



## Chương II: Cấu Trúc Hệ Điều Hành

---

- Các thành phần của hệ điều hành
- Các dịch vụ hệ điều hành cung cấp
- Lời gọi hệ thống (System call)
- Các chương trình hệ thống (system programs)
- Cấu trúc hệ thống
- Máy ảo (virtual machine)



## 2.1. Các thành phần của hệ điều hành

### 2.1.1. Quản lý quá trình (process management)

- Quá trình (hay tiến trình – **process**) là gì?
- Quá trình khác chương trình ở điểm gì?
- Một quá trình cần các tài nguyên của hệ thống như CPU, bộ nhớ, file, thiết bị I/O,... để hoàn thành công việc.
- Các nhiệm vụ của thành phần
  - Tạo và hủy quá trình
  - Tạm dừng/thực thi tiếp (suspend/resume) quá trình
  - Cung cấp các cơ chế
    - đồng bộ hoạt động các quá trình (synchronization)
    - giao tiếp giữa các quá trình (interprocess communication)
    - không chế tắc nghẽn (deadlock)



## 2.1. Các thành phần của hệ điều hành

### 2.1.2. Quản lý bộ nhớ chính

- Bộ nhớ chính là **trung tâm** của các thao tác, xử lý
- Để nâng cao hiệu suất sử dụng CPU, hệ điều hành cần quản lý bộ nhớ thích hợp
- Các nhiệm vụ của thành phần
  - Theo dõi, quản lý các vùng nhớ trống và đã cấp phát
  - Quyết định sẽ nạp chương trình nào khi có vùng nhớ trống
  - Cấp phát và thu hồi các vùng nhớ khi cần thiết



## 2.1. Các thành phần của hệ điều hành

### 2.1.3. Quản lý file (file management)

- Hệ thống file (file system)
  - *File*
  - *Thư mục*
- Các dịch vụ mà thành phần cung cấp
  - Tạo và xoá file/thư mục.
  - Các thao tác xử lý file/thư mục (mkdir, rename, copy, move, new,...)
  - “Ánh xạ” file/thư mục vào thiết bị lưu trữ thứ cấp tương ứng
  - Sao lưu và phục hồi dữ liệu



## 2.1. Các thành phần của hệ điều hành

### 2.1.4. Quản lý hệ thống I/O (I/O system management)

- Che dấu sự khác biệt của các thiết bị I/O trước người dùng
- Có chức năng
  - Cơ chế: *buffering, caching, spooling*
  - Cung cấp giao diện chung đến các trình điều khiển thiết bị (*device-driver interface*)
  - Bộ điều khiển các thiết bị (*device driver*) phần cứng.



## 2.1. Các thành phần của hệ điều hành

### 2.1.5. Quản lý hệ thống lưu trữ thứ cấp (secondary storage management)

- Bộ nhớ chính: kích thước nhỏ, là môi trường chứa tin không bền vững => cần hệ thống lưu trữ thứ cấp để lưu trữ bền vững các dữ liệu, chương trình
  - Phương tiện lưu trữ thông dụng là đĩa từ, đĩa quang
  - Nhiệm vụ của hệ điều hành trong quản lý đĩa
    - Quản lý không gian trống trên đĩa (*free space management*)
    - Cấp phát không gian lưu trữ (*storage allocation*)
    - Định thời hoạt động cho đĩa (*disk scheduling*)
- *Sử dụng thường xuyên => ảnh hưởng lớn đến tốc độ của cả hệ thống => cần hiệu quả*



## 2.1. Các thành phần của hệ điều hành

### 2.1.6. Hệ thống bảo vệ

Trong hệ thống cho phép nhiều user hay nhiều process diễn ra đồng thời:

- Kiểm soát quá trình người dùng đăng nhập/xuất và sử dụng hệ thống
- Kiểm soát việc truy cập các tài nguyên trong hệ thống
- Bảo đảm những *user/process* chỉ được phép sử dụng các tài nguyên dành cho nó
- Các nhiệm vụ của hệ thống bảo vệ
  - Cung cấp cơ chế kiểm soát đăng nhập/xuất (*login, log out*)
  - Phân định được sự truy cập tài nguyên *hợp pháp* và *bất hợp pháp* (*authorized/unauthorized*)
  - Phương tiện thi hành các chính sách (*enforcement of policies*)  
Chính sách: cần bảo vệ dữ liệu của ai đối với ai



