CSE 451: Operating Systems
Winter 2022

Module 15
I/O System and Secondary Storage

Gary Kimura

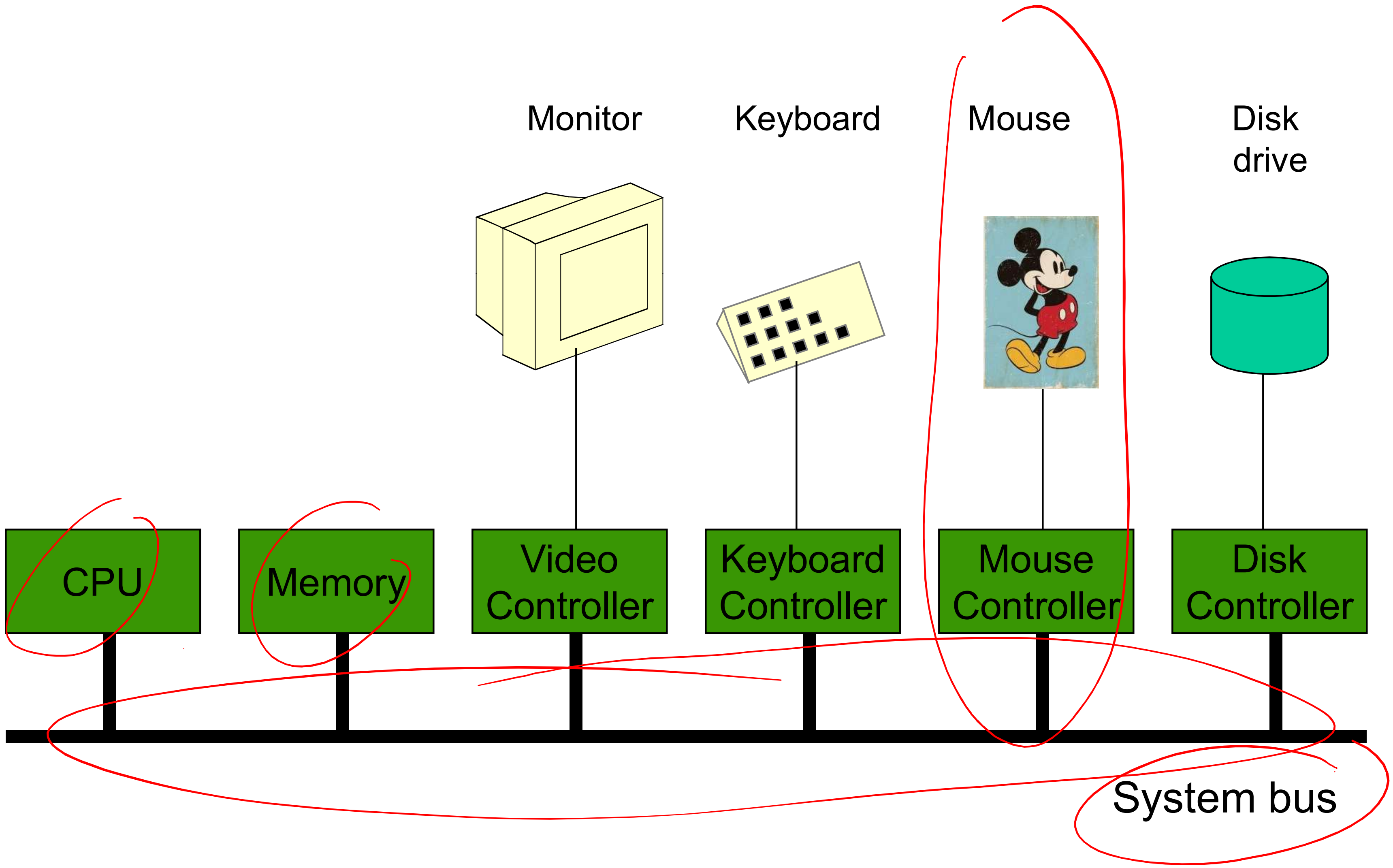
What's Ahead

- High level view of the I/O System
- We will concentrate on two major I/O components
- Secondary Storage
  - Disk Drives and SSD
- File System
  - Objectives /mission
  - Programming Interface
  - On-Disk Structure (Storage layout)

