

# HỆ ĐIỀU HÀNH (OPERATING SYSTEM)

Trình bày: Nguyễn Hoàng Việt  
Khoa Công Nghệ Thông Tin  
Đại Học Cần Thơ

# Chương 11: Hệ thống xuất/nhập (I/O System)

---

- Phần cứng I/O (I/O Hardware)
- Giao diện ứng dụng I/O (Application I/O Interface)
- Hệ thống con I/O trong nhân (Kernel I/O Subsystem)
- Chuyển đổi yêu cầu I/O sang các tác vụ phần cứng (Transforming I/O Request to Hardware Operations)
- Kênh truyền thông (Stream)
- Hiệu năng (Performance)

# Phần cứng xuất/nhập (1)

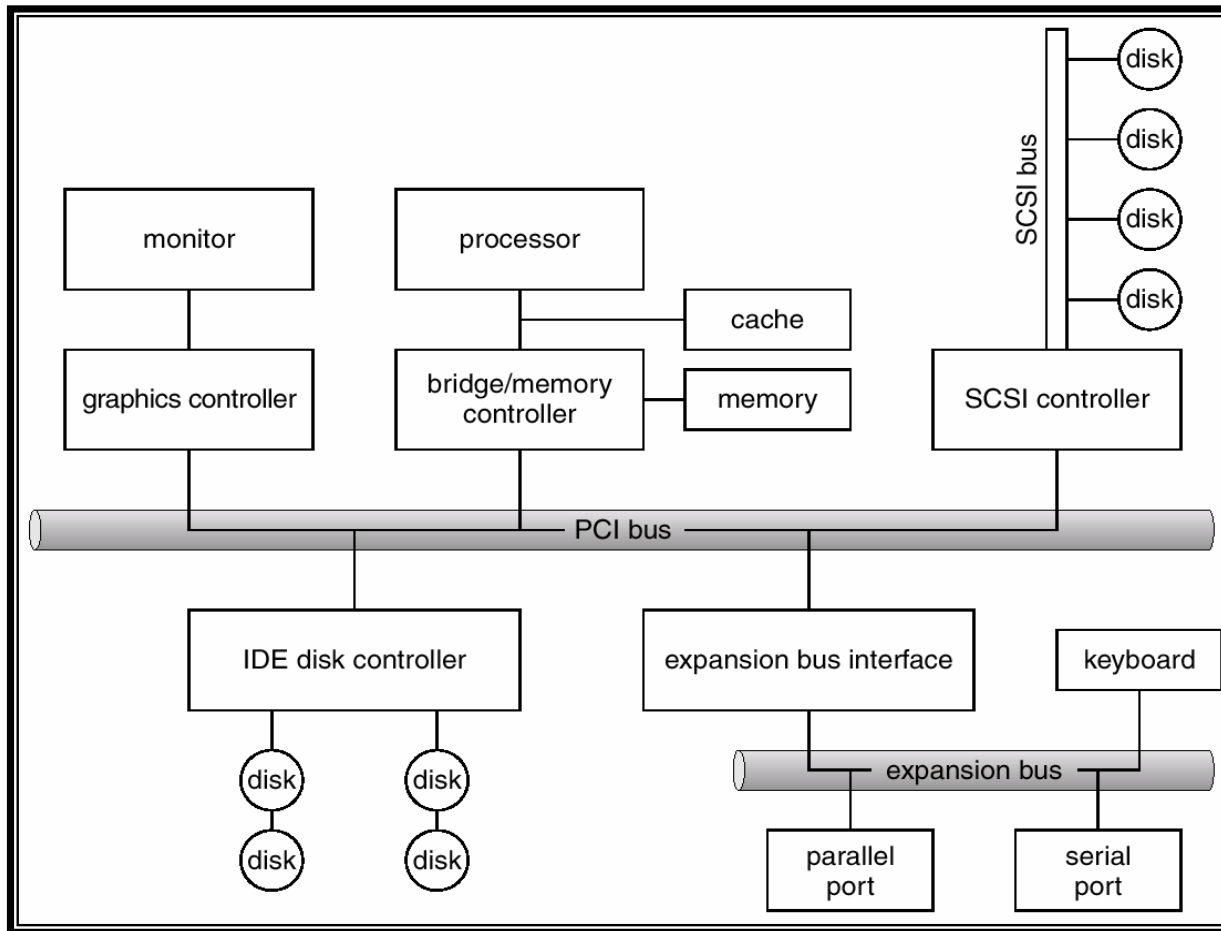
## Tổng quan

---

- Các thiết bị I/O có nhiều vô kể!
- Các khái niệm chung
  - Cổng (Port): một điểm nối kết với thiết bị
  - Bus: kênh vận chuyển thông tin, dữ liệu
  - Controller: bộ điều khiển thiết bị
- Các chỉ thị I/O (instruction) sẽ điều khiển các thiết bị
- Các thiết bị có địa chỉ riêng, được sử dụng bởi
  - Các chỉ thị I/O trực tiếp
  - I/O được ánh xạ qua bộ nhớ

# Phần cứng xuất/nhập (2)

## Cấu trúc bus truyền thống trên máy PC



# Phần cứng xuất/nhập (3)

## Một số cổng I/O của các thiết bị

---

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)
320-32F	hard-disk controller
378-37F	parallel port
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary)

# Phần cứng xuất/nhập (4)

## Các thanh ghi cổng I/O của các thiết bị

---

- Thanh ghi trạng thái (Status Register)
  - Đọc để xác định các trạng thái: một lệnh đã hoàn thành, dữ liệu sẵn dùng, lỗi thiết bị
- Thanh ghi điều khiển (Control Register)
  - Viết để khởi động một lệnh hoặc thay đổi chế độ của thiết bị
- Thanh ghi xuất dữ liệu (Data-In Register)
  - Đọc để nhận dữ liệu vào
- Thanh ghi dữ liệu xuất (Data-out Register)
  - Ghi để gửi dữ liệu xuất

# Phần cứng xuất/nhập (5)

## Thăm dò (Polling)

---

- Giao thức (protocol) cho giao tác giữa máy tính và bộ điều khiển thiết bị (controller).
- Thăm dò xác định trạng thái của thiết bị.
  - command-ready
  - busy
  - error
- Chu trình chờ đợi bận (Busy-waiting or polling loop)
- Lặp việc đọc thanh ghi trạng thái cho đến khi bit bận (busy bit) bị xóa.

