

# Chương 1 (bổ sung)

## B. Cấu Trúc Hệ Điều Hành

- Các thành phần của hệ điều hành
- Các dịch vụ hệ điều hành cung cấp
- Giao diện giữa quá trình và hệ điều hành
- Các chương trình hệ thống (system program)
- Cấu trúc hệ điều hành
- Máy ảo (virtual machine)

# Các thành phần của hệ điều hành (1/7)

## ■ *Quản lý quá trình*

- Quá trình vs chương trình
- 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
  - ▶ Tạo và hủy quá trình
  - ▶ Tạm ngưng / tiếp tục thực thi (suspend/resume) quá trình
  - ▶ Cung cấp các cơ chế
    - đồng bộ hoạt động các quá trình
    - giao tiếp giữa các quá trình
    - xử lý deadlock

# Các thành phần của hệ điều hành (2/7)

## ■ *Quản lý bộ nhớ chính*

- Tùy thuộc kiến trúc máy tính
- Để có hiệu suất sử dụng CPU và thời gian đáp ứng tốt, hệ điều hành cần dùng giải thuật quản lý bộ nhớ thích hợp
- Các nhiệm vụ
  - ▶ 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

# Các thành phần của hệ điều hành (3/7)

## ■ *Quản lý file*

- Hệ thống file (file system)
  - ▶ *File*
  - ▶ *Thư mục*
- Các dịch vụ mà thành phần cung cấp
  - ▶ Tạo và xoá file/thư mục.
  - ▶ Các tác vụ xử lý file/thư mục (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.

# Các thành phần của hệ điều hành (4/7)

## ■ *Quản lý hệ thống I/O*

- Che dấu các đặc trưng riêng biệt của từng thiết bị I/O đối với user
- Có chức năng
  - ▶ Quản lý buffering, caching, spooling
    - Buffer: vùng nhớ để lưu dữ liệu khi chúng được truyền giữa hai thiết bị hay giữa thiết bị và ứng dụng
  - ▶ Cung cấp giao diện chung đến các trình điều khiển thiết bị (device-driver interface)
  - ▶ *Trình điều khiển thiết bị* cho mỗi chủng loại thiết bị phần cứng khác nhau.

# Các thành phần của hệ điều hành (5/7)

## ■ *Quản lý hệ thống lưu trữ thứ cấp*

- 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
  - ▶ Quản lý vùng trống
  - ▶ Cấp phát không gian lưu trữ (storage allocation)
  - ▶ Định thời đĩa (disk scheduling)

# Các thành phần của hệ điều hành (6/7)

## ■ *Hệ thống bảo vệ* (protection system)

Khi hệ thống cho phép nhiều user hay nhiều quá trình

- 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 chỉ những người dùng/quá trình *đủ quyền hạn* mới được phép sử dụng các tài nguyên tương ứng
- Các nhiệm vụ
  - ▶ Cung cấp cơ chế kiểm soát đăng nhập/xuất (login, logout)
  - ▶ 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

# Các thành phần của hệ điều hành (7/7)

## ■ *Trình 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
- Liên hệ chặt chẽ với các thành phần khác của hệ điều hành để thực thi các yêu cầu của người dùng
- Các nhóm lệnh trình thông dịch lệnh để
  - ▶ Tạo, hủy, xem thông tin quá trình, hệ thống
  - ▶ Điều khiển truy cập I/O
  - ▶ Quản lý, truy cập hệ thống lưu trữ thứ cấp
  - ▶ Quản lý, sử dụng bộ nhớ
  - ▶ Truy cập hệ thống file
  - ▶ ...



