# Operating systems 1, Lecture 9

Sanda-Maria AVRAM, Ph. D.

Leave a feedback

1

## The internal structure of the disk and file system

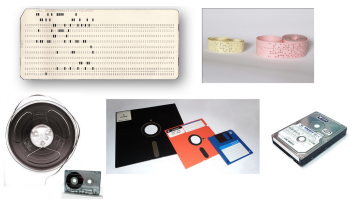
**UNIX**: the i-node mechanism

**DOS**: FAT table

**Windows NT & 2000**: the NTFS mechanism

2

## Evolution of storage media



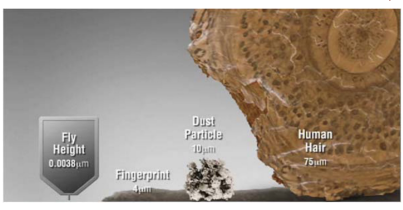
3

## Storage technologies

| Technology | Representation of bits at elementary level | Examples |
|---|---|---|
| Magnetic | - in the form of negatively or positively magnetized particles. | HDD, diskette, tape, magnetic tape |
| Optical | - in the form of microscopic points or cavities that have the role of reflecting light differently. | CD, DVD, mini-disc |
| Electronic | - the presence or absence of an electrical current in certain circuits, or the modulation of the state of certain circuits (closed-open) can also be interpreted as bits. | RAM, flash memory |

4

## The Hard Disk Is Sealed To Protect The Read Heads

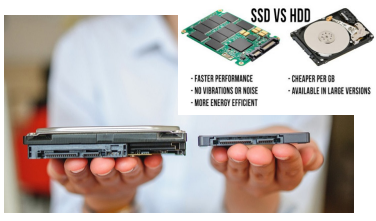



5

## HDD versus SSD

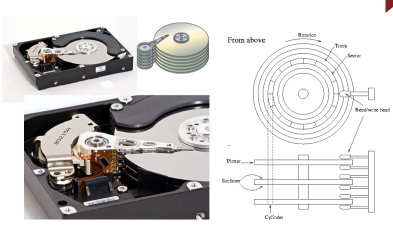

6

## HDD versus SSD




7

## Structure of a HDD




8

## Solid State Disks (SSDs)

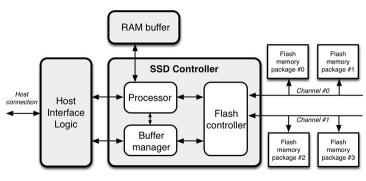
1995 – Replace rotating magnetic media with non-volatile memory (battery backed DRAM)

2009 – Use NAND Multi-Level Cell (2-bit/cell) flash memory

- Sector (4 KB page) addressable, but stores 4-64 "pages" per memory block

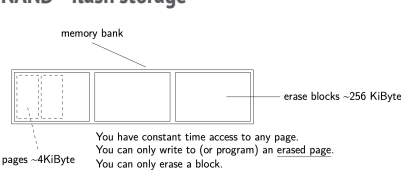
9

## Architecture of a solid-state drive




10

## NAND - flash storage




11

## NAND - flash storage. Lifespan

the cells of NAND-flash modules are wearing off

Each cell has a maximum number of P/E cycles (Program/Erase), after which the cell is considered defective.

The life span of a Samsung 850 PRO with 1TB:

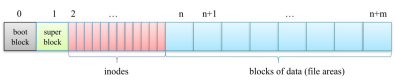
$$\frac{write\ cycles\ \times capacity}{SSD-factor\ \times data\ per\ year} = \frac{3000\times1000\ GB}{5\times1750\ GB} \sim 343\ \text{years}$$

12

## Anatomy of a hybrid hard drive


13

## The internal structure of the UNIX disk




- **Boot block** contains the programs that load the resident part of the Unix operating system.
- **The superblock** contains general information about the file system on disk
- **I-nodes**

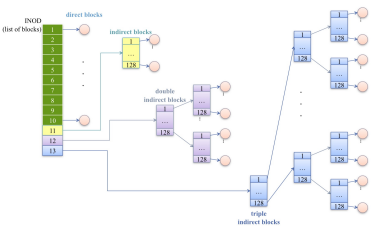
14

## The i-node mechanism

| Name | Explanation |
|---|---|
| mode | file access permissions |
| link count | the number of directories that contain references to this i-number |
| user ID | Owner user ID (UID) |
| group ID | Owner group ID (GID) |
| size | the number of bytes in the file (file length) |
| access time | the time the file was last accessed |
| mod time | the time the file was last modified |
| inode time | the time the i-node was last modified |
| block list | address list of the first few file blocks |
| indirect list | references to the rest of the blocks belonging to the file |

