



Tổng quan Hệ điều hành

Môn học: Hệ điều hành

Nội dung

- Phát biểu hệ điều hành là gì? Giải thích vai trò của hệ điều hành trong hệ thống máy tính
- Trình bày lịch sử phát triển và đặc điểm của một số loại hệ điều hành
- Liệt kê các thành phần chính của một hệ điều hành
- Trình bày đặc điểm của một số kiến trúc hệ điều hành
- Trình bày quá trình khởi động máy tính

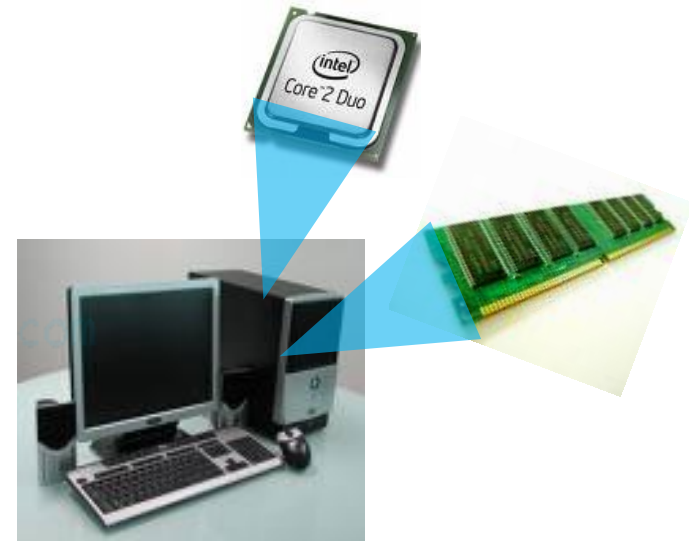


Dẫn nhập

- Đã nghe:
 - Máy tính là thiết bị tính toán nhiều tính năng
 - Máy tính gồm 3 thành phần:
 - Bộ xử lý
 - Bộ nhớ
 - Các thiết bị nhập xuất

- Đã sử dụng máy tính:
 - Chơi game
 - Nghe nhạc
 - Lập trình

- Đã sử dụng máy tính như thế nào, có dễ không?



Hệ điều hành là gì?

Ứng dụng / Người dùng

OS

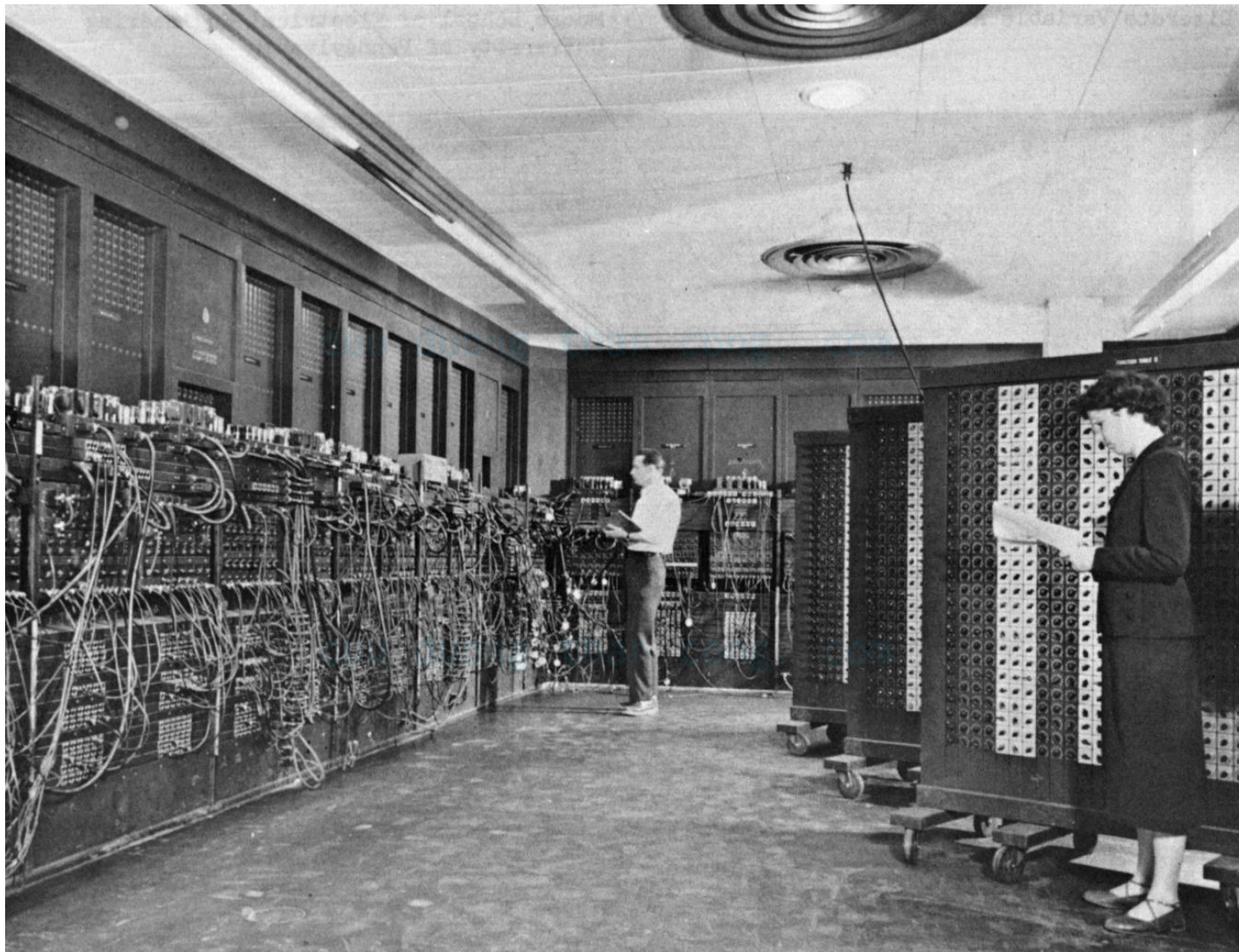
Phần cứng

cuuduongthancong.com

- Một lớp phần mềm ở giữa phần cứng và các chương trình ứng dụng/người dùng.
- Ảo hóa (*virtualize*) các thành phần phần cứng nhằm giúp việc giao tiếp: dễ dàng và an toàn
- Quản lý việc sử dụng chia sẻ các tài nguyên máy tính của các chương trình/người dùng: công bằng và hiệu quả
- Cung cấp một tập các hàm thư viện để đơn giản hóa việc phát triển ứng dụng



ENIAC – Máy tính hiện đại đầu tiên



...hệ thống xử lý theo lô (batch system)

- Máy tính mainframe: cung cấp khả năng tính toán
 - Làm sao chia sẻ thiết bị đắt tiền giữa nhiều người dùng?
 - Đăng kí giờ sử dụng
 - Đưa chương trình cho người sử dụng máy tính
 - Họ chạy chương trình và in kết quả trả lại cho bạn
 - Thông thường, chương trình nằm trong các thẻ đọc (card) được phân cách bởi các thẻ điều khiển (control card). Đầu đọc thẻ (card reader) sẽ đọc dữ liệu trên các thẻ này
- Hệ điều hành (HĐH) được phát triển để thực hiện các công việc một cách tuần tự
 - HĐH luôn nằm trong bộ nhớ (Quản lý thường trú – Resident monitor)



Đa chương (Multiprogramming)

- CPU sẽ rảnh mỗi khi chương trình thực thi cần giao tiếp với thiết bị ngoại vi
 - Ý tưởng: khi một chương trình tương tác với thiết bị ngoại vi thì CPU sẽ phục vụ chương trình khác
- Hệ thống đa chương (Multiprogrammed system) ra đời
 - Nạp đồng thời nhiều chương trình vào đĩa (sau này là vào bộ nhớ RAM)
 - Chuyển sang phục vụ chương trình khác nếu chương trình hiện thời đang tương tác với thiết bị ngoại vi
 - Thiết bị ngoại vi thường chậm hơn đĩa (sau này là bộ nhớ RAM)



Đa chương (Multiprogramming)

- Đặc điểm của hệ thống đa chương
 - Đồng thời thực hiện nhập/xuất của chương trình này và tính toán cho chương trình khác
 - Tiêu chí: *bộ xử lý luôn trong tình trạng làm việc*
 - Phải biết khi nào công việc nhập xuất xong: ngắt vs. polling

cuu duong than cong. com



Chia sẻ thời gian (Time-Sharing)

