

HỆ ĐIỀU HÀNH

Chủ đề 1: Tổng quan

Trường Đại học Nha Trang
Khoa Công nghệ thông tin
Bộ môn Hệ thống thông tin
Giáo viên: Ts.Nguyễn Khắc Cường

CHƯƠNG 1

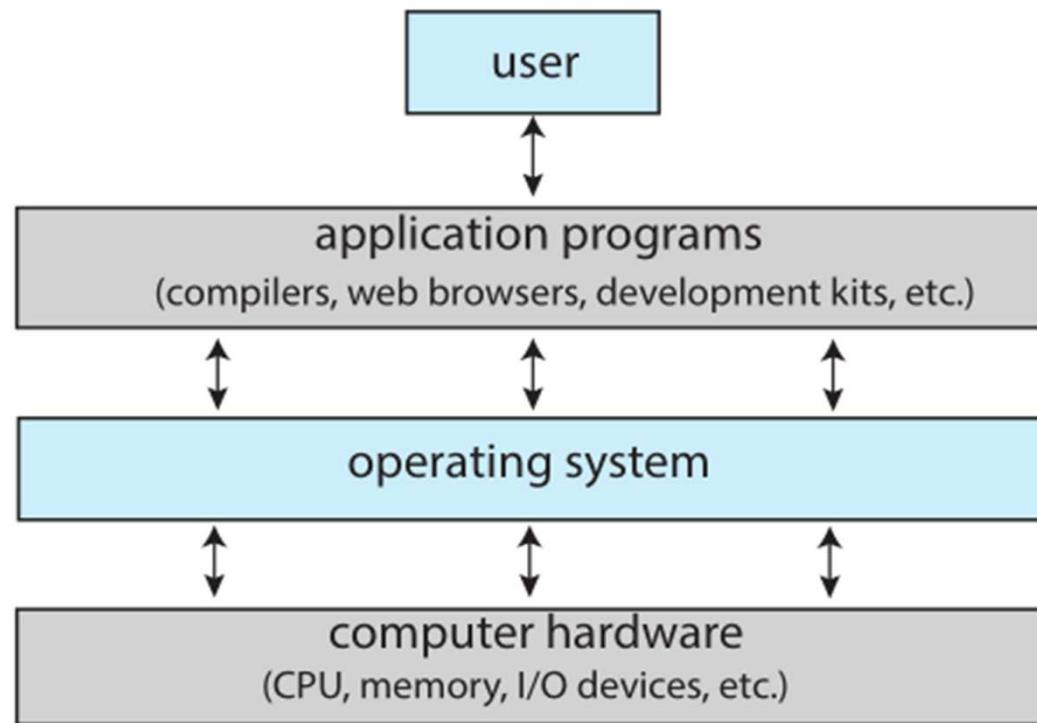
GIỚI THIỆU

Giới thiệu sơ lược HĐH

- HĐH có mặt ở rất nhiều hệ thống trong thực tế:
 - ô tô
 - các thiết bị điện dân dụng (máy điều hòa, máy giặt, ...)
 - điện thoại thông minh
 - máy tính cá nhân
 - máy tính công nghiệp
 - máy tính mạng
 - ...

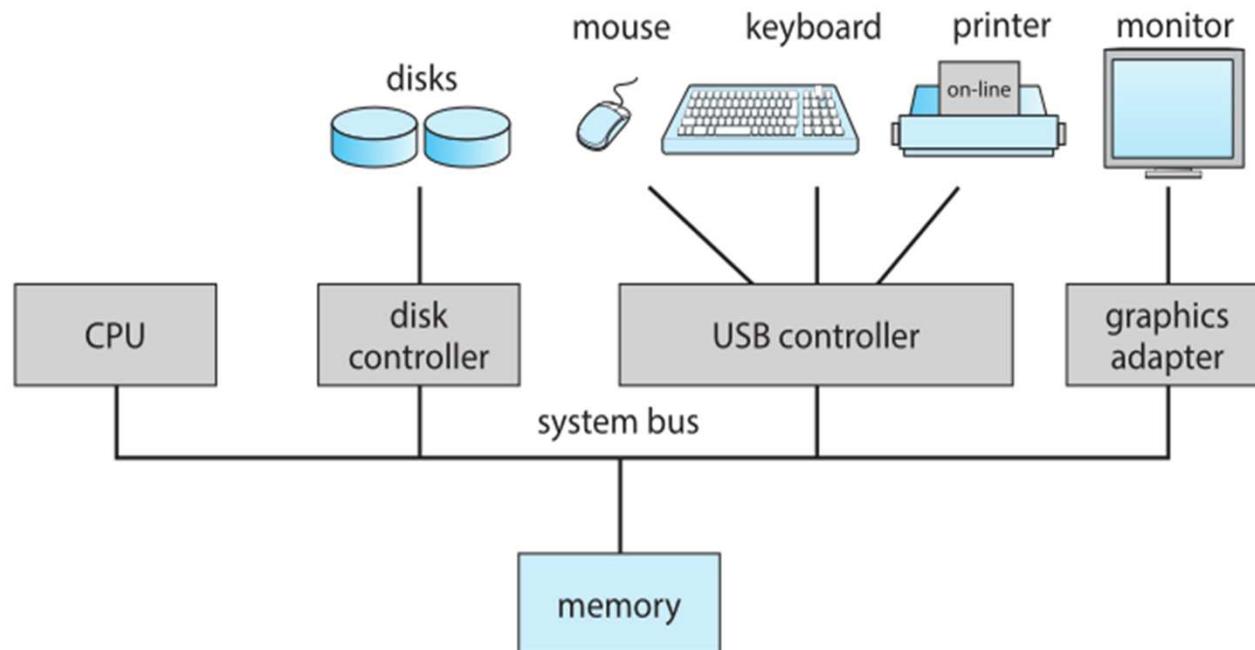
Định nghĩa HĐH

- HĐH là hệ thống phần mềm phức tạp, có nhiệm vụ:
 - quản lý tài nguyên của hệ thống tính toán (CPU time, memory space, storage space, I/O devices, ...)
 - làm bộ phận trung gian giữa người sử dụng và phần cứng hệ thống tính toán



