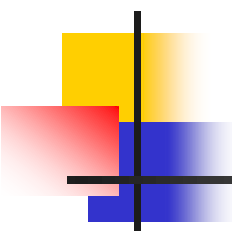


Hệ thống Xuất/Nhập (i/o)

cuu duong than cong . com



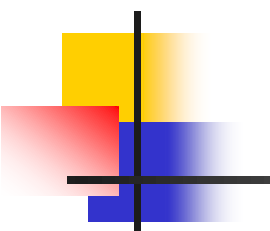


Nội dung

- Thiết bị phần cứng I/O
- Giao diện I/O cho ứng dụng
- Các dịch vụ của I/O subsystem
- Hiệu suất I/O

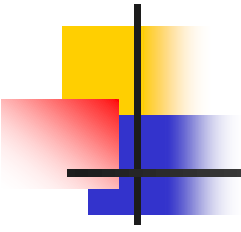
cuu duong than cong . com





Tổng quan

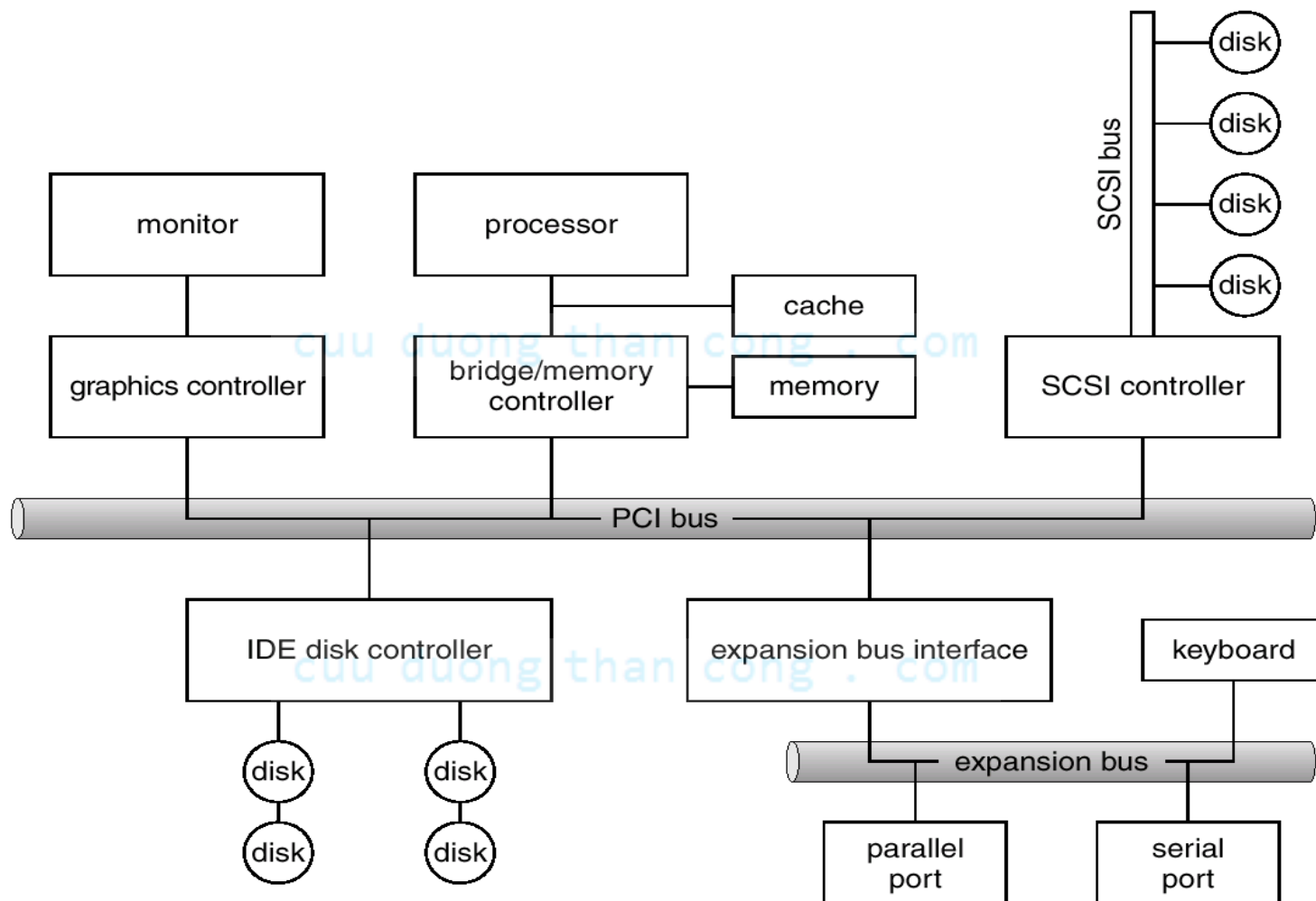
- Các thiết bị I/O khác biệt về chức năng, tốc độ,... nên cần có các phương thức quản lý, điều khiển tương ứng khác nhau
 - Chức năng của I/O subsystem của kernel
- Các trình điều khiển thiết bị (device driver) cung cấp cho I/O subsystem một giao diện thuần nhất để truy cập các thiết bị



