Operating systems

ELTE IK.

Dr. Illés Zoltán

zoltan.illes@elte.hu

Financed from the financial support ELTE won from the Higher Education Restructuring Fund of the Hungarian Government

We have talked about…

- Operating system generations
  - Typical hw constructions
- Op. Systems definitions
- Notions:
  - Files, directories, processes
- System calls
- System structures
  - Today: Client-server model, with layered features

What comes today…

- Data storage types
  - Magnetic tape, magnetic disk, optical storage
- Formatting,
- MBR, partitions
- Boot process
- Disk access, features, schedulers, efficiency
- Reduntant data storage, RAID

Data storage types

- Magnetic storage type
  - Magnetic tapes
  - Magnetic drives
    - Hard disk
    - Floppy disk
- Optical storage type
  - CD, DVD, Blu-Ray, laser, appr. 5xDVD capacity
  - It is based on light(laser, from red to blue) reflection (pit-land)
- Semiconductor principle
  - USB, memory card
  - SSD(Solid State Drive/Disk) disk

Data storage types tomorrow

- Holografic
  - GE communications, 2011, 500GB, holografic storage system, a bit is a hologram
- Biology
- Nano architecture
- ….
- Moore’s law, …”…but…

Physical architecture of magnetic tapes

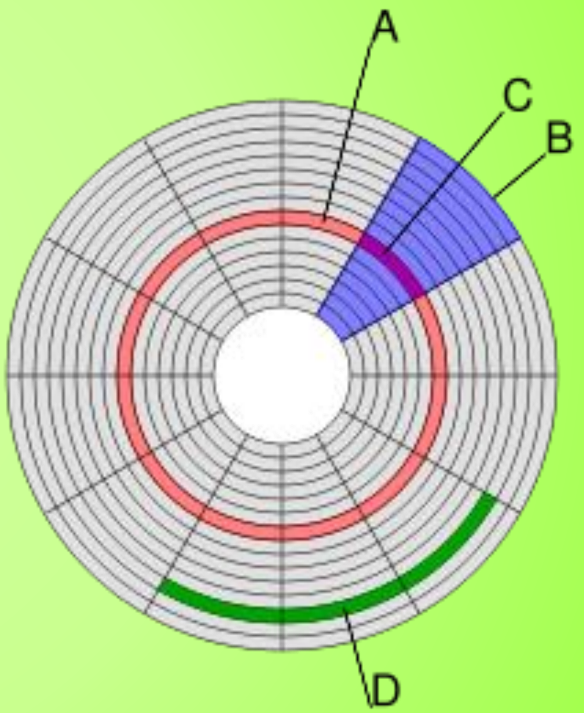
- Magnetic tape- linear storage, architecture
  - Linear Tape File System, XML based structure
  - 9 bits frames (8 bit + 1 bit parity)
  - Frames organized into records
  - Between records: record gap
  - A file is a set of records, between files there is a file gap
  - Beginning of the tape the dir. structure is stored
- Typical usage
  - Security archive (backup)
  - For storing „big data”
- Not so cheep
- Typical size today: DLT (Digital Linear Tape), LTO (Linear Tape-Open) 4 Ultrium 800/1600 GB, LTO5 1.5TB/3TB, LTO6 2.5TB/6.25TB

Architecture of magnetic disks I.

- FDD – Floppy Disk Drive
  - Typically one disk
- HDD – Hard Disk Drive
  - Typically more disk (or one)
- Disk is round – divided into stripes
- Stripes are divided into sectors – intersection is a block
  - cluster – more block
- If the drive has more disks – the stripes above each other : cilinder
- The disk is logicaly a rage of blocks (0–xxx)
- Firmware: hides physical operations

Architecture of magnetic disks II.