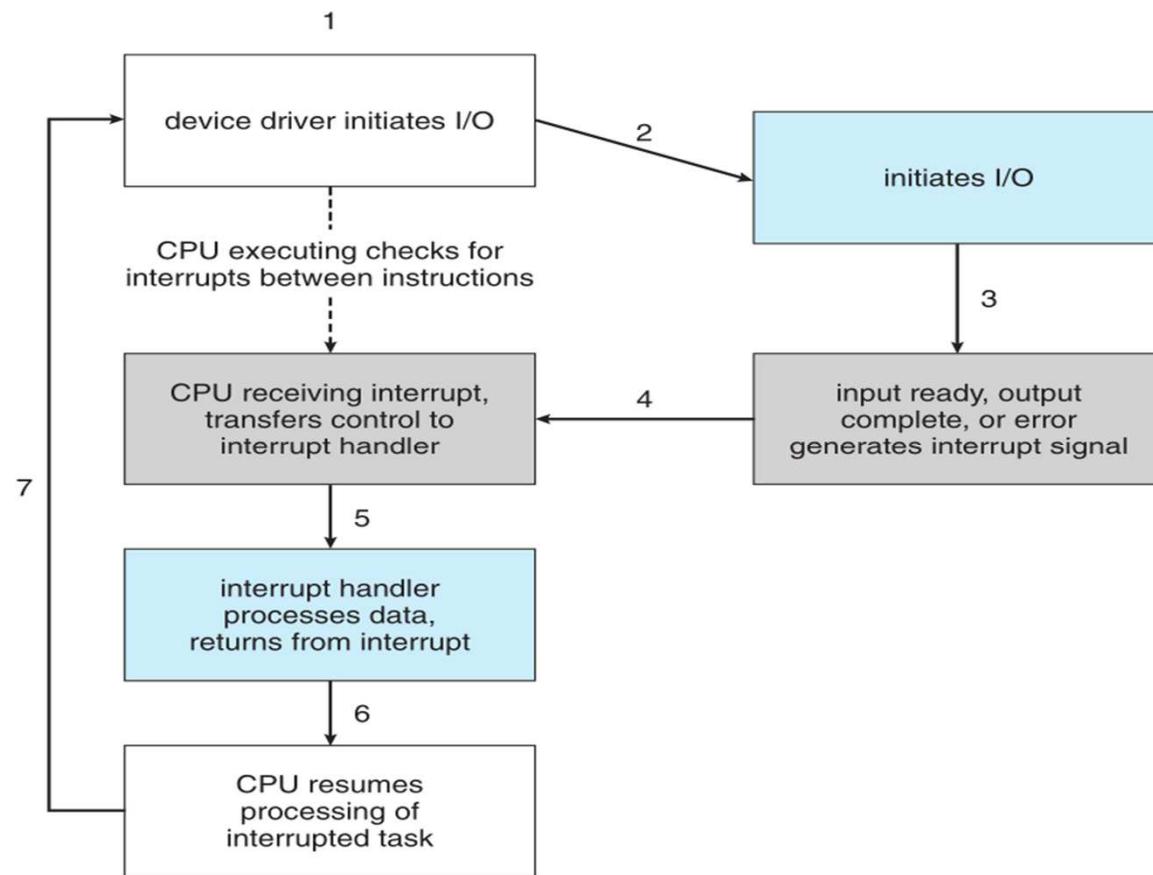
Tác dụng của HĐH - Phần cứng máy tính

- Tại sao cần hiểu cách hoạt động của HĐH?
 - Giúp việc lập trình tạo ra các ứng dụng hoạt động được chính xác, tối ưu, hiệu quả, ít xung đột gây ra lỗi, tăng độ tương thích, ...
 - Hỗ trợ lập trình ở mức hệ thống (bảo mật, điều khiển phần cứng, ...)
- Các phần cứng điển hình của một máy tính



