



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÀI GIẢNG MÔN

HỆ ĐIỀU HÀNH

Giảng viên:

Bộ môn:

Email:

TS. Đào Thị Thúy Quỳnh

Khoa học máy tính - Khoa CNTT1

quynhdao.ptit@gmail.com

1. Silberschatz A., Galvin G., Operating systems concepts, 9th ed, John Willey&Sons, 2013
2. Từ Minh Phương, Bài giảng Hệ điều hành
3. Hà Quang Thụy. Nguyên lý các hệ điều hành. Nxb KHKT 2009
4. Nguyễn Thanh Tùng. Giáo trình hệ điều hành. ĐHBK HN 1999

- ❖ Điểm chuyên cần: 10%
- ❖ Điểm trung bình kiểm tra: 10%
- ❖ Điểm thực hành: 10%
- ❖ Thi cuối kỳ: 70%

1. Chương 1: Giới thiệu chung
2. Chương 2: Hệ thống file
3. Chương 3: Quản lý bộ nhớ
4. Chương 4: Quản lý tiến trình

CHƯƠNG 1: GIỚI THIỆU CHUNG

1. Các thành phần của hệ thống máy tính
2. Khái niệm hệ điều hành
3. Các dịch vụ do HDH cung cấp
4. Giao diện lập trình của HDH
5. Quá trình phát triển và một số khái niệm quan trọng
6. Cấu trúc HDH
7. Một số HDH cụ thể

- **Phần cứng:** cung cấp các tài nguyên cần thiết cho việc tính toán, xử lý dữ liệu
- **Phần mềm:** các chương trình cụ thể. (phần mềm hệ thống và phần mềm ứng dụng),
- HDH: phần mềm đóng vai trò trung gian giữa **phần cứng** và **người sử dụng** chương trình ứng dụng, làm cho việc sử dụng hệ thống máy tính được *tiện lợi* và *hiệu quả*

Người sử dụng
Chương trình ứng dụng, chương trình hệ thống và tiện ích
Hệ điều hành
Phần cứng

- Được định nghĩa thông qua mục đích, vai trò, và chức năng trong hệ thống máy tính
 - HĐH là hệ thống phần mềm đóng vai trò trung gian giữa người sử dụng và phần cứng của máy tính nhằm thực hiện 2 chức năng cơ bản:
 - Quản lý tài nguyên
 - Quản lý việc thực hiện các chương trình
- => Một cách thuận lợi và hiệu quả!

II. KHÁI NIỆM HỆ ĐIỀU HÀNH

1. Quản lý tài nguyên

- Đảm bảo cho tài nguyên hệ thống được sử dụng một cách có ích và hiệu quả
- Các tài nguyên: bộ xử lý (CPU), bộ nhớ chính, bộ nhớ ngoài (các đĩa), các thiết bị vào ra
- Phân phối tài nguyên cho các ứng dụng hiệu quả:
 - Yêu cầu tài nguyên được HDH thu nhận và đáp ứng bằng cách cấp cho chương trình các tài nguyên tương ứng
 - HDH cần lưu trữ tình trạng tài nguyên
- Đảm bảo không xâm phạm tài nguyên cấp cho chương trình khác
- Ví dụ: Lưu trữ thông tin trên đĩa => HDH cần biết những vùng nào trên đĩa chưa được sử dụng để ghi thông tin lên những vùng này. Việc ghi thông tin cũng cần tính toán sao cho quá trình truy cập khi cần có thể thực hiện nhanh nhất.

II. KHÁI NIỆM HỆ ĐIỀU HÀNH

2. Quản lý việc thực hiện các chương trình

- Nhiệm vụ quan trọng nhất của máy tính là thực hiện các chương trình, 1 chương trình đang trong quá trình chạy gọi là tiến trình (process).
- Chương trình cần được quản lý để thực hiện thuận lợi, tránh lỗi, đồng thời đảm bảo môi trường để việc xây dựng và thực hiện chương trình được thuận lợi.
- Để chạy chương trình cần thực hiện một số thao tác nhất định => HĐH giúp việc chạy chương trình dễ dàng hơn, người dùng không cần phải thực hiện thao tác