# Phần cứng xuất/nhập (6)

## Các ngắt (Interrupt)

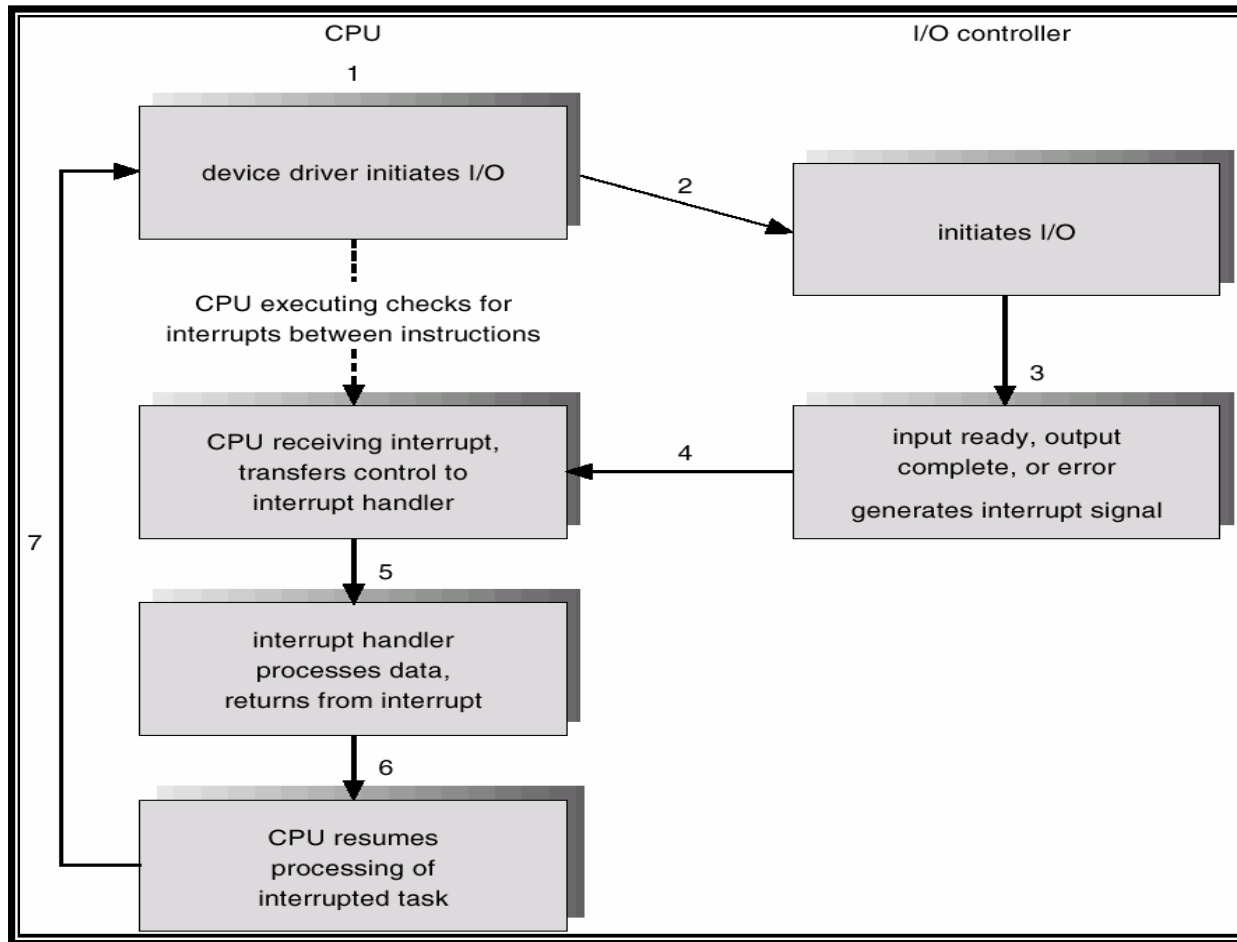
---

- Tuyến yêu cầu ngắt của CPU (CPU interrupt request line) được kích hoạt bởi thiết bị I/O
  - **Nonmaskable interrupt:**
    - ✓ Là các ngắt không bị bỏ qua bởi kỹ thuật che ngắt chuẩn
    - ✓ Thường được dùng để báo hiệu về các sự kiện quan trọng như một lỗi phần cứng không thể phục hồi
  - **Maskable interrupt:**
    - ✓ Các ngắt có thể bị bỏ qua hoặc trì hoãn
    - ✓ Thường được dùng cho các sự kiện không quan trọng
- Cơ chế ngắt:
  - Bộ quản lý ngắt tiếp nhận các ngắt
  - Vector ngắt dùng để phân bổ ngắt đến đúng bộ quản lý ngắt
  - Dựa trên sự ưu tiên
- Cơ chế ngắt cũng được sử dụng cho những tình huống ngoại lệ (chia cho 0), page fault, system calls, timers, ...



# Phần cứng xuất/nhập (7)

## Chu kỳ I/O có phát sinh ngắt



# Phần cứng xuất/nhập (8)

## Bảng vector sự kiện trong bộ xử lý Intel Pentium

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
19Đ31	(Intel reserved, do not use)
32Đ255	maskable interrupts

