

1. Tổng quan

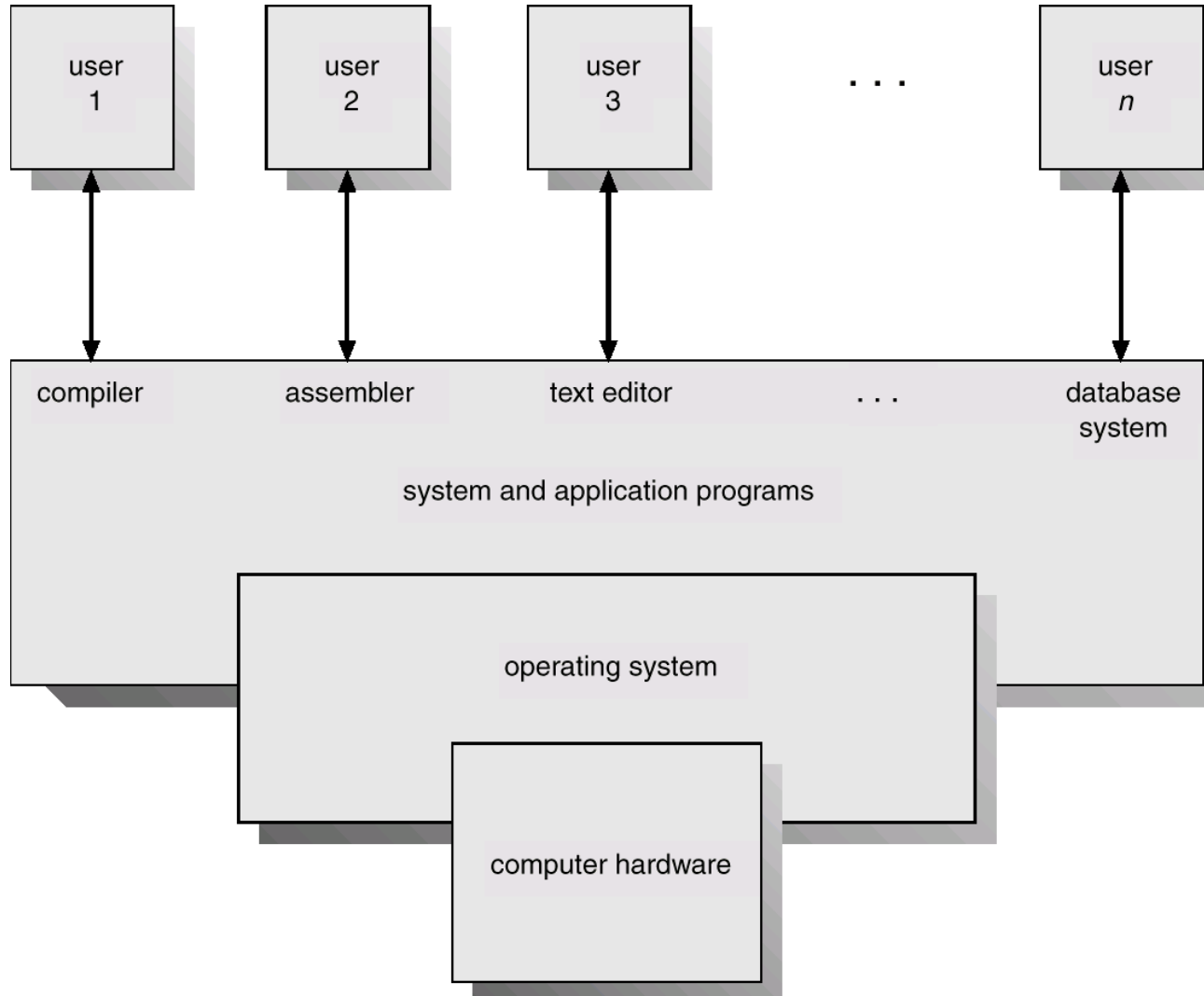
□ Giới thiệu

- Cấu trúc logic của hệ thống máy tính
- Định nghĩa hệ điều hành
- Các chức năng chính của hệ điều hành

□ tiến trình phát triển của hệ điều hành

- Máy tính lớn (mainframe systems)
 - Máy để bàn (desktop systems)
 - Đa xử lý (multiprocessor systems)
 - phân tán (distributed systems)
 - Thời gian thực (real-time systems)
 - Cầm tay (handheld systems)
-

Các thành phần của hệ thống máy tính



Định nghĩa

❑ Hệ điều hành là gì?

- *Chương trình* trung gian giữa phần cứng máy tính và người sử dụng, có chức năng *điều khiển phần cứng* và cung cấp các *dịch vụ cơ bản* cho các ứng dụng.

