Operating Systems: Internals and Design Principles

Chapter 11 I/O Management and Disk Scheduling

Ninth Edition
By William Stallings

Categories of I/O Devices

External devices that engage in I/O with computer systems can be grouped into three categories:

Human readable

- Suitable for communicating with the computer user
- Printers, terminals, video display, keyboard, mouse

Machine readable

- Suitable for communicating with electronic equipment
- Disk drives, USB keys, sensors, controllers

Communication

- Suitable for communicating with remote devices
- Modems, digital line drivers

Differences in I/O Devices

- Devices differ in a number of areas:

Data Rate

- There may be differences of magnitude between the data transfer rates

Application

- The use to which a device is put has an influence on the software

Complexity of Control

- The effect on the operating system is filtered by the complexity of the I/O module that controls the device

Unit of Transfer

- Data may be transferred as a stream of bytes or characters or in larger blocks

Data Representation

- Different data encoding schemes are used by different devices

Error Conditions

- The nature of errors, the way in which they are reported, their consequences, and the available range of responses differs from one device to another

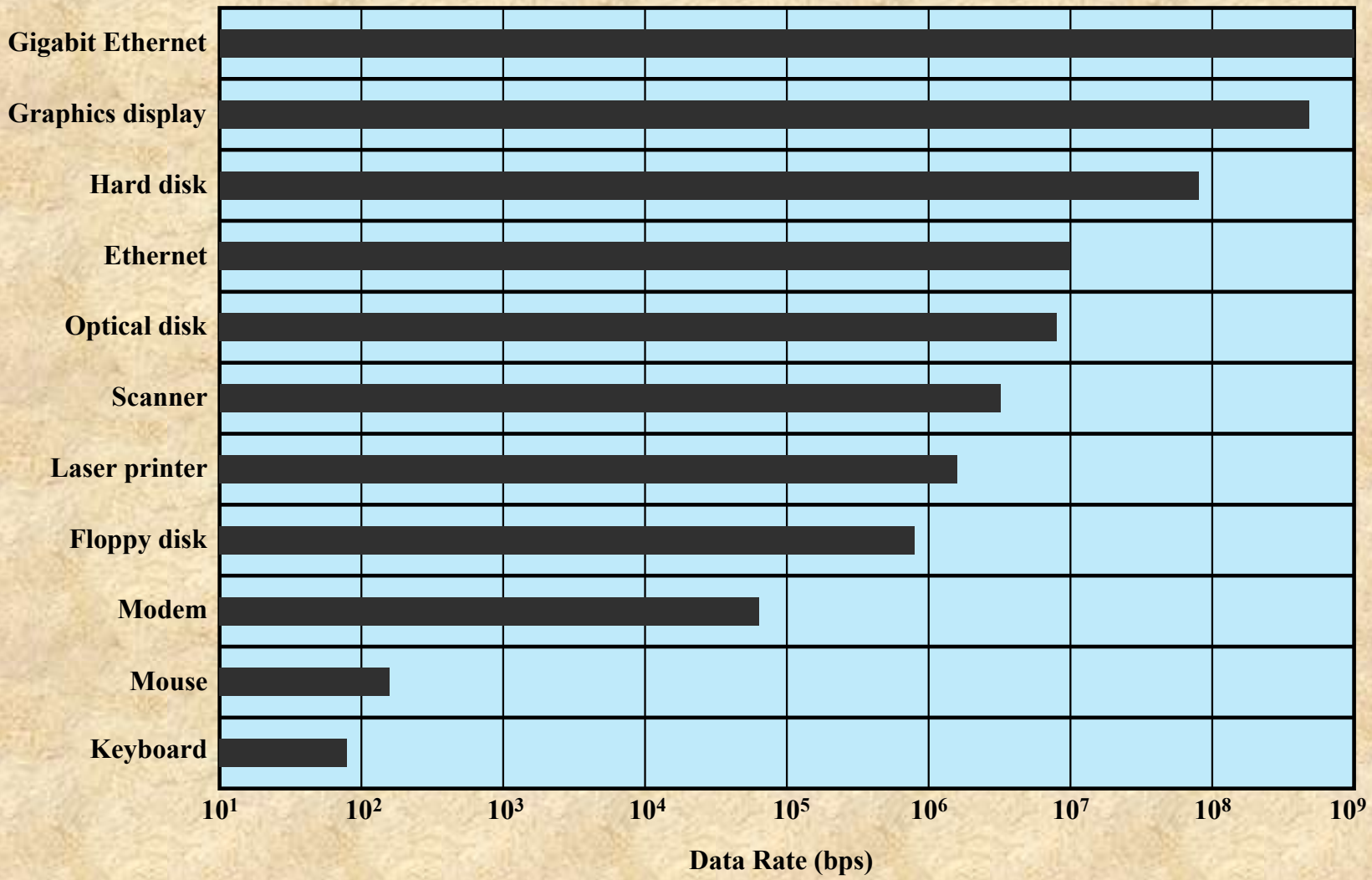


Figure 11.1 Typical I/O Device Data Rates

Organization of the I/O Function

Three techniques for performing I/O are:

- **Programmed I/O**
  - The processor issues an I/O command on behalf of a process to an I/O module; that process then busy waits for the operation to be completed before proceeding
- **Interrupt-driven I/O**
  - The processor issues an I/O command on behalf of a process
    - If non-blocking – processor continues to execute instructions from the process that issued the I/O command
    - If blocking – the next instruction the processor executes is from the OS, which will put the current process in a blocked state and schedule another process
- **Direct Memory Access (DMA)**
  - A DMA module controls the exchange of data between main memory and an I/O module

Table 11.1
I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Evolution of the I/O Function

1. Processor directly controls a peripheral device
2. A controller or I/O module is added
3. Same configuration as step 2, but now interrupts are employed
4. The I/O module is given direct control of memory via DMA
5. The I/O module is enhanced to become a separate processor, with a specialized instruction set tailored for I/O
6. The I/O module has a local memory of its own and is, in fact, a computer in its own right

Evolution of the I/O Function

Along this evolutionary path, more and more of the I/O function is performed without processor involvement. The central processor is increasingly relieved of I/O-related tasks, improving performance. With the last two steps (5 and 6), a major change occurs with the introduction of the concept of an I/O module capable of executing a program

