BÁO CÁO THỰC TẬP 2 NGÀY 15/04/2025

1. Phần 1 ngày 15/04/2025:

Đặc điểm SAS và SATA:

* SAS:

+ Có tốc độ và hiệu năng cao hơn (12Gb/s hoặc 24Gb/s).

+ Độ bền cao, hỗ trợ hoạt động liên tục.

+ Tương thích ngược với ổ SATA (có thể dùng ổ SATA bên bộ điều khiển SAS).

* SATA:

+ Có tốc độ và hiệu năng thấp hơn so với SAS (Thường là 3Gb/s và tối đa là 6Gb/s).

+ Giá thành rẻ.

+ Dung lượng lưu trữ lớn hơn.

Ứng dụng:

* SAS thường được dùng cho doanh nghiệp/máy chủ vì có hiệu năng và tốc độ đọc cao, nhưng chi phí cao và dung lượng ổ đĩa thường nhỏ hơn so với SATA.
* SATA thường được sử dụng cho mục đích cá nhân (laptop, PC, lưu trữ dữ liệu cá nhân, …).

1. Phần 2 ngày 15/04/2025:

* Đơn vị lưu trữ dữ liệu trong ổ cứng:
* Sector: đơn vị lưu trữ nhỏ nhất trong ổ cứng, có kích thước cố định (512 bytes (truyền thống) hoặc 4KB (hiện đại, Advance Format).

+ Sector truyền thống: 512 bytes + 40 byte ECC (Error Correction Code/Mã sửa sai). Giữ mỗi sector có 1 khoảng trắng nhỏ được gọi là Sync/DAM để chứa dữ liệu về vị trí dữ liệu.

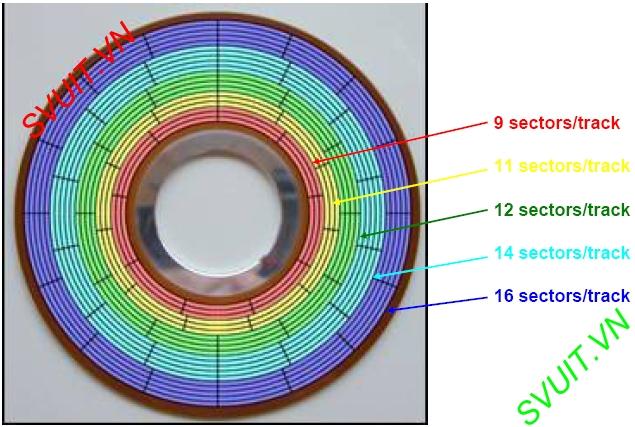
+ Sector hiện đại: 4KB + 100 byte ECC > Tiết kiệm được dung lượng cho ECC hơn và có khả năng sửa lỗi tốt hơn (do kích thước ECC lớn hơn) Sector truyền thống. Kích thước của Sync/DAM không đổi.

+ Cách hoạt động của ECC:

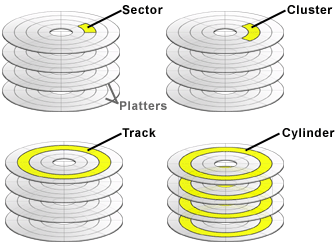
* Chứa 1 đoạn dữ liệu thừa được đặt vào cùng với dữ liệu gốc, sau đó được mã hóa bằng các thuật toán (Hamming code, Reed Solomon, BCH).
* Khi sector được đọc, sẽ đọc luôn cả ECC, nếu dữ liệu thừa trong sector sau khi mã hóa không khớp với ECC thì sẽ được xác định là lỗi
* Nếu dữ liệu nhỏ (tùy theo thuật toán), lỗi sẽ được xác định và xử lý, nếu lỗi vượt quá khả năng sửa, hệ thống sẽ báo lỗi (Uncorrectable error).
* Cluster: thông thường được định dạng là 8 Sector đứng liền kề nhau/Cluster (4KB), được gọi là Cluster (Window) hoặc inode (Linux).

+ Khi lưu trữ, bộ nhớ sẽ ưu tiên ghi dữ liệu vào các bộ Cluster còn trống gần nhất, trong trường hợp các cluster không liền kề nhau thì đó là sự phân mảnh dữ liệu > Gây giảm hiệu suất đọc dữ liệu vì các cluster không liền kề nhau.

* Track: gồm nhiều Cluster tập hợp lại thành 1 vòng tròn, mỗi vòng tròn trên 1 mặt của đĩa từ có kích thước khác nhau > dung lượng của các Track trên 1 mặt đĩa từ cũng khác nhau.



* Cylinder: Tập hợp của các track có cùng bán kính trên từng mặt của đĩa từ. Có đặc điểm là khi muốn đọc dữ liệu của các track trong cùng 1 cylinder thì Head không cần di chuyển vì đã đúng vị trí > tăng hiệu suất vì không có Seek Time.