Thiết bị phần cứng I/O

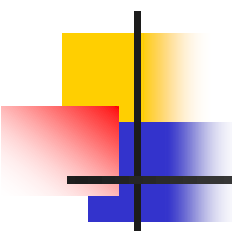
- Kết nối giữa hệ thống máy tính và các thiết bị I/O
 - Port (connection point)
 - Bus (daisy chain, shared direct access)
- Điều khiển
 - Controller (device controller, SCSI host adapter)
- Giao tiếp giữa CPU và thiết bị I/O
 - **I/O port**: dùng lệnh I/O để tác động lên các thanh ghi dữ liệu / trạng thái / lệnh của controller.
 - **Memory-mapped I/O**

Cấu trúc bus trong PC



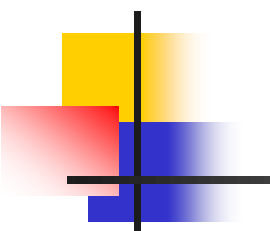
Một số I/O port trong PC

I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary) controller
320-32F	hard-disk controller
378-37F	parallel port controller
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary) controller



Các kỹ thuật thực hiện I/O

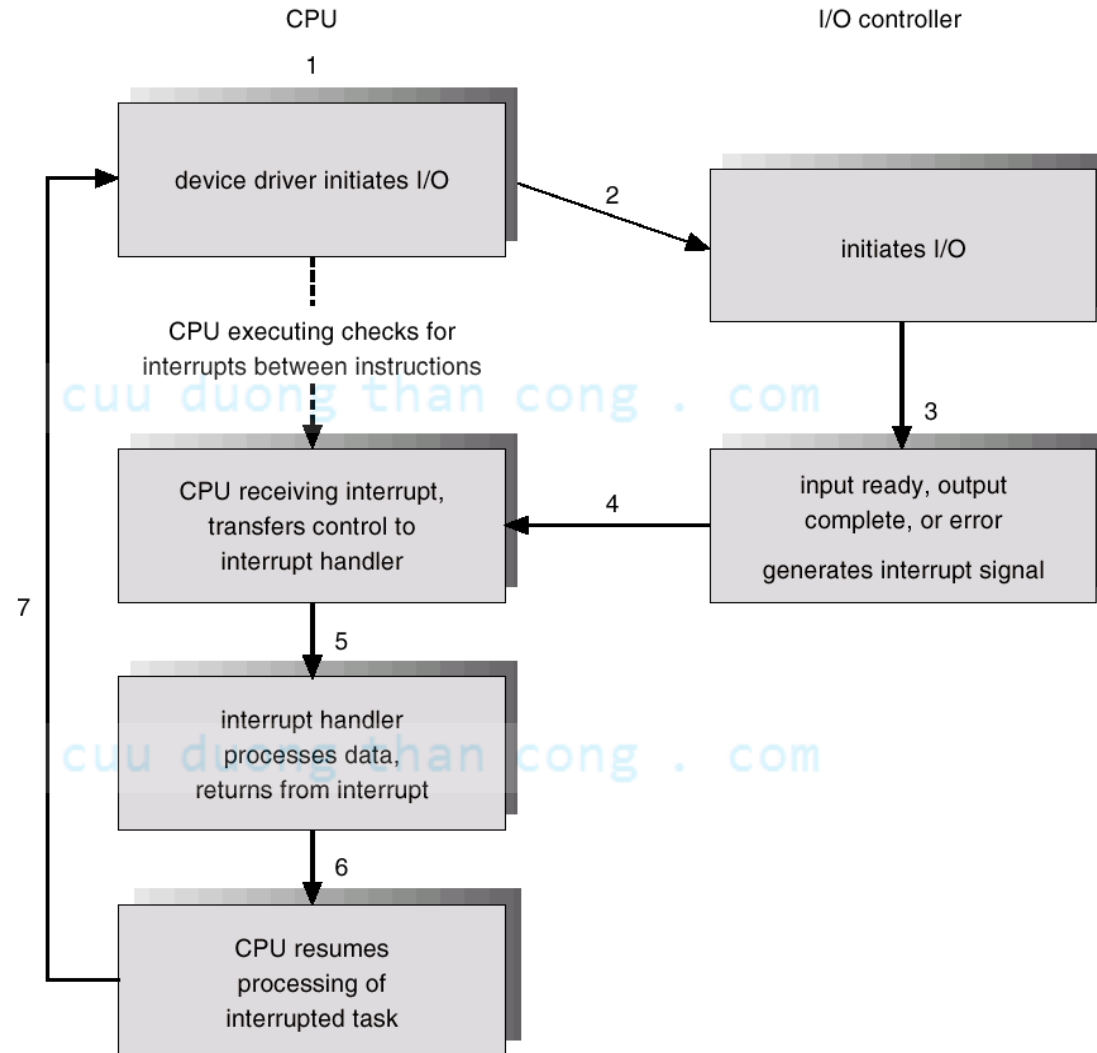
- Kỹ thuật **polling** (busy waiting) hay **programmed I/O**
 - Kiểm tra trạng thái của thiết bị khi muốn thực hiện I/O
 - ready hoặc busy hoặc error
 - Tiêu tốn thời gian trong vòng lặp để kiểm tra trạng thái (busy-wait) và thực hiện I/O.



