Chương 2. Cấu Trúc Hệ Điều Hành

Mục Tiêu

Giới thiệu các **dịch vụ mà hệ điều hành cung cấp** và các **phương pháp thiết kế các kiến trúc** và **cài đặt** hệ điều hành.

Nội Dung

Các thành phần của Hệ điều hành

Các dịch vụ của Hệ điều hành

Lời gọi hệ thống

Các chương trình hệ thống

Kiến trúc hệ điều hành

Thiết kế & Cài đặt hệ điều hành

Các Thành Phần Của Hệ Điều Hành

- Là một hệ thống phức tạp bao gồm nhiều thành phần với input, output và chức năng được định nghĩa rõ ràng:
  1. Quản lý tiến trình (process management)
  2. Quản lý bộ nhớ chính (main-memory management)
  3. Quản lý hệ thống tập tin (file management)
  4. Quản lý hệ thống nhập/xuất (I/O management)
  5. Quản lý hệ thống lưu trữ thứ cấp (secondary storage management)
  6. Hệ thống kết nối mạng (networking)
  7. Hệ thống bảo vệ (protection system)
  8. Giao diện người dùng (user interface)

Quản Lý Tiến Trình

- Tiến trình (process) là một chương trình đang thực thi.
- Tiến 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 tiến trình chịu trách nhiệm thực hiện các tác vụ sau:
  - Tạo và hủy tiến trình.
  - Ngừng và tiếp tục tiến trình.
  - Đưa ra các cơ chế để:
    - đồng bộ hóa các tiến trình.
    - thực hiện việc giao tiếp giữa các tiến trình.
    - chống deadlock.

Quản Lý Bộ Nhớ Chính

- 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 chịu trách nhiệm thực hiện các tác vụ:
  - Theo dõi phần nào của bộ nhớ đang được sử dụng bởi tiến trình nào.
  - Quyết định tiến 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.

Quản Lý Tập Tin

- Một tập tin:
  - là một tập hợp các thông tin có liên quan với nhau,
  - dùng để lưu các chương trình hoặc dữ liệu trong các thiết bị lưu trữ, như đĩa, băng từ.
- Bộ quản lý tập tin chịu trách nhiệm thực hiện các tác vụ:
  - Tạo và xóa tập tin, thư mục.
  - 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.
  - Sao lưu tập tin lên các phương tiện lưu trữ không bay hơi.

Quản Lý Hệ Thống Nhập/Xuất

- 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.
- Thành phần quản lý hệ thống xuất/nhập giao tiếp với các thành phần khác của hệ thống để quản lý các thiết bị, chuyển tải dữ liệu, và phát hiện các hoàn thành xuất/nhập.

Quản Lý Hệ Thống Lưu Trữ Thứ Cấp

- Bộ nhớ chính bị bay hơi và quá nhỏ để chứa tất cả dữ liệu và chương trình lâu dài
  $\Rightarrow$ dùng thiết bị lưu trữ thứ cấp 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 chịu trách nhiệm thực hiện các tác vụ:
  - Quản lý không gian còn trống
  - Cấp phát không gian lưu trữ
  - Định thời sử dụng đĩa

Hệ Thống Nối Kết Mạng – Phân Tán

- Hệ thống phân tán là tập hợp các bộ xử lý không dùng chung bộ nhớ hoặc xung đồng hồ.
- 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: FTP, NFS, HTTP
- Hệ thống phân tán cho phép người dùng truy cập nhiều loại tài nguyên hệ thống khác nhau, giúp:
  - Tăng tốc độ tính toán
  - Tăng mức độ sẵn dùng của dữ liệu
  - Tăng độ tin cậy

Hệ Thống Bảo Vệ

- Khái niệm bảo vệ nhằm ám chỉ cơ chế điều khiển truy cập từ các chương trình, tiến 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 có nguy cơ bị chiếm bất hợp pháp
  - cung cấp các phương tiện để bảo vệ an ninh