❑ Mục tiêu

- Giúp người dùng dễ dàng sử dụng hệ thống.
- Quản lý và cấp phát tài nguyên hệ thống một cách hiệu quả.

Người dùng

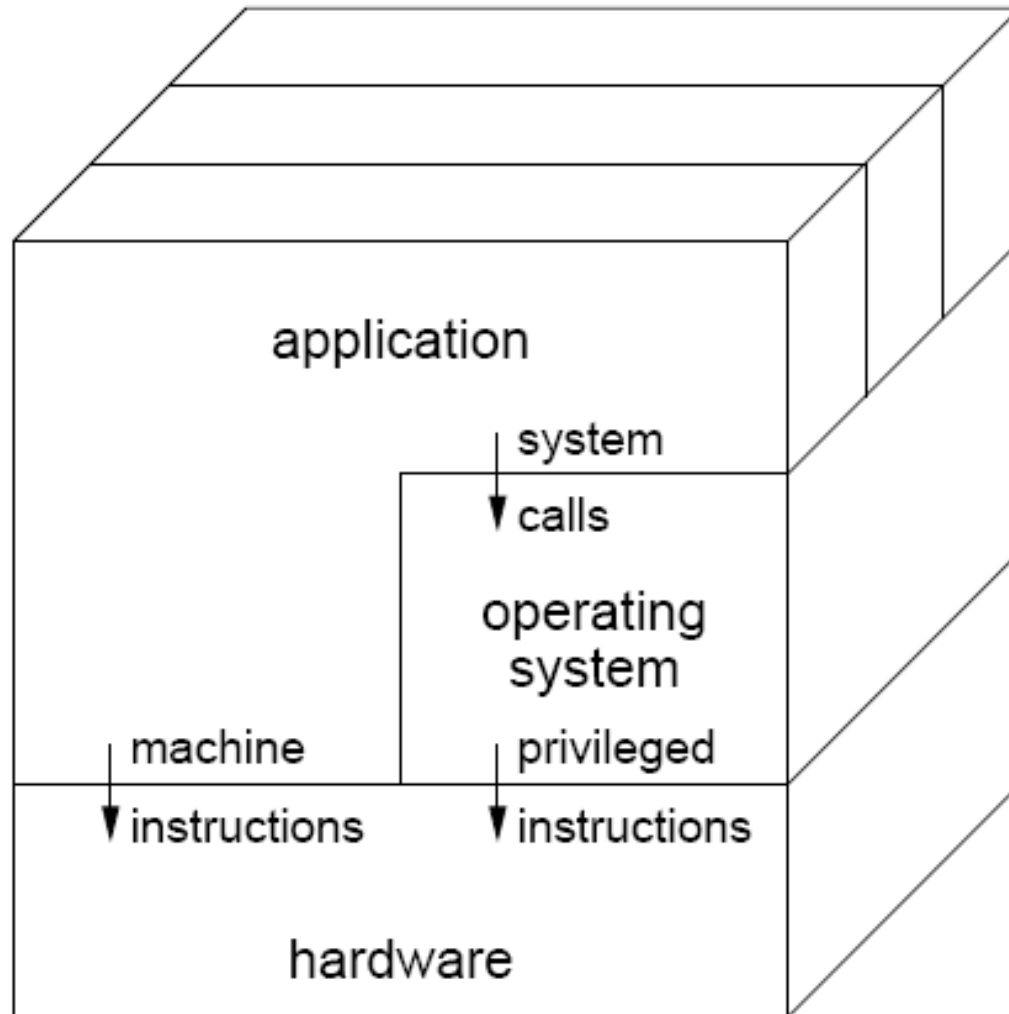


Các ứng dụng

Hệ Điều Hành

Phần cứng

Định nghĩa (tt)



Hình của Dror G. Feitelson

Lịch sử phát triển OS

❑ *Máy tính lớn* (mainframes)

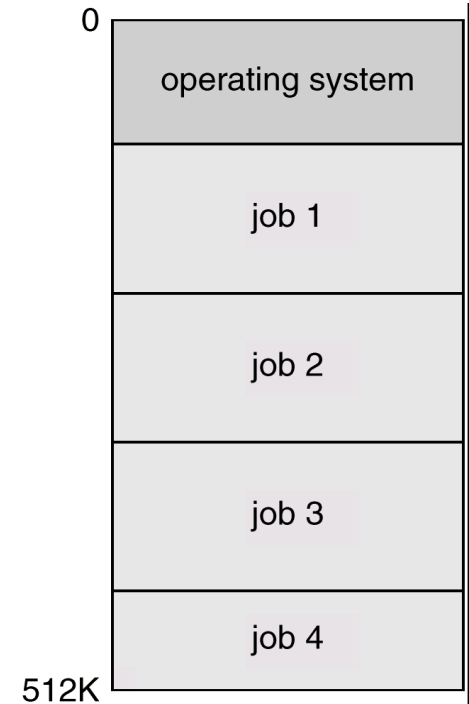
- Xử lý bó (batch)
- Đa chương (multiprogrammed)
- Đa nhiệm (time-sharing, multitasking)