Interrupts

- Interrupt = ngắt
- Ví dụ sử dụng ngắt trong I/O



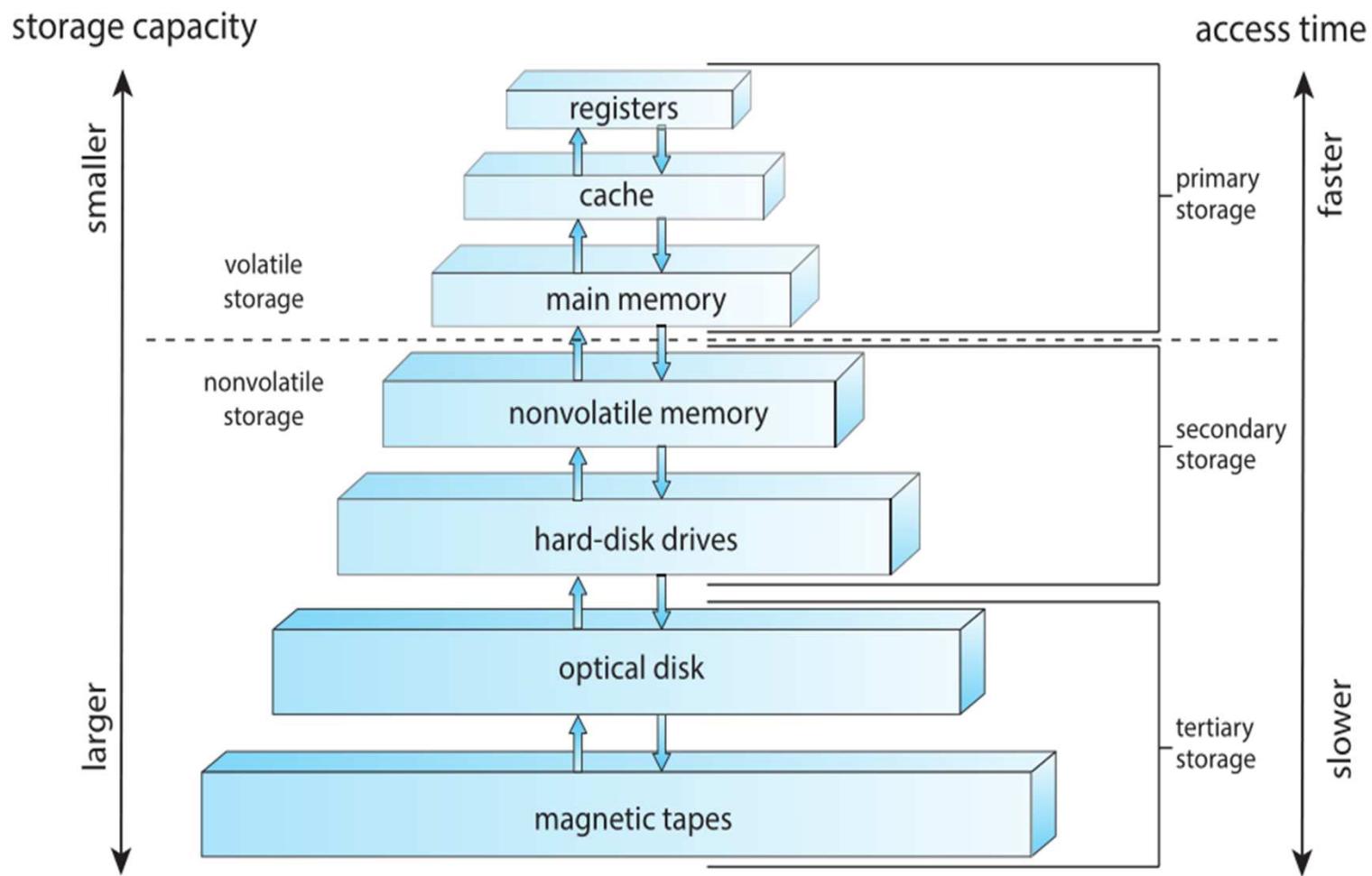
Interrupts

- Bảng event-vector của CPU Intel

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts

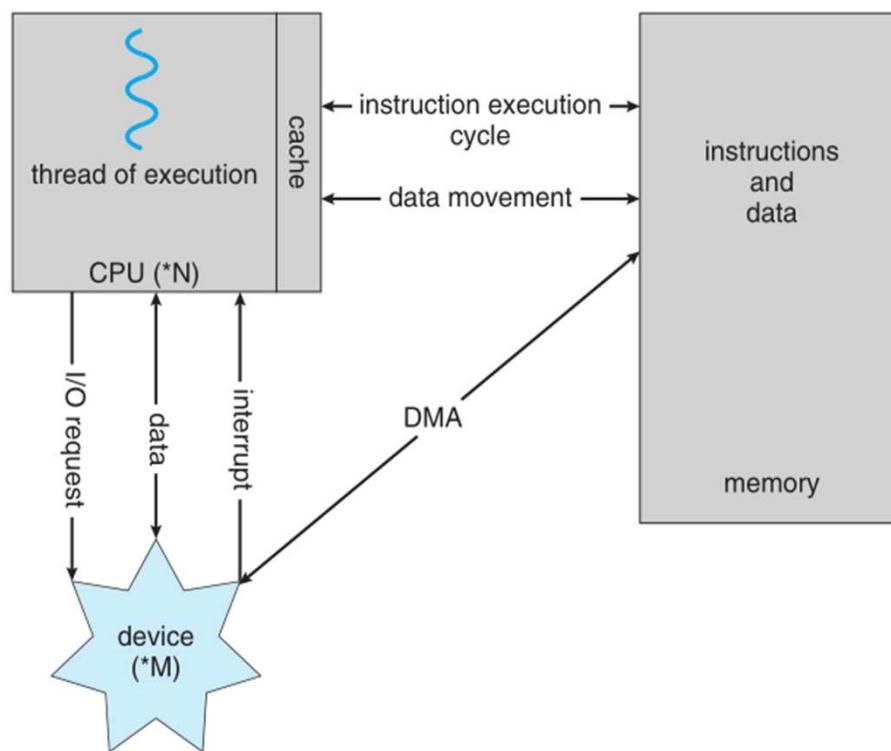
Cấu trúc lưu trữ

- Bảng phân cấp các loại thiết bị lưu trữ



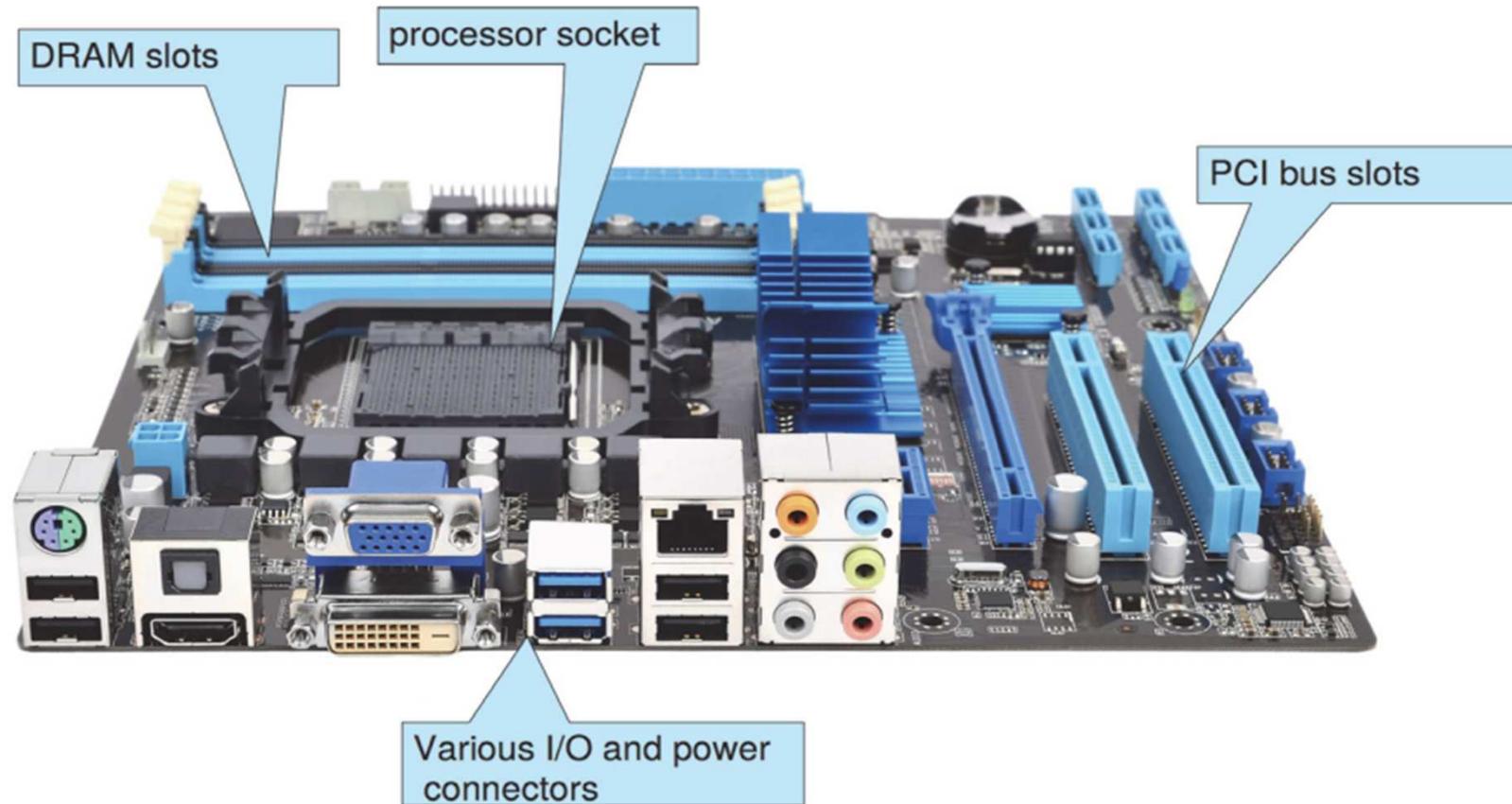
Hoạt động I/O

- Mục đích: thực hiện truy xuất data giữa các thiết bị
- DMA = Direct Memory Access: truy xuất trực tiếp data không cần sự can thiệp của CPU (dùng trong trường hợp cần truy xuất data số lượng lớn, tốn nhiều thời gian)
- VD:



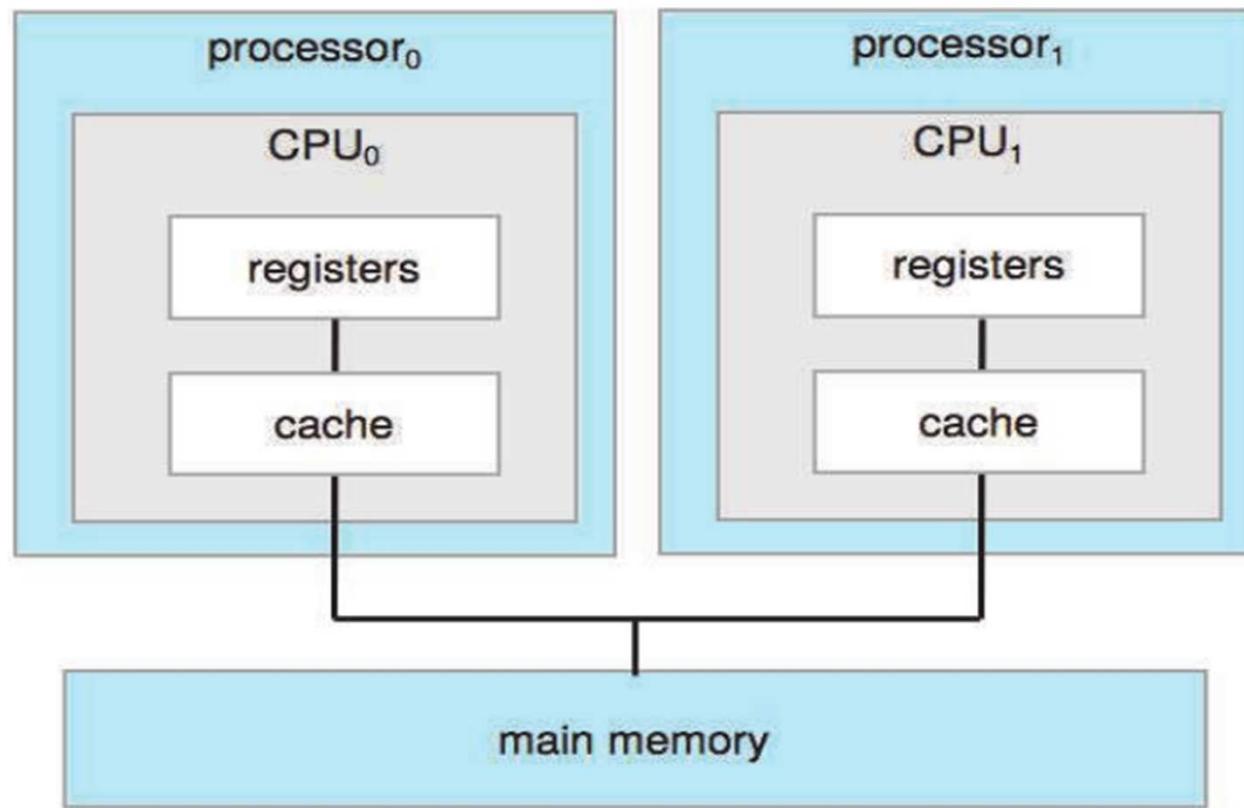
Các loại kiến trúc máy tính

- Một số thuật ngữ cơ bản
 - CPU, Processor, Core, Multicore, Multiprocessor



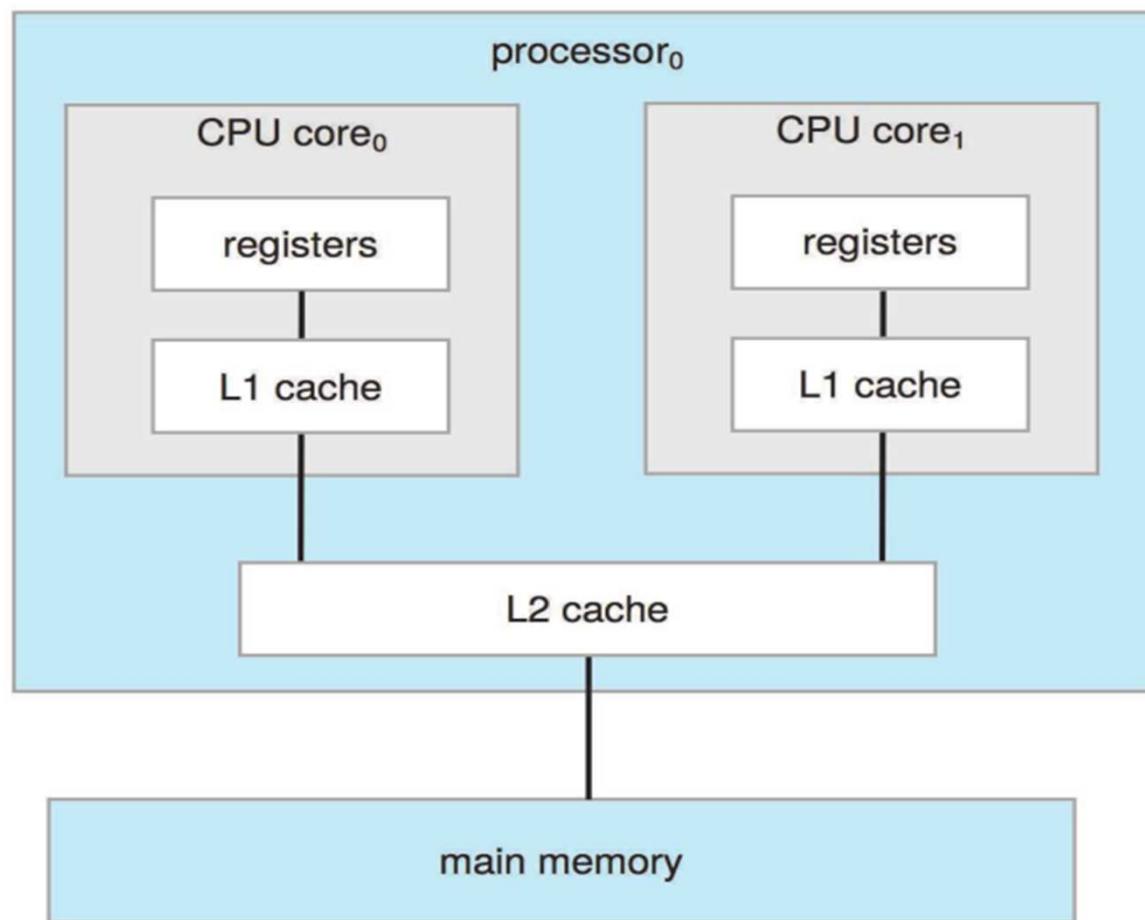
Các loại kiến trúc máy tính

- Single-Processor Systems
- Multiprocessor Systems. Một số ví dụ của kiến trúc này:
 - Symmetric multiprocessing architecture



