



09. I/O System

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

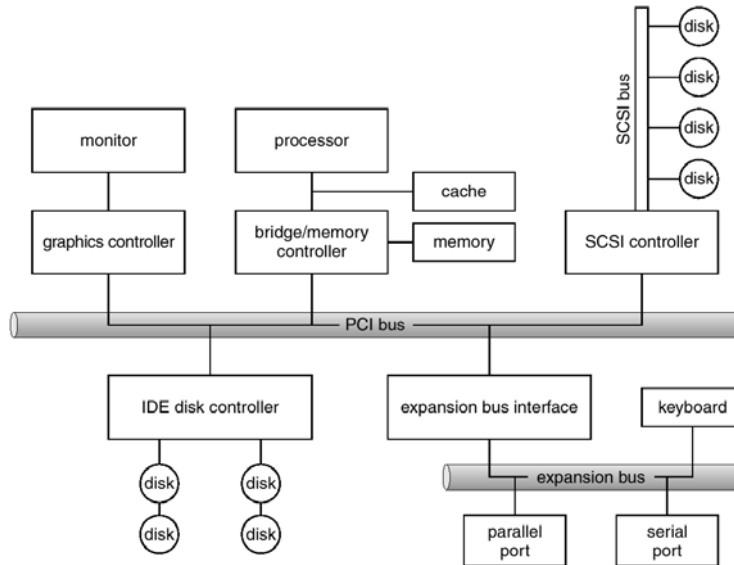


Thiết bị phần cứng I/O

- Các thiết bị I/O rất nhiều, đa chủng loại, khác biệt về nhiều mặt (hình dáng, kích thước, chức năng, tốc độ,...)
⇒ cần có các phương thức quản lý, điều khiển tương ứng khác nhau. Các phương thức này tạo nên lớp *I/O subsystem* của kernel.
- Điểm “*hội tụ*” chung cho các thiết bị I/O
 - Port
 - Bus (daisy chain, shared direct access)
 - Controller (device controller, SCSI host adapter)
- Cơ chế giao tiếp giữa CPU và thiết bị I/O
 - Dùng lệnh I/O để tác động trực tiếp lên các thanh ghi (dữ liệu, trạng thái, lệnh) của controller.
 - Memory-mapped I/O



Cấu trúc Bus trong PC



Khoa Công Nghệ Thông Tin – Đại Học Bách Khoa Tp.HCM

-X.III.3-



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

Khoa Công Nghệ Thông Tin – Đại Học Bách Khoa Tp.HCM

-X.III.4-

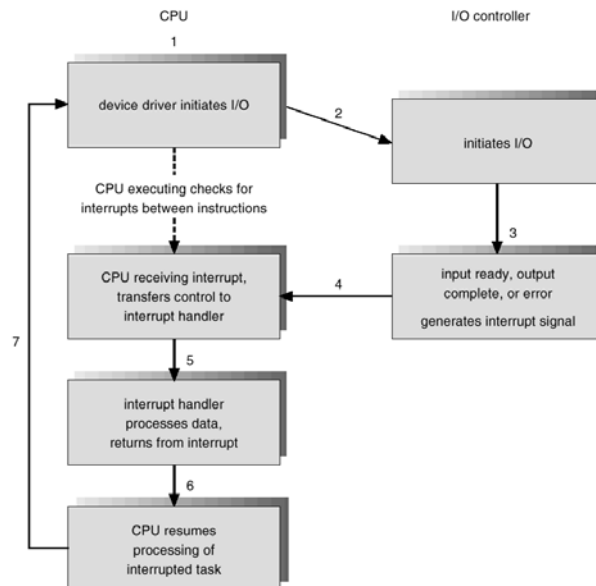


Các cơ chế thực hiện I/O

- Cơ chế polling (busy-waiting)
 - Kiểm tra trạng thái của thiết bị khi muốn thực hiện I/O
 - Ready hoặc Busy hoặc Error
 - Tiêu tốn thời gian lặp quay vòng để kiểm tra trạng thái (busy-wait) và thực hiện I/O.
- Cơ chế ngắt quãng (interrupt)
 - CPU có một ngõ Interrupt Request (IR), được kích hoạt bởi thiết bị I/O.
 - Nếu có ngắt xảy ra (IR = active) \Rightarrow chuyển quyền điều khiển cho trình điều khiển ngắt (interrupt handler)
 - Các ngắt có thể bị che (maskable) hoặc không bị che (non-maskable)
 - Hệ thống có một bảng vector ngắt chứa địa chỉ các trình phục vụ ngắt (interrupt routine).
 - Cơ chế ngắt cũng có thể dùng xử lý các sự kiện khác trong hệ thống (lỗi chia cho 0, lỗi vi phạm vùng nhớ,...)



Chu kỳ I/O với ngắt quãng





Bảng vector ngắt trên 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

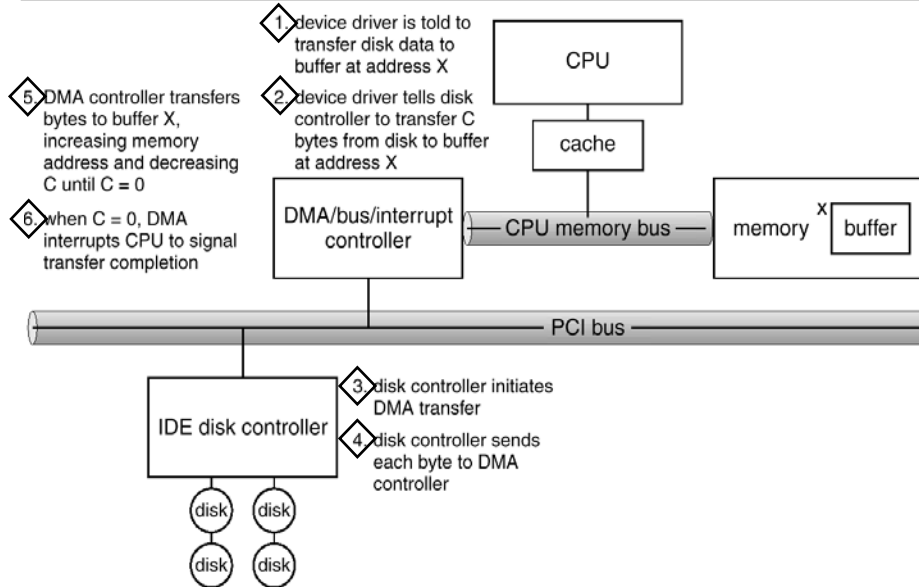


Các cơ chế I/O (t.t)