❑ (Mainframe) *Batch systems*

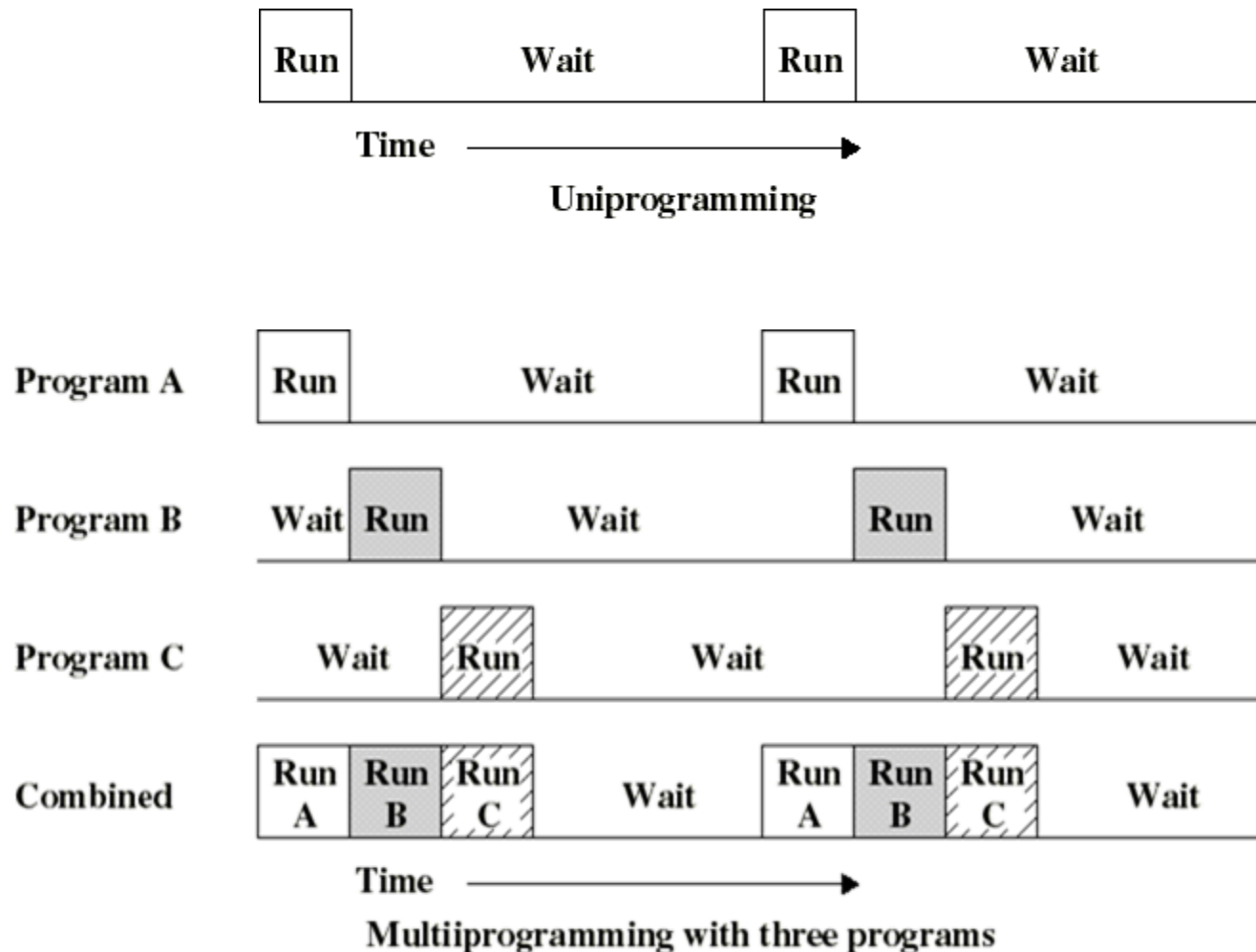
- I/O: card đọc lỗ, băng từ (tapes), line printer
 - Cần có người vận hành (user \neq operator)
 - Giảm setup time bằng cách ghép nhóm công việc (batching)
 - Vd: ghép các công việc cùng sử dụng trình biên dịch Fortran
 - Tự động nạp lần lượt các chương trình
-

Lịch sử phát triển hệ điều hành (tt)