# Phần cứng xuất/nhập (9)

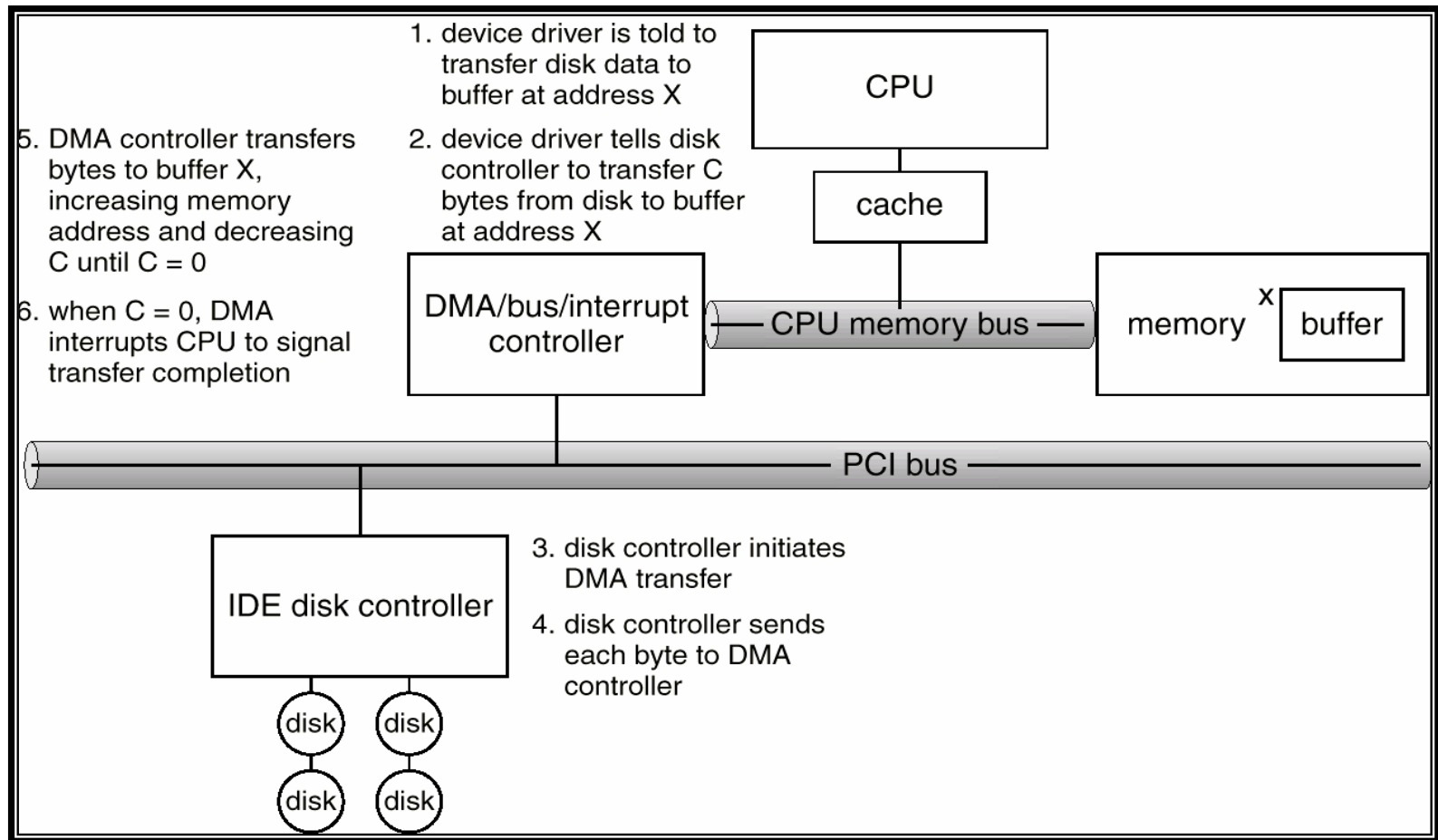
## Truy xuất trực tiếp bộ nhớ (DMA-Direct Memory Access)

---

- I/O được lập trình (Programmed I/O – PIO)
  - Kiểm tra bit trạng thái và đặt dữ liệu vào thanh ghi bộ điều khiển từng byte một.
  - PIO không hữu dụng khi chuyển một lượng dữ liệu lớn.
- Bộ điều khiển DMA (DMA controller)
  - Được sử dụng chuyển một lượng dữ liệu lớn.
  - Bỏ qua CPU để chuyển dữ liệu trực tiếp từ thiết bị I/O đến bộ nhớ.
  - Cho phép CPU có thể được dùng cho việc khác

# Phần cứng xuất/nhập (10)

## 6 bước để chuyển dữ liệu kiểu DMA



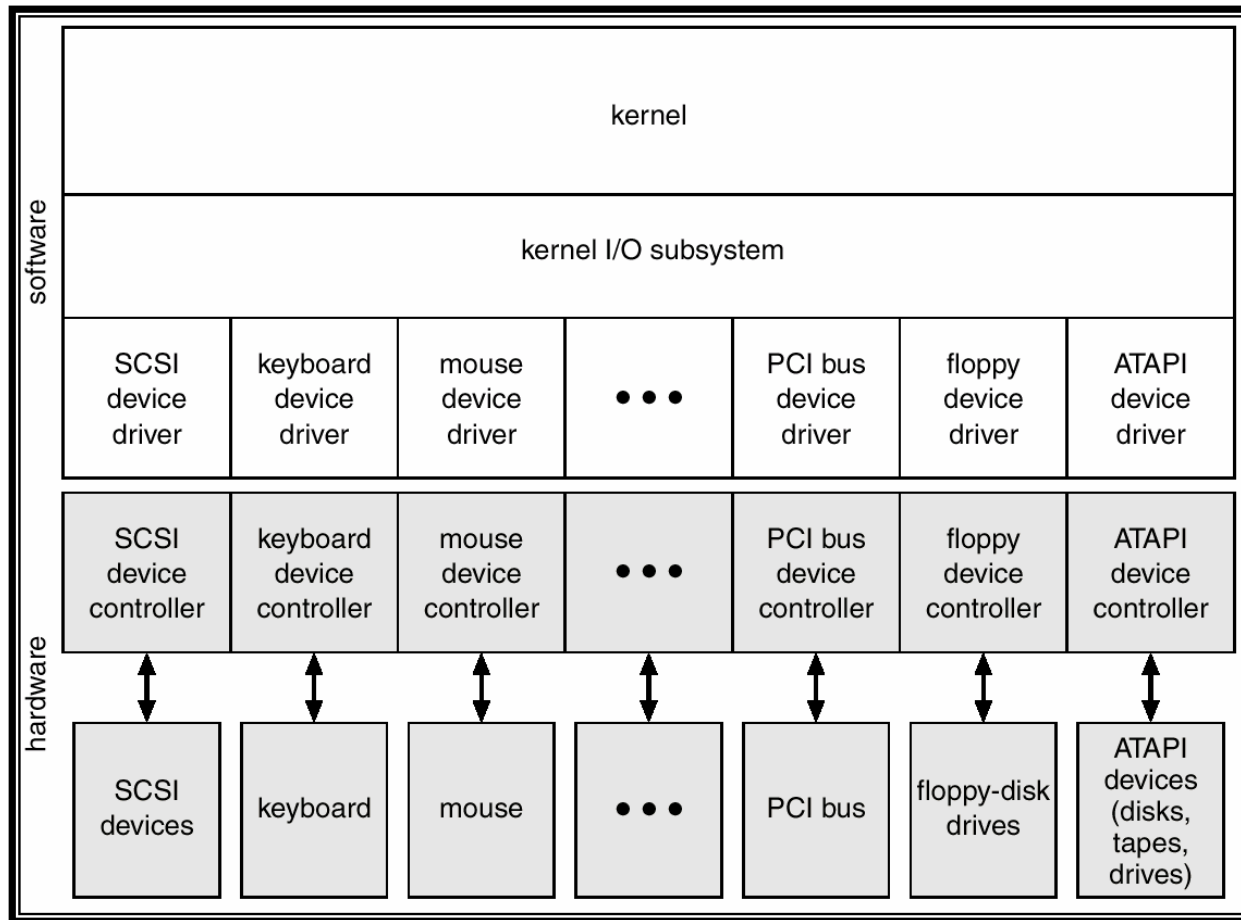
# Giao diện ứng dụng I/O (1)