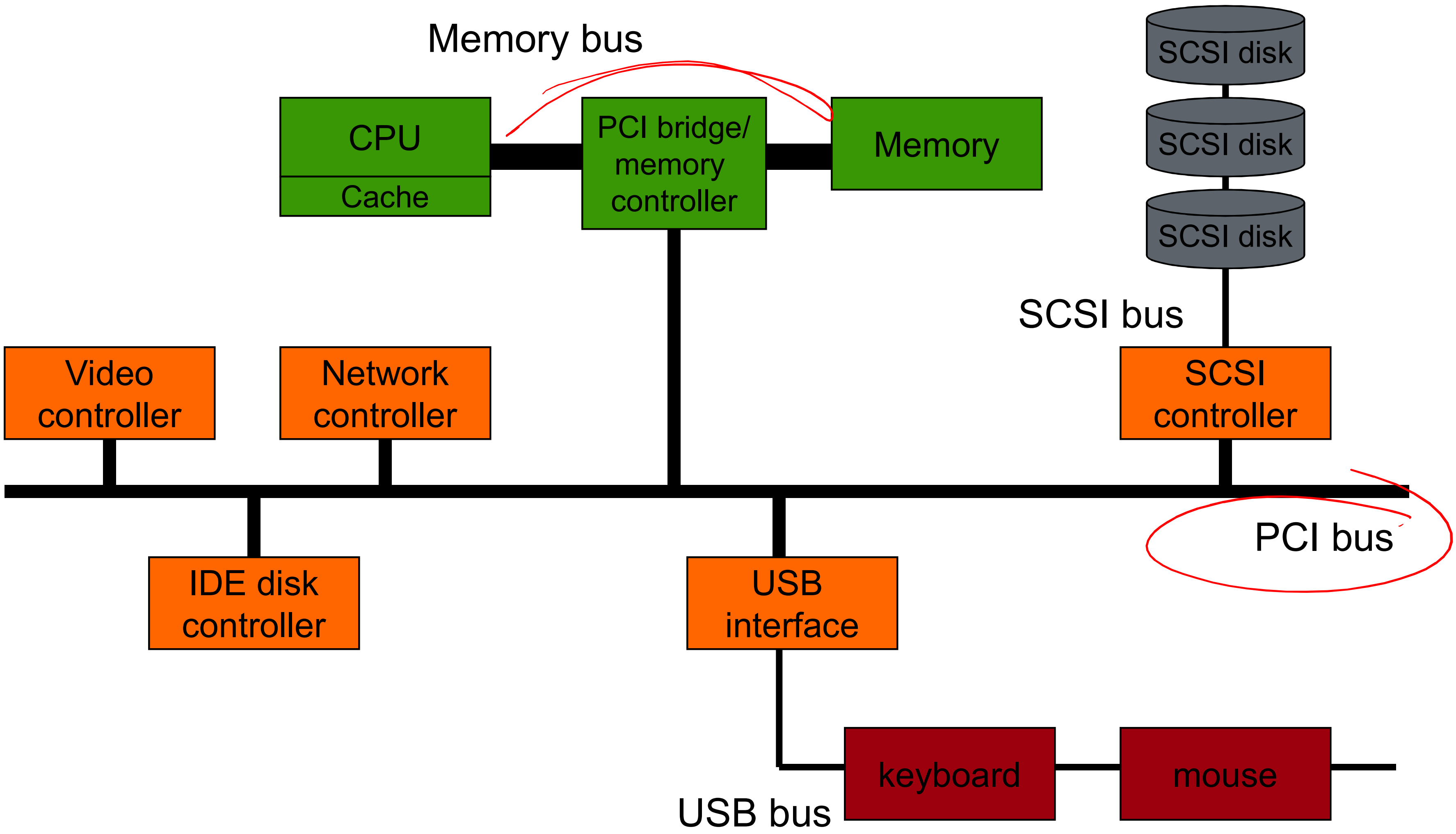
But first the I/O System Hardware Environment

