



# CT107. HỆ ĐIỀU HÀNH

## CHƯƠNG 1 - GIỚI THIỆU HỆ ĐIỀU HÀNH

### Giảng viên

Phạm Thế Phi, Lê Văn Lâm

Trần Công Ấn, Phạm Thị Xuân Diễm

{lvlam, ptphi, tcan, ptxdiem}@cit.ctu.edu.vn

Khoa Công Nghệ Thông Tin & Truyền Thông

Đại học Cần Thơ

2015

# MỤC TIÊU

MỤC TIÊU CỦA MỖI CHƯƠNG,

- Tại sao lại học chương này ?
- Chương này giúp ta cái gì ?

Giúp sinh viên hiểu rõ **Hệ điều hành là gì** và **vai trò của Hệ điều hành** trong các hệ thống máy tính và các môi trường điện toán.

# NỘI DUNG

HỆ ĐIỀU HÀNH LÀ GÌ?

TỔ CHỨC CỦA MỘT HỆ THỐNG MÁY TÍNH

CÁC LOẠI HỆ THỐNG MÁY TÍNH

SỰ PHÁT TRIỂN CỦA HỆ ĐIỀU HÀNH

CÁC MÔI TRƯỜNG ĐIỆN TOÁN

# HỆ ĐIỀU HÀNH LÀ GÌ?

– Phần cứng phải có phần mềm để điều khiển (firmware, HĐH)

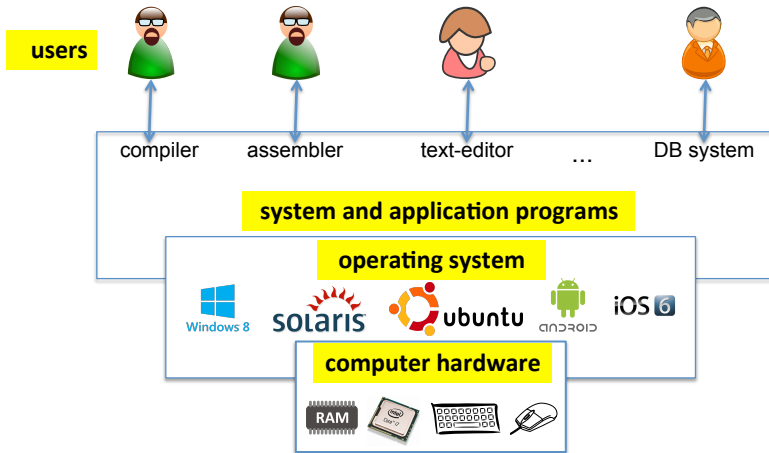
- ▶ là một chương trình quản lý tài nguyên của máy tính, đóng vai trò như một lớp trung gian giữa người sử dụng máy tính và phần cứng của máy tính. (giới thiệu máy tính và các ứng dụng)
- ▶ Mục tiêu:
  - ▶ Cung cấp phương tiện giao tiếp giữa người dùng và máy tính.
  - ▶ **Nhận và thực thi các yêu cầu của người dùng** một cách hiệu quả, nhanh chóng và dễ dàng thông qua các chương trình ứng dụng.
  - ▶ **Quản lý và sử dụng tài nguyên** máy tính một cách hiệu quả.

# THÀNH PHẦN CỦA MỘT HỆ THỐNG MÁY TÍNH

1. Phần cứng (hardware): cung cấp các tài nguyên cơ bản cho việc tính toán (CPU, bộ nhớ, I/O).
2. Hệ điều hành (OS): kiểm soát và điều phối việc sử dụng phần cứng của chương trình ứng dụng của người dùng.
3. Các chương trình hệ thống và ứng dụng (system and application programs): sử dụng tài nguyên hệ thống để giải quyết các vấn đề tính toán của người dùng.
4. Người dùng (user): con người hoặc các thiết bị có nhu cầu tính toán sử dụng các chương trình máy tính.

