**BÁO CÁO THỰC TẬP P1 NGÀY 16/04/2025**

1. **Quy trình một máy tính Boot hệ thống từ lúc nhấn nút khởi động nguồn tới khi vào hệ điều hành (16/04/2025):**
2. **Cấp nguồn và khởi động:**

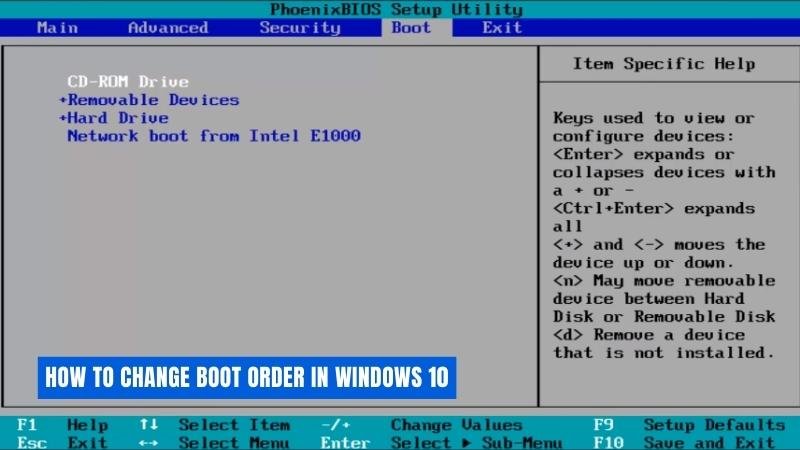
* Khi người dùng nhấn nút nguồn, bộ nguồn sẽ gửi dòng điện áp chính để khởi động mainboard.
* Mạch VRM trên mainboard sẽ cấp nguồn VCORE cho cho CPU và gửi tín hiệu **PowerGood** đến chipset nam. Tín hiệu PowerGood xác nhận bộ nguồn đang hoạt động tốt.
* Mạch **clockgen** hoạt động và cung cấp **xung clock** cho các thành phần trên **mainboard**.
* Khi nhận được xung clock và tín hiệu PowerGood, **chipset nam** hoạt động và tạo tín hiệu reset hệ thống, sau đó mở lệnh cho **chipset bắc** hoạt động.
* Chipset bắc sau đó tạo ra tín hiệu **RESET** đến CPU để khởi động CPU. Khi IC đếm thời gian của mainboard nhận được tín hiệu PowerGood, sẽ ngừng gửi tín hiệu RESET đến CPU.
* Mainboard gửi tín hiệu đến bộ cấp nguồn (PSU). **PSU** nhận tín hiệu, cấp nguồn điện thích hợp và trả tín hiệu về mainboard.
* Sau khi mainboard nhận được tín hiệu nguồn điện ổn định, nó sẽ thực hiện cấp nguồn cho các bộ phận như core, clock, chipset, bộ nhớ, …
* Một số hệ thống con, bao gồm các hệ thống con tự trị (như Intel ME hoặc AMD PSP), có thể khởi động trước bộ xử lý chính.
* Các hệ thống **AMD-based**:
* Khi hệ thống khởi chạy, PSP là thành phần đầu tiên được kích hoạt. Nó chạy mã nguồn khởi động từ BOOT ROM trên chip. BOOT ROM trên chip chứa boot code để khởi động PSP và bắt đầu quy trình khởi tạo hệ thống.
* Sau khi khởi động từ BOOT ROM trên chip, PSP tìm **Bảng phần mềm nhúng** trong BOOT ROM ngoài chip. Bảng phần mềm nhúng chứa thông tin về **cách thực thi PSP Firmware.** PSP nạp và thực thi phần mềm cơ sở này. Phần mềm cơ sở PSP chứa các chức năng nâng cao hơn, bao gồm xác minh bảo mật, quản lý phần cứng và chuẩn bị cho các bước khởi động tiếp theo.
* PSP phân tích Bảng danh mục PSP để tìm địa chỉ của **ABL Stages (AMD Boot Loader Stages)**. ABL Stages là các đoạn mã phần mềm được thiết kế để thực hiện các bước khởi tạo cụ thể (thiết lập bộ nhớ, khởi tạo phần cứng, nạp BIOS, …). ABL thực hiện từng giai đoạn theo thứ tự, đảm bảo hệ thống được khởi tạo đúng cách.
* Một trong những nhiệm vụ chính của ABL là khởi tạo bộ nhớ chính DRAM. Bao gồm cấu hình bộ điều khiển bộ nhớ (memory controller) và kiểm tra tính toàn vẹn của DRAM. Sau khi DRAM sẵn sàng, ABL Stages tìm BIOS Image trong BOOT ROM ngoài chip và nạp vào DRAM.
* Các hệ thống **Intel-based**:
* Khi hệ thống khởi chạy, chipset là thành phần đầu tiên được kích hoạt. Truy cập BOOT ROM (thường là chip SPI Flash ngoài chip và tìm Intel Flash Descriptor). Intel Flash Descriptor là một cấu trúc dữ liệu trong BOOT ROM, chứa thông tin về cách tổ chức nội dung của BOOT ROM, bao gồm vị trí của CSME firmware, BIOS region, và các thành phần khác.
* Sau khi xác định vị trí của **CSME firmware** thông qua Intel Flash Descriptor, chipset sao chép firmware này từ BOOT ROM vào bộ nhớ nội bộ của chipset (thường là SRAM hoặc bộ nhớ dành riêng cho Intel ME).
* Chipset ánh xạ (map) **BIOS region** (phần chứa hình ảnh BIOS/UEFI) từ BOOT ROM vào bộ nhớ hệ thống (thường là DRAM hoặc một vùng bộ nhớ tạm thời). Điều này cho phép CPU chính truy cập và thực thi BIOS image sau khi nó được kích hoạt. Tại thời điểm này, DRAM có thể chưa được khởi tạo đầy đủ, nên chipset có thể sử dụng các kỹ thuật như **Cache-as-RAM** (CAR) để tạo một vùng bộ nhớ tạm thời nếu cần. (Không giống như hệ thống AMD, nơi DRAM được khởi tạo sớm bởi PSP và ABL).
* Chipset hoặc Intel ME tìm **Firmware Interface Table (FIT)** trong BOOT ROM. FIT chứa thông tin về vị trí của các microcode updates (bản cập nhật mã vi mô) dành cho CPU. Các microcode updates được nạp vào CPU để sửa lỗi phần cứng, cải thiện hiệu suất, hoặc tăng cường bảo mật. **(Microcode updates phải được áp dụng mỗi lần hệ thống khởi động, vì chúng không được lưu vĩnh viễn trong CPU mà chỉ được nạp tạm thời trong quá trình khởi động)**.