---

- Các lời gọi hệ thống I/O sẽ thu gọn các hành xử của thiết bị theo một số ít dạng tổng quát, che đi các khác biệt của phần cứng.
- Lớp trình điều khiển thiết bị (Device-driver layer) sẽ che dấu sự khác biệt giữa những bộ điều khiển thiết bị I/O (I/O controllers) ra khỏi nhân (kernel) của hệ điều hành
- Các thiết bị trong hệ thống khác nhau theo nhiều hướng
  - Luồng ký tự (Character-stream) hay là khối (block)
  - Truy xuất tuần tự hay ngẫu nhiên (Sequential or random-access)
  - Đồng bộ (synchronous) hay không đồng bộ (asynchronous)
  - Chia sẻ hay tận hiến (Sharable or dedicated)
  - Tốc độ hoạt động (Speed of operation)
  - read-write, read only, hay write only

# Giao diện ứng dụng I/O (2)

## Một kiến trúc nhân I/O (A Kernel I/O Structure)



# Giao diện ứng dụng I/O (3)

## Các tính chất của 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

# Giao diện ứng dụng I/O (4)

## Các thiết bị dạng khối và dạng ký tự

---

- Các thiết bị dạng khối (block) bao gồm các ổ đĩa
  - Chuyển từng khối dữ liệu
  - Các lệnh bao gồm: read, write, seek
  - Vào ra thô hoặc truy xuất thông qua hệ thống tập tin
  - Có thể sử dụng kỹ thuật Memory-mapped file
- Các thiết bị dạng ký tự (character) bao gồm bàn phím, chuột, các cổng tuần tự (serial ports)
  - Chuyển từng byte dữ liệu
  - Các lệnh bao gồm **get**, **put**
  - Các thư viện hàm được đặt trên cùng cho phép tương tác dạng dòng lệnh (truy xuất và sửa đổi 1 dòng (line) mỗi lần)



# Giao diện ứng dụng I/O (5)

## Các thiết bị mạng (Network Device)

---

- Quá nhiều biến thể, từ dạng khối đến dạng ký tự. Do đó mỗi thiết bị mạng cần có giao diện riêng
- Unix và Windows NT/9x/2000 có thêm vào giao diện socket
  - Tách biệt giữa giao thức mạng và việc vận hành mạng
- Có một dải rộng các hướng tiếp cận để sử dụng thiết bị mạng (pipes, FIFOs, streams, message queues, mailboxes)

# Giao diện ứng dụng I/O (6)

## Đồng hồ và bộ định thời (Clock & Timer)

---

- Phần cứng clock và timer thường cung cấp 3 chức năng cơ bản:
  - Thời gian hiện tại
  - Lượng thời gian đã trôi qua
  - Thiết đặt timer để kích hoạt một thao tác tại thời điểm xác định.
- Hai chức năng sau được cung cấp bởi Bộ định thời có thể lập trình (programmable interval timer)
  - Thường được dùng để đếm thời gian, tạo chu kỳ các ngắt (ví dụ cho CPU scheduling)
- Hàm ioctl (trong UNIX) được sử dụng để làm nhiều thao tác hệ thống I/O như thao tác lên đồng hồ và bộ định thời

# Giao diện ứng dụng I/O (7)

## I/O nghẽn và không nghẽn

---

- **Lời gọi I/O nghẽn (I/O blocking call):**
  - Quá trình phát ra I/O nghẽn sẽ treo cho đến khi thao tác I/O hoàn thành.
  - Dễ hiểu và dễ sử dụng.
  - Không thỏa mãn một số yêu cầu, ví dụ giao diện người dùng nhận từ bàn phím và chuột cùng với xử lý và hiển thị lên màn hình.
- **Lời gọi I/O Không nghẽn (I/O nonblocking call):**
  - I/O không nghẽn không ngừng thực thi ứng dụng một khoảng thời gian.
  - Thay vào đó, nó trở về (return) nhanh với giá trị bằng số lượng byte đã được đọc/ghi.
  - Cài đặt dùng kỹ thuật đa luồng (multi-threading): một số luồng thực hiện lời gọi nghẽn, trong khi các luồng khác tiếp tục thực thi.
- **Lời gọi I/O dị bộ (I/O asynchronous call):**
  - Lời gọi này trở về ngay lập tức, không chờ đợi thao tác I/O hoàn thành ⇒ quá trình vẫn tiếp tục chạy trong khi thao tác I/O đang thực thi.
  - Hệ thống con I/O sẽ báo hiệu cho quá trình khi thao tác I/O hoàn thành.
  - Khó sử dụng.