- Major components of a computer system:
  CPU, memories (primary/secondary), I/O system
- I/O devices:
  - Block devices – store information in fixed-sized blocks; typical sizes: 128-4096 bytes
  - Character devices – delivers/accepts stream of characters (bytes)
- Device controllers:
  - Connects physical device to system bus (Minicomputers, PCs)
  - Mainframes use a more complex model:
    Multiple buses and specialized I/O computers (I/O channels)
- Communication:
  - Memory-mapped I/O, controller registers
  - Direct Memory Access - DMA

I/O Hardware - Single Bus



I/O Hardware - Multiple Buses



Diversity among I/O Devices

The I/O System has to consider device characteristics:

- Data rate:
  - may vary by several orders of magnitude
- Complexity of control:
  - exclusive vs. shared devices
- Unit of transfer:
  - stream of bytes vs. block-I/O
- Data representations:
  - character encoding, error codes, parity conventions
- Error conditions:
  - consequences, range of responses
- Applications:
  - impact on resource scheduling, buffering schemes

input
output
both

The I/O System's role

The I/O System is essentially the glue that holds this all together. By providing

1. A programming interface for applications to use (hopefully not too big, also not too cumbersome). Some functions are straightforward, like open, close, read, and write. And some are device specific like ioctl's.

2. A way for the myriad of hardware devices to attach to the system through device drivers.

In the Windows, its I/O System, File Systems, and Device Drivers are the largest pieces of code in the OS.

Organization of the I/O Function

- Programmed I/O with polling:
  - The processor issues an I/O command on behalf of a process
  - The process busy waits for completion of the operation before proceeding
- Interrupt-driven I/O:
  - The processor issues an I/O command and continues to execute
  - The I/O module interrupts the processor when it has finished I/O
  - The initiator process may be suspended pending the interrupt
- Direct memory access (DMA):
  - A DMA module controls exchange of data between I/O module and main memory
  - The processor requests transfer of a block of data from DMA and is interrupted only after the entire block has been transferred

I/O System and Secondary Storage

- That was our quick overview of the I/O System.

  There is a lot more to discuss if time permits. Such as following a full I/O request from the time the User issues the request, until it causes some action at the hardware level, and then the return trip back to the User.

- But now onto Secondary Storage

Secondary storage

- Secondary storage typically:
  - is anything that is outside of "primary memory"
  - does not permit direct execution of instructions or data retrieval via machine load/store instructions
- Characteristics:
  - it's large: 500-2000GB
  - it's cheap: $0.05-$0.10/GB for hard drives from Dell (at 2TB size)
  - it's persistent: data survives power loss
  - it's slow: milliseconds to access
    - why is this slow??
  - it *does* fail, if rarely
    - big failures (drive dies; MTBF ~3 years)
      - if you have 100K drives and MTBF is 3 years, that's 1 "big failure" every 15 minutes!
    - little failures (read/write errors, one byte in 10^{13})

Another trip down memory lane …

IBM 2314

About the size of 6 refrigerators

8 x 29MB (M!)

Required similar-sized air condx!