1. **BIOS/UEFI thực thi:**

* **CPU chính bắt đầu hoạt động:**
* Sau khi tín hiệu RESET được gỡ bỏ và DRAM sẵn sàng, CPU chính được kích hoạt và bắt đầu thực thi mã từ BIOS image/UEFI firmware đã được nạp vào DRAM hoặc ánh xạ vào bộ nhớ từ trước.
* Điểm bắt đầu thường nằm tại **vector reset** (địa chỉ bộ nhớ cố định, ví dụ: 0xFFFFFFF0 trên Intel) để CPU nhảy đến mã khởi động của BIOS/UEFI.
* **POST (Power-On Self-Test)**:
* BIOS/UEFI thực hiện **POST**, một chuỗi kiểm tra phần cứng để đảm bảo các thành phần chính (CPU, RAM, GPU, bàn phím, ổ cứng, v.v.) hoạt động bình thường.
* Nếu phát hiện lỗi, BIOS có thể phát tín hiệu beep code hoặc hiển thị mã lỗi trên màn hình (nếu có thiết bị hiển thị).
* POST cũng khởi tạo các thiết bị ngoại vi cơ bản (như bàn phím, chuột) để chuẩn bị cho quá trình tiếp theo.
* **Khởi tạo phần cứng bổ sung**:
* BIOS/UEFI cấu hình các thành phần phần cứng khác, bao gồm:
  + **Bộ điều khiển bộ nhớ** (memory controller) để tối ưu hóa hiệu suất RAM.
  + **Bus hệ thống** (PCIe, USB, SATA) để giao tiếp với các thiết bị ngoại vi.
  + **Card đồ họa** để hiển thị giao diện người dùng (nếu có).
* UEFI hiện đại có thể hiển thị giao diện đồ họa để người dùng cấu hình các thiết lập (như overclock, boot order).

