Operating System

Nguyen Tri Thanh
ntthanh@vnu.edu.vn

Storage Systems

File-System Structure
File-System Implementation
Directory Implementation
Allocation Methods
Free-Space Management
Efficiency and Performance
Recovery

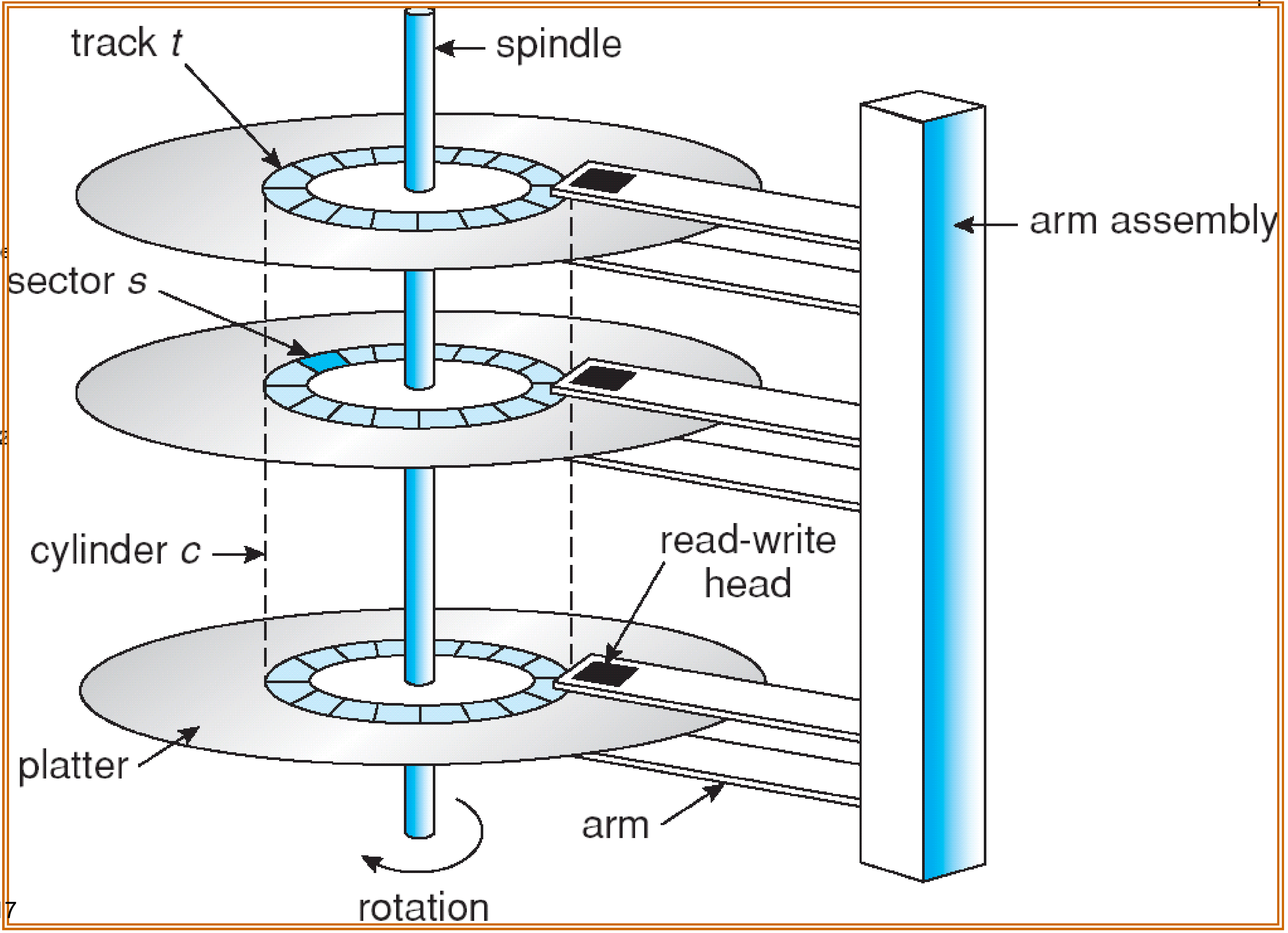
Objectives

- Introduce the list of mass storage devices
- Introduce the structure/organization of disks
- Introduce disk scheduling algorithms
- Introduce reliable storages
- Introduce non-violate storages
- Implement disk scheduling algorithms

Reference

- Chapter 12 of Operating System Concepts

Moving-head Disk Mechanism



(entire

(512

Overview of Mass Storage Structure

- Magnetic disks provide bulk of secondary storage of modern computers
  - rotate at 60 to 300 rounds per second
  - **Transfer rate**
    - rate of data flow between drive and computer
  - **Positioning time** (**random-access time**)
    - time to move disk arm to desired cylinder (**seek time**) and
    - time for desired sector to rotate under the disk head (**rotational latency**)
  - **Head crash**
    - disk head making contact with the disk surface
    - That's bad

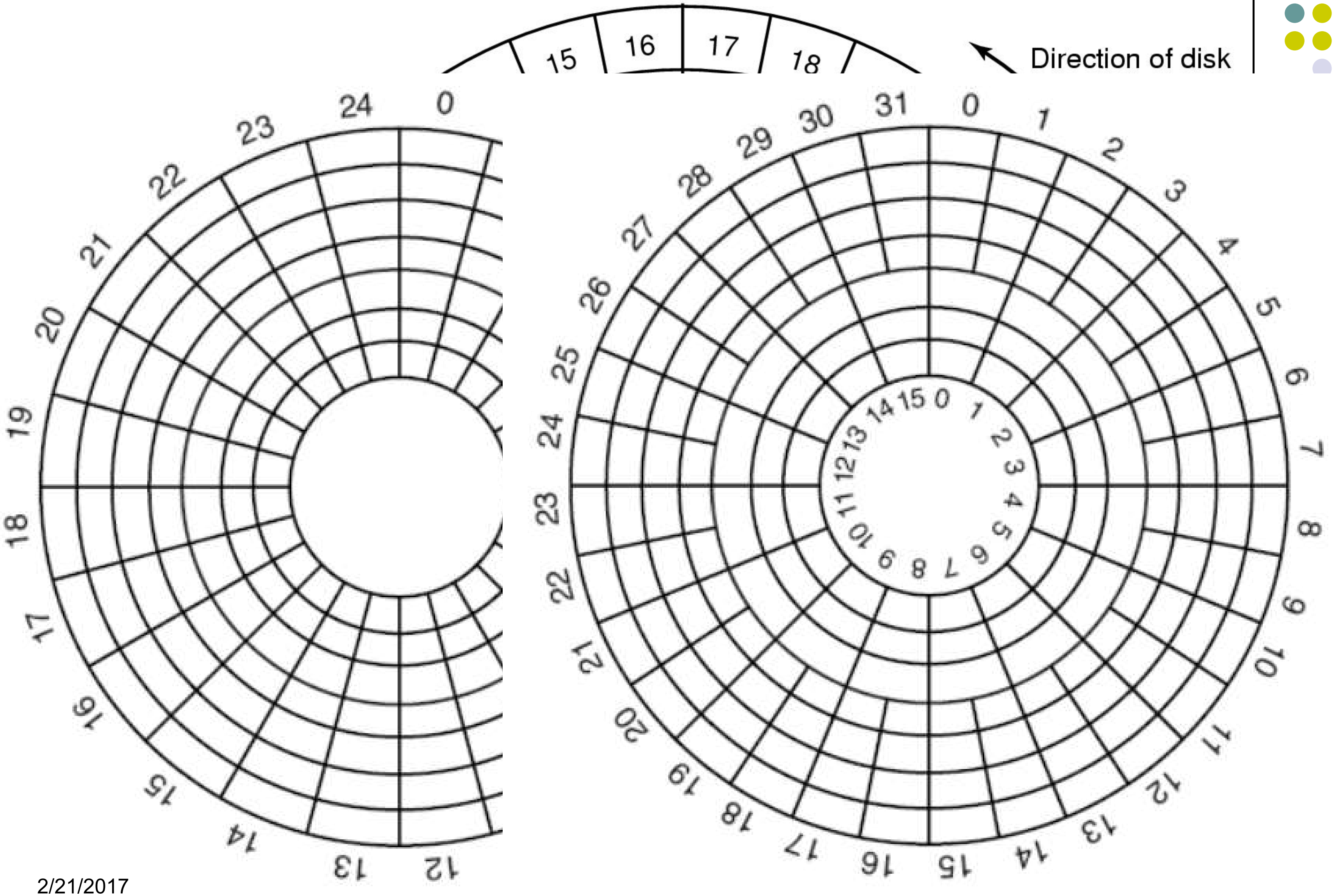
Overview of Mass Storage Structure (cont'd)

- Disks can be removable
- Drive attached to computer via **I/O bus**
  - EIDE, ATA, SATA, USB, Fibre Channel, SCSI
  - Host controller
    - computer uses bus to talk to
  - Disk controller
    - built into drive or storage array

Disk Structure

- Disk drives are treated as
  - a large 1-dimensional arrays of *logical blocks*
    - a logical block is the smallest unit of transfer
  - array of logical blocks is mapped into the sectors of the disk sequentially.
    - Sector 0 is the first sector of the first track on the outermost cylinder
    - Mapping proceeds in order through that track
    - then the rest of the tracks in that cylinder,
    - and then through the rest of the cylinders from outermost to innermost.

