Operating system

Part IX: IO Devices

--- Use HDD as instance to understand the mapping between file and blocks

--- Trend: Attempt to manage devices uniformly

Goals

- Know the basic concepts related to IO
  - Types of devices
  - General framework to connect devices with co mputers
  - How to control the devices
- Know the techniques related to Hard Disk
  - So as to provide the basis for file system

File concept
[We all know “file”, but what is it essentially?]

- Anything you are interested in and want to **st ore permanently** could be a “file”!
  - Long-term Information Storage
    - Must store large amounts of data
    - Information stored must survive the termination of th e process using it
    - Multiple processes must be able to access the information concurrently
- Each file is a named collection of data stored in a permanent device

We need store file permanently in secondary storage media, like Hard Disk

Processes

File System Calls

User's (process's) view of files

File 1

File 2

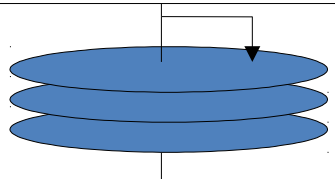
?

Mapping 2 – from file to Hdisk

Disk Driver

Disk Controller

Disk



I reorganize the IO + File system

- Because **Mapping 2** and **Mapping 1** (with HDisk as instance) could l[illegible] owing similar ide[illegible]

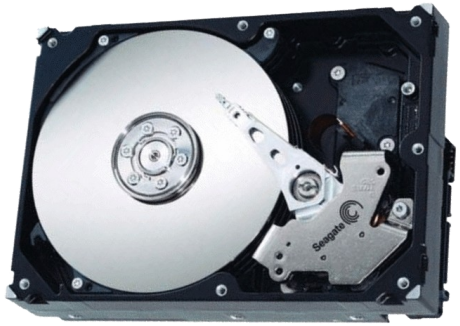
- They both
  - **From logic program/file to linear addressed space**
    - **Allocation** and **Address Translation**
    - **Indexing data structure (Tree)**
      - Some data structures are also needed and kept in MM to lo cate those programs

you'd better remember this and follow this to understand the [illegible] parts!

Basic IO

- General structure to connect devices –
  - Abstraction/Interface in OS to communicate with the diverse devices
    - IO devices - **Categories** of devices
    - **IO operations**
      - **Connect** all together & structure of IO module in OS
      - **Communication** types between CPU and IO devices
- Taking (Magnetic) Disk for instance
  - So-called "linear address sector space"
  - Organize sectors into partitions, and so-called "linear addressed block space"
  - Optical disk is similar

- We have many different device



Three Device Types

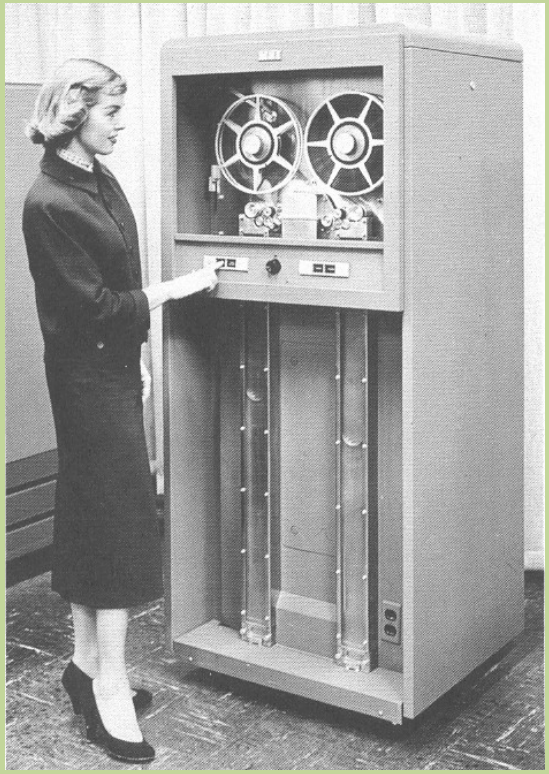
- Most operating system have three device types:
  1. **Character** devices [字符设备]
     - Character devices are read and written directly without buffering
       - Used for serial-line types of devices (e.g. USB port)
  2. **Block** devices: [块设备]
     - Block devices can only be written to and read from in multiples of the block size, typically 512 or 1024 bytes
       - Used for mass-storage (E.g. **hard disks**, tapes and CDROM)
  3. **Network** devices
     - Network devices are accessed via the socket interface
       - Used for network interfaces (E.g. Ethernet card)

PPTs from others\paul.rutgers.edu_cs519_S02\IO-File.ppt

Sequential Access Devices

Sequential Access = In order to access specific information, the device must sequentially pass through all preceding in formation

- 9 Track Tape (Reel to Reel [逐
- Cartridge Tapes



Direct Access Devices

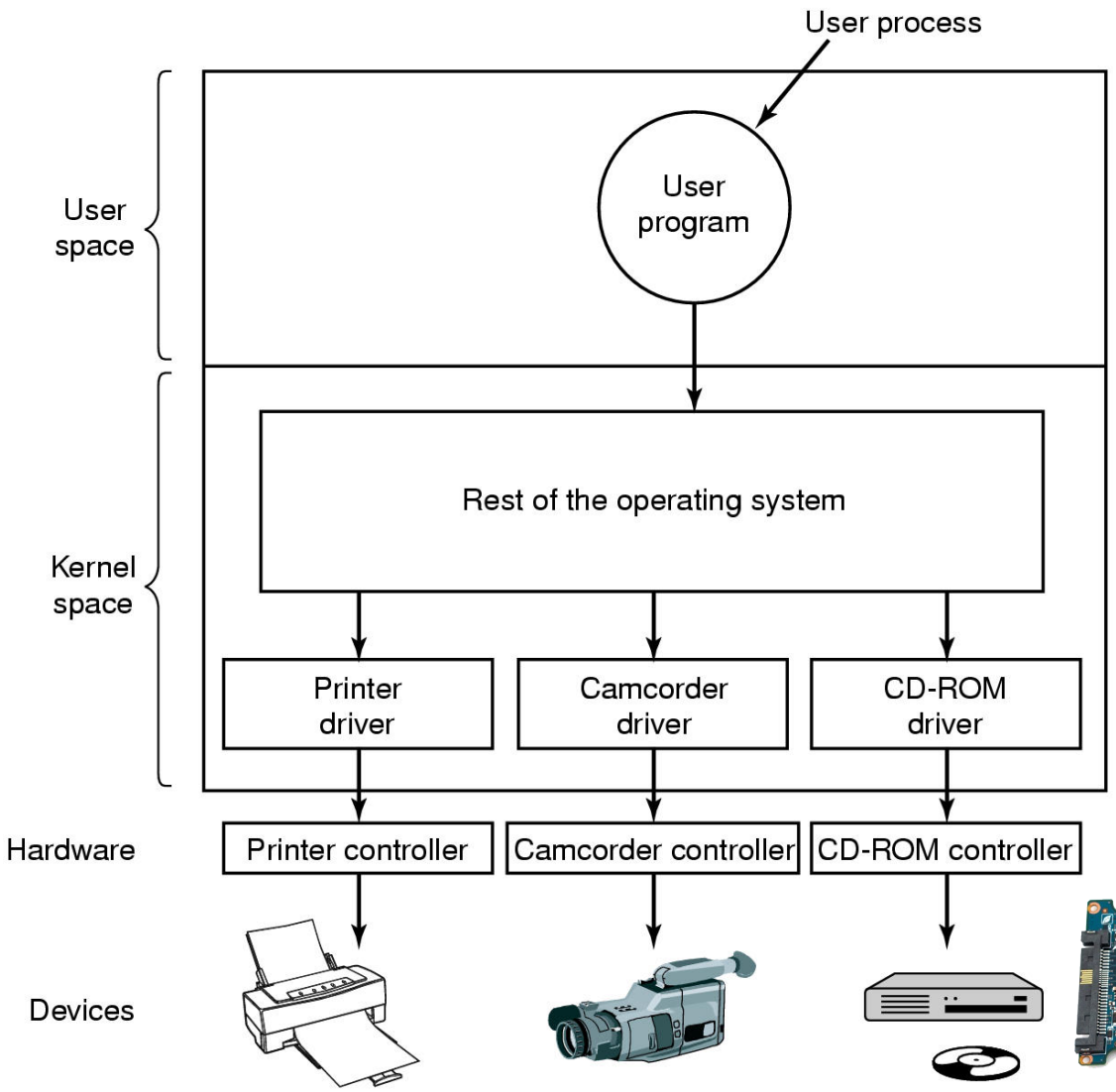
Direct Access = The specific information is accessed directly

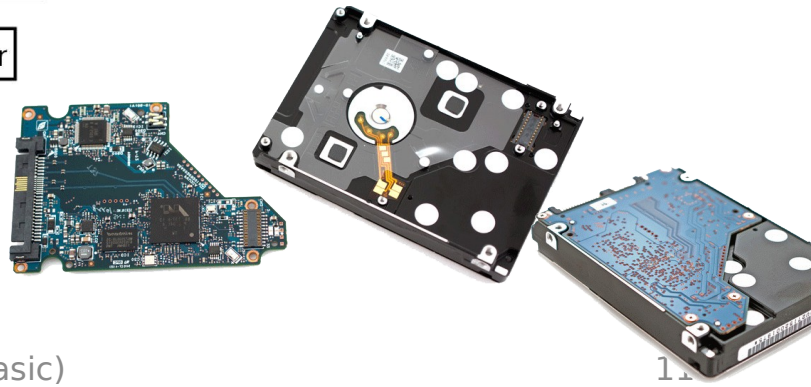
PPTs\Part XII\Part XII magnetic media].p