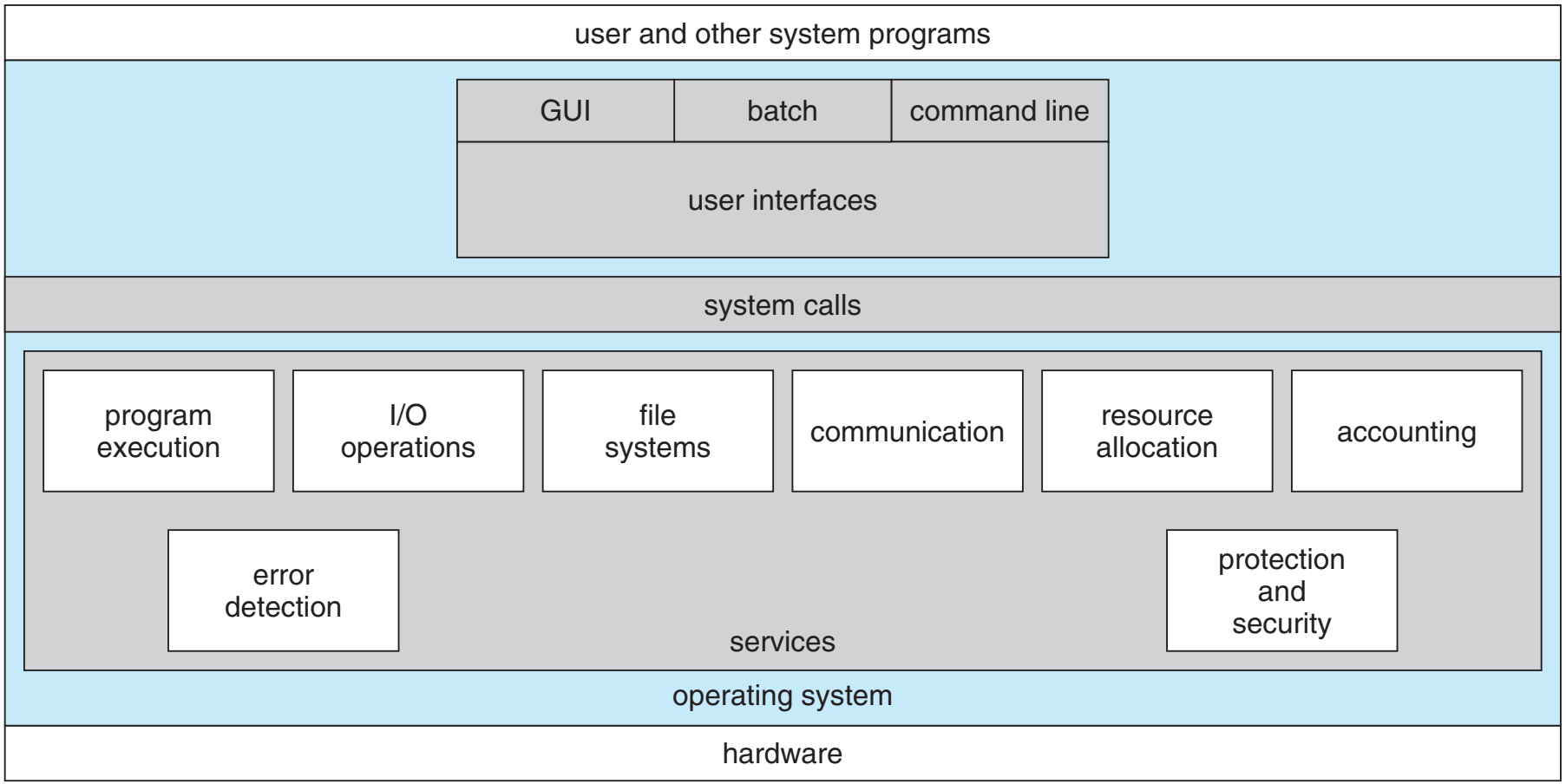
Thông Dịch Lệnh (Command-line Interpreter)

- Nhận và thực hiện các câu lệnh điều khiển của người dùng để thực hiện các tác vụ như: quản lý tiến trình, quản lý vào/ra, quản lý bộ nhớ, truy cập hệ thống tập tin, ...
- Được cài đặt trong kernel (DOS) hoặc qua các chương trình hệ thống (Widows, Unix), còn được gọi là shell.
- Có 2 loại lệnh cơ bản: các lệnh cung cấp bởi shell (built-in) hay tên của một chương trình.
  - Sử dụng các lệnh cung cấp qua chương trình cho phép thêm các lệnh vào hệ thống mà không cần phải cập nhật lại shell.
  - Dung lượng shell nhỏ.

Môi trường nền (Desktop Environment)

- Giao diện người dùng theo dạng đồ họa (GUI): Windows DE, GNOME DE, KDE.
- Môi trường nền điển hình cung cấp các icons, windows, toolbars, folders, wallpapers, và khả năng drag and drop.
- Môi trường nền 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 Hệ Điều Hành



Dịch Vụ Cho Chương Trình & Người Dùng

- Giao diện người dùng: command line, batch interface, GUI
- Thực thi chương trình: nạp chương trình vào bộ nhớ và thực thi
- Thao tác I/O: cung cấp các phương tiện để thực hiện các 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 tiến trình đang thực thi trên cùng một máy tính hoặc trên nhiều hệ thống được kết nối với nhau qua mạng máy tính (dùng p/p bộ nhớ chia sẻ hoặc chuyển thông điệp)
- Phát hiện lỗi: phát hiện lỗi phát sinh tại CPU và bộ nhớ, tại thiết bị I/O hoặc tại chương trình người dùng để bảo đảm tính toán chính xác

Dịch Vụ Cho Hệ Thống

- Một số chức năng không nhằm hỗ trợ người dùng mà dùng để đảm bảo cho hoạt động hiệu quả của hệ thống bao gồm:
  - 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

- Là giao diện giữa tiến trình và hệ điều hành, dùng để gọi các dịch vụ của HĐH.
- Về cơ bản, được hỗ trợ dưới dạng các chỉ thị assembler.
- Các lời gọi hệ thống còn được cài đặt bằng các ngôn ngữ cấp cao hơn (C, C++), gọi là các giao diện lập trình ứng dụng (API)
- Một số API phổ biến:
  - Windows API (cho HĐH Windows)
  - POSIX API (cho POSIX-Based systems như Linux, Unix, MacOS)
  - Java API (cho Java Virtual Machine)

