

Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of *logical blocks*, where the logical block is the smallest unit of transfer.
- The 1-dimensional 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.

Disk Scheduling

- The operating system is responsible for using hardware efficiently - for the disk drivers, this means having a fast access time.
- Access time has two major components
 - *Seek time* is the time that the disk moves the heads to the cylinder containing the desired sector.
 - *Rotational latency* is an additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time \approx seek distance
- Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

Disk Scheduling (Cont.)

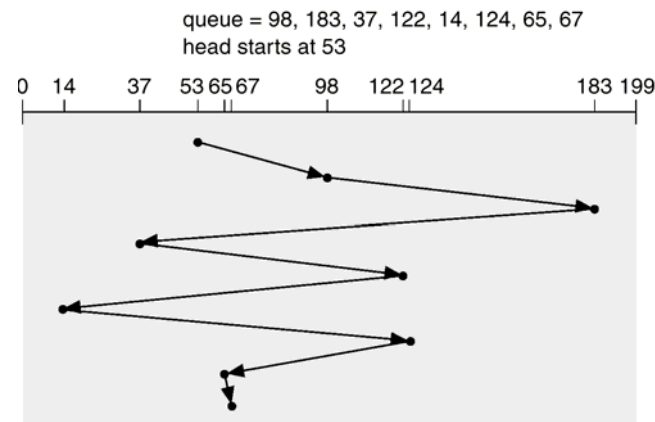
- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue (0-199).

98, 183, 37, 122, 14, 124, 65, 67

Head pointer 53

FCFS

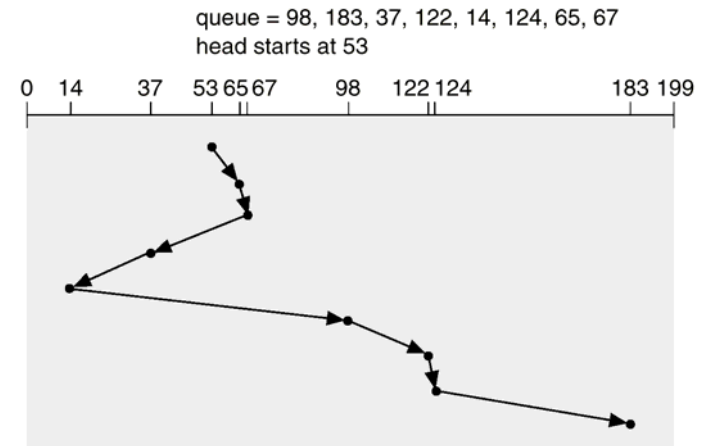
First Come First Served. Illustration shows total head movement of 640 cylinders.



SSTF

- Shortest Seek Time First - 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.

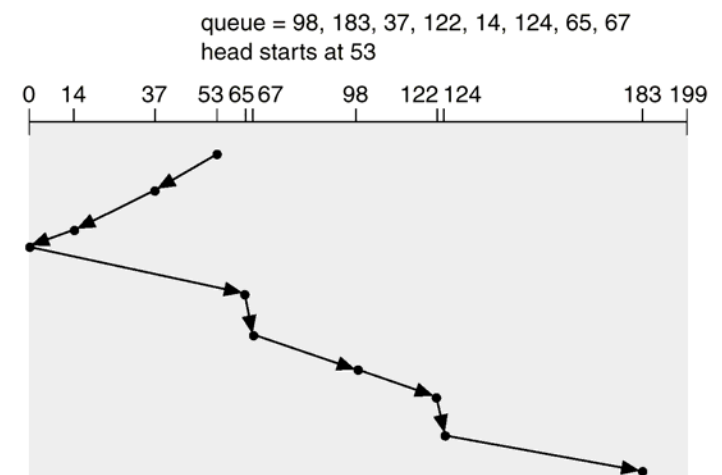
SSTF (Cont.)



SCAN

- The disk arm starts at one end of the disk and moves toward the other end, servicing requests until it will get to the other end of the disk, where the head movement is reversed and the servicing continues.
- Sometimes called the *elevator algorithm*.
- Illustration shows total head movement of 208 cylinders.