Examples

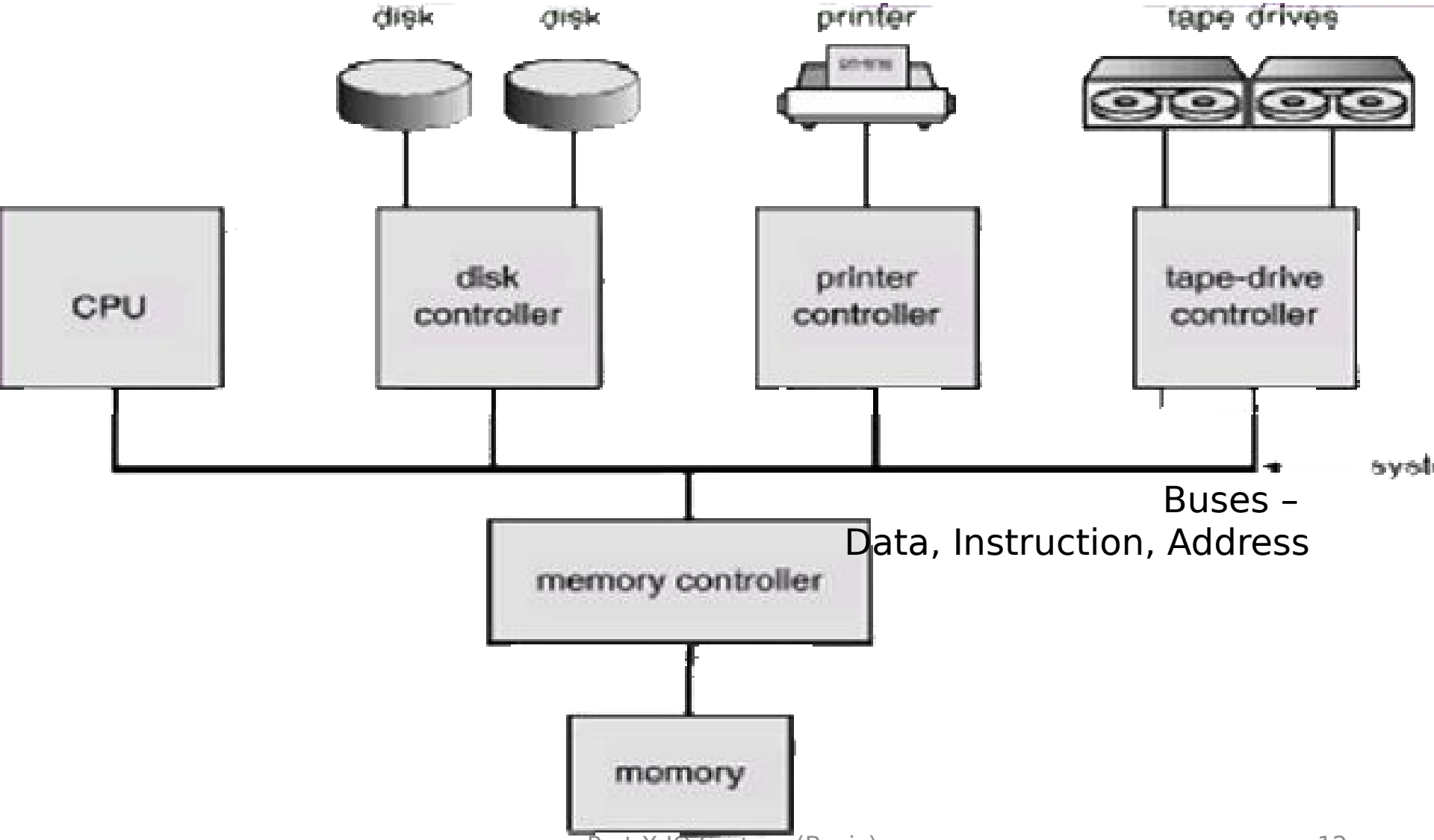
- floppy disk drives
- hard disk drives
- cartridge disk drives
- CD ROM and DVD drives

They share similar connection architecture





MM is also connected with controller



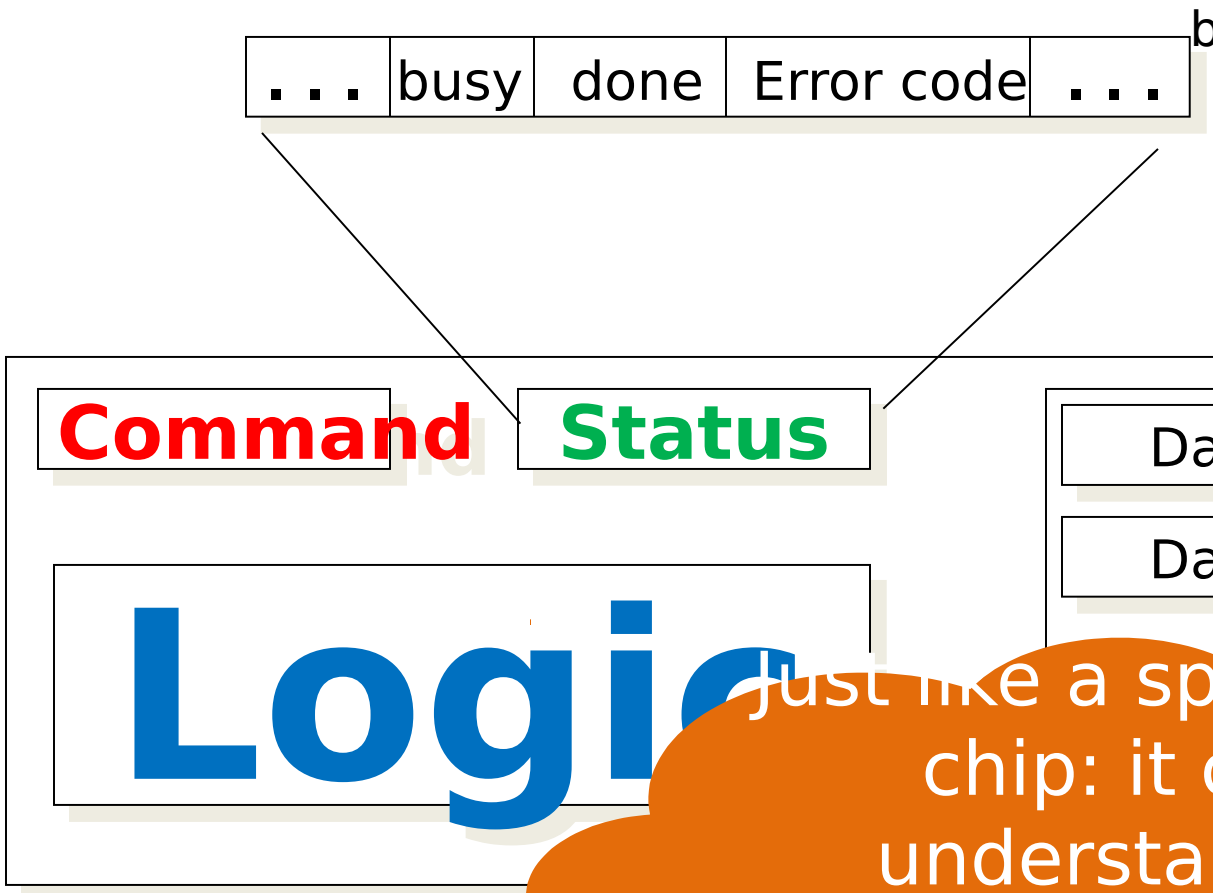
Buses –
Data, Instruction, Address

Device **controller** [控制器]

- **Device controller**: hardware that connects the device to the computer.
  - continuously monitors and controls the operation o f the device.
  - provides an interface to the computer.
- Controller's tasks
  - Control the physical operation of the device
  - Convert serial bit stream to block of bytes
  - Perform error correction as necessary
- Since several devices need to be connected to a computer, they are connected trough the **bus**.