# Hệ thống con I/O trong nhân (1)

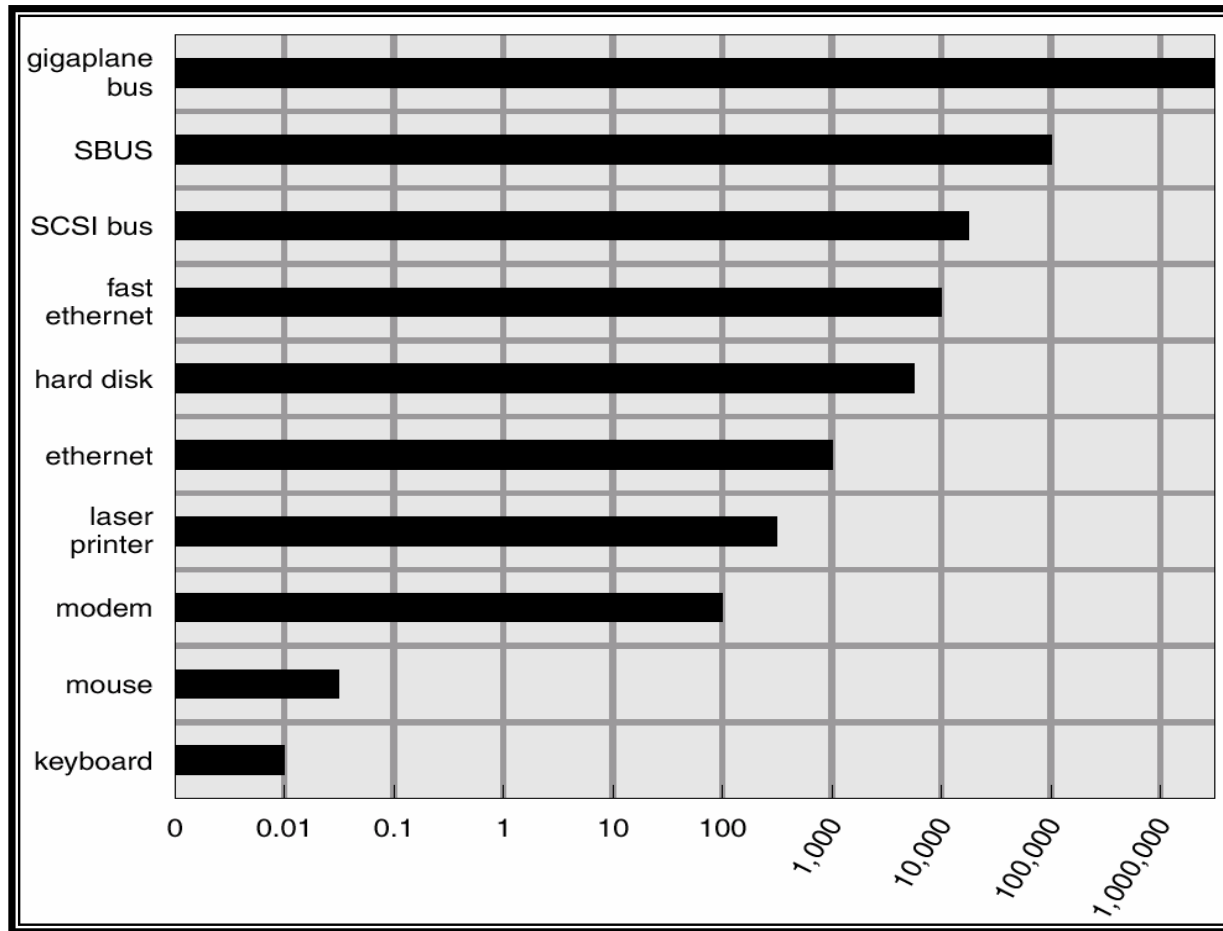
## (Kernel I/O Subsystem)

---

- Cung cấp nhiều dịch vụ liên quan đến I/O
  - Xây dựng trên nền phần cứng và chương trình điều khiển thiết bị (device driver)
- Định thời (Scheduling)
  - Một số yêu cầu I/O được sắp thứ tự trong hàng đợi của từng thiết bị
  - Một số hệ điều hành thích sử dụng sự công bằng
- Trữ đệm (Buffering): lưu dữ liệu trong bộ nhớ trong khi chuyển dữ liệu giữa các thiết bị
  - Để đối phó với sự bất tương xứng về tốc độ thiết bị
  - Để đối phó với sự bất tương xứng về kích thước đơn vị truyền nhận của thiết bị
  - Để duy trì việc sao chép có ngữ nghĩa (copy semantics)

# Hệ thống con I/O trong nhân (2)

## Tốc độ truyền dữ liệu trong Sun Enterprise 6000



# Hệ thống con I/O trong nhân (3)

---

- Trữ tạm (caching): bộ nhớ truy xuất nhanh, dùng để lưu bản sao của dữ liệu
  - Bản sao trên bộ nhớ truy xuất nhanh sẽ được đọc nhanh hơn.
  - Đây là chìa khóa để tăng hiệu năng.
- Spooling: là bộ đệm lưu giữ tạm dữ liệu xuất đến thiết bị
  - Nếu thiết bị có thể phục vụ chỉ một yêu cầu tại một thời điểm
  - Ví dụ: In ấn - dữ liệu xuất của mỗi ứng dụng được đưa vào (được spooled vào 1 file trên đĩa.
- Giữ chỗ thiết bị (Device Reservation) - cung cấp truy xuất mang tính chất độc quyền (exclusive) đến thiết bị
  - Các lời gọi hệ thống để cấp phát và thu hồi.
  - Canh chừng deadlock.

