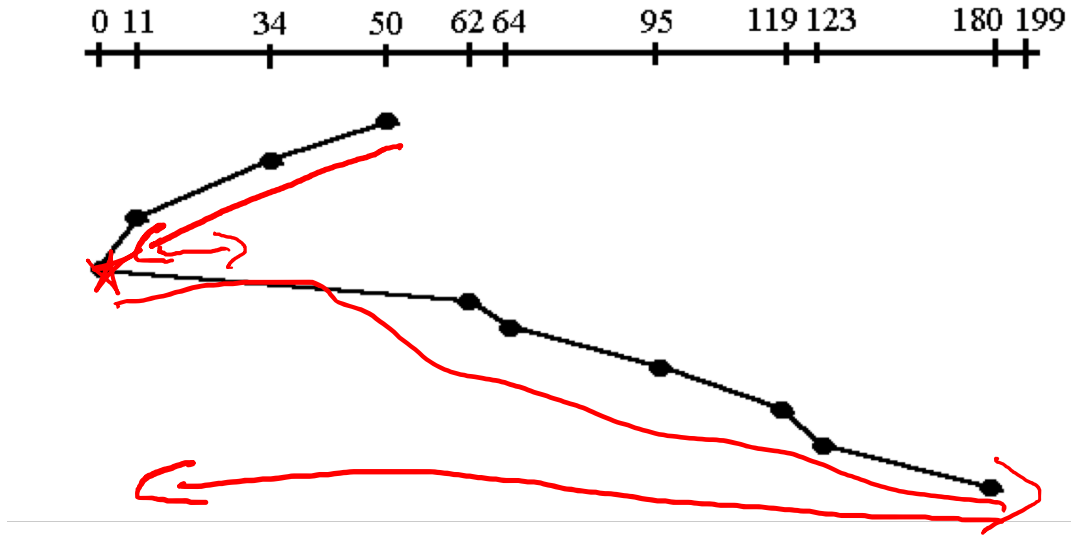


SCAN ("Elevator Algorithm")

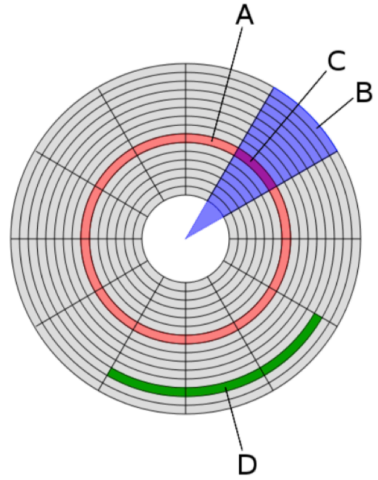
- disk arm moves in one direction, servicing closest request in the direction of travel
- continue until…
  - last track is reached, or
  - there are no more requests to service (in that direction)

- **Pros?**

- **Cons?**



Disk Scheduling Policies *(cont.)*

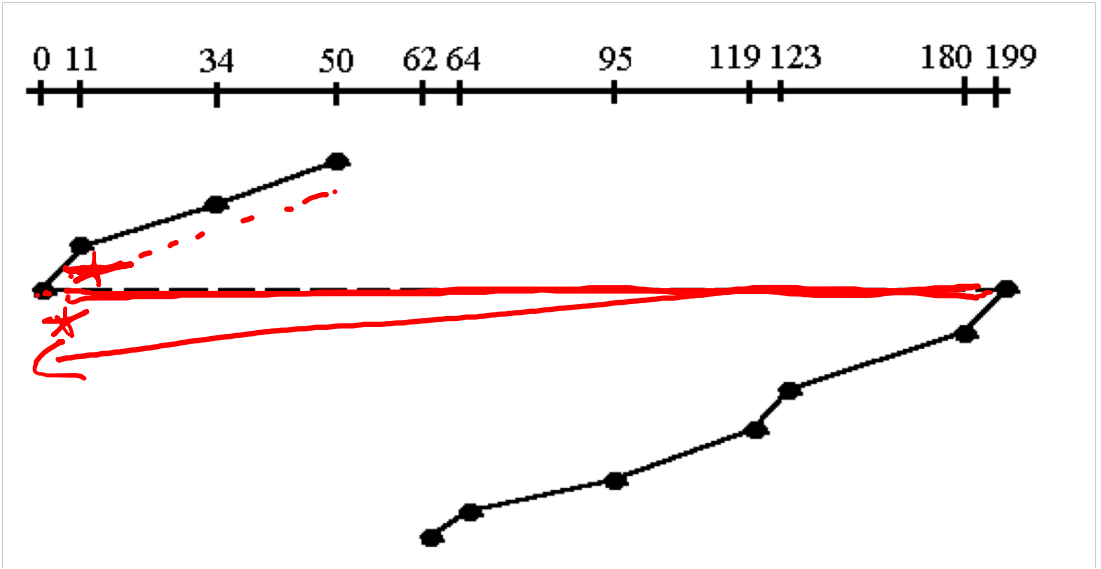


C-SCAN (Circular Scan)

- similar to SCAN… but never change directions
- when the last track is reached, reset the head ("wrap around")

- **Pros?**

- **Cons?**



+ Many more… N-Step-SCAN, FSCAN, C-LOOK, etc…

Extras

Topics to explore when going beyond the basics...

Flash Memory

- **EEPROM** (Electronically Erasable Programmable Read Only Memory)
  - **Ex:** NAND Flash
  - READ performance
    - Random READ: 25 s, Sequential READ: 25ns
  - WRITE performance
    - PROGRAM PAGE: 25 s, BLOCK ERASE: 1.5ms
  - Endurance: 100,000 PROGRAM/ERASE cycles

In The News
Tesla's cars use **Linux** and do **a huge amount of logging**. According to 057 Technology's Jason Hughes, "The information logged here is pretty much useless on production vehicles. Unless a developer has a specific reason for enabling it, it does the customer no good. These logs are also rarely downloaded by Tesla." That **excessive logging is causing the flash memory in the Media Control Unit (MCU) v1 to fail**, which causes the car to **lose touch screen functionality, which in a Tesla controls most everything**. V1 units are in model S and X Teslas made before 2018.
— https://www.tomshardware.com/news/flash-memory-wear-killing-older-teslas-due-to-excessive-data-logging-report

GANGRENE: Exploring the Mortality of Flash Memory
https://www.usenix.org/system/files/conference/hotsec12/hotsec12-final4.pdf