



# Chương 9: I/O System

---

*Tìm hiểu cơ chế Hệ điều hành điều khiển I/O devices*

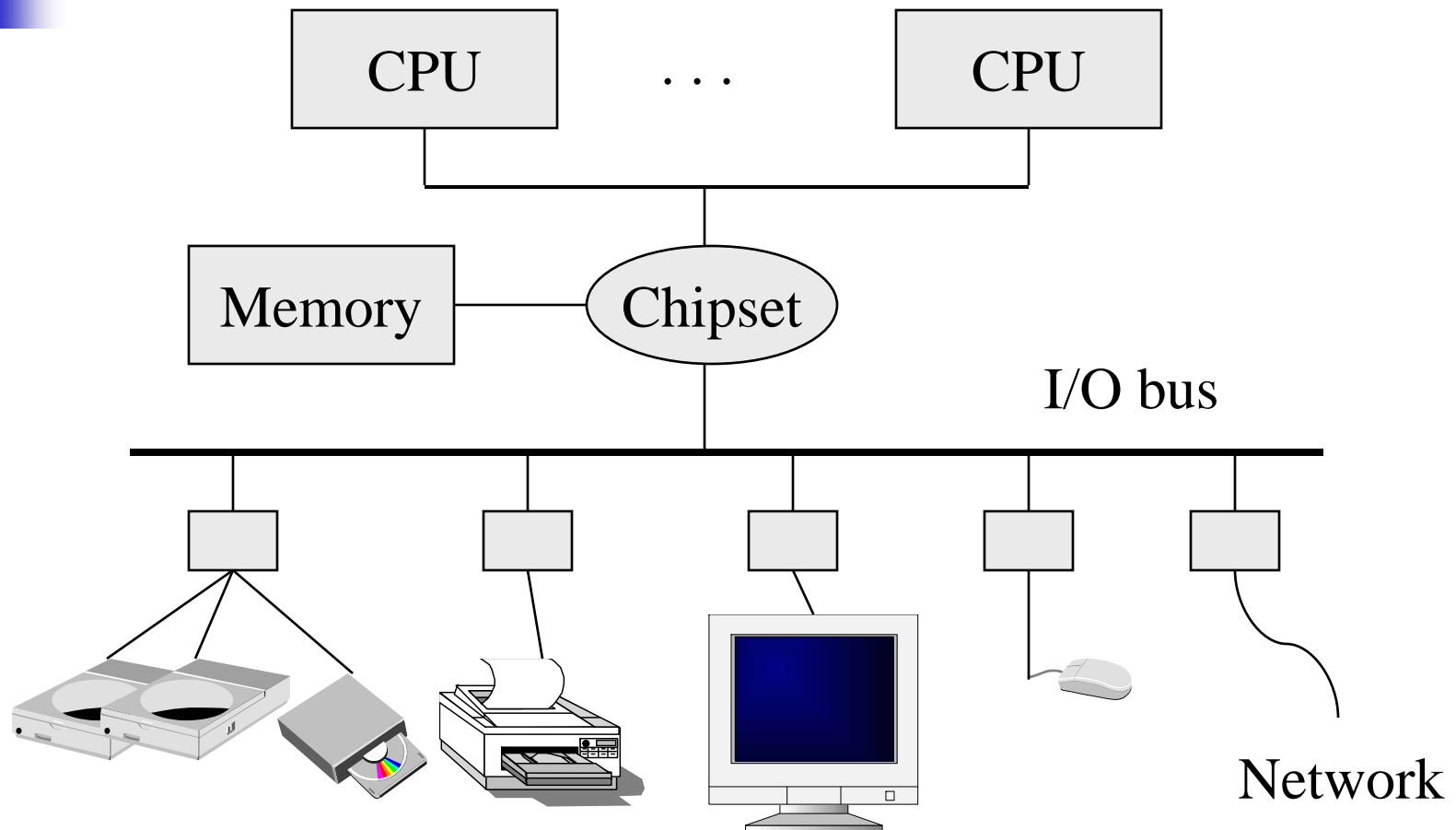


# Nội dung

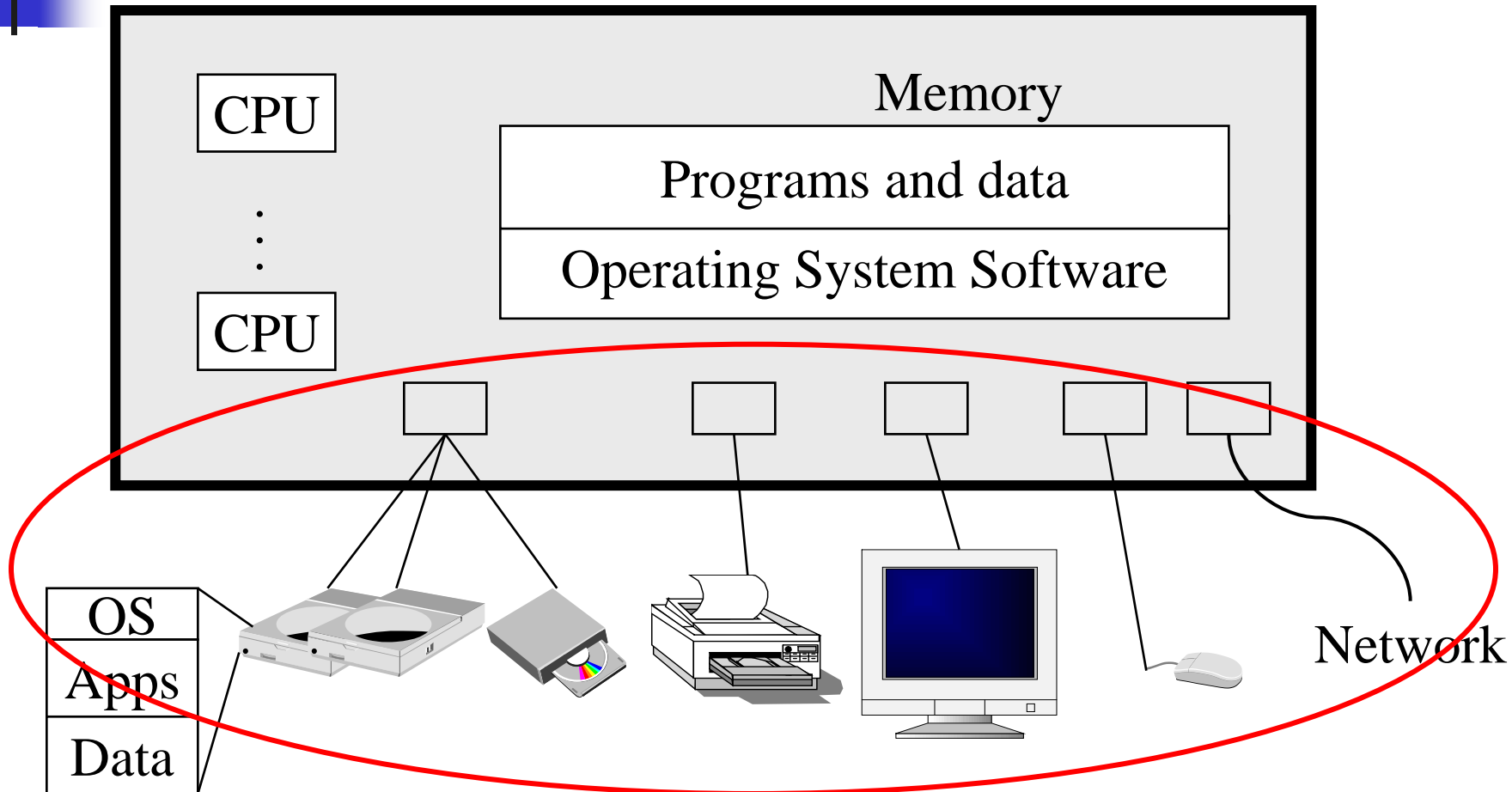
---

- Các khái niệm I/O cơ bản
- Ngắt
- DMA
- Memory Mapped I/O (thiết bị được ánh xạ bởi bộ nhớ)

# Máy tính theo quan điểm phần cứng



# Một hệ thống máy tính (hộp đen)





# 1. Phần cứng I/O

---

- Phần cứng hỗ trợ vào/ra cơ bản
  - ports (cổng), buses (bus), devices (các thiết bị) and controllers (bộ điều khiển)
- Phần mềm vào/ra
  - Các bộ điều khiển ngắt, Các Driver của thiết bị, Phần mềm độc lập với thiết bị, Phần mềm vào/ra trong không gian người dùng
- Các thiết bị vào/ra
  - Các thiết bị khối, Các thiết bị đầu cuối (dạng ký tự), Thiết bị đồ họa giao tiếp người dùng (màn hình, máy in, máy chiếu...), Các thiết bị đầu cuối mạng



