

Bài 06

HỆ ĐIỀU HÀNH

Giảng viên: ...

Email: ...



Khoa Công Nghệ Thông Tin
Trường Đại Học Khoa Học Tự Nhiên

Nội Dung

- Giới thiệu về hệ điều hành
- Lịch sử phát triển của hệ điều hành
- Phân loại HĐH
- Kiến trúc của HĐH
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

Nội Dung

- Giới thiệu về hệ điều hành
- Lịch sử phát triển của hệ điều hành
- Phân loại HĐH
- Kiến trúc của HĐH
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

Định nghĩa



- Máy tính hiện đại gồm có:
 - Một hoặc nhiều bộ xử lý.
 - Bộ nhớ (RAM)
 - Lưu trữ (đĩa cứng)
 - Máy in
 - Các thiết bị nhập/xuất.
- Để quản lý những thành phần khác nhau này, cần một loại phần mềm đặc biệt: **Hệ điều hành – HĐH**

Định nghĩa

- **Hệ điều hành** là một *phần mềm* điều khiển và quản lý các thiết bị phần cứng và các tài nguyên phần mềm trên máy tính.
- Hệ điều hành đóng vai trò **trung gian** giữa *người dùng* và *phần cứng*.
 - Các chương trình thông thường sẽ không tương tác trực tiếp với phần cứng mà thông qua hệ điều hành.
- Hệ điều hành cung cấp:
 - Quản lý/thực thi các chương trình, ứng dụng.
 - Khiến cho việc sử dụng máy tính thuận tiện hơn.
 - Giải quyết các vấn đề của người dùng dễ dàng hơn.
 - Sử dụng phần cứng máy tính theo một cách hiệu quả hơn.
- Ví dụ:
 - Windows (nhiều phiên bản khác nhau), chủ yếu dành cho PC
 - UNIX: là phần lõi của Mac OS và Solaris
 - Linux

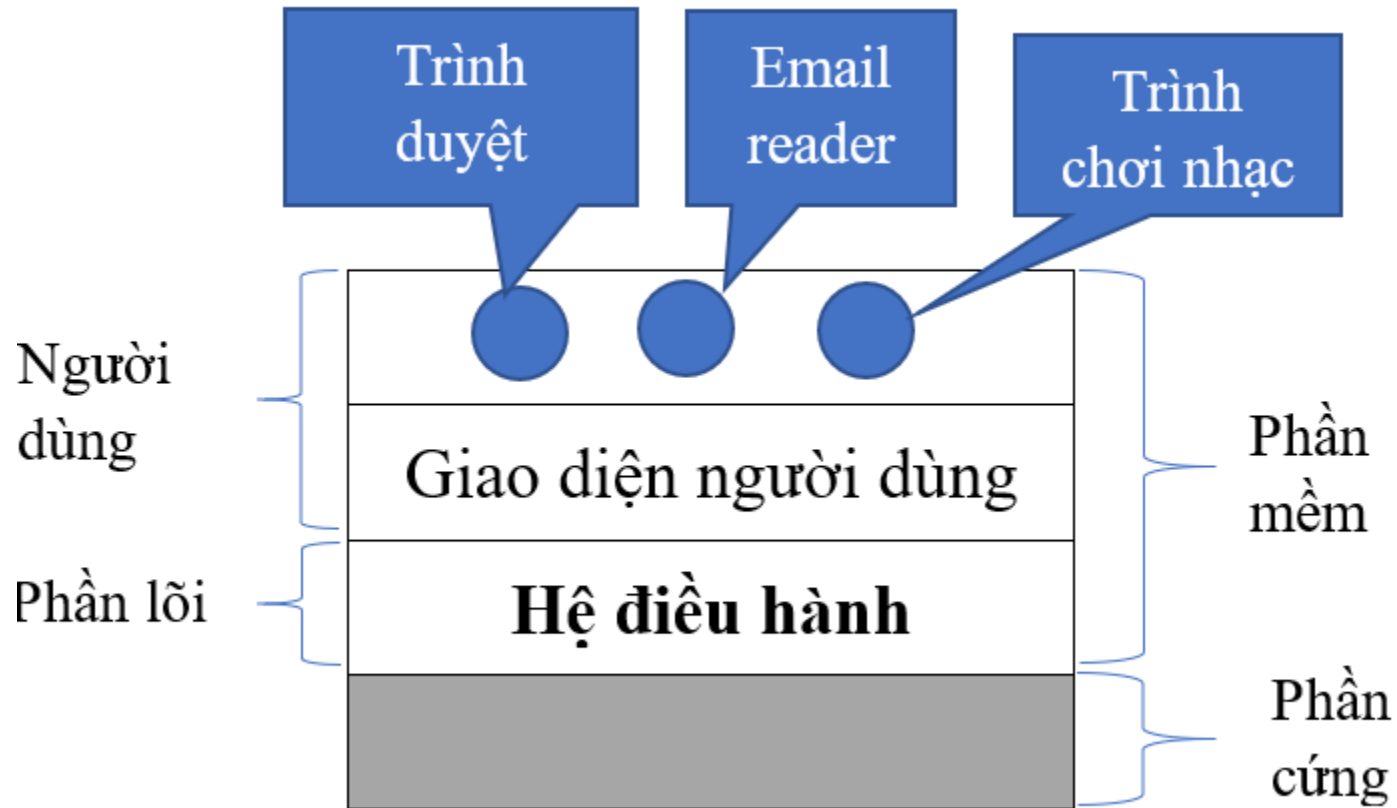
Lợi ích của hệ điều hành

- Đối với người lập trình
 - Dễ dàng hơn trong việc lập trình
 - Chỉ thấy mức trừu tượng cao, không cần phải biết chi tiết phần cứng.
 - Vd: tập tin chứ không phải các blocks trên ổ cứng.
 - Tính tương thích

Lợi ích của hệ điều hành

- Đối với người sử dụng máy tính
 - Dễ dàng sử dụng máy tính
 - Bạn có thể hình dung việc sử dụng máy tính không cần hệ điều hành?
 - An toàn
 - HĐH bảo vệ chương trình giữa các chương trình khác nhau.
 - HĐH bảo vệ người dùng giữa các người dùng khác nhau.

Vị trí của hệ điều hành



Cấu trúc của hệ thống máy tính

- Phần cứng: cung cấp các tài nguyên tính toán cơ bản
 - CPU, bộ nhớ, thiết bị nhập/xuất
- Hệ điều hành
 - Kiểm soát và điều phối việc sử dụng phần cứng giữa các ứng dụng và người dùng khác nhau
- Chương trình ứng dụng: định nghĩa cách mà tài nguyên hệ thống được sử dụng để giải quyết các bài toán cho người dùng.
 - Chương trình xử lý văn bản, trình biên dịch, trình duyệt web, game,...
- Người dùng cuối:
 - Con người, máy tính khác,...

Mục tiêu của HĐH

- Sự thuận tiện
 - Khiến việc sử dụng máy tính thuận tiện hơn
- Sự trừu tượng
 - Mô hình lập trình độc lập với phần cứng
- Tính hiệu quả
 - Cho phép máy tính dùng theo một cách hiệu quả
- Khả năng tiến hóa
 - Cho phép phát triển, thử nghiệm, và giới thiệu các hàm hệ thống mới mà không cần tương tác với các dịch vụ cũ.
- Bảo vệ

Hệ điều hành cung cấp

- Hỗ trợ tạo chương trình:
 - Soạn thảo, biên dịch, liên kết, debug.
- Thực thi chương trình
 - Nạp chương trình vào bộ nhớ, nhập xuất, khởi tạo file
- Thao tác nhập xuất và file
 - Xử lý những dạng nhập xuất đặc biệt và định dạng file.
- Truy cập hệ thống:
 - Giải quyết xung đột trong việc sử dụng tài nguyên.
 - Có cơ chế bảo vệ khi truy cập tài nguyên và dữ liệu.

Nội Dung

- Giới thiệu về hệ điều hành
- **Lịch sử phát triển của hệ điều hành**
- Phân loại HĐH
- Kiến trúc của HĐH
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

Lịch sử phát triển của hệ điều hành

- HĐH phải thích nghi với sự phát triển của phần cứng những loại phần cứng mới:
 - Ký tự vs. đồ họa (DOS vs Windows)
- HĐH phải cung cấp những dịch vụ mới.
 - Hỗ trợ Internet.
- HĐH phải có khả năng mở rộng, dễ dàng thay đổi, nâng cấp
 - Dùng mô hình lập trình hướng đối tượng.
 - Xây dựng theo module với giai diện rõ ràng

Thế hệ đầu tiên (1945-1955)

- Chưa có hệ điều hành và ngôn ngữ lập trình
- Máy tính rất lớn, dùng ống chân không và bảng điều khiển.
- Cách thức hoạt động của máy tính:
 - Làm việc hoàn toàn trên mã máy.
 - Người dùng kết nối các dây điện trên bảng điều khiển theo mã máy để điều khiển máy hoạt động.
 - Các chương trình là độc lập nhau, chỉ có thể chạy 1 chương trình tại một thời điểm.
- Được sử dụng cho việc tính toán toán học: tính sin, cos, logarithm.

Thế hệ thứ 2 (1955-1965)

- Sử dụng *bóng bán dẫn*.
- Máy tính đủ tin cậy để được bán đến các khách hàng, thay vì chỉ nằm trong phòng thí nghiệm.
- Vẫn rất đắt và khó dùng: chỉ các công ty lớn và cơ quan chính phủ sử dụng.
- Bắt đầu có sự phân biệt giữa: người thiết kế, người xây dựng, người vận hành, người lập trình, người bảo trì
- Để chạy một chương trình:
 - Lập trình viên viết một chương trình trên giấy (FORTRAN hay hợp ngữ)
 - Tạo nên các thẻ đục lỗ dựa trên chương trình trên giấy.
 - Đem các thẻ đục lỗ đến phòng Input và đưa cho nhân viên.
 - Nhân viên này sẽ đưa các thẻ đục lỗ này vào máy để thực hiện theo lịch.
 - Kết quả sẽ được in ra và mang đến phòng output để lập trình viên lấy.
- HĐH này lúc này đóng vai trò hỗ trợ chuyển đổi từ mã hợp ngữ sang mã máy và nhập xuất

Thế hệ thứ 3 (1965-1980)

- Máy tính lúc này sử dụng các mạch tích hợp, có thể thực hiện nhiều chương trình cùng một lúc.
- Máy tính lúc này gọi là minicomputer, rẻ hơn nhiều so với lúc trước. Tuy nhiên vẫn còn khá đắt cho người dùng cá nhân.
- Hệ điều hành ra đời nhằm điều phối, kiểm soát hoạt động và giải quyết các yêu cầu tranh chấp thiết bị.