SCAN (Cont.)

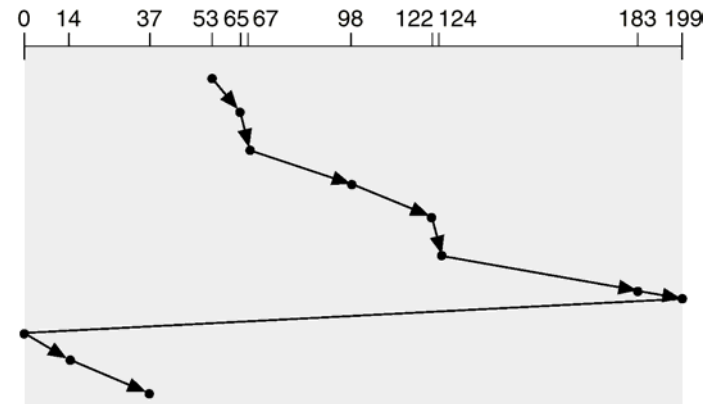


C-SCAN

- 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. However, when it reaches the other end, it immediately will return to the beginning of the disk, without servicing any requests on the return trip.
- Treats the cylinders as a wraparound circular list from the first cylinder to the last one.

C-SCAN (Cont.)

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

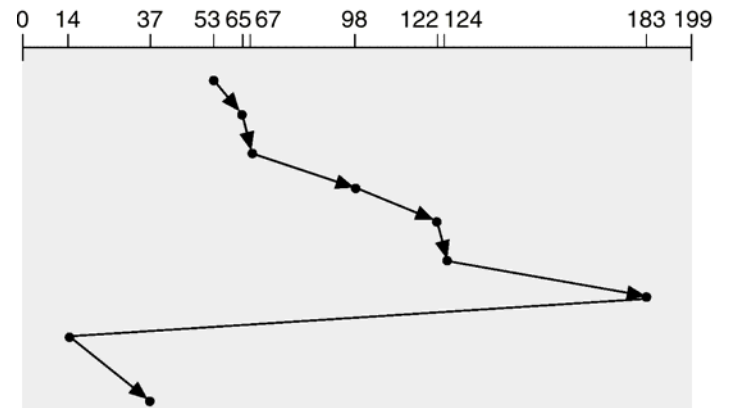


C-LOOK

- A version of C-SCAN
- Arm goes only 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.
- Similarly, LOOK is a version of SCAN which only goes as far as the last request in each direction.

C-LOOK (Cont.)

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



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.

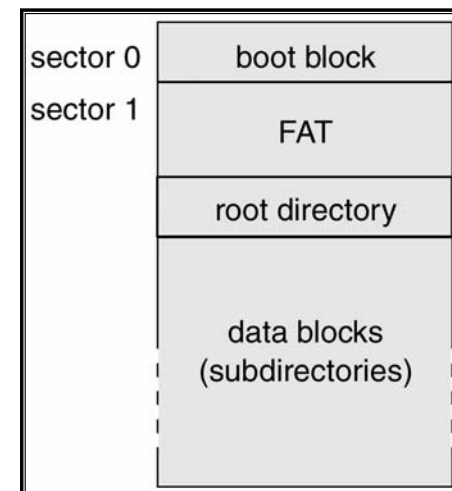
Linux -- Anticipatory Scheduling

- After servicing a request ... WAIT.
 - Yes, this means do nothing even though there is work to be done.
- If a nearby request occurs soon, service it.
- If after waiting this short time nothing occurs, C-LOOK.
- Windows is still with C-LOOK.

Disk Management

- *Low-level formatting, or physical formatting* - Dividing a disk into sectors that the disk controller can read and write.
- To use a disk to hold files, the operating system still needs to record its own data structures on the disk.
 - *Partition* the disk into one or more groups of cylinders.
 - *Logical formatting* or “making a file system” on those partitions.
- Boot block initializes system.
 - The bootstrap is stored in ROM.
 - Bootstrap loads the boot program from the boot block.
 - The boot block can be on the disk, a floppy diskette or a CD-ROM.
- Methods such as *sector sparing* used to handle bad blocks.