~ Dưới sự hỗ trợ của OS, mọi các nhà lập trình ứng dụng chỉ cần lập trình mà không cần tương tác vs phần cứng

# THÀNH PHẦN CỦA MỘT HỆ THỐNG MÁY TÍNH



# TỪ GÓC NHÌN CỦA NGƯỜI DÙNG

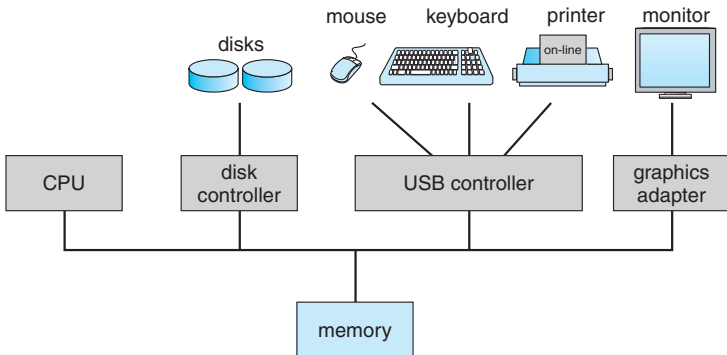
- ▶ Cái nhìn của người dùng đối với hệ điều hành phụ thuộc vào giao diện (interface) mà người dùng sử dụng.
  - ▶ Máy tính cá nhân (PC): tiện lợi, dễ sử dụng, hiệu năng cao, không quan tâm đến việc chia sẻ tài nguyên.
  - ▶ Shared-computer (mainframe, minicomputer): tận dụng các nguồn tài nguyên và chia sẻ công bằng.
  - ▶ Các trạm làm việc (workstation): hài hòa giữa việc sử dụng tài nguyên cá nhân và tận dụng tài nguyên chia sẻ.
  - ▶ Thiết bị cầm tay (handheld): thiết kế cho nhu cầu sử dụng cá nhân và cân đối giữa hiệu năng và năng lượng.

## TỪ GÓC NHÌN HỆ THỐNG – ĐỊNH NGHĨA HĐH

- ▶ là **bộ cấp phát tài nguyên**: quản lý và cung cấp các nguồn tài nguyên
- ▶ là một **chương trình điều khiển**: điều khiển các thiết bị nhập/xuất (phần cứng) và sự thực thi của các chương trình người dùng (phần mềm)
- ▶ là **nhân (kernel) của hệ thống máy tính**: là chương trình duy nhất chạy thường trực toàn thời gian (các chương trình khác gọi là các *chương trình ứng dụng*)

## TỔ CHỨC CỦA MỘT HỆ THỐNG MÁY TÍNH

- Bao gồm 1 hoặc vài bộ xử lý (CPU); bộ nhớ chính (RAM); các thiết bị I/O như đĩa từ, bàn phím chuột, màn hình, máy in; các bộ điều khiển thiết bị, ...



## TỔ CHỨC CỦA MỘT HỆ THỐNG MÁY TÍNH

- ▶ Các thiết bị I/O và CPU có thể thực thi đồng thời.
- ▶ Mỗi bộ điều khiển thiết bị <sup>driver</sup> sẽ điều khiển một loại thiết bị nào đó.
- ▶ CPU, bộ nhớ chính và các bộ điều khiển thiết bị được nối kết với nhau thông qua một bus chung.
- ▶ Mỗi bộ điều khiển thiết bị có một bộ nhớ đệm riêng.
- ▶ Các bộ điều khiển thiết bị thông báo với CPU sau khi thực hiện xong tác vụ bằng cách sử dụng các ngắt (interrupt).

— Mỗi hệ điều hành sẽ có một bộ driver phổ biến, để chạy các device phổ biến

## QUÁ TRÌNH KHỞI ĐỘNG CỦA MÁY TÍNH

- ▶ Khi máy tính khởi động, nó cần 1 chương trình mỗi (initial program/bootstrap).
- ▶ Chương trình này có chức năng tương đối đơn giản: định vị kernel của HĐH, nạp kernel vào bộ nhớ, khởi động kernel và sau đó nhường quyền điều khiển cho HĐH.
- ▶ Bootstrap được lưu trữ trong ROM (read-only memory) hoặc (EPROM), còn được gọi là firmware.
- ▶ Sau khi kernel của HĐH được thực thi, HĐH sẽ bắt đầu cung cấp các dịch vụ thông qua các lời gọi hệ thống.