PPTs from cheng.cms.stu.edu/chap04.p

Device Controller Interface



busy	done	
0	0	idle
0	1	finished
1	0	working
1	1	(undefined)

Just like a special CPU chip: it could understand the instruction (with parameters), and drive the devices

- The device communicates with the computer via a communication point called a **port**. [端口号]

- Exchange data with CPU via registers
  - By writing into these registers
    - OS can command the device to deliver or accept data, to switch the device on or off
  - By reading from the registers
    - OS can learn the status of the device

PPTs from others\epcc.sjtu.edu.cn_~os\OS16.ppt

Device I/O Port Locations on PCs (partial)

I/O address range (hexadecimal)	device
000–00F	DMA controller
020–021	interrupt controller
040–043	timer
200–20F	game controller
2F8–2FF	serial port (secondary)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)

We also need device **Drivers** [驱动程序]

- The software that talks to the device controllers
  - Device specific
  - Tailored to individual device characteristics
  - Written by device manufacturers
  - Part of the OS Kernel

- Know about the details of the devices
  - Disk driver knows about sectors, tracks, cylinders, heads, arm motions, motor drives
  - Mouse driver knows about button pressed

User/OS method interface

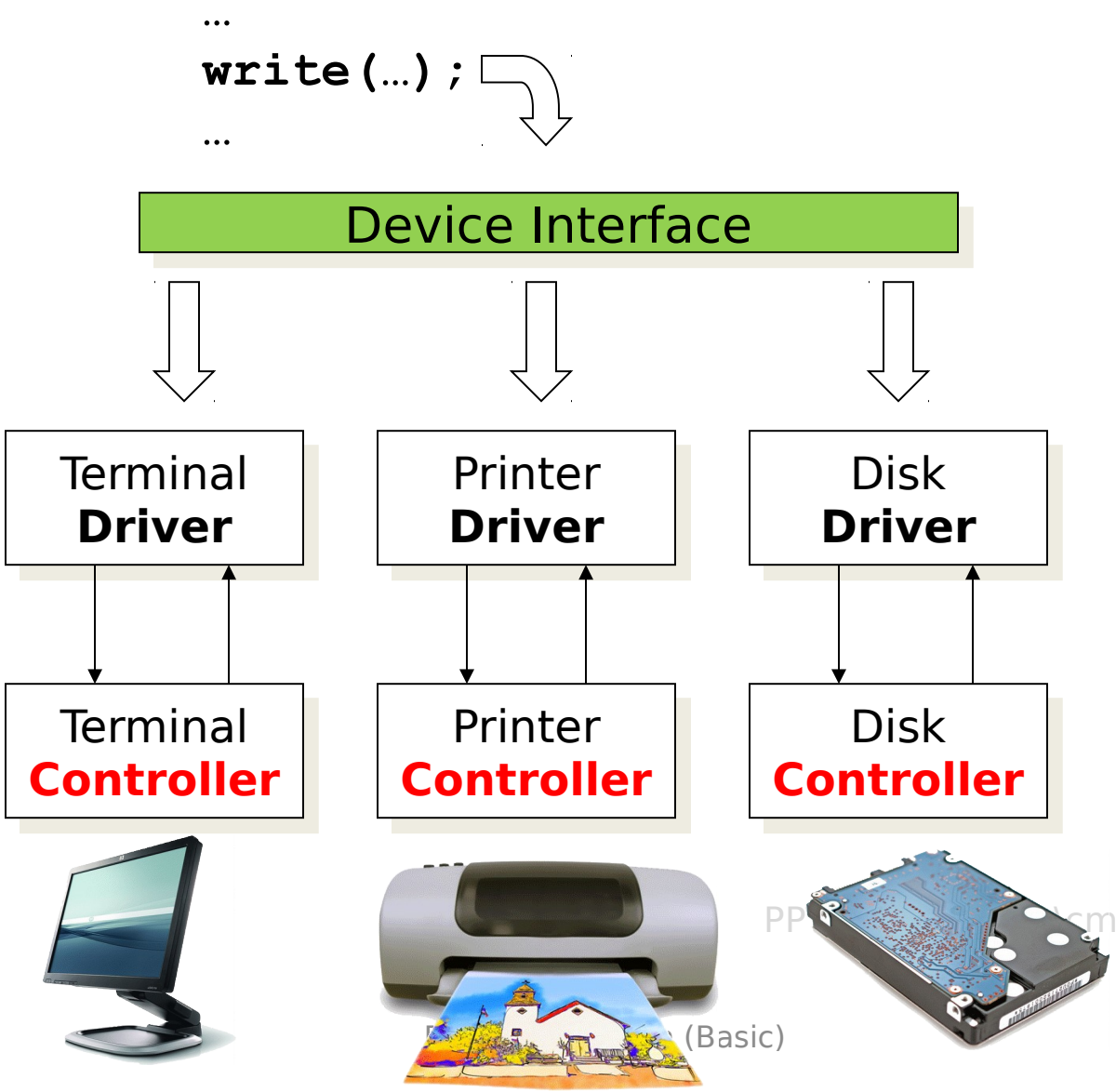
- The same interface is used to access devices (like disks and network lines) and more abstr act resources like files

- 4 main methods:
  - **open**(), **close**(), **read**(), **write**()

- Semantics depend on the type of the device (block, char, net)
  - These methods are system calls because they ar e the methods the OS provides to all processes.

PPTs from others\paul.rutgers.edu_cs519_S02\IO-File.ppt

The Device Driver Interface



Now we know how to connect **Hdisk**

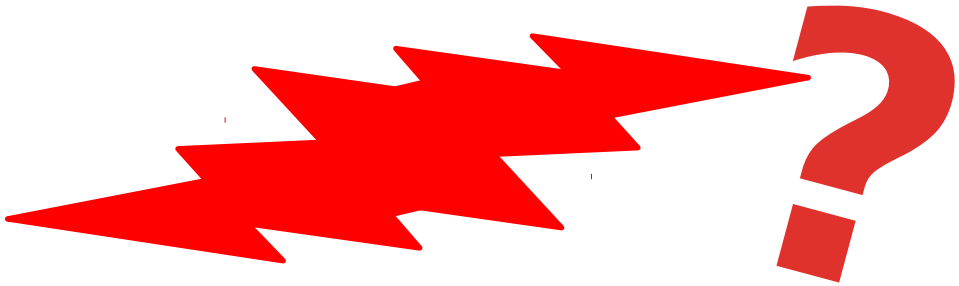
Processes

File System Calls

User's (process's) view of files

File 1

File 2

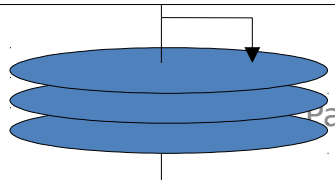


What does the Hdisk space look like?

Disk Driver

Disk Controller

Disk



Basic IO

- General structure to connect devices –
  - Abstraction/Interface in OS to communicate with the diverse devices
    - IO devices - **Categories** of devices
    - **IO operations**
      - **Connect** all together & structure of IO module in OS
      - **Communication** types between CPU and IO devices
- Taking (Magnetic) Disk for instance
  - So-called "linear address sector space"
  - Organize sectors into partitions, and so-called "linear addressed block space"
  - Optical disk is similar

The basis of magnetic storage media

- Magnetic storage stores data by magnetizi ng microscopic particles on the disk or tap e surface

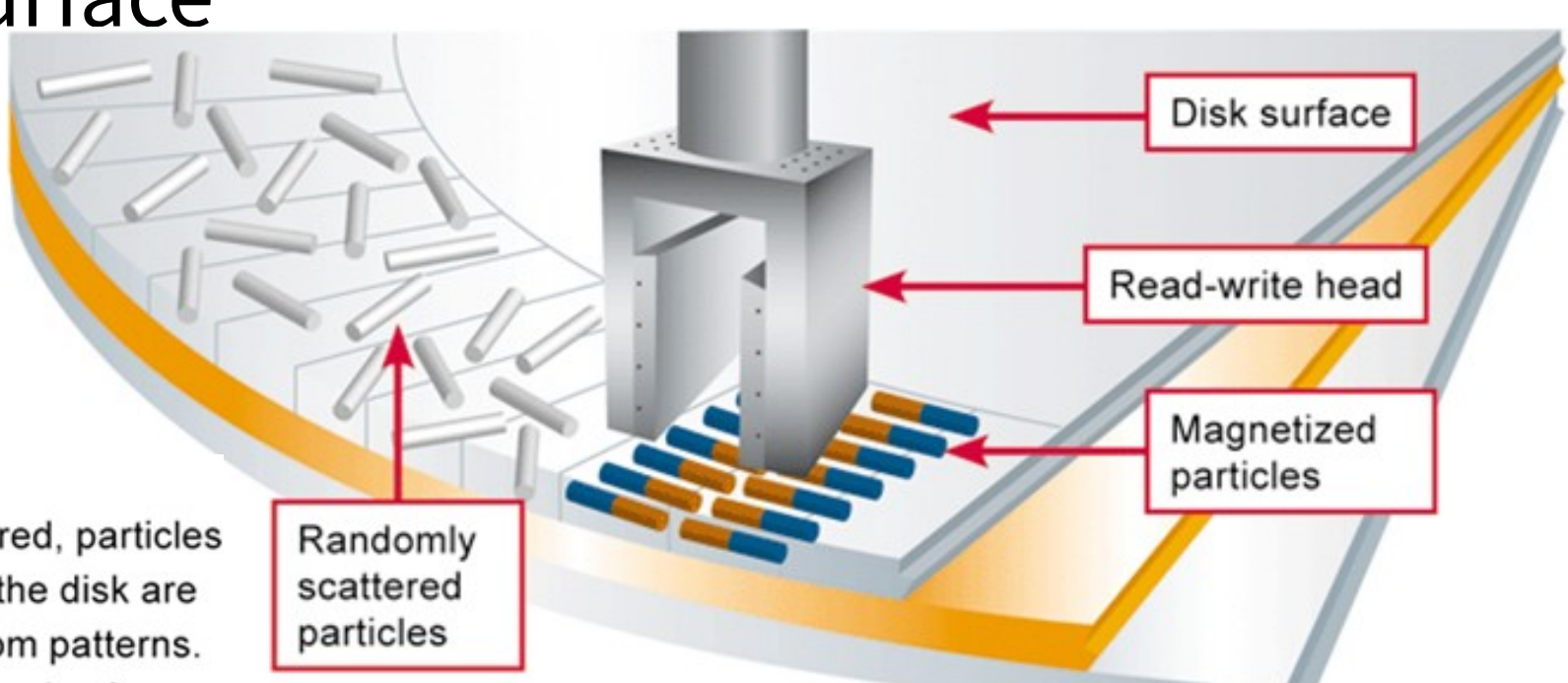


FIGURE 2-23

Before data is stored, particles on the surface of the disk are scattered in random patterns. The disk drive's read-write head magnetizes the particles, and orients them in a positive (north) or negative (south) direction to represent 0 and 1 bits.

