

HỆ ĐIỀU HÀNH (OPERATING SYSTEM)

Trình bày: Nguyễn Hoàng Việt
Khoa Công Nghệ Thông Tin
Đại Học Cần Thơ

Chương 2: Cấu trúc của hệ điều hành

- Các thành phần của hệ điều hành
- Các dịch vụ mà hệ điều hành cung cấp
- Lời gọi hệ thống
- Các chương trình hệ thống
- Cấu trúc hệ điều hành
- Cài đặt hệ điều hành
- Tạo môi trường hệ thống

Các thành phần của HĐH (1)

- HĐH là một hệ thống lớn và phức tạp bao gồm nhiều thành phần, mỗi thành phần bao gồm có inputs, outputs và functions được định nghĩa rõ ràng
- Các thành phần của HĐH:
 - Quản lý quá trình
 - Quản lý bộ nhớ chính
 - Quản lý tập tin
 - Quản lý hệ thống xuất/nhập
 - Quản lý bộ nhớ thứ cấp
 - Nối mạng
 - Hệ thống bảo vệ
 - Giao diện người dùng

Các thành phần của HĐH (2)

Quản lý quá trình (Process Management)

- Quá trình (process) là một chương trình đang thực thi.
- Quá trình cần các tài nguyên để thực hiện tác vụ của nó: thời gian phục vụ của CPU, bộ nhớ, tập tin, thiết bị vào ra.
- Bộ quản lý quá trình (Process Manager) chịu trách nhiệm thực hiện các hoạt động sau:
 - Tạo và hủy quá trình.
 - Ngừng và tiếp tục quá trình.
 - Đưa ra các cơ chế để:
 - ✓ Đồng bộ hóa các quá trình
 - ✓ Thực hiện việc giao tiếp giữa các quá trình
 - ✓ Chống deadlock

Các thành phần của HĐH (3)

Quản lý bộ nhớ chính (Main-Memory Management)

- Bộ nhớ là một mảng lớn các words hoặc bytes, với địa chỉ riêng biệt.
 - Là kho chứa dữ liệu truy cập nhanh, được chia sẻ bởi CPU và các thiết bị vào ra.
 - Là thiết bị lưu trữ bay hơi (volatile storage device), sẽ bị mất nội dung khi hệ thống gặp sự cố.
- Bộ quản lý bộ nhớ chính (Main-Memory Manager) chịu trách nhiệm thực hiện các hoạt động sau:
 - Theo dõi phần nào của bộ nhớ đang được sử dụng bởi ai.
 - Quyết định quá trình nào sẽ được nạp vào bộ nhớ khi không gian nhớ còn chỗ trống.
 - Cấp phát và thu hồi không gian nhớ khi cần thiết.

Các thành phần của HĐH (4)

Quản lý tập tin (File Management)

- Một tập tin (File) là một tập hợp các thông tin có liên quan với nhau, được định nghĩa bởi người tạo ra nó.
- Tập tin thường được dùng để lưu các chương trình hoặc dữ liệu trong các thiết bị lưu trữ, như đĩa.
- Bộ quản lý tập tin (File Manager) chịu trách nhiệm thực hiện các hoạt động sau:
 - Tạo và xóa tập tin.
 - Tạo và xóa thư mục (directory, folder).
 - Hỗ trợ các cơ sở cho việc thao tác trên tập tin và thư mục.
 - Ánh xạ tập tin lên các thiết bị lưu trữ thứ cấp.
 - Chép phòng hờ (backup) tập tin lên các phương tiện lưu trữ ổn định (không bay hơi).

Các thành phần của HĐH (5)

Quản lý hệ thống xuất/nhập (I/O System Management)

- Hệ thống xuất/nhập bao gồm:
 - Hệ thống lưu trữ đệm (Buffer-Caching System: buffering, caching, spooling).
 - Giao diện điều khiển thiết bị tổng quát (General Device-Driver Interface).
 - Trình điều khiển thiết bị (driver) cho các thiết bị cụ thể.
- Hệ thống xuất/nhập giao tiếp với các thành phần khác của hệ thống, quản lý các thiết bị, chuyển tải dữ liệu, và phát hiện một hoàn thành xuất/nhập.

Các thành phần của HĐH (6)

Quản lý bộ nhớ thứ cấp (Secondary-Storage Management)

- Bộ nhớ chính (Primary Storage) bị bay hơi và quá nhỏ để chứa tất cả dữ liệu và chương trình lâu dài → dùng thiết bị lưu trữ thứ cấp (Secondary Storage) để hỗ trợ.
 - Hầu hết sử dụng đĩa từ làm thiết bị lưu trữ trực tuyến chính yếu cho cả dữ liệu và chương trình.
- Bộ quản lý đĩa (Disk Manager) chịu trách nhiệm thực hiện các hoạt động sau:
 - Quản lý không gian còn trống
 - Cấp phát không gian lưu trữ
 - Định thời sử dụng đĩa

Các thành phần của HĐH (7)

Kết nối mạng (Networking) - Các hệ thống phân tán

- Hệ thống phân tán (Distributed System) là tập hợp các bộ xử lý không dùng chung bộ nhớ hoặc xung đồng hồ. Mỗi bộ xử lý cũng có bộ nhớ riêng.
- Các bộ xử lý trong hệ thống được nối kết thông qua một mạng truyền thông (Communication Network).
- Giao tiếp được thực hiện thông qua các giao thức (protocol): FTP, NFS, http...
- Hệ thống phân tán cho phép người dùng truy cập nhiều loại tài nguyên hệ thống khác nhau.
- Việc truy cập đến các tài nguyên chia sẻ sẽ giúp:
 - Tăng tốc độ tính toán
 - Tăng mức độ sẵn dùng của dữ liệu
 - Tăng độ tin cậy

Các thành phần của HĐH (7)

Hệ thống bảo vệ (Protection System)

- Khái niệm **bảo vệ** nhằm chỉ cơ chế điều khiển truy cập từ các chương trình, quá trình hoặc người dùng đến tài nguyên của cả hệ thống và của người dùng.
- Cơ chế bảo vệ phải:
 - Phân biệt được việc truy cập có thẩm quyền hay không.
 - Xác định những quyền điều khiển được áp đặt.
 - Cung cấp các phương tiện để bảo vệ an ninh.