# Hệ thống con I/O trong nhân (4)

## Kiểm soát lỗi (Error Handling)

---

- Hệ điều hành có thể khôi phục lại sau khi bị sự cố vào ra như: đọc đĩa thất bại, thiết bị không sẵn dùng, hành động ghi bị thất bại âm thầm...
- Hầu hết các yêu cầu I/O bị thất bại đều trả về một mã lỗi
- Nhật ký lỗi của hệ thống (system error logs) sẽ giữ lại tất cả các báo cáo về sự cố

# Hệ thống con I/O trong nhân (5)

## Cấu trúc dữ liệu I/O trong nhân

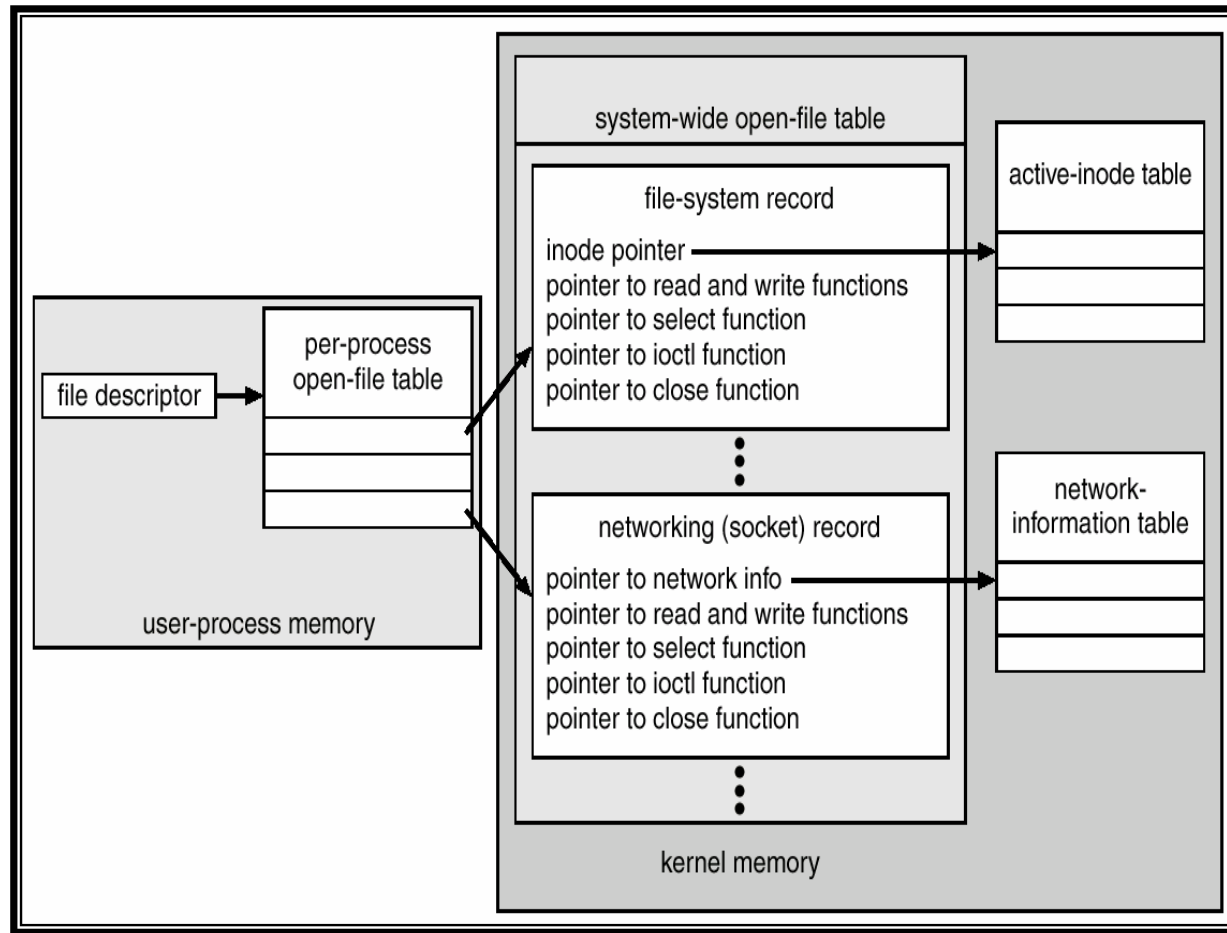
---

- Nhân giữ thông tin về trạng thái của các bộ phận I/O, bao gồm bảng quản lý các file đang mở, các kết nối mạng, giao tiếp thiết bị dạng ký tự, ...
- Rất nhiều cấu trúc dữ liệu phức tạp được dùng để theo dõi:
  - Quản lý không gian tên cho tập tin và thiết bị
  - Điều khiển truy xuất đến tập tin và thiết bị.
  - Điều khiển các thao tác
  - Cấp phát không gian hệ thống tập tin
  - Cấp phát thiết bị
  - Buffering , caching, spooling
  - Định thời xuất nhập
  - Theo dõi trạng thái thiết bị, quản lý lỗi, phục hồi sau lỗi
  - Cài đặt và cấu hình trình điều khiển thiết bị (device driver)
- Các phương pháp được dùng để cài đặt hệ thống I/O:
  - Hướng đối tượng (object-oriented): Unix
  - Phương pháp chuyển thông (message passing): Windows NT



