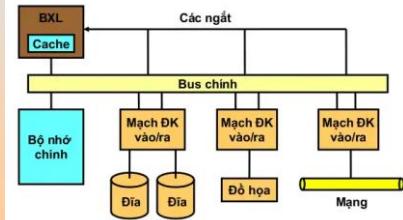


## Chương 6

# Hệ thống vào/ra (Input/Output Systems)



### Mục tiêu

- Khám phá cấu trúc I/O subsystem của HĐH
- Thảo luận các nguyên lý của phần cứng vào-ra và sự phức tạp của nó.
- Cung cấp chi tiết về các khía cạnh hiệu năng của phần cứng và phần mềm vào-ra.

### Nội dung:

- |   |                           |
|---|---------------------------|
| ■ Phần cứng vào-ra  | I/O Hardware              |
| ■ Giao diện vào-ra ứng dụng   | Application I/O Interface |
| ■ Hệ thống con vào-ra   | Kernel I/O Subsystem      |
| ■ Chuyển yêu cầu vào/ra tới thao tác phần cứng Transforming I/O Requests to Hardware Operations |                           |
| ■ Dòng/Luồng  | Streams                   |
| ■ Hiệu năng   | Performance               |

### Mở đầu

- Hai công việc chính của một máy tính:
  - Vào-ra
  - Xử lý
- Trong nhiều trường hợp, công việc chính là vào-ra
  - duyệt một trang web, chỉnh sửa một file
- Vai trò của HĐH trong vào-ra của máy tính là quản lý và điều khiển các thiết bị vào-ra và các hoạt động vào-ra.
- Các thiết bị vào-ra rất đa dạng về chức năng và tốc độ
  - cần các phương pháp điều khiển khác nhau, tạo thành *hệ thống con vào-ra (I/O subsystem)* của kernel.

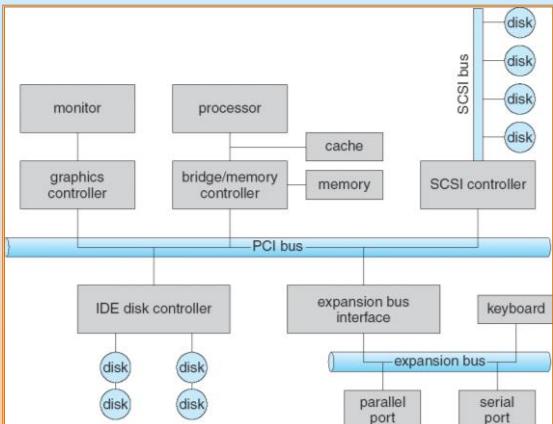
## 6.1. I/O Hardware

- Có vô số loại thiết bị vào-ra
- Các khái niệm chung:
  - **Port:** cổng giao tiếp giữa thiết bị vào-ra với máy tính, vd: serial, parallel, usb, PS/2 ports...
  - **Bus (daisy chain or shared direct access):** tập dây dẫn truyền thông tin giữa các thành phần trong máy tính và trong các thiết bị vào-ra, vd: PCI bus, SCSI bus
  - **Controller (host adapter):** tập hợp các thiết bị điện tử điều khiển cổng, bus, hoặc thiết bị vào-ra; controller có 1 hay nhiều thanh ghi dành cho các tín hiệu dữ liệu và điều khiển; vd: serial-port controller là 1 chip.

## I/O hardware (tiếp)

- Bộ điều khiển thiết bị (device controller) hỗ trợ các giao thức kết nối, buffering, caching...
- Các lệnh vào-ra (I/O instructions) điều khiển các thiết bị
- Các thiết bị có địa chỉ, được sử dụng bởi:
  - Các lệnh vào-ra trực tiếp: chuyển dữ liệu qua các cổng vào-ra
  - Vào-ra theo bản đồ bộ nhớ (**Memory-mapped I/O**)
- Có bộ điều khiển thiết bị có thể hỗ trợ cả 2 kỹ thuật vào-ra trên:
  - graphics controller có các cổng vào-ra cho các hoạt động điều khiển cơ bản, đồng thời có vùng bộ nhớ ảnh xạ lớn để lưu nội dung màn hình.
  - các tiến trình ghi dữ liệu lên vùng nhớ trên, controller sẽ tạo ảnh màn hình theo dữ liệu đó.
  - ghi hàng triệu byte lên vùng nhớ nhanh hơn nhiều so với thực hiện hàng triệu lệnh vào-ra trực tiếp.