Thế hệ thứ 4 (1980-nay)

- Máy tính lúc này gọi là máy tính cá nhân (Personal computer – PC). Giống với thế hệ thứ 3, nhưng rẻ hơn nhiều. Cá nhân có thể sở hữu
- Năm 1974 máy vi tính đầu tiên ra đời
 - Intel 8080, CPU 8 bit đầu tiên.
 - Dừng đĩa mềm.
 - Dừng HĐH **CP/M (Control Program for Microcomputers)**
- Đầu những năm 1980: máy tính cá nhân của IBM
 - Dừng HĐH **DOS (Disk Operating System)**
 - và **MS-DOS (Microsoft Disk Operating System)**
 - **FDOS (Disk Operating System)**
- Năm 1983: máy IBM PC/AT (Dùng CPU Intel 80286)
- 1985-1995: Windows
- Pentium PC:
 - UNIX, Linux, Windows 2000

Nội Dung

- Giới thiệu về hệ điều hành
- Lịch sử phát triển của hệ điều hành
- **Phân loại HĐH**
- Kiến trúc của HĐH
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

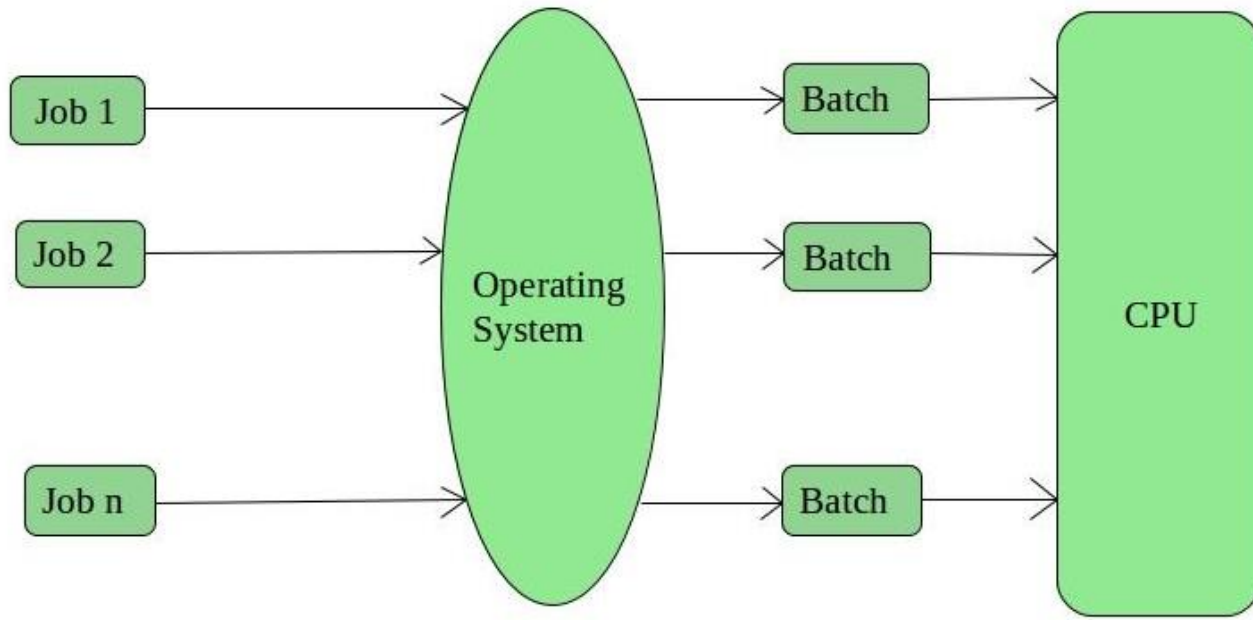
PHÂN LOẠI
THEO HÌNH THỨC XỬ LÝ



Phân loại theo hình thức xử lý

- Hệ thống xử lý theo lô (batch system)
- Hệ thống đa chương (multiprogramming system)
- Hệ thống chia sẻ thời gian (time-sharing system)
- Hệ thống song song (parallel system)
- Hệ thống phân tán (distributed system)
- Hệ thống mạng (network operating system)
- Hệ thống thời gian thực (real-time system)

Batch Operating System



- Hệ điều hành này không tương tác trực tiếp với máy.
- Hệ điều hành sẽ gom các tác vụ tương tự với nhau lại. Rồi đưa cho máy tính xử lý.
- **Ví dụ:** hệ thống trả lương, giao dịch ngân hàng.

Batch Operating System

- **Ưu điểm:**

- Bộ xử lý trong hệ thống này biết được thời gian thực hiện của mỗi tác vụ khi nó trong hàng đợi.
- Nhiều người dùng có thể chia sẻ hệ thống này.
- Thời gian rảnh của CPU giảm do nhóm các tác vụ lại.
- Dễ dàng quản lý một khối lượng công việc lớn nhưng có tính lặp lại.

- **Khuyết điểm:**

- Thiếu sự tương tác giữa người dùng và các tác vụ.
- **Tốc độ của các thiết bị nhập/xuất chậm hơn nhiều so với CPU, làm tăng thời gian rảnh của CPU**
- Khó debug.
- Các tác vụ khác đôi khi phải đợi một khoảng thời gian không xác định nếu một tác vụ bất kỳ có lỗi.

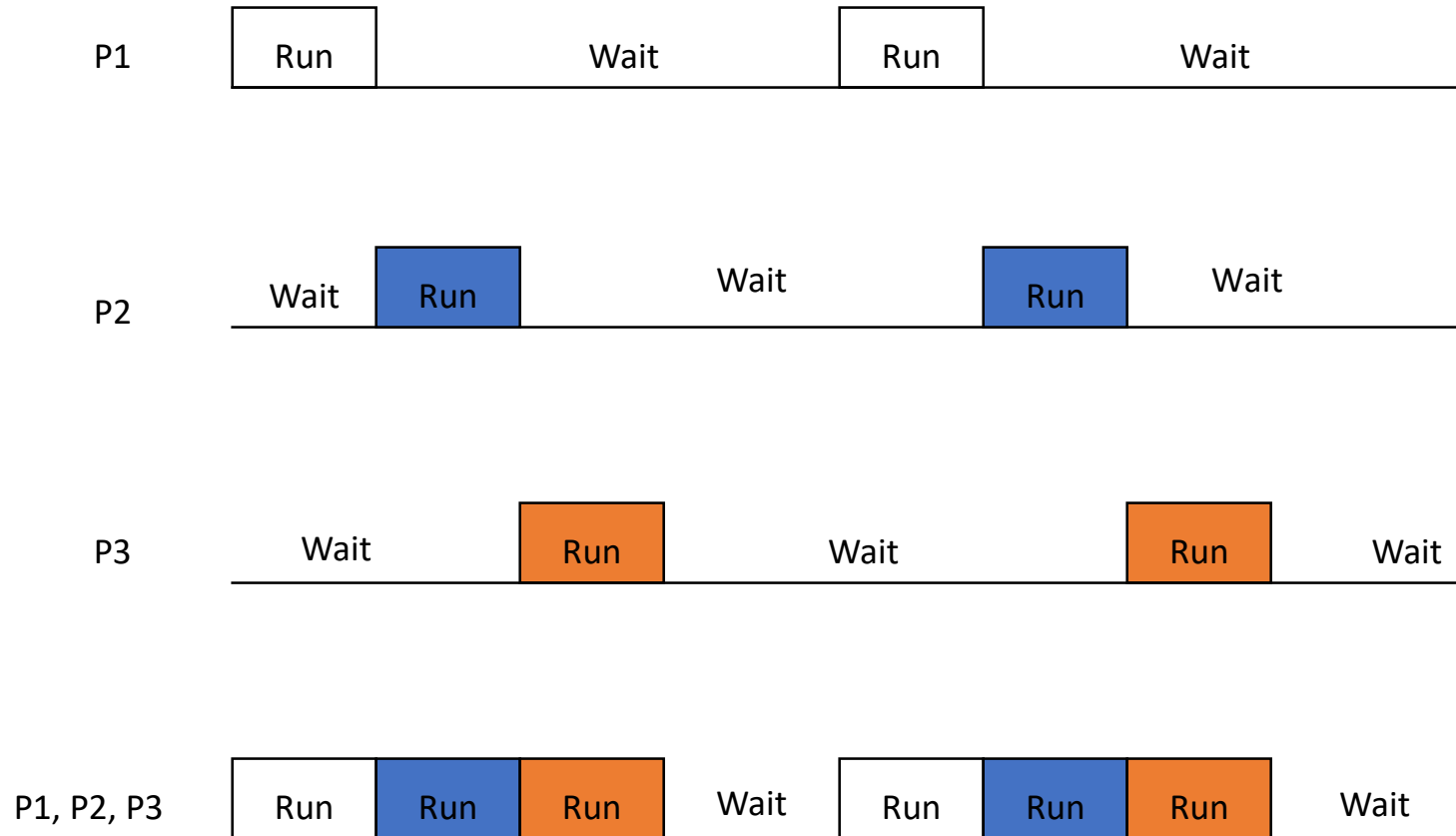
Khó xác định độ ưu tiên.



Multiprogramming system

- Trong hệ thống xử lý theo lô, CPU không được sử dụng nhiều do phải chờ các thao tác nhập xuất.
- *Hệ thống đa chương* khắc phục vấn đề trên.
- Một chương trình sẽ chuyển đổi giữa hai trạng thái:
 - Chờ người dùng nhập liệu.
 - Xử lý tính toán bởi CPU.
- Khi một chương trình đang chờ người dùng nhập liệu thì máy tính có thể cho phép chương trình khác thực thi CPU.
- Nhờ đó có thể tận dụng tối đa năng lực của CPU.

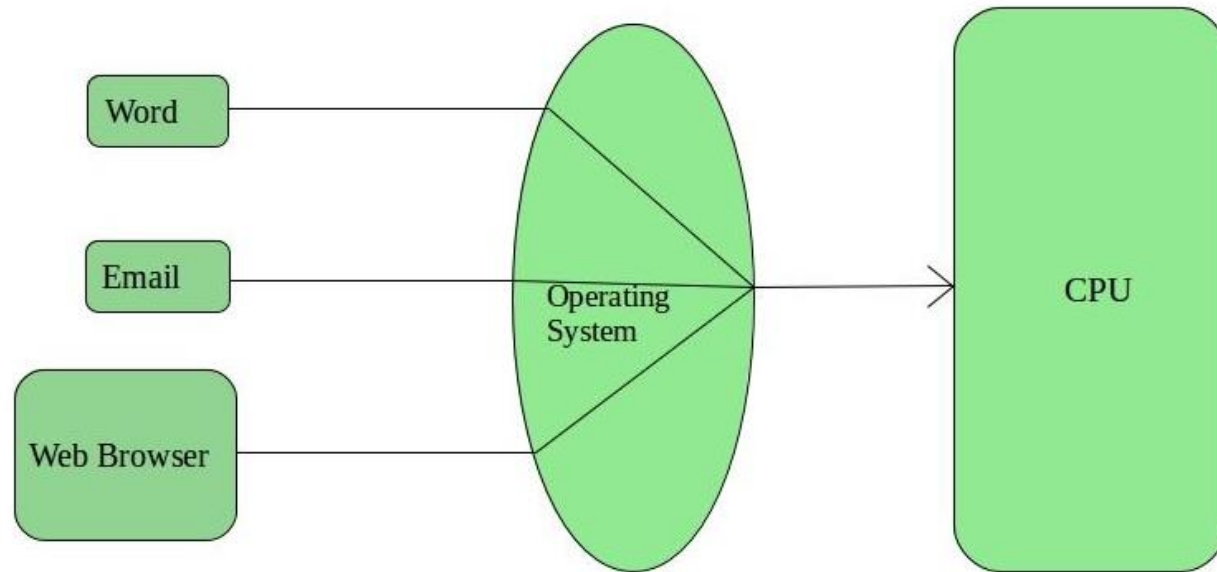
Ví dụ hệ thống đa chương



Time-Sharing Operating Systems

- Cho phép nhiều tác vụ khác nhau cùng sử dụng hệ thống tại một thời điểm
- Thời gian được chia thành các *khoảng bằng nhau*
- Mỗi chương trình chỉ được chiếm CPU một *khoảng thời gian nhất định*. Mục tiêu để chương trình nào cũng có thể thực hiện sau một khoảng thời gian biết trước.