magnetism ['mægnitizəm]
n. 磁 (性), 磁力 [学]; 磁力

Magnetic Disk

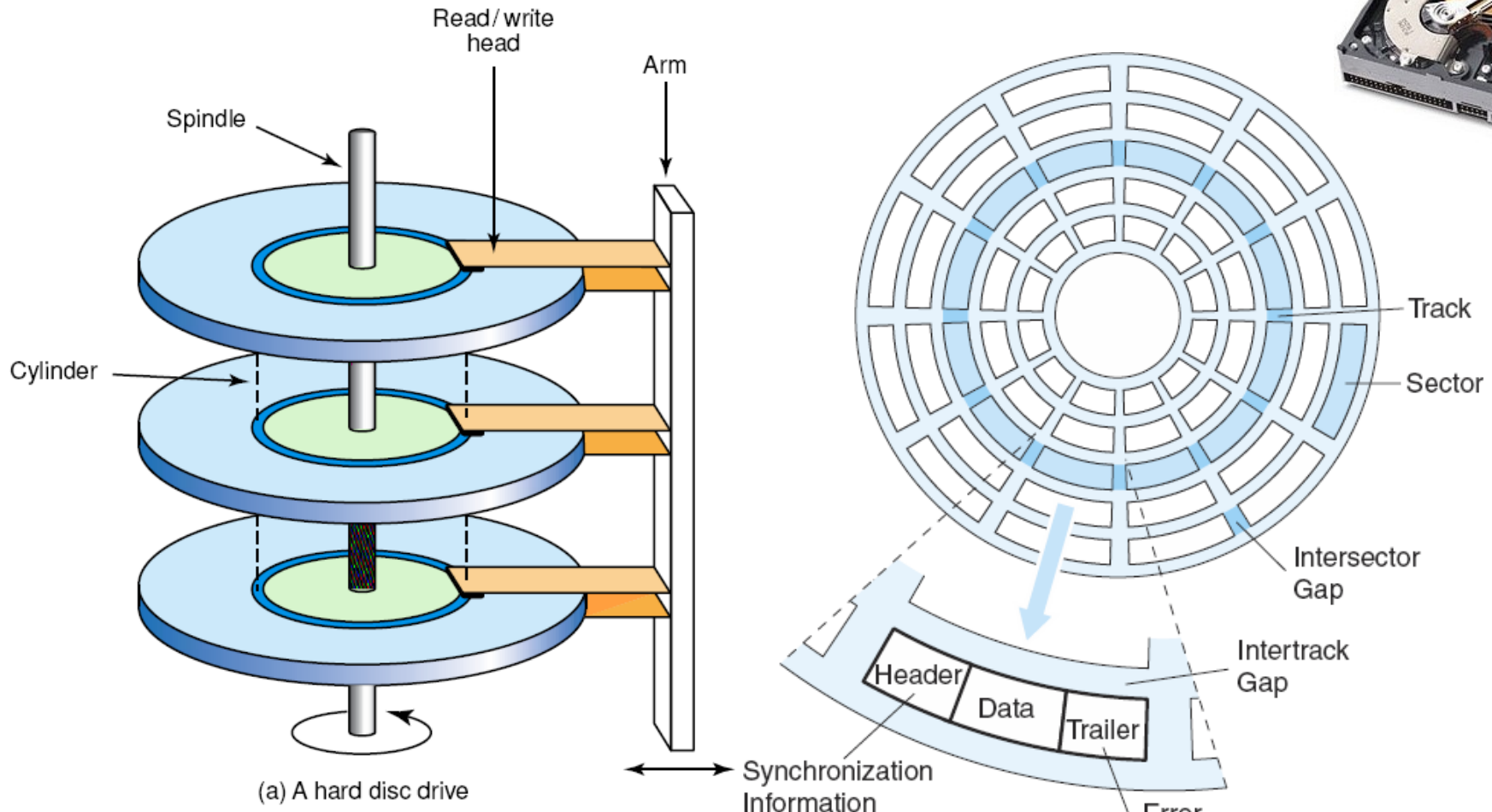


Figure 5.5 The organization of a magnetic disk

FIGURE 7.9
Disk Sectors Showing Intersector Gaps and Logical Sector Format

Computer_Science_Illuminated_N.Dale_J.Lewis. 2002

Spindle: 轴
Cylinder['silində]:

Cont'

- Numbering for th
  - (**c**ylinder #, **h**ead #, **s**

organized by the predefined rules, like the inner most one is numbered 0

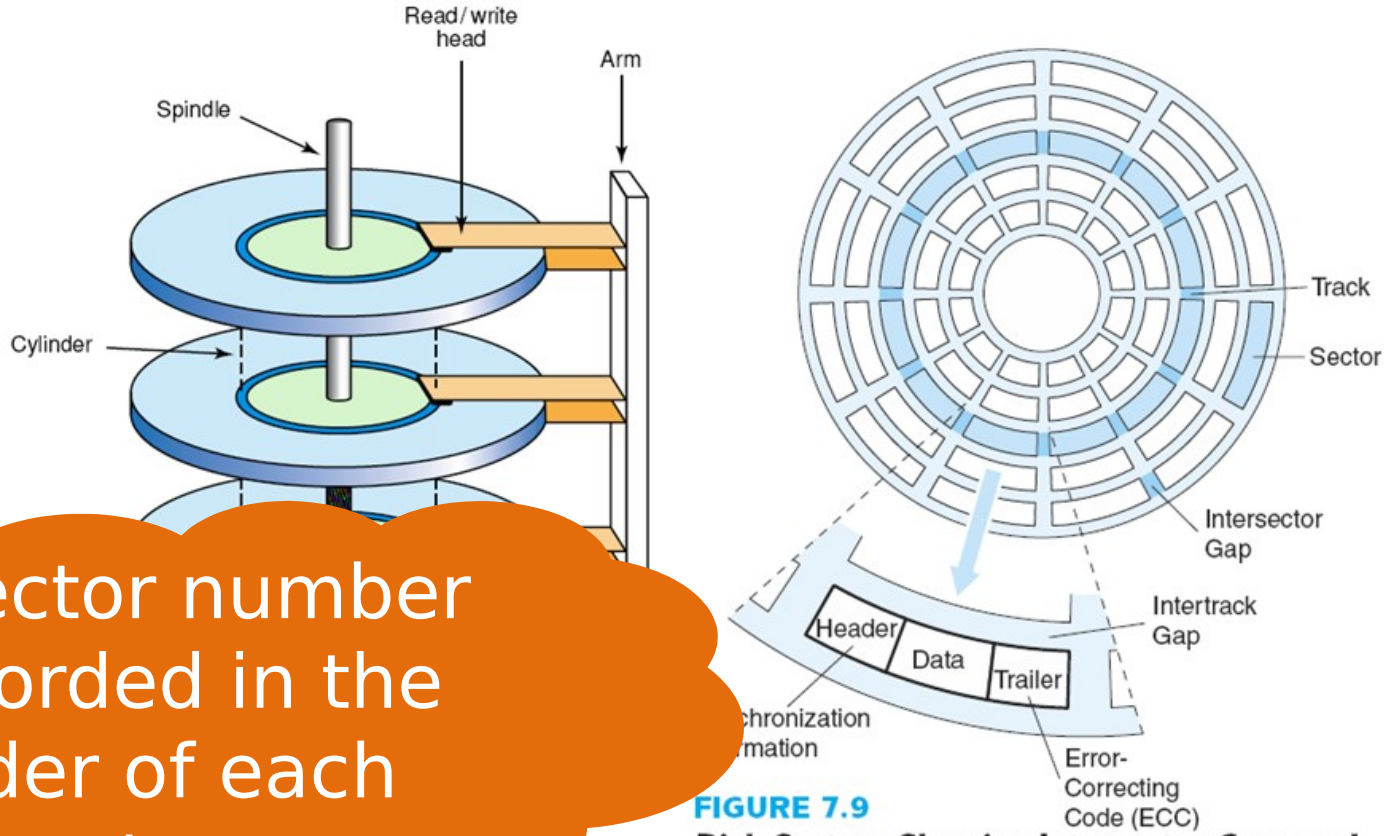


FIGURE 7.9
Disk Sectors Showing Intersector Gaps and Logical Sector Format

The sector number is recorded in the header of each sector.

Mapping files to blocks and sectors?

Processes

File System Calls

User's (process's) view of files

File 1

File 2

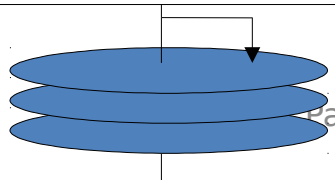
?

Map files to sectors?

Disk Driver

Disk Controller

Disk



Cylinders,
Header/tracks,
Sectors

0	1	2	3	4	5
6	7	8	9	10	11

Sector space (like storage units in MM)

LBA now for CHS

https://en.wikipedia.org/wiki/Logical_block_addressing

- Logical block addressing
  - Logical block addressing (LBA) is a common scheme used for specifying the location of blocks of data stored on computer storage devices, generally secondary storage systems such as hard disk drives.