Windows Disk Layout



Data Striping

- Data Striping is a method of concatenating multiple drives into one logical storage unit. Striping involves partitioning each drive's storage space into stripes. These stripes are then interleaved, so that the combined space is considered as one drive.
- Most multi-user operating systems today, like Unix, Windows2000 and Netware, support overlapped disk I/O operations across multiple drives. However, in order to maximize throughput for the disk subsystem, the I/O load must be balanced across all the drives so that each drive can be kept busy as much as possible. In a multiple drive system without striping, the disk I/O load is never perfectly balanced. Some drives will contain data files which are frequently accessed and some drives will only rarely be accessed.

Striping of Large Records

- In single-user systems which access large records, small stripes (typically one 512-byte sector in length) can be used so that each record will span across all the drives in the array, each drive storing part of the data from the record. This causes long record accesses to be performed faster, since the data transfer occurs in parallel on multiple drives.
- Applications such as on-demand video/audio, medical imaging and data acquisition, which utilize long record accesses, will achieve optimum performance with small stripe arrays.

RAID

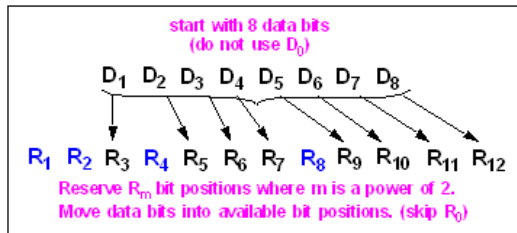
- RAID is Redundant Arrays of Inexpensive Disks. Was suggested on 1987 by Patterson, Gibson and Katz at the University of California Berkeley and nowadays is widely used.
- The basic idea of RAID was to combine multiple small, inexpensive disk drives into an array of disk drives which yields a better fault tolerance.
- This array of drives appears to the computer as a single logical storage unit or drive.

The different RAID levels

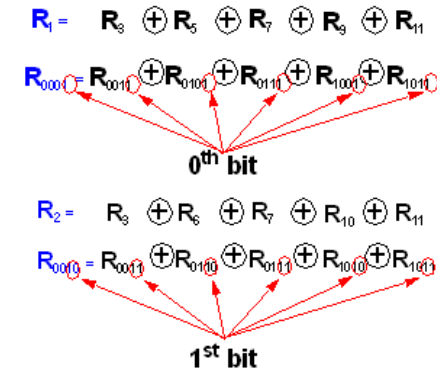
- RAID-0 is commonly referred to as striping. It is not redundant, hence does not truly fit the "RAID" acronym. In level 0, Since no redundant information is stored, performance is very good, but the failure of any disk in the array results in data loss.
- RAID-1 is commonly referred to as mirroring. It provides redundancy by writing all data to two or more drives. The performance of a level 1 array tends to be faster on reads and slower on writes compared to a single drive, but if either drive fails, no data is lost.
- RAID-2 is of little use. It uses Hamming error correction codes, which can use less redundant disks than full mirroring (e.g. in order to backup 4 disks, only 3 are needed), but cannot fix all of errors.

Hamming error correction codes

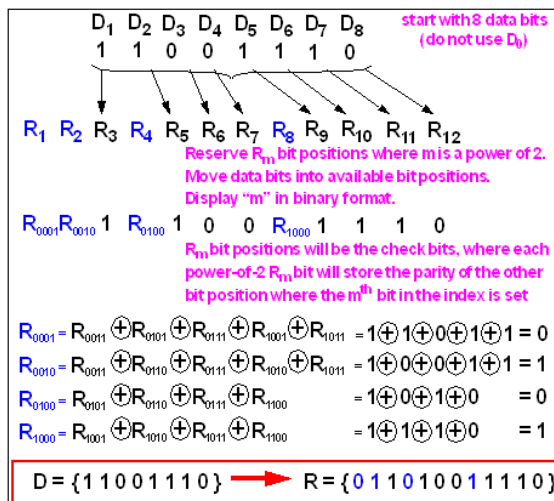
- Hamming Error Correcting Code (ECC) maps a given data vector into a longer codeword.