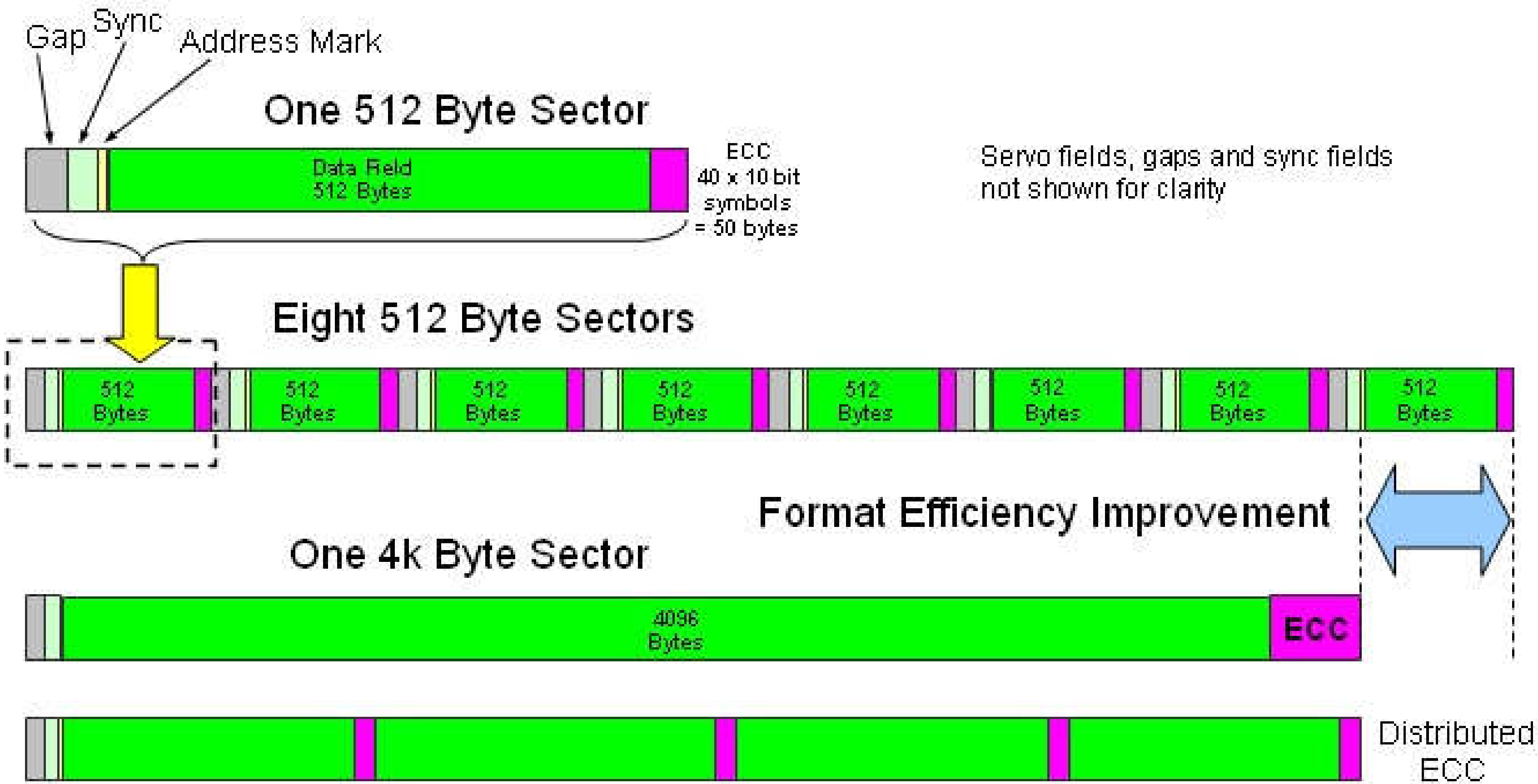
Sectors



Question

- Which is the reason why the sector numbers of different cylinders are not the same?
  A. to increase security
  B. to increase disk size
  C. to increase transfer rate
  D. to reduce waiting time

Sectors



A number of sectors in each cylinder is not numbered (unused)

Question

- Which is the reason why a number of sectors in each cylinder is unused?
  A. to increase security
  B. to be used to recover bad sectors
  C. to be used as buffer
  D. to be used by operating system for logic formatting

Disk Scheduling

Disk Scheduling

- The operating system is responsible for using hardware efficiently
  - for the disk drives, this means having a fast access time and disk bandwidth
- Access time has two major components
  - *Seek time*
    - the time for the disk are to move the heads to the cylinder containing the desired sector.
  - *Rotational latency*
    - the additional time waiting for the disk to rotate the desired sector to the disk head.

Disk Scheduling (Cont.)

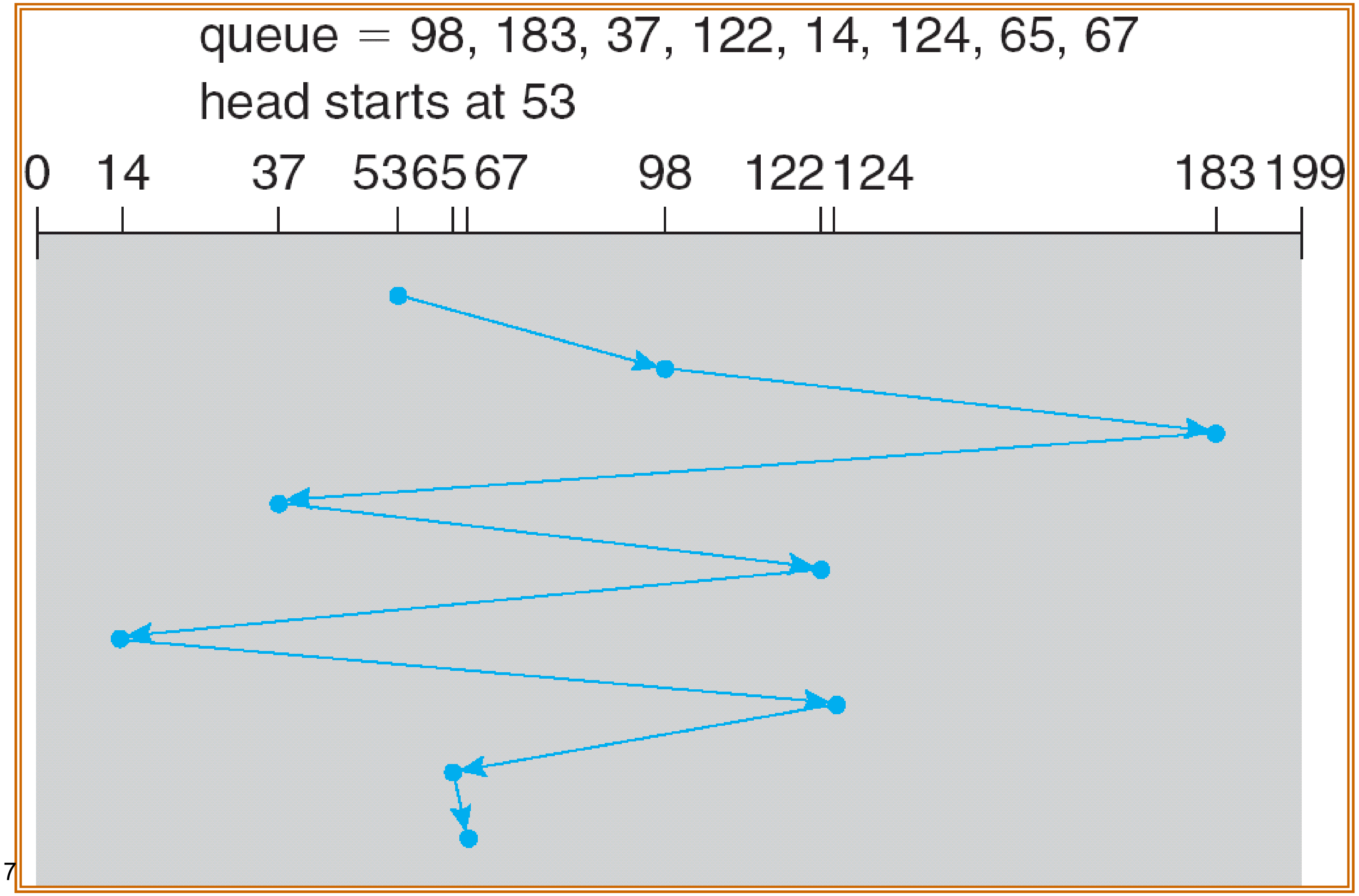
- Target
  - Minimize seek time
  - Seek time ≈ seek distance
- Disk bandwidth
  - (total number of bytes transferred) / (total time between the first request for service and the completion of the last transfer)

Disk Scheduling Algorithms

- Several algorithms exist to schedule the servicing of disk I/O requests
- We illustrate them with a request queue (0-199)
  - 98, 183, 37, 122, 14, 124, 65, 67
  - Current head pointer 53

FCFS

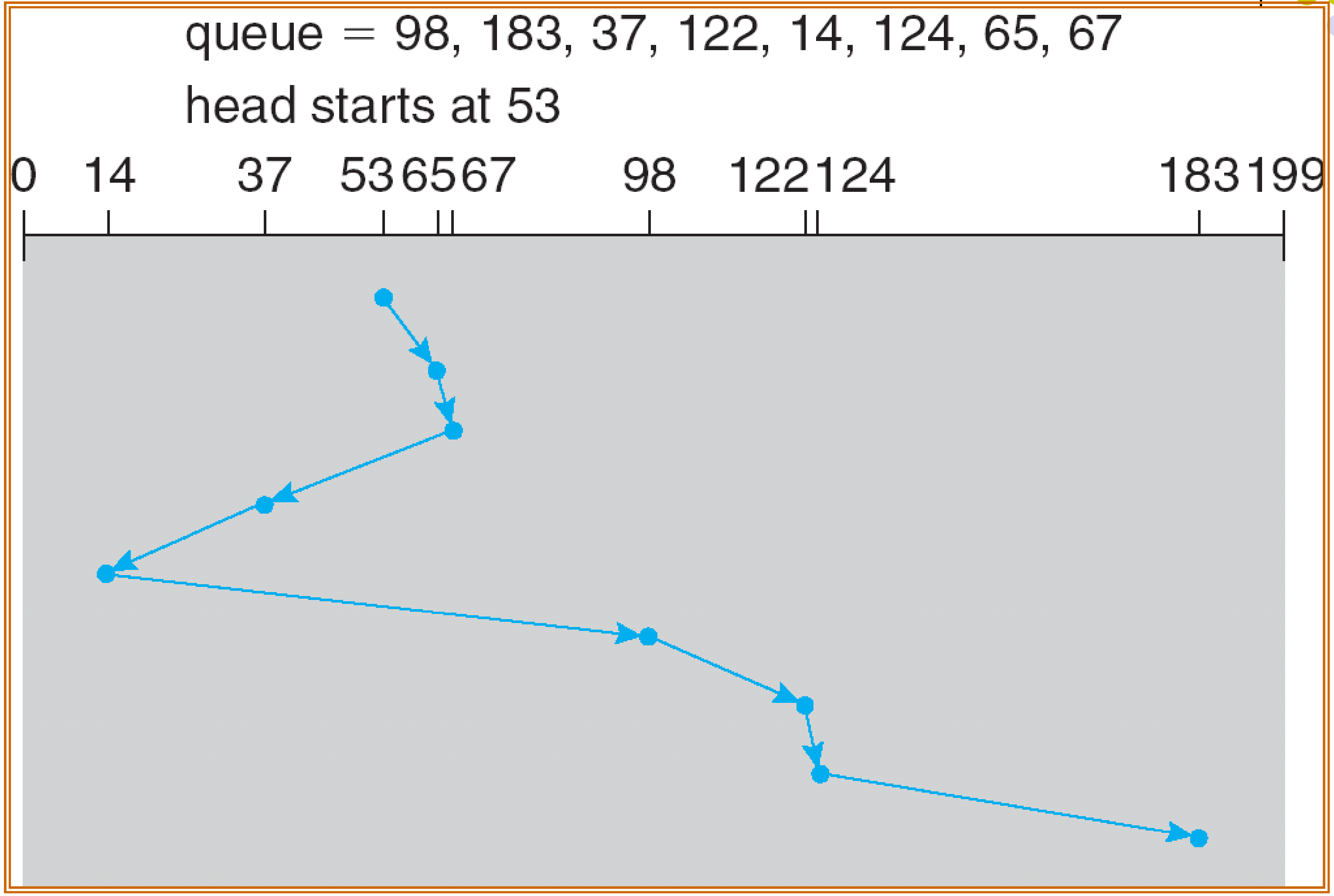
Illustration shows total head movement of 640 cylinders.



Shortest Seek Time First (SSTF)

- Selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of SJF scheduling;
  - may cause starvation of some requests
- Illustration shows total head movement of 236 cylinders.

SSTF (Cont.)