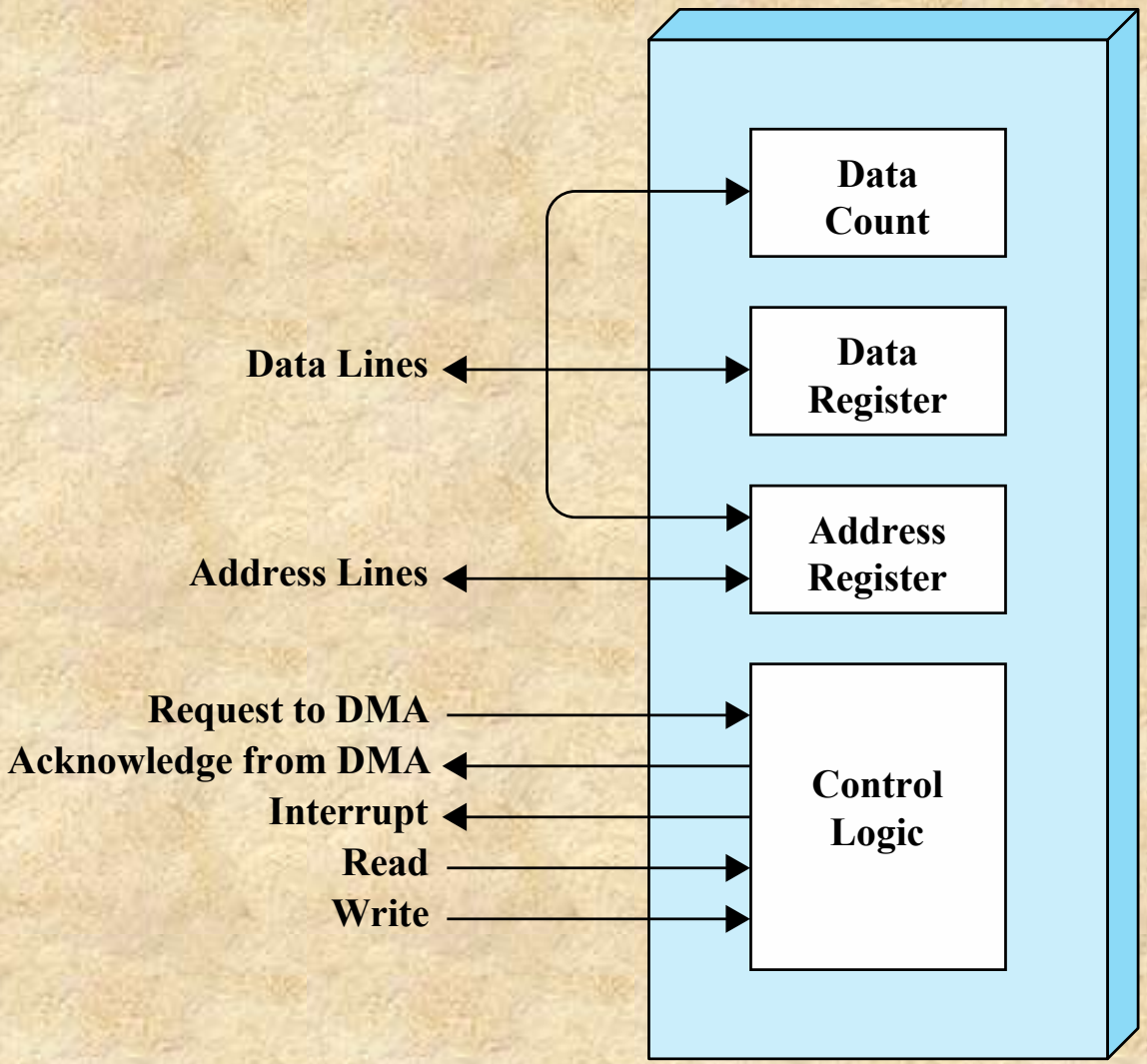
For all of the modules described in steps 4 through 6, the term *direct memory access* is appropriate, because all of these types involve direct control of main memory by the I/O module. Also, the I/O module in step 5 is often referred to as an **I/O channel**, and that in step 6 as an **I/O processor**; however, each term is, on occasion, applied to both situations. In the latter part of this section, we will use the term I/O channel to refer to both types of I/O modules

DMA

The DMA technique works as follows. When the processor wishes to read or write a block of data, it issues a command to the DMA module by sending to the DMA module the following information:

- Whether a read or write is requested, using the read or write control line between the processor and the DMA module
- The address of the I/O device involved, communicated on the data lines
- The starting location in memory to read from or write to, communicated on the data lines and stored by the DMA module in its address register
- The number of words to be read or written, again communicated via the data
- lines and stored in the data count register

The processor then continues with other work. It has delegated this I/O operation to the DMA module. The DMA module transfers the entire block of data, one word at a time, directly to or from memory, without going through the processor. When the transfer is complete, the DMA module sends an interrupt signal to the processor. Thus, the processor is involved only at the beginning and end of the transfer



The DMA unit is capable of mimicking the processor and, indeed, of taking over control of the system bus just like a processor

- It needs to do this to transfer data to and from memory over the system bus

Figure 11.2 Typical DMA Block Diagram

Design Objectives

Efficiency

- Major effort in I/O design
- Important because I/O operations often form a bottleneck
- Most I/O devices are extremely slow compared with main memory and the processor
- The area that has received the most attention is disk I/O

Generality

- Desirable to handle all devices in a uniform manner
- Applies to the way processes view I/O devices and the way the operating system manages I/O devices and operations
- Diversity of devices makes it difficult to achieve true generality
- Use a hierarchical, modular approach to the design of the I/O function

Disk Performance Parameters

- The actual details of disk I/O operation depend on the:
  - Computer system
  - Operating system
  - Nature of the I/O channel and disk controller hardware

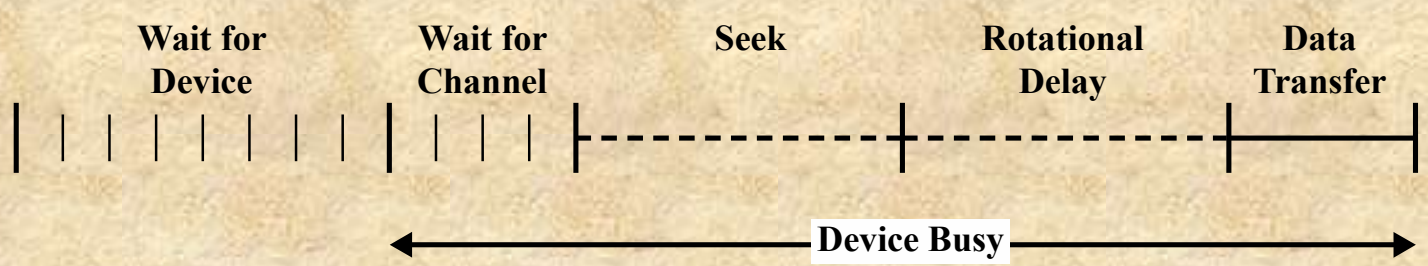
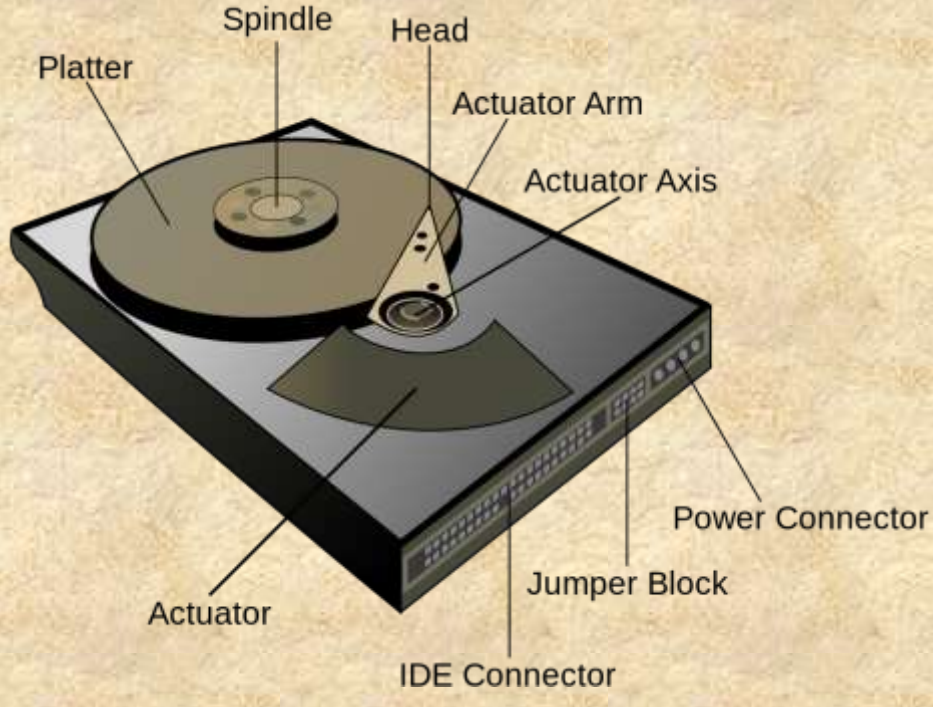
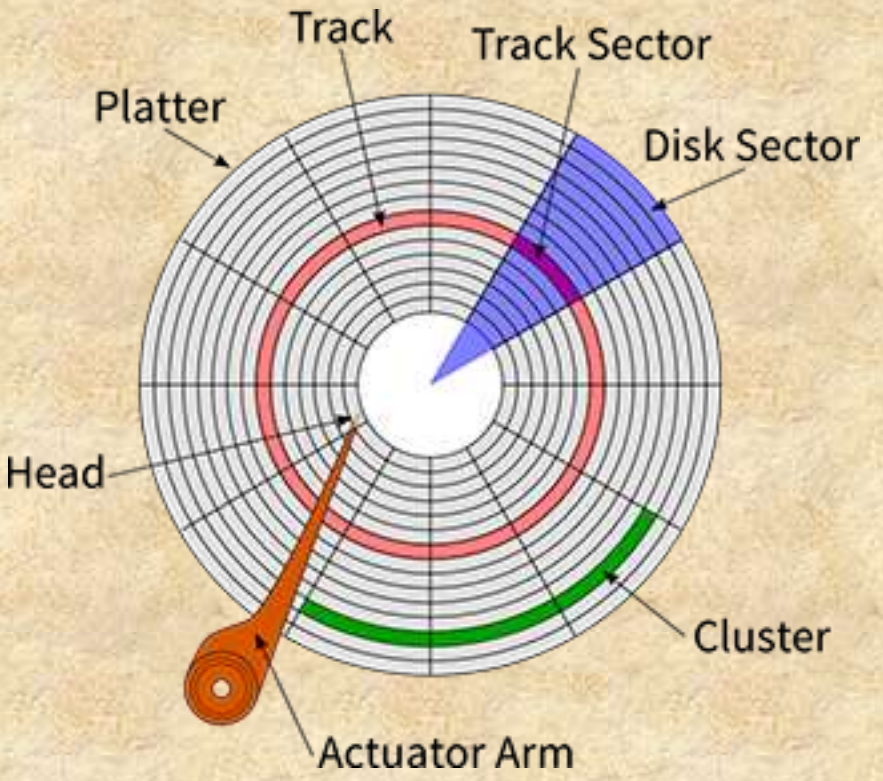