CHS tuples can be mapped to LBA address with the following formula:[5][6]

$$LBA = (C \times HPC + H) \times SPT + (S - 1)$$

where

- C, H and S are the cylinder number, the head number, and the sector number
- LBA is the logical block address
- HPC is the maximum number of heads per cylinder (reported by disk drive, typically 16 fo
- SPT is the maximum number of sectors per track (reported by disk drive, typically 63 for

LBA addresses can be mapped to CHS tuples with the following formula ("mod" is the modulo o and "÷" is integer division, i.e. the quotient of the division where any fractional part is discard

$$C = LBA \div (HPC \times SPT)$$
$$H = (LBA \div SPT) \bmod HPC$$
$$S = (LBA \bmod SPT) + 1$$

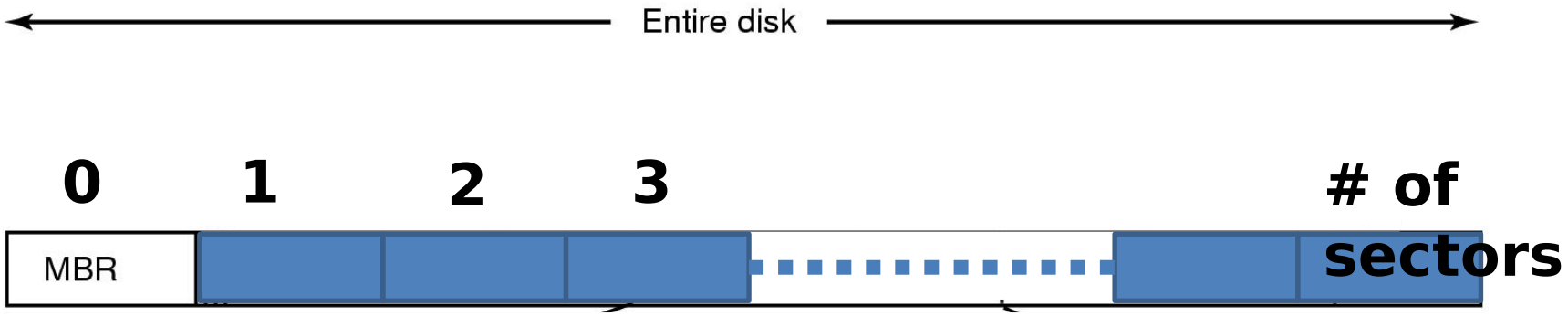
LBA and CHS equivalence with 16 heads per cylinder

LBA value	CHS tuple
0	0, 0, 1
1	0, 0, 2
2	0, 0, 3
62	0, 0, 63
63	0, 1, 1
945	0, 15, 1
1007	0, 15, 63
1008	1, 0, 1
1070	1, 0, 63
1071	1, 1, 1
1133	1, 1, 63
1134	1, 2, 1
2015	1, 15, 63
2016	2, 0, 1
16,127	15, 15, 63
16,128	16, 0, 1
32,255	31, 15, 63
32,256	32, 0, 1
16,450,559	16319, 15, 63
16,514,063	16382, 15, 63

Basic IO

- General structure to connect devices –
  - Abstraction/Interface in OS to communicate with the diverse devices
    - IO devices - **Categories** of devices
    - **IO operations**
      - **Connect** all together & structure of IO module in OS
      - **Communication** types between CPU and IO devices
- Taking (Magnetic) Disk for instance
  - So-called "linear address **sector** space"
    - Like the storage units in MM
  - Organize sectors into partitions, and so-called "linear addressed **block** space"
  - Optical disk is similar

Some special sectors in HD



- After LLF (low level formatting, which has usually been carried out by HD manufacturer like Seagate, WD Western Digital) lt your disk is a collecti on of numbered sectors
  - Divide a disk into sectors that the **controller** can read and write (write C.H.S into header, etc.)
- Sector 0 is called the **M**aster **B**oot **R**ecord (**MBR** : 主引导记录) which is used when booting the computer
  - Some information about the manufacturing – date, size …
  - At the end, the table of **PARTITION**s – "logic disks" in th e HD

Sector is too small!

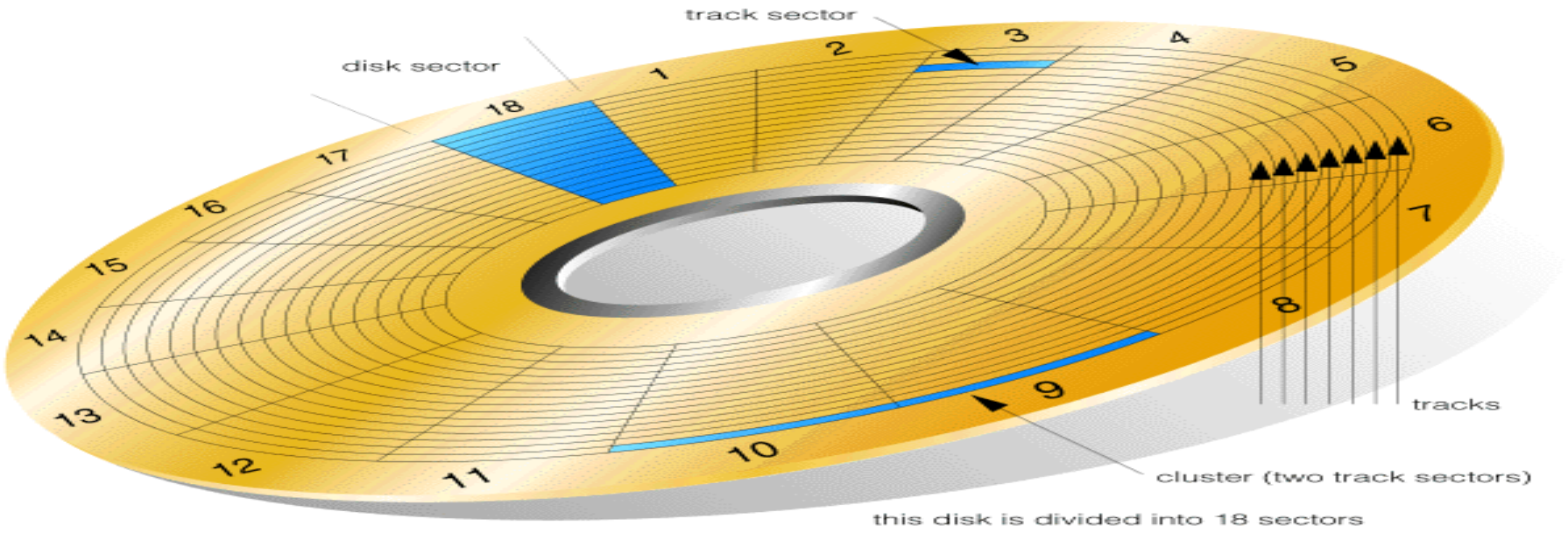
Several sectors are combined to create clusters or bloc
ks

**cluster (Windows and Macs) or block (UNIX) = The n
umber of sectors which is allocated on the disk e
ach time a file needs space on the disk.**

NTFS?

Windows 95 (later versions) and Windows 98 using FAT32

- 1 cluster = 8 sectors (4K bytes)



Disk Partition [磁盘分区]

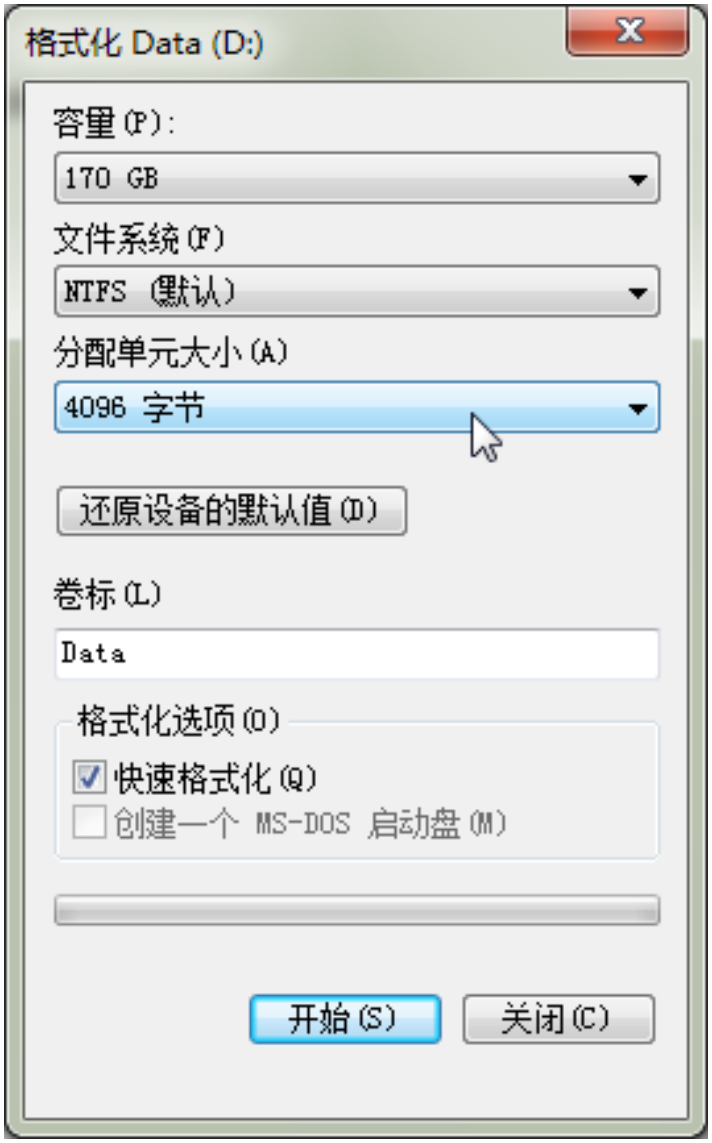
- To use a disk to hold files, OS still needs to rec ord its own data structures on the disk
- **Partition** the disk int ylinders
  - Each partition can
- **Logical formatting** or "making syste m"
  - Store the initial file-sys disk…
    - Maps of free and a ed later)
    - Initial empty directory