- Vấn đề
 - Làm sao chia sẻ cùng một máy tính (lúc đó rất đắt) giữa nhiều người dùng và vẫn duy trì giao diện giao tiếp với người dùng?
- Chia sẻ thời gian cuuduongthancong.com
 - Nối nhiều thiết bị đầu cuối đầu cuối đến một máy tính
 - Điều phối sử dụng máy tính cho nhiều người dùng
 - Chuyển đổi phục vụ giữa các chương trình người dùng sao cho đủ nhanh để người sử dụng có thể tương tác với chương trình trong khi chúng đang chạy (tạo cảm giác mỗi người dùng đang dùng máy riêng của mình)



HỆ THỐNG THỜI GIAN THỰC (Real-Time OS)

- Thực thi các ứng dụng có yêu cầu nghiêm ngặt về thời gian hoàn thành
- *Hard real-time system*
 - Hệ thống điều khiển bay, các hệ thống điều khiển công nghiệp, v.v..
 - Gây thảm họa nếu ta trễ hạn
 - Thách thức: làm sao không trễ hạn mà không phung phí nhiều tài nguyên
- *Soft real-time system*
 - Ứng dụng multimedia
 - Có thể gây khó chịu nhưng không đến nỗi thảm họa nếu bị quá hạn đôi chút!
 - Thách thức: khi hệ thống quá tải



Hệ điều hành song song (Parallel OS)

- Vài 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
- Cần HĐH hỗ trợ đồng bộ và truyền thông nhanh
- Nhiều kiến trúc song song khác nhau



HỆ THỐNG phân tán (Distributed OS)

- Clustering – Hệ thống phân tán cục bộ
 - Dùng nhiều máy nhỏ để phục vụ các công việc lớn
 - Rẻ hơn là dùng một máy tính lớn
 - Độ tin cậy cao hơn, tăng khả năng mở rộng
- Hệ thống phân tán trên diện rộng
 - Cho phép sử dụng tài nguyên phân tán
 - VD. sử dụng PC để truy cập Web
 - Không cần mang nhiều thông tin cần thiết theo mình
- HĐH hỗ trợ truyền thông và chia sẻ tài nguyên phân tán
 - VD. hệ lưu trữ tập tin trên mạng
- Quan tâm tính thực thi (mặc dù tăng tốc không phải là mục tiêu của HĐH này), độ tin cậy cao, sử dụng nguồn tài nguyên đa dạng

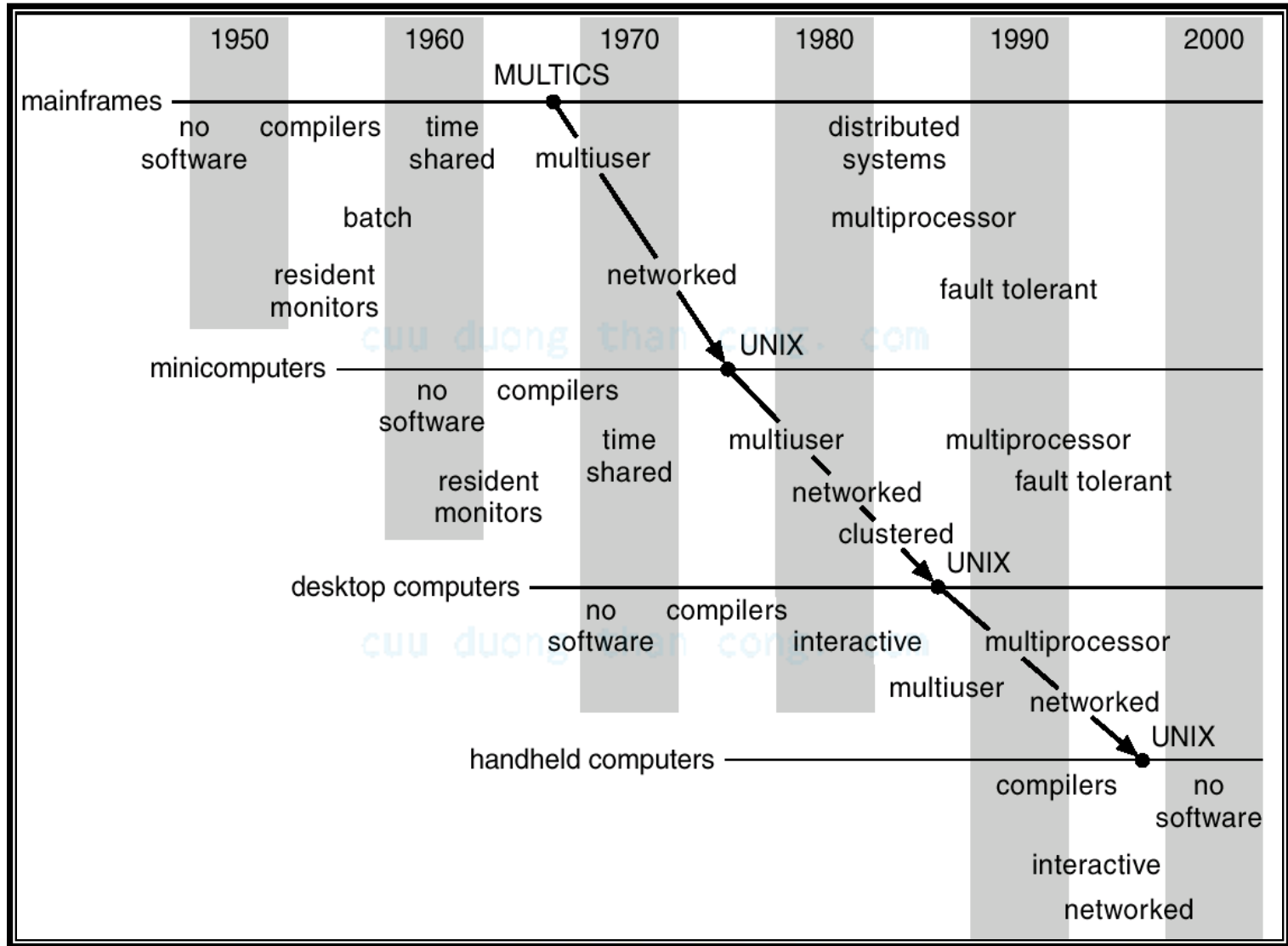


HỆ THỐNG nhúng (Embedded OS)

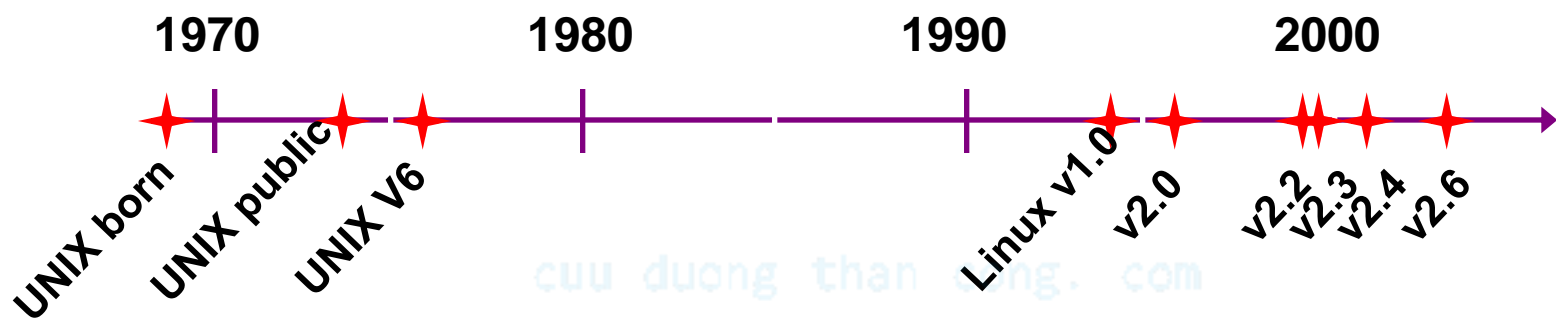
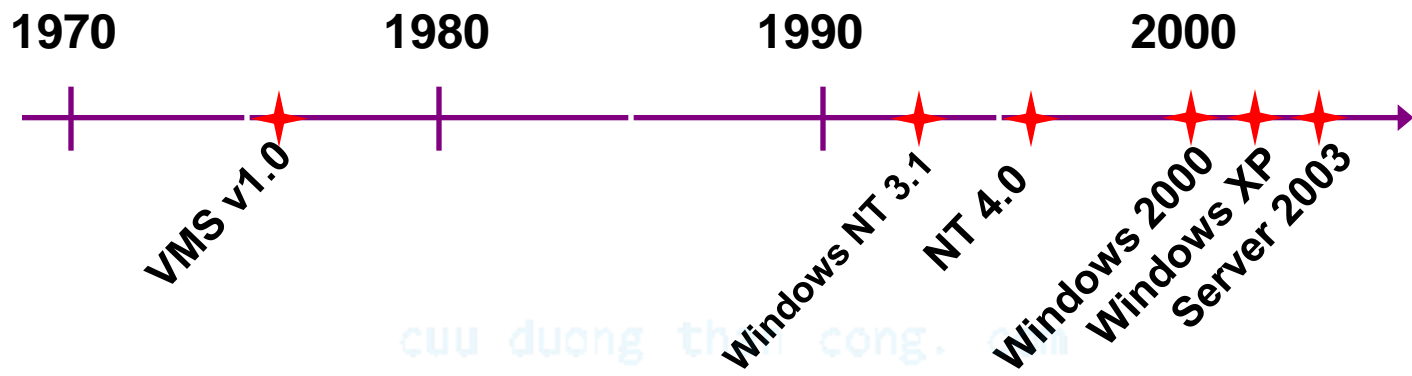
- Phát triển rộng khắp
 - Hiện thời: ĐTDD và PDA
 - Tương lai, các thiết bị tính toán ở mọi nơi
- Đặc tính
 - Tài nguyên hạn chế: CPU chậm, bộ nhớ nhỏ, không ổ đĩa, v.v.
 - Làm sao ta có thể chạy các chương trình mạnh hơn trong điều kiện tài nguyên phần cứng hạn chế?
 - Sử dụng nhiều thiết bị...
 - Tăng thêm các tiện ích trên thiết bị
- HĐH giúp quản lý năng lượng, tính di động, tìm kiếm tài nguyên, v.v.v.