### Cấu trúc bus PC thông thường



### Một số địa chỉ các cổng vào-ra thiết bị trên 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)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)

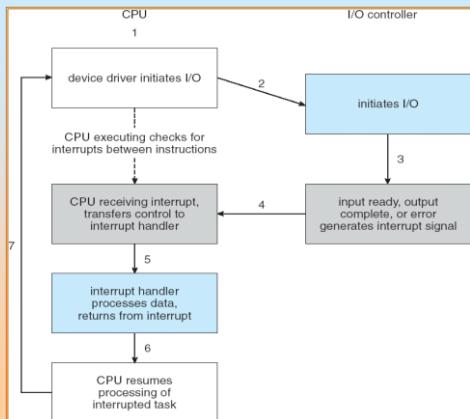
## Polling - thăm dò lần lượt

- CPU cần xác định trạng thái của thiết bị  $\Rightarrow$  thực hiện Polling
  - Command-ready: controller sẵn sàng thực hiện lệnh
  - Busy: controller đang bận
  - Error
- Là vòng lặp busy-wait đợi vào-ra từ thiết bị:
  - đọc thanh ghi trạng thái lặp đi lặp lại cho đến khi bit *busy*= 0 để thực hiện tiếp lệnh vào-ra;
  - vd: đọc trạng thái của serial port để tránh tràn buffer gây mất dữ liệu.
- Nói chung hoạt động polling là có hiệu quả, nó chỉ không hiệu quả khi được cố gắng thực hiện lặp lại nhiều lần mà hiếm khi tìm thấy một thiết bị sẵn sàng.
- Giải pháp: Mỗi controller nên có cơ chế báo trạng thái rõ ràng cho CPU.

## Interrupts - Ngắt

- Thiết bị vào-ra yêu cầu ngắt CPU. Thực hiện xong mỗi lệnh, CPU đọc dòng yêu cầu ngắt (trên CPU) chứa các ngắt đến.
- Các ngắt có thể là:
  - Ngắt không che được (non-maskable): báo các tình trạng lỗi phải xử lý ngay, vd: lỗi chia 0, lỗi stack, page fault
  - Ngắt che được (maskable): có thể bị bỏ qua hoặc bị hoãn xử lý, vd: ngắt gây ra bởi thiết bị
- Khi có ngắt:
  - CPU lưu trạng thái đang làm việc, vd: đ/c trả về, các g/t hiện thời
  - Nhảy đến thường trình xử lý ngắt (có địa chỉ xác định trong bộ nhớ)
  - Trình xử lý ngắt (Interrupt handler) xác định nguyên nhân ngắt, xử lý và thực hiện lệnh trả về từ ngắt
  - CPU khôi phục trạng thái trước ngắt

## Vòng lặp xử lý ngắt vào-ra



## Interrupts (tiếp)

- Trình xử lý ngắt tìm nguyên nhân ngắt trong bảng vector ngắt
- Vector ngắt gửi các ngắt tới đúng trình xử lý nó dựa trên mức ưu tiên ngắt
- Cơ chế ngắt cũng được dùng để xử lý ngoại lệ

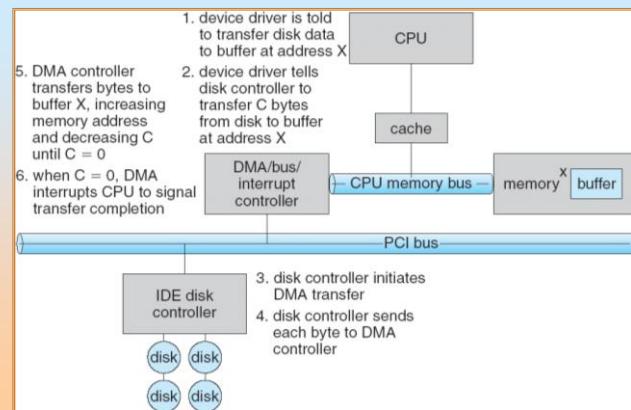
## Bảng vector ngắt của bộ VXL Intel Pentium