# CÁC LOẠI HỆ THỐNG MÁY TÍNH

- ▶ Vai trò, chức năng và kiến trúc của HĐH phụ thuộc vào kiến trúc của hệ thống máy tính.
- ▶ Các hệ thống máy tính có thể chia làm 2 loại:
  1. Các hệ thống đa dụng: mainframe, desktop, multi-processor, distributed, clustered.
  2. Các hệ thống chuyên dụng: real-time, multimedia, handheld.

# HỆ THỐNG BÓ (BATCH)

- ▶ là hệ điều hành thô sơ đầu tiên
- ▶ người dùng không giao tiếp trực tiếp với máy tính mà thông qua người điều khiển (operator)
- ▶ rút ngắn thời gian thiết lập chương trình (setup time) bằng cách bó lại (batch) các công việc tương tự nhau
- ▶ tự động phân dãy công việc, chuyển quyền điều khiển một cách tự động từ một công việc đến một công việc khác thông qua bộ giám sát thường trú của HĐH
- ▶ trong hệ thống này, CPU thường xuyên rảnh vì tốc độ CPU nhanh hơn rất nhiều so với các thiết bị nhập xuất cơ khí

# HỆ THỐNG BÓ – SƠ ĐỒ BỘ NHỚ

Hệ điều hành

Vùng nhớ dành cho  
chương trình của  
người sử dụng

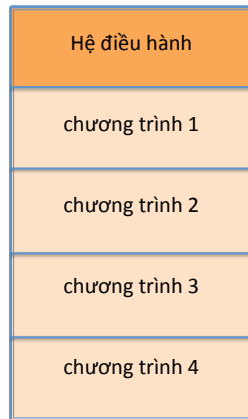


**IBM 7094 Mainframe** (Source: IBM)

1997.

# HỆ THỐNG ĐA CHƯƠNG (MULTI-PROGRAMMING)

- ▶ Sự ra đời của công nghệ đĩa là cơ sở cho các hệ thống đa chương: Các công việc có thể được lưu trữ và truy xuất một cách không tuần tự trên hệ thống đĩa.
- ▶ Một số công việc được lưu trong bộ nhớ chính.
- ▶ CPU được điều phối thực hiện một công việc khác nếu công việc hiện hành đang chờ đợi một thao tác xuất/nhập.
- ▶ Ưu điểm: Tận dụng thời gian rỗi của CPU.



# HỆ THỐNG ĐA CHƯƠNG – YÊU CẦU ĐỐI VỚI HĐH

- ▶ Các hoạt động vào ra (I/O): phải được cung cấp bởi hệ thống.
- ▶ Quản lý bộ nhớ: hệ thống phải cấp phát bộ nhớ cho nhiều tiến trình.
- ▶ Định thời cho CPU: hệ thống phải chọn trong số các công việc đang sẵn sàng một công việc để giao CPU cho nó sử dụng.
- ▶ Một chương trình đang thực thi trong hệ thống chỉ nhường lại CPU cho chương trình khác khi nó hoàn thành hoặc cần thực hiện thao tác I/O.

# HỆ THỐNG CHIA THỜI GIAN (TIME-SHARING)

- ▶ Là sự mở rộng luận lý của hệ thống đa chương.
- ▶ Nhằm tăng hiệu suất sử dụng các tài nguyên trong hệ thống.
- ▶ Cho phép nhiều người dùng chia sẻ máy tính tại 1 thời điểm bằng cách **phân chia thời gian sử dụng các tài nguyên**. *Tạo cảm giác mới & dùng có riêng mỗi CPU*
- ▶ **CPU** sẽ được điều phối cho nhiều công việc đang nằm trong bộ nhớ và trong đĩa (CPU chỉ được cung cấp cho công việc nào đang nằm trong bộ nhớ).
- ▶ **Công việc** sẽ được hoán chuyển giữa bộ nhớ và đĩa.
- ▶ **Giao tiếp trực tuyến** giữa hệ thống và người dùng được cung cấp; khi hệ điều hành hoàn thành thực thi một lệnh, nó sẽ tìm một “lệnh điều khiển” của người dùng từ bàn phím.