Các thành phần của HĐH (8)

User Interface - Thông dịch lệnh (Command-Interpreter)

- Các câu lệnh được chuyển cho HĐH theo các câu lệnh điều khiển nhằm thực hiện việc:
 - Tạo và quản lý quá trình
 - Kiểm soát hệ thống I/O
 - Quản lý hệ thống lưu trữ thứ cấp
 - Quản lý bộ nhớ chính
 - Truy cập hệ thống tập tin
 - Bảo vệ
 - Kết nối mạng
- Chương trình làm nhiệm vụ đọc và thông dịch các câu lệnh điều khiển được gọi theo nhiều kiểu khác nhau:
 - Trình thông dịch lệnh (command-line interpreter)
 - shell (trong UNIX)

Các thành phần của HĐH (9)

User Interface - Môi trường nền (Desktop Environment)

- Giao diện người dùng theo dạng đồ họa (GUI -Graphical User Interface): Windows DE, GNOME DE, KDE.
- DE điển hình cung cấp các **icons**, **windows**, **toolbars**, **folders**, **wallpapers**, và khả năng **drag and drop**.
- DE bao gồm:
 - **window manager** (như Metacity hoặc Kwin),
 - **file manager** (như Konqueror hoặc Nautilus),
 - tập hợp các **themes**, các chương trình và các thư viện cho việc quản lý desktop.

Các dịch vụ của HĐH

(OS Services)

Cung cấp các dịch vụ cho chương trình và cho người dùng chương trình:

- Giao diện người dùng: command line, batch interface, GUI
- Thực thi chương trình: nạp chương trình vào bộ nhớ và chạy nó
- Thao tác vào ra: do chương trình người dùng không thể thực hiện trực tiếp các thao tác I/O, HĐH phải cung cấp phương tiện để thực hiện thao tác I/O
- Thao tác hệ thống tập tin: cung cấp khả năng có thể lập trình để đọc, ghi, tạo và xóa tập tin
- Giao tiếp: chuyển thông tin giữa các quá trình đang thực thi trên cùng một máy tính hoặc trên nhiều hệ thống được kết nối với nhau qua mạng máy tính (thông qua các phương pháp **bộ nhớ chia sẻ** hoặc **chuyển thông điệp**)
- Phát hiện lỗi: bảo đảm tính toán chính xác bằng cách phát hiện lỗi phát sinh tại CPU và bộ nhớ, tại thiết bị vào ra hoặc tại chương trình người dùng

Các chức năng khác của HĐH

Một số chức năng (function) khác tồn tại không nhằm hỗ trợ người dùng mà dùng để đảm bảo cho hoạt động hiệu quả của hệ thống:

- Cấp phát tài nguyên: cấp tài nguyên cho nhiều người dùng hoặc nhiều công việc đang chạy song song.
- Tính chi phí: theo dõi và ghi lại người dùng nào đã sử dụng tài nguyên gì của hệ thống để làm cơ sở tính tiền sử dụng hệ thống hoặc thống kê sử dụng.
- Bảo vệ: đảm bảo rằng tất cả truy cập đến hệ thống đều được kiểm soát.

Lời gọi hệ thống (1)

(System Call)

- Lời gọi hệ thống là giao diện giữa quá trình và HĐH.
 - Thông thường, lời gọi hệ thống tồn tại dưới dạng các chỉ thị dạng ngôn ngữ assembler
 - Ngoài ra lời gọi hệ thống còn được cài đặt dưới dạng các hàm (function) dạng ngôn ngữ cấp cao như C hoặc C++ (UNIX)
- Cách 1 lời gọi hệ thống được dùng: xem 1 chương trình đọc dữ liệu từ 1 tập tin và copy chúng vào 1 tập tin khác
 - Get tên của tập tin nhập và tập tin xuất
 - Open tập tin nhập
 - Create tập tin xuất
 - Read từ tập tin nhập
 - Write vào tập tin xuất
 - Close cả 2 tập tin

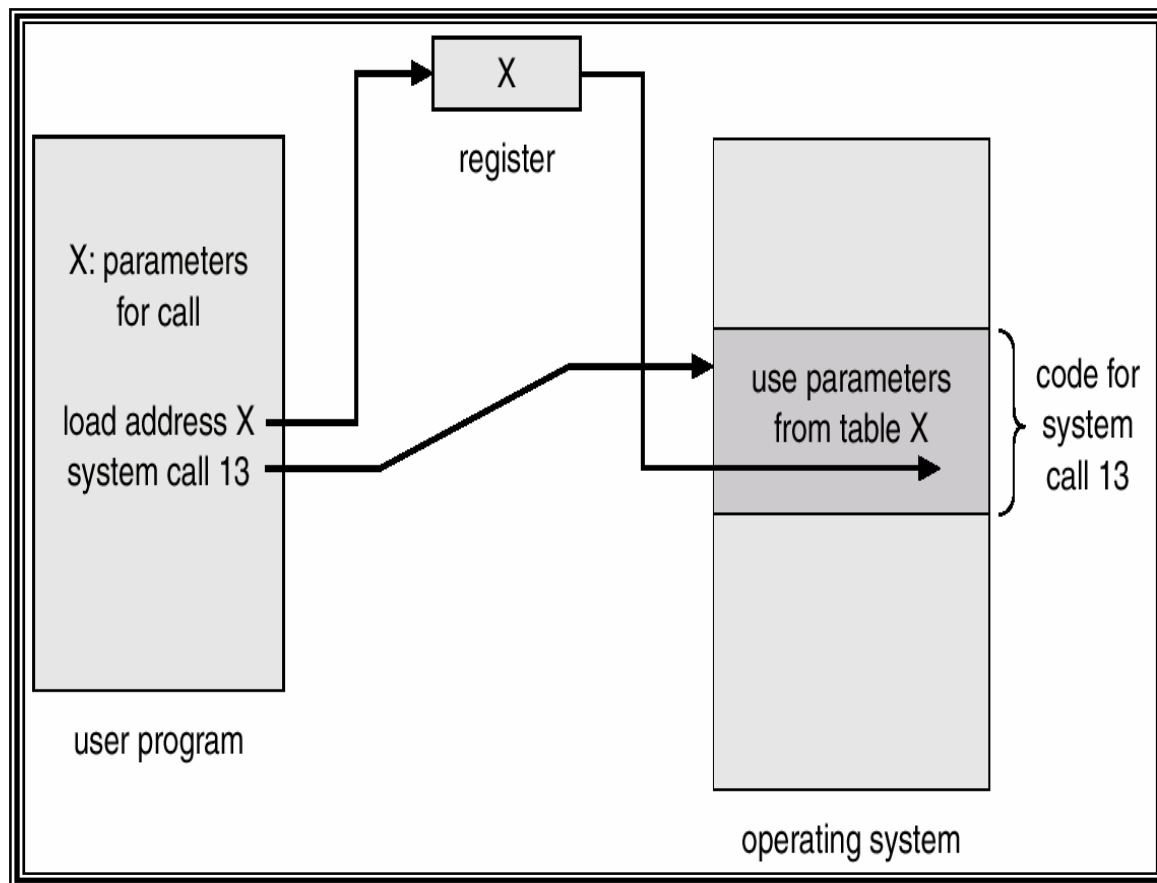
Lời gọi hệ thống (2)