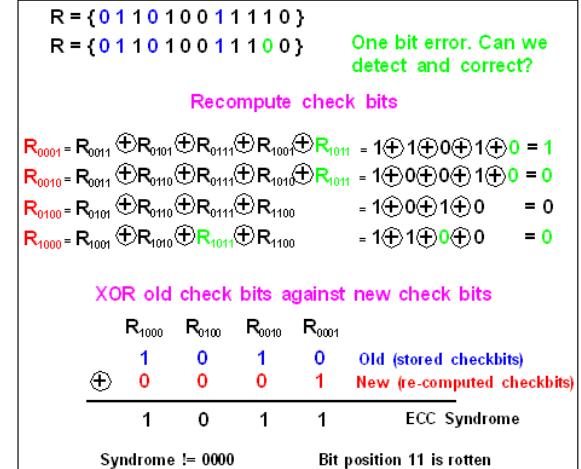
ECC additional bits



ECC procedure



ECC correction procedure



Multi bit error

$R = \{011010011110\}$
 $R = \{011010011101\}$ **Multi bit error. Can we detect and correct?**

Recompute check bits

$R_{0001} = R_{0011} \oplus R_{0101} \oplus R_{0111} \oplus R_{1001} \oplus R_{1011} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$
 $R_{0010} = R_{0011} \oplus R_{0110} \oplus R_{0111} \oplus R_{1010} \oplus R_{1011} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$
 $R_{0100} = R_{0101} \oplus R_{0110} \oplus R_{0111} \oplus R_{1100} = 1 \oplus 0 \oplus 1 \oplus 1 = 1$
 $R_{1000} = R_{1001} \oplus R_{1010} \oplus R_{1011} \oplus R_{1100} = 1 \oplus 1 \oplus 0 \oplus 1 = 1$

XOR old check bits against new check bits

	R_{1000}	R_{0100}	R_{0010}	R_{0001}	
	1	0	1	0	Old (stored checkbits)
\oplus	1	1	0	1	New (re-computed checkbits)
	0	1	1	1	Difference!

Oops, Bit position 7 is NOT rotten

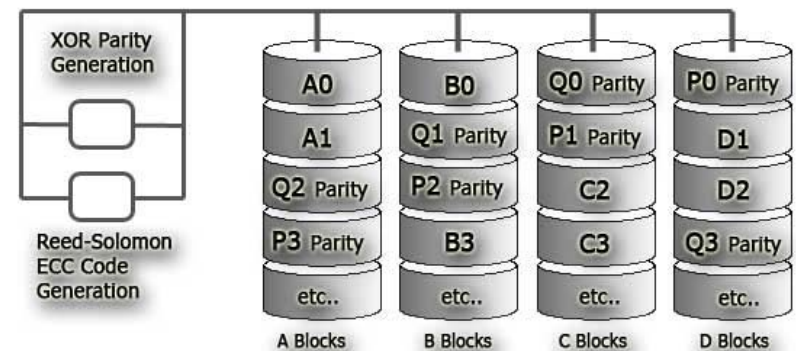
The different RAID levels (Cont.)

- RAID-3 stripes data at a byte level across several drives, with parity-bit stored on one drive. The parity information allows recovery from the failure of any single drive. It is otherwise similar to level 4.
- RAID-4 stripes data at a block level across several drives, with parity stored on one drive. The performance of a level 4 array is very good for reads (the same as level 0). Writes, however, require that parity data be updated each time. Hence, does not support multiple simultaneous write operations. This slows small random writes, in particular, though large writes or sequential writes are fairly fast. Because only one drive in the array stores redundant data, the cost per megabyte of a level 4 array can be fairly low.

The different RAID levels (Cont.)

- RAID-5 is similar to level 4, but distributes parity among the drives. This can speed small writes in multiprocessing systems, since the parity disk does not become a bottleneck. Because parity data must be skipped on each drive during reads, however, the performance for reads tends to be considerably lower than a level 4 array. The cost per megabyte is the same as for level 4.
- RAID level 6 is similar to RAID level 5; however it allows extra fault tolerance by using a second independent parity scheme. In RAID 6 data is striped on a block level across a set of drives and a second set of parity is calculated according to ECC and written across all the drives.