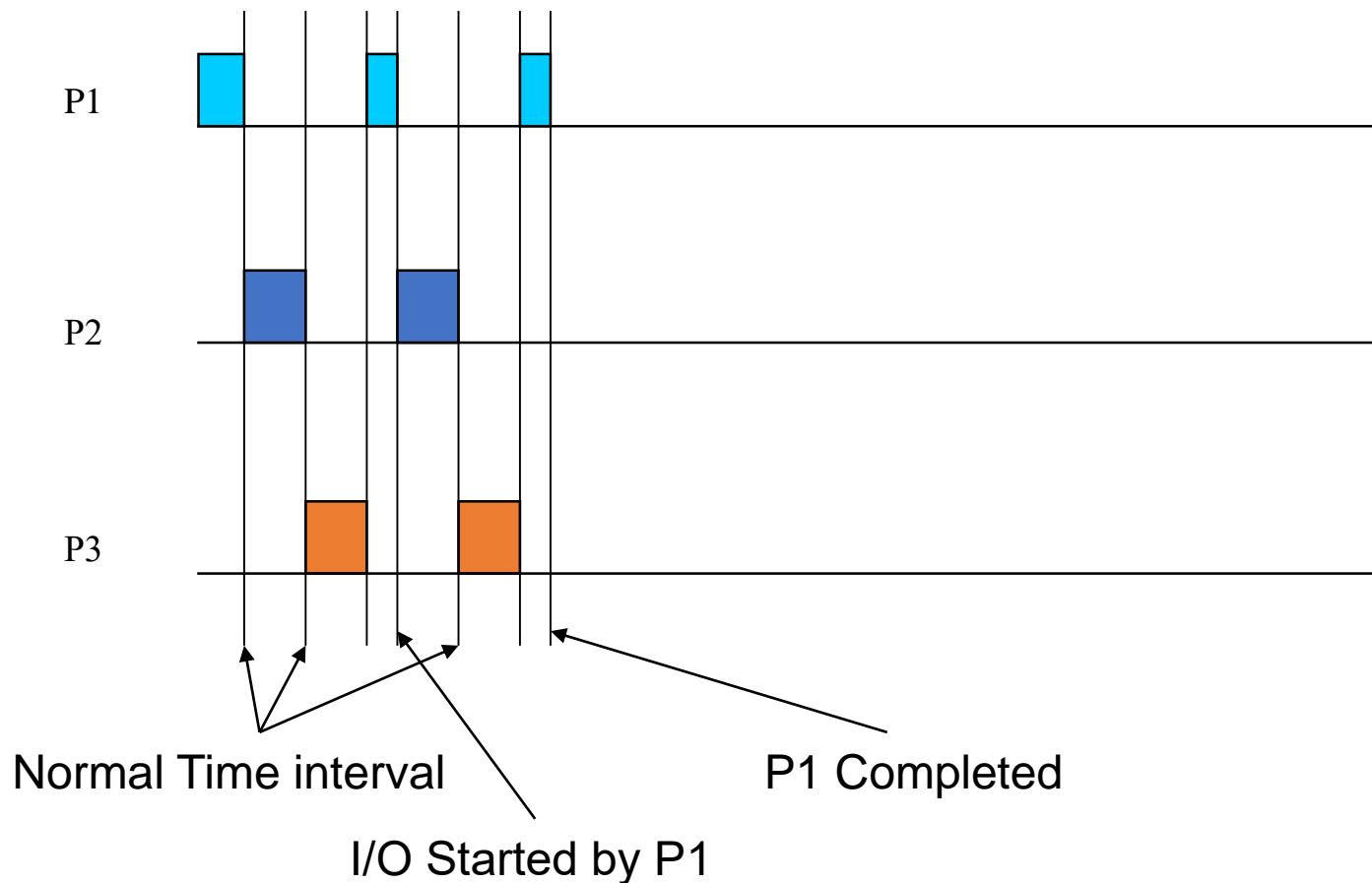
Time-Sharing Operating Systems



- Khác với HĐH đa chương: HĐH chia sẻ thời gian có thể ép buộc một chương trình đang chiếm CPU phải tạm dừng vì đã hết thời gian cấp cho nó. Còn HĐH đa chương thì không thế mà nó phải chờ cho đến khi chương trình tự rời bỏ CPU.

Time-Sharing Operating Systems

- Ví dụ về HĐH chia sẻ thời gian



Time-Sharing Operating Systems

- **Ưu điểm:**

- Từng tác vụ có cơ hội đều nhau.
- Giảm thời gian rảnh của CPU.
- Tránh việc lặp lại các chương trình

- **Khuyết điểm:**

- Độ tin cậy không cao
- Tính bảo mật và toàn vẹn của chương trình và dữ liệu không bảo đảm
- Gặp vấn đề khi trao đổi dữ liệu

- Ví dụ: Multics, Unix

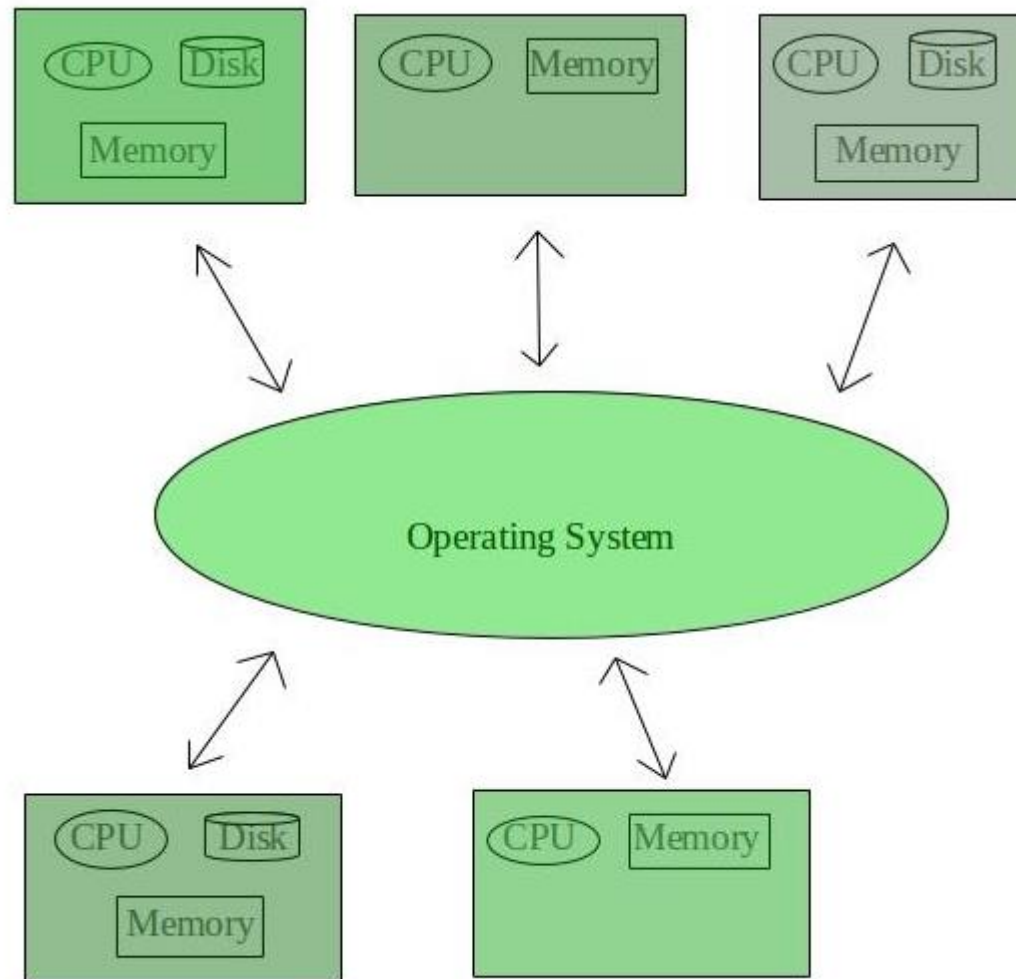
Đa chương và chia sẻ thời gian

- Mục tiêu của hệ đa chương
 - Tận dụng tối đa CPU.
- Mục tiêu của hệ chia sẻ thời gian
 - Giảm thiểu thời gian phản hồi cho người dùng
- **Câu hỏi**
 - Hệ nào cho throughput tốt hơn?
 - Throughput là số chương trình hoàn thành trên một đơn vị thời gian.

Distributed Operating System

- Đây là mô hình trong đó các *tác vụ* được phân tán để chạy trên *nhiều máy tính khác nhau*.
- Các máy tính này độc lập nhau, có bộ nhớ và CPU riêng. Nhưng được kết nối mạng với nhau. Còn được gọi là **loosely coupled systems**
- Người sử dụng không nhìn thấy tài nguyên của các máy tính vật lý bên dưới, mà chỉ thấy như một máy tính bình thường đang có, nhưng với năng lực lớn hơn nhiều

Distributed Operating System



Distributed Operating System

- **Ưu điểm:**

- Khi một máy tính trong mạng gặp sự cố, sẽ không có ảnh hưởng gì đến hệ thống, do chúng độc lập với nhau.
- Tài nguyên có thể được chia sẻ, dẫn đến tính toán nhanh hơn
- Giảm tải cho máy chủ
- Dễ dàng nâng cấp, mở rộng. Chỉ cần thêm máy mới vào hệ thống.
- Giảm độ trễ khi xử lý dữ liệu.

