**1) Quản lý bộ nhớ (Memory Management)**

**Mô tả & mục tiêu:** HĐH quản lý RAM: cấp phát/thu hồi bộ nhớ cho tiến trình, bảo đảm các tiến trình không ghi đè nhau, tối ưu sử dụng bộ nhớ vật lý và hỗ trợ bộ nhớ ảo (virtual memory). Hệ thống xử lý phân trang (paging), phân đoạn (segmentation), tái phân bố (compaction) và hoán đổi/trang vào-trang ra (swap/page-in, page-out).

**Cách hoạt động (chi tiết):**

* Khi một tiến trình khởi chạy, HĐH tạo không gian địa chỉ cho tiến trình đó (gồm code, data, heap, stack).
* HĐH theo dõi các khung (frames) vật lý và các trang (pages) ảo; bảng trang (page table) ánh xạ địa chỉ ảo → địa chỉ vật lý.
* Nếu tiến trình truy cập trang không có trong RAM (page fault), HĐH sẽ dừng tiến trình tạm thời, tải trang từ ổ đĩa vào RAM (page swap) và cập nhật bảng trang.
* Các thuật toán quản lý bộ nhớ (LRU, FIFO, Clock) quyết định trang nào bị đẩy ra khi cần không gian.
* Bộ nhớ ảo cho phép mỗi tiến trình có không gian địa chỉ lớn hơn RAM vật lý, làm tăng tính đa nhiệm.

**Ví dụ thực tế:**

* Mở nhiều tab trình duyệt (Chrome/Edge): mỗi tab là tiến trình hoặc luồng — khi RAM đầy, hệ điều hành (và kết hợp với trình duyệt) sẽ bị "swap" một số trang xuống ổ SSD, dẫn đến hiện tượng máy chậm (page-in/page-out).
* Ứng dụng lớn như IDE (Visual Studio, Android Studio) yêu cầu nhiều bộ nhớ; HĐH dùng phân trang để cấp cho ứng dụng mà không làm sập hệ thống. [GeeksforGeeks+1](https://www.geeksforgeeks.org/operating-systems/memory-management-in-operating-system/?utm_source=chatgpt.com)

**Tóm tắt ngắn:** Quản lý bộ nhớ đảm bảo tiến trình có không gian địa chỉ cần thiết, bảo toàn tính cô lập giữa tiến trình, và tối ưu dùng RAM bằng cơ chế phân trang/hoán đổi — giúp hệ thống đa nhiệm hoạt động trơn tru. [GeeksforGeeks](https://www.geeksforgeeks.org/operating-systems/memory-management-in-operating-system/?utm_source=chatgpt.com)

**2) Quản lý thiết bị nhập / xuất (I/O & Device Management)**

**Mô tả & mục tiêu:** HĐH quản lý mọi thiết bị ngoại vi (bàn phím, chuột, ổ đĩa, card mạng, máy in…). Nó cung cấp giao diện phần mềm (device drivers), cơ chế đệm/buffering, spooling (đặc biệt cho máy in) và sử dụng DMA để truyền dữ liệu hiệu quả. Mục tiêu: ẩn chi tiết phần cứng cho ứng dụng, đồng bộ và xử lý lỗi I/O. [GeeksforGeeks+1](https://www.geeksforgeeks.org/operating-systems/device-management-in-operating-system/?utm_source=chatgpt.com)

**Cách hoạt động (chi tiết):**

* **Device driver:** mã đặc hiệu cho từng thiết bị; HĐH gọi driver để gửi lệnh/nhận dữ liệu.
* **Buffering & Caching:** dữ liệu I/O được lưu tạm vào bộ đệm để giảm số lần truy cập vật lý và tăng tốc.
* **Spooling:** ví dụ in ấn — lệnh in được gửi vào hàng đợi (print spool), hệ thống in xử lý từng job để tránh block ứng dụng gửi lệnh.
* **DMA (Direct Memory Access):** cho phép thiết bị chuyển dữ liệu trực tiếp vào/ra RAM mà không cần CPU thao tác byte-by-byte — giảm tải CPU.
* HĐH cũng quản lý ưu tiên truy cập thiết bị (lock, queue), xử lý lỗi thiết bị và tải/khởi động driver khi cần.

**Ví dụ thực tế:**

* Khi bạn bấm “Print” trên Word, hệ điều hành đẩy job vào print spooler; bạn có thể tiếp tục làm việc trong khi máy in xử lý hàng đợi. Nếu driver máy in lỗi, Windows sẽ báo lỗi và cung cấp tùy chọn cài lại driver.

**Tóm tắt ngắn:** Quản lý I/O ẩn phức tạp phần cứng qua driver, tối ưu truyền dữ liệu qua buffering/DMA và thực thi spooling/queuing để ứng dụng không bị chặn bởi I/O chậm.

**3) Cung cấp giao diện người dùng (User Interface — GUI & CLI)**

**Mô tả & mục tiêu:** HĐH cung cấp cách người dùng tương tác với máy: GUI (Graphical User Interface — cửa sổ, biểu tượng, chuột) hoặc CLI (Command-Line Interface — dòng lệnh). Giao diện làm đơn giản thao tác, quản lý cửa sổ, và chuyển tiếp lệnh người dùng tới hệ thống.