- A: stripe (red)
- B: sector (blue)
- C: block, 512 byte
- D: cluster, n is decided by OS. D=n x C, where n=1.. 128.(green: n=3)
- Cilinder: stripes above each other (red)



Addressing of magnetic disk

- CHS address (Cilinder- Head- Sector)
  - Example: 1.44 MB FD
  - No. Of Cilinders: 80 (0-79)
  - No. Of heads: 2 (0-1)
  - No. Of sectors on a track: 18 (1-18)
  - Total size: 80*2*18=2880 sector * 512byte
- LBA address (Logical Block Addressing)
  - Earlier 28 bits, up to 137GB.
  - Now 48 bits, up to144 PB (Petabyte), (144 000 000 GB)

$$A = (c \cdot N_{\text{heads}} \cdot N_{\text{sectors}}) + (h \cdot N_{\text{sectors}}) + s - 1$$

Optical storage

- A 8 or 12 cm (diagonal) optical disks
  - CD – Compact Disc, DVD –Digital Versatile Disc
  - Size: 650MB – 17 GB
  - Speed: 1x = 150 KB/sec
- Working principle: Laser light reflection differences.
  - The time of the reflected light from a pit is longer than the time from the land (pit-land range)
  - Writable disks: writing process: laser beam (heat) changes (increase) the light refraction coefficient (pit).

Device driver

- A program which makes communication between user application and hardware.
- Part of the kernel.
- During disk read or write there is used DMA (big data)
  - Interrupt message: there is finished the read or write activities.
  - I/O ports are used to setup device parameters.
- Layered architecture

Formatting magnetic disks

- Creating the tracks-sectors architecture
- Typical block size is 512 byte
- Disks are ready to use by manufacturers
- Quick format- Normal format
  - A normal format finds bad sectors on the disk.
- Block (Sector) content= Head information+datablock+footer inf.
  - Block inf.: track no., head no., sector number
  - Footer inf.: error corr. block
- Low level format: Creating sectors, done by manufacturer.

Logical formatting

- Creating partitions
  - Allowed max. 4 on a hard disk.
- 0. sector- MBR (Master Boot Record)
  - Contains 2 parts, like every blocksize is 512 byte
    - First part: bootloader code, max. 446 byte
    - Second part: Max. 4 partitions data (4x 16 byte=64 byte)
    - Record close 2 byte: 0x55,0xAA
  - Primary partition- op. system can boot
  - Extended partition- there are more logical drive
  - Swap partition
- Task of formatting: Creating the necessary data structure on the partition (volume)

MBR architecture

MBR structure

Address			Description		Size (byte)
Hex	Oct	Dec			
0000	0000	0	Loader program code		440 (max. 446)
01B8	0670	440	Optional disk code		4
01BC	0674	444	Typically: 0 x 0000		2
01BE	0676	446	Primary partition table data (4 16 byte long part, IBM partition table schema		64
01FE	0776	510	55h	MBR closing: 0 x AA55	2
01FF	0777	511	AAh		
			446 + 64 + 2 =		512

A partition entry description

- 1. byte: Partition status (80=activ, 0=not boot)
- 2-3-4. byte : Partition begin block CHS address
  - 0-5. bit: No. Of heads
  - 6-15. bit: No. Of cilinders
  - 16-23. bit: No. Of sectors
- 5. byte: Partition type (primary, extended)
- 6-7-8. byte : Partition end block CHS address
- 9-10-11-12. byte: Partition begin LBA addr.
- 13-14-15-16. byte: No of sectors
  - 4 bájt: 2^32 *512 ~ 4 GB *512= 2 TB

The Boot process

- From a boot device, ROM-BIOS loads the „boot program” from MBR at 7c00h address.
- Only one primary partition can be active.
- Boot program loads the first block of active partition into memory.
- This is the operating system boot program, eg:LILO, NTFS boot
- The boot program knows, which files can be loaded to make a „systemstart”
  - Multi layered process, vary from operating systems.

Interleave-Block calculation

- For reading-writing we have to calculate the blocks number.
  - Number of heads, sectors
  - Assume we have 4 head (2 or 4 disks)
  - Lets divide one track into 7 sectors
- Because of the rotation speed the blocks logical order is not same with the physical order! (interleave)
  - 1:2 interleave, every second sector gives the log. order"

	1 sector	2 sector	3 sector	4 sector	5 sector	6 sector	7 sector
1 head.	1	17	5	21	9	25	13
2 head.	2	18	6	22	10	26	14
3 head.	3	19	7	23	11	27	15
4 head.	4	20	8	24	12	28	16

Physical parameters of disk access

- Rotation speed (typical 5400,7200,10000 or 15000 rpm)
  - How many times rounds in a minute.
- Head speed
  - Inside a cilinder the head must not to move.
- The task of disk I/O scheduler is to choose an efficient, quick service order
  - Decrease the time of the disk access
  - Increase the data(read-write) bandwith

Reading-Writing access

- We need for a system call(read):
  - The number of block(s) to read.
  - The address space into write.
  - Number of bytes
- We have more processes…more read-write request
  - Main question: Which one is the first?