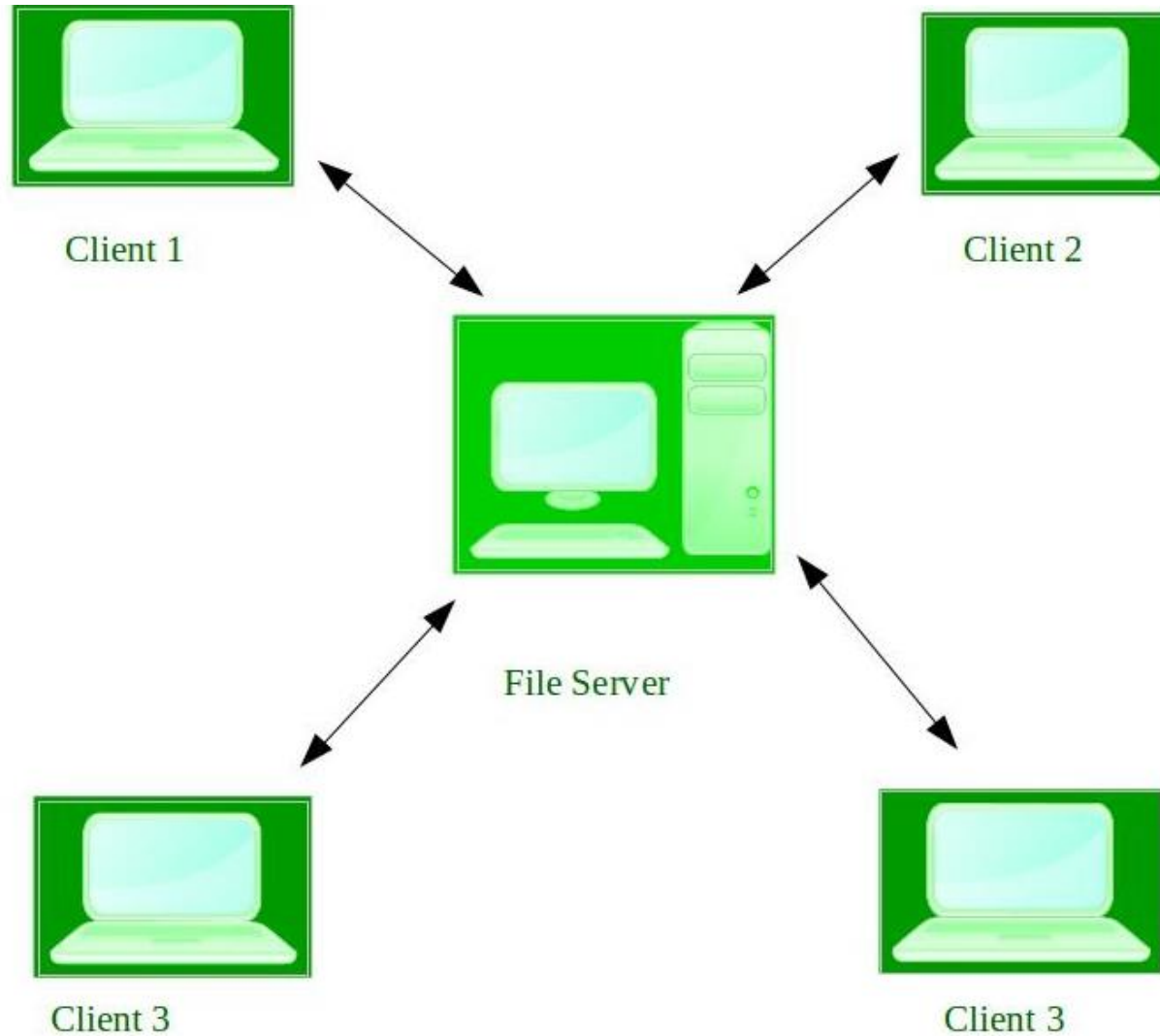
- **Khuyết điểm:**

- Nếu có sự cố đối với hệ thống mạng. Toàn bộ hệ thống sẽ sụp.
- Các phần mềm phức tạp.

Network Operating System

- Chạy trên server và cung cấp cho server khả năng quản lý: dữ liệu, người dùng, ứng dụng, bảo mật, và các chức năng khác về mạng.
- Mục đích chính của HĐH mạng này là cho phép khả năng chia sẻ file, máy in và các tài nguyên khác giữa các máy tính khác nhau trong cùng một hệ thống mạng (Ví dụ LAN)
- Ví dụ: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare

Network Operating System




Network Operating System

- **Ưu điểm:**

- Máy chủ tập trung có độ ổn định cao.
- Các vấn đề bảo mật được quản lý qua server
- Các công nghệ và phần cứng mới dễ dàng được tích hợp vào hệ thống.
- Có thể remote control

- **Khuyết điểm:**

- Server có chi phí cao.
- Người dùng phụ thuộc vào server cho hầu hết các tác vụ.
-  Đòi hỏi phải bảo trì và cập nhật thường xuyên.

Real-Time Operating System

- Loại HĐH này dùng trong có hệ thống thời gian thực.
- Thời gian để xử lý và phản hồi các yêu cầu phải rất nhỏ. Thời gian này gọi là *response time*.
- Dùng trong các hệ thống đòi hỏi ngặt nghèo về thời gian: hệ thống tên lửa, hệ thống điều khiển không lưu, robot.
- Được chia làm hai loại:
 - Hard Real-Time Systems
 - Soft Real-Time Systems

Real-Time Operating System

Hard Real-Time Systems

- Những hệ thống trong đó các yêu cầu về thời gian rất chặt chẽ. Thời gian trễ nhỏ nhất cũng không chấp nhận. Sẽ gây ra thảm họa nếu xử lý trễ. VD: Hệ thống điều khiển bay, hệ thống túi khí ô tô.
- Thiết bị lưu trữ rất giới hạn. Dữ liệu được lưu trữ trong ROM. Bộ nhớ ảo gần như không có.

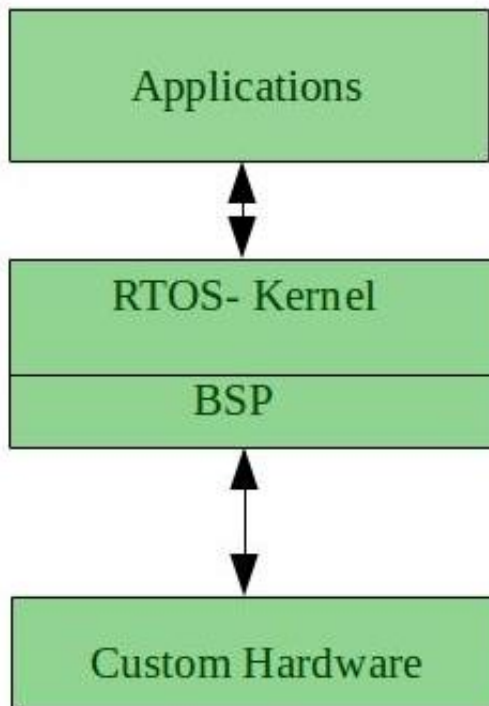
Soft Real-Time System

- Các yêu cầu về thời gian ít chặt chẽ hơn.

• VD: truyền thông, thực tế ảo.

Real-Time Operating System

- **BSP** (board support package) là một tầng phần mềm, chứa các driver phần cứng cho phép hệ điều hành hoạt động trên môi trường phần cứng cụ thể.



Real-Time Operating System

- **Ưu điểm:**

- Tối đa hoạt động: sử dụng các thiết bị và hệ thống một cách tối đa.
- Chuyển đổi tác vụ rất nhanh. Các hệ thống hiện nay chỉ tốn 3ms.
- Tập trung vào việc chạy ứng dụng hơn là lập lịch để chạy ứng dụng.
- Do kích thước nhỏ, thường được dùng trong các hệ thống nhúng.
- Ít lỗi. Do tính chất đặc thù, nên các hệ thống này rất lỗi.
- Cấp phát bộ nhớ tối ưu

- **Khuyết điểm:**

- Các tác vụ có thể thực hiện được rất giới hạn. Rất ít tác vụ có thể chạy đồng thời. Loại hệ thống này tập trung vào việc chạy một (vài) tác vụ để tránh lỗi hơn là chạy nhiều tác vụ.
- Thuật toán phức tạp.
- Sử dụng nhiều tài nguyên

Parallel system

- Nhiều ứng dụng có các công việc có thể thực hiện đồng thời: Dự báo thời tiết, mô phỏng, tính toán lại các bảng tính, ...
- Có thể tăng tốc độ bằng cách chạy các công việc trên các bộ xử lý khác nhau song song đồng thời.
- Cần HĐH và ngôn ngữ lập trình hỗ trợ chia nhỏ công việc thành các hành động song song.

Embedded system

- Phát triển rộng khắp
 - Hiện thời, ĐTDD và PDAs.
 - Tương lai, các thiết bị tính toán ở mọi nơi.
- HĐH giúp quản lý năng lượng, tính di động, tìm kiếm tài nguyên, v.v.

Embedded system

- Đặc tính
 - Tài nguyên hạn hẹp: CPU chậm, bộ nhớ nhỏ, không ổ đĩa hoặc ổ đĩa với kích thước nhỏ (vài chục GB), v.v.
 - Nguồn pin có giới hạn

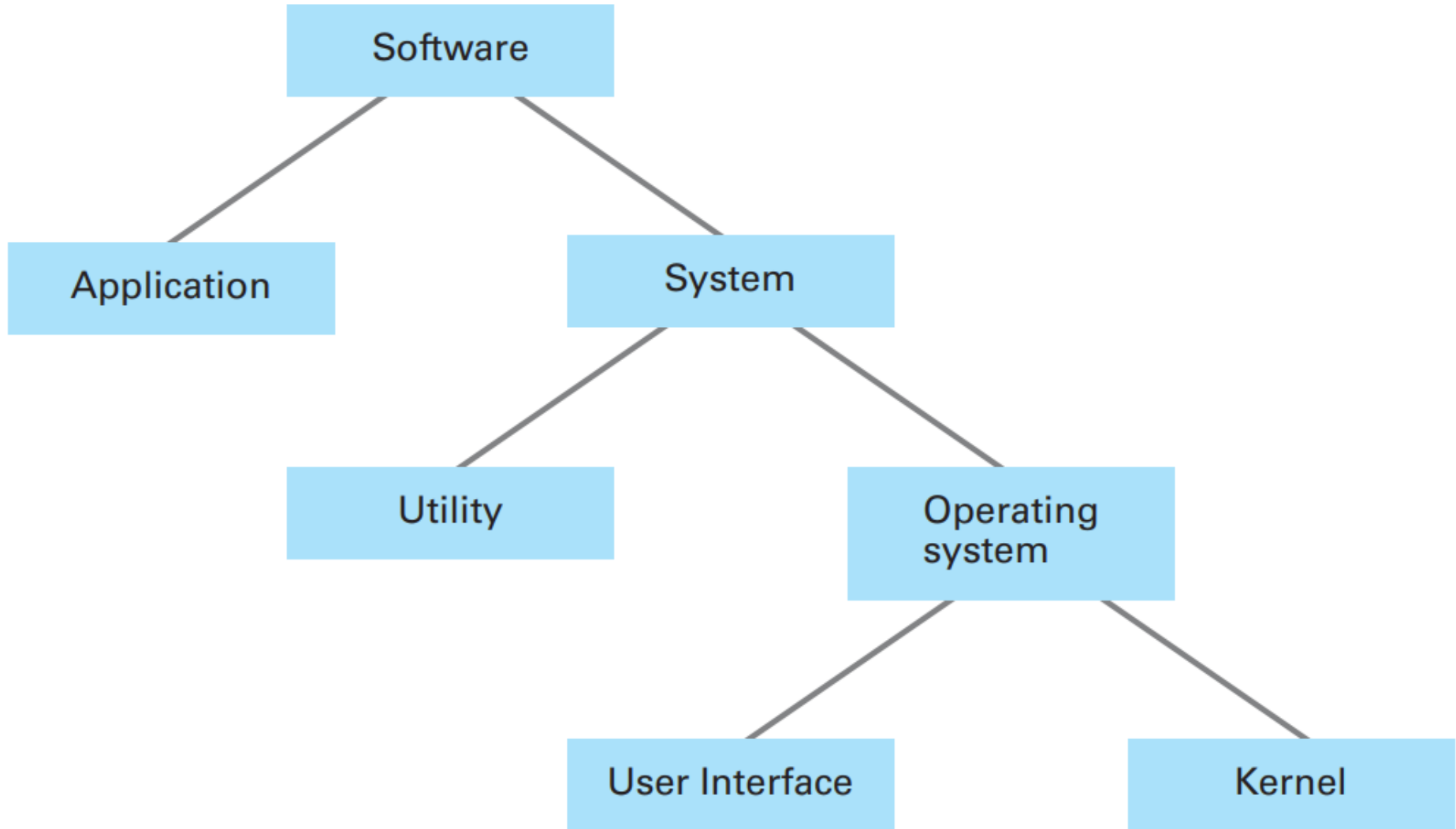
Nội Dung

- Giới thiệu về hệ điều hành
- Lịch sử phát triển của hệ điều hành
- Phân loại HĐH
- **Kiến trúc của HĐH**
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

Kiến trúc của HĐH

- Để hiểu về kiến trúc của HĐH, trước hết hãy hiểu rõ tổng thể về một chương trình trên máy tính nói chung. Sau đó tập trung vào HĐH
- Trước hết, hãy phân loại phần mềm.