Quá trình phát triển các khái niệm và tính năng HĐH



Quá trình phát triển của Windows và Linux



(xem thêm <http://www.levenez.com>)



Câu hỏi

- Làm thế nào các chương trình / ứng dụng người dùng có thể giao tiếp với HĐH?

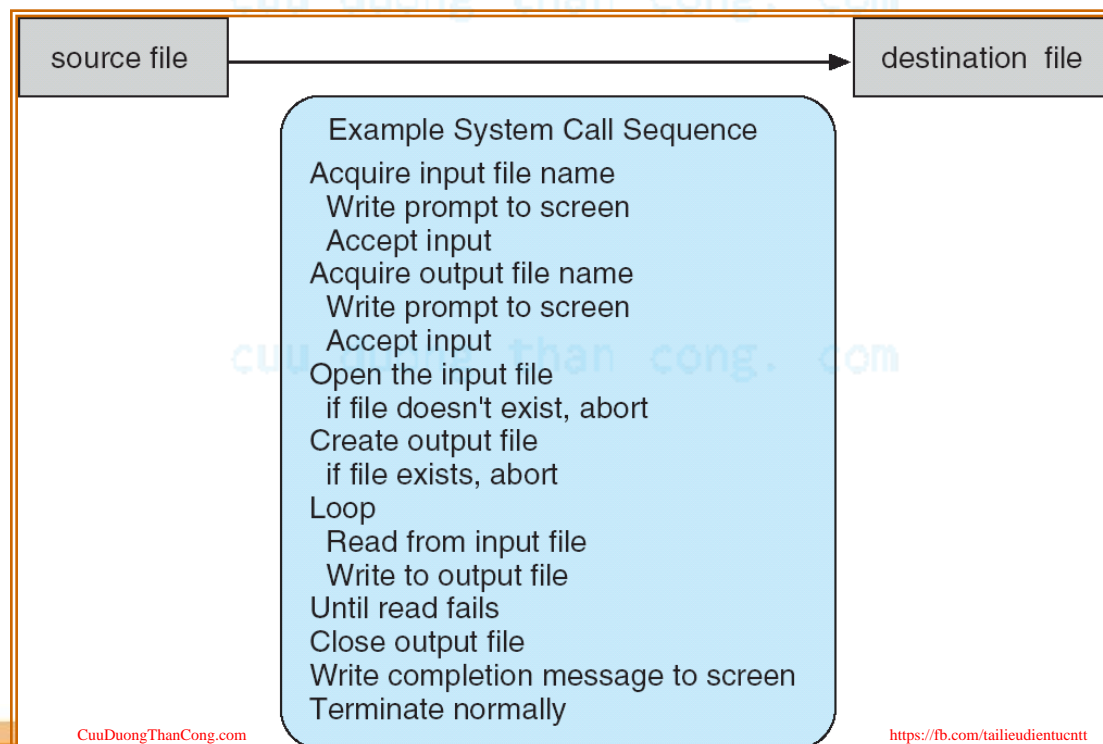
cuu duong than cong. com

cuu duong than cong. com



Lời gọi hệ thống (System call)

- Là tập các hàm cơ bản nhất của hệ điều hành để phục vụ các yêu cầu từ các chương trình người dùng
- Ví dụ, chuỗi các lời gọi hệ thống được thực hiện để sao chép nội dung của một tập tin sang một tập tin khác

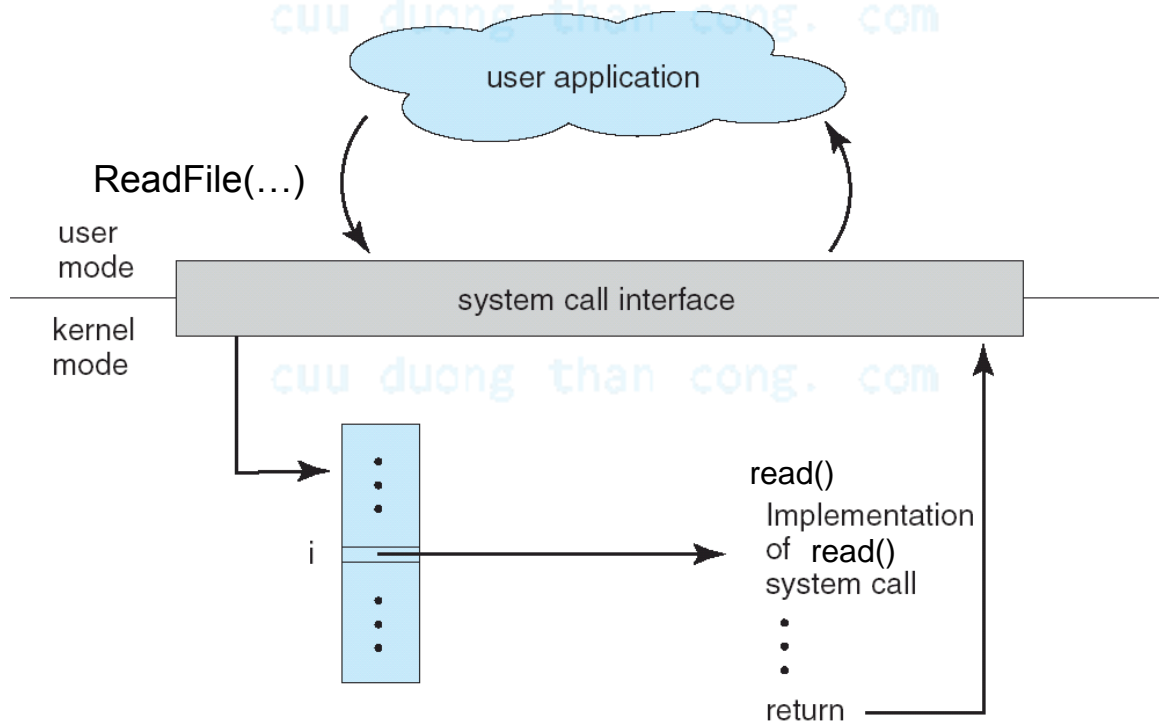
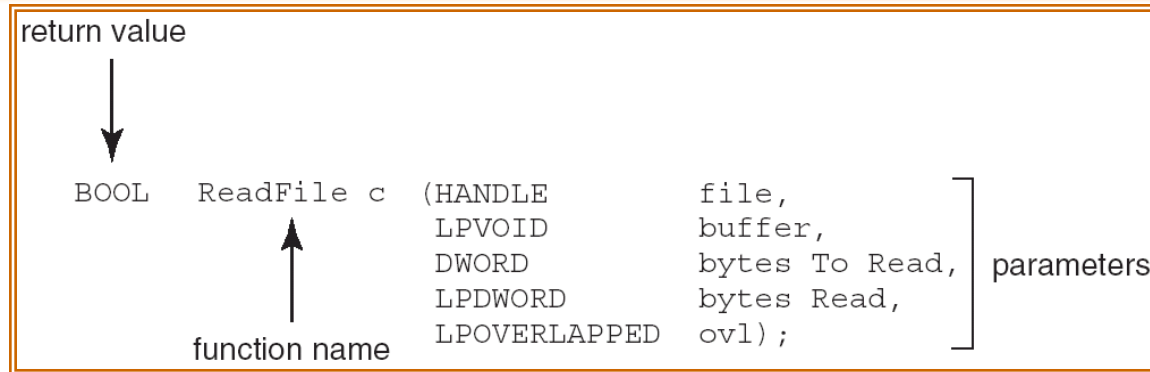