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
19-31	(Intel reserved, do not use)
32-255	maskable interrupts

## Direct Memory Access (DMA)

- Được sử dụng để tránh phải sử dụng vào-ra bằng chương trình (programmed I/O) khi lượng dữ liệu cần chuyển lớn.
- Cần có DMA controller (DMAC)
- Chuyển dữ liệu trực tiếp giữa thiết bị vào-ra và bộ nhớ mà không cần qua CPU.

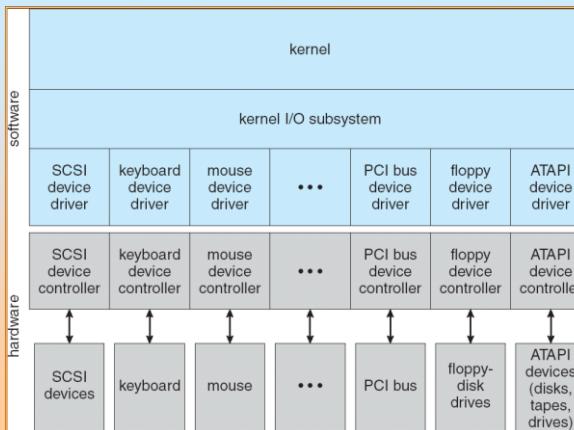
## Sáu bước thực hiện truyền dữ liệu DMA



## 6.2. Giao diện vào-ra ứng dụng

- Làm thế nào để 1 ứng dụng có thể mở một file trên đĩa mà không cần biết loại đĩa là gì?
- Sự khác nhau của các thiết bị vào-ra được gói gọn trong các kernel module được gọi là device driver, thay đổi đối với mỗi thiết bị nhưng tuân theo những giao diện chuẩn.
- Lớp device-driver ẩn đi sự khác nhau trong các I/O controller đối với I/O subsystem của kernel.
- Các thiết bị được phân loại nhiều cách khác nhau
  - Vào-ra theo chuỗi ký tự hoặc khối
  - Truy nhập tuần tự hoặc ngẫu nhiên
  - Có thể chia sẻ hoặc chuyên dụng
  - Tốc độ xử lý
  - read-write, read only, hoặc write only

## Một cấu trúc Kernel I/O



## Đặc điểm của các thiết bị vào-ra

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 thiết bị vào ra theo khôi và theo ký tự

- Các thiết bị vào ra theo khôi chủ yếu gồm các ổ đĩa
  - Các lệnh bao gồm: read, write, seek
  - Các ứng dụng thường truy nhập các thiết bị này qua một giao diện hệ thống file (file-system access) hoặc mảng tuyến tính các khôi (raw I/O access)
  - Có thể truy nhập file theo bản đồ bộ nhớ
- Các thiết bị ký tự bao gồm: bàn phím, chuột, cổng nối tiếp
  - Các lệnh gồm: get, put
  - Trên đỉnh của giao diện này, các thư viện có thể được xây dựng cho phép các thao tác chỉnh sửa và xử lý buffer; vd: khi ấn phím backspace, ký tự trước đó bị loại ra khỏi dòng vào (input stream)

## Các thiết bị mạng

- Hầu hết các HDH cung cấp giao diện vào-ra dùng riêng cho mạng, khác với giao diện vào-ra sử dụng cho các đĩa.
- Unix và Windows NT/9x/2000 dùng giao diện socket
  - Tách riêng giao thức mạng khỏi hoạt động mạng
  - Có bao gồm chức năng select để quản lý tập các socket.
- Nhiều phương pháp giao tiếp liên tiến trình và giao tiếp mạng khác cũng được sử dụng (pipes, FIFOs, streams, queues, mailboxes)