## 2.1. Các thành phần của hệ điều hành

### 2.1.7. *Hệ thống thông dịch lệnh*

- Là giao diện chủ yếu giữa người dùng và OS  
Ví dụ: shell, mouse-based window-and-menu
- Khi user login  
*command line interpreter* (shell) chạy, và chờ nhận lệnh từ người dùng, thực thi lệnh và trả kết quả về.
- Các lệnh -> bộ điều khiển lệnh -> hệ điều hành
- Các lệnh có quan hệ với các việc:
  - Tạo, hủy, và quản lý quá trình, hệ thống
  - Kiểm soát I/O
  - Quản lý bộ lưu trữ thứ cấp
  - Quản lý bộ nhớ chính
  - Truy cập hệ thống file và cơ chế bảo mật





## 2.2. Các dịch vụ hệ điều hành cung cấp

---

- Thực thi chương trình
- Thực hiện các thao tác I/O theo yêu cầu của chương trình
- Các thao tác trên hệ thống file
  - Đọc/ghi hay tạo/xóa file
- Trao đổi thông tin giữa các quá trình qua hai cách:
  - Chia sẻ bộ nhớ (Shared memory)
  - Chuyển thông điệp (Message passing)
- Phát hiện lỗi
  - Trong CPU, bộ nhớ, trên thiết bị I/O (dữ liệu hư, hết giấy,...)
  - Do chương trình: chia cho 0, truy cập đến địa chỉ bộ nhớ không cho phép.



## 2.2. Các dịch vụ hệ điều hành cung cấp

---

Ngoài ra còn các dịch vụ giúp tăng hiệu suất của hệ thống:

- Cấp phát tài nguyên (resource allocation)  
Tài nguyên: CPU, bộ nhớ chính, tape drives,...  
OS có các routine tương ứng
- Kế toán (accounting)  
Nhằm lưu vết user để tính phí hoặc đơn giản để thống kê.
- Bảo vệ (protection)  
Hai quá trình khác nhau không được ảnh hưởng nhau  
Kiểm soát được các truy xuất tài nguyên của hệ thống
- An ninh (security)  
Chỉ các user được phép sử dụng hệ thống mới truy cập được tài nguyên của hệ thống (vd: thông qua username và password)



## 2.3. Lời gọi hệ thống (*System call*)

### *Dùng để giao tiếp giữa quá trình và hệ điều hành*

- Cung cấp giao diện giữa quá trình và hệ điều hành  
Vd: open, read, write file
- Thông thường ở dạng thư viện nhị phân (binary libraries) hay giống như các lệnh hợp ngữ.
- Trong các ngôn ngữ lập trình cấp cao, một số thư viện lập trình được xây dựng dựa trên các thư viện hệ thống (ví dụ Windows API, thư viện GNU C/C++ như glibc, glibc++,...)
- Ba phương pháp **truyền tham số** khi sử dụng system call
  - Qua thanh ghi
  - Qua một vùng nhớ, địa chỉ của vùng nhớ được gửi đến hệ điều hành qua thanh ghi
  - Qua stack



## 2.4. Các chương trình hệ thống

---

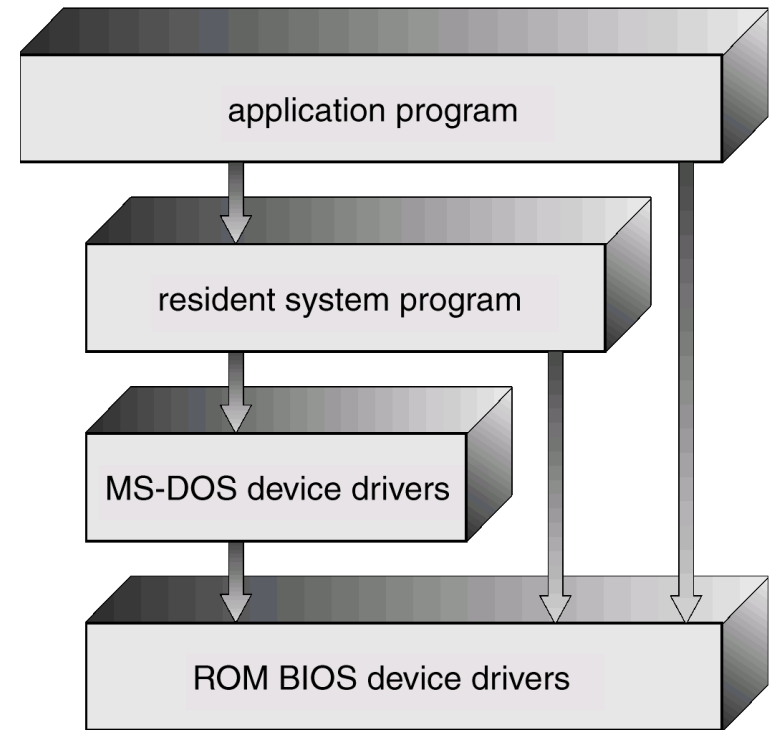
- *Chương trình hệ thống* (*system program*, phân biệt với *application program*) gồm
  - Quản lý hệ thống file: như *create, delete, rename, list*
  - Thông tin trạng thái: như *date, time*, dung lượng bộ nhớ trống
  - Soạn thảo file: như *file editor*
  - Hỗ trợ ngôn ngữ lập trình: như *compiler, assembler, interpreter*
  - Nạp, thực thi, giúp tìm lỗi chương trình: như *loader, debugger*
  - Giao tiếp: như *email, talk, web browser*
  - ...
- Người dùng chủ yếu làm việc thông qua các *system program* (không làm việc “trực tiếp” với các *system call*)