Chuyển tham số giữa chương trình và HĐH

- Ngay cả những chương trình đơn giản cũng dường như khó khăn khi dùng HĐH
 - Tuy nhiên các người dùng thường không nhìn thấy mức độ chi tiết này
 - API (Application Programming Interface) định rõ các hàm Win16/32/64, POSIX, Java APIs
- Có 3 phương thức tổng quát để chuyển các tham số từ quá trình tới HĐH:
 - Chuyển tham số vào các thanh ghi
 - Lưu tham số vào một bảng trong bộ nhớ và địa chỉ của bảng này sẽ được chuyển như là tham số vào thanh ghi
 - Chương trình **push** (trữ) tham số vào **stack**, và HĐH sẽ lấy tham số bằng cách **pop stack**

Lời gọi hệ thống (3)

Chuyển tham số như là một bảng



Lời gọi hệ thống (4)

Các dạng của lời gọi hệ thống

- Điều khiển quá trình (Process control)
- Quản lý file (File management)
- Quản lý thiết bị (Device management)
- Duy trì thông tin trạng thái (Information maintenance)
- Giao tiếp (Communication)

Lời gọi hệ thống (5)

Các dạng của lời gọi hệ thống (tt)

- Process control:
 - end, abort
 - load, execute
 - create process, terminate process
 - get process attributes, set process attributes
 - wait for time
 - wait event, signal event
 - allocate and free memory
- File management:
 - create file, delete file
 - open, close
 - read, write, reposition
 - get file attributes, set file attributes

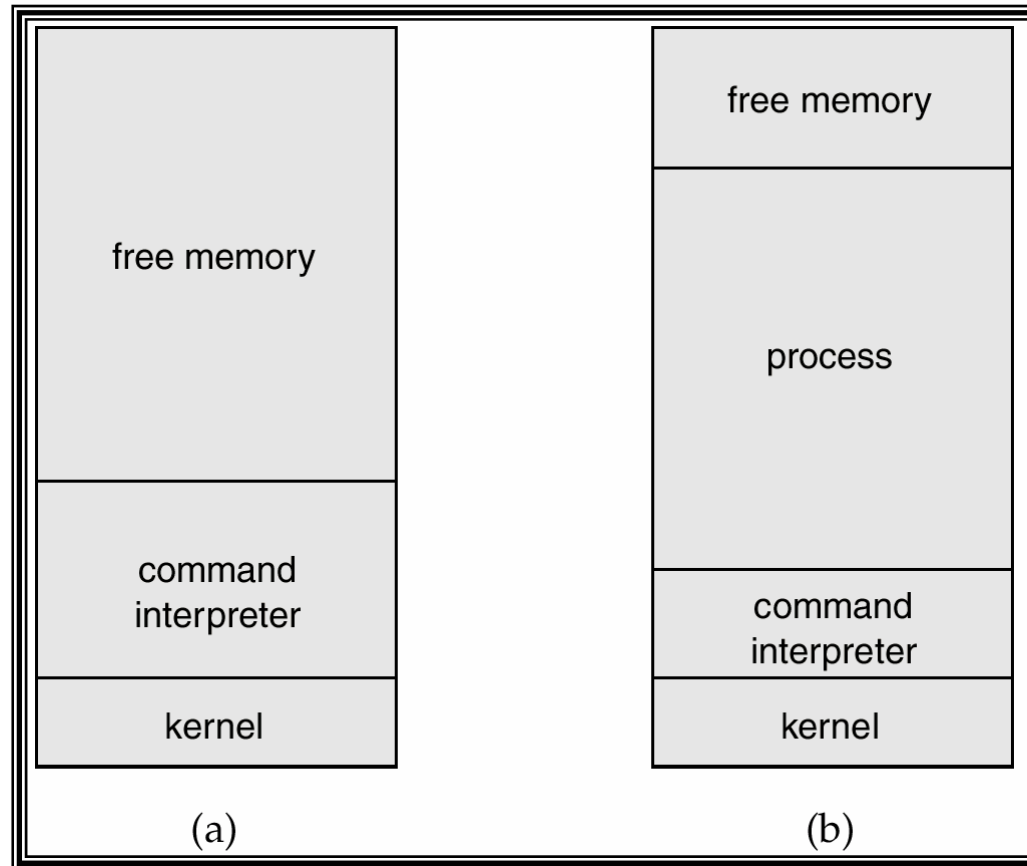
Lời gọi hệ thống (6)

Các dạng của lời gọi hệ thống (tt)

- Device management:
 - request device, release device
 - read, write, reposition
 - get device attributes, set device attributes
 - logically attach or detach devices
- Information maintenance:
 - get time or date, set time or date
 - get system data, set system data
 - get process, file, or device attributes
 - set process, file, or device attributes
- Communications:
 - create, delete communication connection
 - send, receive message
 - Transfer status information
 - Attach or detach remote device

Lời gọi hệ thống (7)

Thực thi chương trình trong MS-DOS

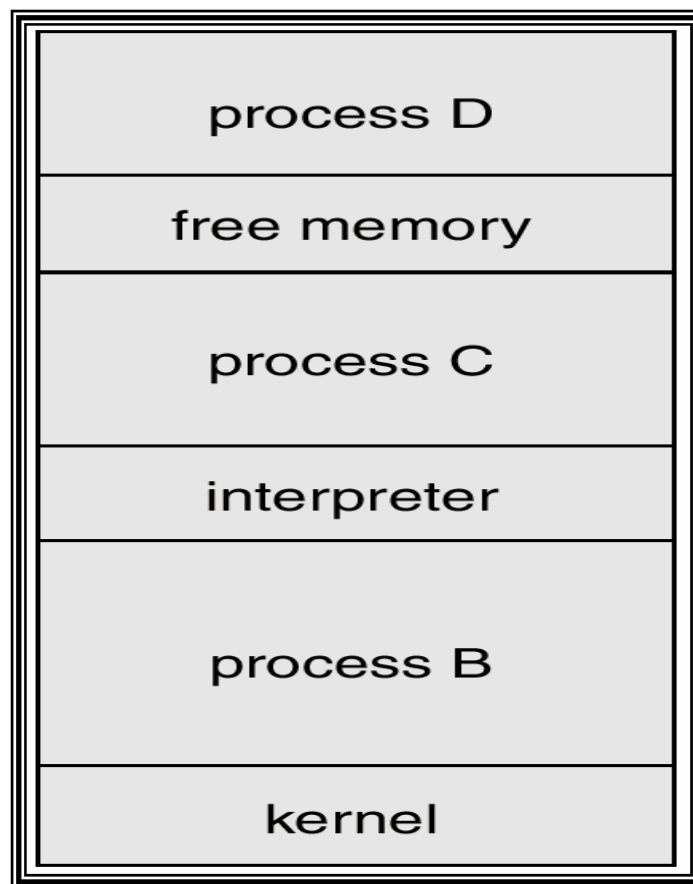


Lúc hệ thống khởi động

Chạy một chương trình

Lời gọi hệ thống (8)