# Hệ thống con I/O trong nhân (6)

## Cấu trúc dữ liệu I/O trong nhân của UNIX



# Hệ thống con I/O trong nhân (7)

## Từ yêu cầu I/O đến thao tác phần cứng

---

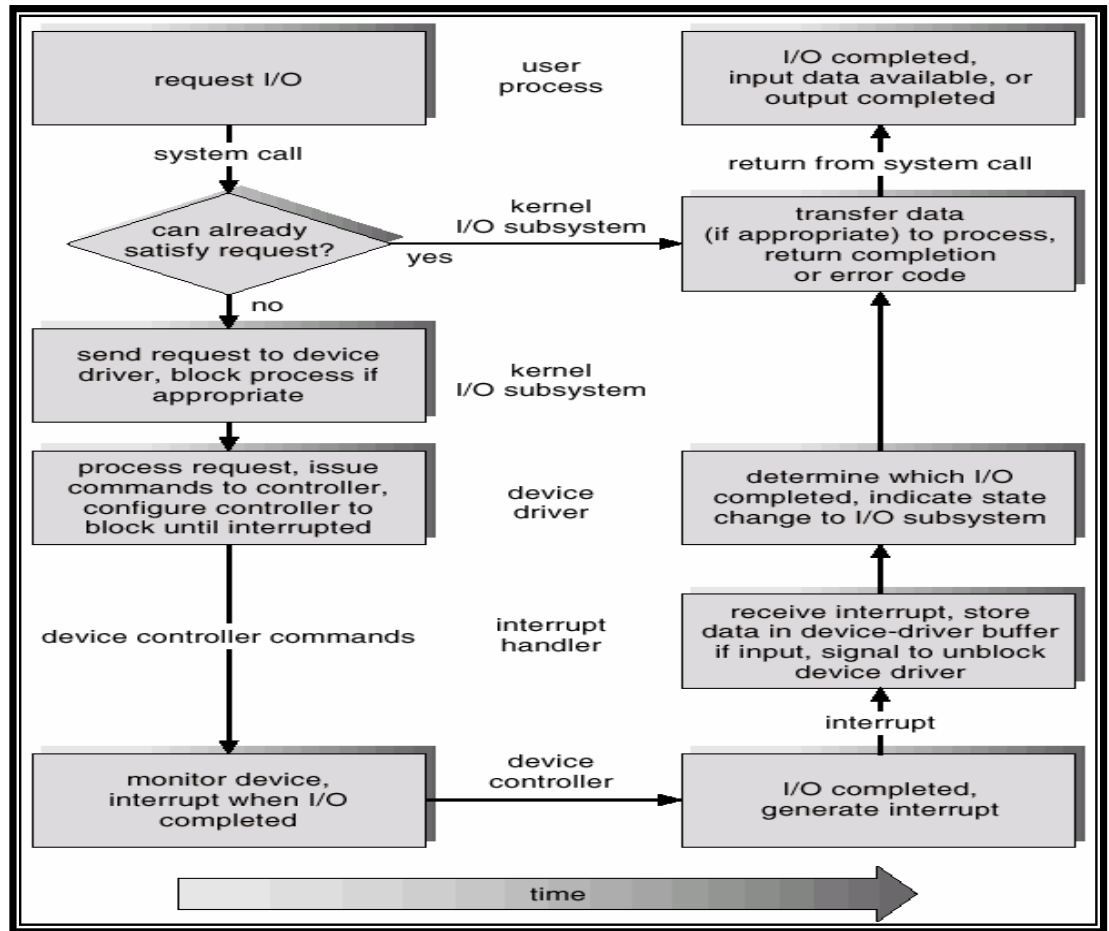
- Xét việc đọc một file từ đĩa vào bộ nhớ của một quá trình:
  - Xác định thiết bị đang lưu file
  - Dịch tên file sang cách thức biểu diễn file của thiết bị
  - Đọc dữ liệu một cách vật lý từ đĩa đến vùng đệm
  - Làm cho dữ liệu sẵn dùng đối với quá trình vừa yêu cầu
  - Trả quyền điều khiển về cho quá trình

# Hệ thống con I/O trong nhân (8)

## Từ yêu cầu I/O đến thao tác phần cứng

Xét việc đọc một file từ đĩa vào bộ nhớ của một quá trình:

- Xác định thiết bị đang lưu file
- Dịch tên file sang cách thức biểu diễn file của thiết bị
- Đọc dữ liệu một cách vật lý từ đĩa đến vùng đệm
- Làm cho dữ liệu sẵn dùng đối với quá trình vừa yêu cầu
- Trả quyền điều khiển về cho quá trình



Vòng đời của một yêu cầu I/O

# Kênh truyền thông (1)

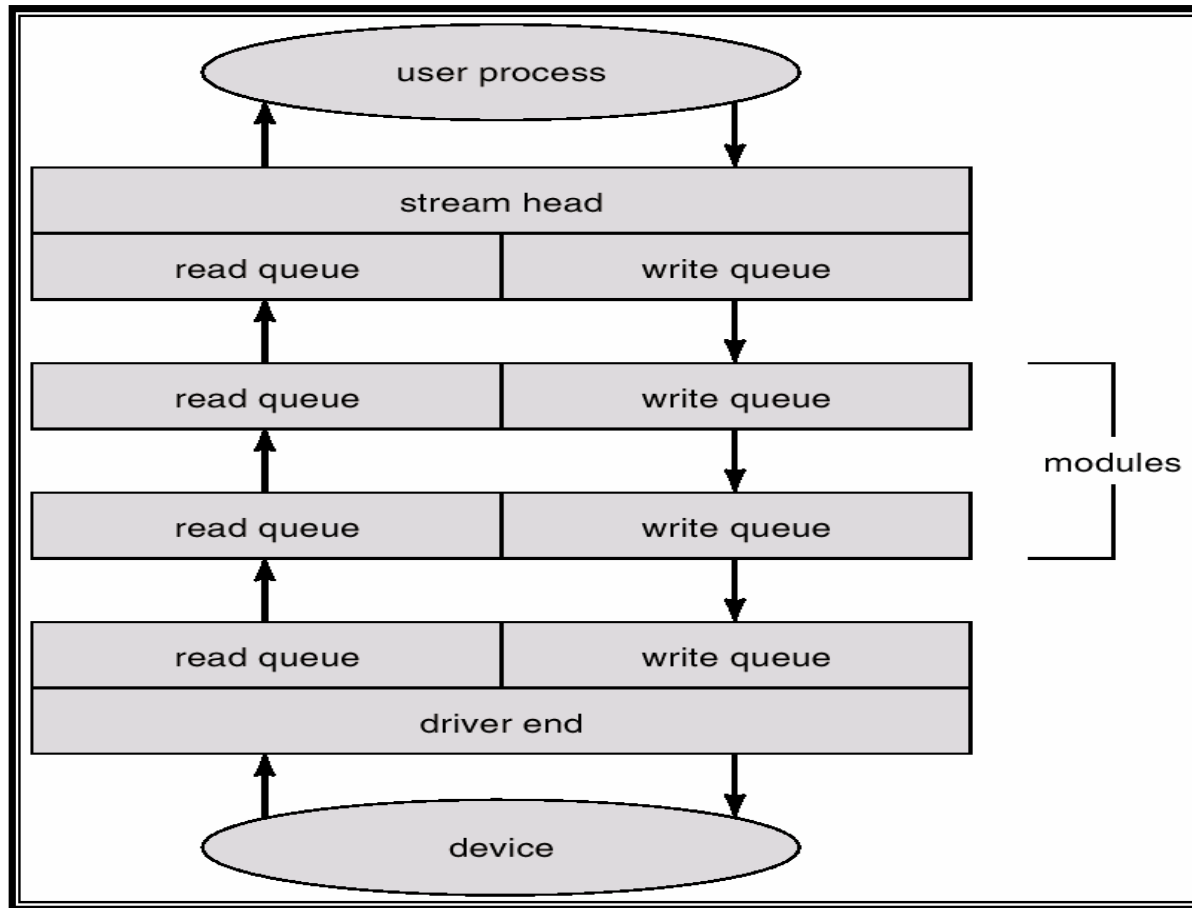
## (STREAM)