Scheduling of disk access

- Low level parameters(kernel)
  - Type of request (read-write)
  - A block begin address, (track, sector, head)
  - DMA address
  - Number of bytes to read-write
- Disk is used by every process
  - Who will be served first?
  - We'll calculate of head position

FCFS scheduler

- First Come - First Service order
- Simple strategy, as the requests comes, they will be served in the same order.
- We sure, all requests will be served!
  - No starve (perish with hunger)
- Do not care about head position.
- Not so efficient.
- Low read-write bandwith.
- General, average service time, low deviation.

SSTF scheduler

- Shortest Seek Time First – SSTF
- We will service that request which is more closer to actual head position.
- Average waiting time is low.
  - But the deviation is high.
- High read–write bandwith
- There is real starve danger (perish with hunger)

Scan scheduler

- SCAN (LOOK) method
- The head is moving all time from inside to outside and back and service the requests of call.
- The head turns back if there are no requests at that direction or at reaching the end position.
- The service of a bad timing request is after a „round".
  - Waiting time is medium, high deviation
- Deviation of access tracks located at the center of disk is low.

C-Scan scheduler

- Circural SCAN, C-SCAN
- Modifying of SCAN, reading-writing service is only in one direction.
- Quicker head movement
- Higher bandwith
- Average waiting time, low deviation.
  - No real bad request

Improve scheduler efficiency

- FCFS method: if we have a request near from actual head position, lets to serve it! (Pick up)
- Usually a file data is located in a block. If we ask to read the first part, in that case the system will read the second part too.
  - Scheduler in advance
- The access of the middle of the disks is very efficient!

Improve scheduler efficiency using memory

- DMA (also memory)
- Memory buffer
  - Double using
  - Read: Scheduler writes information, user reads it.
  - Write: User writes information, schedules reads it
- Disc cache
  - Reads or writes not only the asked data, but the closest integral data too.
  - Gives additional operating system task
    - Eg: Smartdrive

Which scheduler can we choose?

- The algorithms calculates only with the head position
- FCFS is used in a single user system.
- SSTF, There is real danger of starve
- C-Scan , Big I/O bandwith, no starve
- Bild in schedules: eg. SCSI controller
  - Generally OS sends requests one after the other.

The main role of scheduler

- To give quick answer to requests.
- We can (OS too) improve it:
  - Defragmentation
  - Best bandwith in the middle of disks.
  - Access of disks is more quick in the middle too (virtual memory)
  - Disk buffer in the memory.
  - Maybe data compression (more CPU load)

Disk efficiency–Redundant data storage

- Definition: Avoid data loss in case of disk error
- Operating system support
  - Dynamic volume– more than one disks gives a logical volume. The volume size is the sum of disks.
  - Mirroring– Two disks works parallel as one volume. The volume size is one disk (less) size.
  - Bigger CPU load.
- Hardware support
  - Intelligent devices
  - SCSI devices supports (RAID)

Redundant drives

- RAID – Redundant Array of Inexpensive Disks
- First was used at SCSI devices
  - Small Computer System Interface
    - A definition, a standard for data exchange between computers and devices.
    - Usually was used to connect HDD’s to computers.
  - Today SCSI rarely used, instead: SAS controller (Serial Attached SCSI)

RAID

- If an operating system supports it: SoftRaid.
- If a hardware supports it: Hardware Raid- or simply Raid system.
- However in the names inexpensive, but usually more expensive than a normal disk.
- Collect more disks, and gives as a single volume to the operating system.
- There are more type of raid’s:RAID 0-6

RAID 0(striping)

- This is a Raid, but not redundant…
- Like OS supported dynamic volume, but this is supported by a RAID controller. Adds more disks into a collection (Raid Array) and
- It gives the summary of size of individual disks as a new logical disk.
- Striping, a (big) file parts is stored on more disks.
- Quick I/O execution.
- No error correction, no redundancy.