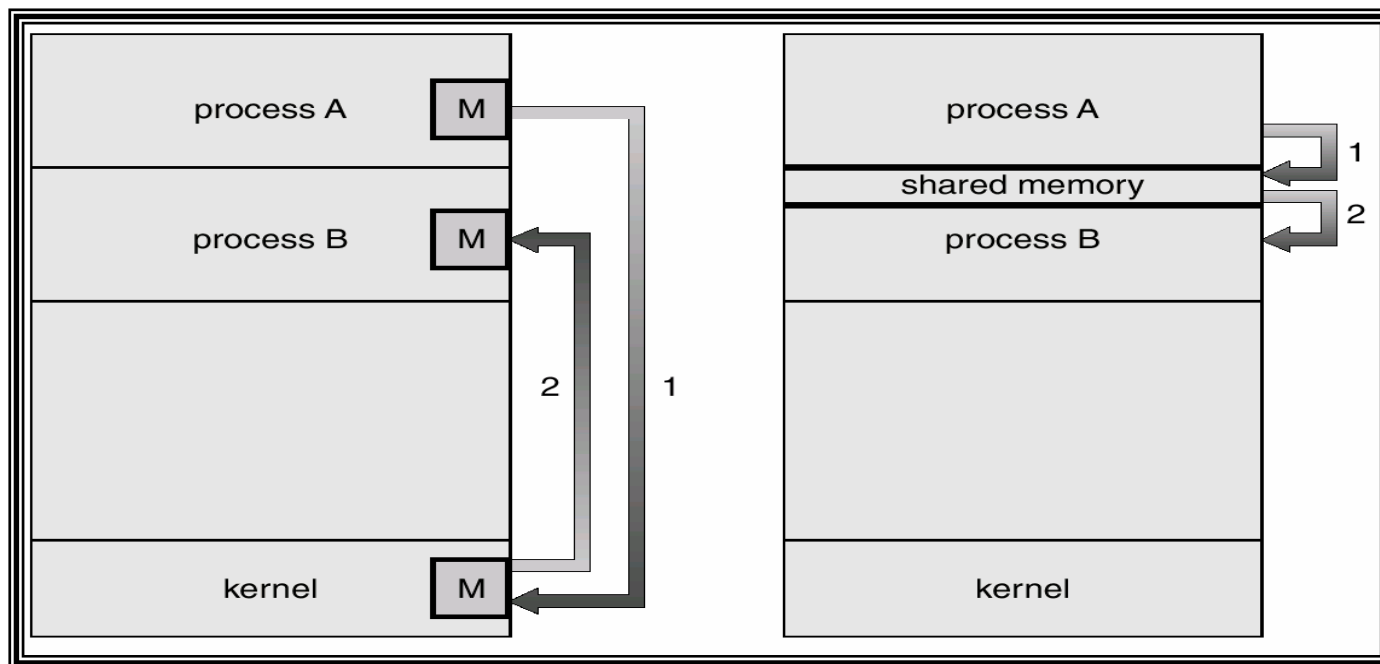
UNIX chạy nhiều chương trình cùng lúc



Lời gọi hệ thống (9)

Các mô hình giao tiếp (Communication Models)

- **Message Passing:** trao đổi thông tin dựa vào phương tiện giao tiếp liên quá trình cung cấp bởi HĐH
- **Shared Memory:** các quá trình dùng Map Memory System Calls để truy cập vào shared memory



Chuyển thông điệp (Message Passing)

Bộ nhớ chia sẻ (Shared Memory)

Chương trình hệ thống (System Program)

- Chương trình hệ thống cung cấp môi trường thuận lợi cho việc phát triển và thực thi chương trình.
- Một số là các giao diện người dùng đơn giản cho các lời gọi hệ thống.
- Người dùng hầu như nhìn HĐH qua các chương trình hệ thống, không qua các lời gọi hệ thống.
- Các loại chương trình hệ thống:
 - Thao tác tập tin
 - Thông tin trạng thái
 - Điều chỉnh tập tin
 - Hỗ trợ ngôn ngữ lập trình
 - Nạp và thực thi chương trình
 - Giao tiếp giữa các quá trình, người dùng, các máy tính

Cấu trúc HĐH (1)