* Head: Đầu đọc ghi, có công dụng đọc dữ liệu ở dạng từ hóa được ghi trên bề mặt của các mặt (platter) trong đĩa từ, có bao nhiêu mặt của đĩa từ hoạt động thì có bấy nhiêu Head được sử dụng.
* Hệ thống File (File System):
* File System dùng để tổ chức và quản lý dữ liệu/không gian lưu trữ, lưu trữ metadata, kiểm soát quyền (đọc/ghi), tăng hiệu suất khi truy cập
* Một số loại File System thông dụng hiện nay:

+ File Allocate Table (FAT): là file system lâu đời nhất, có 3 dòng FAT tiêu chuẩn là FAT12,FAT16, FAT32. Ngoài ra còn 1 số biến thể như exFAT, FATX, …

* FAT12: được dùng nhiều cho đĩa mềm (floppy disk).
* FAT16: được dùng cho các hệ điều hành cũ: MS-DOS, Win 95, …
* FAT32: File system được sử dụng phổ biến nhất của FAT, hỗ trợ phân vùng tối đa lên 2TB và file tối đa 4GB – 1 byte.
* exFAT: được phát triển để thay thế FAT32, được tối ưu tốt cho các thiết bị lưu trữ ngoài và các file có kích thước lớn, hỗ trợ phân vùng lên tới 128PB

+ New Technology File System (NTFS): là file system đang được dùng bởi các hệ điều hành Window mới (Win 7 ~ Win 11, Window Server, …).

* Có hỗ trợ ghi nhật ký (journaling), giúp phục hồi dữ liệu nhanh sau các sự cố bất ngờ.
* Hỗ trợ phân vùng tối đa 16 Exabyte, kích thước file cao tối đa 16 exabyte và không giới hạn số lượng file.
* Hỗ trợ phân quyền cho từng người hoặc nhóm người dùng (read, write, execute, modify, full control).
* Quản lý tốt metadata của các file trong ổ cứng bằng Master File Table (MFT).

+ Extended File System (ext): được dùng cho các hệ điều hành sử dụng Linux, có 4 phiên bản ext/ext2/ext3/ext4. Trong đó ext4 đang được sử dụng rộng rãi nhất tại thời điểm hiện tại,.

* Sử dụng journaling để ghi nhật ký.
* Hỗ trợ phân vùng tối đa lên tới 1 exabyte. Kích thước tối đa của 1 file là 16TB.
* Hỗ trợ số lượng inode lớn, cho phép lưu trữ hàng tỷ file.
* Extent: Thay vì ánh xạ từng block, ext4 dùng extent (các vùng liên tục) để lưu trữ dữ liệu, giảm phân mảnh và tăng tốc độ truy xuất.
* Delayed allocation: Trì hoãn việc phân bổ block cho đến khi dữ liệu được ghi ra ổ, cải thiện hiệu suất và giảm phân mảnh.
* Multiblock allocation: Phân bổ nhiều block cùng lúc, tối ưu cho việc ghi tệp lớn.
* Fast fsck: Công cụ kiểm tra hệ thống tệp (fsck) chạy nhanh hơn nhờ các cải tiến trong quản lý siêu dữ liệu.
* Phân quyền người dùng theo mô hình Unix (owner, group, other) với quyền Read/Write/Excute.
* Cấu trúc boot:
* MBR: lưu trữ tại sector đầu tiên của ổ cứng.

+ Bao gồm 3 thành phần chính:

* Bootstrap code: có kích thước 446 byte, chịu trách nhiệm boot giai đoạn đầu của thiết bị. Bao gồm các lệnh tìm kiếm phân vùng khởi động (bootable section) và chuyển quyền điều khiển sang mã khởi động của phân vùng đó.
* Partition Table (64 byte): Bảng phân vùng chứa thông tin về tối đa 4 phân vùng chính (primary partitions) trên ổ đĩa. Mỗi mục trong bảng phân vùng (16 byte) bao gồm:

Trạng thái phân vùng (1 byte): Cho biết phân vùng có phải là phân vùng khởi động (bootable) hay không (ví dụ: 0x80 là bootable, 0x00 là không).

Loại phân vùng (1 byte): Xác định kiểu phân vùng (ví dụ: FAT32, NTFS, Linux ext4, v.v.).

Địa chỉ bắt đầu (3 byte): LBA (Logical Block Addressing) hoặc CHS (Cylinder-Head-Sector) của sector đầu tiên của phân vùng.

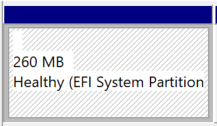
Kích thước phân vùng (4 byte): Số sector của phân vùng.

Địa chỉ kết thúc (3 byte): Thông tin về sector cuối của phân vùng (dạng CHS).

* Boot Signature (2 byte): Đây là dấu hiệu nhận biết MBR hợp lệ, luôn có giá trị 0x55AA. Nếu giá trị này không đúng, BIOS/UEFI sẽ coi MBR bị lỗi.