1. **Tìm và nạp Bootloader:**

* **Xác định thiết bị khởi động (Boot Device)**:
* BIOS/UEFI kiểm tra **boot order** được cấu hình trong **firmware**, sau đó thực hiện truy cập và tìm boot sector hoặc partition table (MBR hoặc GPT) trên thiết bị đó để xác định vị trí của bootloader.



* **Nạp bootloader:**
* **Bootloader** là một chương trình nhỏ nằm trong boot sector của thiết bị khởi động (GRUB, Window Boot Manager).
* BIOS/UEFI nạp bootloader vào **RAM** và **chuyển quyền điều khiển cho nó.**
* Trên hệ thống UEFI, bootloader **thường là một tệp EFI (ví dụ: \EFI\BOOT\BOOTX64.EFI trên ổ đĩa EFI System Partition)**.

1. **Bootloader khởi động hệ điều hành:**

* **Khởi tạo môi trường hệ điều hành:** Bootloader cấu hình các tham số cần thiết cho hệ điều hành:
* **Chế độ CPU** (Real Mode, Protected Mode, hoặc Long Mode).
* Bảng mô tả phần cứng (như **ACPI tables**).
* Vị trí của **kernel** trong bộ nhớ.

Bootloader nạp **kernel** (nhân hệ điều hành) và các mô-đun khởi động cần thiết (như initrd/initramfs trên Linux) vào RAM.

* **Chuyển giao quyền điều khiển**:
* Bootloader **chuyển quyền điều khiển** sang kernel của hệ điều hành.
* Kernel bắt đầu thực thi, khởi tạo các trình điều khiển (drivers) cho phần cứng, thiết lập hệ thống tệp (file system), và nạp các dịch vụ hệ thống.

1. **Hệ điều hành khởi động hoàn tất**

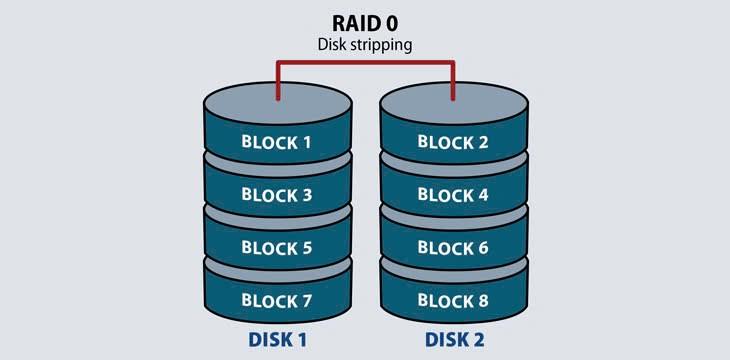
* **Khởi tạo hệ thống**: Kernel khởi tạo các thành phần cốt lõi như:
  + - **Quản lý bộ nhớ** (memory management).
    - **Quản lý tiến trình** (process management).
    - **Hệ thống tệp** (file system) và các thiết bị lưu trữ.

Các trình điều khiển phần cứng (như GPU, mạng, âm thanh) được nạp để đảm bảo hệ thống hoạt động đầy đủ.

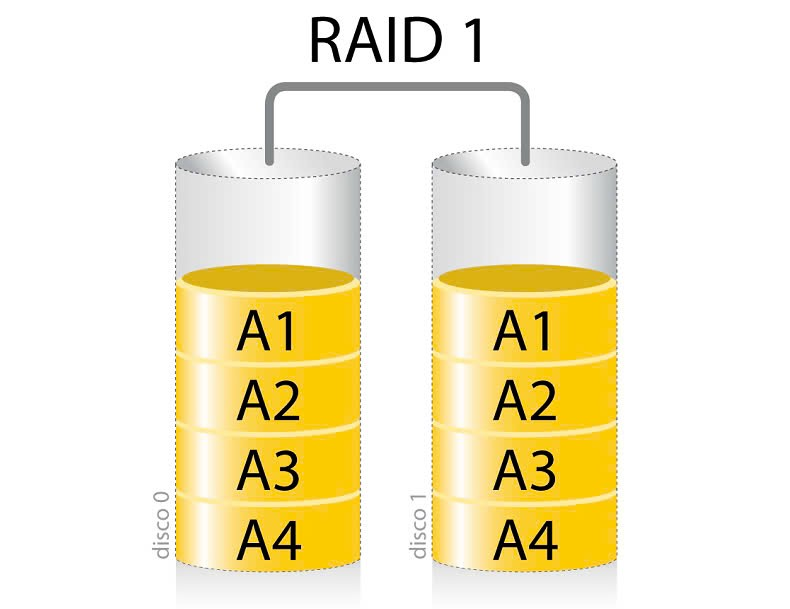
* **Nạp giao diện người dùng**:
  + Hệ điều hành khởi động **shell** hoặc **môi trường đồ họa** (như Windows Desktop, Linux Desktop Environment).
  + Người dùng thấy màn hình đăng nhập hoặc màn hình chính của hệ điều hành, đánh dấu việc hoàn tất quá trình khởi động.