# Các dịch vụ hệ điều hành cung cấp (1/2)

- *Một số dịch vụ chủ yếu* mà người dùng hay chương trình cần
  - Thực thi chương trình
  - Thực hiện các tác vụ I/O do yêu cầu của chương trình
  - Các tác vụ lên file
    - ▶ Đọc/ghi hay tạo/xóa file
  - Giao tiếp, trao đổi thông tin giữa các quá trình
    - ▶ Shared memory
    - ▶ Message passing
  - Phát hiện lỗi
    - ▶ Trên thiết bị I/O: dữ liệu hư, hết giấy,...
    - ▶ Chương trình ứng dụng: chia cho 0, truy cập đến địa chỉ bộ nhớ không được phép

# Các dịch vụ hệ điều hành cung cấp (2/2)

## ■ *Các chức năng khác*

- Cấp phát tài nguyên
  - ▶ Tài nguyên: tape drives,...
  - ▶ OS có các routine tương ứng
- Kế toán (accounting)
  - ▶ Ví dụ để tính phí

# Giao diện giữa quá trình và hệ điều hành

## ■ *System call*

- 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 library)
- 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
  - ▶ Truyền tham số qua **thanh ghi**
  - ▶ Truyền tham số thông qua một **vùng nhớ**, địa chỉ của vùng nhớ được gửi đến hệ điều hành qua thanh ghi
  - ▶ Truyền tham số qua **stack**

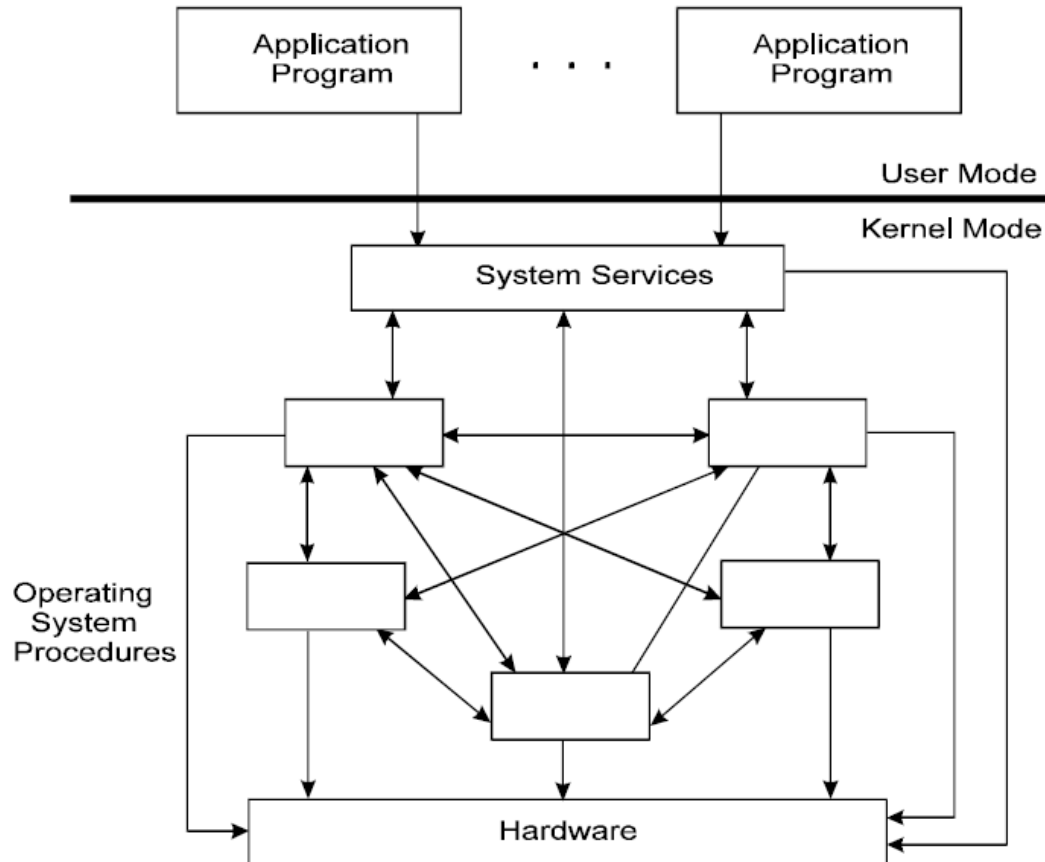
# Các chương trình hệ thống

- *Chương trình hệ thống* (phân biệt với application program) gồm
  - Quản lý file: như create, delete, rename, list
  - Thông tin trạng thái hệ thống: như 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 cuối chủ yếu làm việc thông qua các system program (không sử dụng “trực tiếp” các system call).