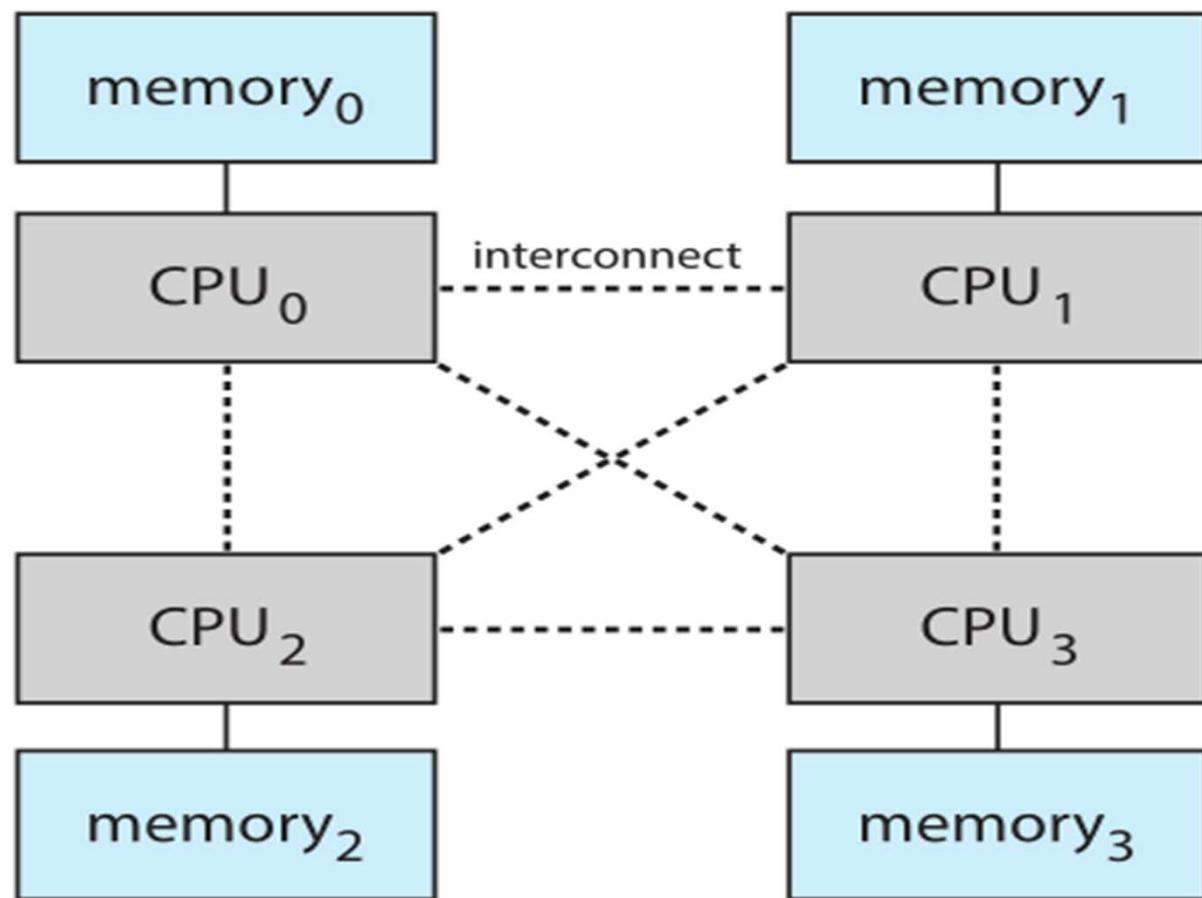
Các loại kiến trúc máy tính

- Multiprocessor Systems. Một số ví dụ của kiến trúc này:
 - Dual-core



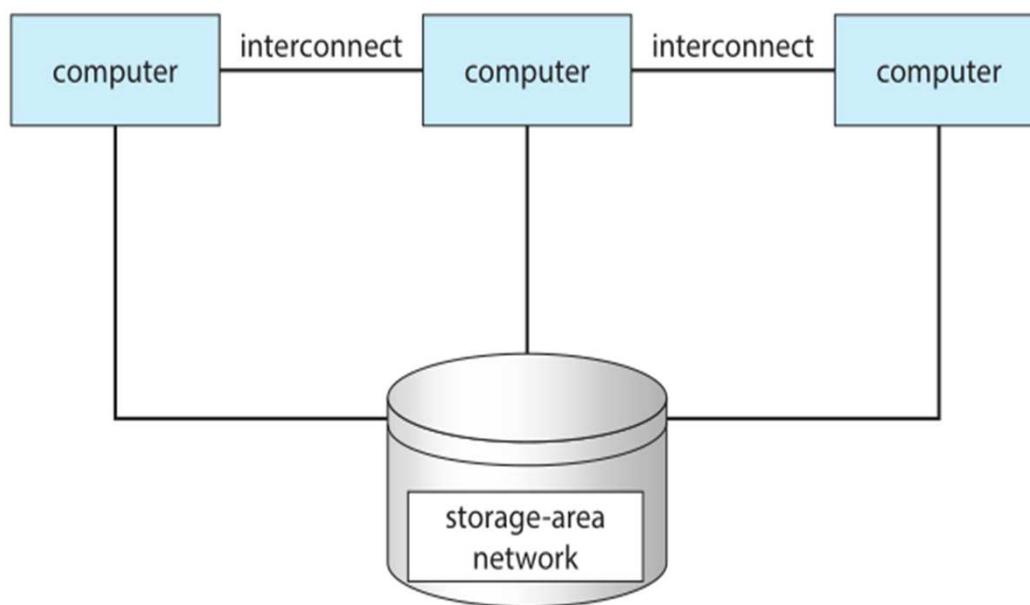
Các loại kiến trúc máy tính

- Multiprocessor Systems. Một số ví dụ của kiến trúc này:
 - NUMA (Non-Uniform Memory Access) multiprocessing architecture