Figure 11.6 Timing of a Disk I/O Transfer

Hard Disc Drive



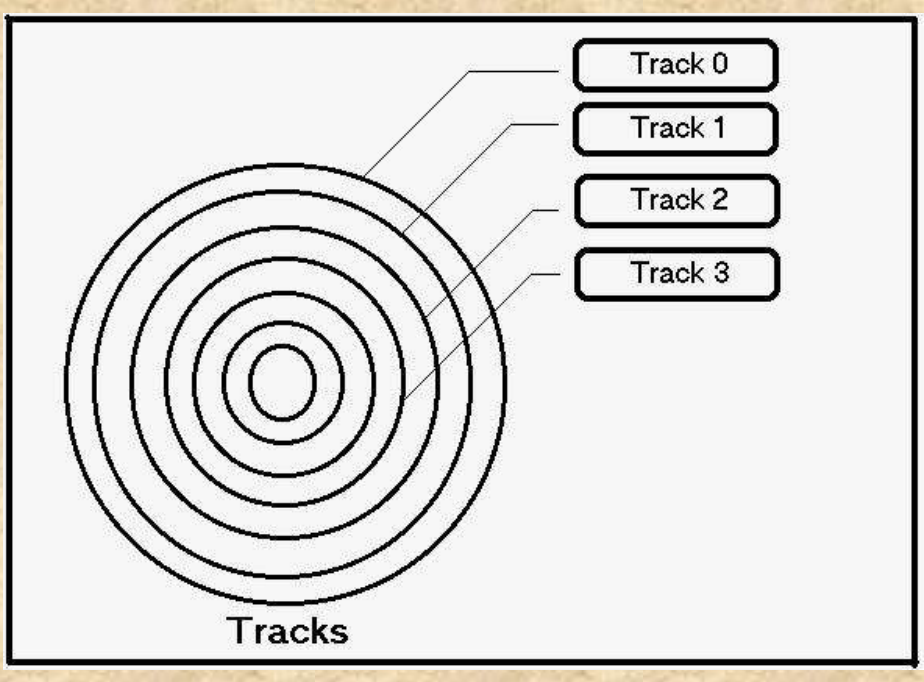


https://en.wikipedia.org/wiki/Hard_disk_drive

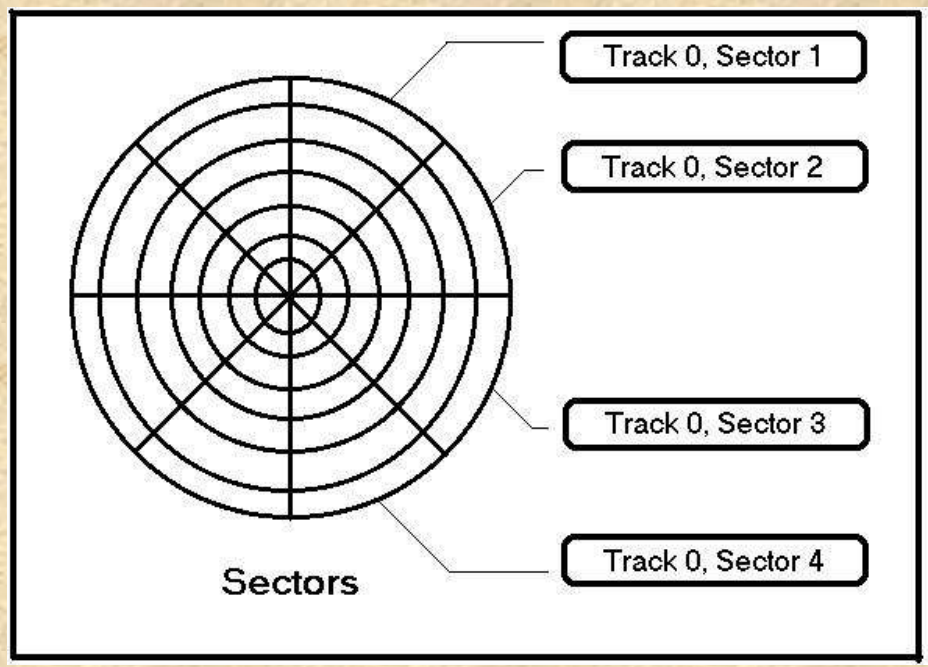
Inside of Hard Drive
https://www.youtube.com/watch?v=9eMWG3fwiEU&ab_channel=joshuamarius

https://www.datarecovery.net/articles/hard-drive-sector-damage.aspx

Hard Disc Drive



https://www.datadoctor.biz/data_recovery_programming_book_chapter2-page16.html



https://www.datadoctor.biz/data_recovery_programming_book_chapter2-page17.html

Disk Performance Parameters

- When the disk drive is operating, the disk is rotating at constant speed
- To read or write the head must be positioned at the desired track and at the beginning of the desired sector on that track
- Track selection involves moving the head in a movable-head system or electronically selecting one head on a fixed-head system
- On a movable-head system the time it takes to position the head at the track is known as **seek time**
- The time it takes for the beginning of the sector to reach the head is known as **rotational delay**
- The sum of the seek time and the rotational delay equals the **access time**

Seek Time

- The time required to move the disk arm to the required track
- Consists of two key components:
  - The initial startup time
  - And the time taken to traverse the tracks that have to be crossed once the access arm is up to speed
    - Including the settling time: time after positioning the head over the target track until track identification is confirmed
- Much improvement comes from smaller and lighter disk components
- A typical average seek time on contemporary hard disks is under 10ms

Disk Performance

- Rotational delay
  - The time required for the addressed area of the disk to rotate into a position where it is accessible by the read/write head
  - Disks rotate at speeds ranging from 3,600 rpm (for handheld devices such as digital cameras) up to 15,000 rpm
  - RPM for revolutions per minute

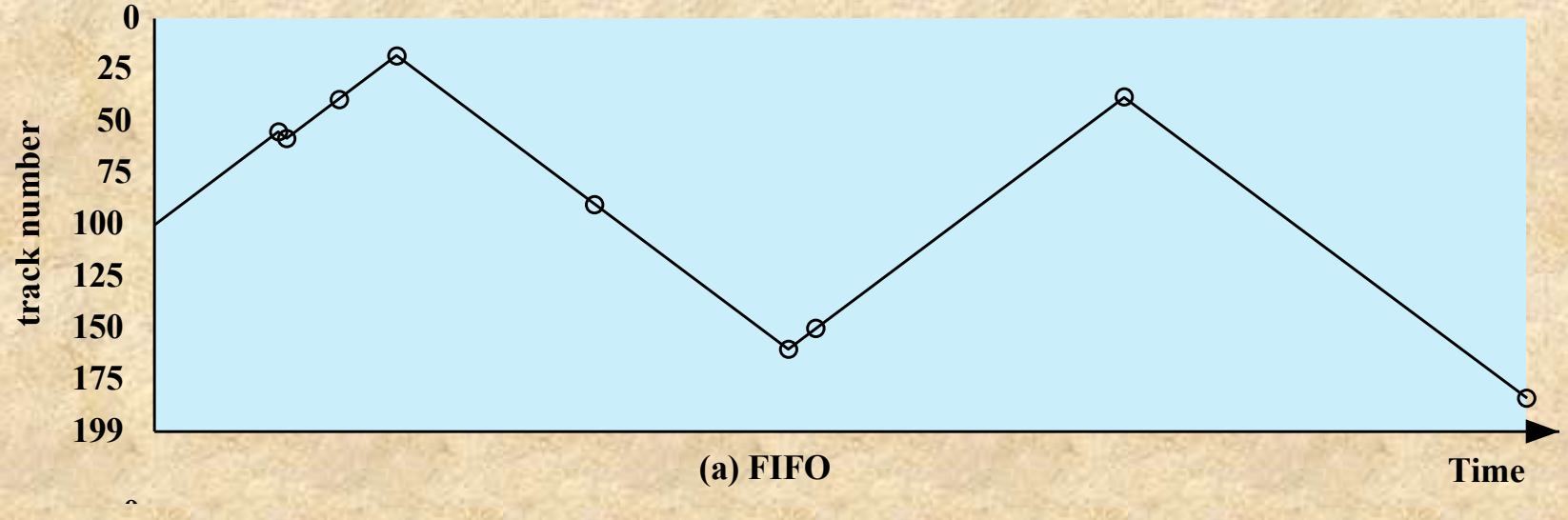
The requested tracks, in the order received by the disk scheduler, are 55, 58, 39, 18, 90, 160, 150, 38, 184.