Lời Gọi Hệ Thống – Ví Dụ

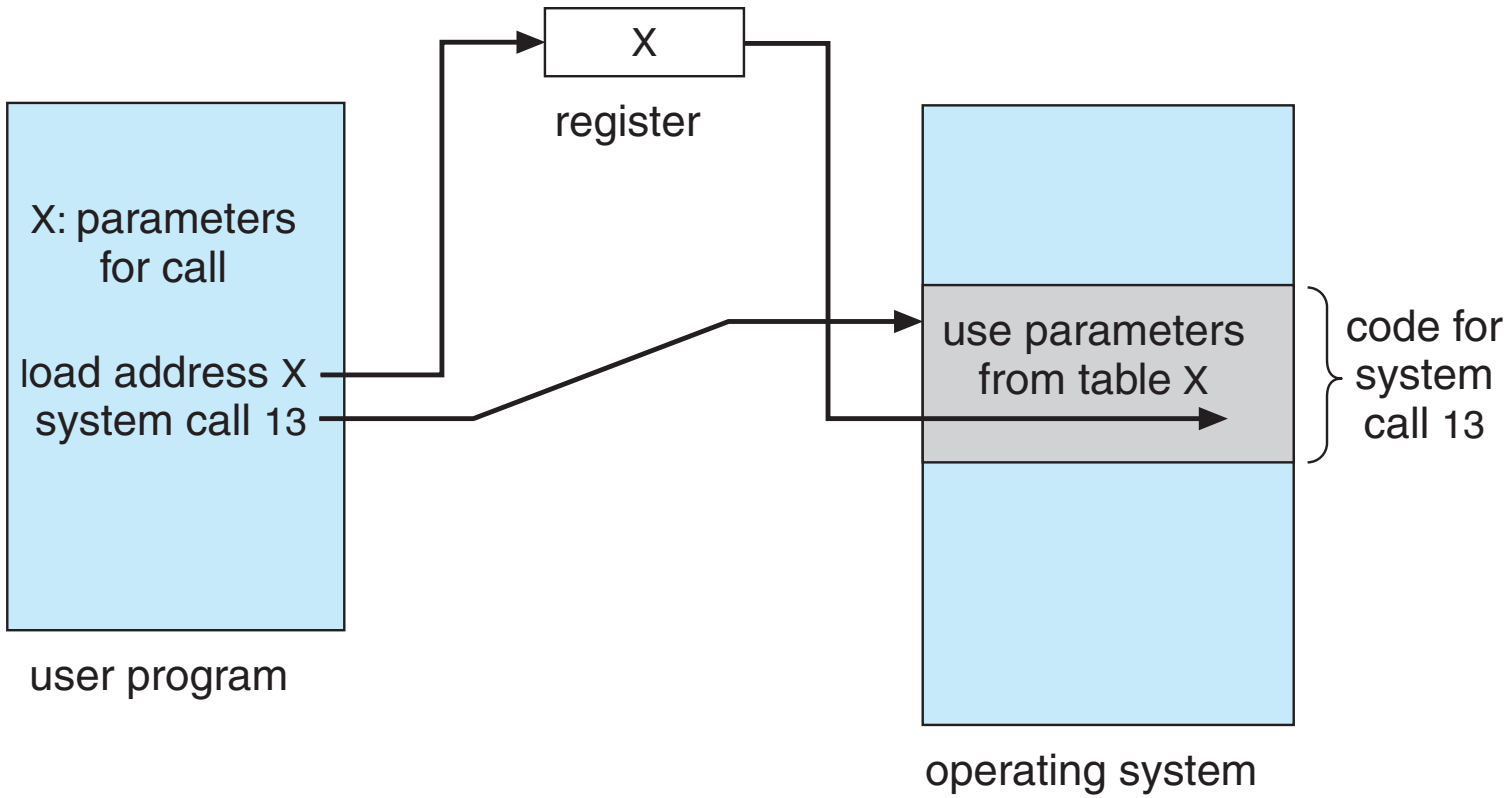
Acquire input filename
  Write prompt to screen
  Accept input
Acquire output filename
  Write prompt to screen
  Accept output
Check existence of input file
  if input file doesn't exist
    Write prompt to screen, abort
Check existence of output file
  if output file exists
    Write prompt to screen, abort
Copy input file to output file
Terminate normally

```
echo -n "Source filename: "
  read src
echo -n "Target filename: "
  read targ
if [ ! -f $src ] then
  echo ”Error, $src doesn’t exist”
  exit 1
elif [ -f $targ ] then
  echo “Error, $targ exist”
  exit 2
fi
cp $src $targ
echo "Sucessfully”
```

Truyền Tham Số Cho Lời Gọi Hệ Thống

- Một lời gọi hệ thống thường kèm theo các tham số.
- Có 3 phương pháp tổng quát để truyền tham số:
  1. Truyền qua thanh ghi: giới hạn số lượng tham số vì số thanh ghi tương đối ít.
  2. Truyền qua bộ nhớ: các tham số được lưu vào 1 bảng hay khối trong bộ nhớ và địa chỉ của bảng/khối được chuyển vào thanh ghi như là 1 tham số.
  3. Truyền bằng stack: chương trình push tham số vào stack và hệ điều hành sẽ lấy tham số bằng cách pop stack.