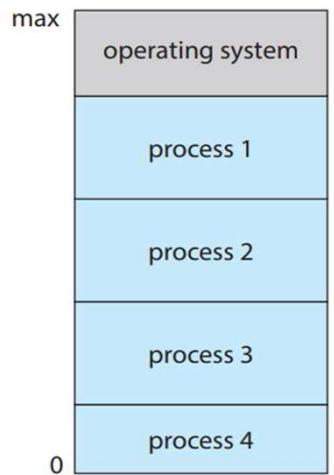
Các loại kiến trúc máy tính

- Multiprocessor Systems. Một số ví dụ của kiến trúc này:
 - Blade servers
 - Mỗi Blade server bao gồm nhiều multiprocessor systems
 - Mỗi multiprocessor system là một hệ thống riêng biệt, chạy OS riêng, nối với nhau bằng network
 - Clustered Systems



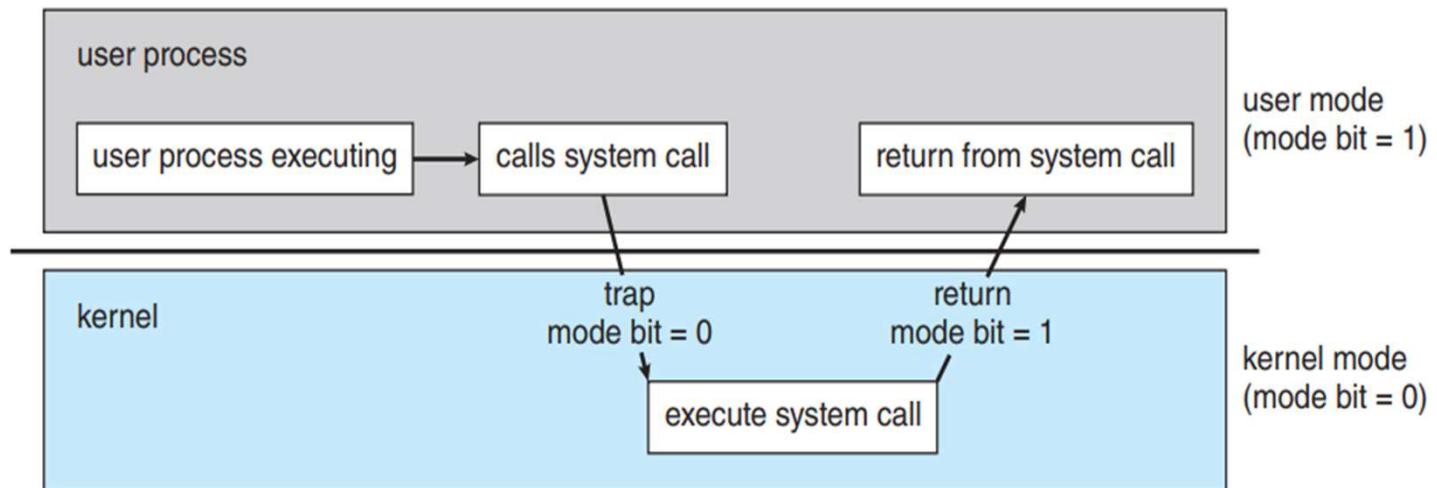
Multiprogramming và Multitasking

- Kỹ thuật giúp HĐH thực hiện nhiều chương trình cùng một lúc, đáp ứng nhu cầu
 - User cần chạy nhiều chương trình cùng lúc
 - Các hardware cần thực hiện nhiều yêu cầu liên lạc, truy xuất lẫn nhau
- Multiprogramming (đa chương):
 - Các program được tổ chức thành các process khi nạp vào bộ nhớ
 - Khi thực hiện I/O, CPU được chuyển cho tiến trình khác
- Multitasking (đa nhiệm):
 - Là kỹ thuật mở rộng của đa chương
 - Mỗi tiến trình được thực hiện trong một khoảng thời gian nhất định và chuyển CPU cho tiến trình khác



Dual-mode và Multimode

- Là sự phân cấp để program của user và HĐH không xâm phạm lẫn nhau
- Dual-mode:



- Multimode:
 - Là sự mở rộng của Dual-mode thành nhiều hơn 2 modes: Kernel mode, Virtual machine mode, User mode