- ❑ (Mainframe) *Multiprogrammed systems*
 - Nhiều công việc được nạp đồng thời vào bộ nhớ chính
 - Thời gian xử lý của CPU được phân chia giữa các công việc đó
 - Tận dụng được thời gian rảnh, tăng *hiệu suất sử dụng* CPU (CPU utilization)
 - Yêu cầu đối với hệ điều hành
 - Định thời công việc (job scheduling): chọn job trong job pool trên đĩa và nạp nó vào bộ nhớ để thực thi.
 - Quản lý bộ nhớ (memory management)
 - Định thời CPU (CPU scheduling)
 - Cấp phát tài nguyên (đĩa, máy in,...)
 - Bảo vệ



Lịch sử phát triển hệ điều hành (tt)



Lịch sử phát triển hệ điều hành (tt)

❑ (Mainframe) *Time-sharing systems*

- Multiprogrammed systems không cung cấp khả năng tương tác hiệu quả với users
 - CPU luân phiên thực thi giữa các công việc
 - Mỗi công việc được chia một phần nhỏ thời gian CPU (*time slice*, *quantum time*)
 - Cung cấp tương tác giữa user và hệ thống với *thời gian đáp ứng* (response time) nhỏ (1 s)
 - Một công việc chỉ được chiếm CPU khi nó nằm trong bộ nhớ chính.
 - Khi cần thiết, một công việc nào đó có thể được chuyển từ bộ nhớ chính ra thiết bị lưu trữ (swapping), nhường bộ nhớ chính cho công việc khác.
-

Lịch sử phát triển hệ điều hành (tt)

- ❑ Yêu cầu đối với OS trong hệ thống time-sharing
 - Định thời công việc (job scheduling)
 - Quản lý bộ nhớ (memory management)
 - Virtual memory
 - Quản lý các tiến trình (process management)
 - Định thời CPU
 - Đồng bộ các tiến trình (synchronization)
 - Giao tiếp giữa các tiến trình (process communication)
 - Tránh deadlock
 - Quản lý hệ thống file, hệ thống lưu trữ
 - Cấp phát hợp lý các tài nguyên
 - Bảo vệ (protection)
-

Lịch sử phát triển hệ điều hành (tt)

- ❑ *Máy để bàn* (desktop system, personal computer)
 - Nhiều thiết bị I/O: bàn phím, chuột, màn hình, máy in,...
 - Phục vụ người dùng đơn lẻ.
 - Mục tiêu chính của OS
 - Thuận tiện cho user và khả năng tương tác cao.
 - Không cần tối ưu hiệu suất sử dụng CPU và thiết bị ngoại vi.
 - Nhiều hệ điều hành khác nhau (MS Windows, Mac OS, Solaris, Linux,...).
-

Lịch sử phát triển hệ điều hành (tt)

- ❑ *Hệ thống song song* (parallel, multiprocessor, hay tightly-coupled system)
 - Nhiều CPU
 - Chia sẻ computer bus, clock
 - Ưu điểm
 - *System throughput*: càng nhiều processor thì càng nhanh xong công việc
 - Multiprocessor system ít tốn kém hơn multiple single-processor system: vì có thể dùng chung tài nguyên (đĩa,...)
 - *Độ tin cậy*: khi một processor hỏng thì công việc của nó được chia sẻ giữa các processor còn lại
-

Lịch sử phát triển hệ điều hành (tt)

- ❑ Phân loại hệ thống song song
 - *Đa xử lý đối xứng* (symmetric multiprocessor - SMP)
 - Mỗi processor vận hành một identical copy của hệ điều hành
 - Các copy giao tiếp với nhau khi cần
 - *Đa xử lý bất đối xứng* (asymmetric multiprocessor)
 - Mỗi processor thực thi một công việc khác nhau
 - Master processor định thời và phân công việc cho các slave processors
-

Lịch sử phát triển hệ điều hành (tt)

- ❑ *Hệ thống phân tán* (distributed system, loosely-coupled system)
 - Mỗi processor có bộ nhớ riêng, các processor giao tiếp qua các kênh nối như mạng, bus tốc độ cao, leased line
 - Người dùng chỉ thấy một hệ thống đơn nhất
 - Ưu điểm
 - Chia sẻ tài nguyên (resource sharing)
 - Chia sẻ sức mạnh tính toán (computational sharing)
 - Độ tin cậy cao (high reliability)
 - *Độ sẵn sàng* cao (high availability): các dịch vụ của hệ thống được cung cấp liên tục cho dù một thành phần hardware trở nên hỏng
-

Lịch sử phát triển hệ điều hành (tt)

❑ Hệ thống phân tán (tt)