II. KHÁI NIỆM HỆ ĐIỀU HÀNH

2. Quản lý việc thực hiện các chương trình

- Để tạo môi trường thuận lợi cho chương trình, hđh tạo ra các máy ảo:
 - Là máy logic với các tài nguyên ảo
 - Tài nguyên ảo mô phỏng tài nguyên thực được thực hiện bằng phần mềm
 - Cung cấp các dịch vụ cơ bản như tài nguyên thực
 - Dễ sử dụng hơn, số lượng tài nguyên ảo có thể lớn hơn số lượng tài nguyên thực.
- Một số máy ảo tốt nhất hiện nay: VirtualBox (Windows/Mac/Linux); Parallels (Windows/Mac/Linux); VMware (Windows/Linux, Basic); QEMU (Linux); Boot Camp (Windows/macOS); Windows Virtual PC (Windows).

III. CÁC DỊCH VỤ DO HDH CUNG CẤP

- Một trong những nhiệm vụ chủ yếu của HDH là tạo ra môi trường thuận lợi cho các chương trình khác thực hiện và giúp người sử dụng hệ thống dễ dàng.
- Các dịch vụ có thể thay đổi theo từng HDH. Một số HDH có thể cung cấp nhiều dịch vụ khi hệ điều hành khác có thể cung cấp ít dịch vụ hơn.
 - Ví dụ như MS-DOS không cung cấp dịch vụ về bảo mật trong khi Windows NT lại rất chú trọng tới dịch vụ này.
- Một số dịch vụ thường gặp của hệ điều hành =>

- **Tải và chạy chương trình:**
- Để thực hiện, chương trình được tải từ đĩa vào bộ nhớ, sau đó được trao quyền thực hiện các lệnh. Khi thực hiện xong, cần giải phóng bộ nhớ và các tài nguyên
- Toàn bộ quá trình này tương đối phức tạp song lại diễn ra thường xuyên.
- => HDH sẽ thực hiện công việc phức tạp và lặp đi lặp lại này
- Do HDH là chương trình đầu tiên được thực hiện khi khởi động hệ thống nên HDH tự tải mình vào bộ nhớ
- Nhờ có HDH, lập trình viên, người sử dụng không cần quan tâm chi tiết đến việc tải và chạy chương trình.

III. CÁC DỊCH VỤ DO HDH CUNG CẤP

- **Giao diện với người dùng:** cho phép giao tiếp giữa HDH và người dùng:
 - Dưới dạng dòng lệnh (command-line): cho phép người dùng chỉ thị cho HDH bằng cách gõ lệnh dưới dạng văn bản. Ví dụ: chtr cmd.exe của Windows.
 - Giao diện đồ họa (Graphic User Interface-GUI): sử dụng hệ thống cửa sổ, thực đơn và thiết bị trỏ chuột, kết hợp với bàn phím để giao tiếp với hệ thống.
- **Thực hiện các thao tác vào/ ra dữ liệu:**
 - Người dùng và chương trình trong khi thực hiện có thể có nhu cầu I/O dữ liệu với các đĩa và thiết bị ngoại vi. Để tránh cho chương trình không phải làm việc với phần cứng, yêu cầu I/O sẽ được giao cho hệ điều hành thực hiện.

III. CÁC DỊCH VỤ DO HDH CUNG CẤP

- **Làm việc với hệ thống file:** nhu cầu đọc, ghi, tạo, xóa, chép file hoặc làm việc với thư mục; quản lý quyền truy cập, sao lưu.
- **Phát hiện và xử lý lỗi:**
 - Phát hiện và xử lý kịp thời các lỗi xuất hiện trong phần cứng cũng như phần mềm
 - => Đảm bảo cho hệ thống hoạt động ổn định, an toàn
 - Ví dụ: các lỗi phần cứng như hết bộ nhớ, mất điện, máy in hết mực, hết giấy,...
- **Truyền thông:**
 - Cung cấp dịch vụ cho phép thiết lập liên lạc và truyền thông tin dưới dạng thông điệp hoặc qua BN dùng chung.

III. CÁC DỊCH VỤ DO HDH CUNG CẤP

- **Cấp phát tài nguyên:**

- Trong các hệ thống cho phép nhiều chương trình thực hiện đồng thời cần có cơ chế cấp phát và phân phối tài nguyên hợp lý => người dùng và trình ứng dụng không phải tự thực hiện việc cấp phát tài nguyên mà vẫn đảm bảo cấp phát công bằng và hiệu quả.