- Cơ chế DMA (Direct Memory Access)
 - Các cơ chế polling và interrupt, gọi chung là programmed I/O, không thích hợp khi thực hiện di chuyển khối lượng lớn dữ liệu.
 - Cơ chế DMA cần có phần cứng hỗ trợ đặc biệt, đó là DMA controller
 - Cơ chế 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



Khoa Công Nghệ Thông Tin – Đại Học Bách Khoa Tp.HCM

-X III.9-



Giao diện I/O cấp ứng dụng

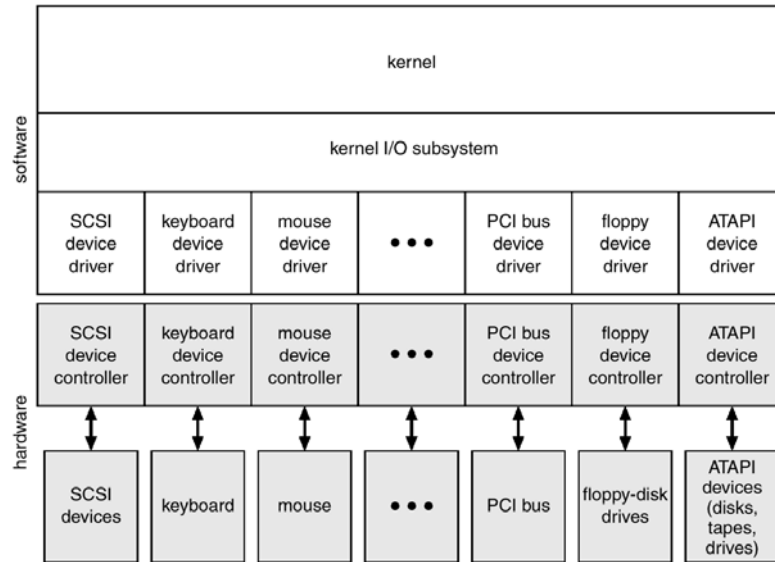
- ❑ **Mục tiêu:** OS cung cấp một giao diện I/O chuẩn hóa, thuần nhất cho các ứng dụng.
 - Ví dụ: một ứng dụng in tài liệu ra máy in mà không cần biết hiệu máy in, đặc tính máy in,...
- ❑ Giao diện làm việc đó là các **I/O system call** của OS.
- ❑ Trình điều khiển thiết bị (device driver) sẽ là cầu nối giữa kernel và các bộ điều khiển thiết bị (device controller).
- ❑ Đặc tính của thiết bị rất đa dạng
 - Character-stream vs. Block
 - Sequential vs. Random-Access
 - Sharable vs. Dedicated
 - Tốc độ truy xuất
 - Read-write, Read Only, Write-Only.

Khoa Công Nghệ Thông Tin – Đại Học Bách Khoa Tp.HCM

-X III.10-



Cấu trúc I/O cấp kernel



Khoa Công Nghệ Thông Tin – Đại Học Bách Khoa Tp.HCM

-X III.1.1-



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

Khoa Công Nghệ Thông Tin – Đại Học Bách Khoa Tp.HCM

-X III.1.2-



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.



Các kiểu truy xuất 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.
- **Non-blocking** – process sẽ tiếp tục thực thi ngay sau lệnh gọi I/O.
 - Ví dụ: data copy (buffered I/O)
 - Thường hiện thực với multi-threading.
 - 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.

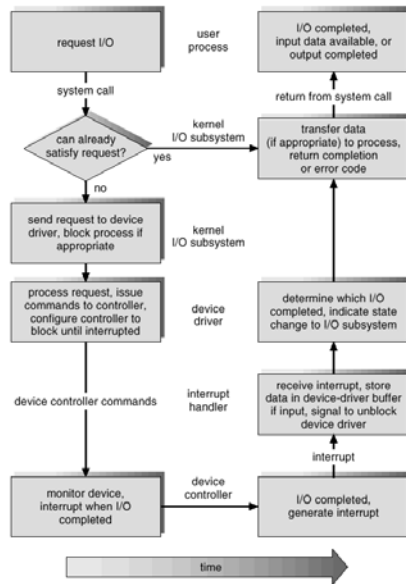


Bộ phận I/O subsystem ở kernel

- ❑ 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 can recover from disk read, device unavailable, transient write failures
 - Most return an error number or code when I/O request fails
 - System error logs hold problem reports



Chu kỳ của một yêu cầu I/O





Hiệu suất thực hiện I/O

- Hiệu suất I/O ảnh hưởng rất lớn đến hiệu suất toàn hệ thống. Tốc độ I/O thường rất chậm.
 - Yêu cầu 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. Riêng các thiết bị mạng thì phụ thuộc vào băng thông của mạng máy tính.
- Tăng hiệu suất thực hiện 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 cơ chế polling,...)
 - Sử dụng DMA nếu có thể.



Ví dụ giao tiếp qua mạng