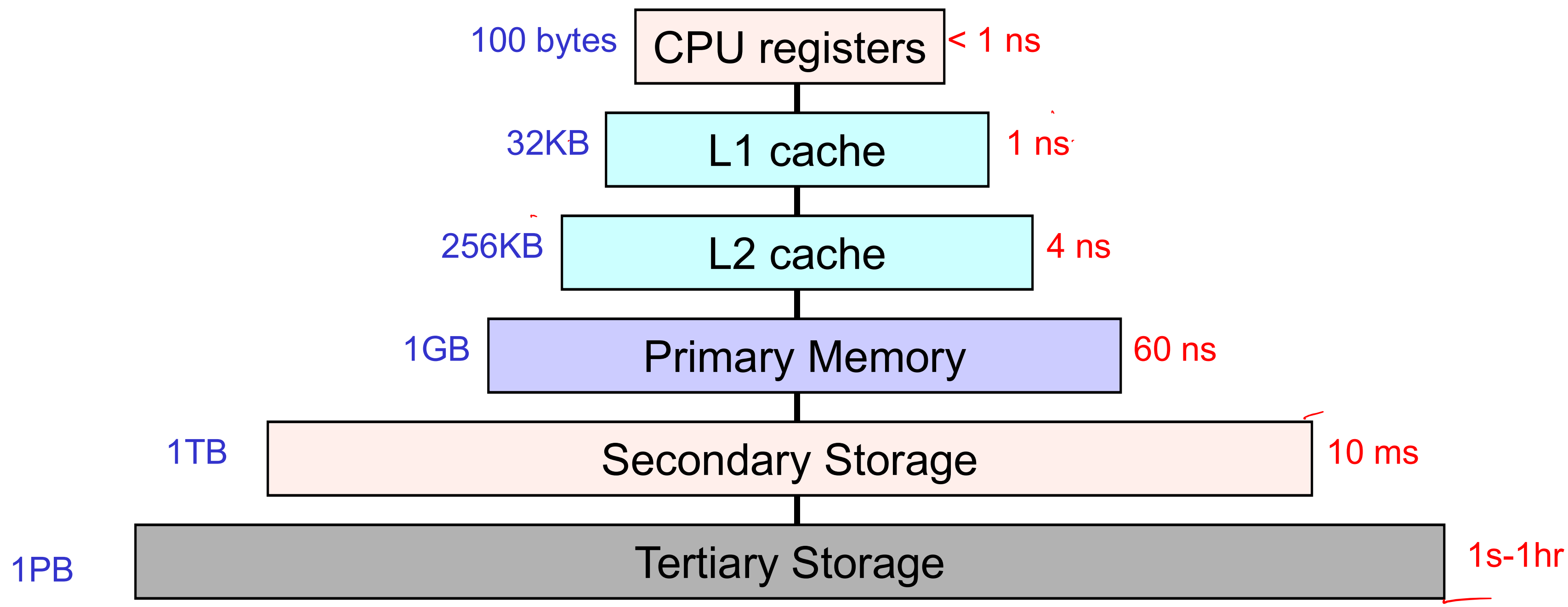
.01% (not 1% – .01%!) the capacity of this $100 4"x6"x1" item

Disk trends

- Disk capacity, 1975-1989
  - doubled every 3+ years
  - 25% improvement each year
  - factor of 10 every decade
  - Still exponential, but far less rapid than processor performance
- Disk capacity, 1990-recently
  - doubling every 12 months
  - 100% improvement each year
  - factor of 1000 every decade
  - Capacity growth 10x as fast as processor performance!