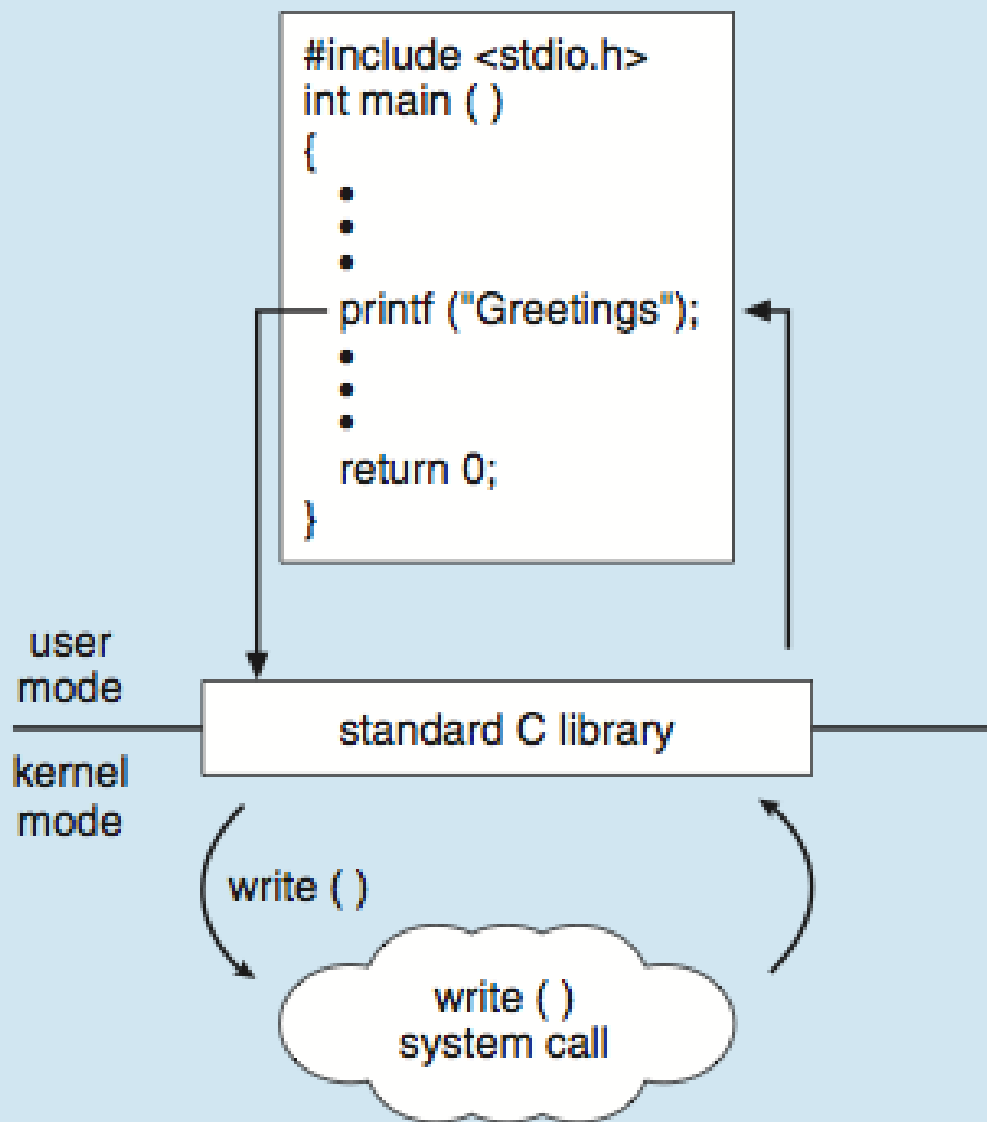
- **Dịch vụ an ninh và bảo mật:**

- Đối với hệ thống nhiều người dùng thường xuất hiện yêu cầu bảo mật thông tin, tức là người dùng này không tiếp cận được thông tin của người khác nếu không được cho phép.
 - Cần đảm bảo để tiến trình không truy cập trái phép tài nguyên (như vùng nhớ, file mở) của tiến trình khác hay chính HDH sẽ thực hiện bằng cách kiểm soát truy cập tới tài nguyên.