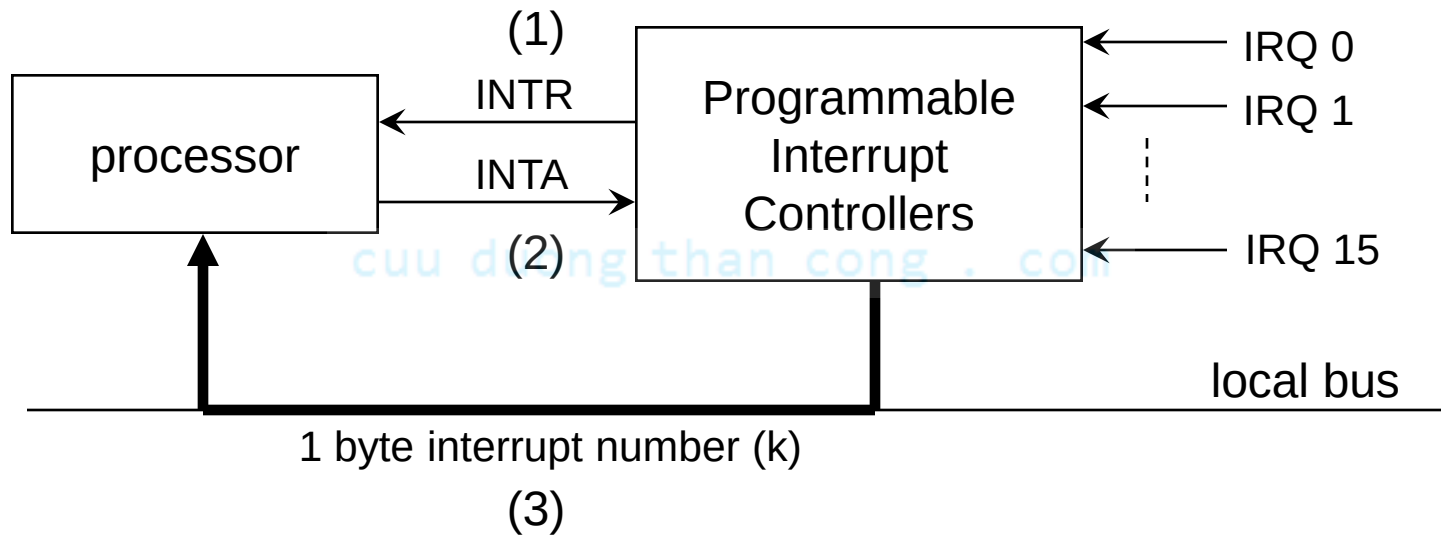
Các kỹ thuật thực hiện I/O (tt.)