Các mô hình hệ thống phân tán

– *Client-server*

- Server: cung cấp dịch vụ
- Client: có thể sử dụng dịch vụ của server

– *Peer-to-peer* (P2P)

- Các *peer* (máy tính trong hệ thống) đều ngang hàng nhau
- Không có cơ sở dữ liệu tập trung
- Các peer là tự trị
- Vd: Gnutella

(Napster không phải là hệ thống P2P đúng nghĩa vì có cơ sở dữ liệu tập trung)

Lịch sử phát triển hệ điều hành (tt)

□ *Hệ thống thời gian thực* (real-time system)

- Sử dụng trong các thiết bị chuyên dụng như điều khiển các thử nghiệm khoa học, điều khiển trong y khoa, dây chuyền công nghiệp, thiết bị gia dụng
- Ràng buộc về thời gian: hard và soft real-time

Phân loại

– *Hard real-time*

- Hạn chế (hoặc không có) bộ nhớ phụ, tất cả dữ liệu nằm trong bộ nhớ chính (RAM hoặc ROM)
- Yêu cầu về thời gian đáp ứng/xử lý rất nghiêm ngặt, thường sử dụng trong điều khiển công nghiệp, robotics,...

– *Soft real-time*

- Thường được dùng trong lĩnh vực multimedia, virtual reality với yêu cầu mềm dẻo hơn về thời gian đáp ứng
-

Lịch sử phát triển hệ điều hành (tt)

❑ *Thiết bị cầm tay* (handheld system)

- Personal digital assistant (PDA): Palm, Pocket-PC
 - Điện thoại di động (cellular phones)
 - Đặc trưng
 - Bộ nhớ nhỏ (512 KB – 128 MB)
 - Tốc độ processor thấp (để ít tốn pin)
 - Màn hình hiển thị có kích thước nhỏ và độ phân giải thấp.
 - Có thể dùng các công nghệ kết nối như IrDA, Bluetooth, wireless
-

2. Cấu Trúc Hệ Điều Hành

1. Các chức năng chính của hệ điều hành
 2. Các dịch vụ hệ điều hành cung cấp
 3. Giao tiếp giữa tiến trình và hệ điều hành
 4. Các chương trình hệ thống (system programs)
 5. Cấu trúc logic của hệ thống
 6. Máy ảo (virtual machine)
-

2.1. Các chức năng chính của hệ điều hành

a) Quản lý tiến trình (process management)

- tiến trình vs. chương trình
 - Một tiến 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 tiến trình
 - Tạm ngưng/tiếp tục thực thi (suspend/resume) tiến trình
 - Cung cấp các cơ chế
 - đồng bộ hoạt động các tiến trình (synchronization)
 - giao tiếp giữa các tiến trình (interprocess communication)
 - khống chế deadlock
-

Các chức năng chính của hệ điều hành (tt)

b) Quản lý bộ nhớ chính

- 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 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
-

Các chức năng chính của hệ điều hành (tt)

c) 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 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 chức năng chính của hệ điều hành (tt)

d) Quản lý hệ thống I/O (I/O system management)

- Che dấu các đặc trưng riêng biệt của từng thiết bị I/O
 - Có chức năng
 - Cơ chế: buffering, caching, spooling
 - Cung cấp giao diện chung đến các trình điều khiển thiết bị (device-driver interface)
 - *Trình điều khiển thiết bị* (device driver) cho mỗi chủng loại thiết bị phần cứng khác nhau.
-

Các chức năng chính của hệ điều hành (tt)

e) 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 thành phần
 - Quản lý vùng trống (free space management)
 - Cấp phát không gian lưu trữ (storage allocation)
 - Định thời đĩa (disk scheduling)
-

Các chức năng chính của hệ điều hành (tt)

f) Hệ thống bảo vệ

Khi hệ thống cho phép nhiều user hay nhiều tiến trình

- Kiểm soát tiến 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/tiến trình *đủ quyền hạn* mới được phép sử dụng các tài nguyên tương ứng
 - Các nhiệm vụ của thành phần
 - 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
-

Các chức năng chính của hệ điều hành (tt)

g) 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 tiến 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
 - ...
-

2.2. Các dịch vụ hệ điều hành cung cấp

- a) 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 hệ thống file
 - Đọc/ghi hay tạo/xóa file
 - Cơ chế giao tiếp, trao đổi thông tin giữa các tiến 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 (tt)

b) Các chức năng khác (giúp hệ điều hành chạy hữu hiệu)