Truyền Tham Số Qua Bộ Nhớ – Ví Dụ

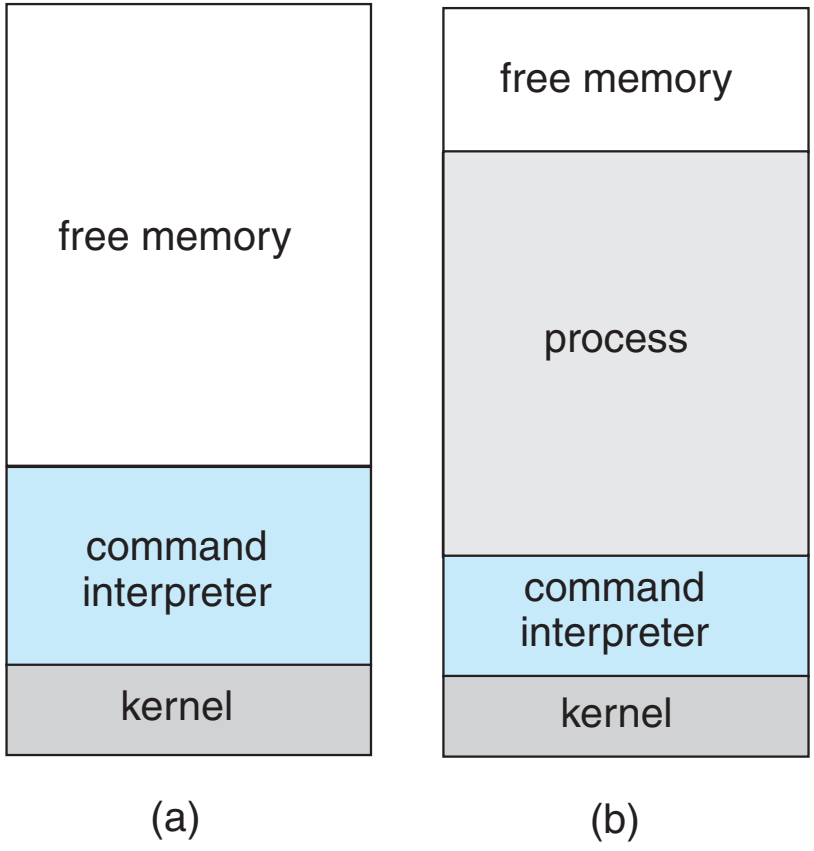


Các Kiểu Lời Gọi Hệ Thống

1. Điều khiển tiến trình (process control)
2. Quản lý file (file management)
3. Quản lý thiết bị (device management)
4. Duy trì thông tin trạng thái (information maintenance)
5. Giao tiếp (communication)
6. Một số HĐH còn cung cấp lời gọi hệ thống để sử dụng các dịch vụ bảo vệ (protection) của HĐH

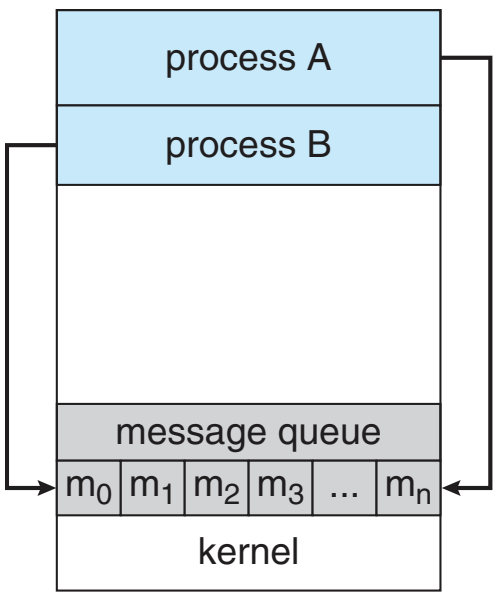
Thực Thi Chương Trình Trong MS-DOS

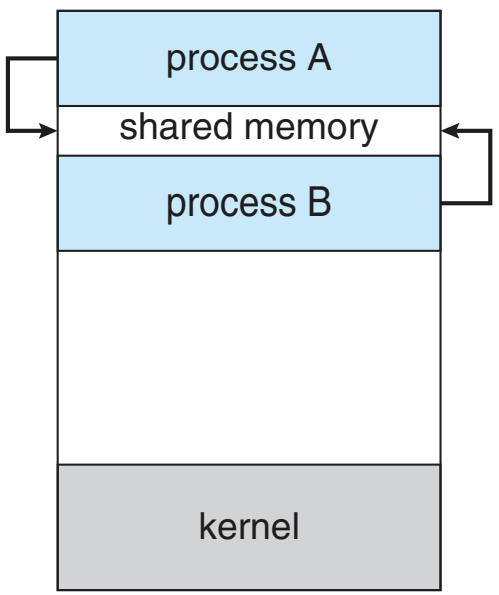
- Thực thi chương trình trong MS-DOS:
  a) Lúc hệ thống khởi động, và
  b) Lúc chạy một chương trình.
- MS-DOS là một chương trình đơn nhiệm: tại 1 thời điểm chỉ tối đa 1 chương trình được thực thi.