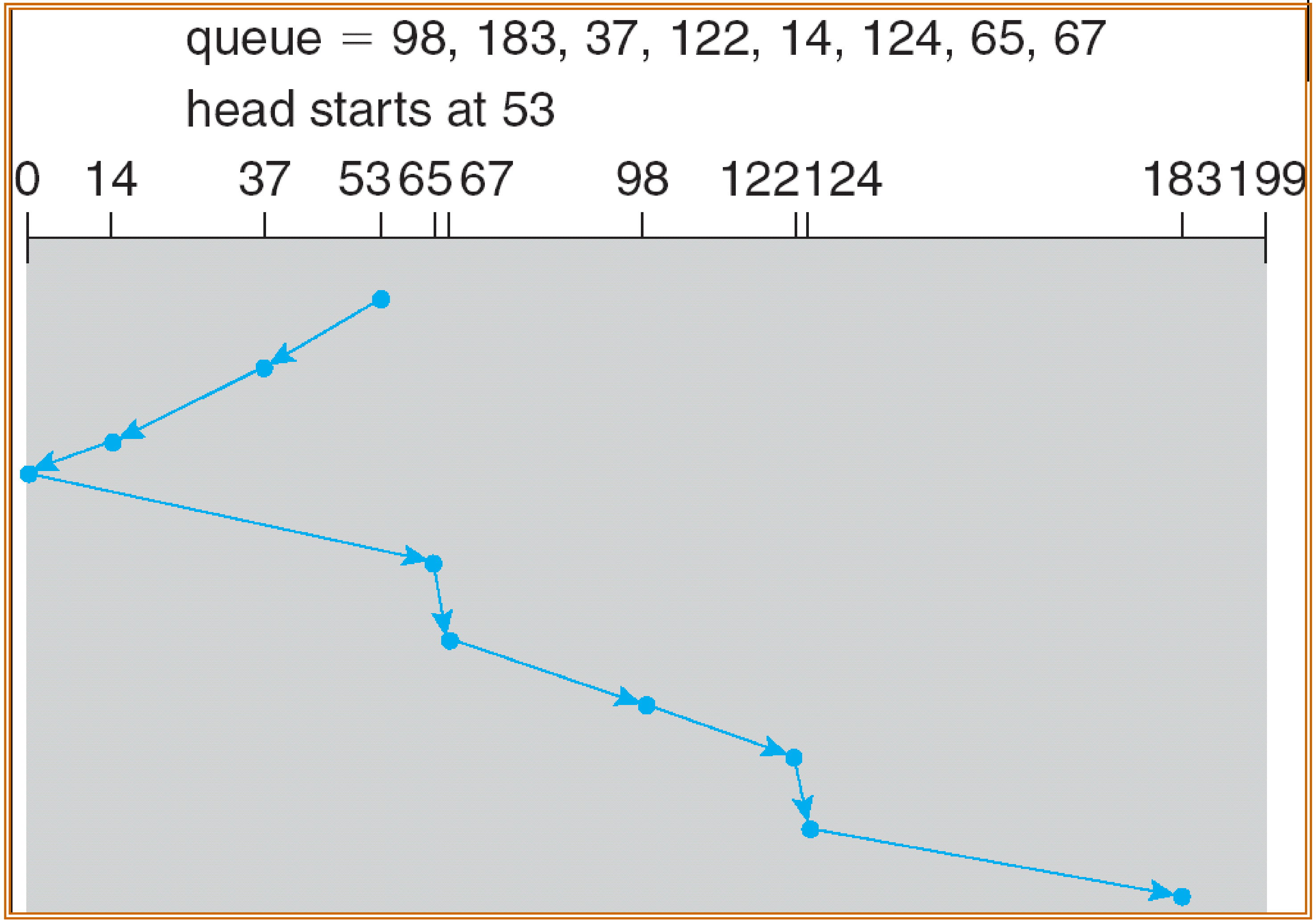


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0 14 37 53 65 67 98 122 124 183 199

SCAN

- The disk arm starts at one end of the disk, and moves toward the other end,
  - servicing requests until it gets to the other end of the disk,
  - head movement is reversed and servicing continues.
- Sometimes called the *elevator algorithm*
- Illustration shows total head movement of 236 cylinders

SCAN (Cont.)

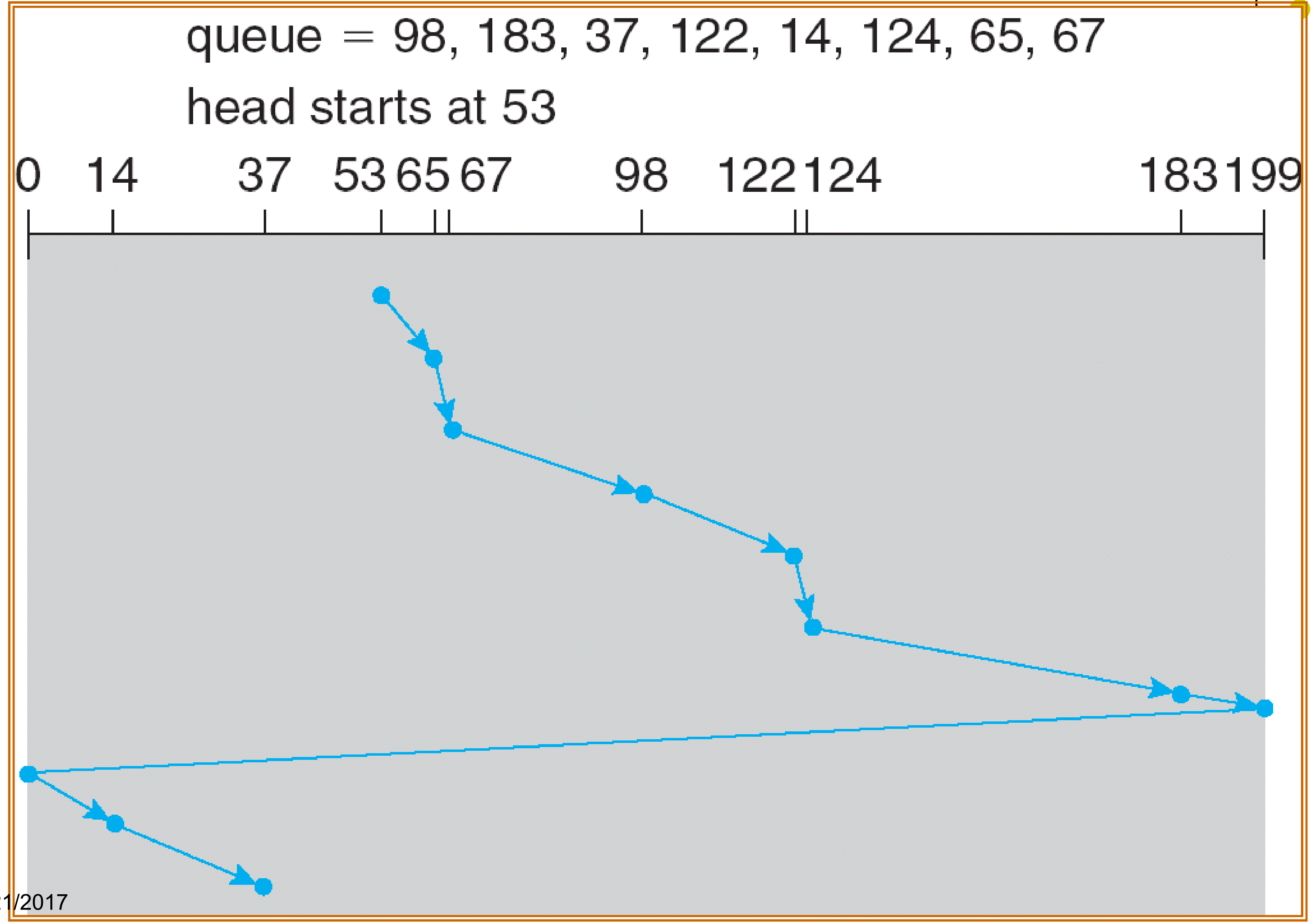


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0 14 37 53 65 67 98 122 124 183 199

C-SCAN

- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other
  - servicing requests as it goes
  - When it reaches the other end, however,
    - it immediately returns to the beginning of the disk,
    - without servicing any requests on the return trip.
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one

C-SCAN (Cont.)



queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0 14 37 53 65 67 98 122 124 183 199