- Kỹ thuật I/O dùng **ngắt quãng** (interrupt-driven I/O)
 - CPU có một ngõ Interrupt Request (IR), được kích hoạt bởi thiết bị I/O
 - Nếu ngắt xảy ra (IR = active), CPU chuyển quyền điều khiển cho trình phục vụ ngắt (interrupt handler)
 - Các ngắt có thể che được (maskable) hoặc không che được (non-maskable)
 - Hệ thống có một bảng vector ngắt chứa địa chỉ các trình phục vụ ngắt
- Ngắt cũng có thể dùng xử lý các sự kiện khác trong hệ thống (lỗi chia cho 0, lỗi vi phạm vùng nhớ,...)

Chu trình I/O với ngắt quãng



PC interrupt



(4) Take exception to address $k*4$

Bảng vector Pentium processor

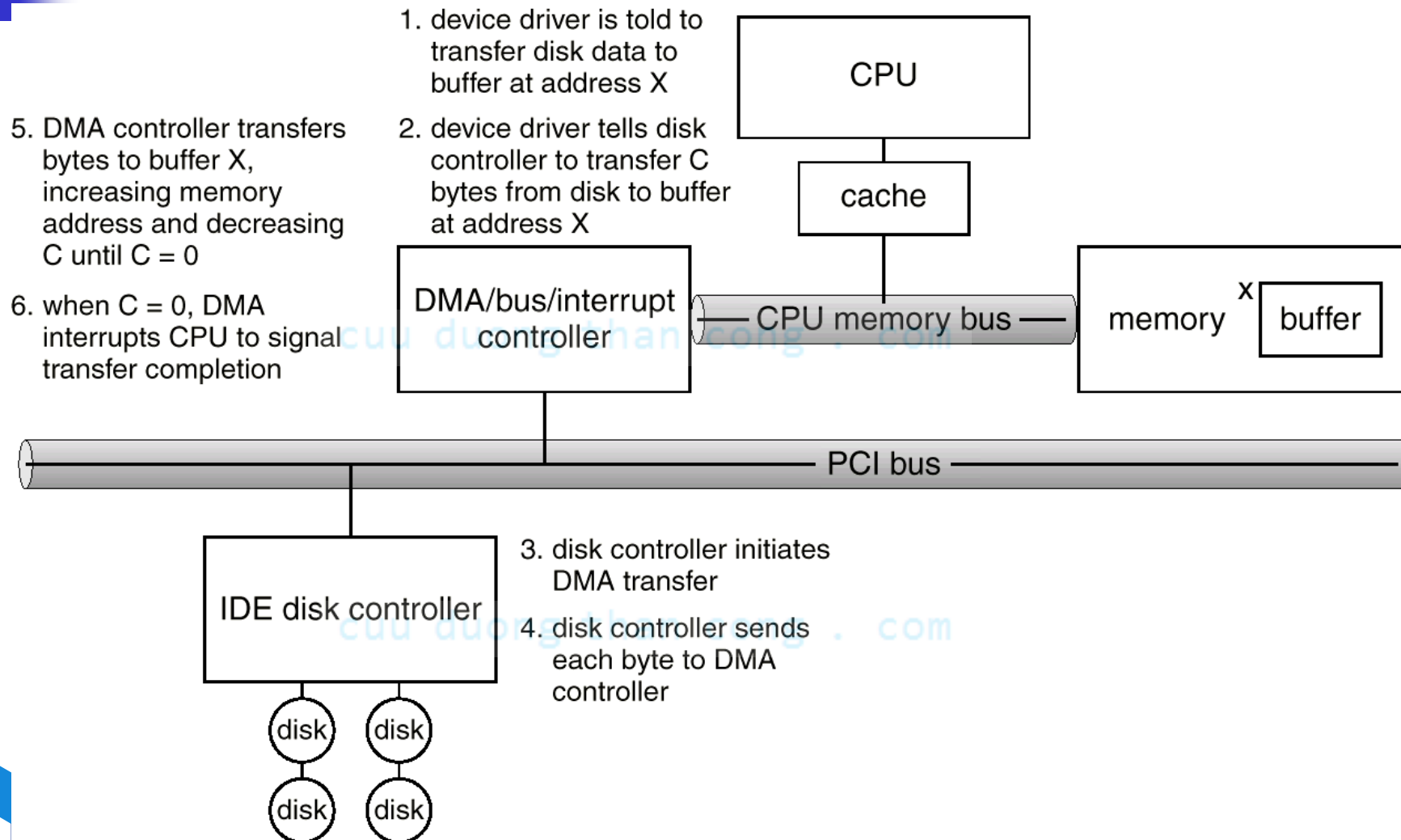
vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
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