Phân loại phần mềm



Phân loại phần mềm

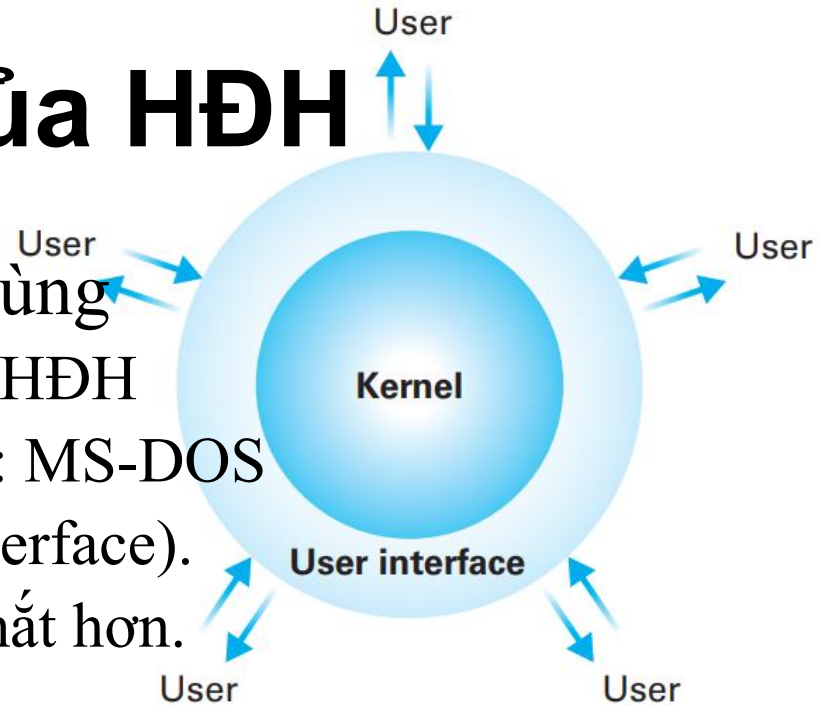
- Application software: phần mềm ứng dụng là phần mềm thực hiện trực tiếp công việc, đáp ứng yêu cầu cụ thể của người dùng.
 - VD: Microsoft word, Paint, Chrome
- System software: phần mềm hệ thống quản lý và điều hành phần cứng máy tính. Cung cấp nền tảng cho phần mềm ứng dụng thực hiện.
 - VD: Windows 10

Phân loại phần mềm

- Phần mềm hệ thống chia làm 2 loại:
 - Hệ điều hành.
 - Utility software – Phần mềm tiện ích:
 - Cung cấp cho máy tính những tính năng nền tảng, nhưng không nằm trong HĐH. Ví dụ: phần mềm để format đĩa cứng.

Các thành phần của HĐH

- User interface: giao diện người dùng
 - Đảm nhận giao tiếp giữa người và HĐH
 - Thế hệ cũ: shell, chỉ gồm text. VD: MS-DOS
 - Thế hệ mới: GUI (Graphic user interface).
Giao diện đồ họa, trực quan và bắt mắt hơn.
VD: Windows 10



- Kernel: phần lõi của HĐH. Chứa những thành phần cơ bản nhất liên quan đến việc cài đặt máy tính
 - File manager: Quản lý việc lưu trữ và truy xuất tập tin.
 - Device drivers: đảm nhận việc giao tiếp trực tiếp hay gián tiếp thông qua bộ điều khiển với các thiết bị ngoại vi trên máy tính.
 - Memory manager: quản lý việc sử dụng bộ nhớ.

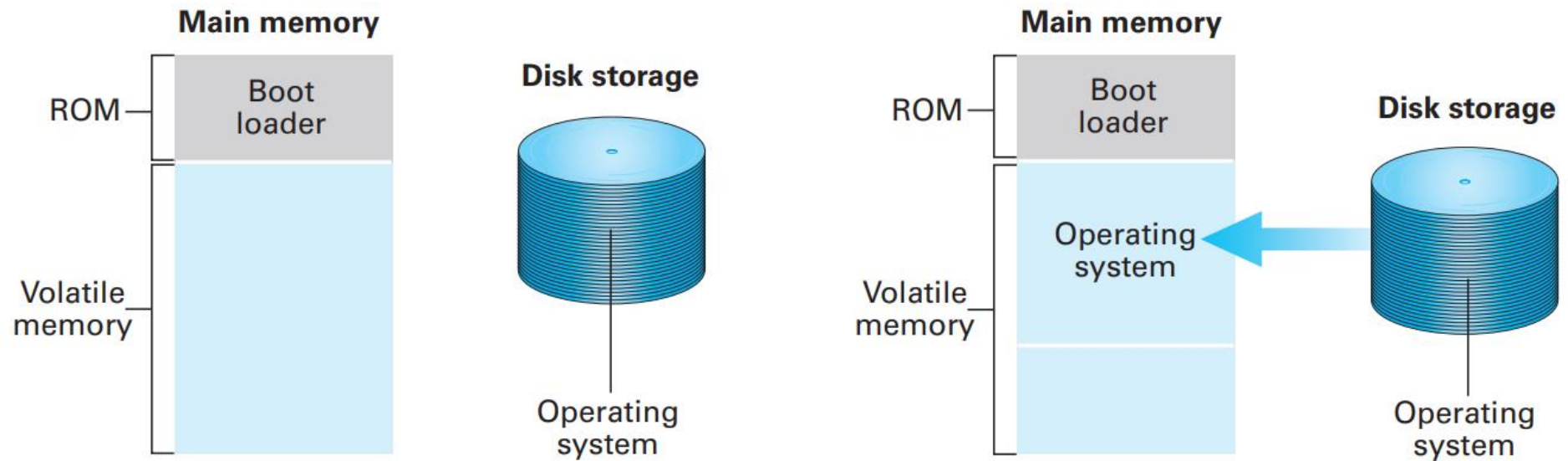
Nội Dung

- Giới thiệu về hệ điều hành
- Lịch sử phát triển của hệ điều hành
- Phân loại HĐH
- Kiến trúc của HĐH
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

HĐH hoạt động như thế nào

- HĐH được lưu trong ổ cứng. Để HĐH có thể hoạt động, cần phải sao chép nó vào bộ nhớ chính (RAM)
- Booting:
 - Được thực hiện mỗi lần mở máy.
 - Quá trình trên được khởi động bởi boot loader lưu trữ trên ROM.
 - Boot loader sẽ điều khiển quá trình sao chép HĐH từ nơi lưu trữ (đĩa cứng) sang bộ nhớ (RAM)

Booting



- Bước 1: máy tính bắt đầu cách thực thi chương trình boot loader nằm sẵn trong bộ nhớ ROM, HĐH đã được lưu trữ trong ổ đĩa.
- Bước 2: Boot Loader sẽ điều khiển việc copy HĐH sang bộ nhớ chính và sau đó trả quyền điều khiển về cho HĐH.

Processes

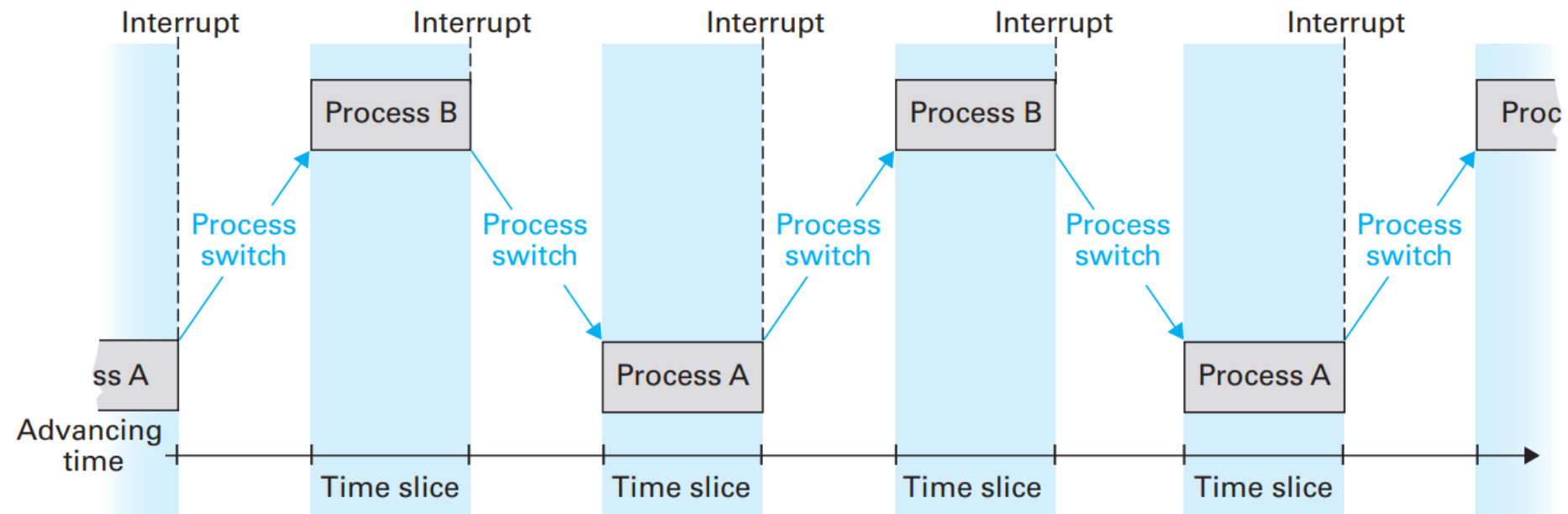
- **Process:** hoạt động thực thi chương trình dưới sự kiểm soát của HĐH
- **Process State:** trạng thái hiện hành của một hoạt động
 - Program counter: vị trí hiện tại
 - Các thanh ghi
 - Các phần bộ nhớ liên quan
- Nhiều process sẽ chạy cùng lúc. Chúng sẽ cạnh tranh các tài nguyên của máy tính.
- HĐH cần quản lý các process:
 - Mỗi process có tài nguyên cần thiết để dùng
 - Các process độc lập không cản trở hoạt động của nhau.
 - Các process có thể trao đổi thông tin với nhau nếu cần thiết

Quản lý các process

- Quá trình thực thi của các process được quản lý bởi *Scheduler* và *Dispatcher* nằm trong nhân của hệ điều hành.
- Để theo dõi các process, HĐH sẽ dùng *process table* lưu trong bộ nhớ
- **Scheduler:** khi một chương trình yêu cầu thực thi, scheduler sẽ thêm thông tin về process đó trong *process table* và xóa thông tin đó khi process hoàn thành. Thông tin này bao gồm:
 - Vùng bộ nhớ được gán cho process
 - Độ ưu tiên của process
 - Process *sẵn sàng* hay *đợi*. Sẵn sàng cho biết rằng việc thực thi của process có thể tiếp tục. Trạng thái đợi khi việc thực thi bị trì hoãn cho đến khi một sự kiện xảy ra. VD: nhấn phím.

Quản lý các process

- **Dispatcher:** quản lý việc cấp phát time slice cho các processes thực thi trong process table
 - CPU sẽ ngừng xử lý tiếp khi hết time slice



Quản lý việc cạnh tranh giữa các process

Khi có nhiều process cùng yêu cầu đến một tài nguyên, HĐH phải có cơ chế giải quyết để tránh ùn tắc.