**Cách hoạt động (chi tiết):**

* **GUI:** thành phần quản lý cửa sổ (window manager / desktop environment) xử lý các sự kiện chuột/phím, vẽ giao diện trên màn hình, quản lý vị trí/cỡ cửa sổ và compositing.
* **CLI:** trình thông dịch lệnh (shell) nhận chuỗi ký tự, phân tích cú pháp và gọi hệ thống (system calls) để thực thi chương trình hoặc thao tác tệp.
* Giao diện tích hợp với quản lý tiến trình (khởi chạy/tắt ứng dụng), quản lý tệp (mở/lưu), và quản lý thiết bị (nhập liệu/hiển thị). **Ví dụ thực tế:**
* **GUI:** Windows Explorer trên Windows hoặc Finder trên macOS — khi bạn kéo-thả file, đó là chức năng desktop environment phối hợp với hệ thống tệp để di chuyển dữ liệu.
* **CLI:** Mở Terminal trên Linux và gõ ls / grep — shell gửi yêu cầu tới kernel để đọc thư mục và trả kết quả lên màn hình.

**Tóm tắt ngắn:** Giao diện người dùng (GUI/CLI) là lớp mà người dùng thấy — chịu trách nhiệm nhận lệnh/sự kiện và chuyển đổi chúng thành thao tác ở cấp hệ thống, làm trải nghiệm máy tính thân thiện và hiệu quả.

**4) Quản lý hệ thống tệp (File System Management)**

**Mô tả & mục tiêu:** HĐH tổ chức, lưu trữ và truy xuất dữ liệu trên lưu trữ lâu dài (HDD/SSD). Bao gồm cấu trúc thư mục, phân quyền, metadata (timestamps, kích thước), và tính năng như journaling để bảo toàn dữ liệu khi có lỗi. Hệ thống tệp khác nhau (NTFS, ext4, FAT32) có đặc điểm riêng.

**Cách hoạt động (chi tiết):**

* HĐH giữ **siêu dữ liệu** (inode trong UNIX-like) — thông tin file: chủ sở hữu, quyền, con trỏ tới block vật lý.
* **Bộ nhớ đệm cho hệ thống tệp (file cache):** giảm truy vấn đĩa bằng cách giữ dữ liệu nóng trong RAM.
* **Journaling:** trước khi thay đổi cấu trúc tệp, hệ thống ghi một bản ghi (journal); nếu bị crash, HĐH dùng journal để phục hồi trạng thái nhất quán.
* **Quản lý phân vùng & mount:** HĐH mount (gắn) phân vùng/thiết bị để trình bày chúng trong cây thư mục chung

**Ví dụ thực tế:**

* Khi bạn nhấn “Save” trong Word, hệ điều hành xử lý tạo/ghi file: đặt entry trong thư mục, ghi block dữ liệu lên ổ và cập nhật metadata (kích thước, thời gian). Trên hệ thống ext4/NTFS, journaling giúp tránh file bị hỏng khi mất điện đột ngột.
* Khi bạn cắm USB, hệ điều hành mount phân vùng và hiển thị nó dưới dạng ổ đĩa (Windows: E:, Linux: /media/usb) để bạn thao tác file. **Tóm tắt ngắn:** Hệ thống tệp do HĐH quản lý tổ chức dữ liệu lên ổ đĩa, quản lý metadata/quyền, tối ưu truy cập bằng cache và đảm bảo nhất quán qua cơ chế như journaling.

**5) Quản lý tiến trình (Process Management)**

**Mô tả & mục tiêu:** HĐH tạo/huỷ/nạp/chuyển trạng thái tiến trình, lập lịch CPU (scheduling), quản lý đồng bộ và giao tiếp giữa tiến trình (IPC), cùng xử lý context switching. Mục tiêu: phân phối CPU hợp lý, giữ hiệu suất và đáp ứng tính ưu tiên/độ trễ.

**Cách hoạt động (chi tiết):**

* **Tạo tiến trình:** khi chạy một chương trình, HĐH tạo PCB (Process Control Block) lưu trạng thái (registers, PC, priority, tài nguyên dùng).
* **Lập lịch (scheduling):** thuật toán (Round Robin, Priority, SJF, Multilevel Queue...) quyết định tiến trình nào được cấp CPU và trong bao lâu (time quantum). Round Robin phổ biến cho hệ tương tác.
* **Context switch:** khi HĐH chuyển CPU từ tiến trình A sang B, nó lưu trạng thái A và nạp trạng thái B — đó là context switching.
* **Đồng bộ & IPC:** mutex, semaphore, pipe, message queue, shared memory dùng để tránh race condition và trao đổi dữ liệu giữa tiến trình.
* **Trạng thái tiến trình:** new → ready → running → waiting (blocked) → terminated. HĐH quản lý chuyển trạng thái dựa trên I/O, time slice, tín hiệu

**Ví dụ thực tế:**

* Khi hệ thống chạy nhiều ứng dụng (browser, music player, editor), HĐH chia CPU theo time slice; nếu bạn mở game nặng, scheduler có thể ưu tiên tiến trình game (priority) để giảm lag.
* Khi một tiến trình chờ đọc dữ liệu từ mạng (I/O blocked), scheduler cho CPU chạy tiến trình khác; khi dữ liệu đến, tiến trình từ blocked → ready → chạy tiếp.

**Tóm tắt ngắn:** Quản lý tiến trình điều phối CPU giữa nhiều chương trình, xử lý tạo/hủy, context switch, lập lịch và giao tiếp giữa tiến trình để hệ thống đa nhiệm hoạt động hiệu quả.