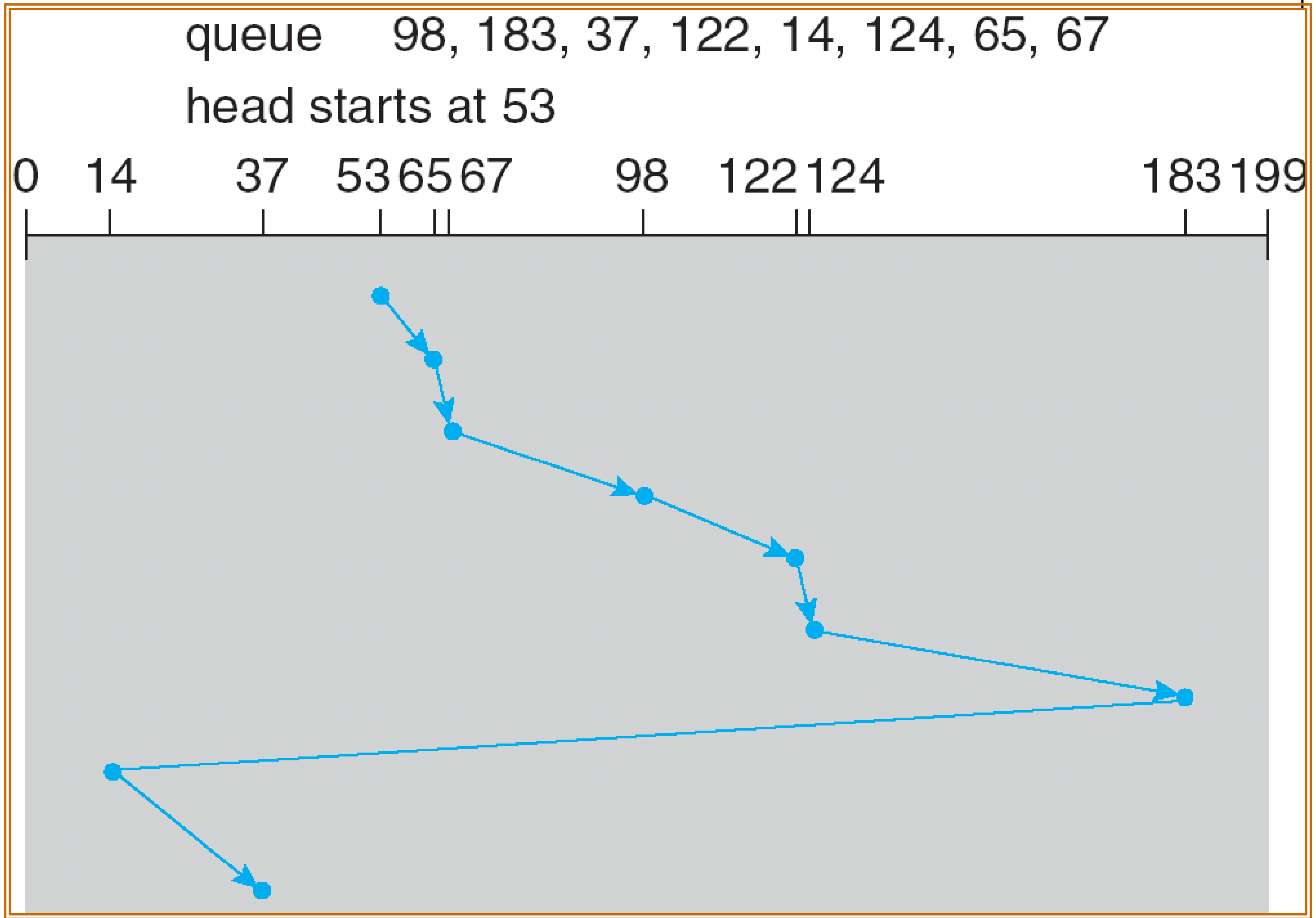
LOOK

- Version of SCAN
- Arm only goes as far as the last request in each direction,

C-LOOK

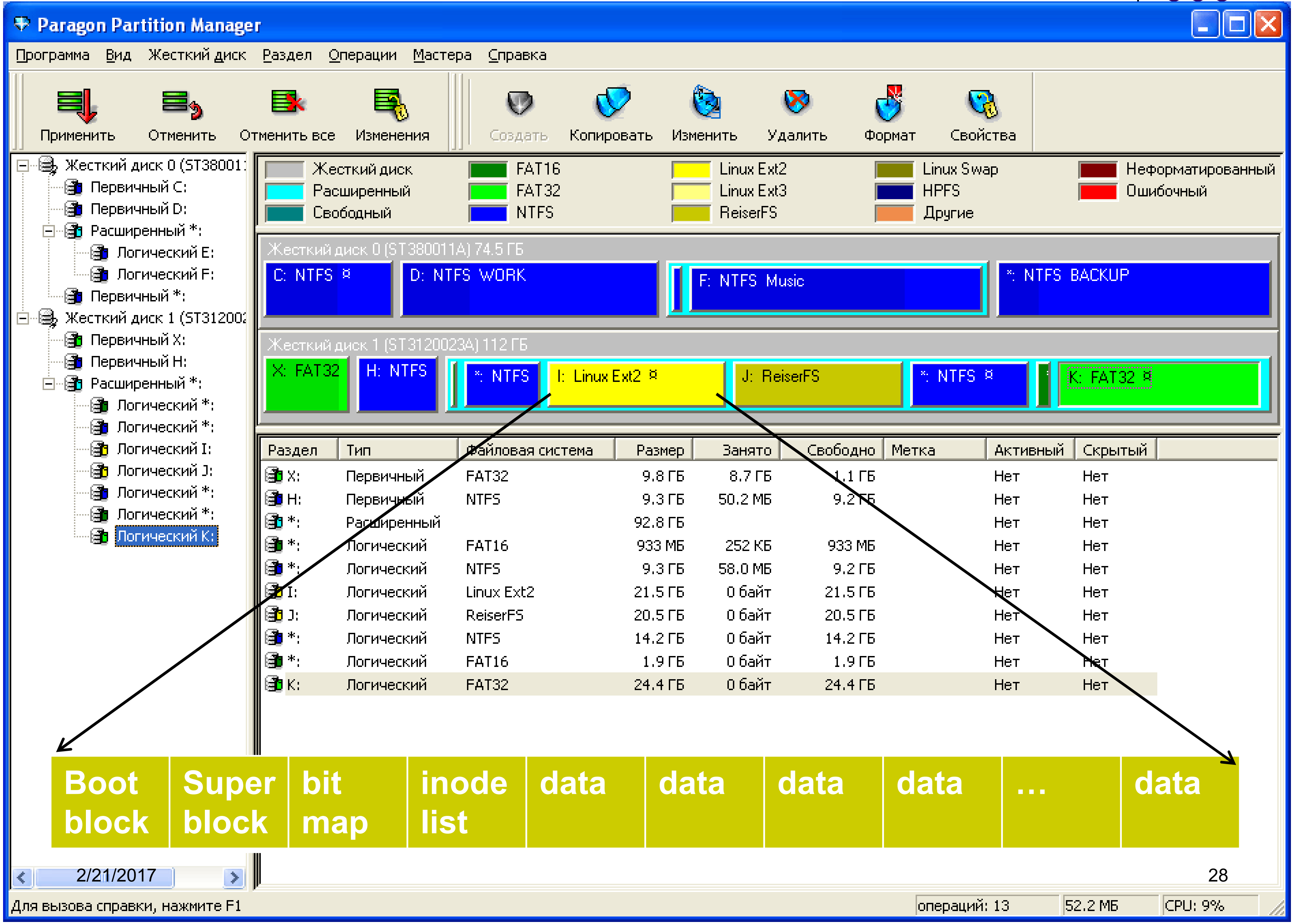
- Version of C-SCAN
- Arm only goes as far as the last request in each direction,
  - then reverses direction immediately,
  - without first going all the way to the end of the disk.

C-LOOK (Cont.)


queue 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0 14 37 53 65 67 98 122 124 183 199

Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk.
- Performance depends on the number and types of requests.
- Requests for disk service can be influenced by the file-allocation method.
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary.
- Either SSTF or LOOK is a reasonable choice for the default algorithm.


Paragon Partition Manager
Программа Вид Жесткий диск Раздел Операции Мастера Справка
Применить Отменить Отменить все Изменения Создать Копировать Изменить Удалить Формат Свойства
Жесткий диск 0 (ST38001
Первичный C:
Первичный D:
Расширенный *:
Логический E:
Логический F:
Первичный *:
Жесткий диск 1 (ST31200
Первичный X:
Первичный H:
Расширенный *:
Логический *:
Логический *:
Логический I:
Логический J:
Логический *:
Логический *:
Логический K:
Жесткий диск Расширенный Свободный FAT16 FAT32 NTFS Linux Ext2 Linux Ext3 ReiserFS Linux Swap HPFS Другие Неформатированный Ошибочный
Жесткий диск 0 (ST380011A) 74.5 ГБ
C: NTFS
D: NTFS WORK
F: NTFS Music
*: NTFS BACKUP
Жесткий диск 1 (ST3120023A) 112 ГБ
X: FAT32
H: NTFS
*: NTFS
I: Linux Ext2
J: ReiserFS
*: NTFS
K: FAT32
Раздел Тип Файловая система Размер Занято Свободно Метка Активный Скрытый
X: Первичный FAT32 9.8 ГБ 8.7 ГБ 1.1 ГБ Нет Нет
H: Первичный NTFS 9.3 ГБ 50.2 МБ 9.2 ГБ Нет Нет
*: Расширенный 92.8 ГБ Нет Нет
*: Логический FAT16 933 МБ 252 КБ 933 МБ Нет Нет
*: Логический NTFS 9.3 ГБ 58.0 МБ 9.2 ГБ Нет Нет
I: Логический Linux Ext2 21.5 ГБ 0 байт 21.5 ГБ Нет Нет
J: Логический ReiserFS 20.5 ГБ 0 байт 20.5 ГБ Нет Нет
*: Логический NTFS 14.2 ГБ 0 байт 14.2 ГБ Нет Нет
*: Логический FAT16 1.9 ГБ 0 байт 1.9 ГБ Нет Нет
K: Логический FAT32 24.4 ГБ 0 байт 24.4 ГБ Нет Нет
Boot block
Super block
bit map
inode list
data
data
data
data
…
data
Для вызова справки, нажмите F1
операций: 13
52.2 МБ
CPU: 9%

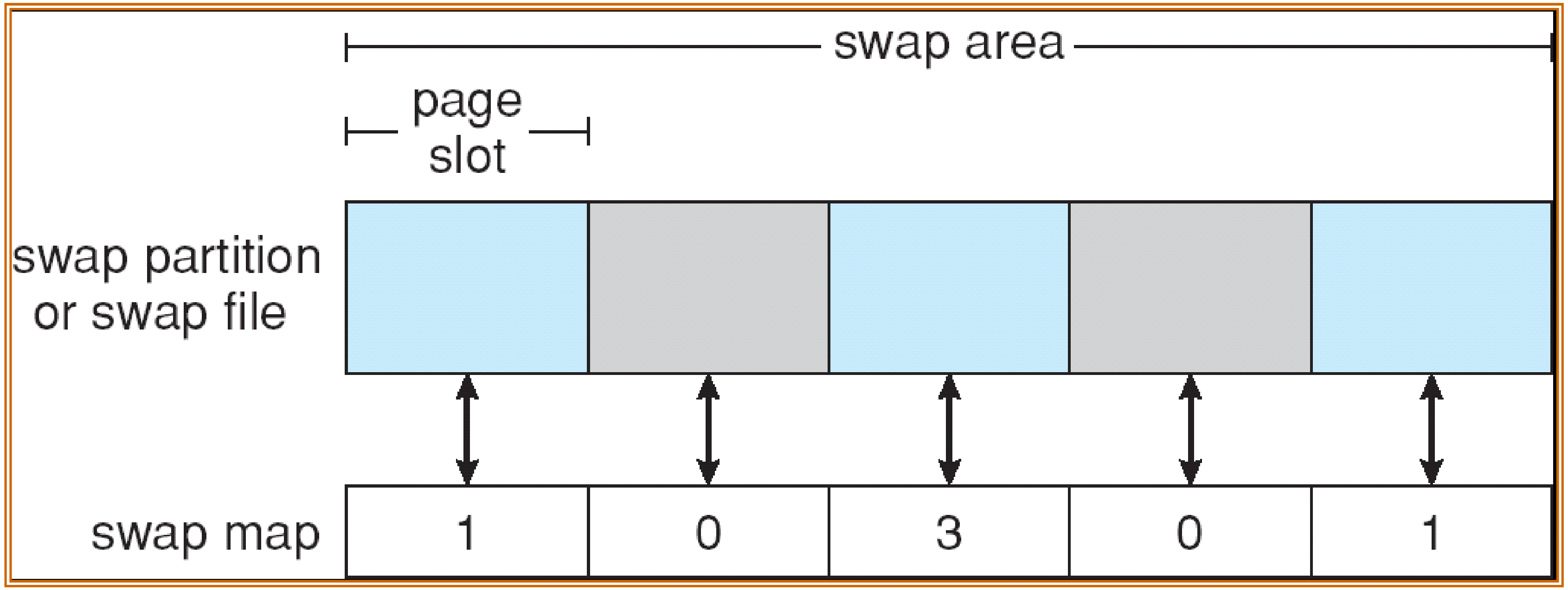
Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory.
- Swap-space can be
  - carved out of the normal file system,
  - more commonly, it can be in a separate disk partition.

Swap-Space Management

- Swap-space management
  - 4.3BSD allocates swap space when process starts; holds *text segment* (the program) and *data segment.*
  - Kernel uses *swap maps* to track swap-space use.
  - Solaris 2 allocates swap space only when a page is forced out of physical memory, not when the virtual memory page is first created.