- **Semaphore:** dùng flag để đánh dấu tài nguyên có đang được sử dụng hay không.
 - 0: tài nguyên có thể dùng. 1: tài nguyên đang được dùng bởi process khác.
 - Khi một process yêu cầu tài nguyên, nó sẽ kiểm tra flag. Nếu bằng 1, không thể truy cập, đợi. Nếu bằng 0, truy cập và set lại bằng 1.
- **Critical Region:** Các chỉ thị lệnh chỉ nên được thực thi bởi 1 process tại 1 thời điểm.
- **Mutual exclusion:** yêu cầu rằng chỉ một process tại 1 thời điểm được truy xuất đến critical region

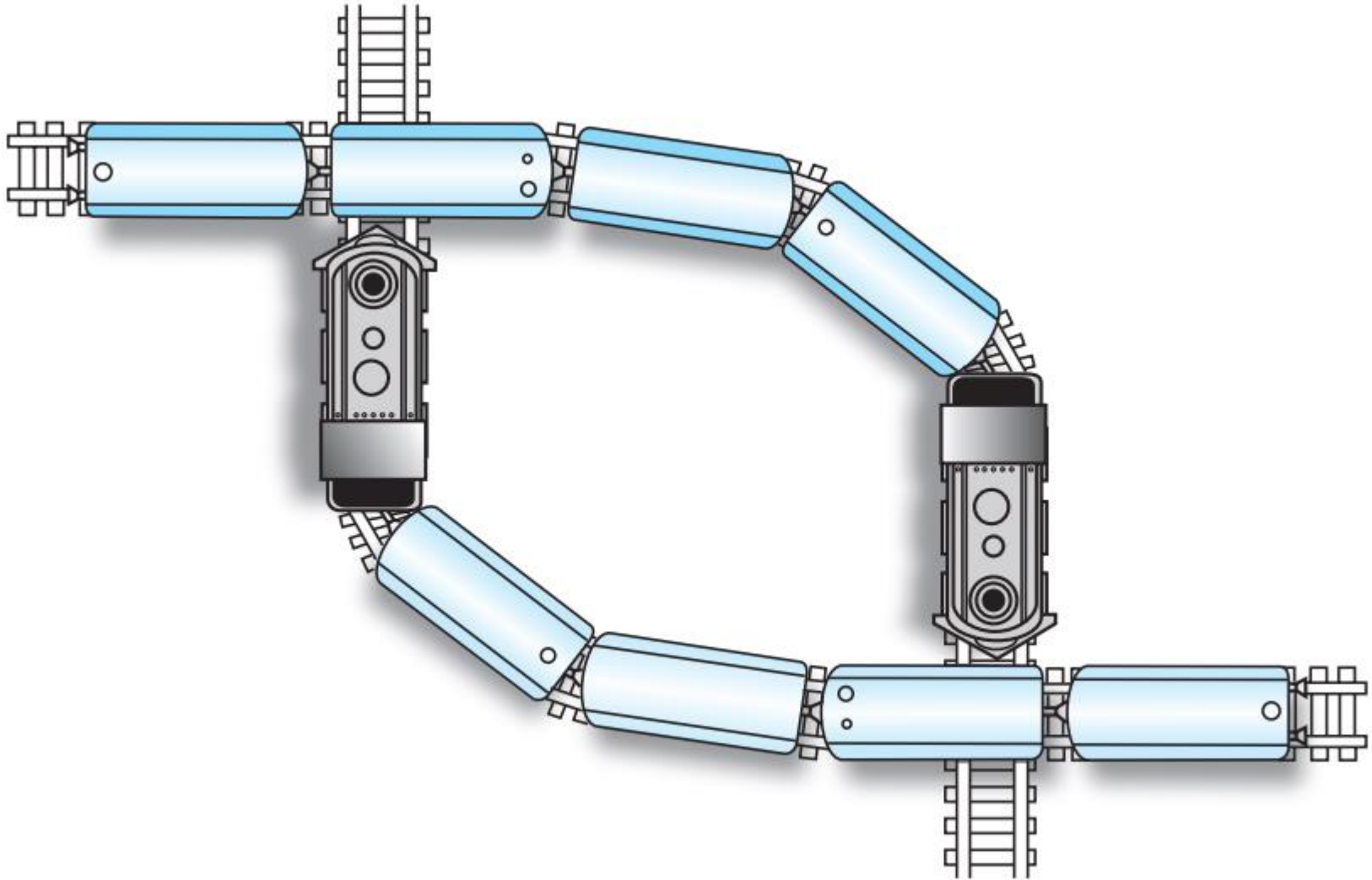
Deadlock

- Vấn đề khác xảy ra trong quá trình cấp phát tài nguyên cho các process có cạnh tranh.
- Xảy ra khi: hai hay nhiều process không thể tiếp tục vì phải chờ tài nguyên cấp cho process kia.

VD: process A có quyền truy cập đến máy in, nhưng phải đợi để truy cập đến ổ đĩa quang. Process B có quyền truy cập đến ổ đĩa quang nhưng đợi truy cập đến máy in. → Cả hai đợi nhau mãi mãi.

- Điều kiện xảy ra deadlock
 1. Có sự cạnh tranh cho tài nguyên (tài nguyên này chỉ cho phép 1 process sử dụng tại 1 thời điểm)
 2. Tài nguyên được yêu cầu 1 phần. Process đã nhận 1 phần tài nguyên và sau đó sẽ yêu cầu thêm
 3. Tài nguyên được cấp phát không thể lấy về một cách cưỡng ép

Deadlock



Bảo mật

- HĐH quản lý các hoạt động trên máy tính.
Nên có vai trò quan trọng trong việc bảo mật.
- Tấn công từ bên ngoài
 - Vấn đề
 - Mật khẩu không an toàn
 - Phần mềm lấy cắp mật khẩu
 - Giải quyết
 - Auditing software: phát hiện các hành vi bất thường nhằm đánh cắp mật khẩu.

Bảo mật

- Tấn công từ bên trong
 - Vấn đề: những processes bất thường
 - Giải pháp: quản lý việc hoạt động của các process thông qua có mode và chỉ thị lệnh có quyền hạn khác nhau.

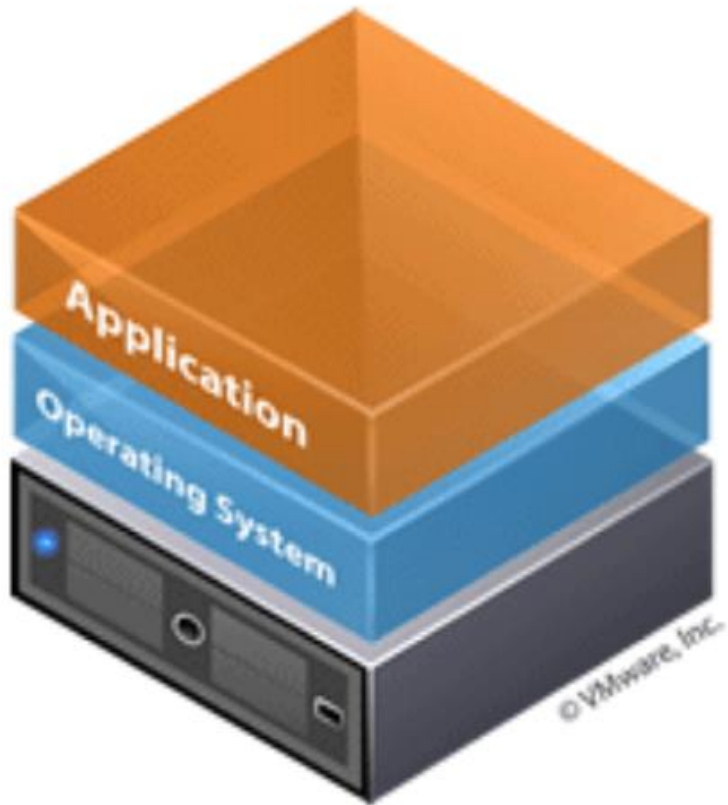
Nội Dung

- Giới thiệu về hệ điều hành
- Lịch sử phát triển của hệ điều hành
- Phân loại HĐH
- Kiến trúc của HĐH
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

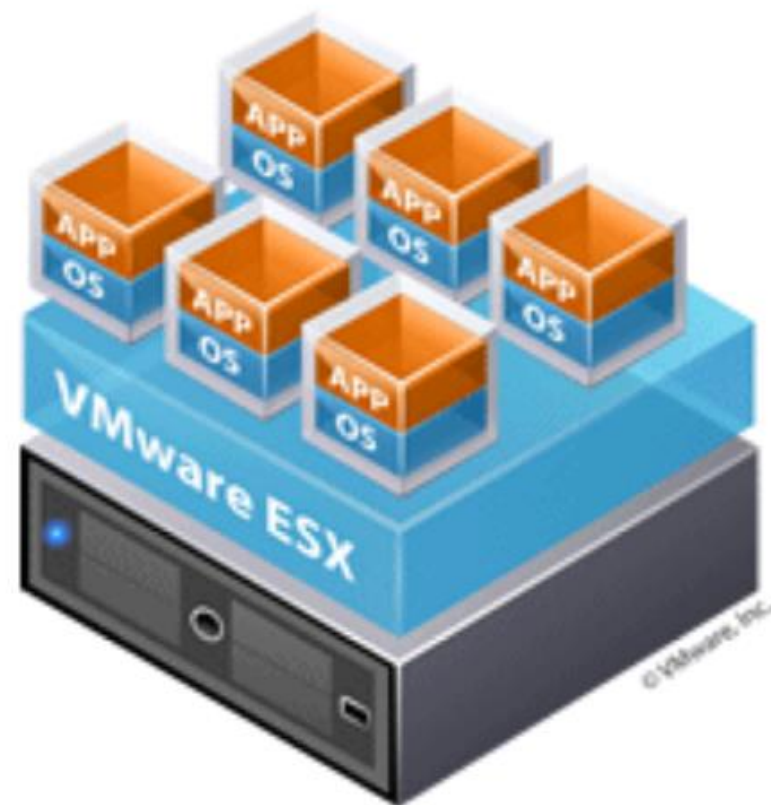
Khái niệm ảo hóa

- *Ảo hóa* là quá trình tạo ra một phiên bản *ảo* (không thực) của một thực thể *nào đó*. Ảo trong trường hợp này mang ý nghĩa rất giống với bản gốc, hầu như không phân biệt được.
- *Ảo hóa* là một công nghệ được thiết kế để tạo ra *tầng trung gian* giữa hệ thống *phần cứng máy tính* và *phần mềm chạy* trên nó
- Nguồn gốc:
 - Từ những mainframe 1960s
 - Dùng để ám chỉ việc chi tài nguyên của mainframe cho các ứng dụng khách nhau