Hàm thư viện lập trình

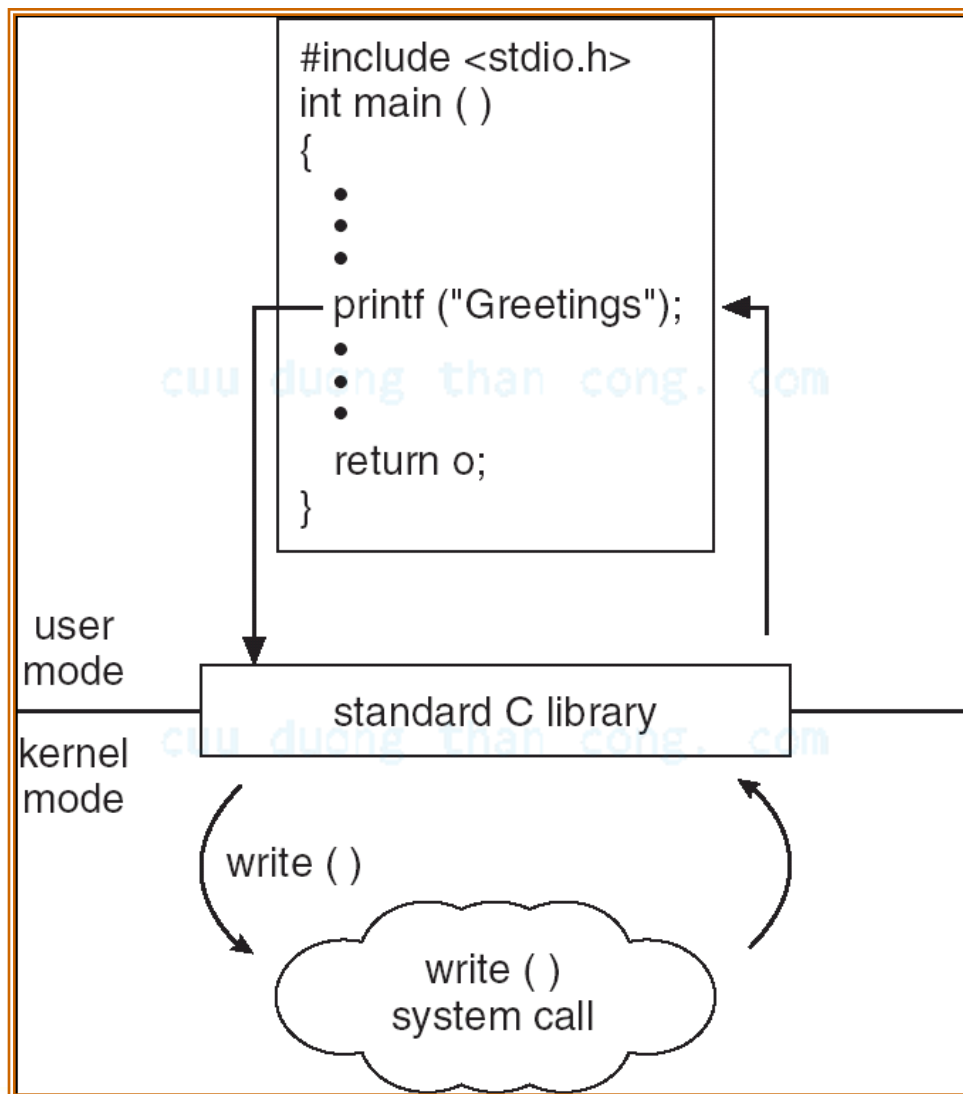
- Thông thường, chương trình người dùng không gọi trực tiếp các lời gọi hệ thống của hệ điều hành
- Hệ điều hành thường cung cấp bộ thư viện các hàm lập trình, chẳng hạn Win32 API, POSIX API (application programming interface), giúp việc lập trình dễ dàng hơn phải dùng các lời gọi hệ thống
- Tuy nhiên, thông thường người lập trình thường dùng các hàm thư viện của các ngôn ngữ lập trình như thư viện C, Java,... (do dễ sử dụng hơn các hàm thư viện của hệ điều hành)
- Cho dù là sử dụng hàm ngôn ngữ lập trình hay hàm thư viện hệ điều hành thì cuối cùng cũng sẽ chuyển thành các lời gọi hệ thống tương ứng



API và Lời gọi hệ thống

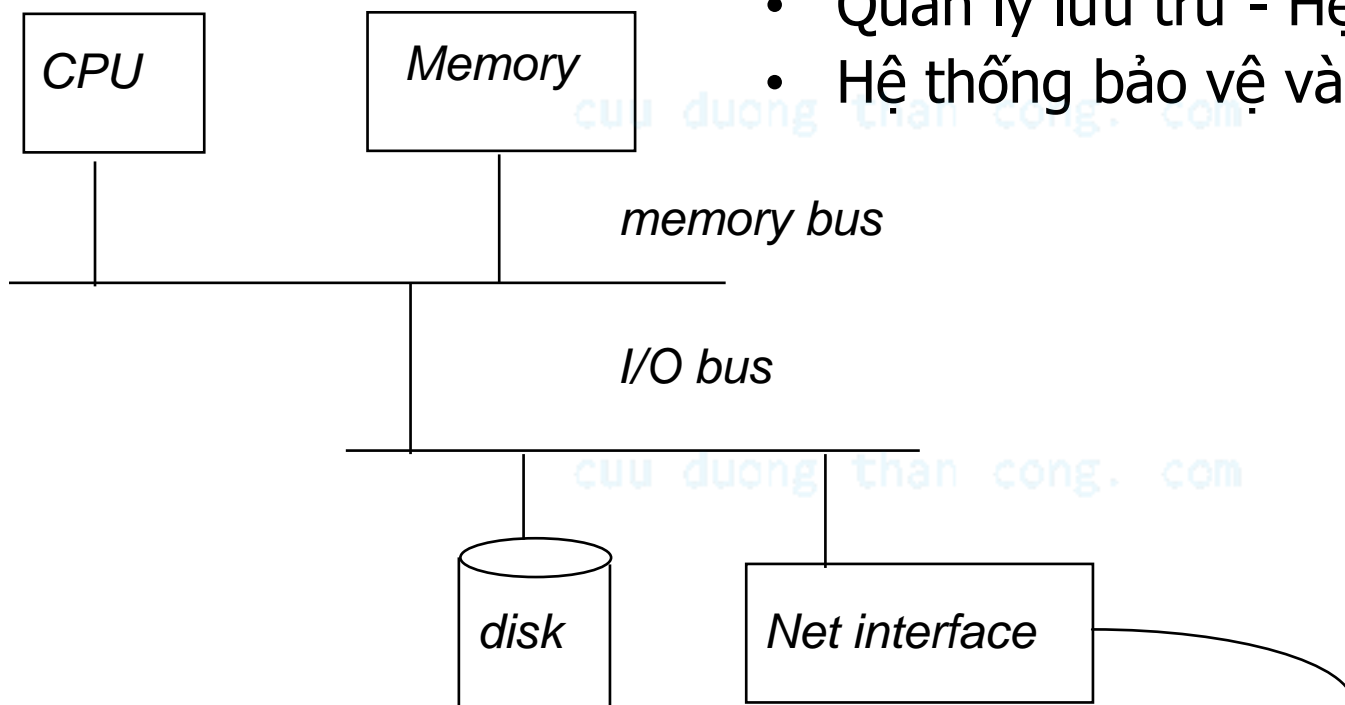


Hàm thư viện C và Lời gọi hệ thống



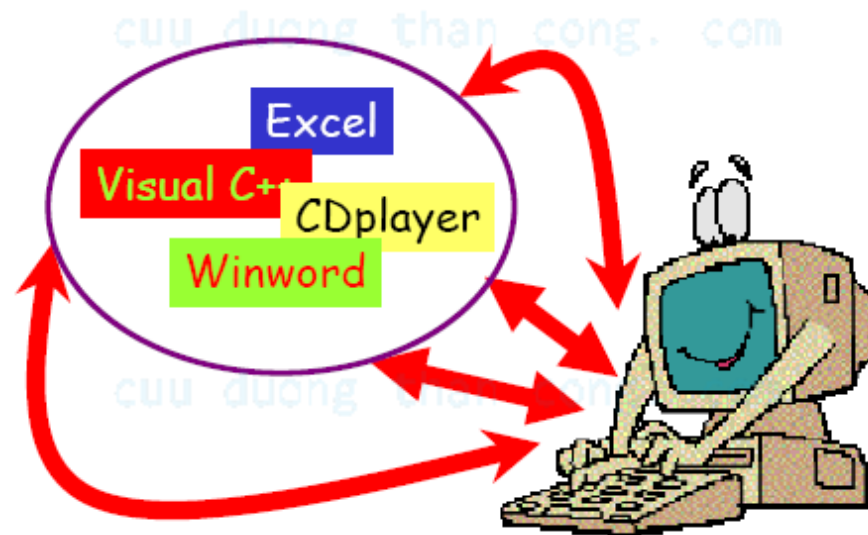
Các thành phần của hệ điều hành

- Quản lý bộ xử lý – Quản lý tiến trình
- Quản lý bộ nhớ
- Quản lý nhập xuất
- Quản lý lưu trữ - Hệ thống tập tin
- Hệ thống bảo vệ và bảo mật



Quản lý tiến trình

- Tiến trình là chương trình đang thực thi.
- Một tiến trình cần sử dụng các tài nguyên: CPU, bộ nhớ, tập tin, thiết bị nhập xuất để hoàn tất công việc của nó
- Hệ thống đa chương: sẽ có nhiều tiến trình chạy cùng lúc.