## 2.5. Cấu trúc hệ thống

### ➤ Cấu trúc đơn giản (monolithic)

- MS-DOS: khi thiết kế, do giới hạn về dung lượng bộ nhớ nên *không phân chia thành các module* (modularization) và chưa phân chia rõ chức năng giữa các phần của hệ thống.



Cấu trúc phân tầng của MS-DOS

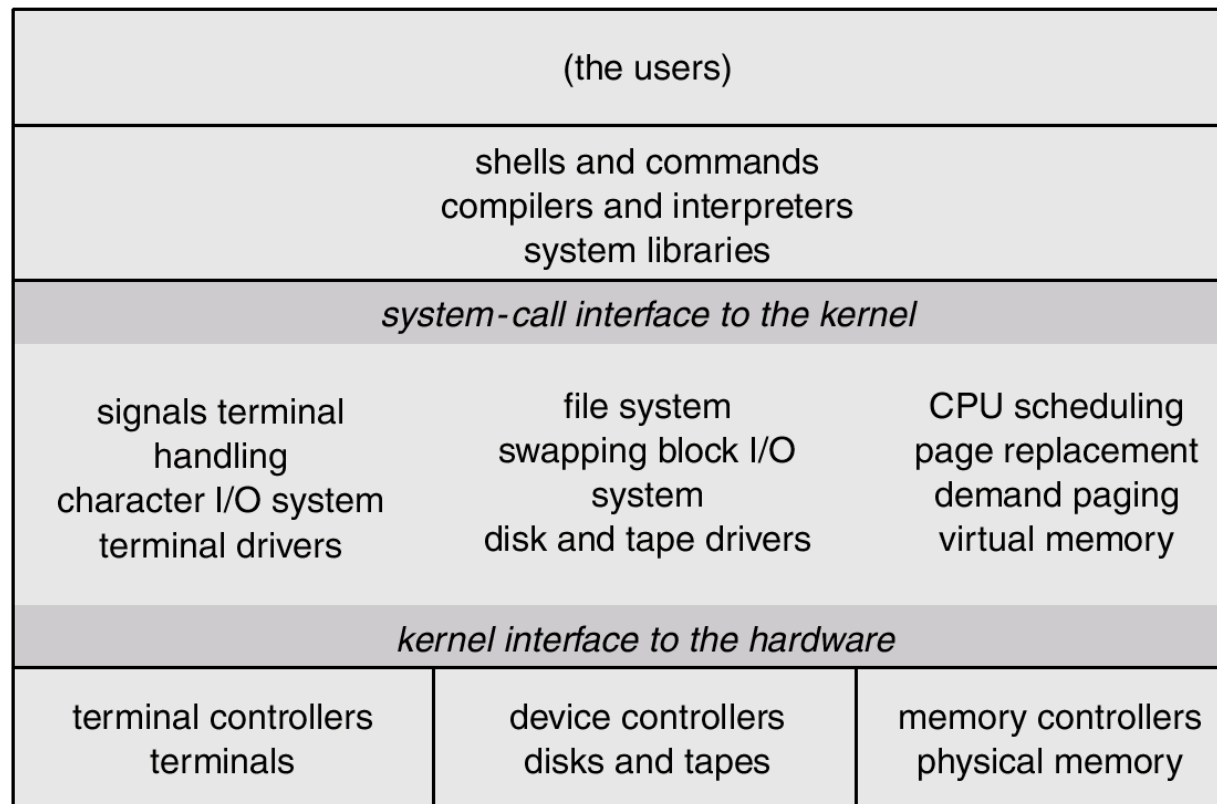


## 2.5. Cấu trúc hệ thống

### ➤ Cấu trúc đơn giản (monolithic)

UNIX: gồm hai phần có thể tách rời nhau

- Nhân (cung cấp file system, CPU scheduling, memory management, và một số chức năng khác) và system program