1. **Báo cáo ngày 16/04/2025:**
2. **RAID (Redundant Array of Independent Disks):**

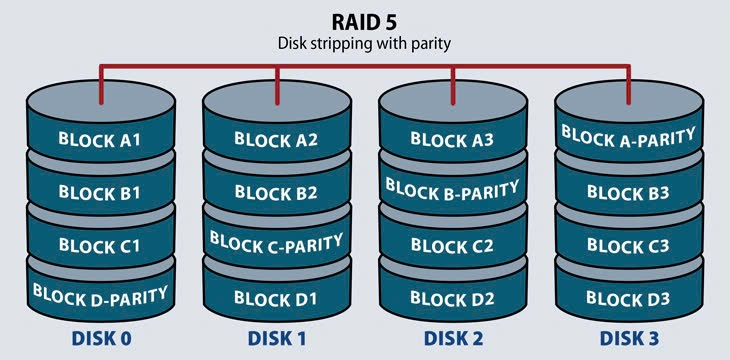
* RAID là công nghệ lưu trữ dữ liệu trong nhiều ổ đĩa để tăng hiệu suất, độ tin cậy hoặc kết hợp cả 2. RAID được thiết kế để bảo vệ trước lỗi ổ đĩa và cải thiện tốc độ truy cập.
* Các ổ đĩa vật lý được kết hợp lại thành **ổ phi vật lý (Logical disk)** dựa trên thuật toán của RAID.
* Các mức RAID phổ biến và cách hoạt động:
* **RAID 0 (Striping):**
* **Cách hoạt động:** Dữ liệu được chia nhỏ và lưu trữ trên nhiều ổ đĩa.
* **Ưu điểm:** tăng tốc độ đọc ghi do xử lý bằng nhiều luồng song song.
* **Nhược điểm:** Nếu một ổ hỏng, toàn bộ dữ liệu sẽ hỏng.
* **Ứng dụng:** phù hợp các yêu cầu cần hiệu suất cao (edit video, xử lý dữ liệu, …).



* **RAID 1 (Mirroring):**
* **Cách hoạt động:** Dữ liệu được mirror thành bản sao và lưu trên 2 hoặc nhiều ổ đĩa.
* **Ưu điểm:** độ tin cậy cao, khi 1 ổ hỏng, dữ liệu vẫn còn ở ổ còn lại.
* **Nhược điểm:** Dung lượng lưu trữ bị giảm (do chia nhỏ để mirror).
* **Ứng dụng:** phù hợp các hệ thống cần độ chính xác và tin cậy cao (máy chủ, lưu trữ, …).



* **RAID 5 (Striping with Parity):**
* **Cách hoạt động:** Dữ liệu được ghi vào cùng với thông tin kiểm tra lỗi (parity) được phân phối trên ít nhất 3 ổ đĩa. Nếu một ổ hỏng, có thể restore lại từ phần dữ liệu parity.
* **Ưu điểm:** Kết hợp giữa tốc độ đọc cao (stripping) và khả năng dự phòng tốt (parity), chỉ mất dung lượng 1 ổ cho parity.
* **Nhược điểm:** Tốc độ ghi chậm hơn do tính toán parity, nếu 2 ổ hỏng cùng lúc, dữ liệu sẽ mất.
* **Ứng dụng:** Phổ biến trong máy chủ và hệ thống lưu trữ doanh nghiệp.