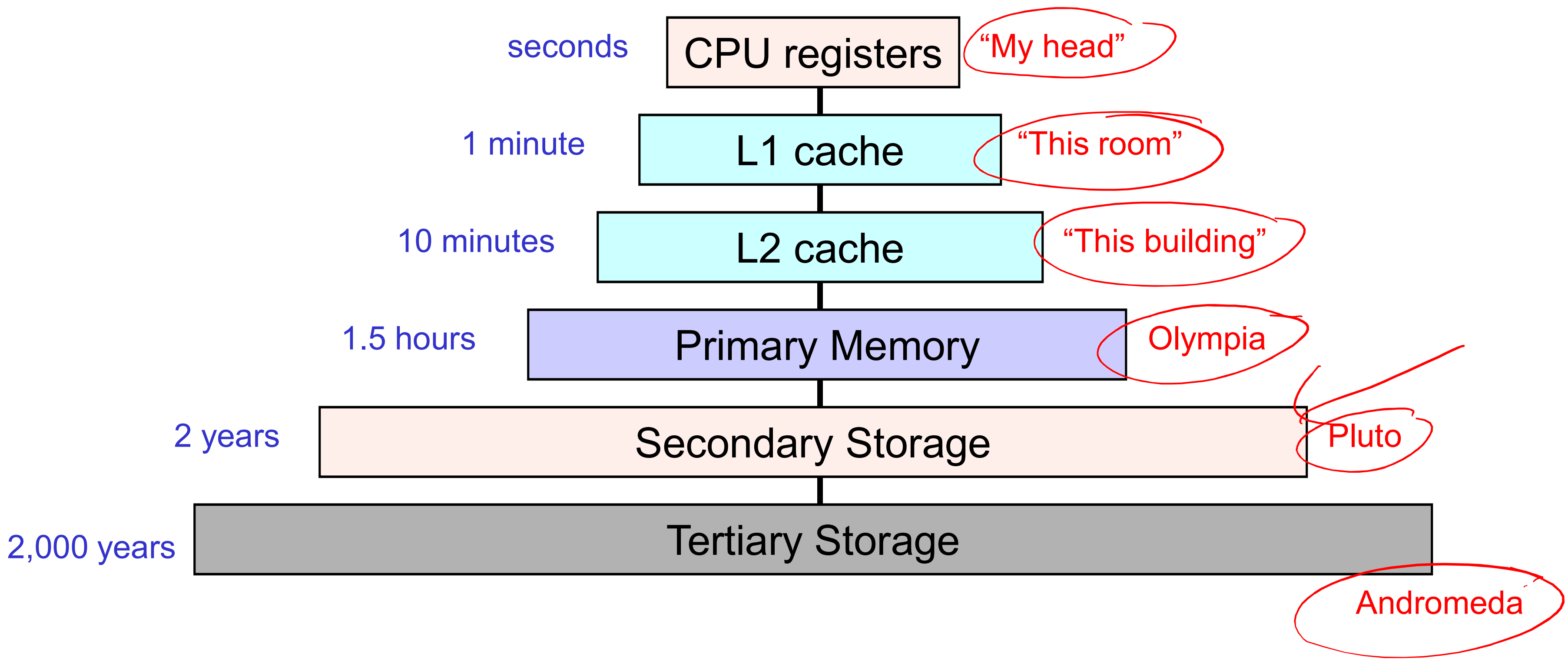
- Only a few years ago, we purchased disks by the megabyte (and it hurt!)
- Today, 1 GB (a billion bytes) costs ~~$1~~ ~~$0.50~~ $0.10-$0.05 from Dell (except you have to buy in ~~increments of~~ ~~40~~ ~~80~~ 250 2000 GB)
  - => 1 TB costs ~~$1K~~ ~~$500~~ $100, 1 PB costs ~~$1M~~ ~~$500K~~ $100K
- Technology is amazing
  - Flying a 747 6” above the ground
  - Reading/writing a strip of postage stamps
- But …
  - Jets do crash …