We'll learn FAT, i-node, NTFS – the way to organize files - later

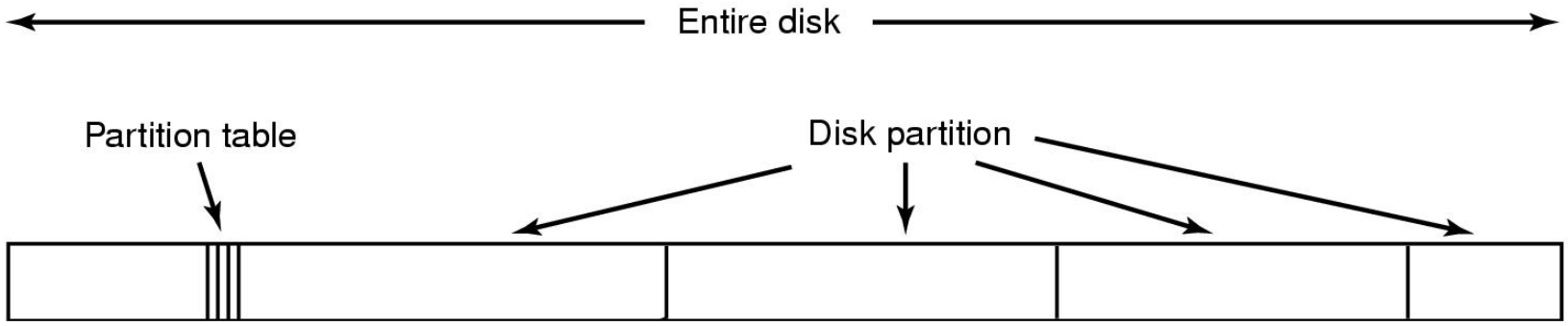
In fact this is like organize storage units of MM into frames

First, organize sectors in a partition into **blocks**

- The popular size in Windows now is 4KB (4096 bytes)
  - Which means a block usually contains 8 sectors
- You can set the File System you want to use for this partition
  - NTFS (NT File System) here
- You can also label it
  - Volume index here, "Data"



PARTITIONs in HD



MBR

- Do you remember "What does OS do after turning-on but before your OS?"
  - After POST, BIOS (CMOS) will check the HDs to locate the OS (s) by reading the MBR of each HD
    - The partition table in MBR contains all the location information (starting CHS) of partitions
  - Each partition is further organized to contain the files based on a complex data structure called **FILE SYSTEM**
    - If in primary partition, bootstrap loader reads necessary OS's codes with help of file system into main memory, and CPU executes those programs – you can use OS now

http://en.wikipedia.org/wiki/Master_boot_record

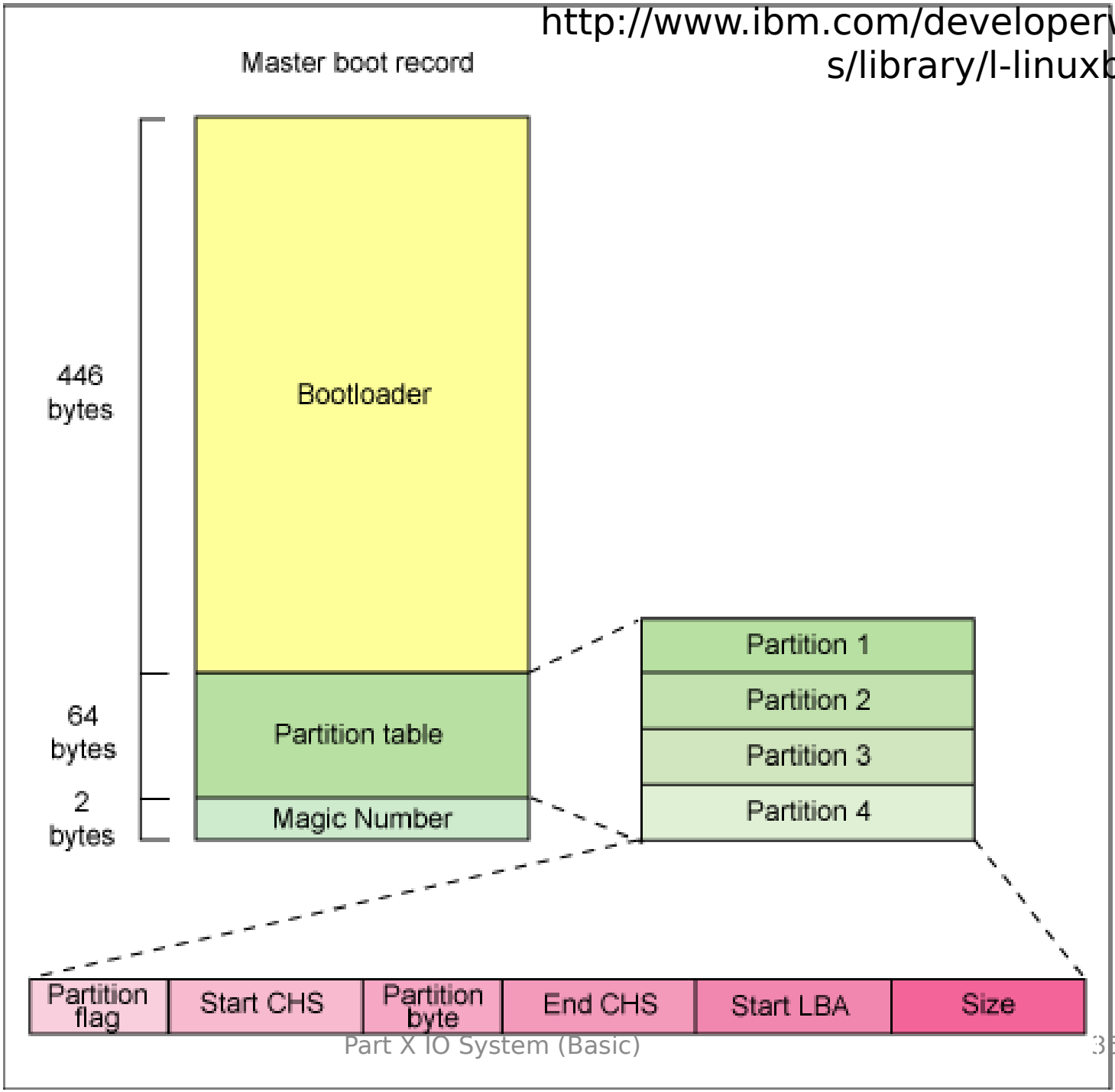
MBR – for demonstration

bytes for "partition entry" are used to indicate the 1st sector number of the corresponding partition

MBR

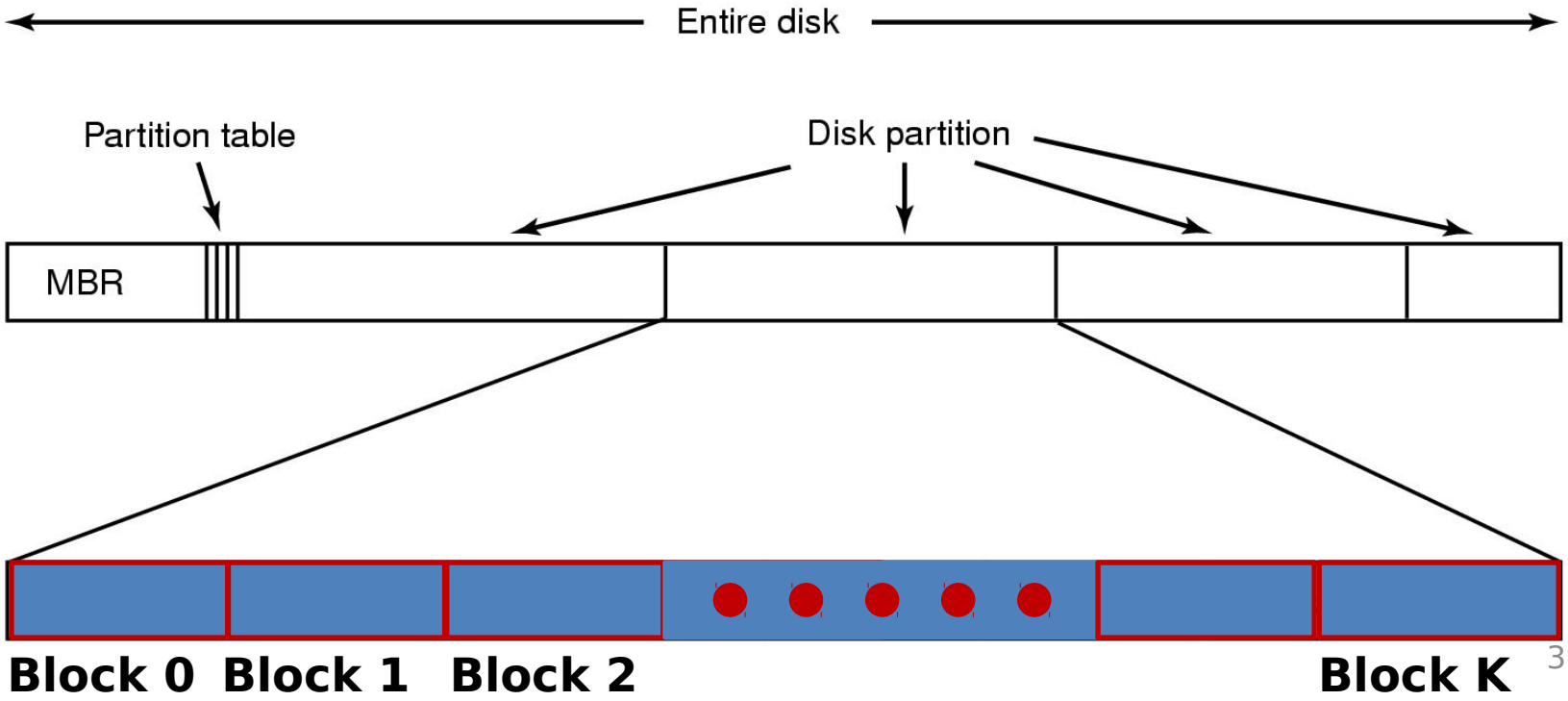
				Size in bytes
				446
				16
+1CEh	+462	Partition entry #2	*Partition table* (for primary partitions)	16
+1DEh	+478	Partition entry #3		16
+1EEh	+494	Partition entry #4		16
+1FEh	+510	55h	*Boot signature*[nb 1]	2
+1FFh	+511	AAh		
Total size: 446 + 4*16 + 2				**512**