* Ngoài ra còn 1 số mức RAID như **RAID 6**, **RAID 10**, …
* **Nguyên lý hoạt động chung**
* RAID sử dụng phần cứng (bộ điều khiển RAID) hoặc phần mềm để quản lý cách dữ liệu được ghi và đọc từ các ổ đĩa.
* Dữ liệu được phân chia (striping), sao chép (mirroring), hoặc kết hợp với thông tin parity để đảm bảo an toàn và hiệu suất.
* Khi một ổ đĩa bị lỗi, tùy thuộc vào mức RAID, hệ thống có thể khôi phục dữ liệu từ các ổ còn lại hoặc tiếp tục hoạt động mà không mất dữ liệu.
* **Lưu ý khi sử dụng RAID**
* RAID không thay thế sao lưu (backup) vì không bảo vệ dữ liệu trước các lỗi logic, xóa nhầm, hoặc hỏng đồng thời nhiều ổ.
* Lựa chọn mức RAID phụ thuộc vào nhu cầu về hiệu suất, độ tin cậy, và chi phí.
* Cần ổ đĩa có dung lượng và tốc độ tương đương để RAID hoạt động hiệu quả.

1. **Format ổ đĩa:**

* **Format (định dạng)** ổ đĩa là quá trình chuẩn bị ổ đĩa để lưu trữ dữ liệu bằng cách:
* Tạo hệ thống tệp (file system) như NTFS, FAT32, exFAT, hoặc ext4 để hệ điều hành có thể quản lý dữ liệu.
* Xóa toàn bộ dữ liệu hiện có trên ổ đĩa (hoặc đánh dấu là không thể truy cập).
* Thiết lập cấu trúc lưu trữ, bao gồm bảng phân bổ tệp (**file allocation table**) và các siêu dữ liệu (metadata) cần thiết.
* Lựa chọn Unit Size: lựa chọn kích thước của cluster trên ổ đĩa để tối ưu hóa hiệu suất và chống phân mảnh.
* (Tùy chọn) Kiểm tra và đánh dấu các bad sector (khu vực lỗi) trên ổ đĩa.
* Quá trình format có thể được thực hiện bằng hai phương pháp chính: quick format và **format thường (full format).**
* **HDD**: Có 2 phương pháp format là quick format và full format:
* **Quick format:** là quá trình format nhanh, chỉ xóa dữ liệu ở **bảng phân bố file (MFT)** như metadata, con trỏ. Dữ liệu vẫn còn được **lưu lại ở dạng raw** và **có thể phục hồi** bằng phần mềm chuyên dụng.
* **Full Format:** là quá trình **format toàn diện ổ đĩa**, không chỉ xóa MFT mà có khả năng kiểm tra toàn bộ bề mặt HDD và đánh dấu bad sector, một số hệ thống tệp như NTFS, … có khả năng **ghi đè toàn bộ dữ liệu (zero-fill)** nhằm mục đích xóa dữ liệu triệt để hơn.

Quick format có tốc độ cao hơn và thường được dùng cho những ổ đĩa cần dùng ngay. Full format và quá trình dọn dẹp hoàn toàn ổ đĩa và thường có tốc độ chậm hơn quick format.

* **SSD:**
* **Quick Format:**
  + Xóa bảng phân bổ tệp và siêu dữ liệu của hệ thống tệp cũ.
  + Tạo hệ thống tệp mới với các cấu trúc siêu dữ liệu (như Master File Table trong NTFS).
  + Không ghi đè dữ liệu cũ; dữ liệu vẫn tồn tại trên các ô nhớ flash nhưng được đánh dấu là không thể truy cập.
* **Full Format:**
  + Tương tự quick format, xóa siêu dữ liệu và tạo hệ thống tệp mới.
  + Trên Windows (với NTFS), full format ghi số 0 (zero-fill) lên toàn bộ ổ SSD để kiểm tra tính toàn vẹn của các ô nhớ và xóa dữ liệu cũ.
  + Không kiểm tra bad sector như HDD, vì SSD quản lý lỗi ô nhớ thông qua cơ chế **wear leveling** và **bad block management** nội bộ.
* Lựa chọn Unit Size:
* **4KB**: Lựa chọn mặc định, phù hợp cho hầu hết các trường hợp (hệ điều hành, ổ đa năng, SSD, lưu trữ tệp nhỏ hoặc hỗn hợp).
* **8KB hoặc lớn hơn (16KB, 64KB)**: Phù hợp cho ổ đĩa lưu trữ tệp lớn (video, game, sao lưu), máy chủ, hoặc NAS.