(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	**Average seek length**	27.5	**Average seek length**	27.8	**Average seek length**	35.8

Table 11.2 Comparison of Disk Scheduling Algorithms

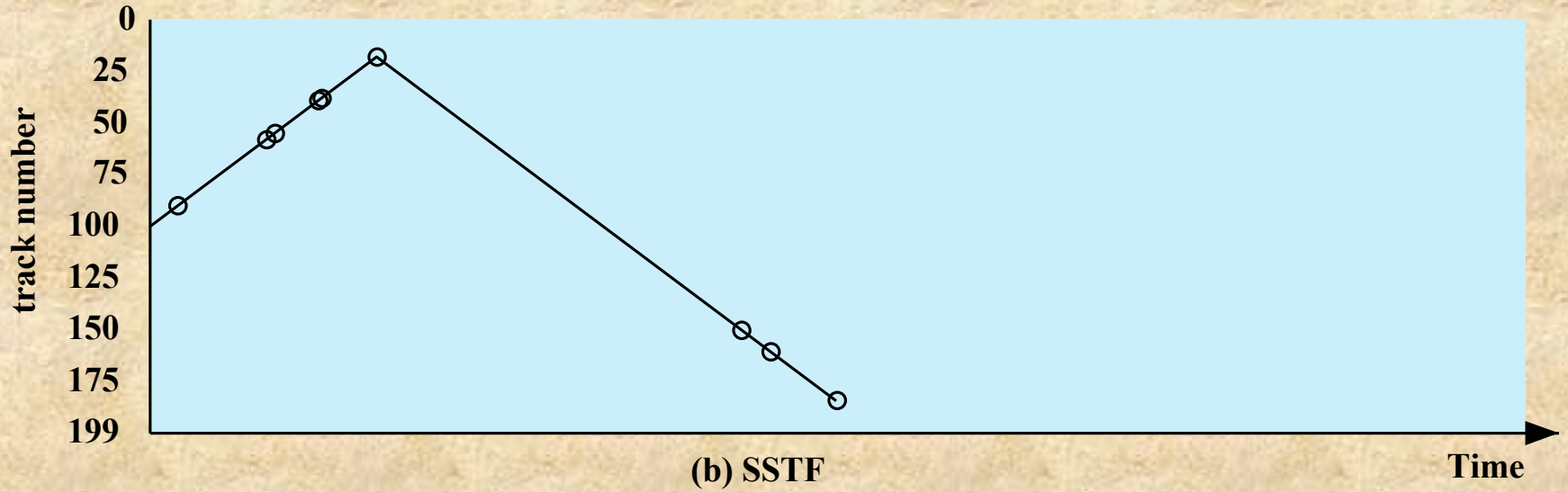
First-In, First-Out (FIFO)

- Processes in sequential order
- Fair to all processes
- Approximates random scheduling in performance if there are many processes competing for the disk
- Also called first come first serve (FCFS)



Shortest Service Time First (SSTF)

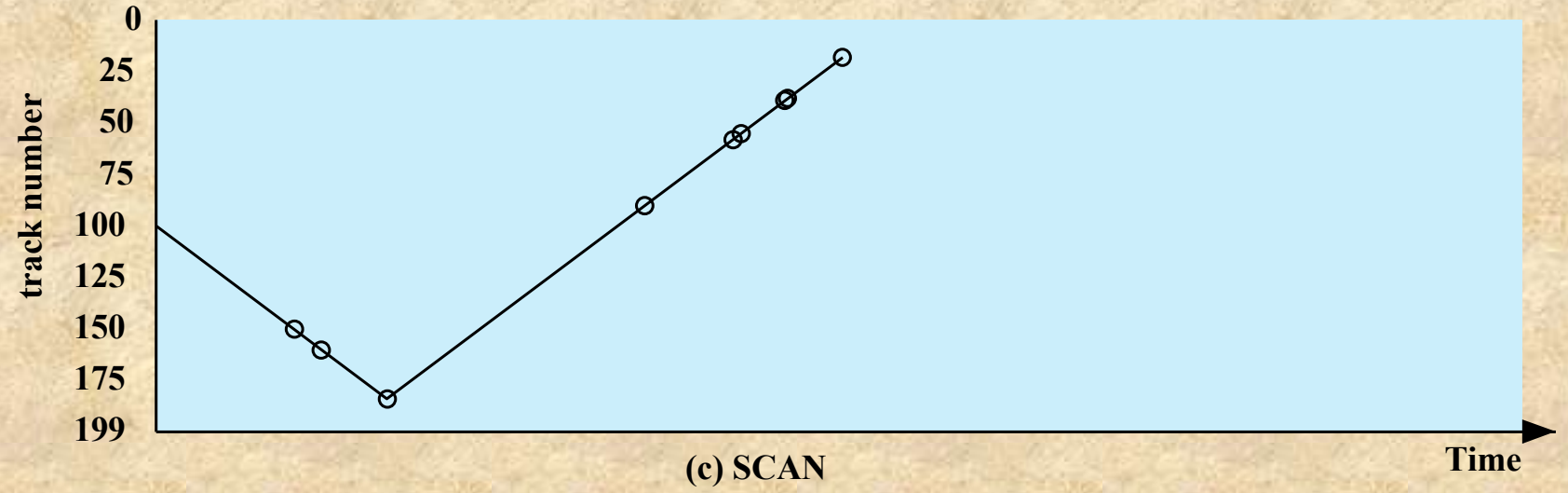
- Select the disk I/O request that requires the least movement of the disk arm from its current position
- Always choose the minimum seek time



(b) SSTF

SCAN

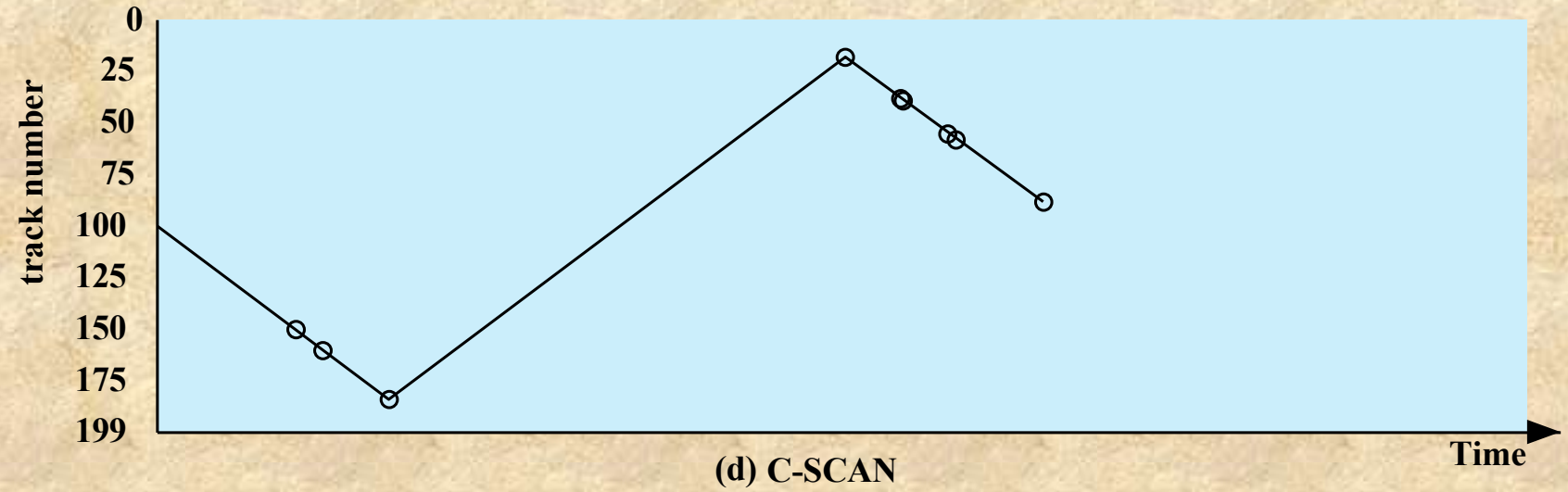
- Also known as the **elevator** algorithm
- Arm moves in one direction and satisfies all outstanding requests until it reaches the last track in that direction, then the direction is reversed
- **LOOK** refinement: reverse the direction after reaching the last requested track, as shown in the figure below
- Favors jobs whose requests are for tracks nearest to both innermost and outermost tracks and favors the latest-arriving jobs