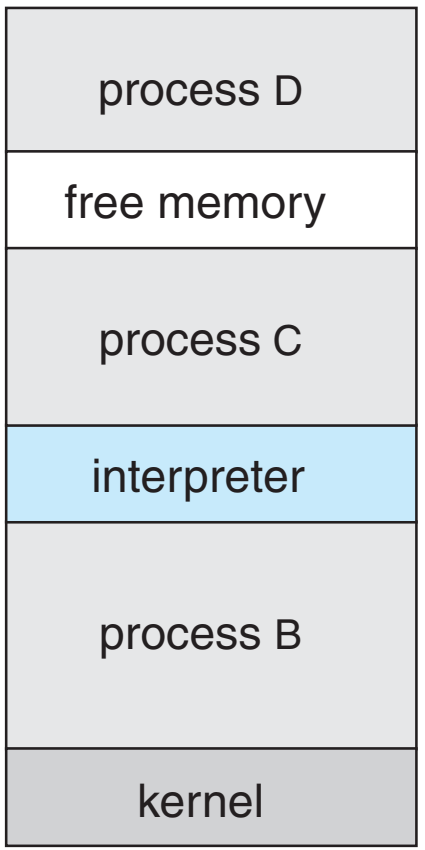


Thực Thi Chương Trình Trong UNIX

- UNIX là một chương trình đa nhiệm: tại 1 thời điểm có thể có nhiều chương trình được thực thi.
- Cần cơ chế thực hiện giao tiếp giữa các tiến trình: chuyển thông điệp hay bộ nhớ chia sẻ.







Các Chương Trình Hệ Thống

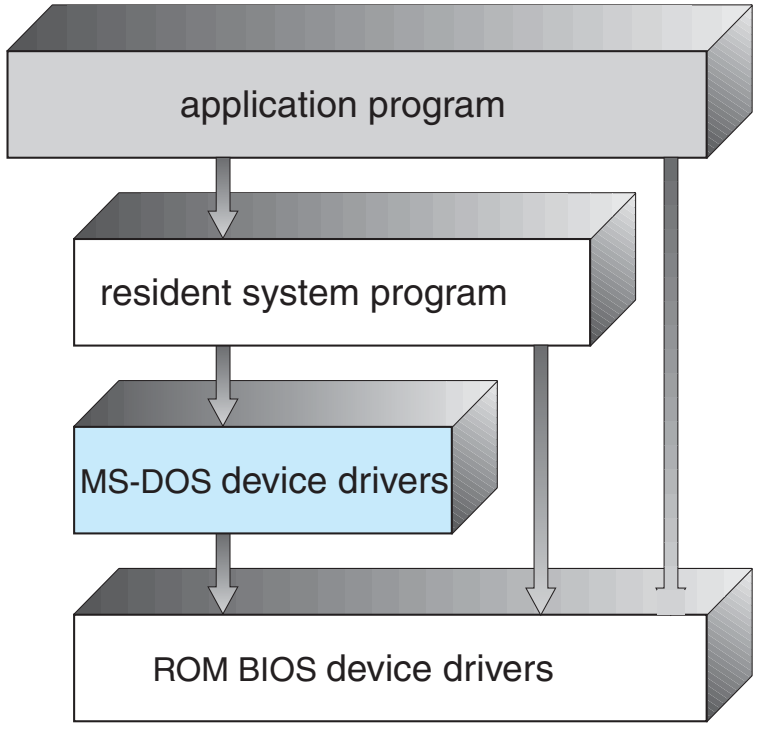
- Cung cấp môi trường thuận lợi cho việc phát triển và thực thi chương trình và đơn giản hóa cho các lời gọi hệ thống đối với người dùng.
- Các loại chương trình hệ thống:
  - Thao tác tập tin và điều chỉnh tập tin
  - Thông tin trạng thái
  - Hỗ trợ ngôn ngữ lập trình
  - Nạp và thực thi chương trình
  - Giao tiếp giữa các tiến trình, người dùng, các máy tính
- Người dùng hầu như nhìn HĐH qua các chương trình hệ thống, không qua các lời gọi hệ thống.

Kiến Trúc Hệ Điều Hành

- Là cách thức tổ chức các thành phần HĐH để xác định đặc quyền mà mỗi thành phần thực hiện.
- Ba loại kiến trúc:
  - Nguyên khối (monolithic): tất cả các thành phần chứa trong nhân (kernel)
  - Phân tầng (layered): phương pháp trên-xuống (top-down), tách biệt các chức năng và các đặc điểm trong các thành phần
  - Vi nhân (microkernel): chỉ những thành phần chủ yếu bao gồm trong kernel