IV. GIAO DIỆN LẬP TRÌNH CỦA HDH

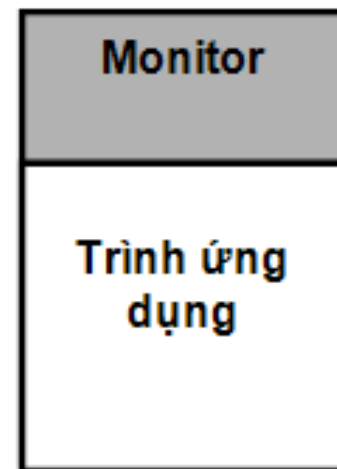
- Để các chương trình có thể sử dụng được những dịch vụ, HDH cung cấp giao diện lập trình.
- Giao diện này bao gồm các lời gọi hệ thống (system call) mà chương trình sử dụng yêu cầu một dịch vụ nào đó từ phía HDH.
- Lời gọi hệ thống: các lệnh đặc biệt mà CTUD gọi khi cần yêu cầu HDH thực hiện một việc gì đó
- Lời gọi hệ thống được thực hiện qua những thư viện hàm gọi là thư viện hệ thống. Các hàm này sẽ giúp người lập trình gọi lời gọi hệ thống tương ứng của hệ điều hành.

IV. GIAO DIỆN LẬP TRÌNH CỦA HDH

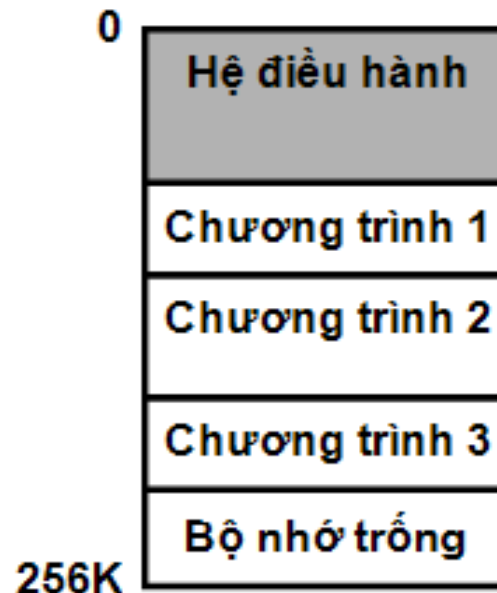


- ❖ Các hệ thống đơn giản (những năm 40-50 của thế kỷ trước): tốc độ xử lý của máy tính rất thấp, việc vào/ra được thực hiện thủ công và khó khăn.
 - ❖ Việc nạp chương trình được thực hiện nhờ công tắc, mạch hàn sẵn, bìa đục lỗ. Trong thời kỳ này, lập trình viên tương tác trực tiếp với phần cứng.
- => Máy tính thời kỳ này chưa có HDH.

- Xử lý theo mẻ:
 - Chương trình được phân thành các *mẻ*: gồm những chương trình có yêu cầu giống nhau
 - Toàn bộ mẻ được nạp vào băng từ và được tải vào máy để thực hiện lần lượt
- *Chương trình giám sát* (monitor): mỗi khi một chương trình của mẻ kết thúc, chương trình giám sát tự động nạp chương trình tiếp theo vào máy và cho phép nó chạy => Giảm đáng kể thời gian chuyển đổi giữa hai chương trình trong cùng một mẻ
- Trình giám sát là dạng đơn giản nhất của HDH
 - Nhược điểm: Mỗi khi có yêu cầu vào/ra, CPU phải dừng việc xử lý dữ liệu để chờ quá trình vào ra kết thúc. Do tốc độ vào/ra thấp hơn tốc độ CPU rất nhiều nên CPU thường xuyên phải chờ đợi 1 thời gian dài. => hiệu suất CPU thấp.