Khái niệm ảo hóa



Traditional Architecture



Virtual Architecture

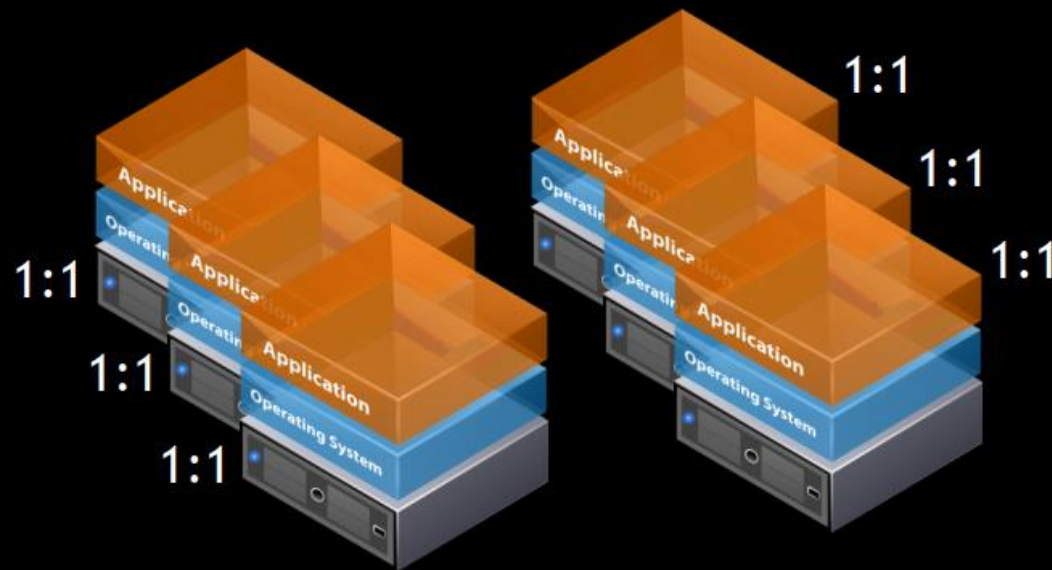
Lịch sử phát triển của ảo hóa

- 1960: thời đại của các mainframe
 - Tính toán tập trung (tại các mainframe)
 - Ảo hóa xuất hiện.
 - Thín client (người dùng chỉ dùng các thiết bị để giao tiếp với mainframe, việc xử lý chính nằm ở mainframe)
- 1980: thời đại của máy tính cá nhân
 - Tính toán phi tập trung: việc tính toán nằm ở từng PC
- 2000: PC phát triển lộn xộn
 - Vỡ bong bóng dot com.
- Hiện nay: cloud
 - Tính toán tập trung
 - Ảo hóa và thín client.
 - Internet of thing

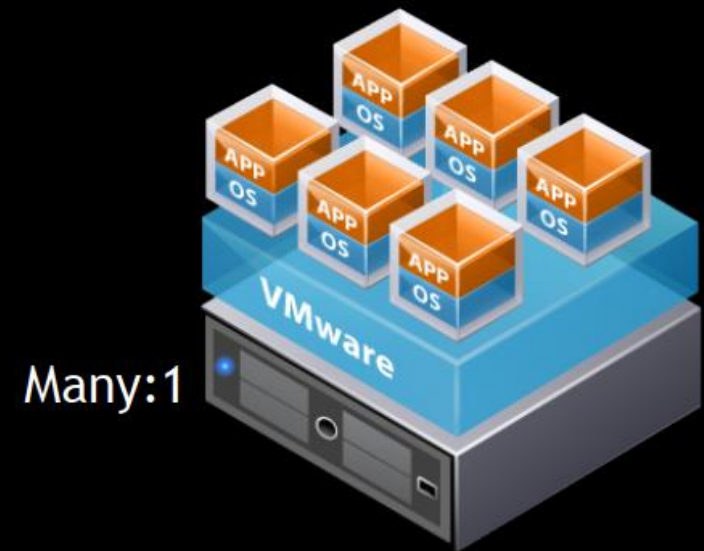
Tại sao cần ảo hóa

- Tối ưu hóa tài nguyên:
 - Tăng cường mật độ: Tăng cường việc tối ưu hóa tài nguyên nhưng không hi sinh hiệu năng.

Physical World

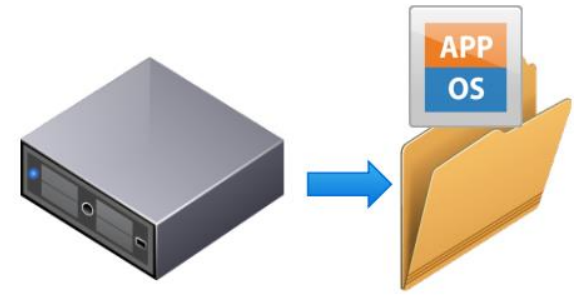


Virtual World



Tại sao cần ảo hóa

- Tối ưu hóa tài nguyên:
 - Sử dụng phần cứng năng suất hơn. Thông kê vào năm 2007:
 - 75% máy chủ hoạt động dưới 5% công suất
 - 25% máy chủ hoạt động dưới 0.5% công suất
- Portability - Phân chia:
 - Chạy nhiều hệ điều hành trên cùng một máy
 - Chia sẻ tài nguyên vật lý giữa các máy ảo
- Portability - Dễ di chuyển:
 - Toàn bộ máy ảo được lưu thành một file.
 - Di chuyển, sao chép, kết xuất dễ dàng như trên file
- Bảo mật:
 - Phần cứng cô lập với hệ điều hành
 - Phục hồi dễ dàng như trên file
- Agnostic – sự che đậy
 - Di chuyển các máy ảo giữa các máy chủ vật lý



Các loại ảo hóa

- ***Ảo hóa máy tính để bàn:***

- Tách môi trường trong máy tính để bàn ra khỏi phần cứng vật lý mà người dùng đang tương tác.
- Các tài nguyên tính toán (CPU, RAM,...) không nằm ở máy tính của người dùng mà nằm trên máy chủ.
- Giúp tiết kiệm chi phí, linh hoạt khi cần thay đổi phần cứng

- ***Ảo hóa hệ thống mạng***

- là một tiến trình hợp nhất tài nguyên, thiết bị mạng cả phần cứng lẫn phần mềm thành một hệ thống mạng ảo.
- Sau đó, các tài nguyên này sẽ được phân chia thành các channel và gắn với một máy chủ hoặc một thiết bị nào đó

- ***Ảo hóa phần cứng***

- Quen thuộc nhất.
- VD: VirtualBox, VMware.

Nội Dung

- Giới thiệu về hệ điều hành
- Lịch sử phát triển của hệ điều hành
- Kiến trúc của HĐH
- Các khái niệm quan trọng trong HĐH
- Công nghệ ảo hóa
- Công nghệ đám mây.

Điện toán đám mây

- **Điện toán đám mây:** là mô hình cung cấp tài nguyên máy tính (CPU, RAM, phần mềm, dịch vụ) cho người dùng thông qua một hệ thống mạng (thường là Internet)
- Người dùng có thể truy cập vào bất cứ tài nguyên nào trên đám mây. Vào bất kỳ thời điểm nào và ở bất kỳ đâu, chỉ cần kết nối với hệ thống mạng.



Tại sao dùng điện toán đám mây?

- **Tiết kiệm chi phí.** Thay vì xây dựng một hệ thống máy tính lớn, mà nhiều khả năng không dùng hết sức mạng phần cứng → Thuê hệ thống máy khác để sử dụng, chỉ phải trả phần thực sự dùng.
- **Tiện lợi:** Người dùng có thể nhanh chóng truy cập, sử dụng tài nguyên thông qua internet mà không cần cài đặt phức tạp
- **An toàn và liên tục:** Mọi dữ liệu được đồng bộ hóa trên đám mây. Giúp đảm bảo độ an toàn cao hơn, tránh trường hợp mất dữ liệu do hư hỏng ổ cứng. Ngoài ra, nhà cung cấp sẽ sao lưu định kỳ và có các phương thức bảo mật để bảo vệ dữ liệu tốt hơn.
- **Triển khai nhanh chóng ở bất kỳ đâu:** Chỉ với một vài thao tác đơn giản để triển khai chúng mọi nơi. Điều này đồng nghĩa với việc người dùng sẽ có được trải nghiệm tốt hơn với độ trễ thấp hơn.

Điện toán đám mây – Phân loại



Private Cloud
Community Cloud
Public Cloud
Hybrid Cloud

- Private cloud: tài nguyên tính toán được cung cấp cho 1 tổ chức duy nhất.



Public cloud

- Mô hình triển khai *điện toán đám mây* sử dụng phổ biến nhất hiện nay. Thường dùng cho B2C (Business to Consumer)
- Các tài nguyên trên *Public Cloud* đều nằm trên cùng một *hệ thống Cloud*. Tất cả người dùng sẽ dùng chung tài nguyên.
- Nhà cung cấp dịch vụ sẽ trực tiếp quản lý và bảo vệ dữ liệu trên đám mây
- **Ưu điểm:**
 - Phục vụ được nhiều đối tượng người dùng, không bị giới hạn về không gian, thời gian.
 - Đặc biệt **Public Cloud** có chi phí đầu tư thấp. Tiết kiệm được hệ thống máy chủ, giảm gánh nặng quản lý, cơ sở hạ tầng.
 - Public cloud có thể co giãn theo nhu cầu thực tế của người sử dụng.
- **Nhược điểm:**
 - Mất an toàn và khó kiểm soát dữ liệu



PRIVATE CLOUD

Private cloud

- **Private Cloud** thường được cung cấp cho các doanh nghiệp để đảm bảo an toàn dữ liệu. **Private cloud** sẽ được bảo vệ bên trong tường lửa của công ty và doanh nghiệp trực tiếp quản lý.
- **Ưu điểm:**
 - Chủ động hơn trong việc sử dụng và quản lý dữ liệu
 - Bảo mật thông tin tốt hơn
- **Nhược điểm:**
 - Gặp khó khăn trong việc triển khai công nghệ
 - Tốn chi phí để xây dựng, duy trì hệ thống
 - Chỉ phục vụ trong nội bộ doanh nghiệp. Những người dùng khác bên ngoài không thể tiếp cận và sử dụng.

Hybrid Cloud



Hybrid Cloud

- **Hybrid Cloud** là sự kết hợp giữa public cloud và private cloud. Nó cho phép người dùng khai thác được điểm mạnh của 2 mô hình trên. Và đồng thời hạn chế được điểm yếu của 2 mô hình đó. Hybrid cloud thường sẽ do doanh nghiệp tạo ra và việc quản lý thông tin. Dữ liệu sẽ được phân chia giữa doanh nghiệp và nhà cung cấp *Public Cloud*.
- **Ưu điểm:**
 - Đảm bảo được an toàn cho các dữ liệu quan trọng
 - Sử dụng được nhiều dịch vụ **điện toán đám mây** mà không bị giới hạn
- **Nhược điểm:**
 - Khó khăn khi triển khai và quản lý hệ thống
 - Tốn nhiều chi phí để xây dựng cơ sở hạ tầng