## 2.5. Cấu trúc hệ thống

- **Cấu trúc phân tầng:** HĐH được chia thành nhiều *lớp* (layer).
- Lớp dưới cùng: hardware
  - Lớp trên cùng là giao tiếp với user
  - Lớp trên chỉ phụ thuộc lớp dưới
  - Một lớp chỉ có thể gọi các hàm của lớp dưới và các hàm của nó được gọi bởi lớp trên
  - Mỗi lớp tương đương một đối tượng trừu tượng: cấu trúc dữ liệu + thao tác
  - Phân lớp có lợi ích gì? Gỡ rối (debugger, kiểm tra hệ thống, thay đổi chức năng)



## 2.5. Cấu trúc hệ thống

### ➤ Cấu trúc phân tầng:

- Lần đầu tiên được áp dụng cho HÐH THE (Technische Hogeschool Eindhoven)

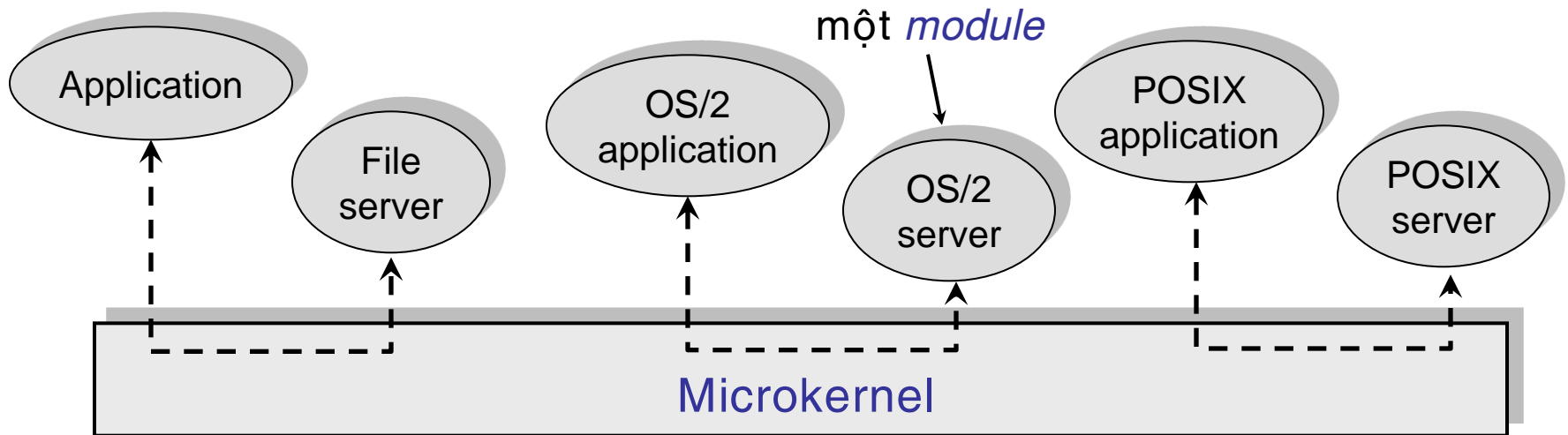
|       |                                 |
|-------|---------------------------------|
| Lớp 5 | user programm                   |
| Lớp 4 | Tạo buffer cho thiết bị I/O     |
| Lớp 3 | Device driver thao tác màn hình |
| Lớp 2 | Quản lý bộ nhớ                  |
| Lớp 1 | Lập lịch CPU                    |
| Lớp 0 | Phần cứng                       |





## 2.5. Cấu trúc hệ thống

- **Vi nhân:** phân chia module theo microkernel (CMU Mach OS, 1980)
- Chuyển một số chức năng của OS từ kernel space sang user space
- Thu gọn kernel => microkernel, microkernel chỉ bao gồm các chức năng tối thiểu như quản lý quá trình, bộ nhớ và cơ chế giao tiếp giữa các quá trình
- Giao tiếp giữa các module qua cơ chế truyền thông điệp