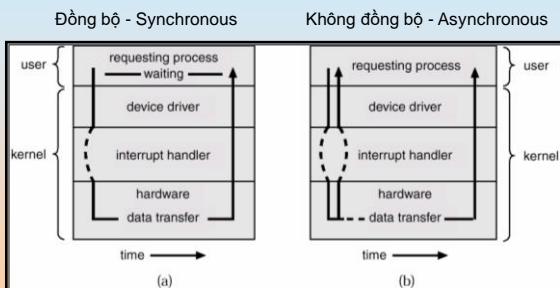
## Clocks và Timers

- Cung cấp thời gian hiện tại, thời gian trôi qua, thiết lập định thời (khởi sự hành động X tại thời điểm T)
- Programmable interval timer (bộ định thời có khoảng thời gian lập trình được) được sử dụng cho việc định thời gian, sinh các ngắt định kỳ.

## Blocking and Nonblocking I/O

- Blocking – tiến trình bị treo đến khi vào-ra kết thúc
  - Dễ hiểu, dễ sử dụng
  - Không thể áp dụng cho một số trường hợp
- Nonblocking – lời gọi vào-ra lặp lại nhiều nhất có thể
  - Sử dụng trong các giao diện người sử dụng, đưa dữ liệu vào bộ nhớ đệm (buffered I/O), vd: nhận dữ liệu vào từ bàn phím và chuột, trong khi xử lý và hiển thị trên màn hình.
  - Được thực hiện thông qua đa luồng
  - Lời gọi vào-ra lặp lại nhanh, với giá trị trả về cho biết bao nhiêu byte đã được chuyển
- Asynchronous – lập tức gọi return, không đợi vào-ra kết thúc, tiến trình vẫn chạy trong khi vào-ra đang được thực hiện → Khó sử dụng
  - I/O subsystem báo cho tiến trình biết khi nào vào-ra kết thúc

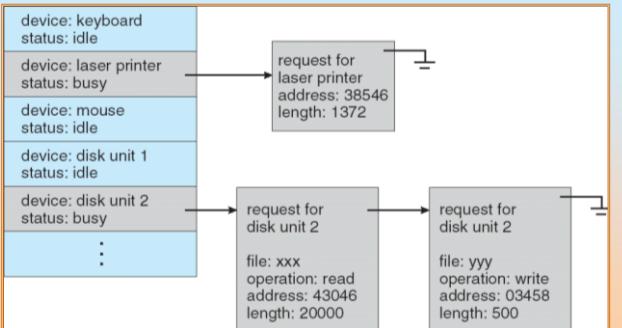
## Hai phương pháp vào-ra



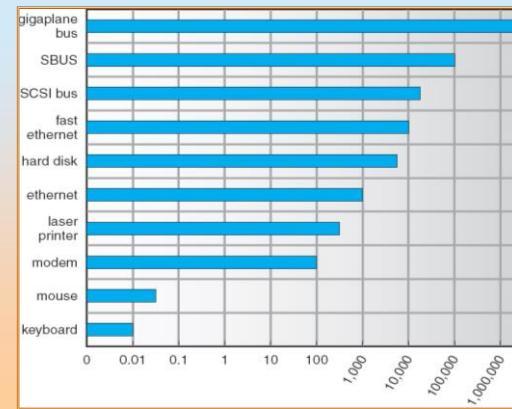
## 6.3. Kernel I/O Subsystem

- Kernel I/O Subsystem cung cấp nhiều dịch vụ liên quan đến vào-ra: scheduling, buffering, caching, spooling, error handling...
- **I/O Scheduling**
  - Lập lịch các yêu cầu vào-ra nhằm xác định một thứ tự thực hiện chúng hợp lý.
  - Thực hiện lập lịch vào-ra bằng cách duy trì một queue chứa các yêu cầu cho mỗi thiết bị
  - HĐH có gắng đạt tối ưu: cải thiện hiệu năng hệ thống, giảm thời gian đáp ứng trung bình...
- **Buffering** – chứa dữ liệu trong bộ nhớ trong khi vẫn truyền dữ liệu giữa các thiết bị
  - Để giải quyết vấn đề không khớp tốc độ giữa các thiết bị
  - Để giải quyết vấn đề không khớp kích thước dữ liệu truyền giữa các thiết bị