Kiến Trúc Hệ Điều Hành MS-DOS

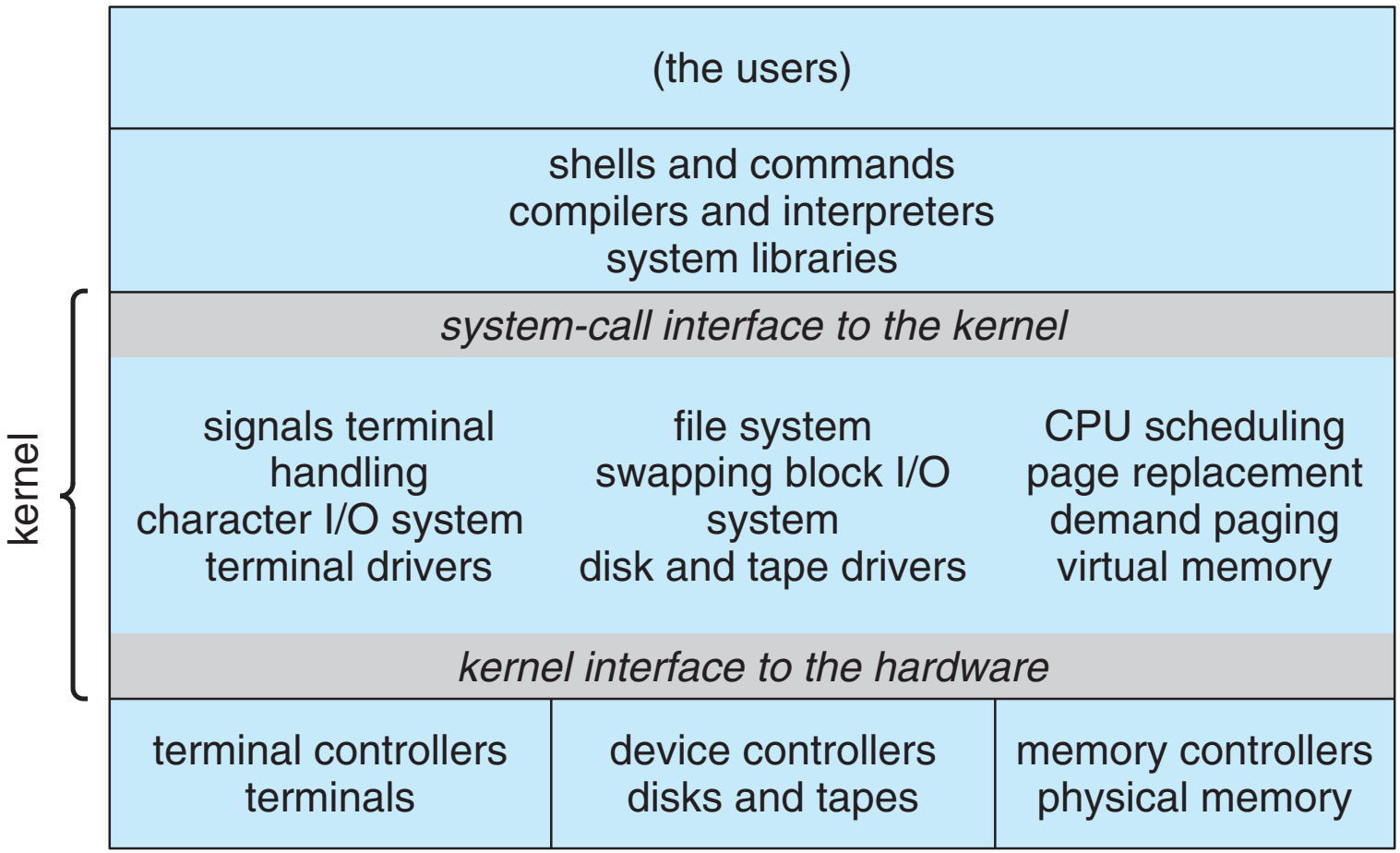
- Không có kiến trúc rõ ràng.
- Được viết để cung cấp nhiều chức năng nhất với dung lượng nhỏ nhất.
- Không được chia thành các modules.
- Mặc dù MS-DOS được tổ chức có cấu trúc, các lớp chức năng cũng như giao diện giữa chúng không được phân chia tốt.



Kiến Trúc Hệ Điều Hành UNIX Cổ Điển

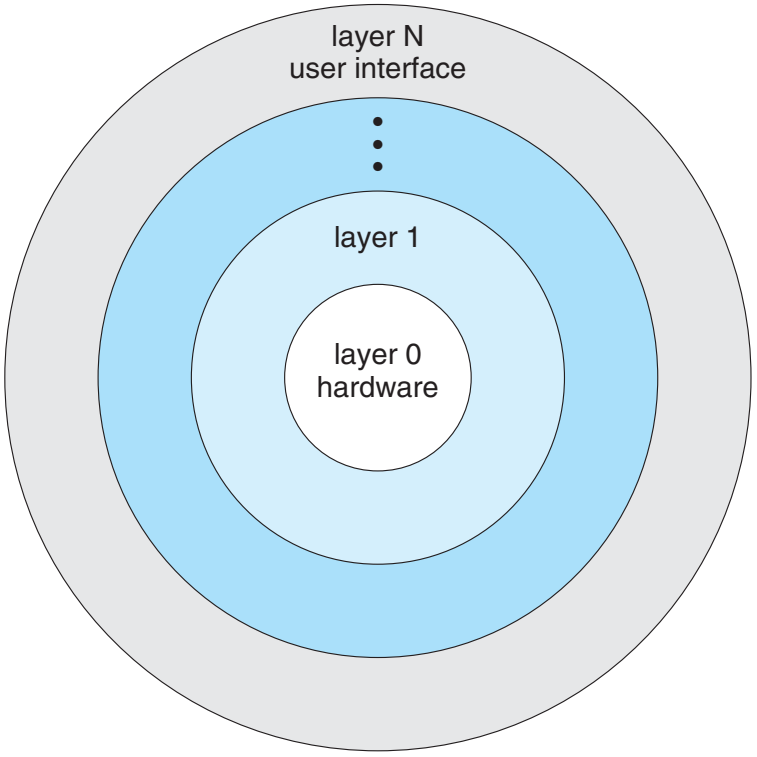
- Hệ điều hành UNIX khởi thủy có kiến trúc giới hạn do những giới hạn về phần cứng.
- Hệ điều hành UNIX bao gồm hai phần tách biệt:
  - Các chương trình hệ thống
  - Nhân (kernel)
    - Bao gồm mọi thứ phía dưới giao diện lời gọi hệ thống và phía trên phần cứng vật lý.
    - Cung cấp cơ chế quản lý hệ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

Kiến Trúc Hệ Điều Hành UNIX Cổ Điển



Kiến Trúc Phân Tầng

- Hệ điều hành được chia thành một số tầng, mỗi tầng được xây dựng trên nền tảng của một tầng khác thấp hơn.
- Tầng thấp nhất là tầng vật lý, tầng cao nhất là giao diện với người dùng.
- Sự phân chia chức năng: mỗi một tầng sẽ sử dụng các hàm (thao tác) và dịch vụ được cung cấp duy nhất bởi tầng phía dưới liền kề nó.