# HỆ THỐNG CHIA THỜI GIAN (TIME-SHARING)

- ▶ Hệ thống phân chia thời gian **phức tạp hơn** hệ thống đa chương:
  - ▶ cơ chế **quản lý bộ nhớ** phức tạp: quản lý cạnh tranh, bảo vệ bộ nhớ
  - ▶ **bộ nhớ ảo**: cho phép tăng số lượng chương trình trong bộ nhớ
  - ▶ cơ chế **định thời vi cho CPU** tinh vi: cung cấp cơ chế đồng bộ hóa, giao tiếp giữa các tiến trình, cơ chế định thời CPU tinh vi, ...
  - ▶ phải cung cấp hệ thống **quản lý đĩa**

# CÁC HỆ THỐNG ĐỂ BÀN (DESKTOP)

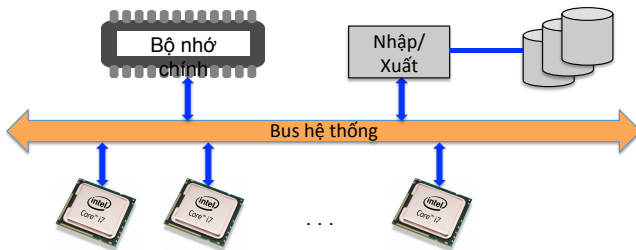
- ▶ Máy tính cá nhân (personal computer): hệ thống máy tính được dành cho một người dùng duy nhất.
- ▶ Các thiết bị xuất/nhập: bàn phím, chuột, màn hình, máy in.
- ▶ Tiện lợi và phản ứng nhanh đối với người dùng.
- ▶ Có thể phỏng theo các kỹ thuật được phát triển cho các hệ thống lớn.
- ▶ Có thể chạy nhiều họ hệ điều hành khác nhau (Windows, MacOS, UNIX, Linux).

# CÁC HỆ THỐNG DA XỬ LÝ (MULTI-PROCESSOR)

- ▶ Là các hệ thống đa xử lý với nhiều hơn một CPU được nối kết chặt chẽ với nhau
- ▶ Còn được gọi là các hệ thống song song hay hệ thống ghép đôi chặt
- ▶ Các processors chia sẻ bộ nhớ và xung đồng hồ; việc giao tiếp diễn ra thông qua bộ nhớ được chia sẻ.
- ▶ Lợi ích của hệ thống song song:
  - ▶ Tăng năng lực xử lý: nhiều công việc được hoàn thành/đơn vị thời gian
  - ▶ Kinh tế: chia sẻ ngoại vi, thiết bị lưu trữ, điện, ...
  - ▶ Tăng tính tin cậy: chỉ giảm cấp xử lý khi có sự cố, cung cấp hệ thống chịu lỗi (fault tolerant)

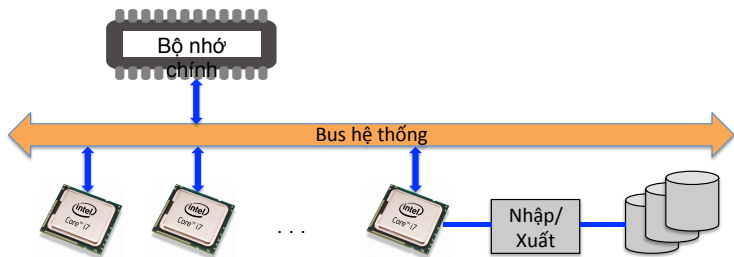
# KIẾN TRÚC HỆ THỐNG DA XỬ LÝ ĐỐI XỨNG

- ▶ Symmetric MultiProcessing – SMP.
- ▶ Mỗi CPU chạy một bản sao giống nhau của hệ điều hành.
- ▶ Nhiều quá trình có thể chạy song song mà không làm giảm hiệu năng của hệ thống.
- ▶ Hầu hết các hệ điều hành hiện đại đều hỗ trợ SMP



# KIẾN TRÚC HỆ THỐNG DA XỬ LÝ BẤT ĐỐI XỨNG

- ▶ Asymmetric multiprocessing – AMP
- ▶ Mỗi CPU được giao một công việc cụ thể; CPU chủ (Master) sẽ lập lịch biểu và giao việc cho các CPU tớ (Slave).
- ▶ Thường phổ biến trong các hệ thống cực lớn.