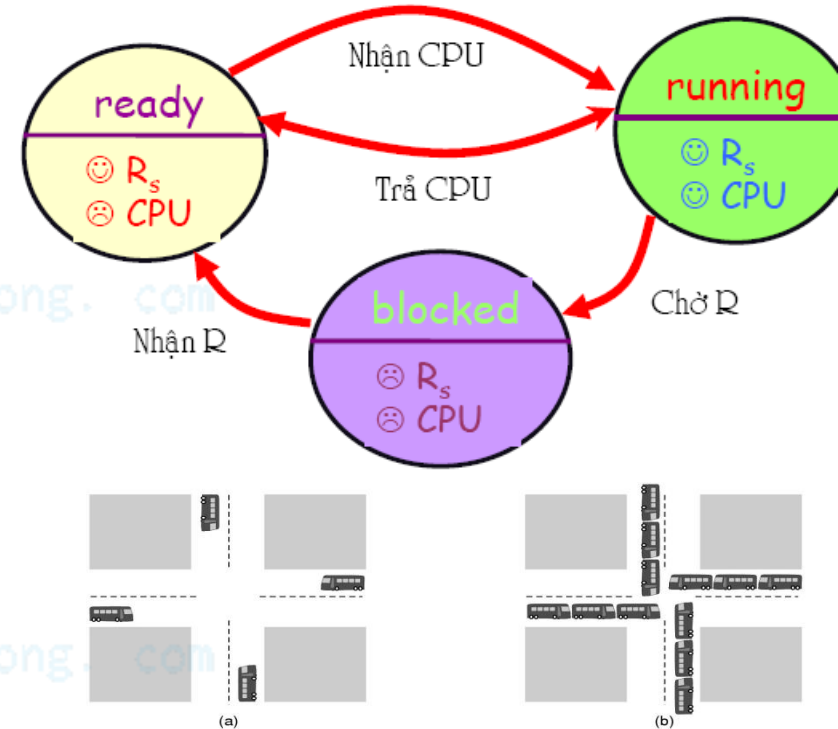


- Số lượng tài nguyên \ll số lượng tiến trình chạy cùng lúc !!!
→ Tranh chấp



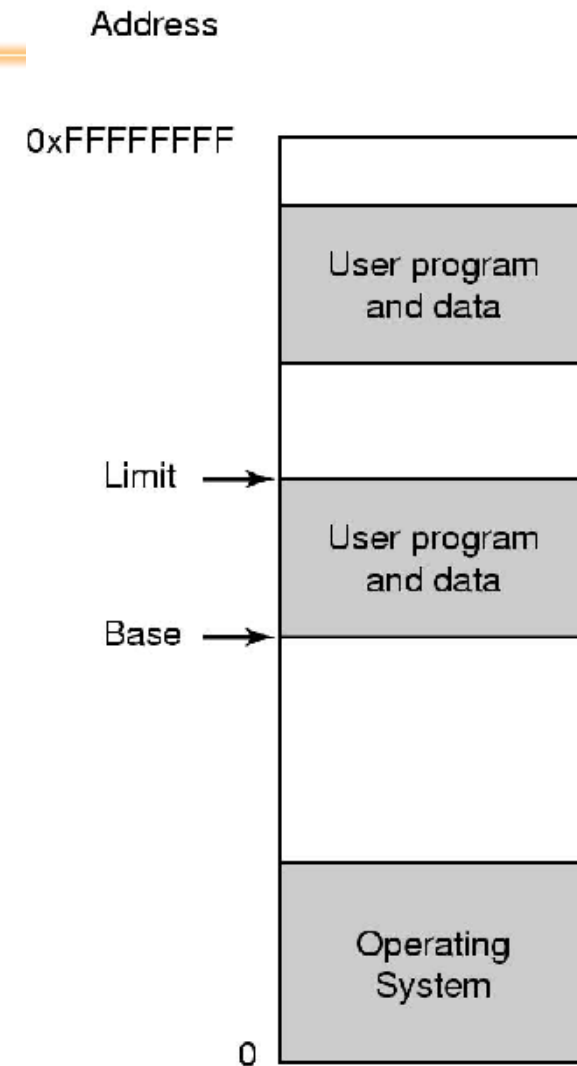
Quản lý tiến trình (tt)

- Vai trò của HĐH trong việc quản lý tiến trình
 - Tạo, hủy, tạm dừng và thực hiện tiếp tiến trình
 - Quản lý trạng thái các tiến trình
 - Điều phối tiến trình: FIFO, Round Robin, SJF, ...
 - Cung cấp cơ chế đồng bộ tiến trình
 - Độc quyền truy xuất
 - Phối hợp hoạt động
 - Cung cấp cơ chế kiểm soát deadlock
 - Cung cấp cách thức trao đổi thông tin giữa các tiến trình
 - Chia sẻ tài nguyên dùng chung
 - Trao đổi thông điệp



Quản lý bộ nhớ

- Mọi chương trình (mã nguồn + dữ liệu) cần được nạp vào bộ nhớ chính để thi hành
- Nhiều tiến trình chạy đồng thời → Quản lý bộ nhớ sao cho tối ưu việc tận dụng CPU và đáp ứng kịp thời cho người sử dụng
- Vai trò của HĐH trong việc quản lý bộ nhớ:
 - Tổ chức cấp phát, thu hồi bộ nhớ khi cần thiết
 - Mô hình cấp phát (liên tục, không liên tục)
 - Quản lý không gian địa chỉ của tiến trình
 - Quản lý bộ nhớ ảo
 - Quyết định chương trình/ một phần chương trình nào được nạp vào/ ra bộ nhớ như thế nào



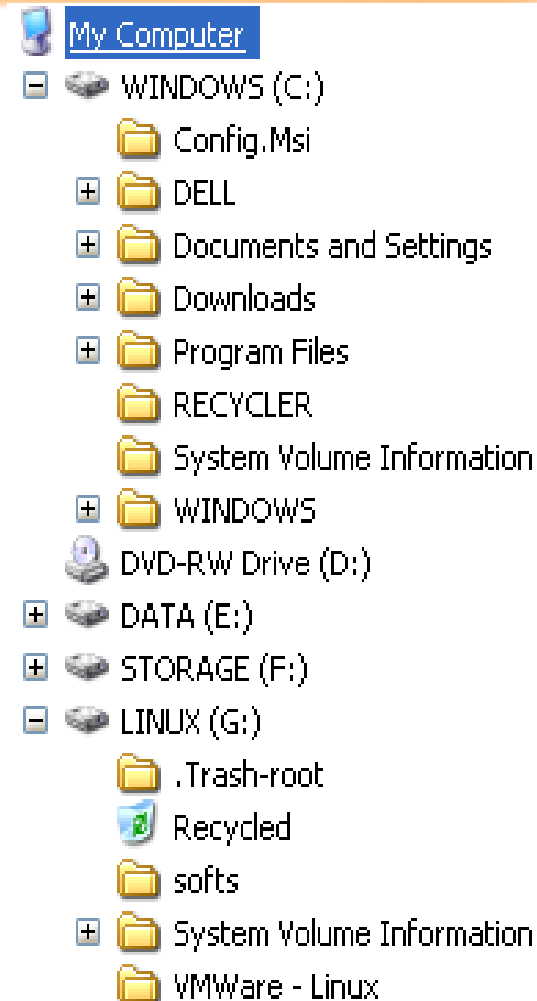
Quản lý nhập xuất

- Hệ thống quản lý nhập xuất chịu trách nhiệm:
 - Tạo môi trường giao tiếp đơn giản, đồng nhất với các thiết bị nhập xuất
 - Trình điều khiển thiết bị: che đi đặc thù phức tạp của các thiết bị nhập xuất đối với người sử dụng
 - Quản lý hiệu quả nhập xuất:
 - Điều phối yêu cầu nhập xuất
 - Tổ chức lưu trữ đệm (buffering, spooling,...)
 - Quản lý lỗi