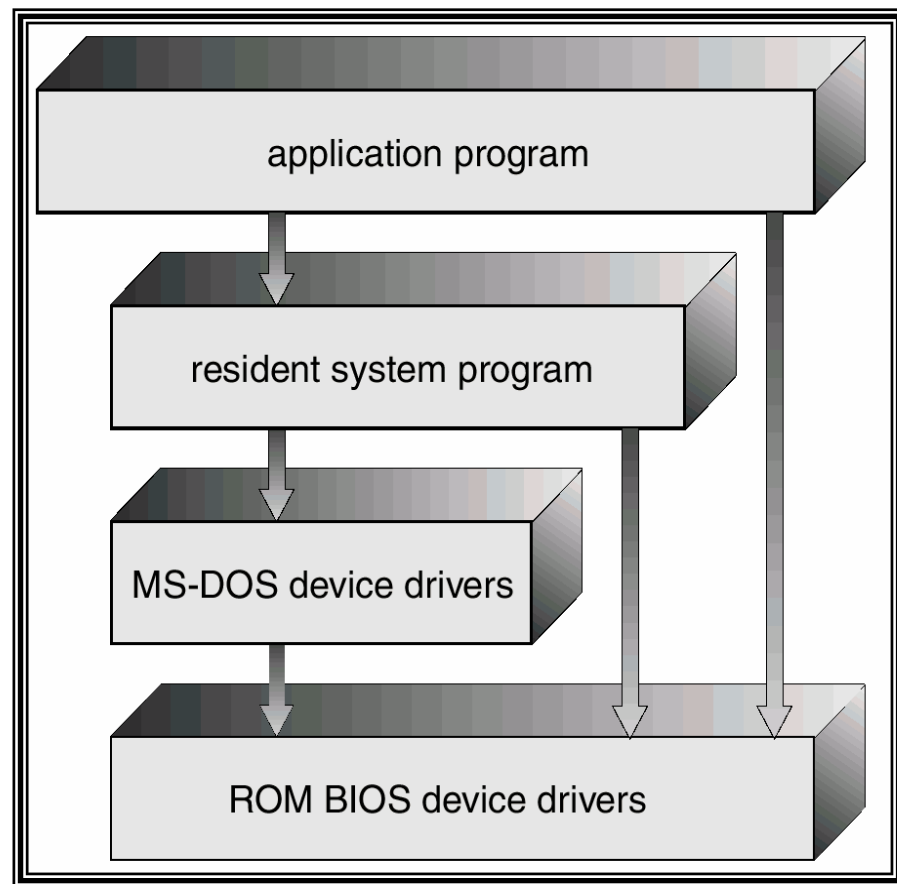
Các cấu trúc HĐH

- Tổ chức của các thành phần HĐH để xác định đặc quyền mà mỗi thành phần thực hiện
- Ba loại cấu trúc:
 - Nguyên khối (Monolithic): tất cả các thành phần chứa trong nhân (kernel)
 - Vi nhân (Microkernel): chỉ những thành phần chủ yếu bao gồm trong kernel
 - Phân tầng (Layered): phương pháp trên-xuống (top-down), tách biệt các chức năng và các đặc điểm trong các thành phần

Cấu trúc HĐH (2)

Cấu trúc hệ thống MS-DOS

- MS-DOS – được viết để cung cấp nhiều chức năng nhất với dung lượng nhỏ nhất.
 - Không được chia thành các modules.
 - Mặc dù MS-DOS có cấu trúc nào đó, các mức chức năng và giao diện giữa chúng không được tách biệt.



Cấu trúc phân tầng của MS-DOS

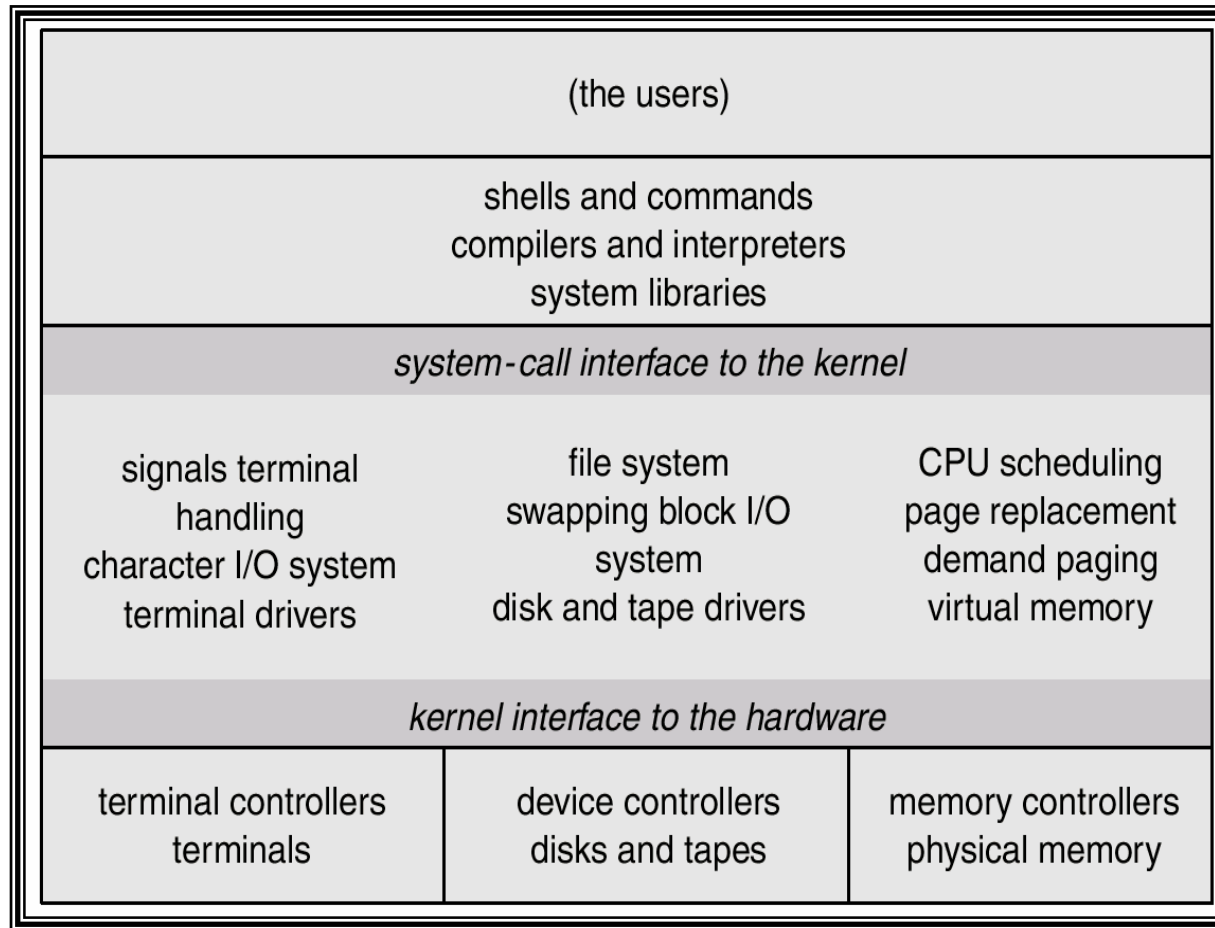
Cấu trúc HĐH (3)

Cấu trúc của hệ thống UNIX

- UNIX – bị giới hạn bởi chức năng phần cứng, HĐH UNIX khởi thủy có cấu trúc giới hạn.
- HĐH UNIX bao gồm hai phần tách biệt.
 - Các chương trình hệ thống
 - Nhân (kernel)
 - ✓ Bao gồm mọi thứ phía dưới giao diện lời gọi hệ thống và phía trên phần cứng vật lý.
 - ✓ Cung cấp cơ chế quản lý hệ thống tập tin, định thời CPU, quản lý bộ nhớ và các chức năng khác của HĐH

Cấu trúc HĐH (4)