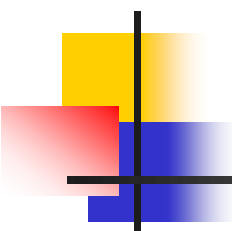


Các kỹ thuật thực hiện I/O (tt.)

- Kỹ thuật I/O dùng **direct memory access (DMA)**
 - Các kỹ thuật polling và interrupt-driven I/O không thích hợp khi thực hiện di chuyển khối lượng lớn dữ liệu.
 - Kỹ thuật DMA cần có phần cứng hỗ trợ đặc biệt, đó là DMA controller
 - Kỹ thuật DMA thực hiện truyền dữ liệu trực tiếp giữa thiết bị I/O và bộ nhớ mà không cần sự can thiệp của CPU.

6 bước thực hiện DMA

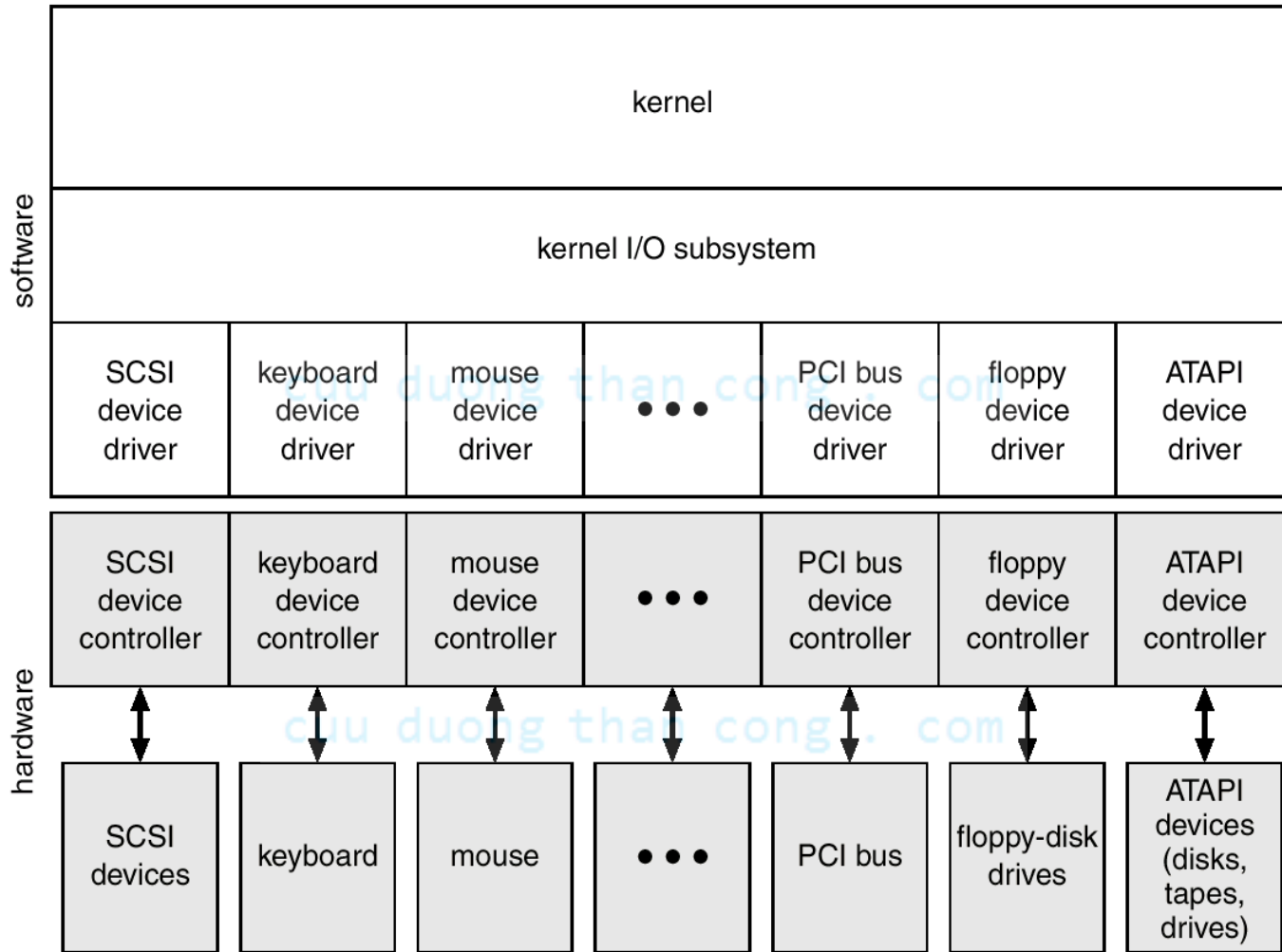




Giao diện I/O cho ứng dụng

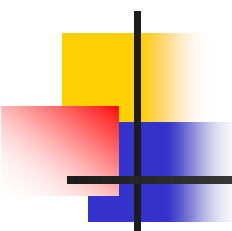
- OS cung cấp một giao diện I/O chuẩn hóa, thuần nhất cho các ứng dụng.
 - Ví dụ: một ứng dụng in tài liệu ra máy in mà không cần biết hiệu máy in, đặc tính máy in,...
- Giao diện làm việc là các I/O **system call** của OS.
- Trình điều khiển thiết bị là “cầu nối” giữa kernel và các bộ điều khiển thiết bị (device controller).

Cấu trúc I/O cấp kernel



Đặc tính của các thiết bị I/O

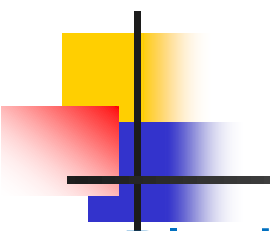
aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read&write	CD-ROM graphics controller disk



Các dịch vụ I/O

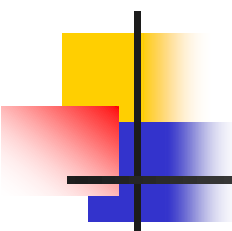
- Giao diện chuẩn cho nhóm thiết bị có liên quan
 - **Thiết bị khối** (block device)
 - Disk
 - Các tác vụ: read, write, seek
 - **Thiết bị kí tự** (character device)
 - Keyboard, mouse, serial port, line printer,...
 - Tác vụ: get, put
 - **Thiết bị mạng** (network device)
 - Block hoặc character.
 - Socket interface trên Unix, Windows/NT,...
- Clock và timer
 - Cung cấp thời gian hiện tại, timer
 - Có thể lập trình được.





System call yêu cầu I/O

- **Blocking**: process bị suspended cho đến khi I/O hoàn tất.
 - Dễ dàng sử dụng
 - Không hiệu quả trong một số trường hợp
 - **Nonblocking**: process sẽ tiếp tục thực thi ngay sau lệnh gọi I/O.
 - Ví dụ: data copy (buffered I/O)
 - Thường hiện thực với multithreading
 - Khó kiểm soát kết quả thực hiện I/O
 - **Asynchronous**: process vẫn thực thi trong lúc hệ thống đang thực hiện I/O.
 - Khó sử dụng
- I/O subsystem báo hiệu cho process khi I/O hoàn tất