http://www.ibm.com/developerworks/library/l-linuxboot/



Disk Space Organization

- Disk can be **PARTITION**ed
  - Each partition can have a different OS and/or diff erent file system
    - One partition can be swap space for main memory
- Each partition has



Organize sectors into blocks?

- Given a partition whose starting sector is 1024, and the size of a block is defined as 4KB, could you determine the sectors for a block 7 in this p artition?
  - Since 4 KB = 8*512B = 8 sectors, the 1st sector of the block 7 should be: 7*8 = 56th sector
  - So the 1st sector of block 7 in this partition is just: 10 24 + 56 = 1080 sector
- If you want to store a file of 17 KB, it’ s easy to know we need ⌈17/4⌉ = 5 blocks.
  - Of course we need record some information to retrieve the blocks use a structure!

We’ll learn those data structures later

Now we have linear addressed block space for files

Processes

File System Calls

User's (process's) view of files

File 1

File 2

Manage files based on blocks? – Need data structures

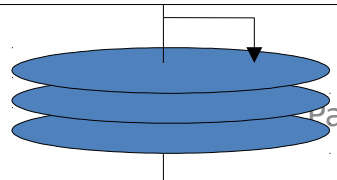
0	1	2	3	4	5

file system's view of the disk

Disk Driver

Disk Controller

Disk

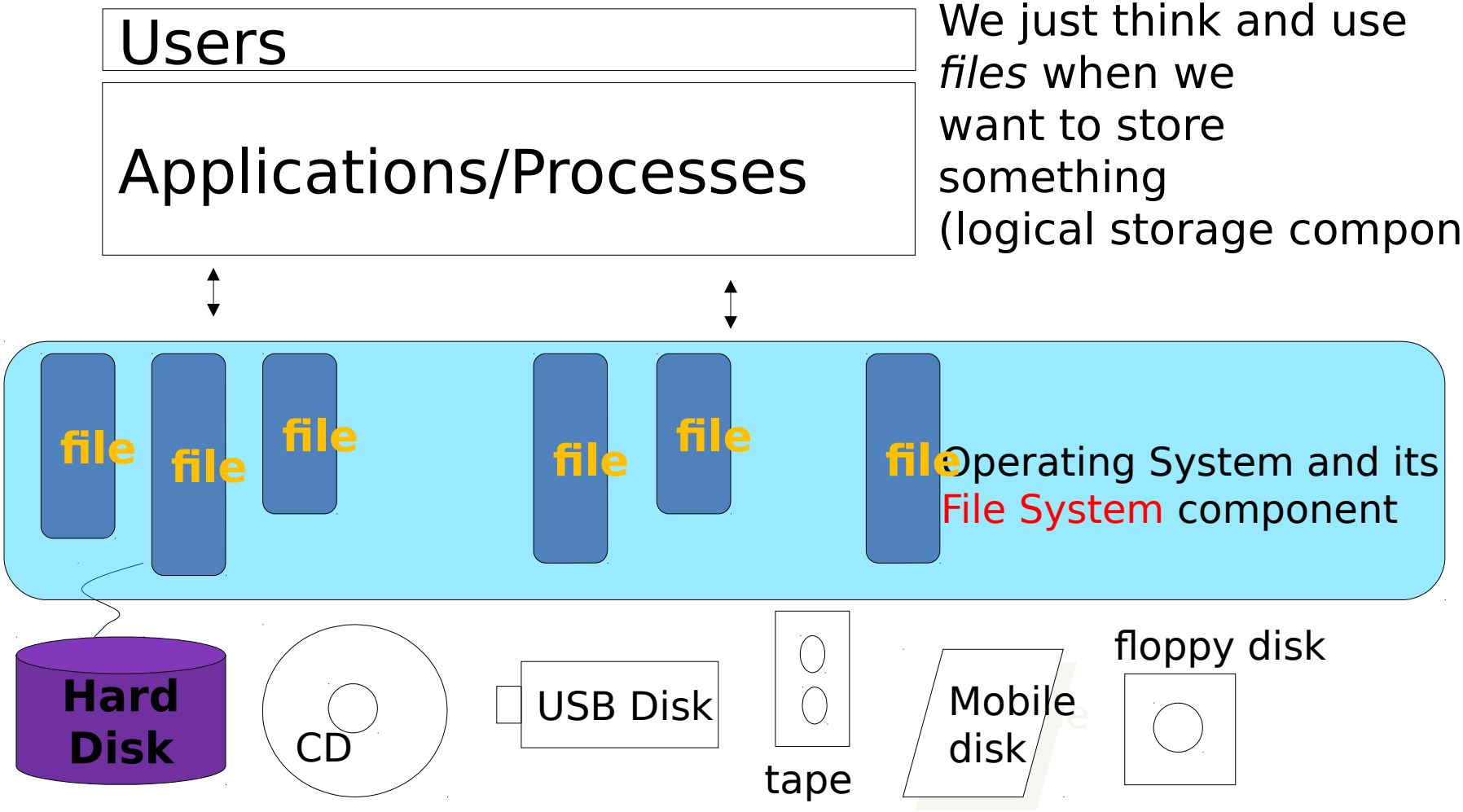


Cylinders,
Header/tracks,
Sectors

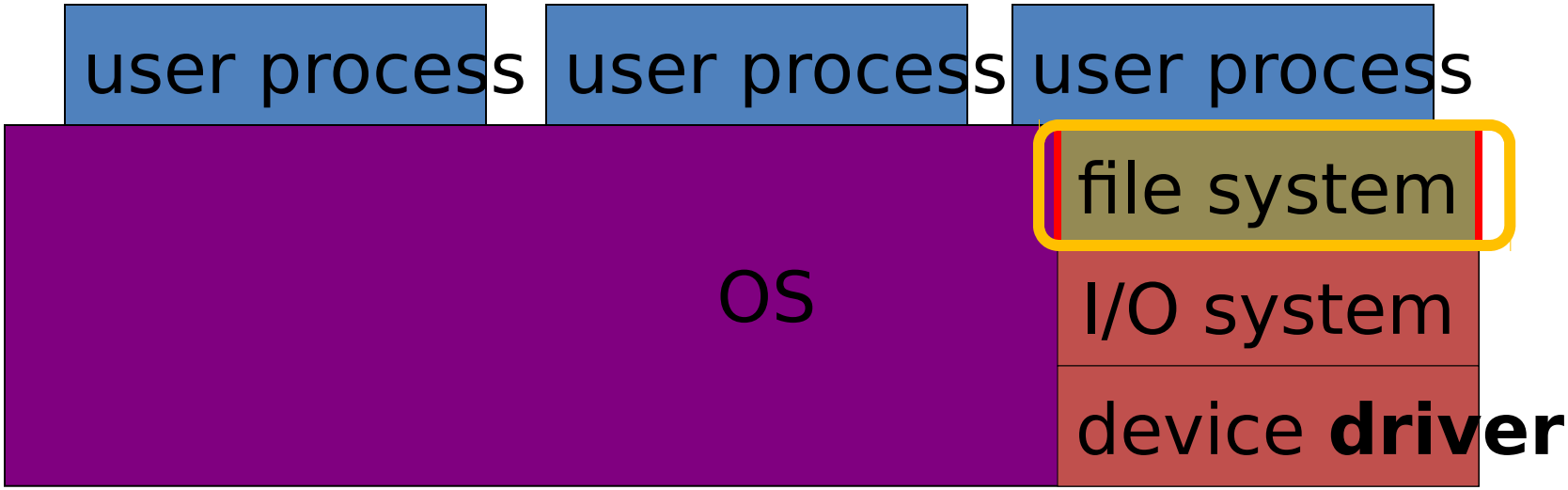
0	1	2	3	4	5
6	7	8	9	10	11

Sector space (like storage units in MM)

Those data structures are maintained by **File system**



I/O System – connect devices together



PPTs.2012\PPTs from others\pages.cs.wisc.edu_~remzi_Classes_537_Fall2005\lecture19.ppt