Data Structures for Swapping on Linux Systems



Reliable storage

(reliable means data is safe even some disks are broken)

RAID Structure

- **RAID=Redundant Array of Inexpensive Disks**
- **RAID** – multiple disk drives provides **reliability** via **redundancy**

- RAID is arranged into six different levels
- There are also combinations

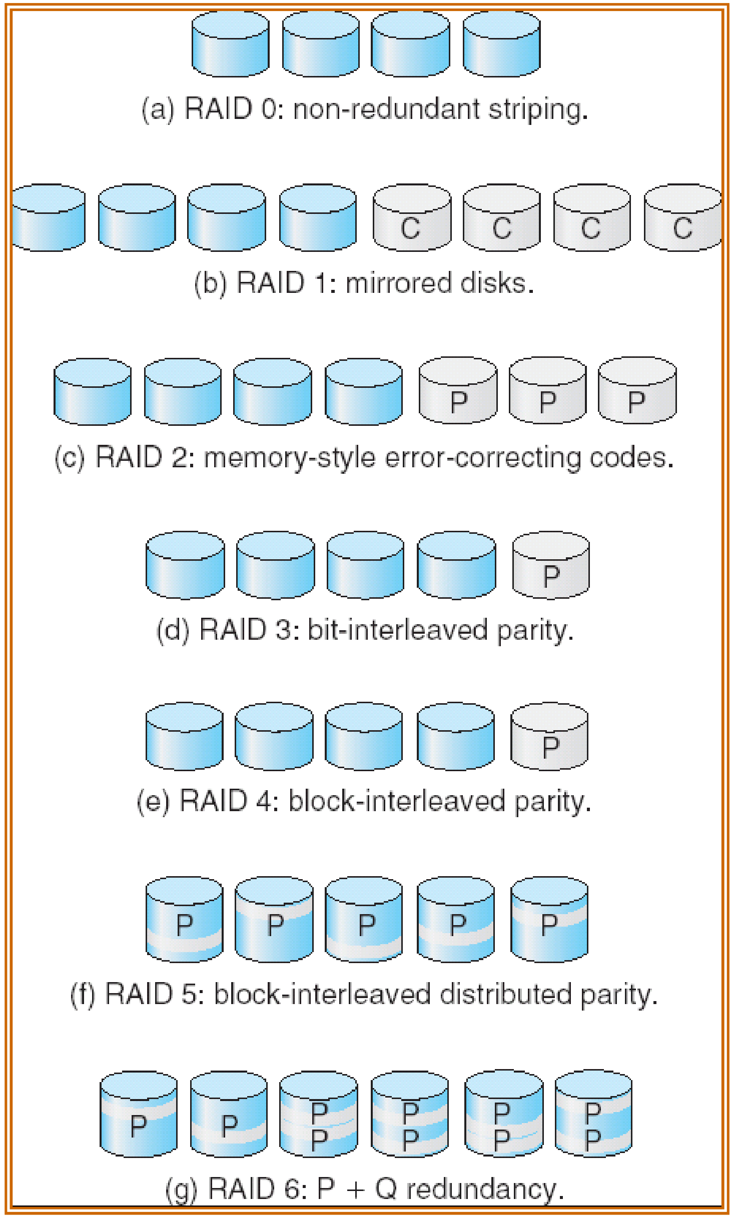
RAID (cont'd)

- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively
- Disk striping uses a group of disks as one storage unit

RAID (cont'd)

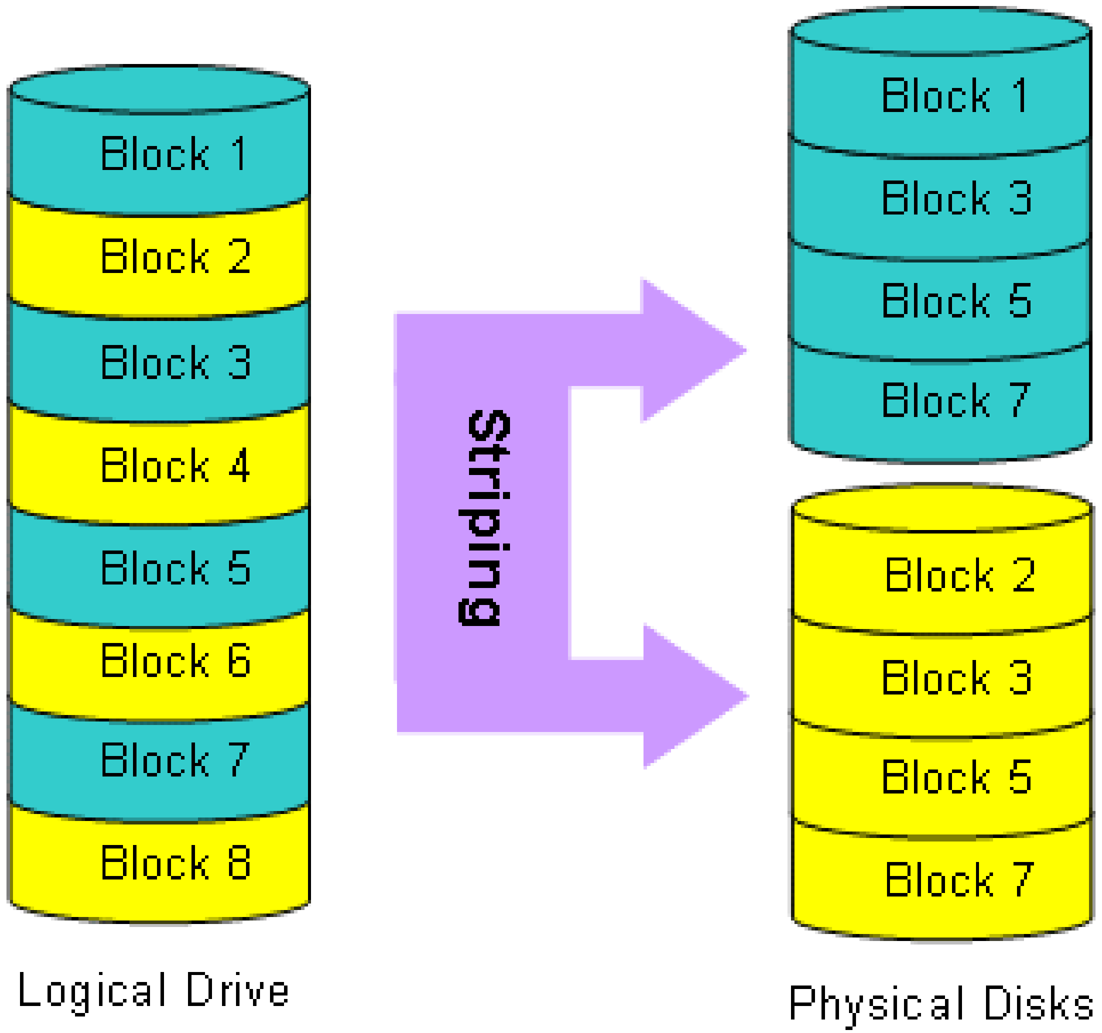
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - *Mirroring* or *shadowing* keeps duplicate of each disk
  - *Block interleaved parity* uses much less redundancy

RAID Levels

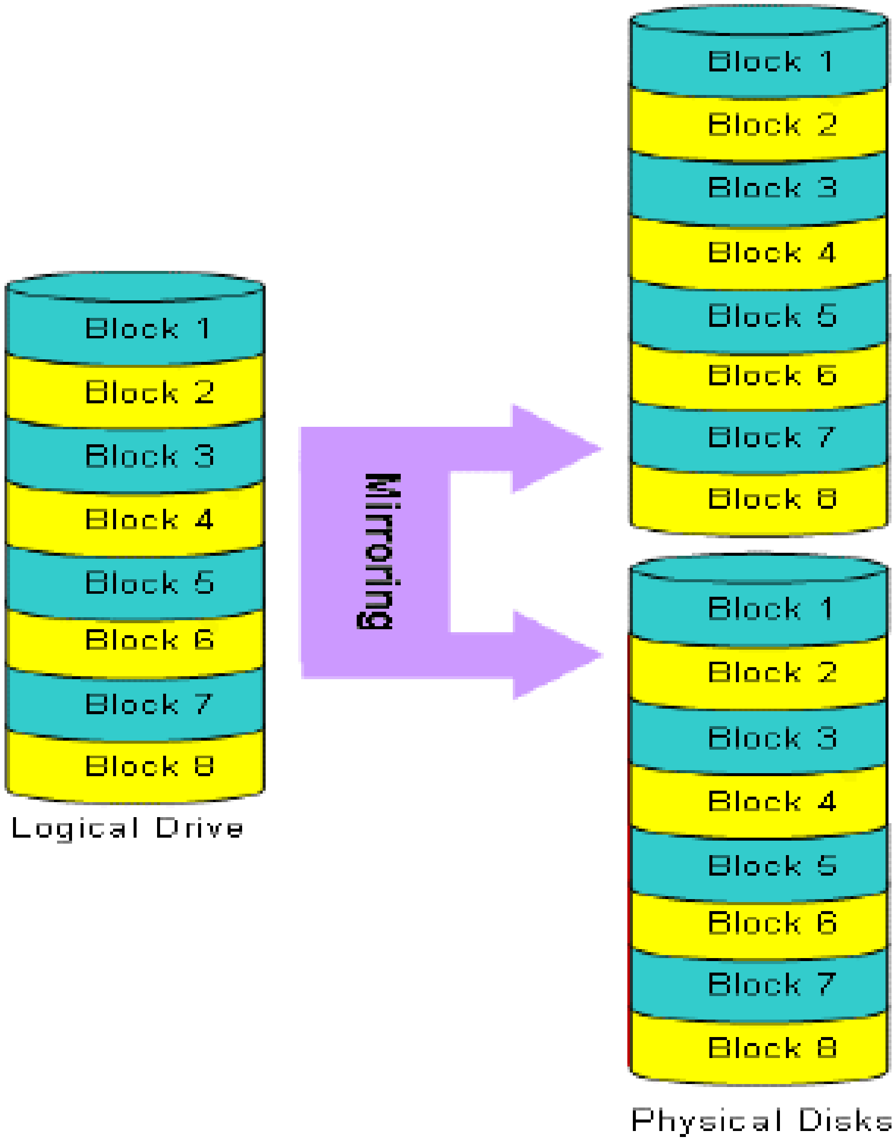


(a) RAID 0: non-redundant striping.
C C C C
(b) RAID 1: mirrored disks.
P P P
(c) RAID 2: memory-style error-correcting codes.
P
(d) RAID 3: bit-interleaved parity.
P
(e) RAID 4: block-interleaved parity.
P P P P P
(f) RAID 5: block-interleaved distributed parity.
P P P P P P P P P P
(g) RAID 6: P + Q redundancy.