Cấu trúc của hệ thống UNIX



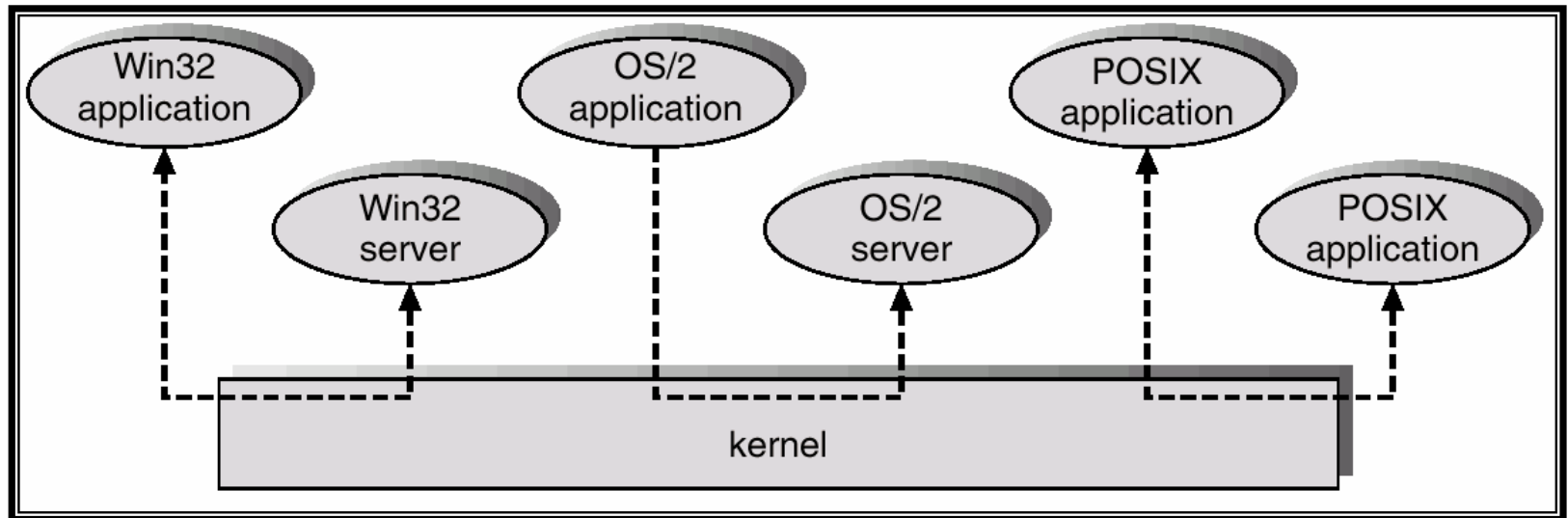
Cấu trúc HĐH - Vi nhân (1)

(Microkernel)

- Di chuyển nhiều chức năng từ nhân lên mức người dùng, giữ lại các phần chính yếu: quản lý quá trình, bộ nhớ, giao tiếp giữa các quá trình → nhân nhỏ hơn
- Giao tiếp giữa các quá trình người dùng được thực hiện bằng cơ chế chuyển thông điệp gián tiếp thông qua nhân
- Lợi ích:
 - Dễ dàng mở rộng HĐH như các dịch vụ mới, đưa vào không gian người dùng
 - Dễ dàng chuyển đổi HĐH sang các kiến trúc mới do nhân nhỏ hơn
 - Tin cậy và an toàn hơn (ít mã lệnh chạy ở mức nhân hơn)

Cấu trúc HĐH - Vi nhân (2)

Kiến trúc Client-Server của Windows NT

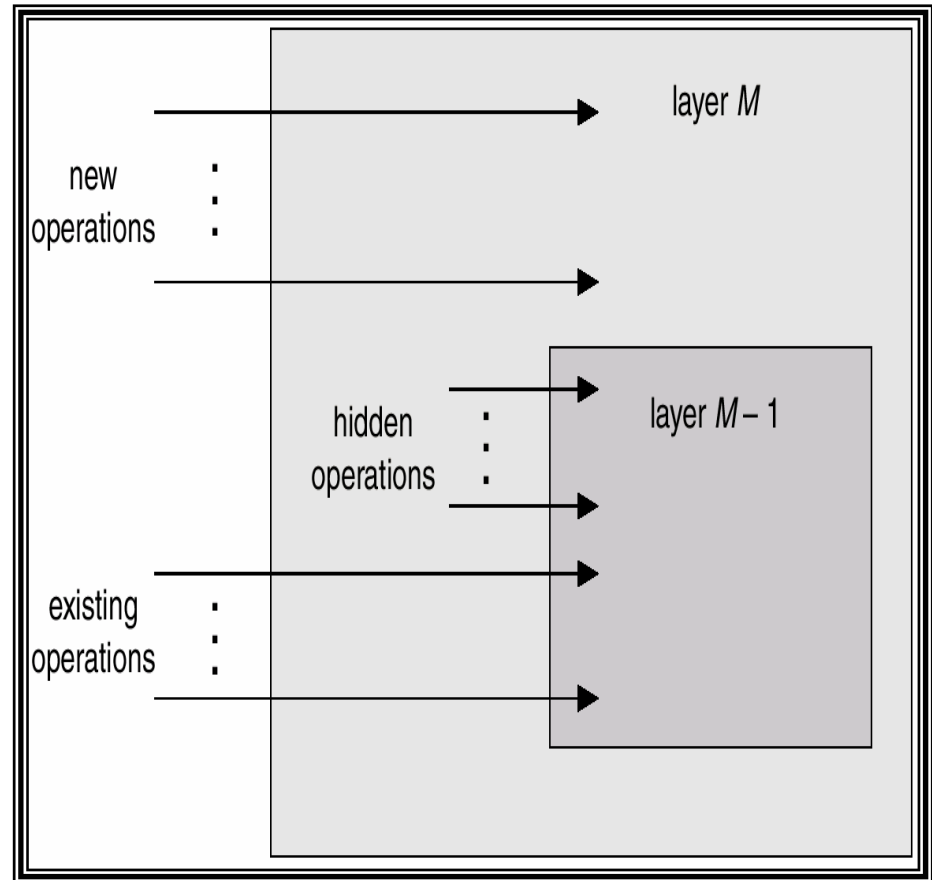


- Các ứng dụng khác nhau (Win32, OS/2, POSIX) chạy trên không gian người dùng
- Server cho mỗi ứng dụng chạy trên không gian người dùng
- Message Passing giữa các client application programs và application servers chạy trong không gian nhân

Cấu trúc HĐH - Phân tầng (1)