## 2.5. Cấu trúc hệ thống

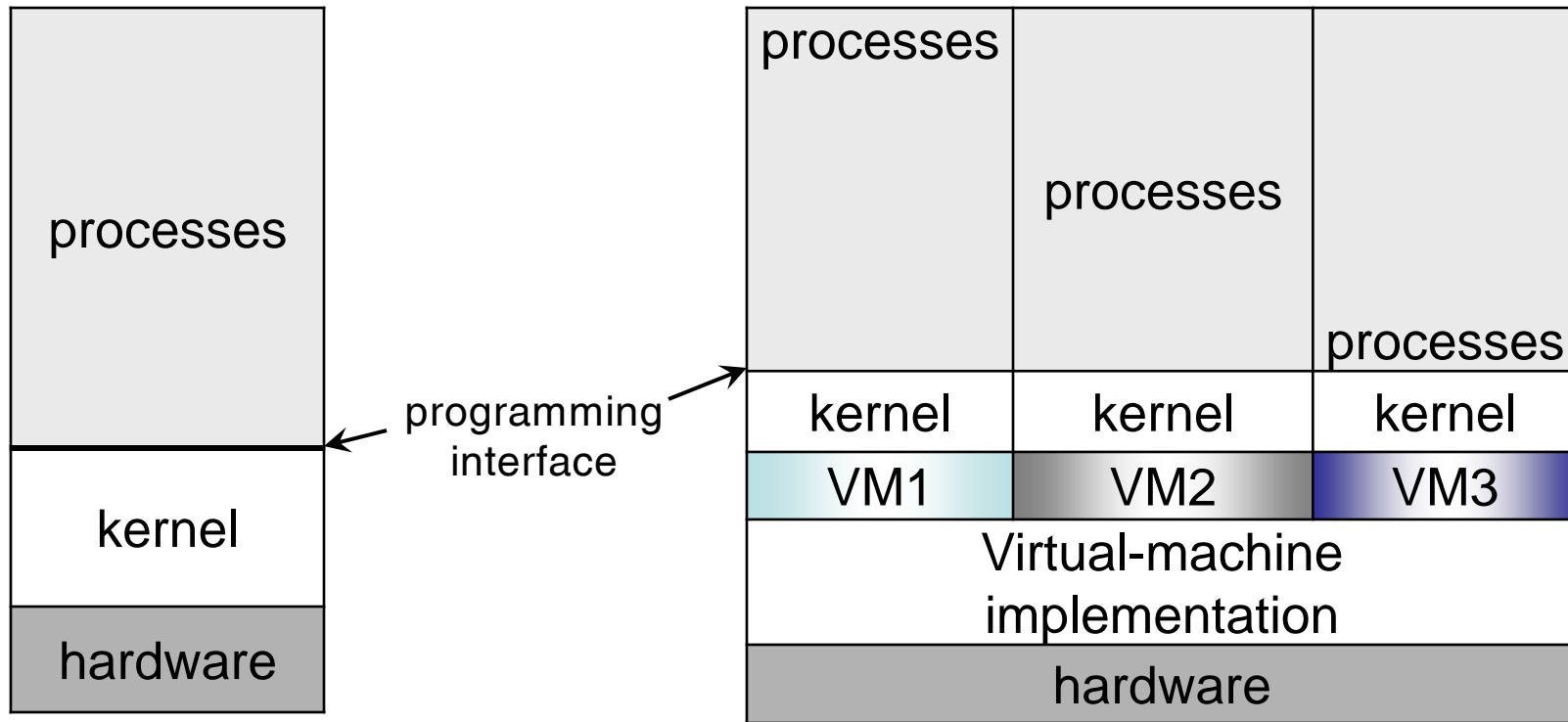
### ➤ Vi nhân:

- Lợi ích: dễ mở rộng HĐH
- Một số HĐH hiện đại sử dụng vi nhân:
  - + Tru64 UNIX (Digital UNIX trước đây): nhân Mach
  - + Apple MacOS Server : nhân Mach
  - + QNX – vi nhân cung cấp: truyền thông điệp, định thời CPU, giao tiếp mạng cấp thấp và ngắt phần cứng
  - + Windows NT: chạy các ứng dụng khác nhau win32, OS/2, POSIX (Portable OS for uniX)



## 2.6. Máy ảo

Từ OS layer đến *máy ảo* (virtual machine)



Non-virtual machine  
system model

Virtual machine system model



## 2.6. Máy ảo

### ➤ Hiện thực ý niệm VM

- Làm thế nào để thực thi một chương trình MS-DOS trên một hệ thống Sun với hệ điều hành Solaris ?
- 1. Tạo một *máy ảo Intel* bên trên hệ điều hành Solaris và hệ thống Sun
- 2. Các lệnh Intel (x86) được máy ảo Intel chuyển thành lệnh tương ứng của hệ thống Sun.