- Cấp phát tài nguyên (resource allocation)
 - Tài nguyên: CPU, bộ nhớ chính, tape drives,...
 - OS có các routine tương ứng
 - Kiểm soát (accounting)
 - Ví dụ: chống can thiệp trái phép, tăng độ tin cậy hệ thống,...
 - Bảo vệ (protection)
 - Các tiến trình lạ nhau không được ảnh hưởng nhau
 - Kiểm soát được các truy cập vào tài nguyên
 - An ninh (security)
 - Chỉ các user được phép sử dụng hệ thống mới truy cập được tài nguyên của hệ thống (vd: thông qua password)
-

2.3. Giao tiếp giữa tiến trình và hệ điều hành

□ *System call*

- Cung cấp giao diện giữa tiến trình và hệ điều hành
 - Vd: open, read, write file
 - Thông thường ở dạng thư viện nhị phân (binary libraries)
 - 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
-

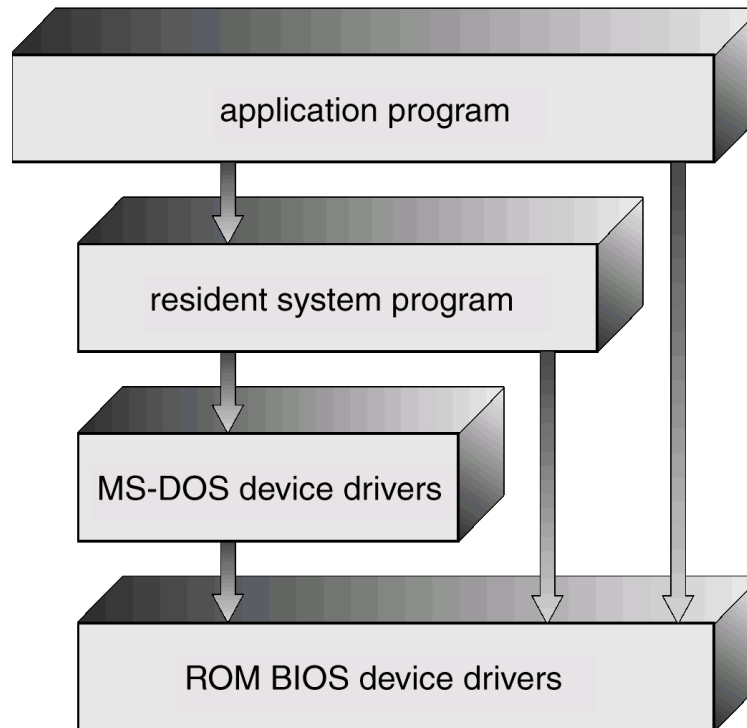
2.4. Các chương trình hệ thống

- ❑ *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 logic của hệ thống

❑ Hệ thống *đơn* (monolithic)

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



Cấu trúc logic của hệ thống (tt)

❑ Hệ thống đơn (monolithic)

– UNIX: gồm hai phần

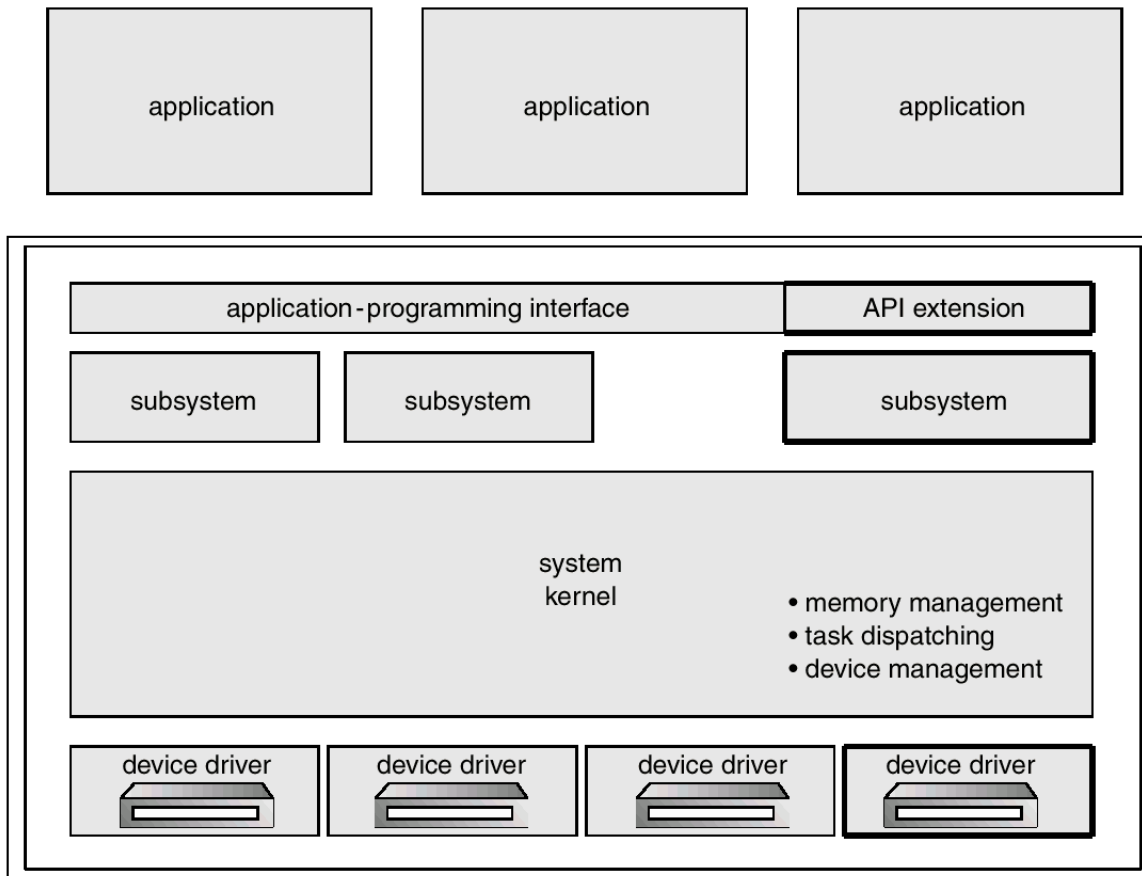
- các system program và kernel (file system, CPU scheduling, **memory** management, và một số chức năng khác)