Quản lý tập tin

- Nhiều loại thiết bị lưu trữ đa dạng về tốc độ truy xuất, đơn vị lưu trữ, phương thức truy xuất (đĩa cứng, USB, CD,...)
- HĐH cung cấp cái nhìn logic và đồng nhất về việc lưu trữ thông tin – **tập tin**
- Tập tin thường được tổ chức trong các thư mục
- Vai trò HĐH trong việc quản lý tập tin:
 - Tổ chức tập tin, thư mục trên đĩa
 - Hỗ trợ các thao tác trên tập tin và thư mục
 - Quản lý quyền truy cập
 - Sao lưu dự phòng tập tin trên các thiết bị lưu trữ



Bảo vệ và bảo mật

- Hệ thống máy tính luôn đứng trước các mối nguy cơ:
 - Khách quan: thiên tai, lỗi sử dụng, lỗi phần cứng, phần mềm.
 - Chủ quan:
 - Tấn công phá hoại: virus, worm, DoS, ...
 - Ăn cắp tài nguyên: trojan horses, trap doors, Man-in-the-middle, ...

cuu duong than cong. com

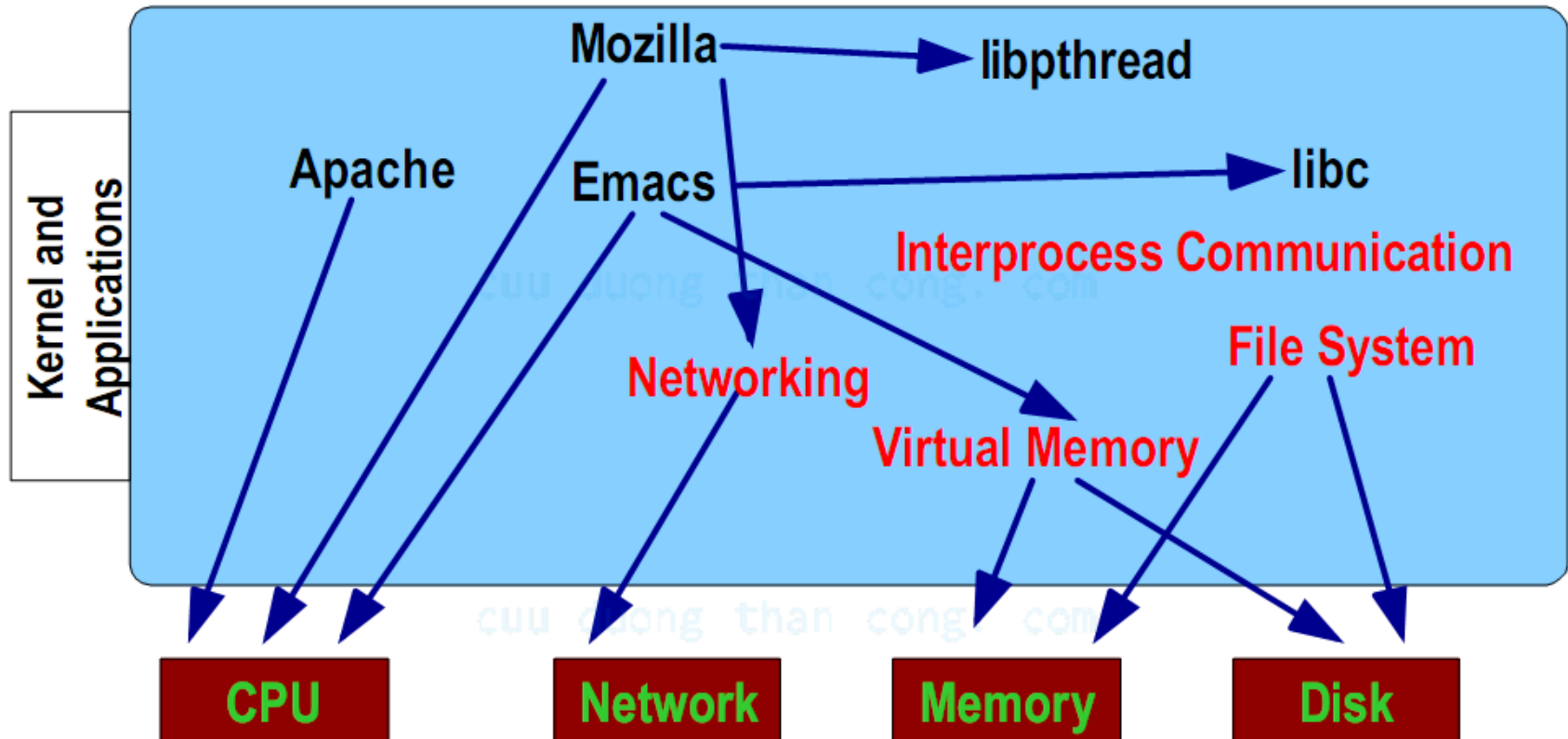


- Bảo vệ (protection) và Bảo mật (security):
 - Kiểm soát quá trình truy xuất tài nguyên của tiến trình/ người dùng
 - Phòng thủ, chống lại các tấn công
- Một số cơ chế:
 - Hoạt động ở 2 chế độ (kernel mode vs. user mode)
 - Sao lưu dự phòng (Backup)
 - Xác thực người dùng (User Authentication)
 - Phân quyền (Authorization), chính sách bảo mật (Policy)
 - Kiểm soát nhật ký (Audit log)



Một số kiến trúc HĐH

Kiến trúc đơn giản



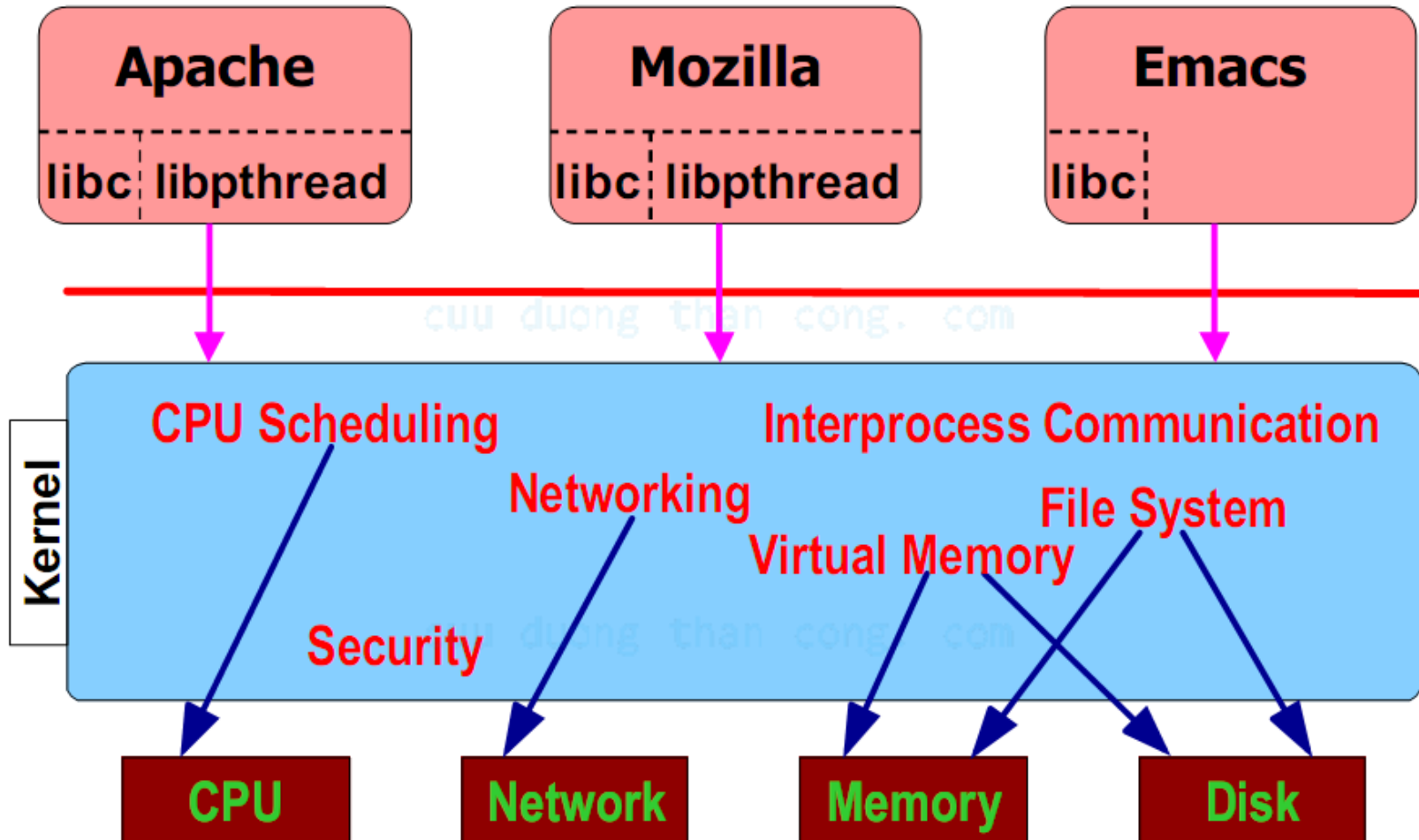
Kiến trúc đơn giản

- Các ứng dụng, thư viện, nhân HĐH nằm chung không gian địa chỉ
- Ưu điểm:
 - Xử lý yêu cầu rất nhanh
 - Dễ mở rộng
 - Phù hợp với môi trường đơn người dùng
- Nhược điểm
 - Không có bảo vệ giữa nhân HĐH và các ứng dụng
 - Những mở rộng có thể gây ra những xung đột
- MS-DOS, Mac OS 9, Windows ME, 98, 95, 3.1, Palm OS và HĐH cho một số hệ thống nhúng