Memory hierarchy

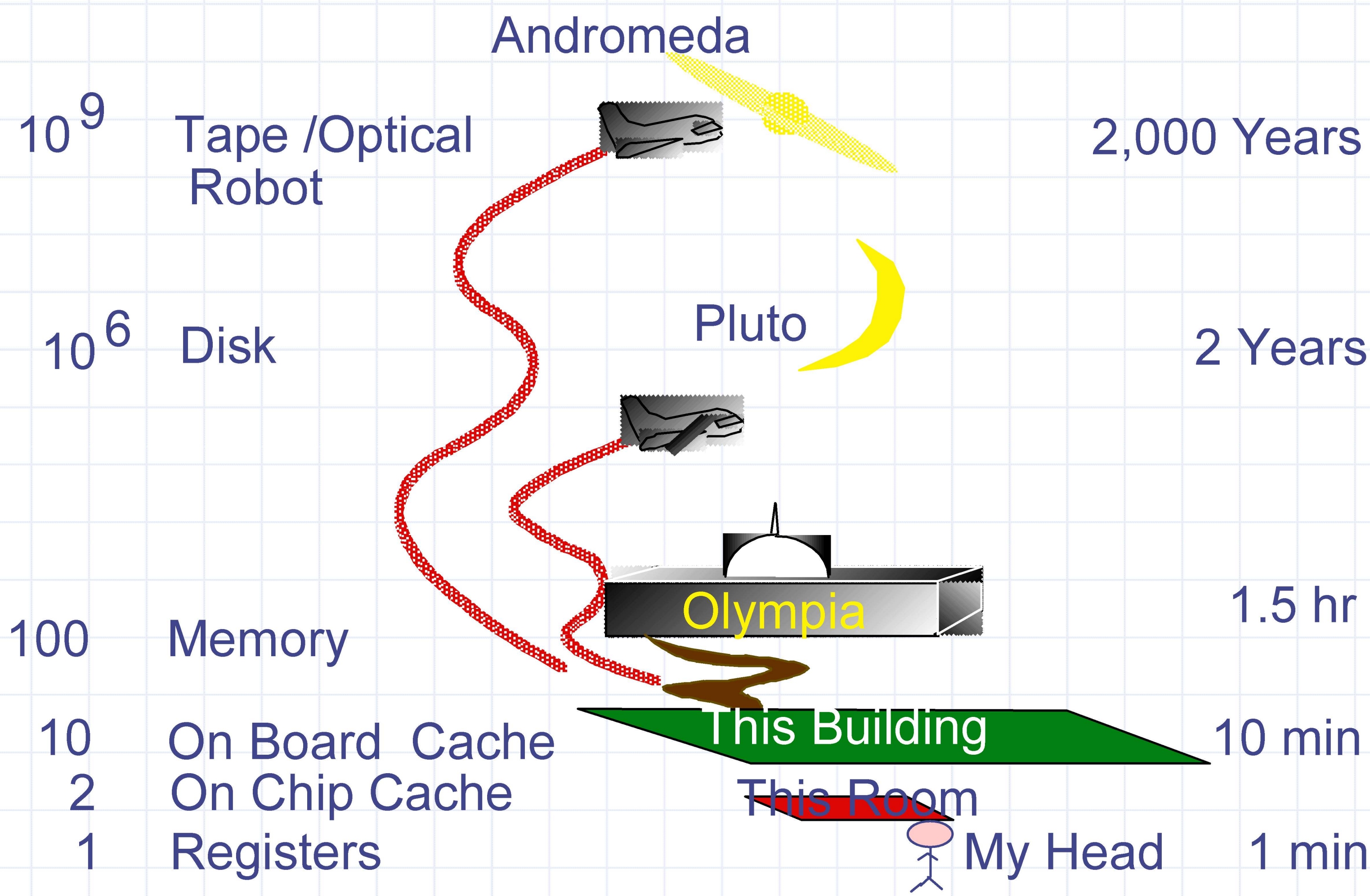


- Each level acts as a cache of lower levels

Memory hierarchy: distance analogy



Storage Latency: How Far Away is the Data?