Xem như được
phân lớp:

(the users)		
shells and commands compilers and interpreters system libraries		
<i>system-call interface to the kernel</i>		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
<i>kernel interface to the hardware</i>		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

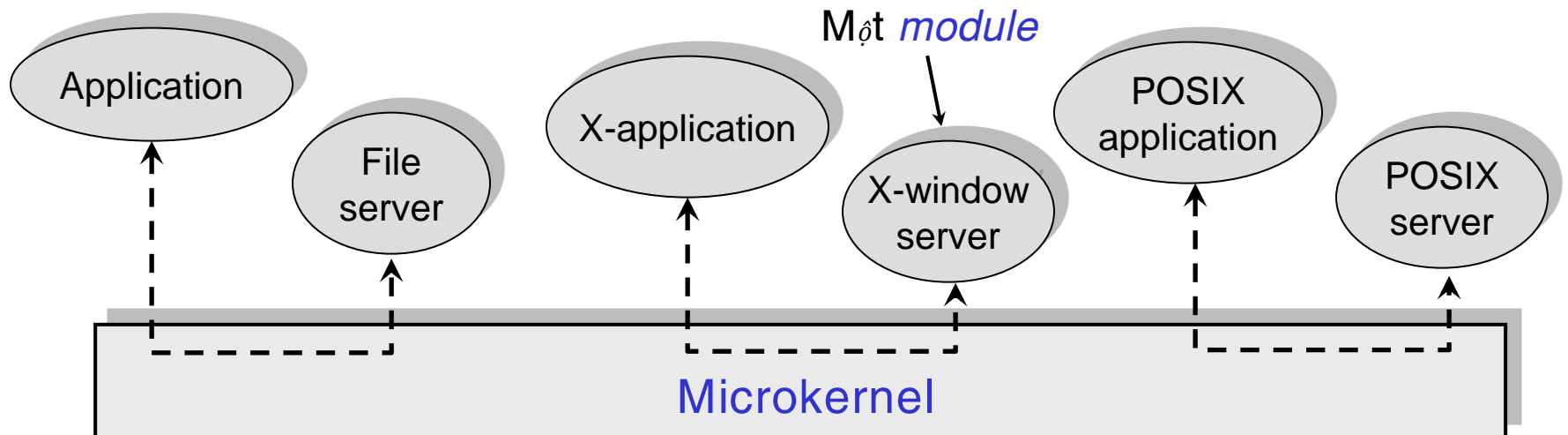
Cấu trúc logic của hệ thống (tt)

- ❑ Thiết kế OS: phân chia module theo cách *phân lớp* (layered).
Vd: hệ điều hành OS/2



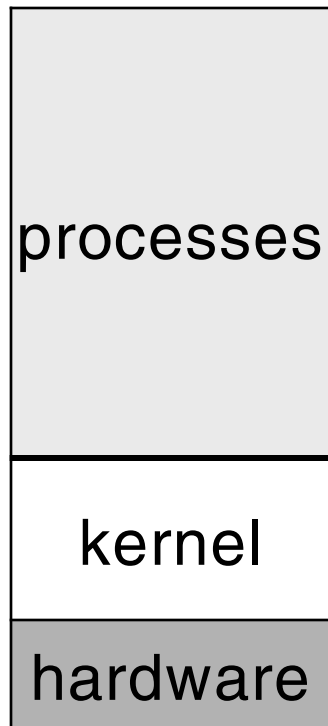
Cấu trúc logic của hệ thống (tt)