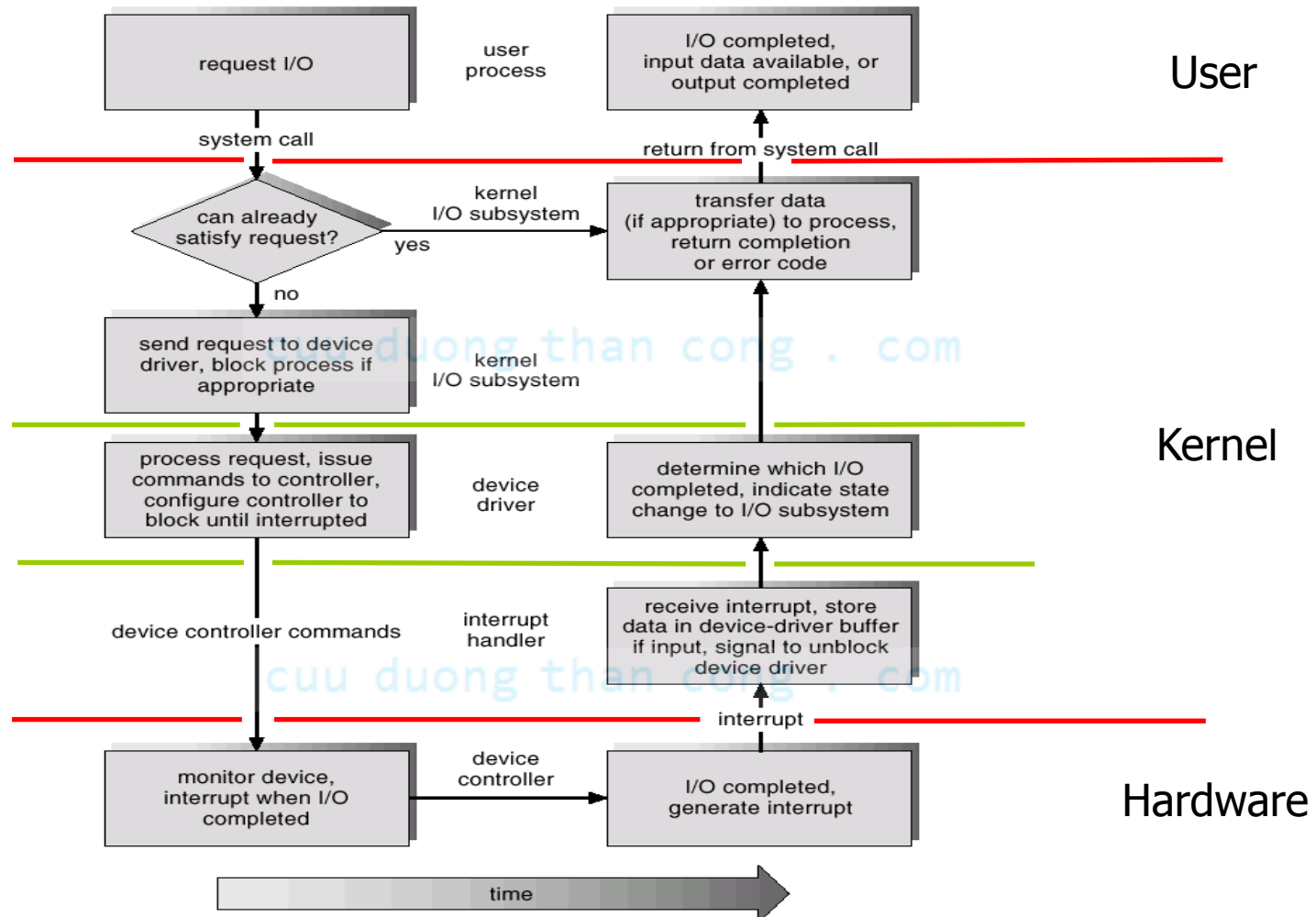
I/O subsystem của kernel

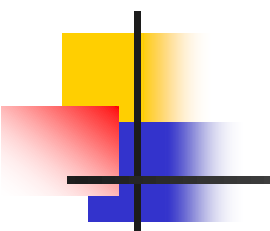
I/O subsystem cung cấp các chức năng

- Định thời các yêu cầu I/O
 - Các yêu cầu I/O xếp hàng tại các hàng đợi của mỗi thiết bị
 - Bảo đảm công bằng, hiệu suất cao.
- Đệm dữ liệu (buffering): lưu dữ liệu tạm thời trong bộ nhớ khi thực hiện I/O
 - Giải quyết trường hợp chênh lệch tốc độ, kích thước dữ liệu khi thực hiện I/O
- Caching
- Spooling
- Xử lý lỗi (error handling)
 - OS sẽ phục hồi lại sau những lỗi đọc đĩa, thiết bị không tồn tại, lỗi đọc/ghi hay vận chuyển dữ liệu
 - Trả về mã lỗi khi yêu cầu truy cập I/O không thành công.
 - Lỗi hệ thống được lưu lại dưới dạng báo cáo trong ghi nhớ lỗi.