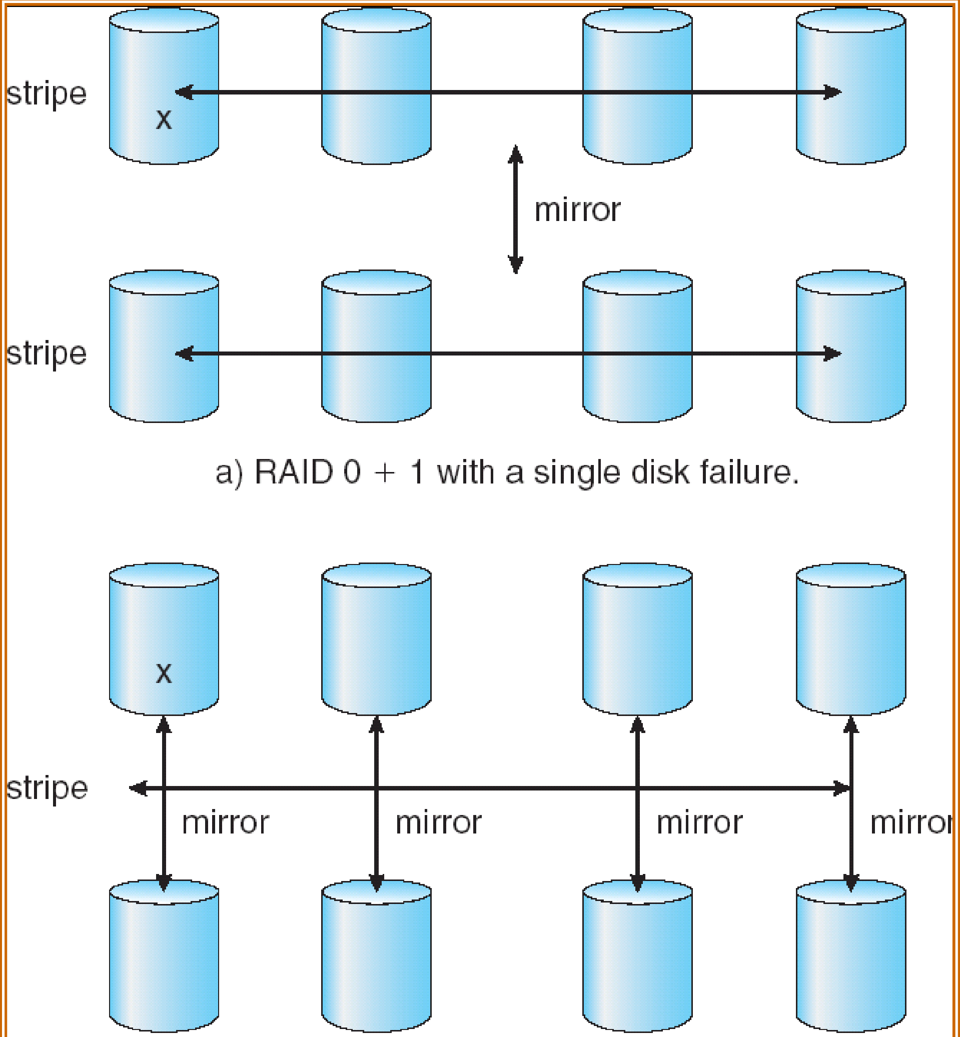
RAID 0 - Stripping



RAID 1 -Mirroring



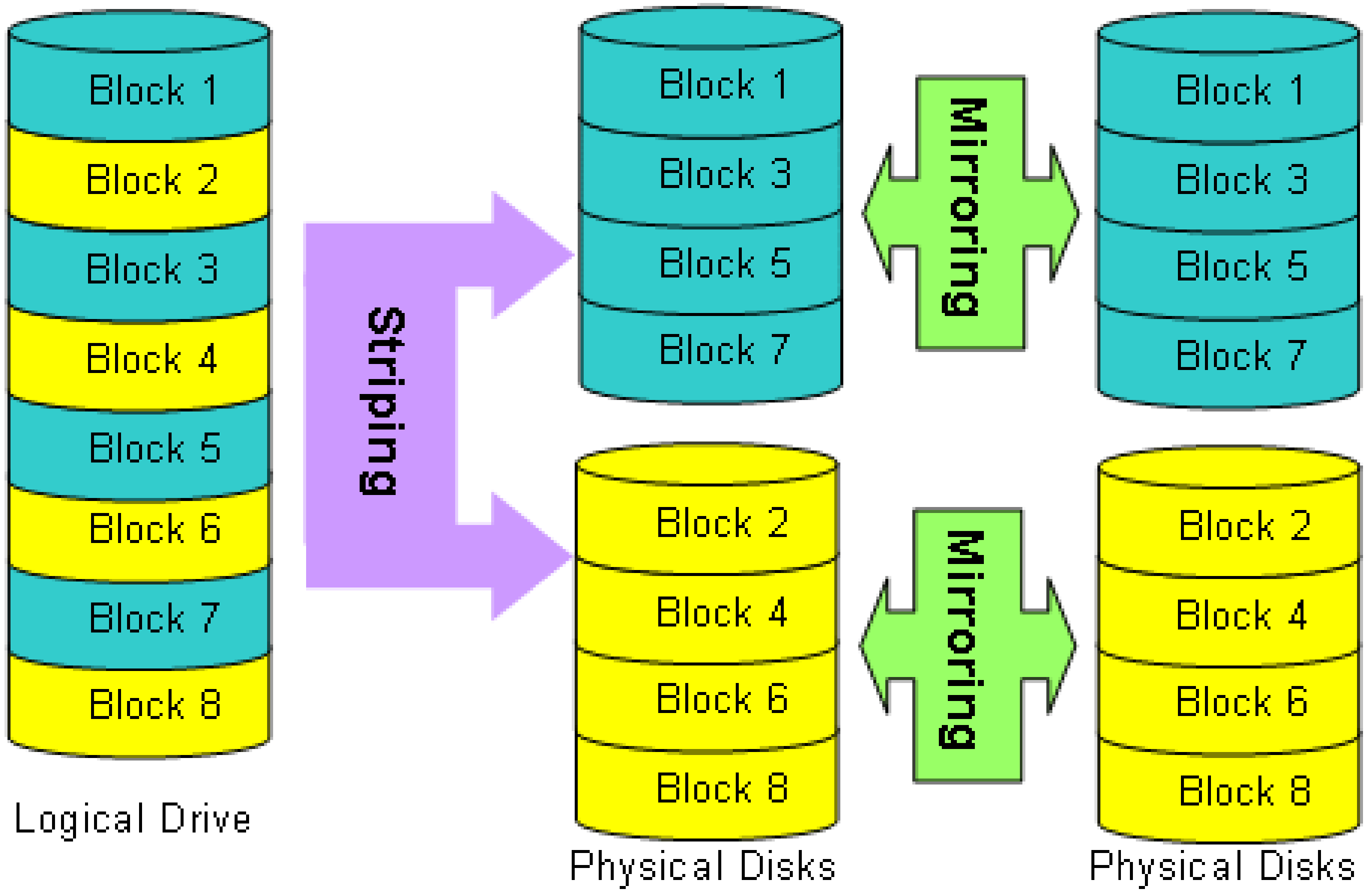
RAID (0 + 1) and (1 + 0)



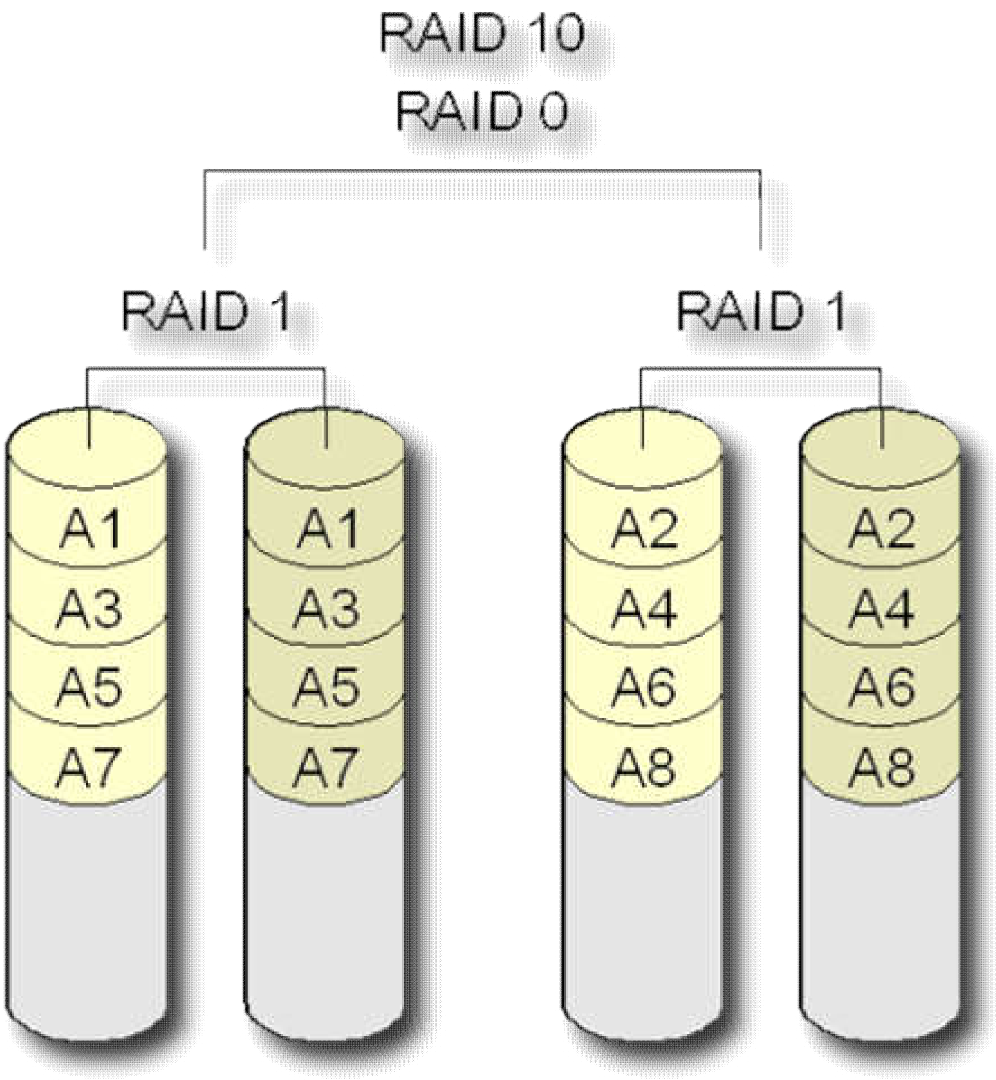
a) RAID 0 + 1 with a single disk failure.

b) RAID 1 + 0 with a single disk failure.

RAID 0+1


Block 1
Block 2
Block 3
Block 4
Block 5
Block 6
Block 7
Block 8
Logical Drive
Striping
Block 1
Block 3
Block 5
Block 7
Block 2
Block 4
Block 6
Block 8
Physical Disks
Mirroring
Mirroring
Block 1
Block 3
Block 5
Block 7
Block 2
Block 4
Block 6
Block 8
Physical Disks

RAID 1+0


RAID 10
RAID 0
RAID 1
RAID 1
A1
A3
A5
A7
A1
A3
A5
A7
A2
A4
A6
A8
A2
A4
A6
A8

Question

- Which is the main reason of stripping data among multiple disks?
  A. increase data volume
  B. increase the total number of files
  C. increase the file size
  D. increase the I/O bandwidth