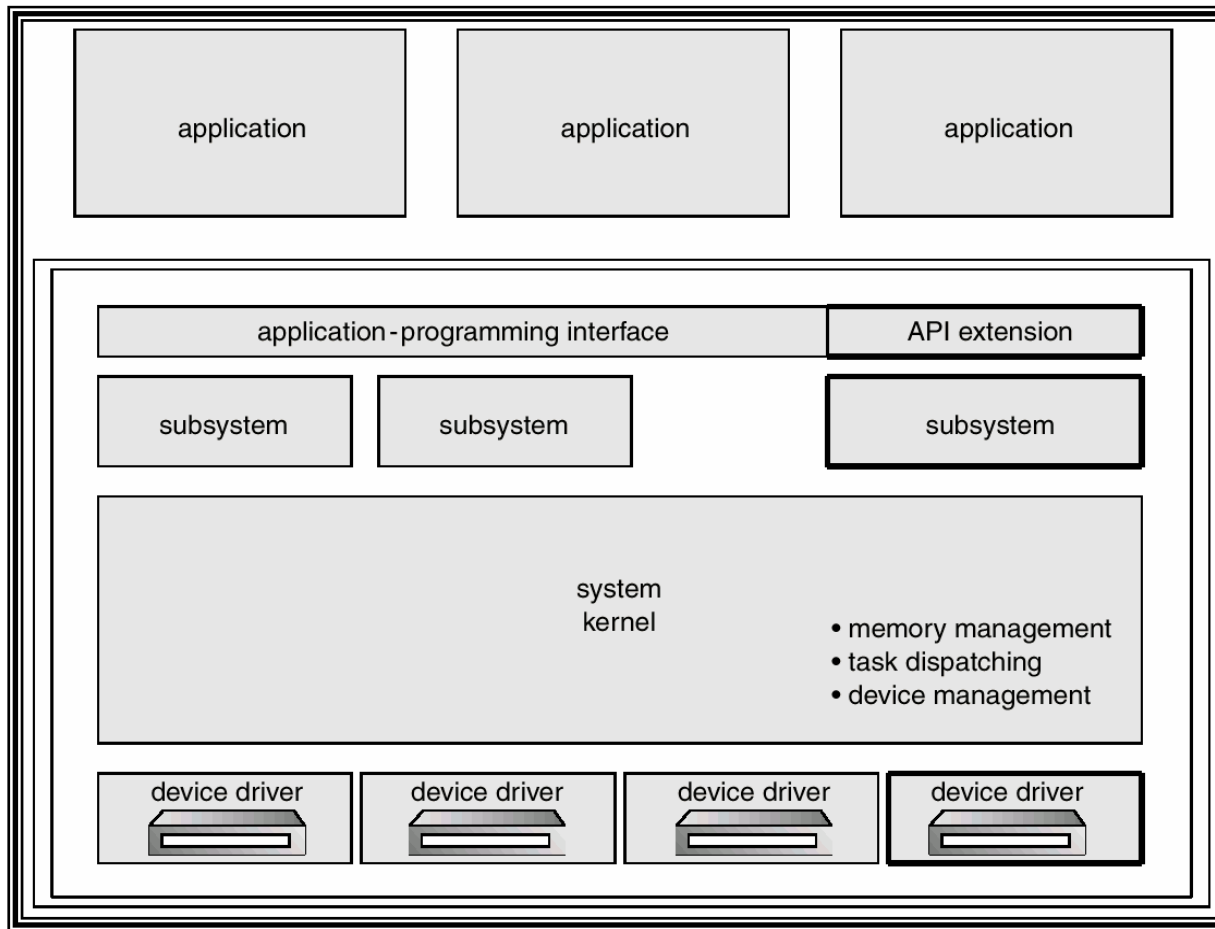
(Layered Structure)

- HĐH được chia thành một số tầng (mức), mỗi tầng được xây dựng trên nền tảng của một tầng khác thấp hơn. Tầng thấp nhất (layer 0) là tầng vật lý, tầng cao nhất (layer N) là giao diện với người dùng.
- Sự phân chia chức năng được thực hiện sao cho mỗi một tầng sẽ sử dụng các chức năng (thao tác) và các dịch vụ được cung cấp chỉ bởi các tầng mức thấp hơn.



Cấu trúc HĐH - Phân tầng (2)

Kiến trúc phân tầng của OS/2



Cấu trúc HĐH - Phân tầng (3)

Thuận lợi và khó khăn

■ Thuận lợi:

- Tính module (modularity) → đơn giản hóa trong việc thiết kế, cài đặt, gỡ rối và kiểm tra hệ thống.
- Đơn giản hóa được thể hiện qua việc có thể sửa đổi, cải tiến tại từng tầng, không ảnh hưởng đến các tầng khác.

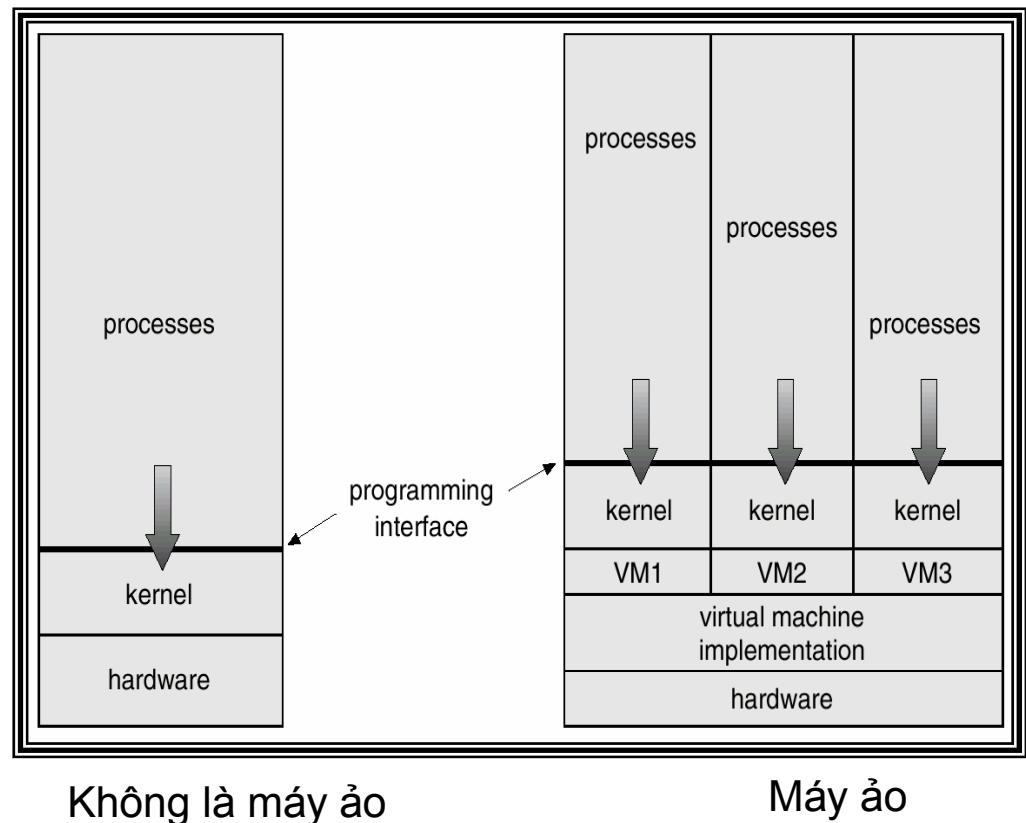
■ Khó khăn:

- Cần phải định nghĩa cẩn thận chức năng các tầng vì mỗi tầng chỉ có thể sử dụng các tầng dưới nó.
- Đôi khi khó khăn trong việc xác định một chức năng của HĐH nằm tại tầng nào.
- Tăng phí tổn về thời gian cho việc gọi các lời gọi hệ thống thông qua nhiều tầng.