Chu trình của một yêu cầu I/O

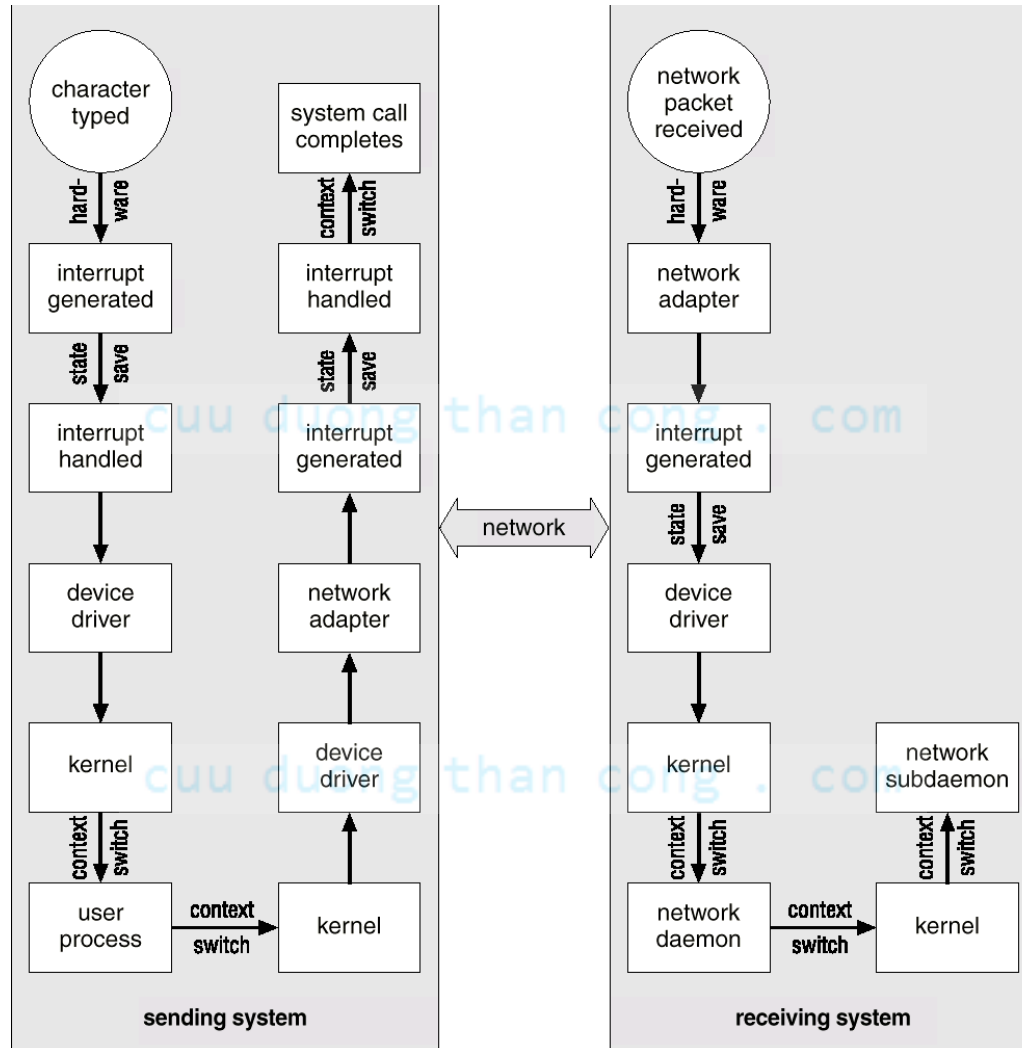




Hiệu suất I/O

- Hiệu suất I/O ảnh hưởng rất lớn đến hiệu suất toàn hệ thống. I/O gây tải cao do
 - CPU thực hiện các lệnh điều khiển của device driver, của kernel I/O code.
 - Chuyển ngữ cảnh vì các I/O interrupt, chi phí copy dữ liệu gửi nhận.
 - Network traffic gây tốc độ context switch cao.
- Tăng hiệu suất I/O
 - Giảm số lần chuyển ngữ cảnh, giảm thiểu quá trình copy dữ liệu (bằng caching,...)
 - Giảm số lần ngắt quãng (truyền khối dữ liệu lớn, dùng các bộ controller thông minh, dùng kỹ thuật polling,...)
 - Sử dụng DMA nếu có thể.

Ví dụ: remote login



I/O system call
to send the character
to the remote machine

subdaemon for the
remote login session