+ Quy trình hoạt động:

* Khi máy tính khởi động, BIOS hoặc UEFI sẽ đọc thông tin của sector đầu tiên.
* BIOS nạp bootstrap code từ MBR vào bộ nhớ (thường tại địa chỉ 0x7C00) và thực thi nó.
* Bootstrap code kiểm tra partition table để tìm phân vùng được đánh dấu là bootable.
* Sau đó, nó chuyển quyền điều khiển sang boot sector của phân vùng khởi động, nơi chứa mã khởi động của hệ điều hành (ví dụ: Windows Boot Manager hoặc GRUB cho Linux).
* Hệ điều hành tiếp tục được nạp vào bộ nhớ để hoàn tất quá trình khởi động.
* UEFI:
* UEFI hoạt động khác với BIOS/MBR ở chỗ nó không dựa vào một mã khởi động cố định trong MBR. Thay vào đó, UEFI sử dụng một quy trình khởi động linh hoạt hơn:
* EFI System Partition (ESP): UEFI yêu cầu một phân vùng đặc biệt gọi là ESP (thường định dạng FAT32, kích thước 100-300MB). Phân vùng này chứa các tệp khởi động (boot loader) của hệ điều hành, ví dụ:
  + Windows: bootmgfw.efi
  + Linux: GRUB hoặc systemd-boot (thường là grubx64.efi).

****

* NVRAM (Non-Volatile RAM): UEFI lưu trữ các biến khởi động (boot entries) trong NVRAM, bao gồm thông tin về vị trí của các tệp khởi động trong ESP. Điều này cho phép UEFI biết chính xác nơi để tìm boot loader.
* Quy trình khởi động:
  + Khi bật máy, UEFI khởi động và thực hiện POST (Power-On Self-Test) để kiểm tra phần cứng.
  + UEFI đọc các biến khởi động từ NVRAM để xác định thứ tự khởi động.
  + UEFI boot file .efi (boot loader) từ ESP vào bộ nhớ.
  + Boot loader tiếp tục boot hệ điều hành (Windows, Linux, v.v.).
* Định dạng GPT: Chiếm 33 sector đầu của ổ cứng, bao gồm:

+ Protective MBR (Sector 0, 512 byte): Để tương thích ngược với các công cụ hoặc hệ thống chỉ hỗ trợ MBR, GPT bắt đầu bằng một Protective MBR ở sector 0.

+ Primary GPT Header (Sector 1, 512 byte):

* Signature: Chuỗi EFI PART (8 byte) để xác nhận đây là GPT.
* Revision: Phiên bản của chuẩn GPT (thường là 1.0).
* Header Size: Kích thước của GPT Header (thường là 92 byte).
* CRC32 Checksum: Giá trị kiểm tra lỗi cho Header.
* Primary LBA: Vị trí của Primary GPT Header (thường là sector 1).
* Backup LBA: Vị trí của Backup GPT Header (thường ở cuối ổ đĩa).
* First Usable LBA: Sector đầu tiên có thể dùng cho phân vùng (thường là sector 34).
* Last Usable LBA: Sector cuối cùng có thể dùng cho phân vùng.
* Disk GUID: Một mã định danh duy nhất cho ổ đĩa.
* Partition Entry LBA: Vị trí bắt đầu của Partition Entry Array (thường là sector 2).
* Number of Partition Entries: Số lượng mục phân vùng (thường là 128).
* Size of Partition Entry: Kích thước mỗi mục phân vùng (thường là 128 byte).
* Partition Table CRC32: Giá trị kiểm tra lỗi cho Partition Entry Array.

+ Partition Entry Array (Sector 2 > 33): Đây là nơi chứa thông tin về các phân vùng trên ổ đĩa. Mỗi mục phân vùng (Partition Entry) có kích thước 128 byte và bao gồm:

* Partition Type GUID: Mã định danh loại phân vùng (ví dụ: phân vùng Windows NTFS, Linux ext4, hoặc EFI System Partition).
* Unique Partition GUID: Mã định danh duy nhất cho phân vùng.
* Starting LBA: Sector bắt đầu của phân vùng.
* Ending LBA: Sector kết thúc của phân vùng.
* Attributes: Các thuộc tính của phân vùng (ví dụ: bootable, read-only).
* Partition Name: Tên phân vùng (tối đa 36 ký tự Unicode).

+ Backup GPT (Cuối ổ đĩa): Lưu lại 1 bản sao của GPT Header và Partiton Entry Array để tăng độ an toàn.

+ Usable Disk Space (Phần còn lại, sector đầu thường là sector 34, sector cuối tùy vào dung lượng ổ đĩa): Chứa dữ liệu, folder, file của người dùng, …

* Ngoài ra còn có 1 số định dạng như APM (Apple Partition Map) của Apply, BDS Disklabel thường được dùng trong các hệ điều hành BSD, …