Cấu trúc HĐH - Máy ảo (1)

(Virtual Machine)

- Máy ảo sử dụng mô hình phân tầng. Nó xem phần cứng và nhân HĐH đều là phần cứng.
- Một máy ảo cung cấp giao diện đồng nhất cho phần cứng thực bên dưới.
- Tài nguyên thực của máy tính được chia sẻ để tạo ra các máy ảo
- HĐH tạo ảo giác nhiều máy (nhiều quá trình), mỗi máy sử dụng phần CPU và bộ nhớ ảo của nó giống như hệ thống truyền thống.



Cấu trúc HĐH - Máy ảo (2)

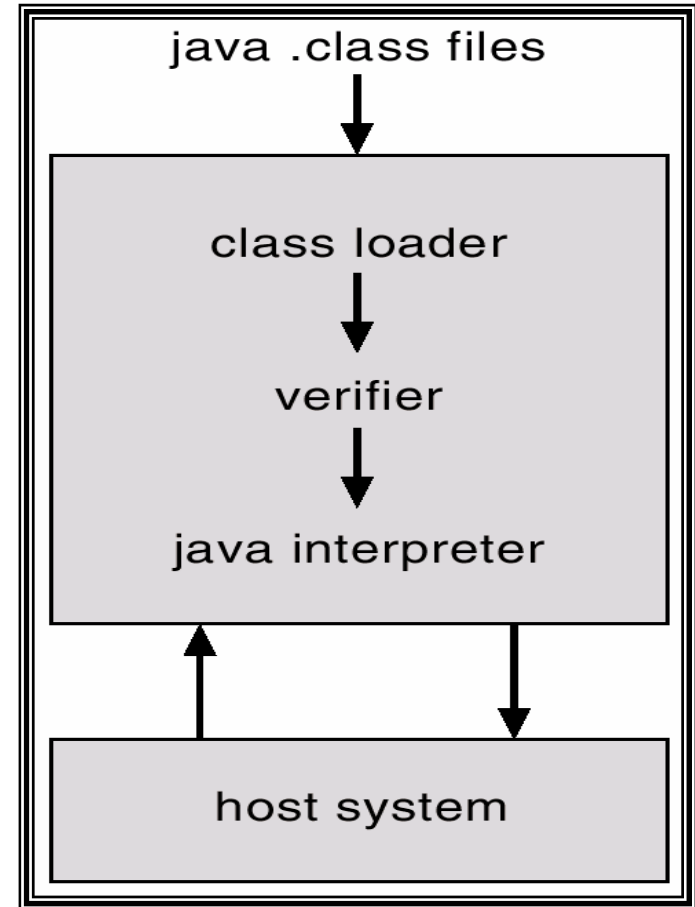
Ưu điểm và nhược điểm của máy ảo

- Máy ảo cung cấp cơ chế bảo vệ tuyệt đối các tài nguyên hệ thống do mỗi máy ảo được tách biệt khỏi tất cả các máy ảo khác. Tuy nhiên, sự tách biệt này không cho phép việc chia sẻ trực tiếp tài nguyên.
- Hệ thống máy ảo là công cụ hoàn hảo cho việc nghiên cứu và phát triển HĐH.
 - Việc phát triển hệ thống được hoàn thành trên máy ảo thay vì trên máy tính vật lý thật, vì vậy nó không ngắt quãng hoạt động bình thường của hệ thống.
- Máy ảo khó cài đặt do yêu cầu đặt ra là phải cung cấp bản sao chính xác phần cứng của máy thật.

Cấu trúc HĐH - Máy ảo (3)

Java Virtual Machine (JVM)

- Các chương trình Java đã biên dịch là các mã-byte nền trung lập (platform-neutral bytecodes) và được thực hiện bởi JVM.
- JVM bao gồm:
 - class loader
 - class verifier
 - runtime interpreter
- Việc đọc và thông dịch bytecodes khi chạy của JVM sẽ làm giảm hiệu suất của ứng dụng. JIT (Just-In-Time) compilers giúp tăng hiệu suất
 - Biến đổi architecture-neutral bytecodes thành ngôn ngữ máy



Cài đặt HĐH (1)

Mục tiêu thiết kế hệ thống

- Mục tiêu người dùng: HĐH phải dễ dùng, dễ học, tin cậy, an toàn và nhanh.
- Mục tiêu hệ thống: HĐH phải dễ thiết kế, cài đặt, bảo trì, cũng như phải linh hoạt, tin cậy, không lỗi và hiệu quả.

Cài đặt HĐH (2)

Các cơ chế và chiến lược

- Cơ chế (mechanism) xác định cách thực hiện một việc
- Chiến lược (policy) quyết định cái gì sẽ được làm
- Việc tách chiến lược ra khỏi cơ chế cho phép đạt được sự linh hoạt tối đa
 - Các quyết định về chiến lược có thể thay đổi sau đó
 - HĐH hành dạng Microkernel cài đặt một tập hợp các hàm cơ (primitive) cho phép tạo các khối cho hầu hết các chiến lược
- Các quyết định về chiến lược phải được xác lập cho tất cả các vấn đề về cấp phát tài nguyên (resource allocation) và định thời (scheduling)

Cài đặt HĐH (3)

Cài đặt hệ thống

- Thay vì được viết bằng hợp ngữ theo truyền thống, ngày nay HĐH có thể được viết bằng các ngôn ngữ lập trình cấp cao.
- Mã lệnh được viết ở ngôn ngữ cấp cao:
 - Có thể được viết nhanh hơn.
 - Gọn gọn hơn.
 - Dễ hiểu và sửa lỗi hơn.
- Một HĐH có thể được chuyển đổi (sang hệ thống phần cứng khác) dễ dàng hơn nhiều nếu nó được viết bằng ngôn ngữ lập trình cấp cao.

Tạo môi trường hệ thống (SYSGEN – System Generation)

- HĐH được thiết kế để có thể chạy trên bất kỳ dạng máy
 - Hệ thống phải được cấu hình cho mỗi dạng máy tính cụ thể - một quá trình được gọi là SYSGEN (System Generation).
- SYSGEN nhận các thông tin về cấu hình cụ thể của phần cứng hệ thống và các tùy chọn cụ thể của HĐH
 - CPU, bộ nhớ, thiết bị
 - HĐH sẽ làm cho tương hợp bằng cách tái biên dịch
- Booting: khởi động máy tính bằng cách nạp kernel
 - Chương trình Bootstrap: các mã lệnh được trữ trong RAM. Nó có thể định vị kernel, nạp nó vào bộ nhớ và cho thực thi