(c) SCAN

C-SCAN (Circular SCAN)

- Restricts scanning to one direction only
- When the last requested track has been visited in one direction, the arm is returned to the opposite end of the disk and the scan begins again
- C-LOOK: similar to LOOK, as shown in the figure below



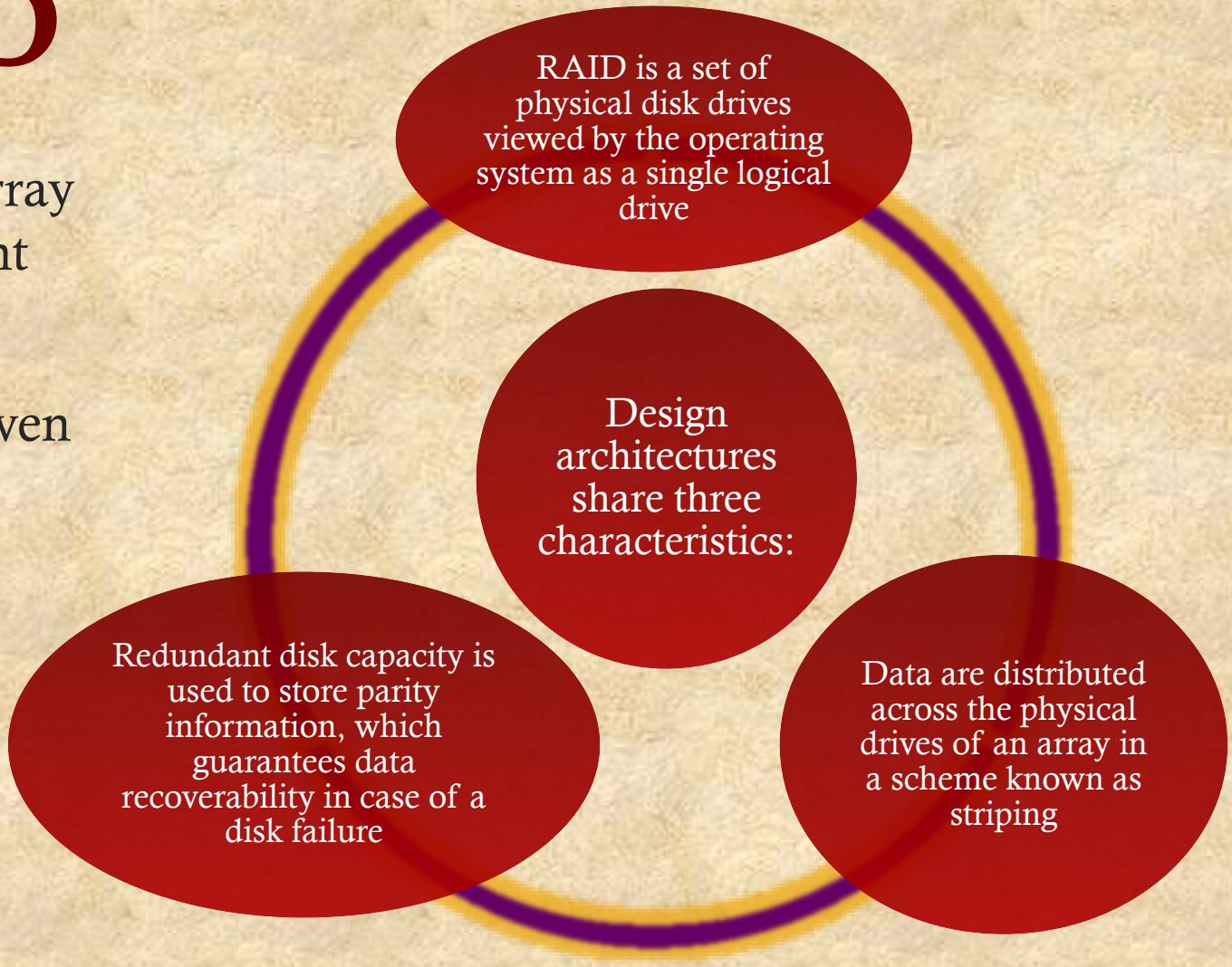
(d) C-SCAN

More Examples of Disk Scheduling

- http://www.cs.iit.edu/~cs561/cs450/disksched/dissched.html
- http://www2.cs.uregina.ca/~hamilton/courses/330/notes/io/node8.html
- For assignment, use LOOK for SCAN and C-LOOK for C-SCAN

RAID

- Redundant Array of Independent Disks
- Consists of seven levels, zero through six



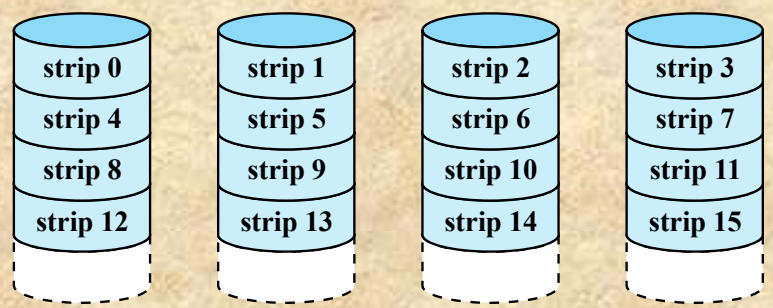
RAID

- The term was originally coined in a paper by a group of researchers at the University of California at Berkeley
  - The paper outlined various configurations and applications and introduced the definitions of the RAID levels
- Strategy employs multiple disk drives and distributes data in such a way as to enable simultaneous access to data from multiple drives
  - Improves I/O performance and allows easier incremental increases in capacity
- The unique contribution is to address effectively the need for redundancy
- Makes use of stored parity information that enables the recovery of data lost due to a disk failure

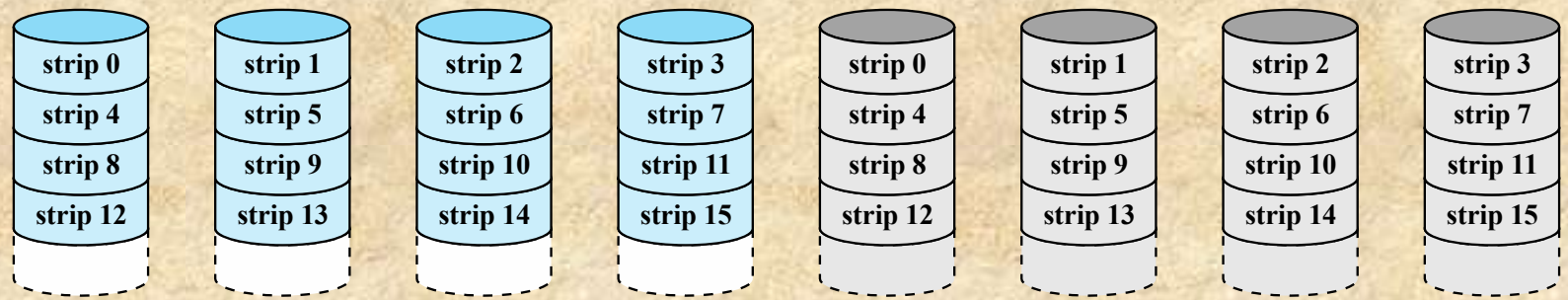
Category	Level	Description	Disks required	Data availability	Large I/O data transfer capacity	Small I/O request rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