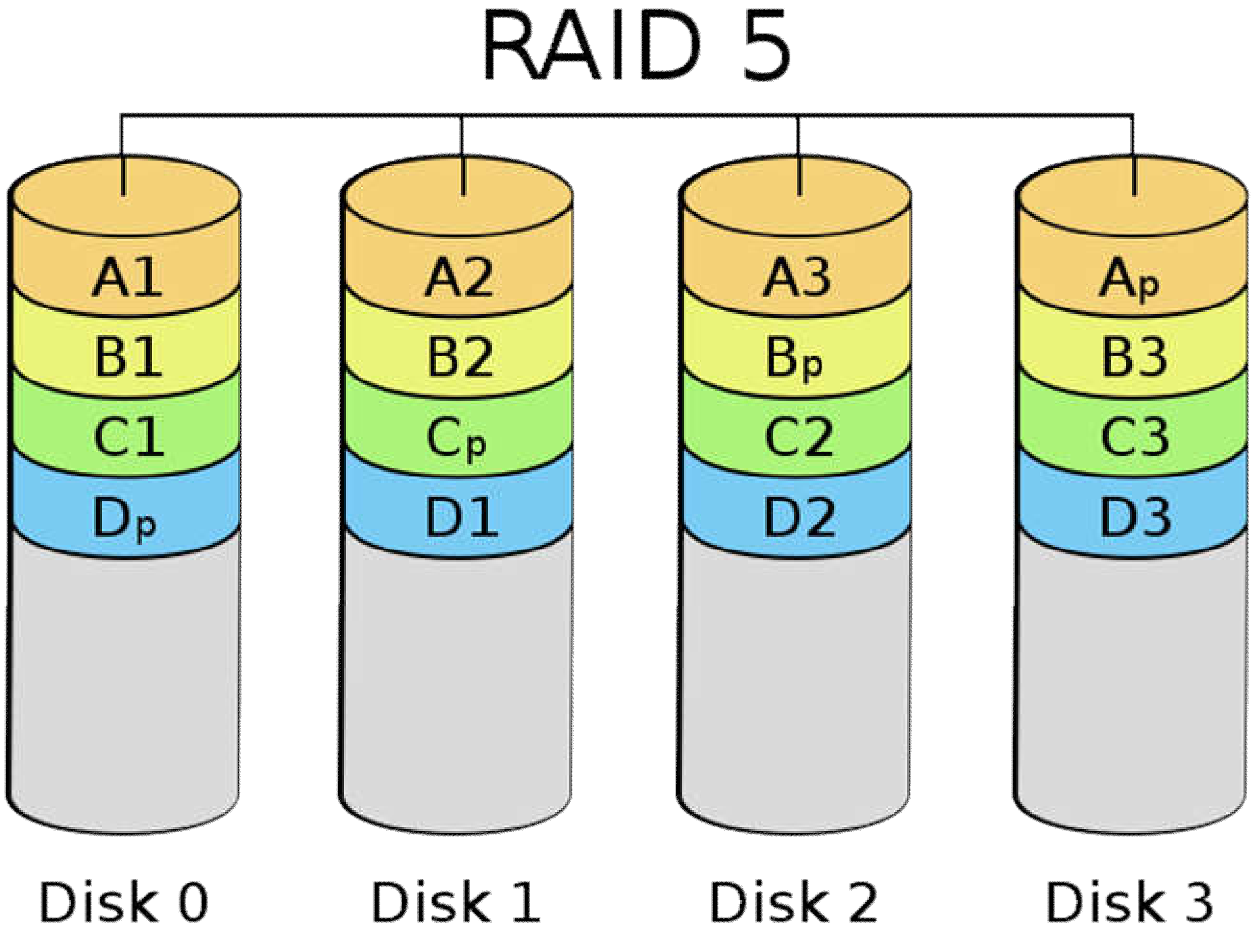
Question

- Which is correct about mirroring data among multiple disks?
  A. support data recovery
  B. increase the total number of files
  C. increase the file size
  D. increase the I/O bandwidth

Question

- How many disks can be broken without losing data in the RAID level 1?
  A. 0
  B. 2
  C. 1
  D. 3

RAID 5



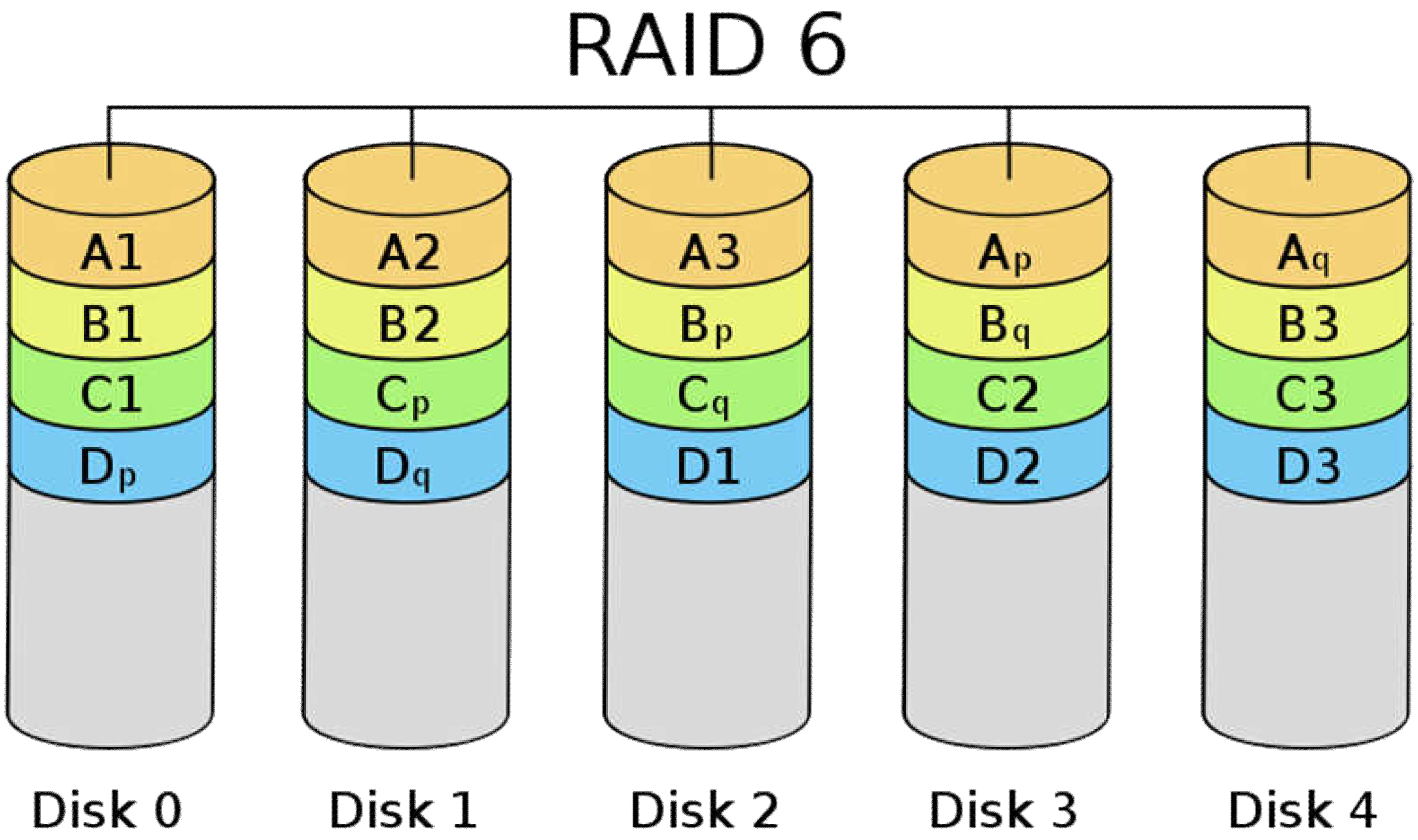
Parity blocks are used instead of mirroring

A1 (1110) A2 (0100) A3 (1001) A_P(0011)

Question

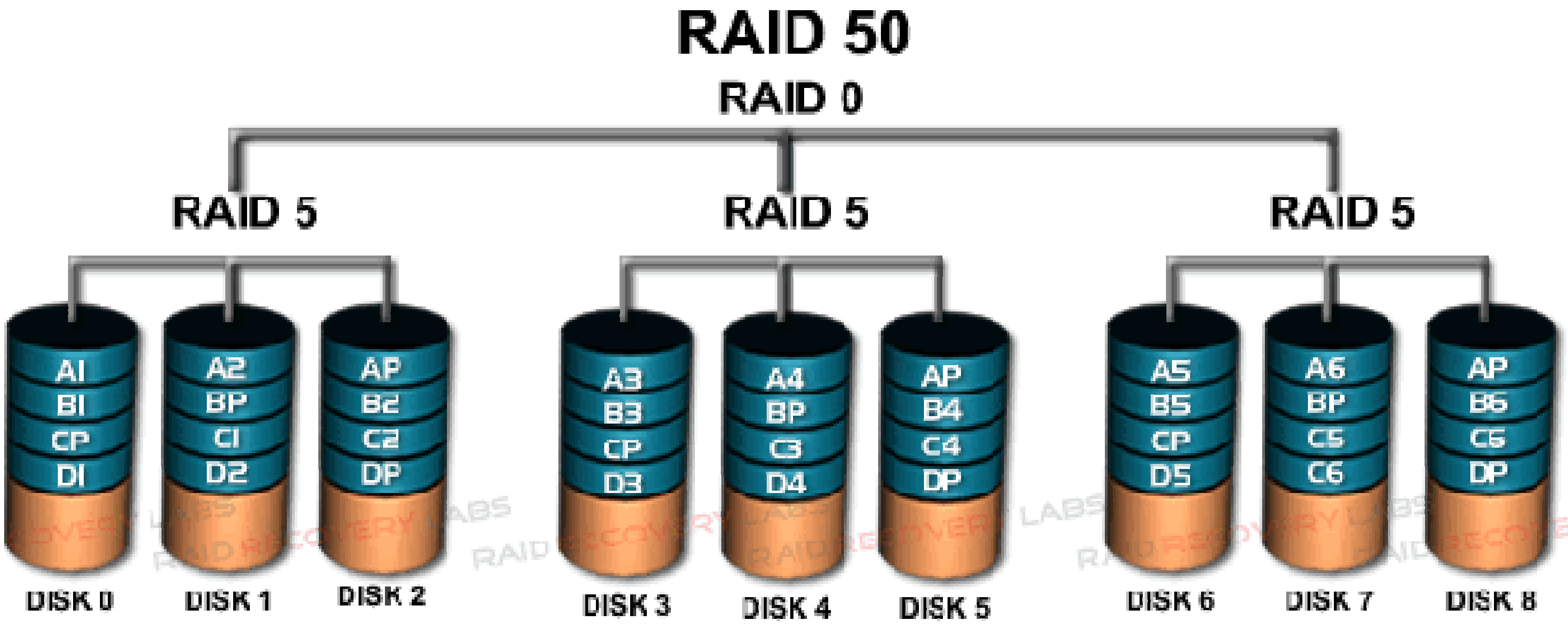
- Which is the most correct about parity blocks?
  A. used to recover data efficiently, similar to mirroring
  B. used to recover disk
  C. used to replace stripping
  D. used to mark file on disk