- ❑ 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
 - Thu gọn kernel → microkernel, microkernel chỉ bao gồm các chức năng tối thiểu như quản lý tiến trình, bộ nhớ và cơ chế giao tiếp giữa các tiến trình
 - Giao tiếp giữa các module qua cơ chế *truyền thông điệp*

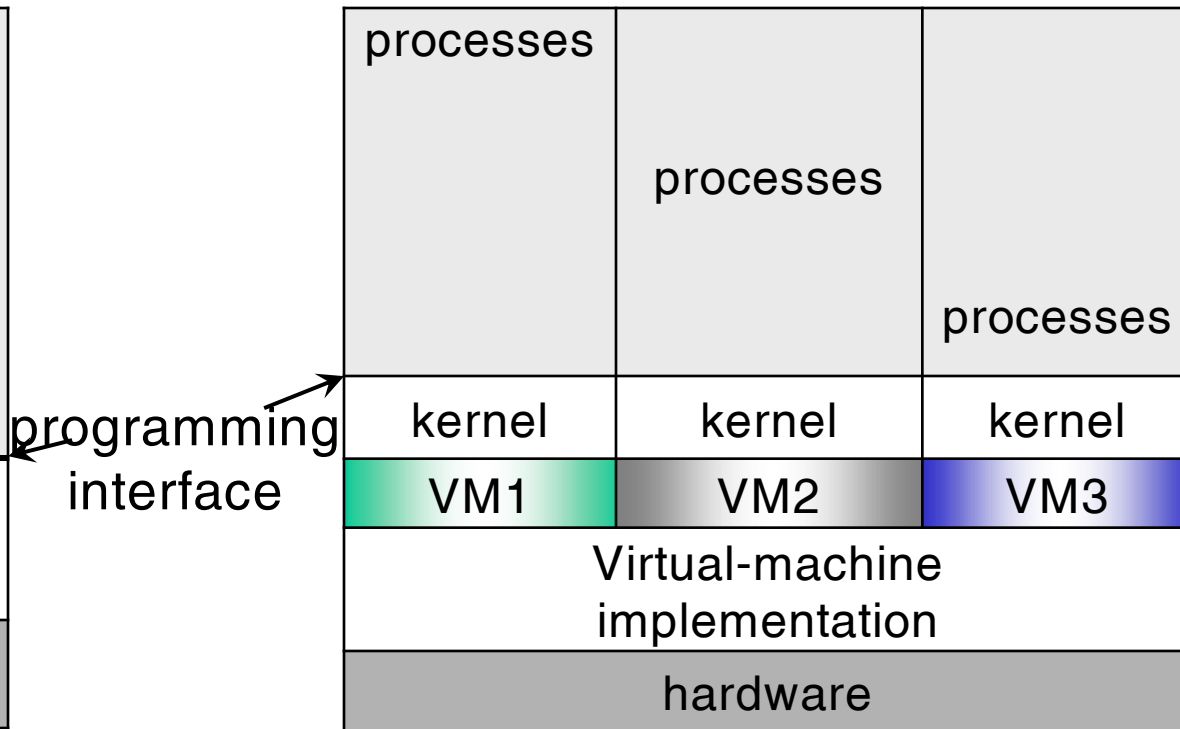


2.6. Máy ảo

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



Non-virtual machine
system model

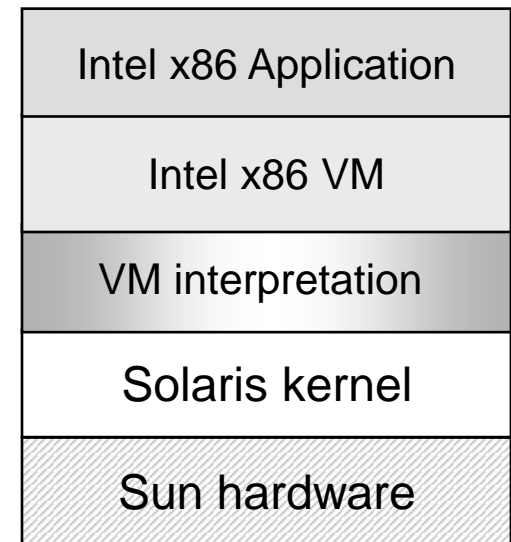


Virtual machine system model

Máy ảo (tt)

❑ 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.



3. Ví dụ các hệ điều hành

❑ 3.1. Hệ điều hành Windows

❑ 3.2. Hệ điều hành Linux

(Sinh viên tự đọc tìm hiểu)