15

## List of blocks in a *i-node*




16

## Internal structures of Windows and DOS file systems

**FAT** (File Alocation Table)

**NTFS** (NT File System)

17

## The internal structure of the FAT disk


- **Boot area** - contains information that is used every time the operating system loads
- **File allocation table (FAT)**
- **The root directory**
- **The area of other files or directories**

18

## Boot area

| Offset | Content | |
|---|---|---|
| +00h | JMP address. Jump to the operating system load routine | |
| +03h | Manufacturer name and version | |
| +0Bh | Number of bytes per sector | |
| +0Dh | Number of sectors per cluster | BPB table |
| +0Eh | Number of reserved sectors (before FAT) | |
| +10h | Number of FATs | |
| +11h | The maximum number of entries in the root directory | BIOS |
| +13h | Total number of sectors | Parameter |
| +15h | Media descriptor | Bloc |
| +16h | The number of sectors in a FAT | |
| +18h | Number of sectors per track | |
| +1Ah | Number of read/write heads | |
| +1Ch | Number of hidden sectors | |
| +1Eh | Boot routine | |

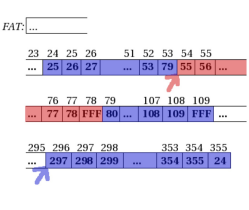
19

## FAT table

- contains disk space allocation information for each file;
- contains as many entries as clusters (groups of sectors) exist on disk;
- each input can handle 12 bits (FAT12), 16 bits (FAT or FAT16) or 32 bits (FAT32).

20

## Example of FAT




21

## Structure of directories

| Offset | Content |
|---|---|
| +00h | File name |
| +08h | Filename extension |
| +0Bh | Attributes |
| +0Ch | 10 bytes booked |
| +16H | The time of the last modification of the file |
| +18h | The date of the last modification of the file |
| +1Ah | The number of the first cluster occupied by the file |
| +1Ch | File size (4 bytes) |

22

## The internal structure of the NTFS disk


- **Boot area** (Partition Boot Sector)
- **MFT** (Master File Table)

**Simple design**: "*everything on the volume is a file and everything in a file is an attribute*"

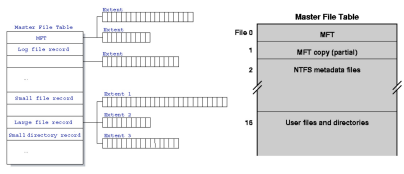
23

## NTFS boot area

| Byte Offset | Field Length | Field Name |
|---|---|---|
| 0x00 | 3 bytes | Jump Instruction |
| 0x03 | LONGLONG | OEM ID |
| 0x0B | 25 bytes | BPB |
| 0x24 | 48 bytes | Extended BPB |
| 0x54 | 426 bytes | Bootstrap Code |
| 0x01FE | WORD | End of Sector Marker |

24

## Master File Table




25

## List of attributes of a file in NTFS

| Attribute Type | Description |
|---|---|
| Standard Information | Includes information about creation time, change, etc. |
| The Attributes list | List of all attributes that are not resident. |
| File Name | |
| Security Descriptor | Represents the file owner |
| *Data* | The contents of the actual file; data stream |
| *Object ID* | A unique volume identifier |
| Logging stream | Data stream with saved operations in the log file |
| *Reparse Point* | Used for mounting volumes. |
| *Index Root* | Used for directories |
| *Index Allocation* | --------- ǁ --------- |
| *Bitmap* | --------- ǁ --------- |
| *Volume Information* | Contains version information |
| *Volume Name* | Contains the volume name |

26

## NTFS system files

| File/th> | Name | MFT index | Description |
|---|---|---|---|
| MFT | $Mft | 0 | Contains MFT |
| MFT2 | $MftMirror | 1 | Copy image of MFT |
| Log file | $LogFile | 2 | List of Transactions Made |
| Volume | $Volume | 3 | Volume information |
| Attribute Definitions | $AttrDef | 4 | A table of attribute names and descriptors |
| Root index file | $ | 5 | The root directory |
| Cluster bitmap | $Bitmap | 6 | Clusters used in volume |
| The boot sector | $Boot | 7 | The code by which the SO is loaded |
| Bad cluster | $BadCluster | 8 | An index of broken clusters |
| Security files | $Secure | 9 | Security information about each file |
| Upcase table | $Upcase | 10 | Convert small case characters into upper case ones |
| NTFS extension | $Extend | 11 | Used for NTFS extensions (ex: $Quota, $ObjId, $Reparse, $UsnJrnl) |
| | | 12-15 | Reserved |

27

# The End