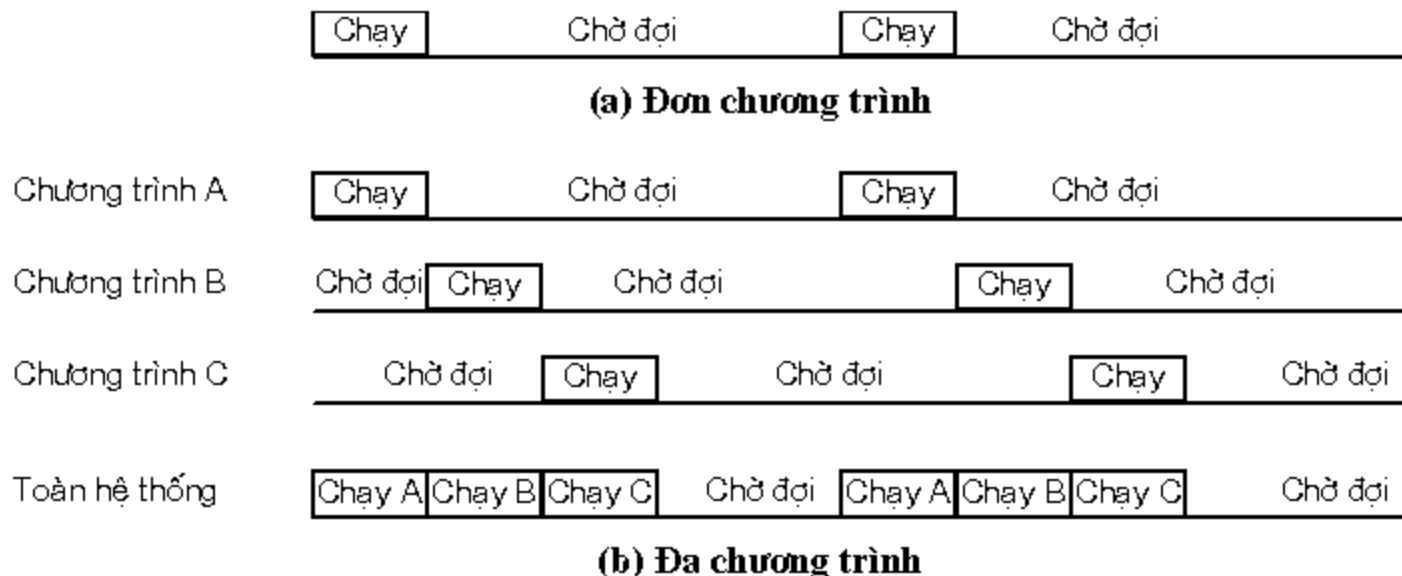


- Đa chương trình (đa nhiệm):
 - Hệ thống chứa đồng thời nhiều chương trình trong bộ nhớ
 - Khi một chương trình phải dừng lại để thực hiện vào ra, HDH sẽ chuyển CPU sang thực hiện một chương trình khác
 - => Giảm thời gian chạy không tải của CPU



V. QUÁ TRÌNH PHÁT TRIỂN

■ Đa chương trình:



- Thời gian chờ đợi của CPU trong chế độ đa chương trình giảm đáng kể so với trong trường hợp đơn chương trình
- HDH phức tạp hơn rất nhiều so với HDH đơn chương trình
- Đòi hỏi hỗ trợ từ phần cứng, đặc biệt khả năng vào/ra bằng ngắt và cơ chế DMA

- Đa chương trình (Hạn chế)
 - Mặc dù đa chương trình cho phép sử dụng hiệu quả CPU và các tài nguyên khác của hệ thống song kỹ thuật này không cho người dùng tương tác với hệ thống.
 - Các máy tính thế hệ sau cho phép máy tính và người dùng làm việc trực tiếp thông qua màn hình và bàn phím.
 - Đối với các hệ thống này thì thời gian từ khi người dùng gõ lệnh cho tới khi máy tính phản xạ lại tương đối nhỏ.
 - Kỹ thuật đa chương trình không đảm bảo được thời gian đáp ứng ngắn như vậy.

■ Chia sẻ thời gian:

- Chia sẻ thời gian có thể coi như đa chương trình cải tiến
- CPU lần lượt thực hiện các công việc khác nhau trong những khoảng thời gian ngắn gọi là lượng tử thời gian
- Chuyển đổi giữa các công việc diễn ra với tần số cao và tốc độ CPU lớn
- => Tất cả người dùng đều có cảm giác máy tính chỉ thực hiện chương trình của mình
- => CPU được chia sẻ giữa những người dùng khác nhau tương tác trực tiếp với hệ thống

- Quản lý tiến trình:
 - Một chương trình đang trong quá trình thực hiện được gọi là tiến trình.