Một số kiến trúc HĐH

Kiến trúc Monolithic



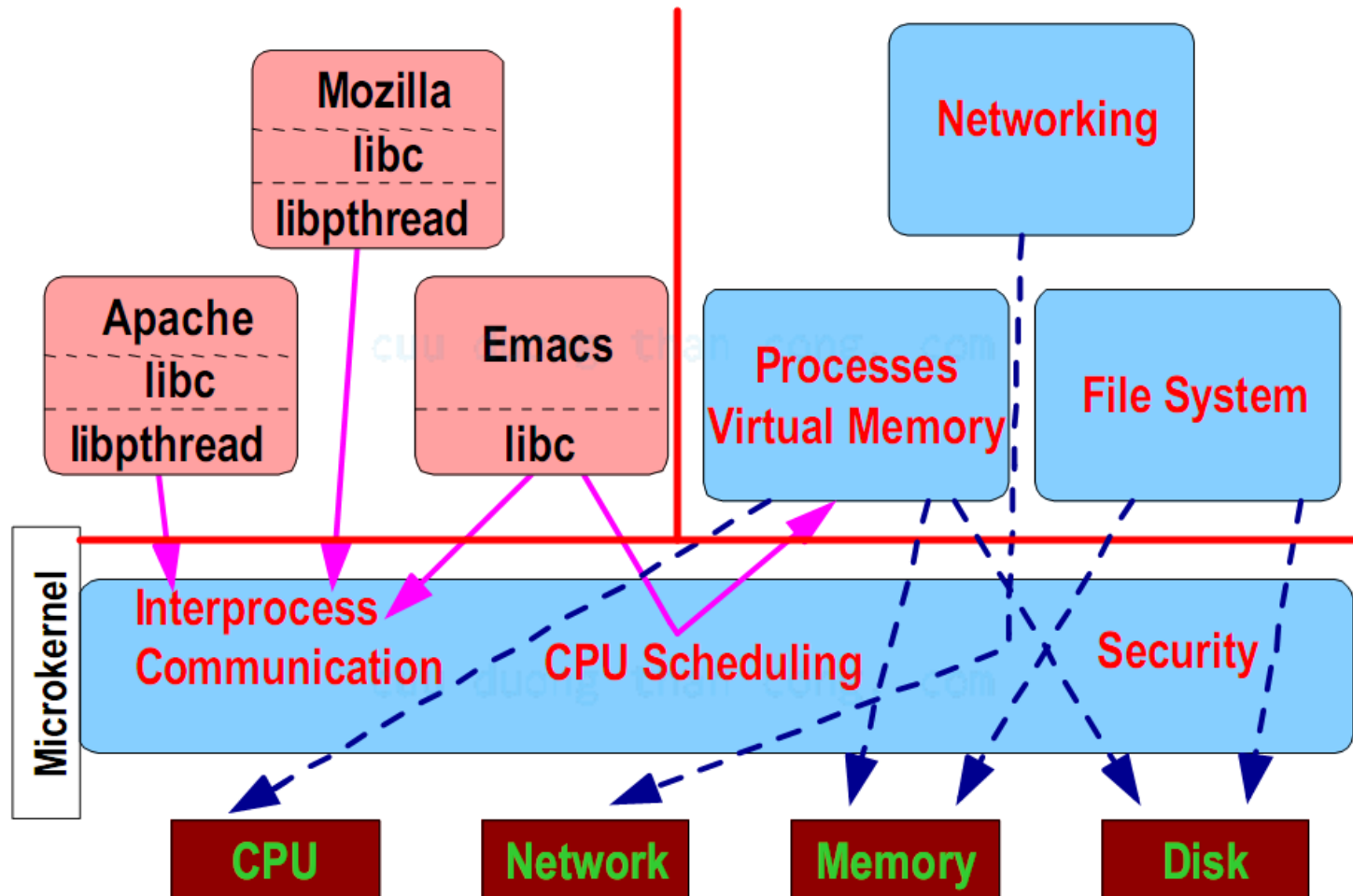
Kiến trúc Monolithic

- Nhân HĐH bao gồm toàn bộ các thành phần quản lý nằm ở không gian riêng biệt với các ứng dụng
- Ưu điểm:
 - Xử lý yêu cầu nhanh
 - Dễ hiểu đối với những người phát triển nhân HĐH
 - Bảo vệ phần nhân HĐH với các ứng dụng
- Nhược điểm:
 - Không có bảo vệ giữa các thành phần trong nhân HĐH
 - Cấu trúc phức tạp, khó mở rộng (do không có ranh giới rõ ràng giữa các thành phần)
- Unix, Windows NT/XP, Linux, BSD



Một số kiến trúc HĐH

Kiến trúc MicroKernel (Client/Server)



Kiến trúc MicroKernel

- Tiêu chí: nhân hệ điều hành (phụ thuộc phần cứng) càng đơn giản càng tốt
- Các thành phần quản lý của HĐH chạy như các dịch vụ server ở mức người dùng
- Một dịch vụ server bị trục trặc không làm “treo” toàn bộ hệ thống
- Các chương trình ứng dụng hoạt động như các client yêu cầu các dịch vụ server bằng các gửi thông điệp thông qua nhân HĐH
- Mach, Chorus, QNX, GNU Hurd