N = number of data disks; m proportional to log N

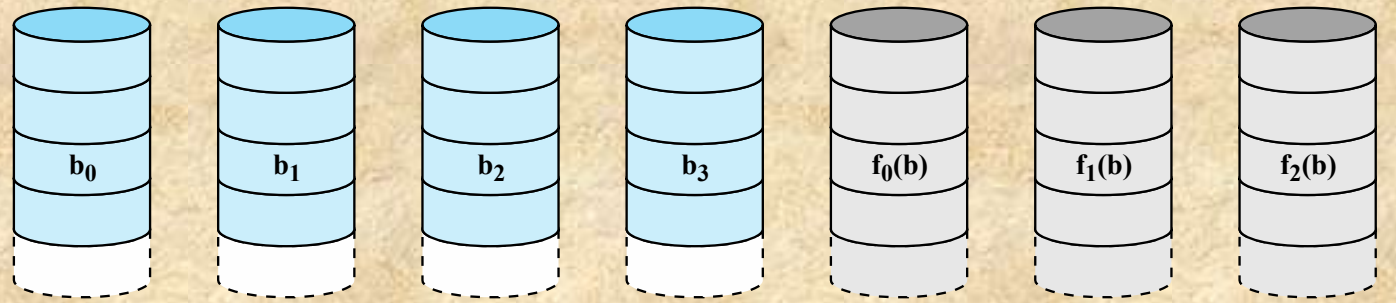
Table 11.4 RAID Levels (Page 498 in textbook)



(a) RAID 0 (non-redundant)

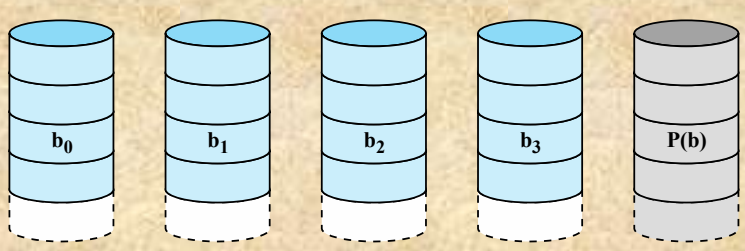


(b) RAID 1 (mirrored)

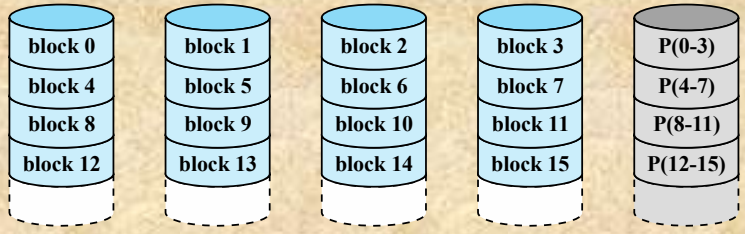


(c) RAID 2 (redundancy through Hamming code)

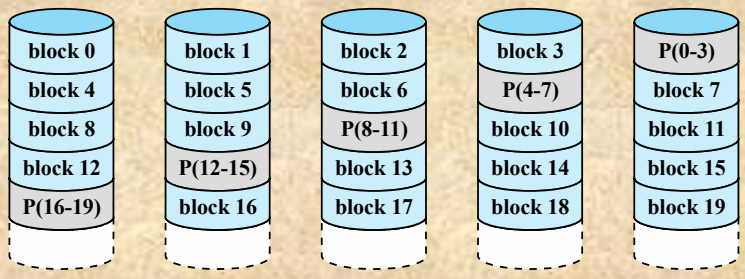
Figure 11.8 RAID Levels (page 1 of 2)



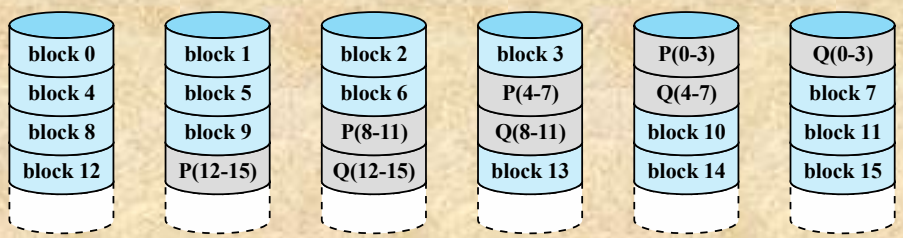
(d) RAID 3 (bit-interleaved parity)



(e) RAID 4 (block-level parity)



(f) RAID 5 (block-level distributed parity)



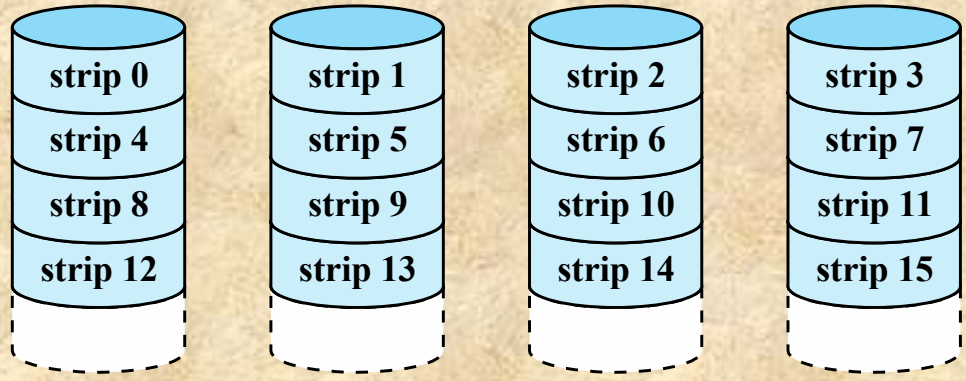
(g) RAID 6 (dual redundancy)

Figure 11.8 RAID Levels (page 2 of 2)

Of the seven RAID levels described, only four are commonly used: RAID levels 0, 1, 5, and 6

RAID Level 0

- Not a true RAID because it does not include redundancy to improve performance or provide data protection
- User and system data are distributed across all of the disks in the array
- Logical disk is divided into strips

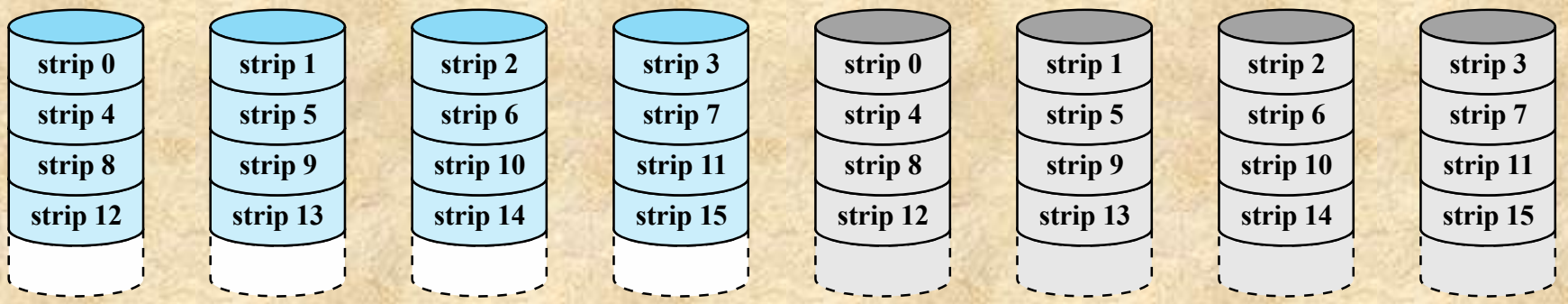


(a) RAID 0 (non-redundant)

In an *n*-disk array, the advantage of this layout is that if a single I/O request consists of multiple logically contiguous strips, then up to *n* strips for that request can be handled in parallel, greatly reducing the I/O transfer time