## Bảng trạng thái thiết bị



## Tốc độ truyền dữ liệu của các thiết bị trên Sun Enterprise 6000



## Kernel I/O Subsystem (tiếp)

- **Caching** – bộ nhớ nhanh giữ bản copy của dữ liệu
  - Luôn chỉ giữ một bản copy
  - Là chìa khóa của hiệu năng
- **Spooling** – giữ output cho một thiết bị
  - Nếu thiết bị chỉ có thể phục vụ một yêu cầu tại một thời điểm
  - vd: Printing
- **Device reservation** – cung cấp sự truy nhập riêng tới một thiết bị
  - Là các lời gọi hệ thống để phân phối và thu hồi các thiết bị cho các tiến trình
  - Là một cách đề phòng bể tắc

## Kernel I/O Subsystem (tiếp)

- **Error handling**
  - HĐH sử dụng bộ nhớ được bảo vệ (protected memory) có thể ngăn ngừa cản thận nhiều loại lỗi phần cứng và lỗi ứng dụng.
  - HĐH có thể phục hồi sau các lỗi đọc đĩa, ghi ảo, thiết bị không khả dụng.
  - Hầu hết trả về một mã lỗi khi yêu cầu vào-rõ lỗi

## Kernel I/O Subsystem (tiếp)

### ■ I/O Protection

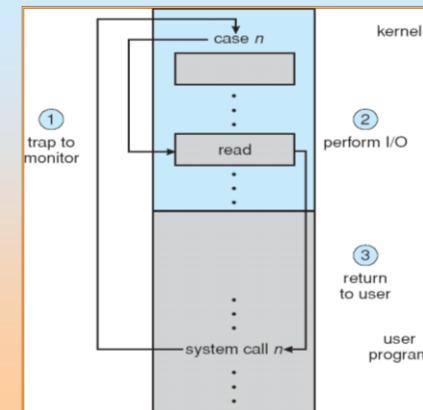
- Tiến trình của người sử dụng có thể vô tình hay cố ý cố gắng phá vỡ hoạt động bình thường thông qua các lệnh vào-ra bất hợp lệ
  - Tất cả các lệnh vào-ra được định nghĩa là đặc quyền
  - Vào-ra phải được thực hiện thông qua các system call.
    - Các vùng nhớ dùng cho ánh xạ (memory-mapped) và cổng vào-ra (I/O port) cũng phải được bảo vệ

## Kernel I/O Subsystem (tiếp)

### ■ Các cấu trúc dữ liệu kernel

- Kernel lưu thông tin trạng thái cho các thành phần vào-ra, bao gồm các bảng file mở, các kết nối mạng, trạng thái thiết bị vào-ra theo ký tự.
- Rất nhiều cấu trúc dữ liệu phức tạp để theo dõi các buffer, phân phối bộ nhớ, các khối nhớ “dirty”.
- Một số phương pháp hướng đối tượng và truyền thông điệp được sử dụng để thực thi vào-ra.

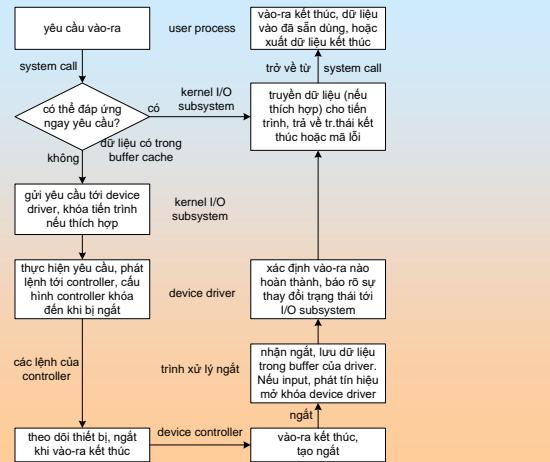
## Sử dụng system call để thực hiện vào-ra