# Cấu trúc hệ điều hành (1/6)

## ■ Hệ thống *đơn* (monolithic)

- Các thành phần được tổ chức tùy tiện, các module có thể tùy tiện gọi module khác



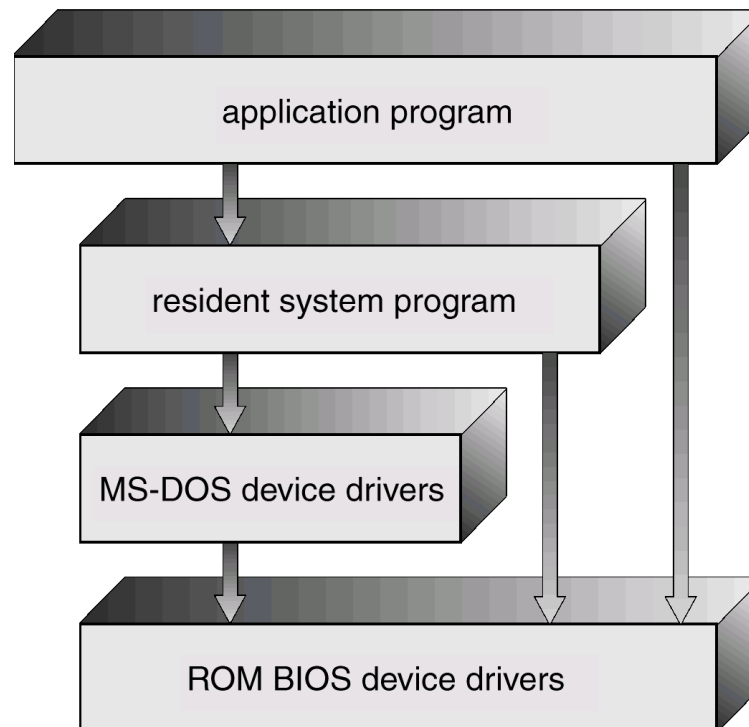
# Cấu trúc hệ điều hành (2/6)

- Trong hệ điều hành **được phân lớp** (**layered** operating system) mỗi lớp gồm có cấu trúc dữ liệu và thủ tục chỉ được gọi bởi các lớp ở mức cao hơn.
  - Lớp thấp nhất là phần cứng
  - Lớp cao nhất là giao diện người dùng
  - Lớp dưới che giấu cấu trúc dữ liệu và cách hiện thực của thủ tục đối với lớp cao hơn

# Cấu trúc hệ điều hành (3/6)

## ■ Hệ thống *đơn* (monolithic)

- MS-DOS: được thiết kế dưới điều kiện giới hạn về dung lượng bộ nhớ – Intel 8088, 1 MB bộ nhớ, không dual mode.
  - ▶ Nhìn lại, có thể *phân lớp*:



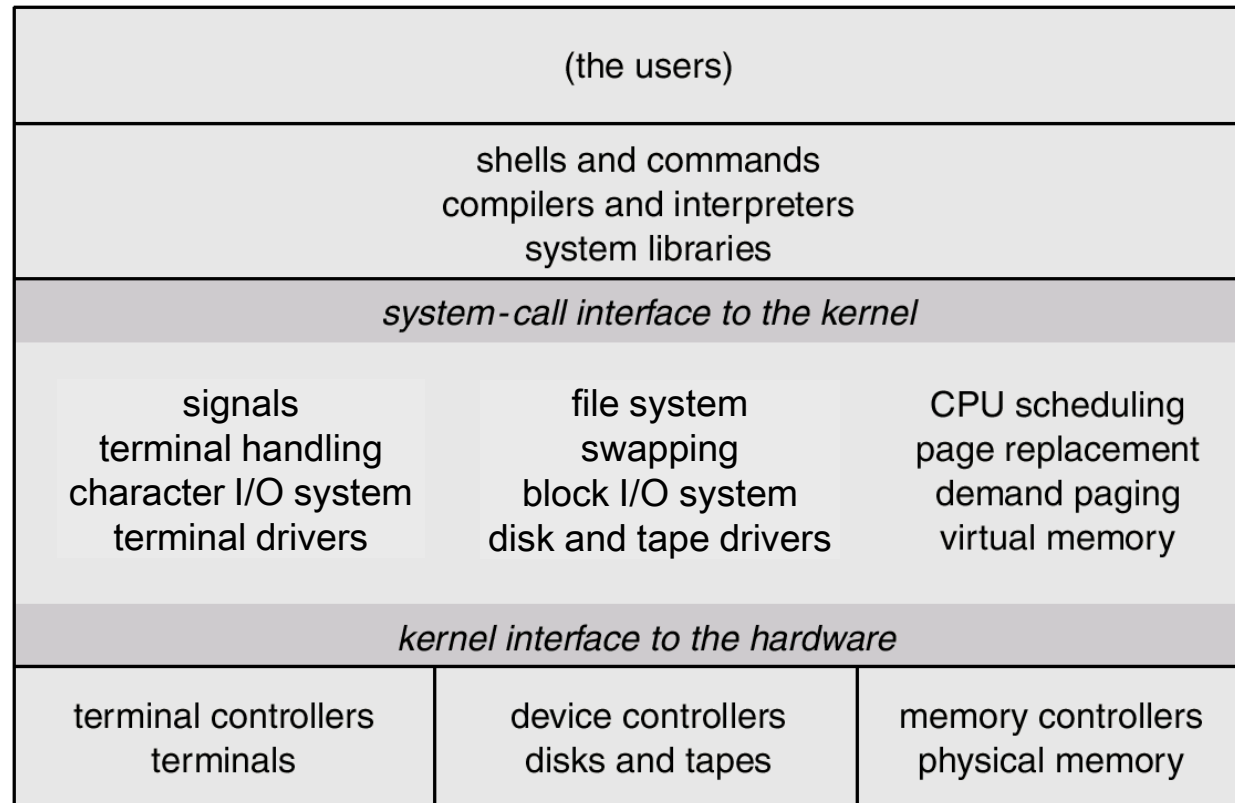
# Cấu trúc hệ điều hành (4/6)

## ■ Hệ thống đơn

- UNIX: gồm hai phần

- ▶ các system program và kernel (file system, CPU scheduling, memory management, và một số chức năng khác)

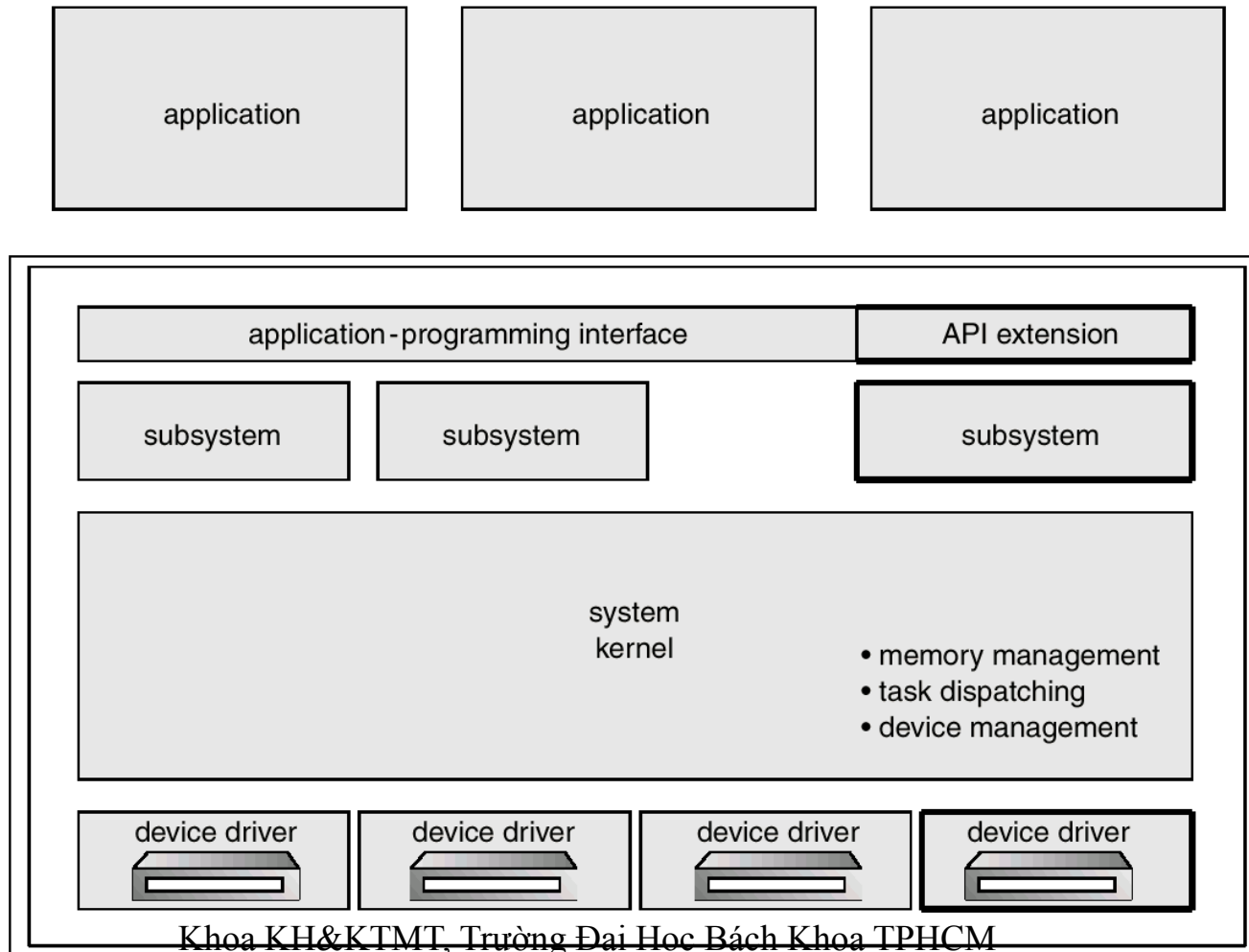
Nhìn lại, có thể phân lớp:





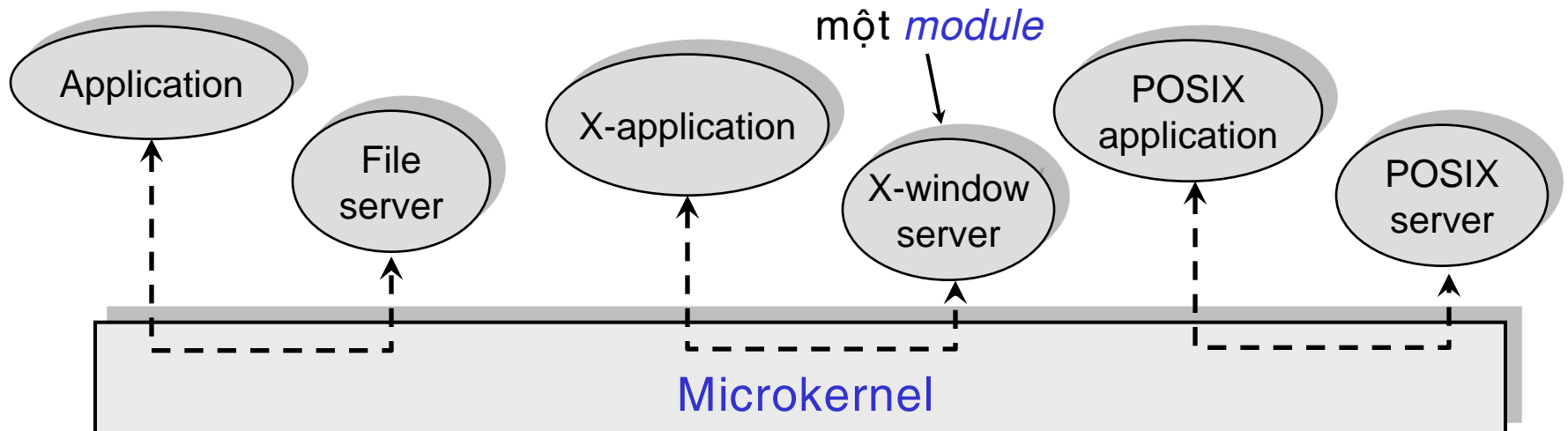
# Cấu trúc hệ điều hành (5/6)

- **Layered approach** -- thiết kế OS bằng cách phân chia module thành nhiều *lớp*. Vd: hệ điều hành OS/2