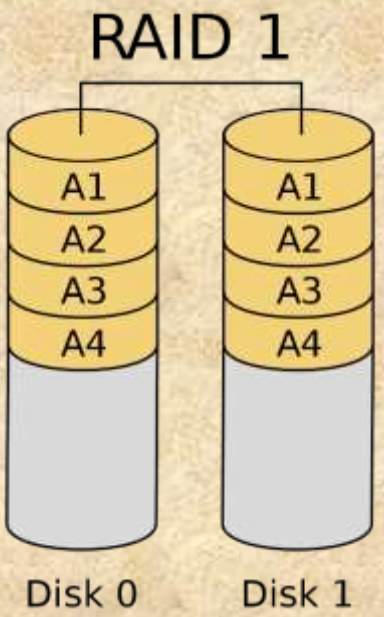
RAID Level 1

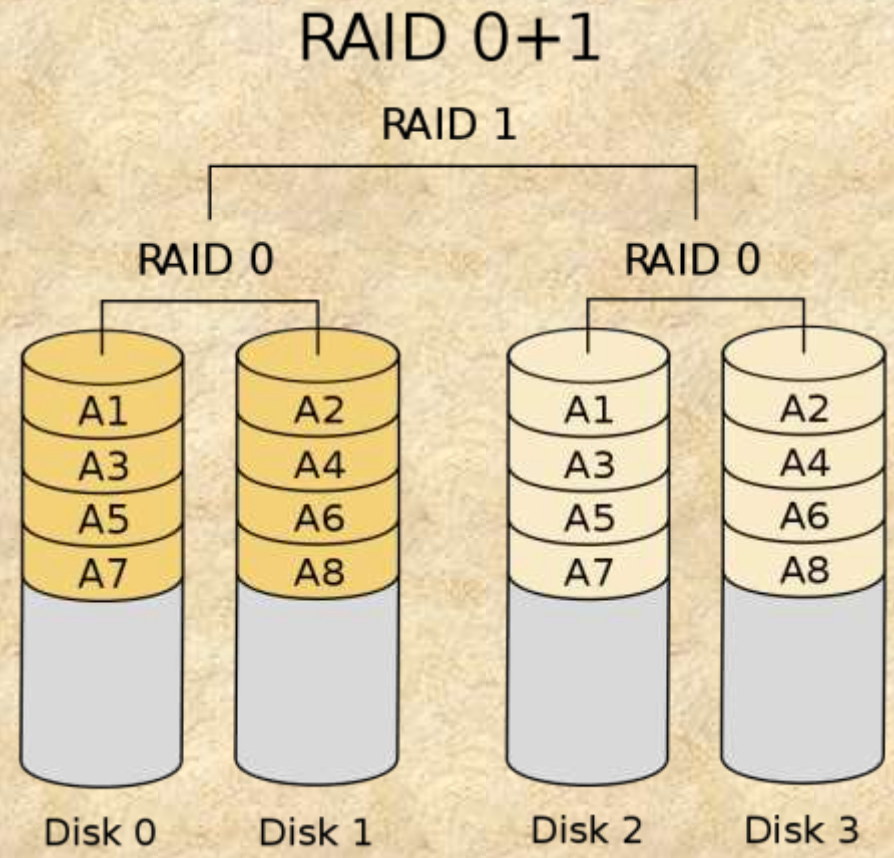
- Redundancy is achieved by the simple expedient of duplicating all the data
- There is no "write penalty" since it RAID 1 does not use parity bits
- When a drive fails the data may still be accessed from the second drive
- The principal disadvantage is the cost



(b) RAID 1 (mirrored)

RAID 0 + 1

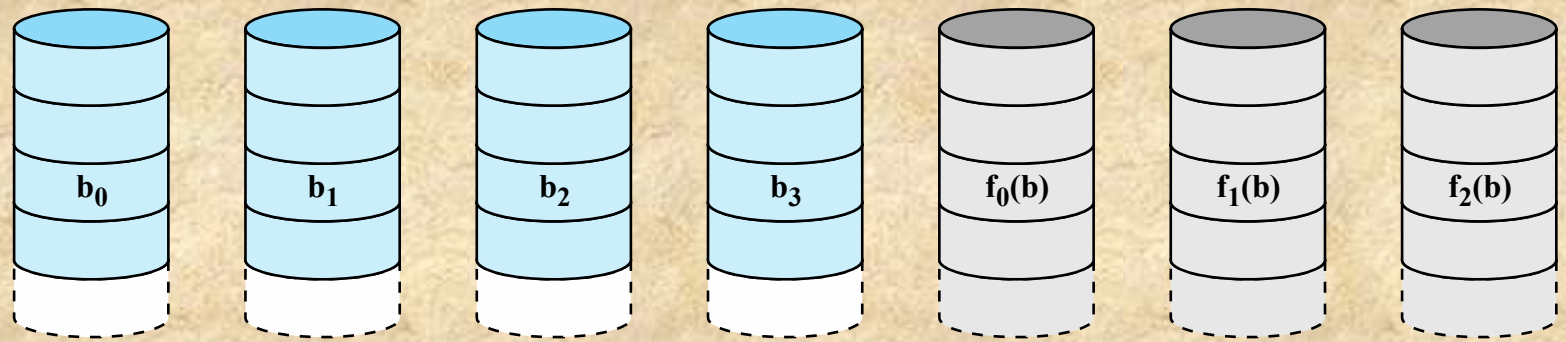




https://en.wikipedia.org/wiki/Standard_RAID_levels
https://en.wikipedia.org/wiki/Nested_RAID_levels

RAID Level 2

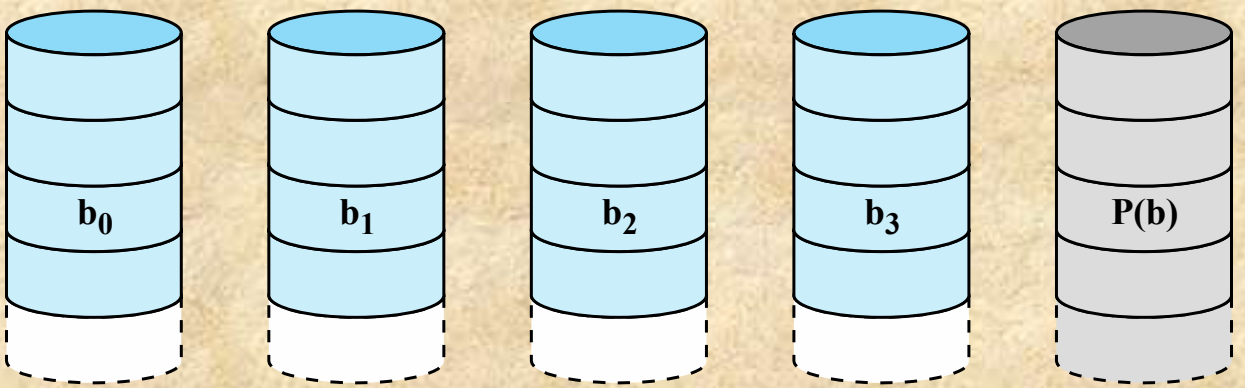
- Makes use of a parallel access technique. All member disks participate in the execution of every I/O request
- Data striping is used (strips are very small, e.g., a single byte or word)
- An error-correcting code is calculated across corresponding bits on each data disk, and the bits of the code are stored in the corresponding bit positions on multiple parity disks. Typically, a Hamming code is used, which can correct single-bit errors and detect double-bit errors
- Effective choice in an environment in which many disk errors occur



(c) RAID 2 (redundancy through Hamming code)