RAID 6

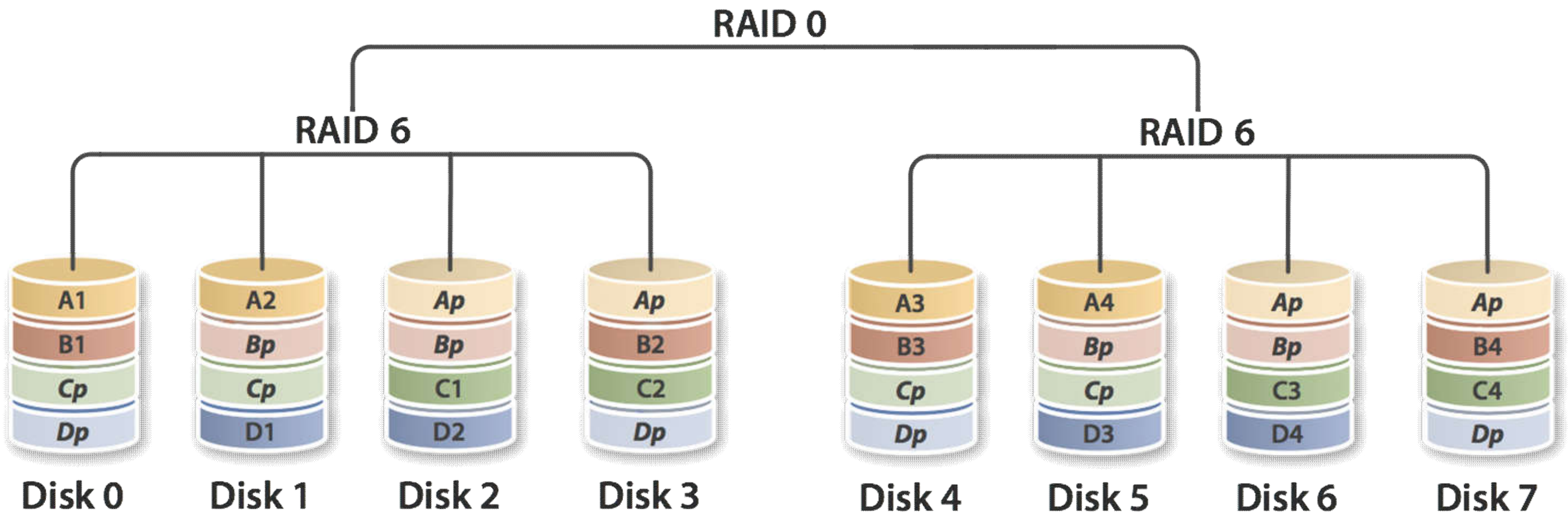


RAID 6
A1
B1
C1
Dp
A2
B2
Cp
Dq
A3
Bp
Cq
D1
Ap
Bq
C2
D2
Aq
B3
C3
D3
Disk 0
Disk 1
Disk 2
Disk 3
Disk 4

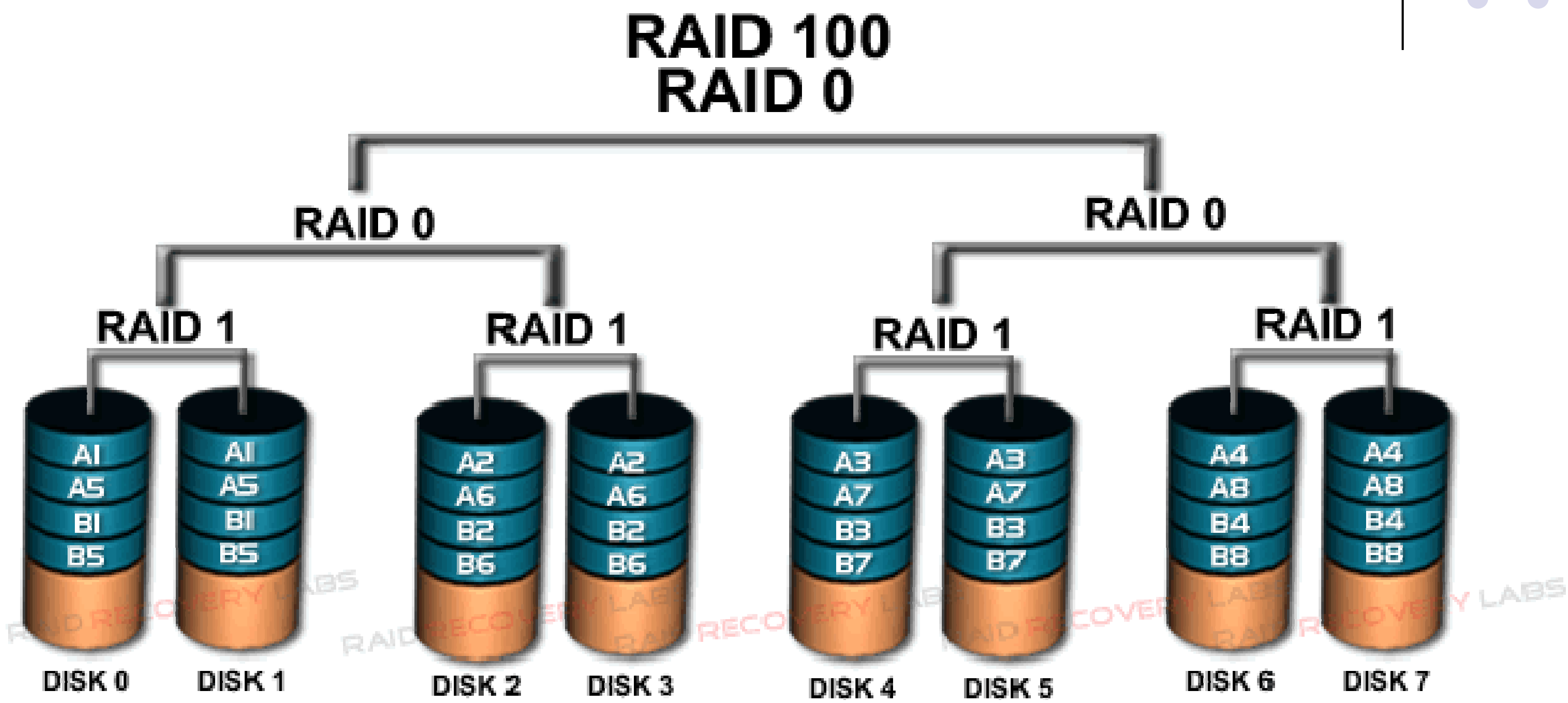
RAID 50


RAID 50
RAID 0
RAID 5
RAID 5
RAID 5
A1
B1
CP
D1
A2
BP
C1
D2
AP
B2
C2
DP
A3
B3
CP
D3
A4
BP
C3
D4
AP
B4
C4
DP
A5
B5
CP
D5
A6
BP
C5
C6
AP
B6
C5
DP
DISK 0
DISK 1
DISK 2
DISK 3
DISK 4
DISK 5
DISK 6
DISK 7
DISK 8

RAID 60


RAID 0
RAID 6
RAID 6
A1
B1
Cp
Dp
Disk 0
A2
Bp
Cp
D1
Disk 1
Ap
Bp
C1
D2
Disk 2
Ap
B2
C2
Dp
Disk 3
A3
B3
Cp
Dp
Disk 4
A4
Bp
Cp
D3
Disk 5
Ap
Bp
C3
D4
Disk 6
Ap
B4
C4
Dp
Disk 7

RAID 100


RAID 100
RAID 0
RAID 0
RAID 0
RAID 1
RAID 1
RAID 1
RAID 1
AI
A5
BI
B5
AI
A5
BI
B5
A2
A6
B2
B6
A2
A6
B2
B6
A3
A7
B3
B7
A3
A7
B3
B7
A4
A8
B4
B8
A4
A8
B4
B8
DISK 0
DISK 1
DISK 2
DISK 3
DISK 4
DISK 5
DISK 6
DISK 7

Stable-Storage

(Stable means the data is safe even the power is suddenly off)

Stable-Storage Implementation

- Write-ahead log scheme requires stable storage
- To implement stable storage
  - Replicate information on more than one nonvolatile storage media with independent failure modes
  - Update information in a controlled manner to ensure that we can recover the stable data after any failure during data transfer or recovery

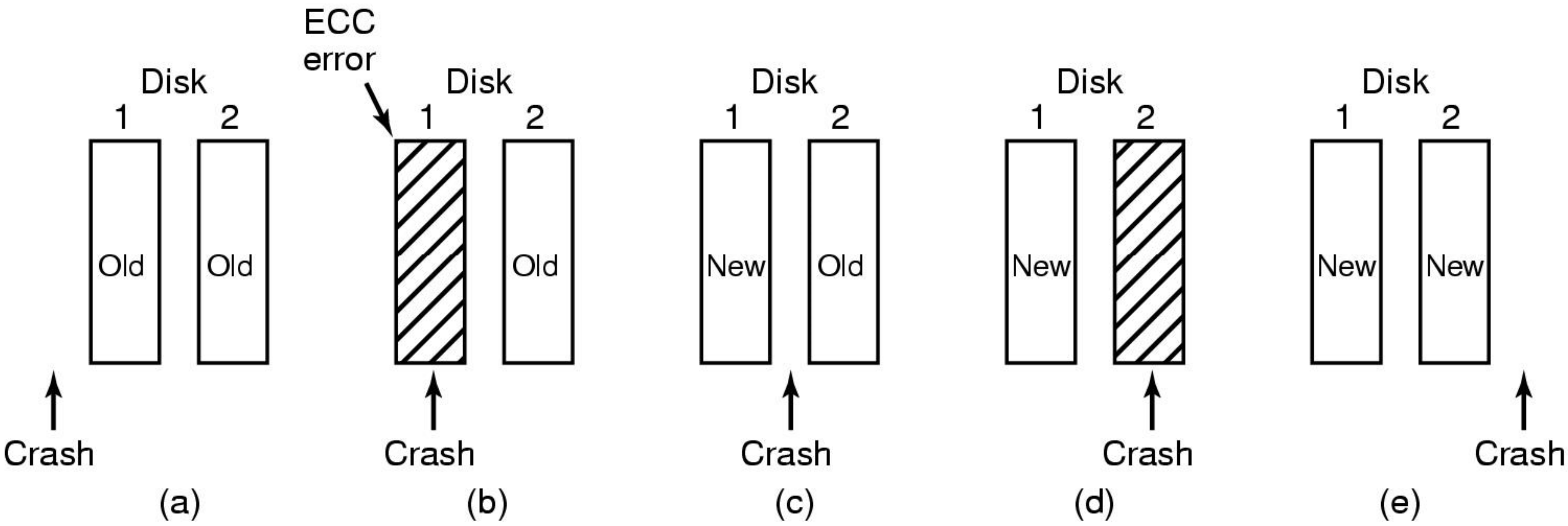
Stable-Storage Implementation

- Write everything twice to separate disks
  - Be sure 1st write does not invalidate previous 2nd copy
  - Read blocks back to validate; then report completion
- Reading both copies
  - If 1st copy okay, use it – i.e., newest value
  - If 2nd copy different or bad, update it with 1st copy
  - If 1st copy is bad; update it with 2nd copy – i.e., *old value*

Stable Storage (continued)

- Crash recovery
  - Scan disks, compare corresponding blocks
  - If one is bad, replace with good one
  - If both good but different, replace 2nd with 1st copy
- Result:
  - If 1st block is good, it contains latest value
  - If not, 2nd block still contains previous value
- An *abstraction* of an *atomic disk write* of a single block
  - Uninterruptible by power failure, etc.

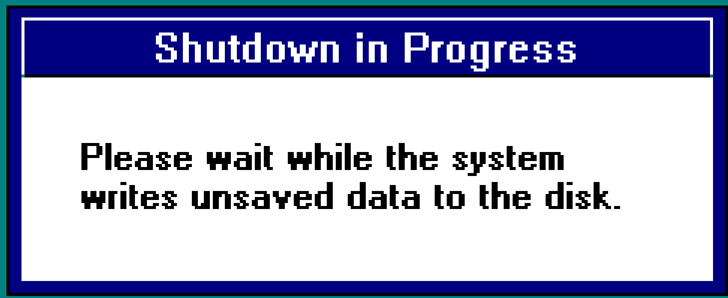
Stable Storage



Analysis of the influence of crashes on stable writes

Tertiary Storage Devices

- Low cost is the defining characteristic of tertiary storage
- Generally, tertiary storage is built using *removable media*
  - CD-ROMs; Floppy, Flash, WORM, tapes


Shutdown in Progress
Please wait while the system
writes unsaved data to the disk.

Question?