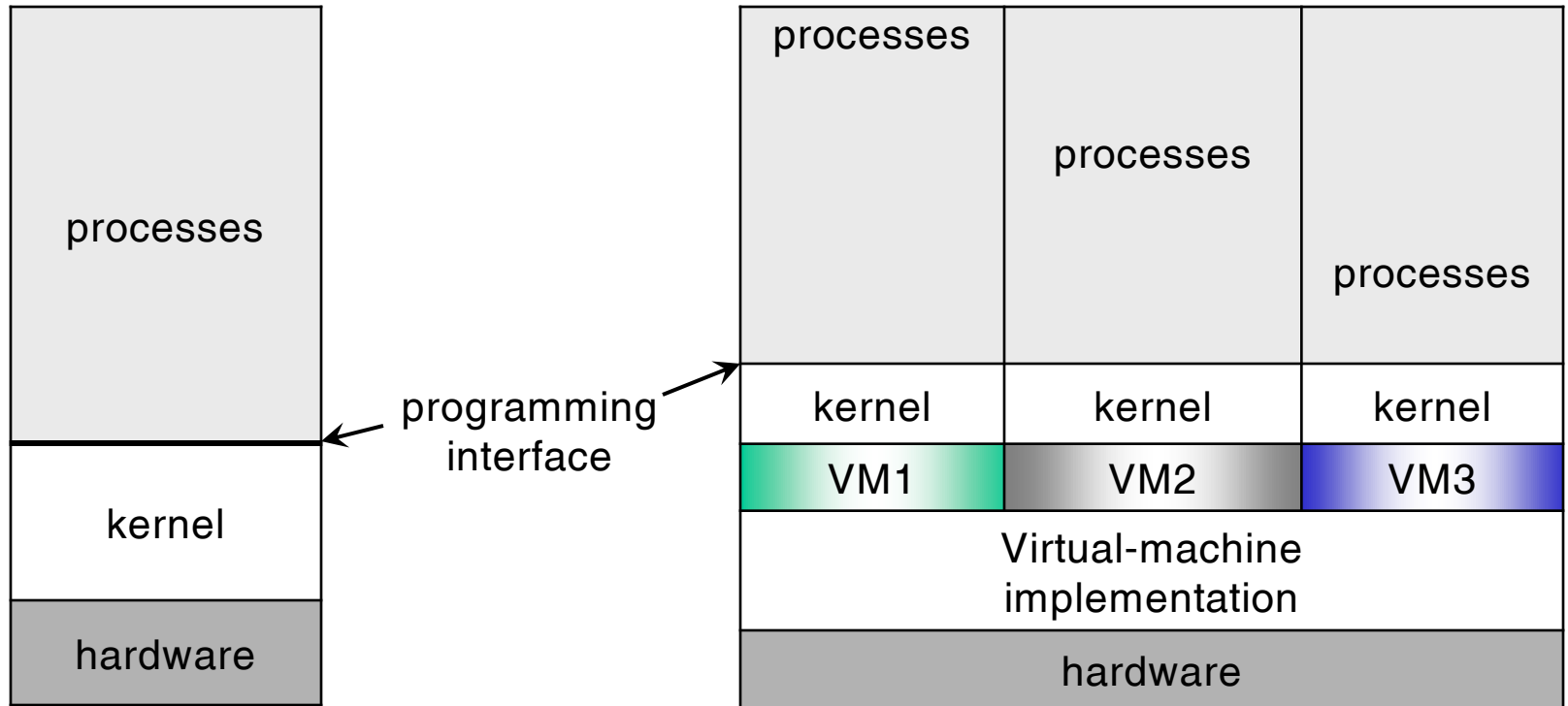
# Cấu trúc hệ điều hành (6/6)

- Thiết kế OS: phân chia module theo *microkernel* (CMU Mach OS, 1980)
  - Chuyển một số chức năng của OS từ kernel space sang user space (vd: file server)
  - Thu gọn kernel → microkernel, chỉ 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 dùng kỹ thuật *truyền thông điệp*



# Máy ảo (1/2)

- Từ OS layer đến *máy ảo* (VM, virtual machine)



Non-virtual machine  
system model

Virtual machine system model

# Máy ảo (2/2)

## ■ 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.