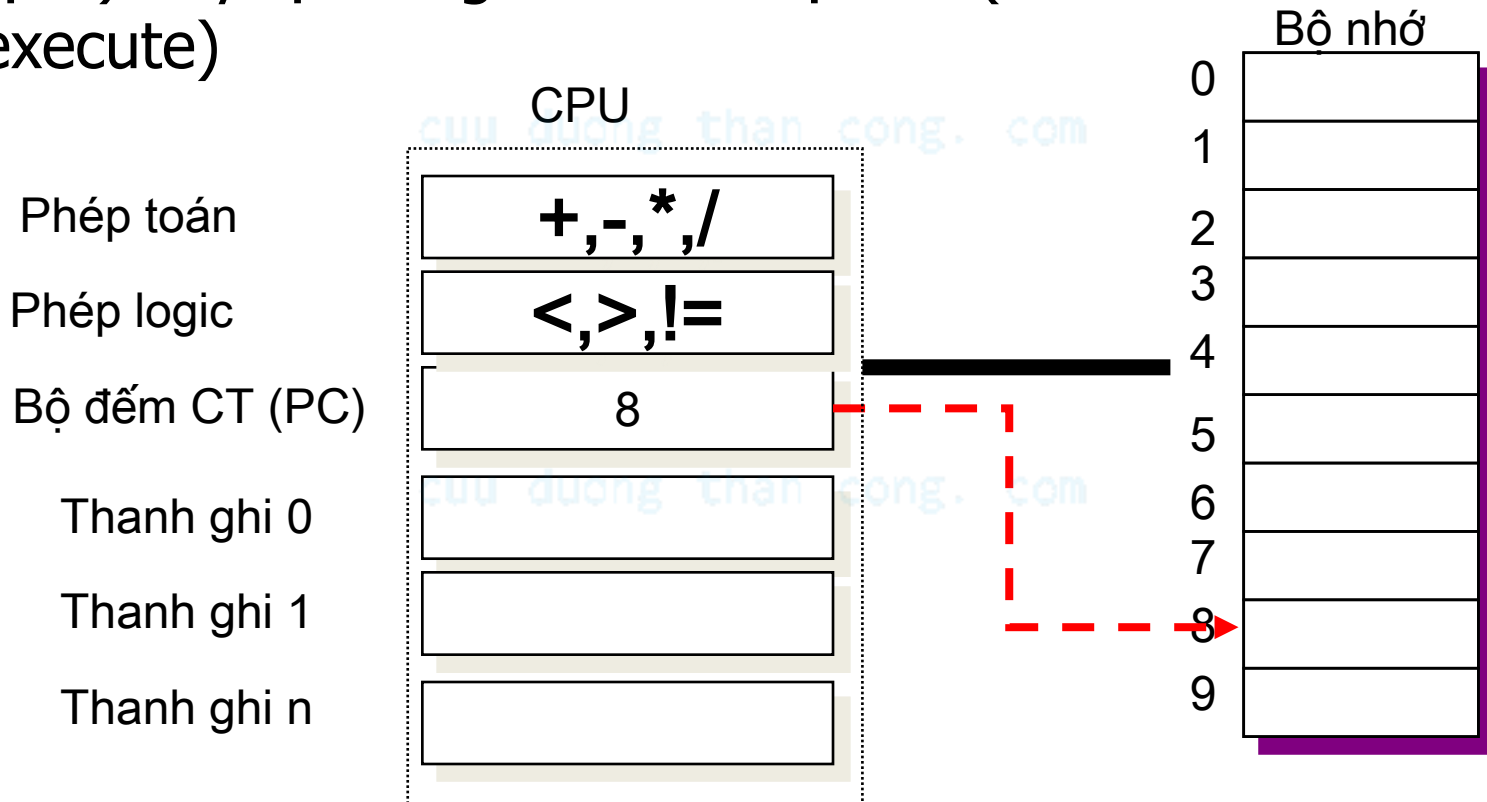
Kiến trúc MicroKernel

- Ưu điểm:
 - Dễ mở rộng
 - Thêm tính năng mới cho HĐH bằng cách thêm 1 dịch vụ server
 - Phần nhân không xác định môi trường HĐH
 - Cho phép hỗ trợ chạy nhiều HĐH
 - Cần 1 server giả lập cho mỗi hệ thống (Mac, Windows, Unix)
 - Tất cả các ứng dụng cùng chạy trên microkernel
 - Có thể tùy chọn HĐH để chạy ứng dụng
 - Các dịch vụ của HĐH độc lập phần cứng
 - Các thành phần của HĐH được bảo vệ tốt
 - Dễ mở rộng cho hệ thống nhiều bộ xử lý và hệ thống phân tán
- Khuyết điểm
 - Xử lý yêu cầu chậm
 - Khó khăn trong việc triển khai hệ thống



Nhắc lại qui trình làm việc của CPU

- Bộ xử lý gồm: ALU, CU, Register
- Quá trình bộ xử lý thực thi yêu cầu (dưới dạng các lệnh): lấy lệnh – giải mã – thực thi (fetch – decode – execute)



Quá trình khởi động máy tính (1)

- Hệ điều hành nằm ở đâu ?
- Làm sao để máy tính nạp và chạy HĐH lúc khởi động?
 - Quá trình để khởi động HĐH gọi là booting
- Quá trình khởi động của các máy hiện đại gồm 3 giai đoạn
 - CPU thực thi lệnh từ địa chỉ cố định biết trước (boot ROM)
 - Firmware nạp boot loader
 - Boot loader nạp HĐH
- (1) CPU thực thi lệnh từ địa chỉ biết trước trong bộ nhớ
 - Địa chỉ vùng nhớ này thường trỏ tới vùng nhớ chỉ đọc (ROM – read-only memory)
 - Với x86, địa chỉ này là 0xFFFF0, trỏ tới địa chỉ chương trình BIOS (basic input-output system) trong ROM



Quá trình khởi động máy tính (2)

- (2) ROM chứa mã nguồn "boot"
 - Loại phần mềm chỉ đọc này gọi là **firmware**
 - Với x86, chương trình BIOS thực hiện lần lượt các công việc:
 - Kiểm tra cấu hình trong CMOS (complementary metal oxide semiconductor)
 - Nạp trình quản lý ngắt (interrupt handler) và các trình điều khiển thiết bị
 - Khởi tạo các thanh ghi và quản lý nguồn cung cấp (power management)
 - Thực hiện quá trình kiểm tra phần cứng (POST – power-on self-test)
 - Hiển thị các thiết lập hệ thống
 - Xác định các thiết bị có khả năng khởi động
 - Tiếp tục quá trình khởi động
 - Nạp và thực thi chương trình boot loader.



Thực thi firmware

Phoenix - AwardBIOS v6.00PG, An Energy Star Ally
Copyright (C) 1984-2002, Phoenix Technologies, LTD



ASUS A7N8X2.0 Deluxe ACPI BIOS Rev 1008

Main Processor : AMD Athlon(tm) XP 2400+
Memory Testing : 1048576K OK

Memory Frequency is at 200 MHz , Dual Channel mode

Primary Master : SAMSUNG SU4084H PM100-21

Primary Slave : SAMSUNG SP4002H QU100-60

Secondary Master : Pioneer DVD-ROM ATAPIModel DVD-105S 0133 E1.33

Secondary Slave : SAMSUNG CF/ATF

Phoenix Technologies, LTD
System Configurations

CPU Type	: AMD Athlon(tm) XP	Base Memory	: 640K
CPU ID	: 0681	Extended Memory	: 1047552K
CPU Clock	: 2000MHz	L1 Cache Size	: 128K
		L2 Cache Size	: 256K

Diskette Drive A	: 1.44M, 3.5 in.	Display Type	: EGA/UGA
Pri. Master Disk	: LBA,ATA 100,40822MB	Serial Port(s)	: 3F8 2F8
Pri. Slave Disk	: LBA,ATA 100,40062MB	Parallel Port(s)	: 378
Pri. Master Disk	: DVD,ATA 33	DDR DIMM at Rows	: 2 3 4 5
Sec. Slave Disk	: CHS,PIO 4, 512MB		

Press DEL to enter SETUP ; press
08/04/2004-nVidia-nForce-A7N8X2.0

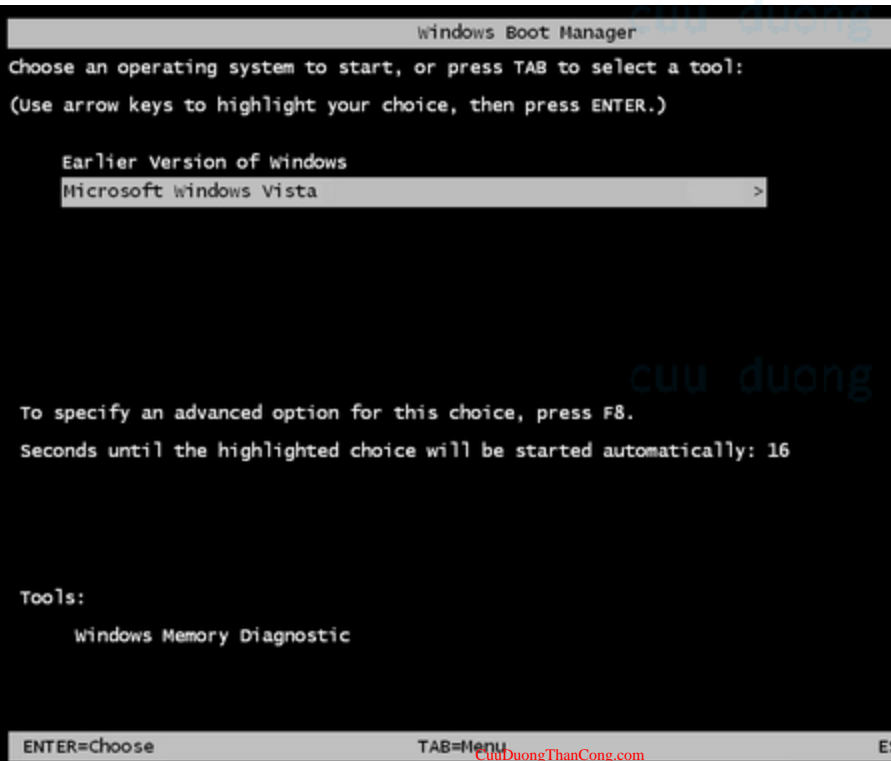
PCI device listing ...

Bus No.	Device No.	Func No.	Vendor/Device	Class	Device Class	IRQ
0	2	0	10DE 0067	0C03	USB 1.0/1.1 OHCI Controller	10
0	2	1	10DE 0067	0C03	USB 1.0/1.1 OHCI Controller	11
0	2	2	10DE 0068	0C03	USB 2.0 EHCI Controller	5
0	9	0	10DE 0065	0101	IDE Controller	14
0	13	0	10DE 006E	0C00	Serial Bus Controller	10
1	8	0	1106 3043	0200	Network Controller	11
1	9	0	1102 0002	0401	Multimedia Device	11



Quá trình khởi động máy tính (3)

- (3) **Boot loader** sau đó nạp phần còn lại của HĐH. Chú ý rằng tại thời điểm này HĐH vẫn chưa chạy
 - Boot loader hiểu được nhiều hệ điều hành khác nhau
 - Boot loader hiểu được nhiều phiên bản khác nhau của các HĐH
 - Đã bao giờ nghe “*dual boot*” ?



Nạp hệ điều hành