Basic IO

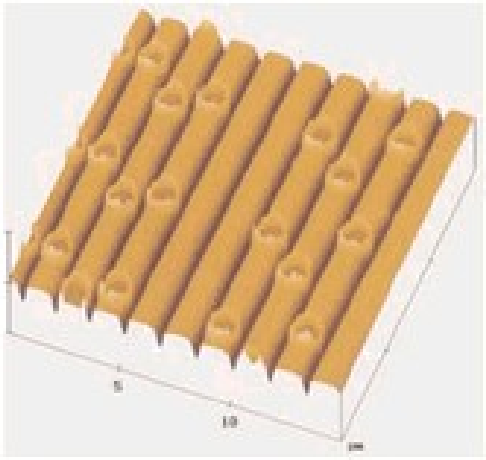
- General structure to connect devices –
  - Abstraction/Interface in OS to communicate with the diverse devices
    - IO devices - **Categories** of devices
    - **IO operations**
      - **Connect** all together & structure of IO module in OS
      - **Communication** types between CPU and IO devices
- Taking (Magnetic) Disk for instance
  - So-called "linear address sector space"
  - Organize sectors into partitions, and so-called "linear addressed block space"
  - Optical disk is similar

CD (Compact Disk)

crater ['kreitə] n. 弹坑；陨石坑
pit n. 坑

FIGURE 2-26

As seen through an electron microscope, the pits on an optical storage disk look like small craters. Each pit is less than 1 micron (one millionth of a meter) in diameter—1,500 pits lined up side by side are about as wide as the head of a pin.



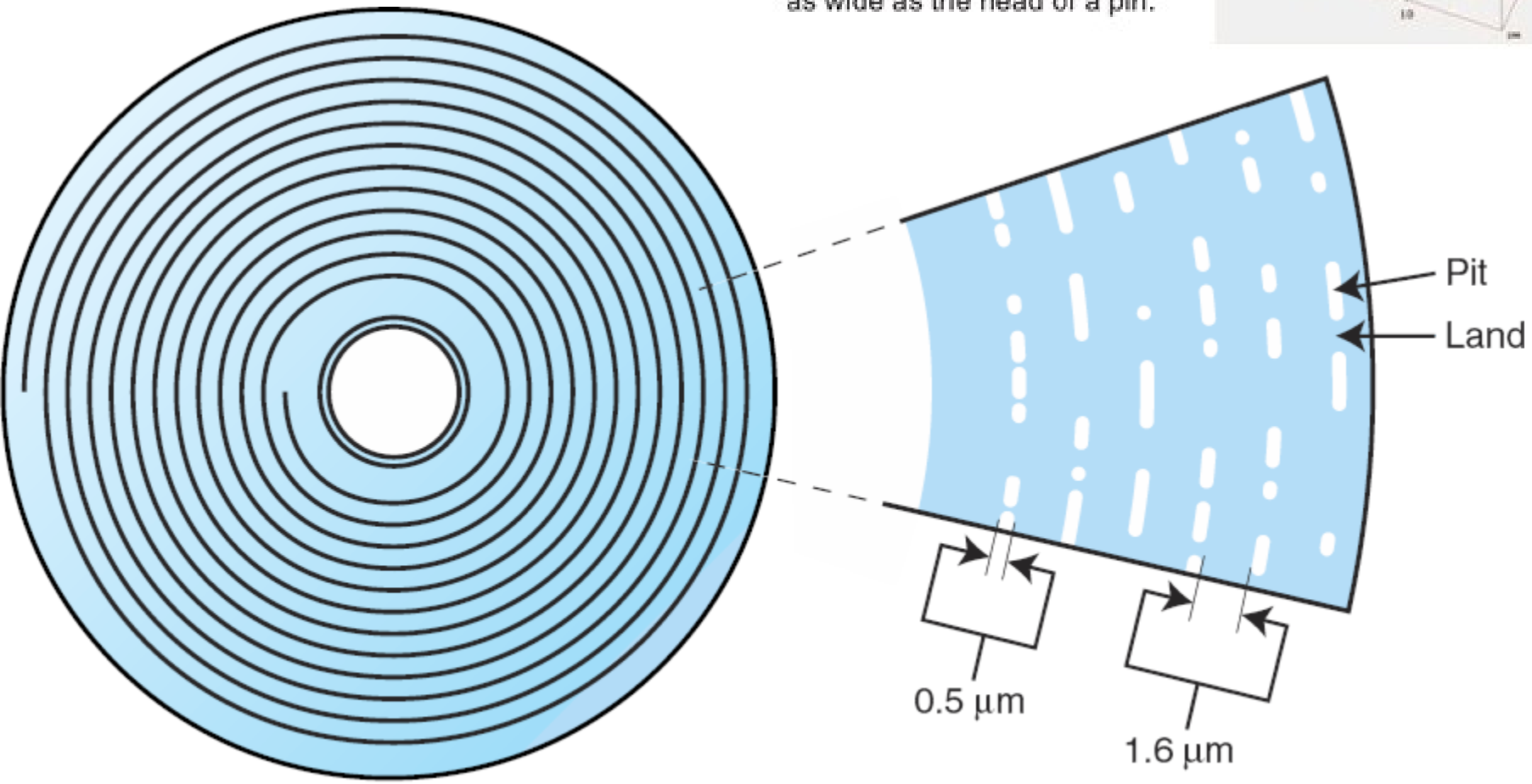
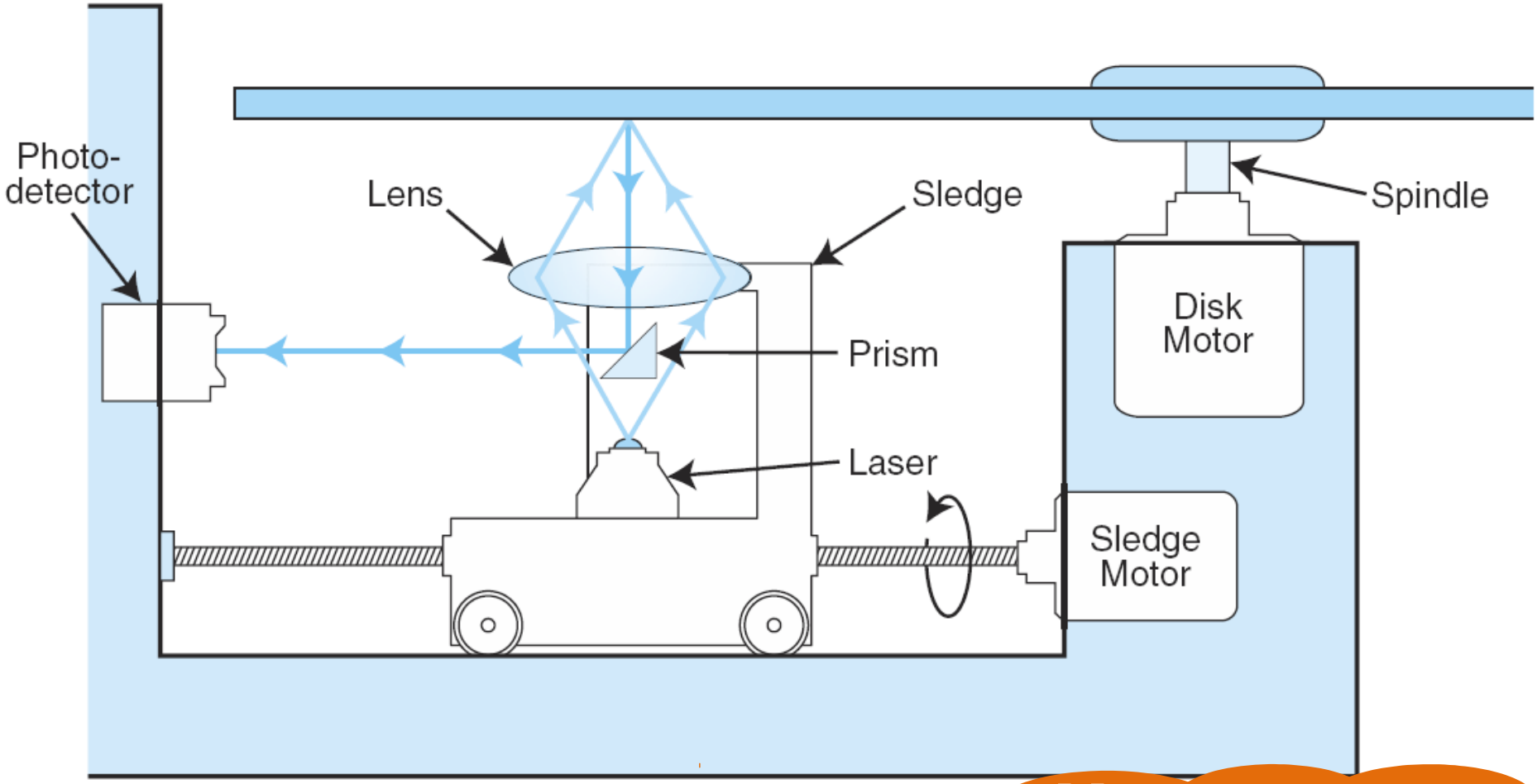


FIGURE 7.14 CD Track Spiral and Track Enlargement

The Internals of a CD-ROM Drive



ledge: 雪橇，雪车

rism: (几何)棱柱（体），角柱（体）

Your responsibility to know others.