RAID 1 (Mirroring)

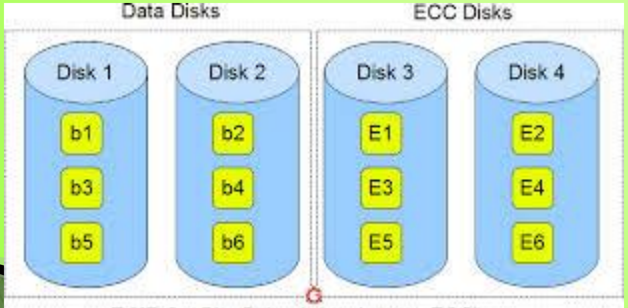
- Creates a logical volume from two independent disks.
- Every data is stored on both disks (mirror)
- The total storage capacity is half size.
- It is expensive. (2 disks, one size)
- Significant redundancy.
  - Occuring at one time some disk error on both disk, data lost.

RAID 1+0, RAID 0+1

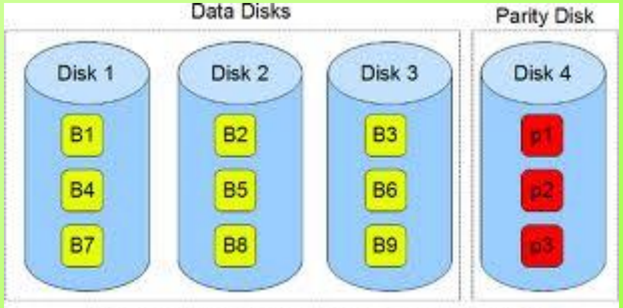
- RAID 1+0: Creating a mirror using RAID 0 volumes.
- RAID 0+1: Creating volumes using RAID 1 (mirrored) disks.
- Nowdays the cotrollers are usually support them, but rarely used because of price!! (mirror, 2 disk 1 size)

RAID 2,3,4

- RAID 2: It contains not only data bits, but error correction bits too. (ECC-Error Correction Code) E.g. 2 data disk, 2 ECC disk
- RAID 3: Only a Plus parity disk, n+1 disk, sum_size=Σ n
- RAID 4: RAID0 implementation with parity disk.
- RAID 2,3,4: Rarely used.



RAID 2 – Bits Striped. (and stores ECC)



RAID 3 – Bytes Striped. (and Dedicated Parity Disk)

RAID 5

- No parity disk, the parity information is striped. (stripe set)
- Data, files etc. is striped also!
- Intensive CPU load (controller has own CPU!!!)
- Redundant storage, occuring 1 disk error do not cause data lost
  - 2 disk errors at the same time occurs data lost!
  - How is it work? (Form the parity and from data bits we can recalculete, recover a lost bit!)
- N disks in a RAID 5 array(N>=3), Total volume size is sum. N-1 disks.

RAID 6

- Add to RAID 5 parity block an error correction code.(+1 disk)
- More intensive CPU load.
- Two disks failure does not cause a „system crash", a data lost!
- Relative expensive
- Capacity of N disk in a RAID 6 array, is the same as N-2 disk capacity.
- In principle we can continue the error correction method (3 disk failure, etc…)

RAID summary

- Most often used RAID types: RAID 1,5
- RAID 6 controllers appeared last 1-2 years.
  - Raid- Inexpensive disk, but in reality the Raid disks more expensive than other, normal disks
  - Raid 6-2 disks spare, not effective used, a little bit more expensive!
- Hot-Swap RAID controllers- Hot-swap disks: during computer work(without switch off), we can safely change the failed disk.

Summary of data storage

- How can we support safe data storage?
- Multilayer result:
  1. Physical disks(HDD)
  2. Hardware RAID
  3. Partitions
  4. Software RAID
  5. Volume Manager in the operating system.
- It is not safe if:
  - eg: Power supply (redundant too) is off, operator mistake, etc.
  - Software attacks, viruses.
- How our data is organised on a „volume”?

Thanks for your attention!

zoltan.illes@elte.hu