Timer

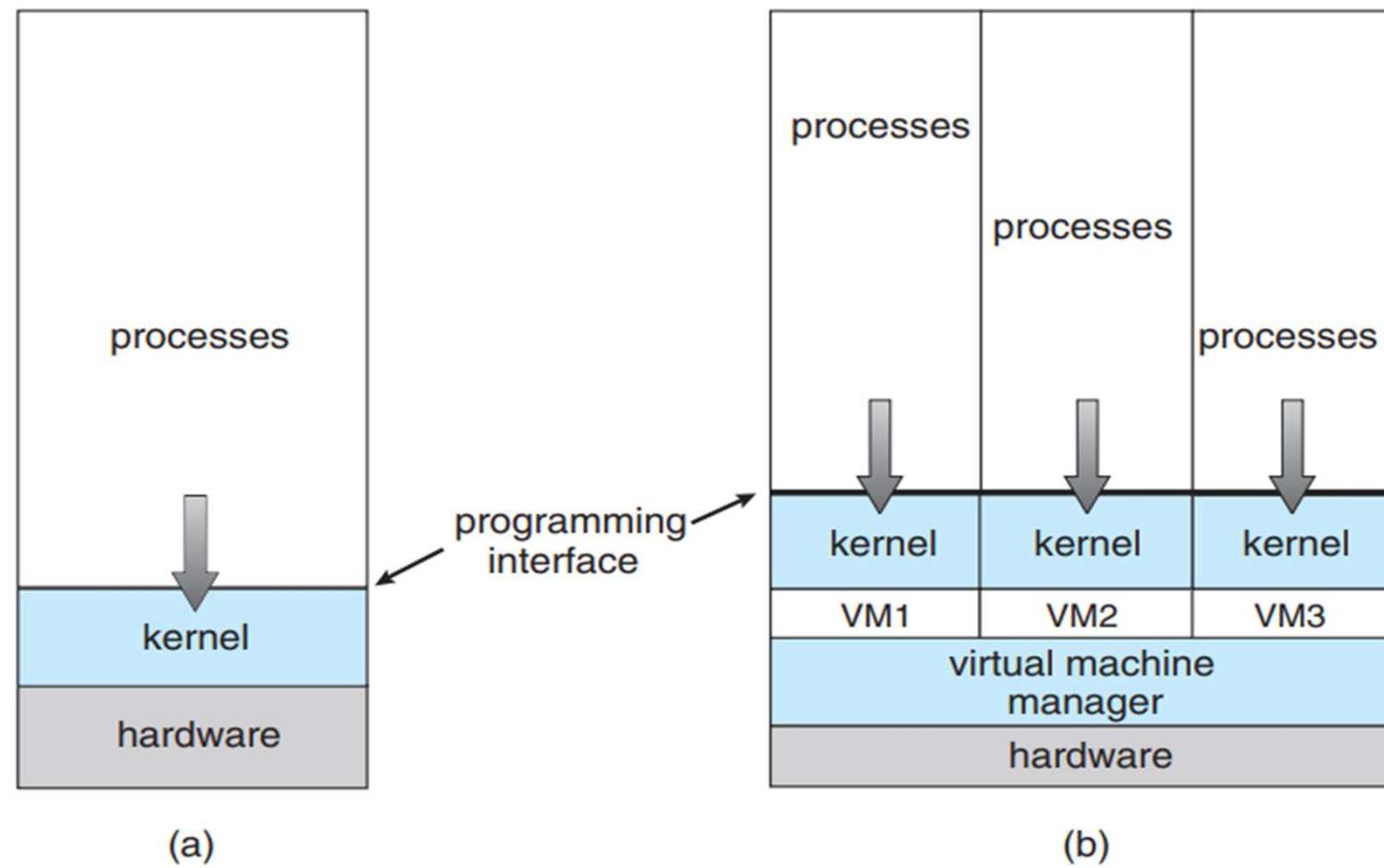
- Là bộ đếm thời gian thực
- Định kỳ phát ra các interrupt đến CPU để:
 - Nhắc HĐH không bỏ sót tiến trình bị chờ đợi vô hạn
 - Chạy chương trình thường trú
 - Chạy các lời gọi hệ thống
 - ...

Các giải pháp quản lý tài nguyên

- Quản lý tiến trình (process management)
 - Quản lý bộ nhớ (memory management)
 - Quản lý hệ thống file (file-system management)
 - Quản lý hệ thống lưu trữ (mass-storage management)
 - Quản lý bộ đệm (cache management)
 - Quản lý hệ thống vào ra (I/O system management)
 - Quản lý an ninh và bảo vệ (security and protection)
- Mỗi loại HĐH cung cấp các thuật toán cụ thể để thực hiện các giải pháp trên sao cho phù hợp với kiến trúc của hệ thống máy tính đang quản lý

Máy ảo (Virtualization)

- Là kỹ thuật dùng một tài nguyên thật để tạo thành nhiều tài nguyên ảo

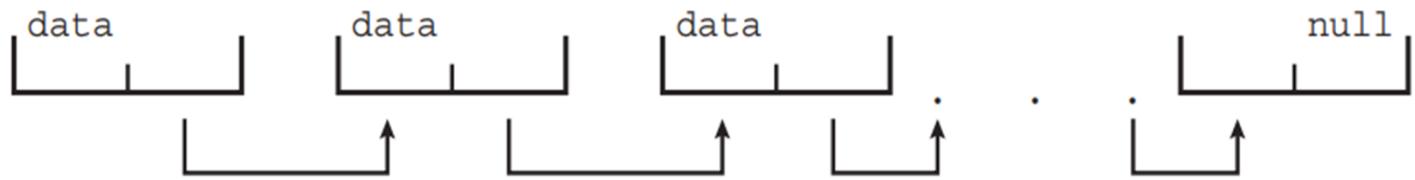


Hệ thống phân tán (Distributed systems)

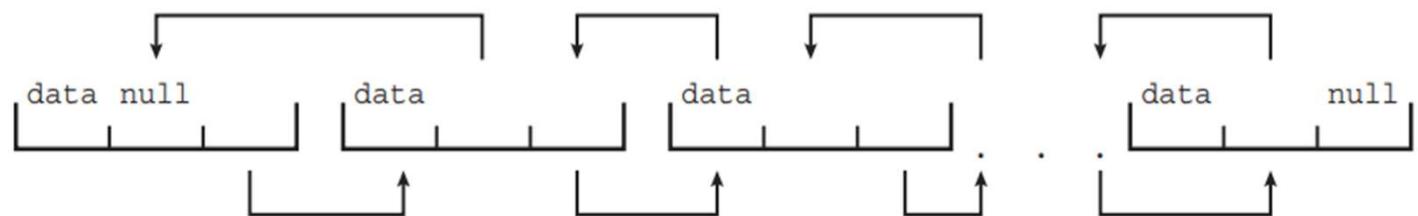
- Là hệ thống kết nối các máy tính có phần cứng khác nhau và HĐH khác nhau lại với nhau
- Cung cấp cho user những lợi ích sau:
 - đa dạng các loại tài nguyên, chức năng, data khác nhau
 - tăng tốc độ xử lý các program của user
 - tăng sự tin cậy trong xử lý và lưu trữ data

Các cấu trúc dữ liệu dùng trong HĐH

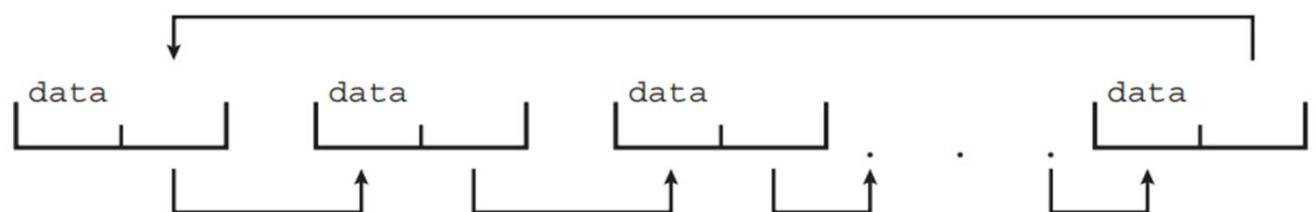
- Array
- List:
 - Kết nối đơn:



- Kết nối đôi:

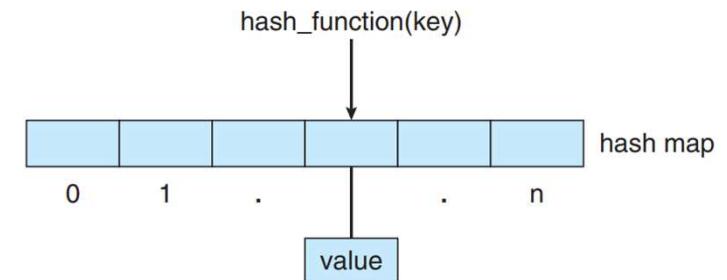
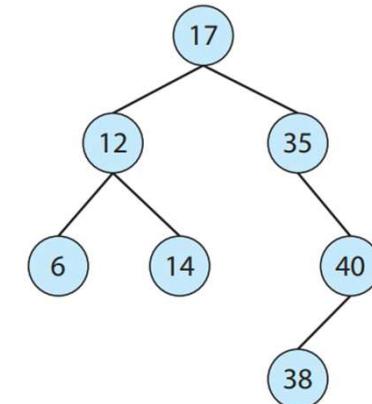


- Kết nối vòng:



Các cấu trúc dữ liệu dùng trong HĐH

- Stack
 - Truy xuất theo nguyên tắc LIFO
 - Thao tác cơ bản: push, pop
- Queue
 - Truy xuất theo nguyên tắc FIFO
- Tree:
 - Cây tổng quát (General tree)
 - Cây nhị phân (Binary search tree)
 - Cây nhị phân cân bằng (Balanced search tree)
 - Hash function
 - Bitmap



Miễn phí vs Mã nguồn mở

- Miễn phí (free)
 - Có thể mã nguồn là mở hay đóng, nhưng chắc chắn là miễn phí
- Mã nguồn mở (open-source)
 - Mã nguồn là mở nhưng chưa chắc là miễn phí
- Chú ý điều này để khi học, nghiên cứu và phát triển về HĐH, có thể chọn sử dụng các HĐH có sẵn một cách đúng đắn mà không vi phạm bản quyền.
- Link tham khảo để hiểu rõ các khái niệm này:
 - <http://www.gnu.org/philosophy/open-source-misses-the-point.en.html>

Các OS không miễn phí và mã nguồn đóng

- Windows của Microsoft là một HĐH đại diện cho các HĐH không miễn phí và mã nguồn đóng
- macOS của Apple thì có kernel (có tên là Darwin) là mã nguồn mở, nhưng các thành phần phát triển trên kernel đó lại là mã nguồn đóng

Red Hat, SUSE , Fedora, Debian, Slackware, Ubuntu

- Giống:
 - Tất cả các HĐH này đều có chung kernel là GNU/Linux
- Khác
 - số lượng người đóng góp phát triển Linux quá nhiều, nên có một số người họp lại tạo thành các nhóm riêng để cùng phát triển Linux và các ứng dụng theo ý của họ
 - mỗi nhóm tự đặt một tên riêng và dùng tên đó để đặt tên cho các distribution của GNU/Linux do họ phát triển
- User khi cần dùng HĐH Linux có thể tùy ý lựa chọn distribution nào có các ứng dụng phù hợp để sử dụng
- Các tiêu chí được lựa chọn là:
 - function, utility, installed applications, hardware support, user interface, and purpose

Sử dụng Linux trong Windows

- Cài đặt phần mềm máy ảo như Virtualbox, VMWare, ...
- Tạo nhiều máy ảo trong các phần mềm đó
- Dùng mỗi máy ảo để
 - nghiên cứu
 - sửa code
 - phát triển thêm ứng dụng, compile, cài đặt thí nghiệm một loại HĐH nào đó

BSD UNIX

- Được phát triển vào năm 1978 bởi AT&T.
- Sau đó, được tiếp tục phát triển và phát hành bởi University of California at Berkeley (UCB) ở dạng không miễn phí
- Đến năm 1994 thì được phát hành ở dạng mã nguồn mở và cũng được chia sẻ để nhiều người đóng góp vào sự phát triển của BSD UNIX
- Tương tự như sự phát triển của GNU/Linux, các nhà phát triển cũng nhóm lại với nhau để tạo ra nhiều distributions của BSD UNIX và được đặt các tên khác nhau.
- Phát triển mạnh nhất và được sử dụng nhiều nhất là các distribution sau:
 - Free BSD , Net BSD , Open BSD , Dragonfly BSD

Code miễn phí của OS Free BSD

- Link cho xem online toàn bộ code của OS Free BSD:
<https://svnweb.freebsd.org>
- Repository cho phép download source code của BSD UNIX
<https://subversion.apache.org/source-code>
- Repository này cung cấp chức năng
 - “pull” : để user download source code
 - “push” : để user upload các code mới phát triển thêm

Code của Darwin (kernel của mac OS)

- Darwin cũng được phát triển từ BSD UNIX
- Link download code Darwin

<http://www.opensource.apple.com/>

- Các tài liệu hướng dẫn và công cụ hỗ trợ cho các nhà phát triển được Apple cung cấp ở link sau:

<http://developer.apple.com>

Solaris

- Solaris là OS thương mại Sun Microsystems. Solaris cũng được phát triển từ UNIX kernel
- Đến năm 2005 thì Sun Microsystems cung cấp dự án OpenSolaris để chia sẻ một phần code của Solaris ở dạng mã nguồn mở
- Hiện nay, có dự án Illumos tiếp tục phát triển Solaris dựa trên dự án OpenSolaris
- Thông tin của OS Solaris được cung cấp ở link sau:
<http://wiki.illumos.org>

Portal của nhiều HĐH và ứng dụng mã nguồn mở và miễn phí

- <http://freshmeat.net/>
- <http://distrowatch.com/>
- Tài liệu và source codes của các projects trong các portals này cho phép mọi người theo dõi được mức độ phát triển của các loại HĐH và các loại ứng dụng
- Giúp nghiên cứu và đóng góp thêm vào sự phát triển của các HĐH và ứng dụng