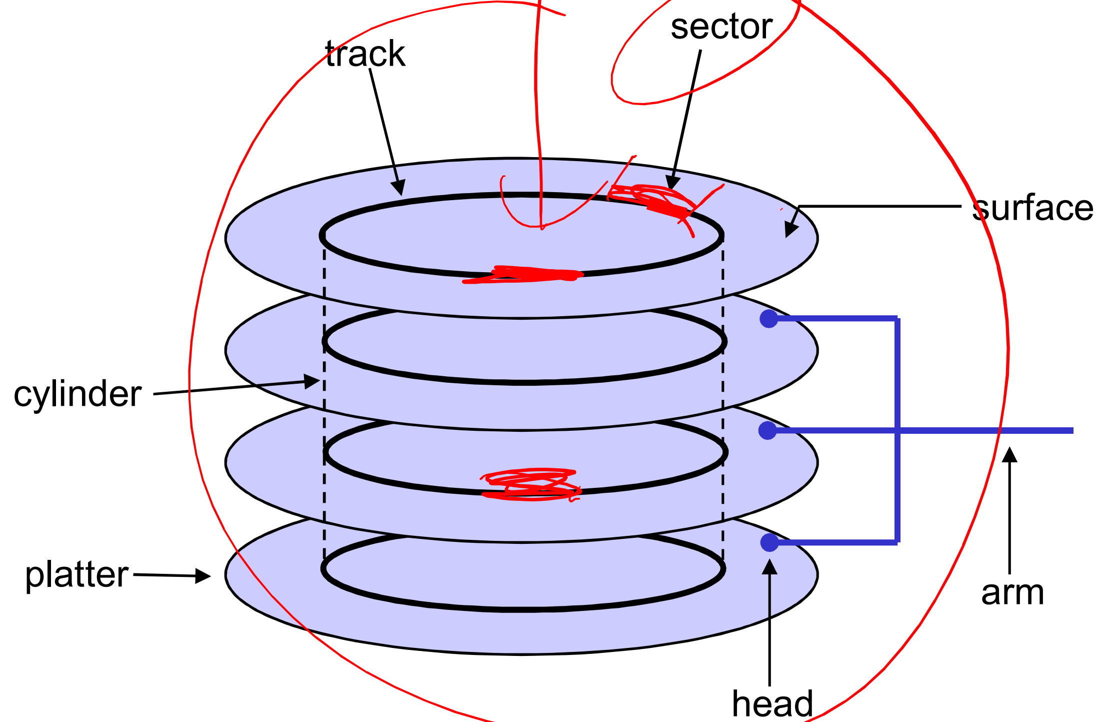


Disks and the OS

- Disks are messy, messy devices
  - errors, bad blocks, missed seeks, etc.
- Job of OS is to hide this mess from higher-level software (disk hardware increasingly helps with this)
  - low-level device drivers (initiate a disk read, etc.)
  - higher-level abstractions (files, databases, etc.)
  - (note that modern disk drives do some of this masking for the OS)
- OS may provide different levels of disk access to different clients
  - physical disk block (surface, cylinder, sector)
  - disk logical block (disk block #)
  - file logical (filename, block or record or byte #)

Physical disk structure

- Disk components
  - platters
  - surfaces
  - tracks
  - sectors
  - cylinders
  - arm
  - heads



Interacting with disks

- In the old days…
  - OS would have to specify cylinder #, sector #, surface #, transfer size
    - i.e., OS needs to know all of the disk parameters
- Modern disks are even more complicated
  - not all sectors are the same size, sectors are remapped, …
  - disk provides a higher-level interface, e.g., SCSI
    - exports data as a logical array of blocks [0 … N]
    - maps logical blocks to cylinder/surface/sector
    - OS only needs to name logical block #, disk maps this to cylinder/surface/sector
    - on-board cache
    - as a result, physical parameters are hidden from OS
      - both good and bad

Disks and the File System

- There is a lot more to say about disks and their peculiar characteristics, but some of it is antiquated because of Solid State Drives (SSD). Hopefully, we will find time to talk about this later.

- From a simple File System perspective. Secondary storage (disk and SSD) are logical blocks that it can read and write.

- And the File System is our next topic.