---

- **STREAM** – một kênh truyền thông hai chiều đồng thời (full-duplex communication channel) giữa một quá trình người dùng và một thiết bị.
- Một STREAM bao gồm:
  - Đầu STREAM (**STREAM head**) giao diện với quá trình người dùng
  - Đuôi driver (**driver end**) giao diện với thiết bị
  - Không có hoặc có các module STREAM giữa hai đầu này.
- **Stream head, driver end** và mỗi module chứa một cặp **read queue** và một **write queue**
- Phương pháp chuyển thông điệp được sử dụng để giao tiếp giữa các hàng đợi (queues)

# Kênh truyền thông (2)

## Cấu trúc của STREAM



# Hiệu năng (1)

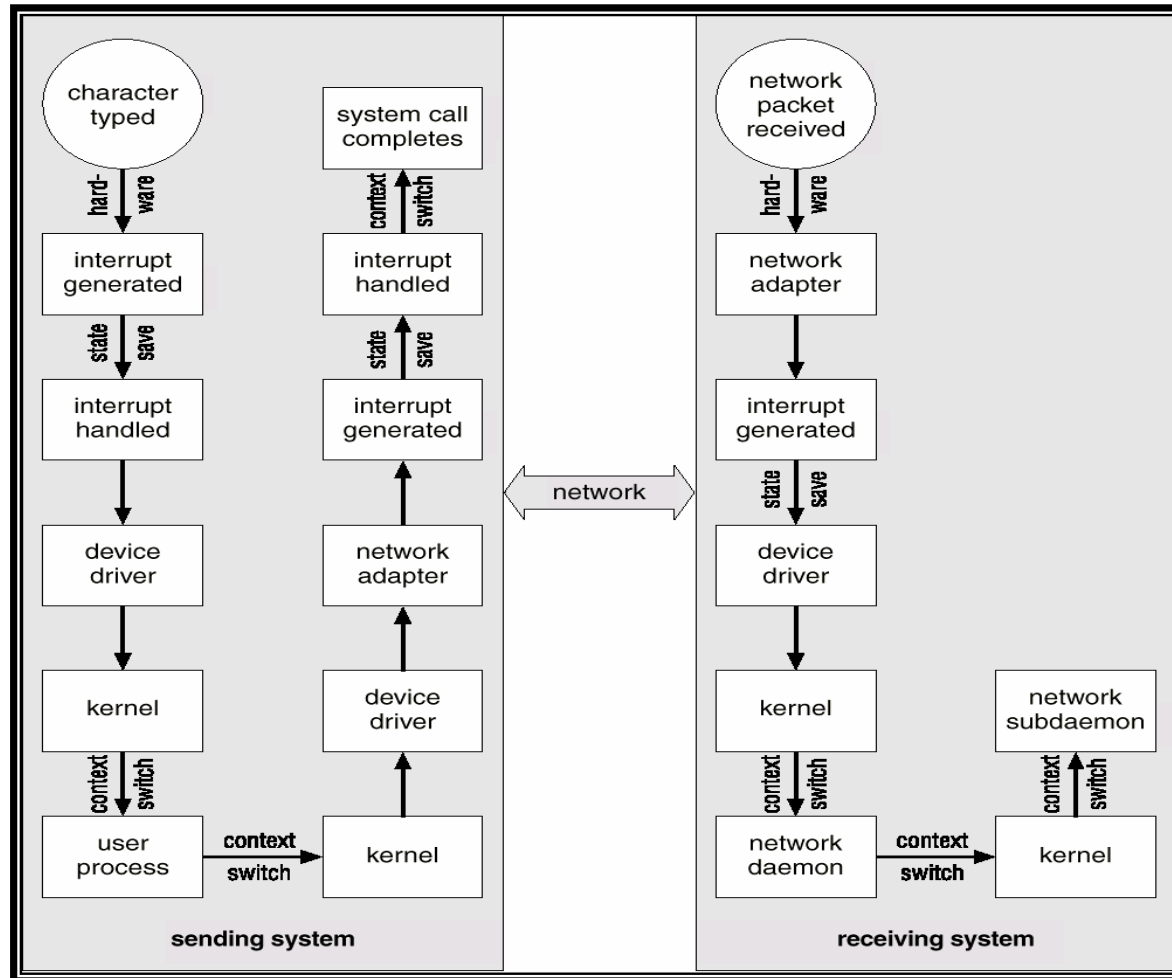
## (Performance)

---

- I/O là một lĩnh vực chính trong công tác bảo đảm hiệu năng hệ thống:
  - Yêu cầu CPU thực thi trình điều khiển thiết bị (device driver), mã lệnh của nhân I/O
  - Chuyển ngữ cảnh đối với các ngắt
  - Sao chép dữ liệu
  - Giao thông trên mạng cũng gây nhiều điều rắc rối

# Hiệu năng (2)

## Giao tiếp giữa các máy tính



# Hiệu năng (3)

## Cải thiện hiệu năng

---

- Giảm số lần chuyển ngữ cảnh
- Giảm việc sao chép dữ liệu
- Giảm các ngắt bằng cách sử dụng các bộ phận truyền dữ liệu với dung lượng lớn, các bộ điều khiển nhanh, polling
- Sử dụng DMA
- Chuyển các hàm cơ sở xử lý (processing primitives) vào phần cứng (device controller)
- Cân bằng CPU, bộ nhớ, bus, và hiệu năng I/O để đạt được thông lượng (throughput) cao nhất



# Hiệu năng (4)

## Sự tiến hóa về chức năng của thiết bị