Chương trình	Tiến trình
<ul style="list-style-type: none">- Là một thực thể tĩnh- Được ghi dưới dạng những bit, những byte trên đĩa	<ul style="list-style-type: none">- Là một thực thể động- Đang tiến hành việc xử lý, tính toán, được cung cấp một số tài nguyên: thời gian CPU, bộ nhớ

- Quản lý tiến trình:
 - Tạo và xoá tiến trình (bao gồm cả tiến trình người dùng và tiến trình hệ thống)
 - Tạm treo và khôi phục các tiến trình bị treo
 - Đồng bộ hoá các tiến trình (lập lịch cho các tiến trình .v.v.)
 - Tạo cơ chế liên lạc giữa các tiến trình
 - Giải quyết các **bế tắc**, ví dụ như khi có xung đột về tài nguyên
 - Bế tắc: là chương trình đang cần tài nguyên nhưng nó chờ đợi mà không được cung cấp.

- Quản lý bộ nhớ :
 - Quản lý, cung cấp và giải phóng
 - **Cung cấp** và **giải phóng** bộ nhớ theo yêu cầu của các tiến trình
 - Quản lý không gian nhớ đã được cấp và không gian còn trống
 - Quản lý việc phân phối bộ nhớ giữa các tiến trình => đảm bảo việc chạy song song giữa nhiều chương trình
 - Tạo ra bộ nhớ ảo và ánh xạ địa chỉ bộ nhớ ảo vào bộ nhớ thực

- Quản lý hệ thống vào ra:
 - Quản lý thông qua các chương trình điều khiển
 - Đơn giản hoá và tăng hiệu quả quá trình trao đổi thông tin giữa các tiến trình với thiết bị vào ra
- Quản lý file và thư mục:
 - Tạo, xóa file và thư mục
 - Đọc ghi file
 - Ánh xạ file và thư mục sang bộ nhớ ngoài

- Hỗ trợ mạng và xử lý phân tán
 - Quản lý thiết bị mạng
 - Hỗ trợ các giao thức truyền thông
 - Quản lý truyền thông, cân bằng tải

=> Thông qua các thành phần điều khiển, giao tiếp mạng.
- Giao diện với người dùng:
 - Đó là hệ thống thông dịch lệnh
 - Giúp cho máy tính hiểu và xử lý được các chỉ thị, các lệnh của người dùng.
 - Ví dụ: bash của Linux, command của window
- Các chương trình tiện ích và ứng dụng

VI. CẤU TRÚC HDH

2. NHÂN CỦA HDH

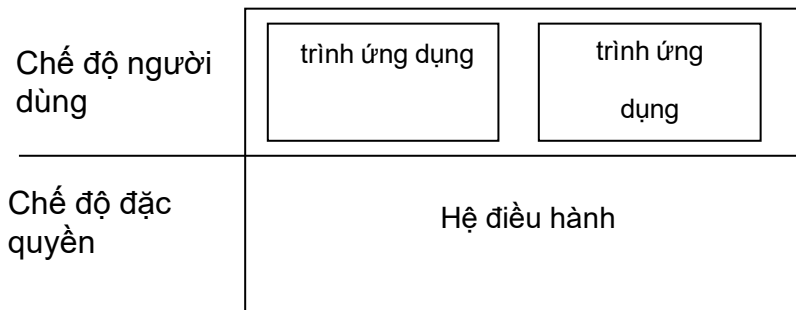
- HDH gồm rất nhiều thành phần, tuy nhiên độ quan trọng của các tp khác nhau, có những tp không thể thiếu là cơ sở cho toàn hệ thống hoạt động, một số tp của HDH cung cấp chức năng kém quan trọng hơn.
- => chỉ tải những thành phần quan trọng không thể thiếu được vào bộ nhớ gọi là nhân.
- *Nhân (kernel) là phần cốt lõi, thực hiện các chức năng cơ bản nhất, quan trọng nhất của HDH và thường xuyên được giữ trong bộ nhớ*
- *Kernel có nhiệm vụ quản lý tài nguyên hệ thống (liên lạc giữa các thành phần phần cứng và phần mềm)*

- Máy tính hiện đại thường được thiết kế với hai chế độ thực hiện chương trình.
 - Nhân chạy trong chế độ đặc quyền – chế độ nhân: là chế độ mà chương trình thực hiện trong đó có đầy đủ quyền truy cập và điều khiển phần cứng máy tính.
 - Chế độ người dùng: chương trình thực hiện trong chế độ người dùng bị hạn chế rất nhiều quyền truy cập và sử dụng phần cứng.
- Việc phân biệt chế độ nhân và chế độ người dùng nhằm mục đích ngăn không cho CTUD vô tình hoặc cố ý thực hiện những thao tác làm ảnh hưởng tới hệ thống.