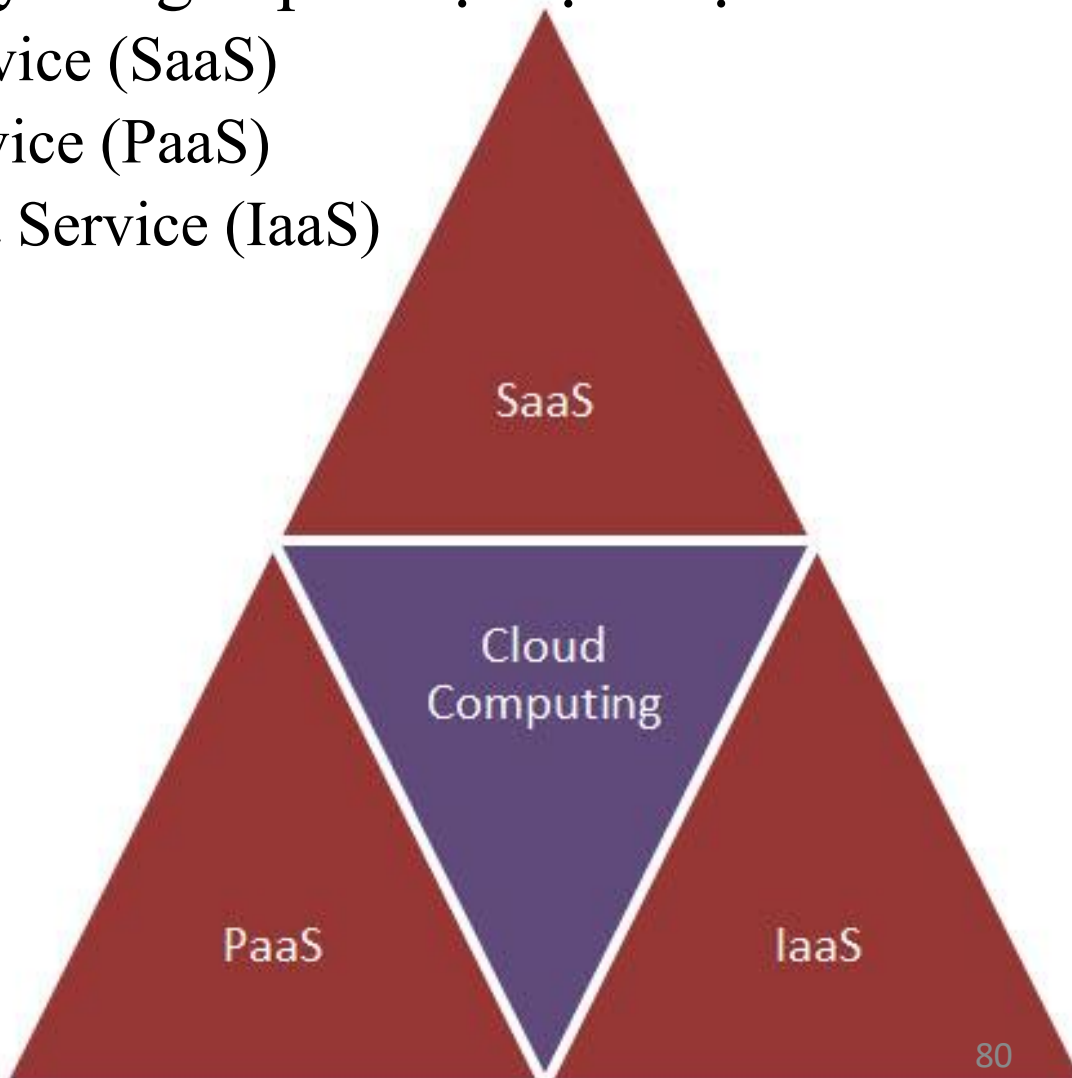


Community cloud

- *Community cloud* được xây dựng nhằm mục đích chia sẻ hạ tầng, dữ liệu cho nhiều tổ chức, người dùng khác nhau. Ví dụ, các doanh nghiệp cùng hoạt động trong ngành giáo dục có thể chia sẻ chung một đám mây để trao đổi dữ liệu cho nhau.
- **Ưu điểm:**
 - Các tổ chức/doanh nghiệp/cá nhân chung lĩnh vực hoạt động có thể chia sẻ dữ liệu, thông tin dễ dàng để phục vụ cho công việc của chính họ.
 - Đảm bảo sự riêng tư, an ninh và tuân thủ các chính sách tốt hơn.
- **Nhược điểm:**
 - Việc điều hành, quản lý tương đối khó khăn.
 - Cần tốn nhiều chi phí để xây dựng, triển khai.

Phân loại dịch vụ

- Điện toán đám mây cung cấp 3 loại dịch vụ chính
 - Software as a Service (SaaS)
 - Platform as a Service (PaaS)
 - Infrastructure as a Service (IaaS)



Infrasructure as a service

- *Infrasructure as a service* (**Iaas**) – Dịch vụ cơ sở hạ tầng
- Iaas là mô hình dịch vụ *pay-per-use* (tức là trả tiền cho những gì sử dụng). Chi phí sử dụng dịch vụ này được tính dựa trên chức năng và lượng tài nguyên mà khách hàng dùng. Theo Amazon thì đây là mức độ cơ bản nhất của điện toán đám mây.
- Nhà cung cấp dịch vụ Iaas sẽ bán cho khách hàng các server, thiết bị mạng, bộ nhớ, CPU, storage, máy tính (máy thật hoặc máy ảo), trang thiết bị trung tâm dữ liệu và một số tính năng bảo vệ an ninh nâng cao.
- Với hạ tầng mà Iaas tạo ra, bạn cần vào đó và thiết lập. Và cài thêm những phần mềm cần thiết khác như web server, database, ... Iaas không được tạo ra để phục vụ cho người dùng cuối. Mà nó để cho các công ty, đơn vị phát hành web sử dụng với mục đích triển khai phần mềm.



Platform as a service

- *Platform as a service (**Paas**) – Dịch vụ nền tảng*
- **Paas** là mô hình dịch vụ giúp các developer có thể phát triển. Nó cho phép triển khai các ứng dụng, website trên đám mây. **Paas** về cơ bản cũng khá giống với **Iaas** nhưng cấp độ cao hơn một chút. **Paas** được trang bị thêm các công cụ phát triển **doanh nghiệp thông minh (BI)**, **middleware** và **nhiều tool khác**. Với **Paas**, bạn sẽ có một nền tảng (Platform) được cài đặt sẵn để phù hợp cho việc phát triển ứng dụng.

Platform as a service

Platform Computing can be compared to your painting class where the teacher gives you paints, brushes etc as a platform to create your painting



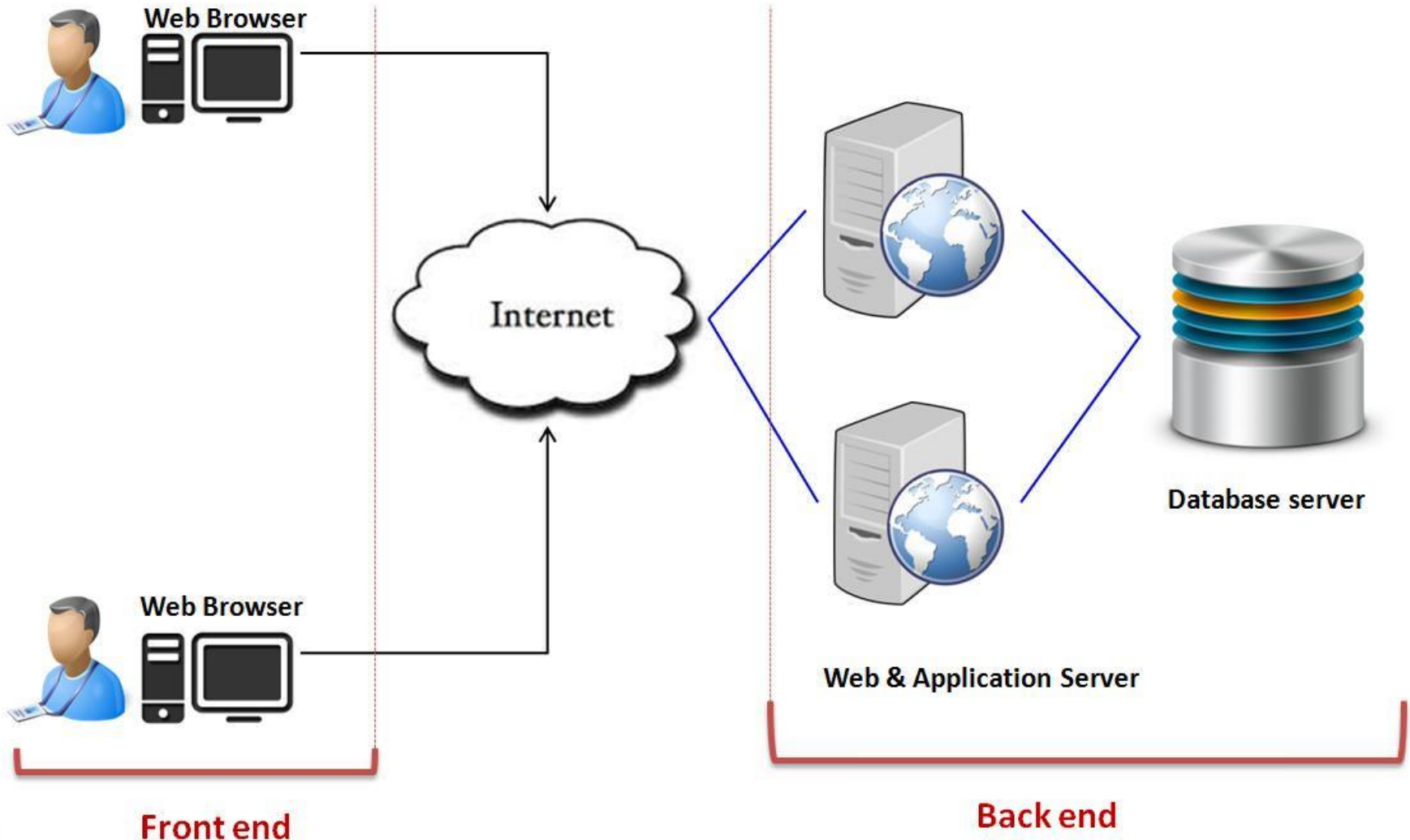
Software as a service

- *Software as a service (**SaaS**) – Dịch vụ phần mềm*
- **SaaS** là một mô hình dịch vụ **điện toán đám mây** cao nhất hiện nay. Cho phép người dùng sử dụng được các ứng dụng dễ dàng trên nền tảng đám mây thông qua internet. Đơn giản hơn, **SaaS** sẽ cung cấp phần mềm/ứng dụng chạy trên internet. Từ đó người dùng cuối (end-user) có thể sử dụng ngay. Nhà cung cấp dịch vụ **SaaS** có thể lưu trữ trên server của họ. Hoặc cho phép người dùng tải xuống và vô hiệu hóa nó khi hết hạn.
- Ví dụ: **Microsoft Office 365**. Đôi khi các web email (**Gmail, Outlook, Yahoo Mail,..**) cũng dùng dịch vụ này. Đây đều là các sản phẩm hoàn chỉnh. Người dùng có thể sử dụng ngay lập tức mà không cần phải thiết lập server để quản lý.
- Tương tự, **OneDrive, Dropbox** cũng là mô hình điện toán đám mây kiểu **SaaS**. Các trang web (phần mềm) này cung cấp không gian lưu trữ cần thiết để bạn có thể upload/download dữ liệu thông qua internet.

Software as a service



Kiến trúc của điện toán đám mây



Kiến trúc của điện toán đám mây

- Gồm hai thành phần: front end và back end
 - Front end: người dùng cuối. Bao gồm giao diện và các ứng dụng cần thiết để truy cập hệ thống điện toán đám mây.
 - Back end: chính là đám mây. Bao gồm các tài nguyên cần thiết cho các dịch vụ mà nó cung cấp.
- Điện toán đám mây không lưu trữ dữ liệu ở một nơi duy nhất, mà phân tán ra nhiều ổ đĩa và máy tính khác nhau. Để có thể phục hồi trong trường hợp dữ liệu gặp sự cố.
- Điện toán đám mây sử dụng công nghệ ảo hóa

Ai cần điện toán đám mây

Điện toán đám mây có thể được sử dụng cho bất kỳ ai:

- Công ty, doanh nghiệp, tổ chức có nhu cầu lưu trữ, chia sẻ dữ liệu.
- Nhà phát triển website, ứng dụng, phần mềm.
- Người dùng cuối (người dùng cá nhân).



Ứng dụng

- Cơ sở dữ liệu đám mây
- Thử nghiệm và phát triển web, ứng dụng
- Phân tích big data
- Lưu trữ dữ liệu cho website (còn gọi là cloud server)
- Lưu trữ, chia sẻ dữ liệu thông qua các nền tảng như: Google Drive, Dropbox, Shutterstock,...
- **Ứng dụng quản lý doanh nghiệp:** Có rất nhiều ứng dụng được thiết kế trên nền tảng đám mây để giúp doanh nghiệp quản lý, duy trì quan hệ với khách hàng tốt hơn.

Hết!