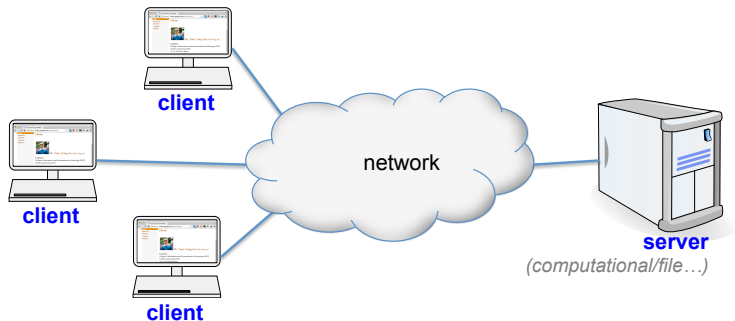


# CÁC HỆ THỐNG PHÂN TÁN (DISTRIBUTED)

- ▶ Phân phối tính toán cho nhiều bộ xử lý vật lý.
- ▶ Còn được gọi là **hệ thống ghép đôi lỏng**: mỗi bộ xử lý có bộ nhớ riêng và giao tiếp với bộ xử lý khác thông qua nhiều đường giao tiếp khác nhau (bus tốc độ cao, đường điện thoại).
- ▶ Các lợi ích của hệ thống phân tán:
  - ▶ Chia sẻ tài nguyên
  - ▶ Tăng tốc độ tính toán – cân bằng tải
  - ▶ Tin cậy
- ▶ Yêu cầu hạ tầng cơ sở mạng: LAN hoặc WAN.

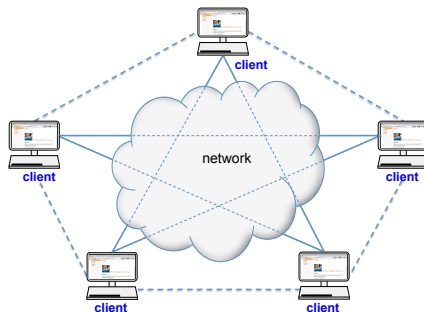
# HỆ THỐNG PHÂN TÁN CLIENT – SERVER

- ▶ Một số hệ thống tập trung hoạt động như hệ máy phục vụ, thỏa mãn các yêu cầu phát sinh bởi hệ thống khách hàng.



# HỆ THỐNG PHÂN TÁN PEER-TO-PEER

- ▶ Các máy tính tham gia vào hệ thống là ngang hàng, không phân biệt client hay server:
  - ▶ nếu **có dịch vụ** thì **đăng ký** với trung tâm tìm kiếm/dịch vụ mạng
  - ▶ nếu muốn **yêu cầu dịch vụ** thì dùng **discovery protocol**



## CÁC HỆ THỐNG CỤM (CLUSTERED)

- ▶ Hai hay nhiều máy tính được **nhóm lại với nhau** sao cho chúng hoạt động **như một máy tính độc nhất**.
- ▶ **Mục đích**: chia sẻ thiết bị lưu trữ, cân bằng tải, xử lý song song.
- ▶ Cung cấp khả năng sẵn dùng, chịu lỗi và độ tin cậy cao.
- ▶ **Ghép cụm bất đối xứng** (asymmetric clustering): các server chạy ứng dụng trong khi một server khác ở trạng thái chờ (hot standby); Khi server hoạt động bị lỗi, server chờ sẽ hoạt động.
- ▶ **Ghép cụm đối xứng** (symmetric clustering): tất cả các hosts cùng chạy ứng dụng và chúng kiểm soát lẫn nhau để thay thế công việc cho nhau.