RAID Level 3

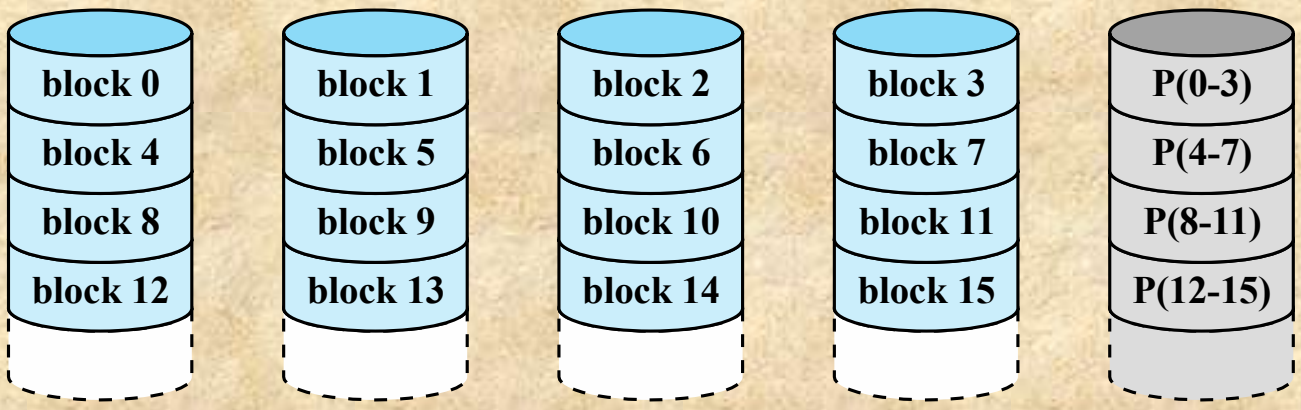
- Requires only a single redundant disk, no matter how large the disk array. Employs parallel access, with data distributed in small strips (e.g., bytes)
- Instead of an error correcting code, a simple parity bit is computed for the set of individual bits in the same position on all data disks
- Can achieve very high data transfer rates
- In the event of a drive failure, the parity drive is accessed, and data is reconstructed from the remaining devices. Once the failed drive is replaced, the missing data can be restored on the new drive and operation resumed.



(d) RAID 3 (bit-interleaved parity)

RAID Level 4

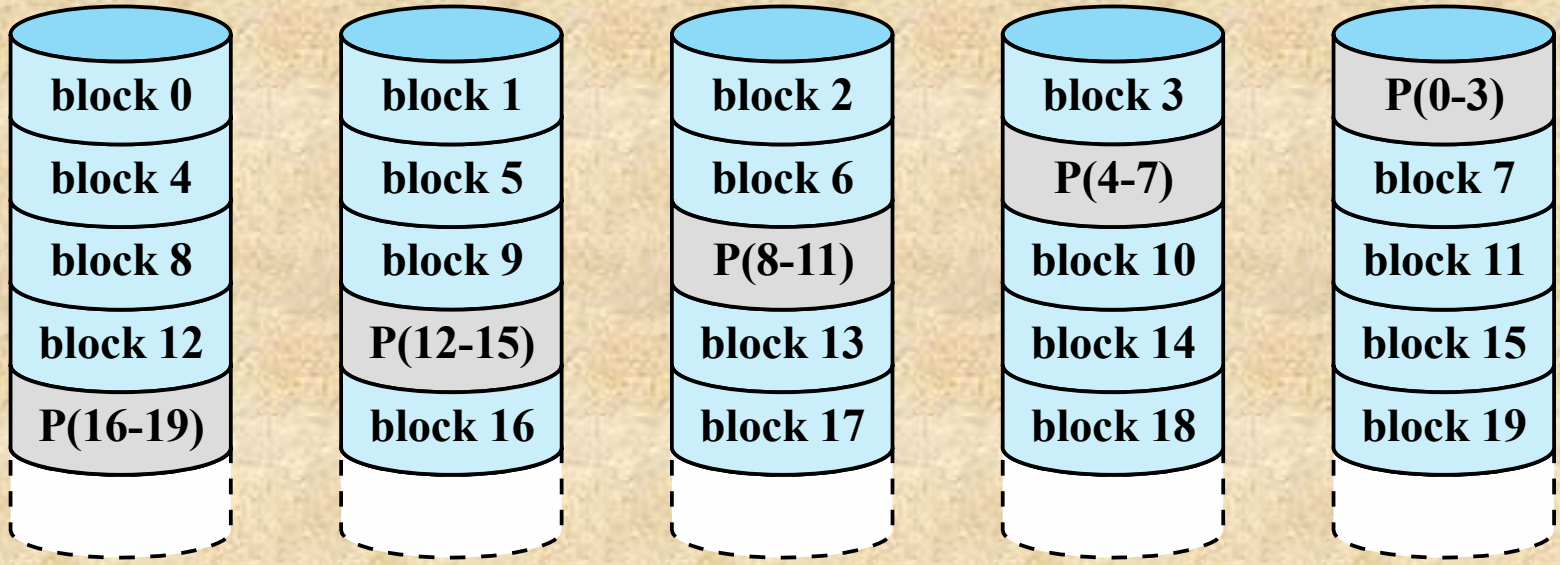
- Makes use of an independent access technique
  - Each member disk operates independently
  - More suitable for applications that require high I/O request rates and are relatively less suitable for applications that require high data transfer rates
- A bit-by-bit parity strip is calculated across corresponding strips on each data disk, and parity bits are stored in the corresponding strip on the parity disk
- Every write operation must involve the parity disk, which therefore can become a bottleneck



(e) RAID 4 (block-level parity)

RAID Level 5

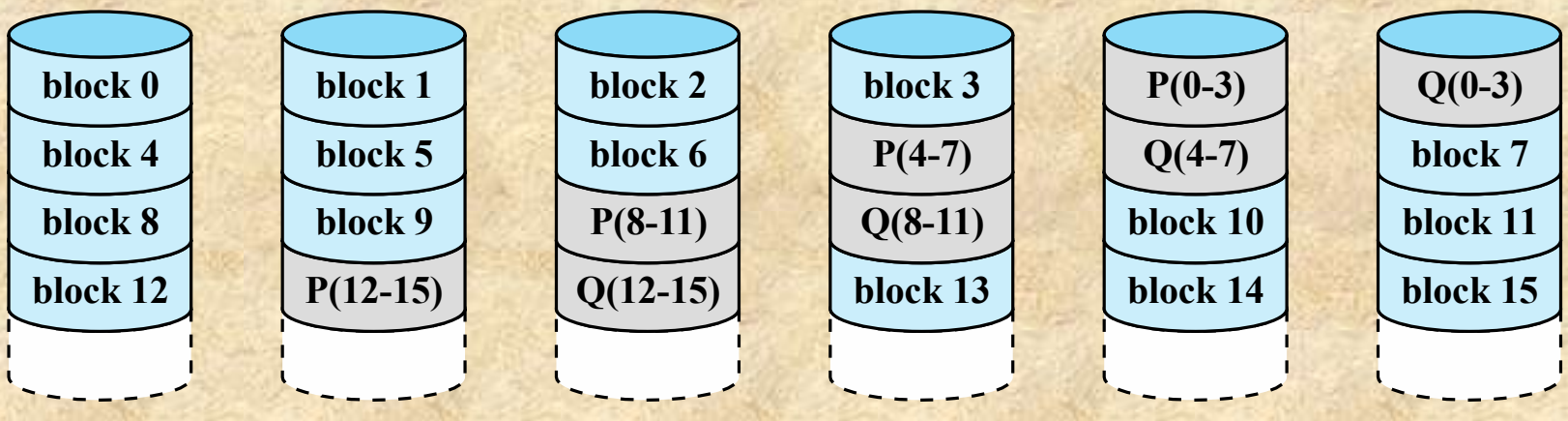
- Similar to RAID-4 but distributes the parity bits across all disks
- Typical allocation is a round-robin scheme
- The distribution of parity strips across all drives avoids the potential I/O bottleneck of the single parity disk found in RAID 4



(f) RAID 5 (block-level distributed parity)

RAID Level 6

- Two different parity calculations are carried out and stored in separate blocks on different disks
- Provides extremely high data availability
- Incurs a substantial write penalty because each write affects two parity blocks



(g) RAID 6 (dual redundancy)

Final Exam
CSIS 2260-010/011/012

- Up to one hour, from 8:30 AM, Apr 15, 2021 (Pacific Time)

https://www.douglascollege.ca/current-students/important-dates-information/exam-schedule

- Blackboard online test: MCQ, true/false, filling in the blank, and short questions
- Don't be late, otherwise you may not be allowed to finish/attend
- You must have ZOOM installed and a working camera. No camera, no exam and a ZERO mark. You will receive the ZOOM link via Email and Blackboard before the exam
- Coverage: all lectures and labs covered in the whole semester; you may expect more questions from the content after the midterm exam. Check the slides for coverage and the corresponding textbook for details
- 30% toward the total evaluation
- Don't forget the assignment

Thank you!