KCS-401

Operating System

Dr. Upasana Pandey

Associate Professor

Department of Computer Sciences & Engineering

IMS Engineering College (College Code: 143)

DISK MANAGEMENT

Disk Formatting :

- Before a disk can store data, it must be divided into sectors that the disk controller can read and write.
- This process is called low-level formatting or physical formatting.
- Low level formatting fills the disk with a special data structure for each sector.
- The data structure for a sector typically consists of a header, a data area (usually 512 bytes in size), and a trailer.
- The header and trailer contain information used by the disk controller, such as sector number and an error-correcting code (ECC).
- When the controller writes a sector of data during normal I/O, the ECC is updated with a value calculated

from all the bytes in the data area.

- To use a disk to hold files, the operating system still needs to record its own data structure on the disk. It does so in two steps :

  (i) The first step is to partition the disk into one or more groups of cylinders. The operating system can treat each partition as though it were a separate disk.

  (ii) After partitioning, the second step is logical formatting (or creation of a file system). Where, the operating system stores the initial file-system data structures onto the disk.

- To increase efficiency, most file systems group block together into larger chunks, frequently called clusters.

(DM.2)

Boot Block :

- For a computer to start running - for instance, when it is powered up or rebooted - it must have an initial program to run.
- This initial bootstrap program tends to be simple. It initializes all aspects of the system, and then starts the operating system.
- To do its job, the bootstrap program finds the operating system kernel on disk, load that kernel into memory, and jumps to an initial address to begin the operating-system execution.
- The bootstrap is stored in read-only memory (ROM).
- Since ROM is read only, it cannot be injected by a computer.

(DM. 3)

BAD BLOCKS :

- Because disks have moving parts and small tolerances they are prone to failure.

- Sometimes the failure is complete; in this case, the disk needs to be replaced and its contents restored from backup media to the new disk.

- Most disks even come from the factory with bad blocks. Depending on the disk and controller in use, these blocks are handled in a variety of ways.

- On simple disks, such as some disks with IDE controllers, bad blocks are handled manually.

- More sophisticated disks, such as the SCSI disks used in high end PCs and most workstations and servers, are smarter about bad block recovery.

(DM.4)

- The controller maintains a list of bad blocks on the disk. The list is initialized during the low-level formatting at the factory and is updated over the life of the disk.

- Low level formatting also sets aside spare sectors not visible to the operating system.

- The controller can be told to replace each bad sector logically with one of the spare sectors. This scheme is known as sector sparing or forwarding.

- A typical bad-sector transaction might be as follows :

  1. The operating system tries to read logical block 87.

  2. The controller calculates the ECC and finds that the sector is bad. It reports

this findings to the operating system.

3. Next time, the system is rebooted, a special command is run to tell the SCSI controller to replace the bad sector with a spare.

4. After that, whenever the system requests logical block 87, the request is translated into the replacement sector's address by the controller.

- Due to optimization factors, most disks are formatted to provide a few spare sector in each cylinder and a spare cylinder as well.

- As an alternative to sector sparing, some controllers can be controllers can be instructed to replace a bad block by sector slipping.

(DM. 6)

- An unrecoverable hard error, however, results in lost data. Whatever file was using that block must be repaired and that requires manual intervention

SWAP SPACE MANAGEMENT

Swap Space :

It is a space on hard disk used as the virtual memory extensions of a computer's real memory. Having a swap space allows your computer to pretend that you have more RAM than you actually have.

Why swap space ?

When the amount of physical memory (RAM) is full and the system needs more memory resources for new process then system may get slow because there is no physical memory available.

Swap Space Management :

- Swap space management is another low-level task of operating system.
- Virtual memory uses disk space as an extension of main memory.
- Since disk acess is much slower than memory access, using swap space significantly decreases system performance
- Goal is to provide the best throughput for the virtual memory system.

Swap Space Use :

- System that implement swapping may use swap space to hold an entire process image, including the code and data segments.

- Paging system may simply store pages that have been pushed out of main memory.

- It is safer to overestimate than to underestimate the amount of swap space required, because if a system runs out of swap space it may be forced to abort processes or may crash entirely.

- Overestimation wastes disk space that could otherwise be used for files, but it does no other harm.

Swap Space Location

- If the swap space is simply a large file within the file system, normal file-system routines can be used to create it, name it, and allocate its space. This approach, through easy to implement but inefficient.

- Problems of External fragmentation, which causes system execution to get slow.

- We can improve performance by caching the block location information in physical memory and by using special tools to allocate physically contiguous blocks for the swap file, but the cost of traversing the file-system data structure still remains.

- Alternatively, swap space can be created in a separate raw partition, as no file system or directory structure is placed in this space.

(SS.4)

- A separate swap-space storage manager is used to allocate and deallocate the blocks from the raw partition as no file system or directory structure is placed in this space.

- Swap space is reinitialized at boot time so any fragmentation is short-lived. This approach creates a fixed amount of swap space during disk partitioning.