**Bài 1:**

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

Hệ điều hành phân bổ và quản lý bộ nhớ RAM cho các tiến trình, sử dụng kỹ thuật như paging và swapping để tạo bộ nhớ ảo, đảm bảo không xung đột và tối ưu hóa sử dụng. Ví dụ thực tế: Trong Windows, khi chạy nhiều ứng dụng như Chrome và Photoshop, OS tự động swap dữ liệu thừa ra ổ cứng để giải phóng RAM, tránh treo máy.

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

Hệ điều hành quản lý thiết bị ngoại vi qua driver, xử lý yêu cầu nhập/xuất, interrupt và buffering để giao tiếp hiệu quả giữa phần cứng và phần mềm. Ví dụ thực tế: Khi kết nối USB, OS nhận diện và cài driver tự động, cho phép đọc/ghi dữ liệu mà không cần người dùng can thiệp thủ công.

**3. Cung cấp giao diện người dùng (User Interface)**

Hệ điều hành cung cấp GUI (Graphical User Interface) hoặc CLI (Command Line Interface) để người dùng tương tác, thực thi lệnh và truy cập tài nguyên. Ví dụ thực tế: Trong macOS, giao diện desktop với icon và menu cho phép kéo-thả file dễ dàng, thay vì gõ lệnh như trong terminal.

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

Hệ điều hành tổ chức, lưu trữ và truy cập file/directory trên ổ đĩa, quản lý quyền truy cập, sao lưu và khôi phục dữ liệu. Ví dụ thực tế: Trong Linux, lệnh "ls" liệt kê file, OS đảm bảo quyền đọc/ghi để tránh xóa nhầm, như hệ thống NTFS trong Windows quản lý phân vùng.

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

Hệ điều hành tạo, lập lịch và chấm dứt tiến trình, hỗ trợ đa nhiệm bằng cách phân bổ CPU thời gian (scheduling) và xử lý đồng bộ. Ví dụ thực tế: Trong Android, OS chạy đồng thời app nhắn tin và chơi nhạc, sử dụng priority scheduling để ưu tiên app foreground mà không làm chậm hệ thống.