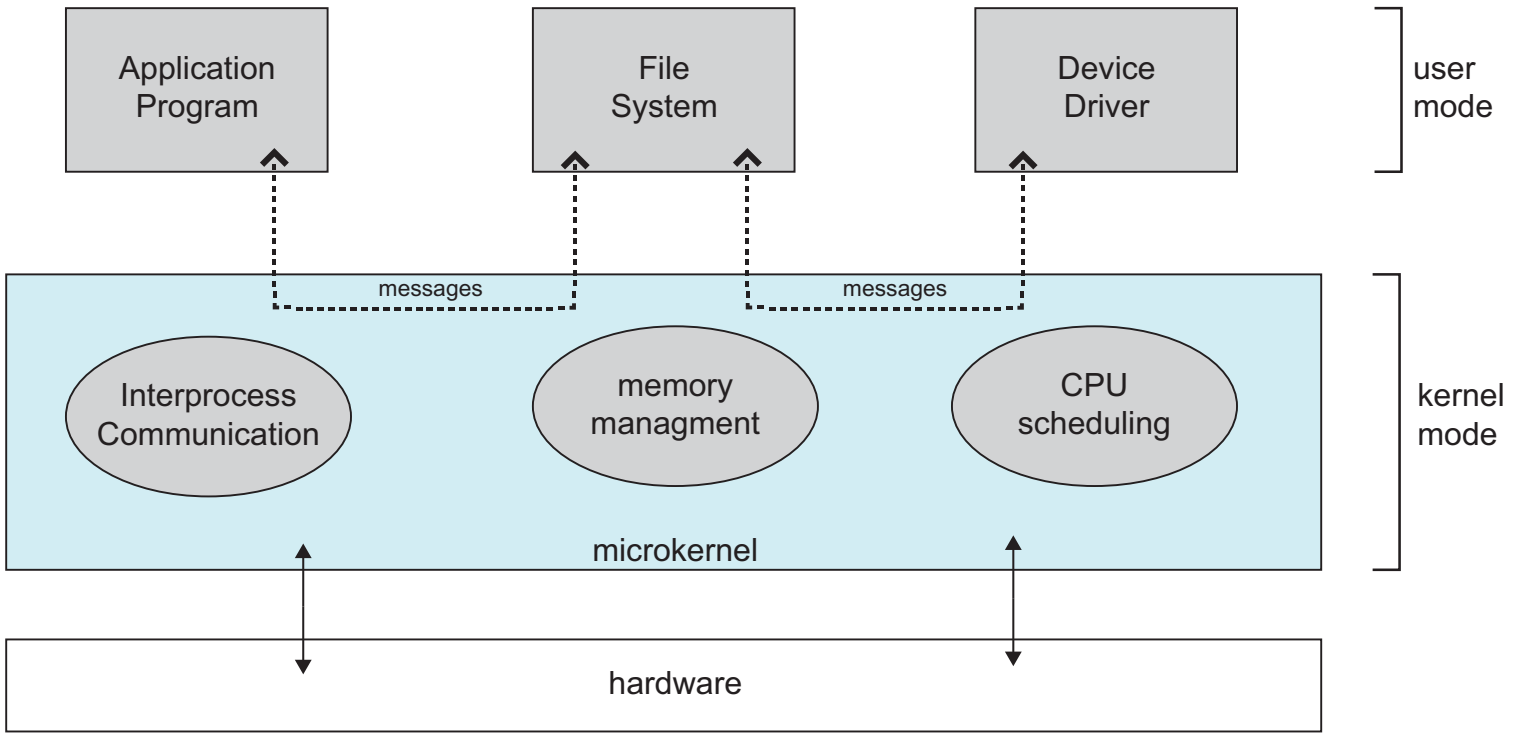
Kiến Trúc Phân Tầng – Ưu Nhược Điểm

- Ưu điểm:
  - Tính module (modularity) $\Rightarrow$ 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.
- Nhược điểm:
  - 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 chi phí cho việc gọi các lời gọi hệ thống thông qua nhiều tầng

Kiến Trúc Vi Nhân

- Di chuyển nhiều chức năng từ nhân lên mức người dùng.
- Việc giao tiếp giữa các module người dùng được thực hiện bằng cách sử dụng cơ chế chuyển thông điệp.
- Lợi ích:
  - dễ dàng mở rộng một microkernel
  - dễ dàng chuyển đổi hệ điều hành sang các kiến trúc mới
  - tin cậy hơn và an toàn hơn (ít mã lệnh chạy ở mức nhân hơn)
- Nhược điểm: chi phí giao tiếp giữa tiến trình của người dùng và nhân.