# CÁC HỆ THỐNG THỜI GIAN THỰC (REAL-TIME)

- ▶ Thường được sử dụng như là một thiết bị điều khiển trong một **ứng dụng dạng chuyên biệt** (special-purpose):
  - ▶ Điều khiển các thí nghiệm khoa học
  - ▶ Các hệ thống điều trị y khoa
  - ▶ Các hệ thống điều khiển trong công nghiệp, quân sự
  - ▶ Một số hệ thống hiển thị, ...
- ▶ Hệ thống có các ràng buộc về thời gian cố định được định nghĩa chính xác.
- ▶ Hai loại hệ thống thời gian thực: cứng (hard) và mềm (soft).

# CÁC HỆ THỐNG THỜI GIAN THỰC “CỨNG”

- ▶ Đảm bảo các tác vụ tới hạn phải **hoàn thành đúng giờ** → các trì hoãn phải bị hạn chế.
- ▶ Hạn chế hoặc ~~không dùng các thiết bị lưu trữ thứ cấp~~, dữ liệu được trữ trong bộ nhớ ngắn kỳ (short-term) hoặc ROM.
- ▶ Mâu thuẫn với các hệ thống chia thời gian → không được hỗ trợ bởi các hệ điều hành đa năng.

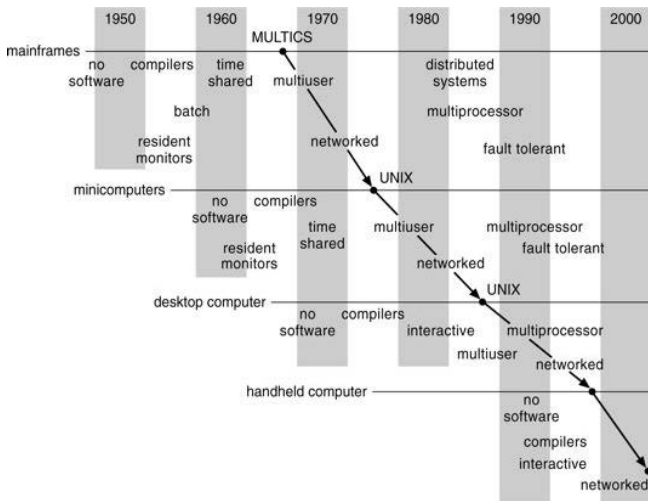
# CÁC HỆ THỐNG THỜI GIAN THỰC “MỀM”

- ▶ Tác vụ thời thực tối hạn **có độ ưu tiên cao hơn** và được duy trì cho đến khi hoàn thành.
- ▶ Có thể được dùng trong các hệ điều hành đa năng.
- ▶ Không hỗ trợ tốt cho thời điểm tối hạn (deadline) → dễ rủi ro → ít được dùng trong điều khiển công nghiệp hoặc robotics.
- ▶ Hữu dụng trong các ứng dụng yêu cầu các tính năng cao cấp của hệ điều hành (đa phương tiện, thực tại ảo).

# CÁC HỆ THỐNG CẦM TAY (HANDHELD)

- ▶ Bao gồm các loại thiết bị như:
  - ▶ Các máy hỗ trợ cá nhân kỹ thuật số (PDA - Personal Digital Assistant).
  - ▶ Điện thoại di động (Cellular phone).
- ▶ Các vấn đề:
  - ▶ Bộ nhớ giới hạn
  - ▶ Các bộ xử lý chậm
  - ▶ Màn hình nhỏ

# SỰ PHÁT TRIỂN CỦA HỆ ĐIỀU HÀNH



# CÁC MÔI TRƯỜNG ĐIỆN TOÁN

Cách HĐH được dùng trong việc thiết đặt môi trường tính toán:

- ▶ Tính toán truyền thống (traditional computing):
  - ▶ Môi trường office, home, thông qua network
- ▶ Tính toán kiểu web (web-based computing):
  - ▶ Mở rộng môi trường tính toán thông qua nền web (web-based)
  - ▶ Hỗ trợ nhiều thiết bị: workstations, handheld PDAs, cellular phones
- ▶ Tính toán kiểu hệ thống nhúng (embedded computing):
  - ▶ Các máy tính chạy các embedded real-time OS
  - ▶ Phục vụ các tác vụ chuyên biệt