VI. CẤU TRÚC HDH

3. MỘT SỐ CẤU TRÚC HDH

- Cấu trúc nguyên khối
 - Toàn bộ chương trình và dữ liệu của HĐH có chung 1 không gian nhớ. Do vậy, có thể coi là một khối duy nhất.
 - HĐH trở thành một tập hợp các thủ tục hay các chương trình con
 - Ưu điểm: nhanh, không mất thời gian giữa các không gian nhớ
 - Nhược điểm: Không an toàn: khi bất kỳ thành phần nào có sự cố thì toàn bộ hệ thống sẽ không hoạt động đc; Ko mềm dẻo và khó sửa đổi, thêm bớt tp nào sẽ ảnh hưởng tới toàn bộ hệ thống, khi có lỗi khó xác định lỗi do tp nào gây ra.

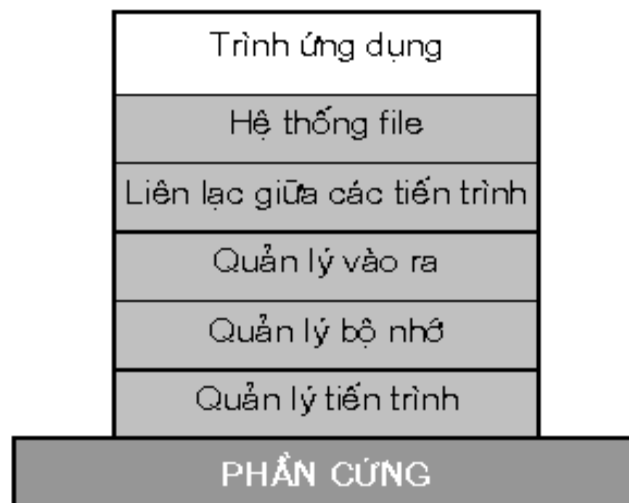


Hình :Cấu trúc nguyên khối

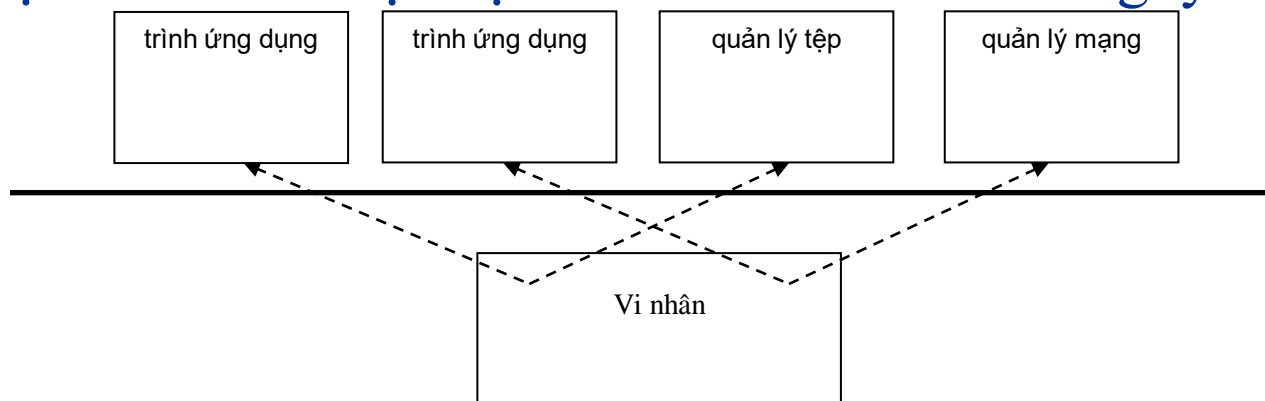
3. MỘT SỐ KIỂU CẤU TRÚC HDH

■ Cấu trúc phân lớp

- Các thành phần được chia thành các lớp nằm chồng lên nhau
- Mỗi lớp chỉ có thể liên lạc với lớp nằm kề bên trên và kề bên dưới
- Mỗi lớp chỉ có thể sử dụng dịch vụ do lớp nằm ngay bên dưới cung cấp
- Ưu điểm: chia nhỏ chức năng, dễ sử dụng, dễ sửa lỗi
- Nhược điểm: khó thiết kế (xác định số lớp, phân chia thành phần các chức năng của mỗi lớp), tốc độ chậm hơn cấu trúc nguyên khối