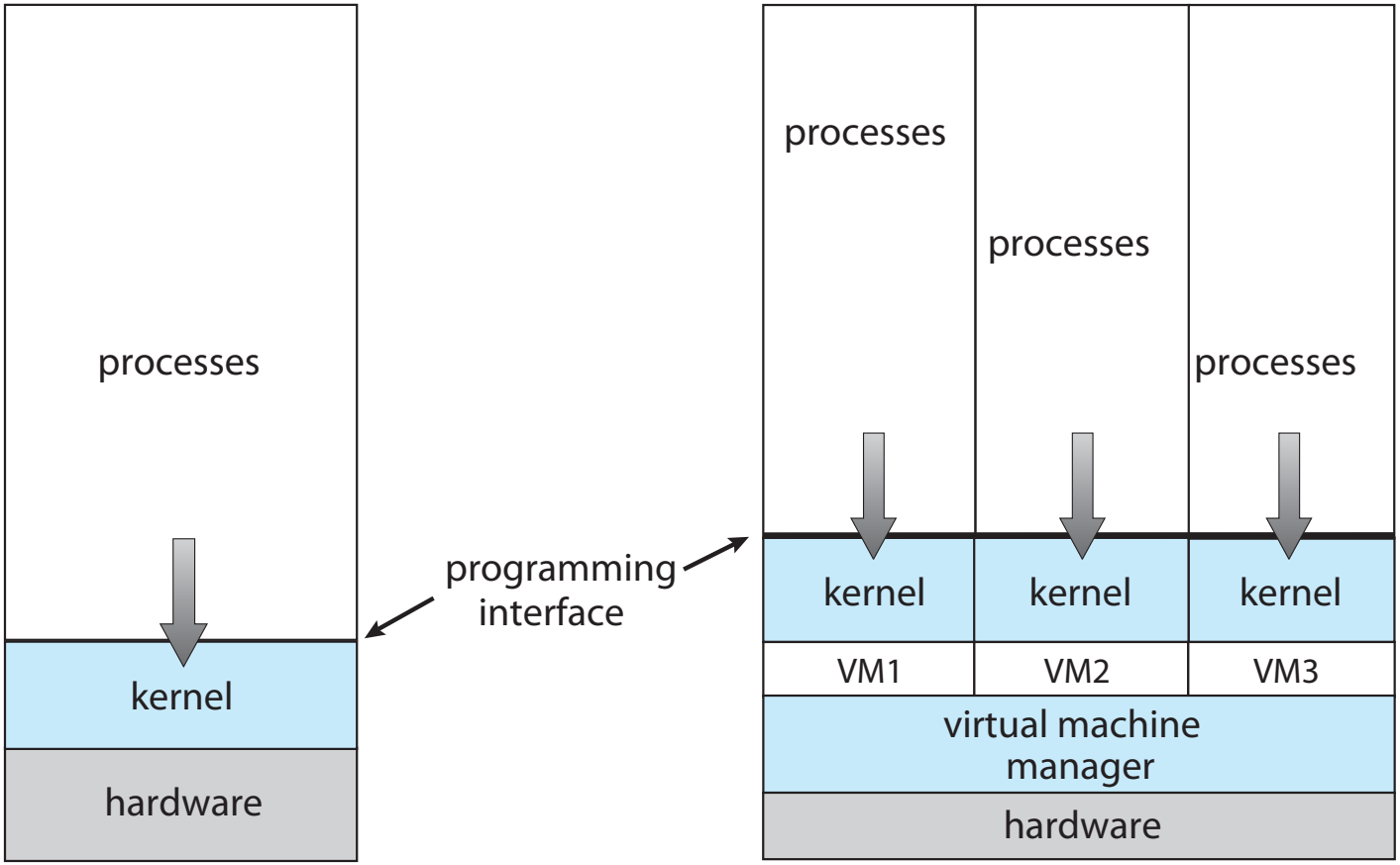
Kiến Trúc Của Một Hệ Vi Nhân Kiểu Mẫu



Máy Ảo (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 cảm giác nhiều máy, mỗi máy sử dụng phần CPU và bộ nhớ ảo của nó giống như hệ thống truyền thống.

Mô Hình Hệ Thống Máy Ảo



Mô hình hệ thống không máy ảo

Mô hình hệ thống máy ảo

Ưu Nhược Điểm Của Hệ Thống 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 $\Rightarrow$ 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.

Thiết Kế Hệ Điều Hành

- Có nhiều hướng tiếp cận.
- Bắt đầu bằng việc xác định các mục tiêu và đặc tả kỹ thuật:
  - phần cứng, kiểu hệ điều hành (batch, time-sharing, distributed, . . .)
- Yêu cầu:
  - Đối với người dùng: HĐH phải dễ dùng, dễ học, tin cậy, an toàn và nhanh.
  - Đối với hệ thống: HĐH phải dễ thiết kế, cài đặt, bảo trì, cũng như phải linh hoạt, tin cậy, không lỗi và hiệu quả.

Cơ Chế Và Chính Sách

- Cơ chế (mechanism) xác định cách thực hiện một việc.
- Chính sách (policy) quyết định cái gì được làm.
- Việc tách chiến lược ra khỏi cơ chế cho phép đạt được sự linh hoạt tối đa:
  - Các quyết định về chính sách có thể thay đổi sau đó
  - HĐH hành dạng microkernel cài đặt một tập hợp các hàm cơ bản (primitive) không có chính sách, cho phép thêm vào các chính sách và cơ chế dựa trên các hàm cơ bản này.
- Các quyết định về chính sách phải được xác lập cho tất cả các vấn đề về cấp phát tài nguyên (resource allocation) và định thời (scheduling).

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.
  - Gói 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.