## 6.4. Chuyển yêu cầu vào-ra tới các hoạt động phần cứng

- Các bước đọc một file từ đĩa cho một tiến trình:
  1. Xác định thiết bị chứa file
  2. Thông dịch tên tương ứng với thiết bị
  3. Đọc dữ liệu từ đĩa vào buffer
  4. Làm cho dữ liệu là khả dụng đối với tiến trình yêu cầu
  5. Trả điều khiển cho tiến trình

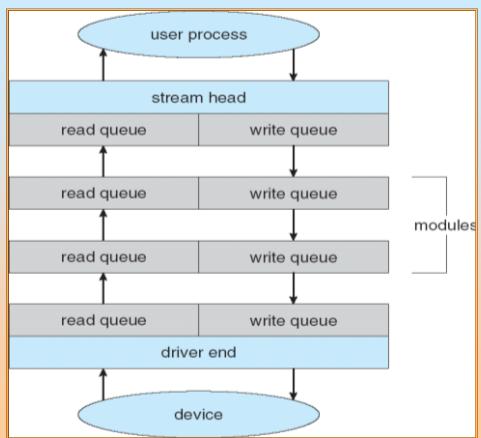
## Vòng đời của một yêu cầu vào-ra



## 6.5. STREAMS (UNIX, Solaris)

- **STREAM** – một kênh giao tiếp có thể giữ và nhận các tín hiệu đồng thời cùng một lúc giữa một tiến trình người sử dụng với một thiết bị (truyền song công hoàn toàn)
- Một STREAM bao gồm:
  - STREAM head giao tiếp với tiến trình người sử dụng
  - driver end giao tiếp với thiết bị
  - các STREAM module nằm giữa 2 phần trên.
- Mỗi module có một **read queue** và một **write queue**
- Phương thức message passing được sử dụng để giao tiếp giữa các queue

## Cấu trúc các STREAM



## 6.6. Performance

- Vào- ra là nhân tố chính trong hiệu năng hệ thống:
  - Luôn yêu cầu CPU thực hiện các mã của device driver, kernel I/O
  - Chuyển ngữ cảnh gây các ngắt
  - Copy dữ liệu luôn chiếm dụng bus
  - Lưu lượng mạng đặc biệt căng thẳng

## Cải thiện hiệu năng

- Giảm số lần chuyển ngữ cảnh
- Giảm copy dữ liệu
- Giảm ngắt bằng cách sử dụng truyền lượng lớn dữ liệu, các controller thông minh, polling
- Sử dụng DMA
- Cân bằng CPU, memory, bus, và hiệu năng vào-ra để đạt thông lượng cao nhất.



## BÀI TẬP

1. Vận chuyển DMA được thực hiện bởi:
  - a. Bộ điều khiển thiết bị.
  - b. CPU
  - c. Bộ nhớ trong
  - d. Câu a, b là sai
2. Trong hệ thống I/O đĩa thời gian để đầu đọc đến đúng khối cần thiết trên một track gọi là:
  - a. Seek time
  - b. Transfer time
  - c. Latency time.
  - d. Không câu nào đúng
3. Trong hệ thống I/O đĩa thời gian để đầu đọc đến đúng track cần thiết trên một đĩa gọi là:
  - a. Seek time.
  - b. Transfer time
  - c. Latency time
  - d. Không câu nào đúng

## BÀI TẬP

4. Tài nguyên của hệ thống bao gồm:
  - a. Bộ nhớ, bộ xử lý, bộ nhớ ngoài, máy in
  - b. Bộ nhớ, bộ xử lý, hệ điều hành, các thiết bị vào ra
  - c. Bộ nhớ, bộ xử lý và các thiết bị vào ra
  - d. Cả 3 đều sai.
5. Trong quản lý thiết bị ngoại vi, các máy tính làm việc theo nguyên tắc phân cấp nào?
  - a. Processor – Thiết bị ngoại vi – Thiết bị điều khiển
  - b. Processor – Thiết bị điều khiển – Thiết bị ngoại vi.
  - c. Thiết bị điều khiển – Thiết bị ngoại vi – Processor
  - d. Không đáp án nào đúng