- Cấu trúc vi nhân (micro kernel)
 - Nhân có kích thước nhỏ, chỉ chứa các chức năng quan trọng nhất
 - Các chức năng còn lại được đặt vào các modul riêng: chạy trong chế độ đặc quyền hoặc người dùng. Khi có yêu cầu từ ứng dụng, nhân sẽ chuyển cho module tương ứng để xử lý và nhận lại kết quả, nhân chủ yếu đóng vai trò trung gian liên lạc.
 - Ưu điểm: mềm dẻo, an toàn
 - Nhược điểm: tốc độ chậm hơn so với cấu trúc nguyên khối



Hình 1.5 Cấu trúc vi nhân

■ UNIX

- Là một hệ điều hành đa nhiệm, được phát triển đầu tiên bởi Ken Thompson, Dennis Ritchie và Douglas McIlroy tại AT & T Bell.
- UNIX được nghiên cứu tại các phòng thí nghiệm năm 1969 và dần cải tiến, phát triển và trở nên phổ biến. Unix lần đầu tiên được lập trình lại bởi Ken Thompson bằng ngôn ngữ C vào năm 1973.
- Tạo ra ngôn ngữ cấp cao trong các hệ điều hành
- Tạo ra hệ thống tập tin phân cấp
- Unix shell đã truyền cảm hứng cho nhiều trình thông dịch dòng lệnh phát triển sau đó.
- Giúp ngôn ngữ lập trình C trở nên phổ biến hơn
- Góp phần vào sự ra mắt của phong trào phần mềm miễn phí

- MINIX (Từ mini-Unix)
 - Là một hệ điều hành máy tính tựa Unix dựa trên kiến trúc micro-kernel.
 - Phiên bản đầu của MINIX được tạo ra bởi Andrew S. Tanenbaum cho mục đích giáo dục như minh họa, phục vụ đào tạo, có thể sử dụng miễn phí.
 - MINIX bây giờ phát triển như là phần mềm nguồn mở.

■ LINUX

- Vào năm 1991 trong khi đang học tại Helsinki - Phần Lan, Linus Torvalds bắt đầu có ý tưởng về một hệ điều hành.
- Do ông cũng nhận thấy hạn chế trong giấy phép của Minix - chỉ cho phép việc sử dụng Minix trong giáo dục mà thôi. Ông bắt đầu viết nên hệ điều hành LINUX phát triển từ MINIX.

■ MS-DOS

- Là sản phẩm của hãng Microsoft và được trang bị cho các máy PC đầu tiên của IBM
- Để có thể chạy trên PC với tài nguyên hạn chế, MS-DOS được xây dựng đơn giản và ít chức năng hơn
- Nhiều giải pháp kỹ thuật trong MS-DOS có nguồn gốc từ UNIS như giao diện lập trình (lời gọi hệ thống), cấu trúc phân cấp của thư mục, bộ dịch lệnh
- Không có các chức năng như bảo mật, hỗ trợ mạng, hỗ trợ nhiều tiến trình

VII. MỘT SỐ HĐH CỤ THỂ

- Windows NT (NT-new technology) là một thành viên của họ điều hành thể hệ mới như Windows 2000, XP, Vista,7.
- Phiên bản đầu tiên được phát hành vào năm 1993
- Đây là hđh sử dụng nhiều kỹ thuật tiên tiến trong lĩnh vực hđh đã được phát triển cho đến thời điểm này gồm các giải pháp lấy từ UNIX
- Là một hệ điều hành đa nhiệm, hỗ trợ mạng, có các chức năng bảo mật, có giao diện đồ họa dưới dạng cửa sổ và được dùng cho cả máy PC yêu cầu độ ổn định cao.