RAID 6



RAID 6

- The basic idea of RAID 6 is to deploy the Hamming ECC algorithm and in addition it uses the parity check bit.
- The purpose of the parity bit is to provide a quick sanity check to ensure that the error that had occurred was not a double bit error.
 - In the case of a single bit error, both the ECC syndrome as well as the parity bit will report that an error had occurred.
 - In the case of a two bit error, the parity bit will returned the same parity as the original codeword, whereas the ECC syndrome will return a bit position that is not zero.
- Any number of odd bit errors is still indistinguishable from a one bit error.
- Many times an entire disk is corrupted and this is very easy to detect. No need for ECC.

RAID Management

- Hardware RAID - The hardware based system manages the RAID subsystem independently from the host and presents to the host only a single disk per RAID array.
- Software RAID - software-based arrays occupy host system memory, consume CPU cycles and are operating system dependent. Hence, degrade overall server performance. Also, unlike hardware-based arrays, the performance of a software-based array is directly dependent on server CPU performance and load.
- Hardware arrays are also highly fault tolerant. Software arrays, will fail to boot if the boot drive in the array fails. An array implemented in software can only be functional when the array software has been read from the disks and is memory-resident. Software-based implementations commonly require a separate boot drive, which is NOT included in the array.

SCSI Devices

- SCSI stands for Small Computer Systems Interface
- It is a standardized way of connecting hardware peripherals to a computer using standardized hardware and control commands.
- SCSI can take the disk scheduling task from the Operating System.
 - SCSI uses C-LOOK.
- The devices on the SCSI bus talks to the computer through the SCSI controller. On modern PCs the SCSI controller is usually connected to the PCI bus either as an on-board solution on motherboards or as a separate card in a PCI slot.
- All devices have the ability to release the controller after being requested to do time consuming operations not requiring the availability of the controller and leaves it free for other devices to use for transferring data or receiving commands.
- Narrow SCSI uses a data pathway of 8 bits. Wide SCSI uses a data pathway of 16 bits. A "very wide" 32-bit form of SCSI was defined as part of the SCSI-2 standard.

IDE Devices

- IDE stands for Integrated Device Electronics and it is also called ATA (AT Attachment) or PATA (Parallel AT Attachment).
- IDE integrates the controller on the disk itself; hence no need for an IDE card.
- The bus width is always 16 bits.
- There are primary and secondary IDE buses with a master device and a slave device for each. I.e. 4 devices at most.
- The priority of the devices is 1. master of primary, 2. slave of primary, 3. master of secondary and 4. slave of secondary.

SCSI vs. IDE

- IDE can have 4 device while SCSI can address 8 devices using Narrow SCSI, 16 devices using Wide SCSI, 32 using Very Wide SCSI and 126 devices using FireWire.
- SCSI is faster and has a wider bandwidth.
- SCSI can queue up to 256 commands per logical unit. The IDE devices lack the intelligence to perform command queuing.
- SCSI hard disk drives aimed at the extreme performance server market have had a lot of research and development time on optimizing seek patterns and rescheduling commands to minimize seek times and maximize throughput.
- Connectors suitable for hot-swapping drives in RAID-systems is something only SCSI can support.
- SCSI devices can sustain higher temperatures and stay mechanically functional despite the expansion of the metal parts with temperature.
- SCSI devices are significantly more expensive.

SATA and SAS

- While ATA is based on a 16 bit parallel interface, Serial Advanced Technology Attachment (SATA) is a single bit serial advancement of the Parallel ATA.
- The transmission time on the cable is no longer a bottleneck, so no need for several parallel bits in a cable.
- While ATA can address 4 devices, the SATA driver can address only one device; hence any additional device requires an additional driver.
- The newer versions of SATA can also support command queuing and hot swapping.
- Serial Attached SCSI (SAS) is the serial version of SCSI.
- SAS can address 16256 devices.
- SAS is designed to support SATA devices.
 - SATA cannot support SAS. SATA integrates the controller on the disk itself; whereas, SAS has no controller on the disk.

Price per Megabyte of Magnetic Hard Disk, From 1981 to 2004