# Các thiết bị

---

- Thiết bị
  - Các thiết bị lưu trữ (đĩa từ, băng từ)
  - Các thiết bị truyền thông tin (card mạng, modem)
  - Các thiết bị giao tiếp với con người (màn hình, bàn phím, chuột)
  - Các thiết bị chuyên dụng (cần điều khiển (VD tay lái trong trò đua ô tô))
- **Các thiết bị khối** (VD, đĩa từ) khác với **Các thiết bị ký tự** (VD, bàn phím)



# Phân lớp thiết bị

---

- Các thiết bị ký tự ; các thiết bị khối
- Các thiết bị truy cập tuần tự ; thiết bị truy cập ngẫu nhiên
- Thiết bị đồng bộ ; thiết bị không đồng bộ
- Thiết bị chuyên dụng ; thiết bị có thể dùng chia sẻ
- Tốc độ hoạt động
- Thiết bị chỉ đọc, chỉ ghi, đọc – ghi.



# Các thiết bị khối

---

- Thiết bị khối – truyền các khối dữ liệu (VD, đĩa từ)
- Các lệnh: read, write, seek (nếu là thiết bị truy cập ngẫu nhiên)
- Các bộ truy cập file được ánh xạ bởi bộ nhớ (Memory-mapped) được xếp vào tầng đầu tiên của các bộ điều khiển (driver) truy cập thiết bị khối.





# Các thiết bị ký tự

---

- Thiết bị ký tự - truyền từng byte một (VD, bàn phím)
- Các lệnh: get, put một ký tự
- Có thể tạo các thư viện trên “đỉnh” của các thiết bị này, để cung cấp việc truy cập từng dòng một bằng cách tạo ra các bộ đệm và các dịch vụ



# Các thiết bị mạng

---

- Network socket interface
- Các lệnh; tạo socket, kết nối socket cục bộ tới một địa chỉ ở xa, gửi đến một socket, nhận từ một socket, lấy thông tin từ một socket



# Giao tiếp giữa Thiết bị-Máy tính/Thiết bị-Thiết bị

---

- Về mặt vật lý: thông qua các tín hiệu được truyền qua dây cáp hoặc qua không khí
- Về mặt logic: thông qua một mối nối (VD, Serial port, RS-232 port)
- Thông thường, nhiều thiết bị được kết nối thông qua một bus
  - Một tập các đường truyền và một tập giao thức để đặc tả các thông điệp được gửi đi trên đường truyền
    - VD, các chuẩn ANSI, IEEE or ISO
    - Các trình điều khiển đĩa sử dụng chuẩn IDE hay SCSI



# Bộ điều khiển thiết bị

- Các thiết bị vào ra thường gồm một thành phần **thiết bị** và một thành phần **điện tử**. Thành phần điện tử được gọi là **Bộ điều khiển thiết bị** hay **card điều khiển** (thường là một bảng mạch IC).
- Thành phần thiết bị chính là thiết bị đó.
- Giao tiếp giữa bộ điều khiển và thiết bị là giao tiếp ở mức thấp. Các bộ điều khiển đã được cài đặt sẵn.
- Ví dụ:
  - Các bộ điều khiển đĩa từ chuyển các chuỗi bit, thành các khối byte, rồi kiểm tra lỗi. Đầu tiên các khối byte được đưa vào bộ đệm trong bộ điều khiển. Sau đó kiểm tra giá trị checksum, nếu không có lỗi thì được chuyển vào bộ nhớ chính.



# Device Controller

---

- Bộ điều khiển đĩa từ— cài đặt cho đĩa từ giao thức thực hiện các việc: kiểm tra lỗi, lấy dữ liệu trước, tạo vùng đệm và caching
- Bộ điều khiển có các thanh ghi phục vụ cho thông tin dữ liệu và điều khiển
- CPU và các bộ điều khiển giao tiếp thông qua các chỉ thị lệnh vào/ra và dữ liệu được đưa vào các thanh ghi này
- **Memory-mapped I/O:** CPU “ghi” (và đọc) các bộ điều khiển giống như việc nó ghi và đọc một vùng bộ nhớ (các chỉ thị VÀO và RA, tương tự như các chỉ thị lệnh TẢI và GHI)

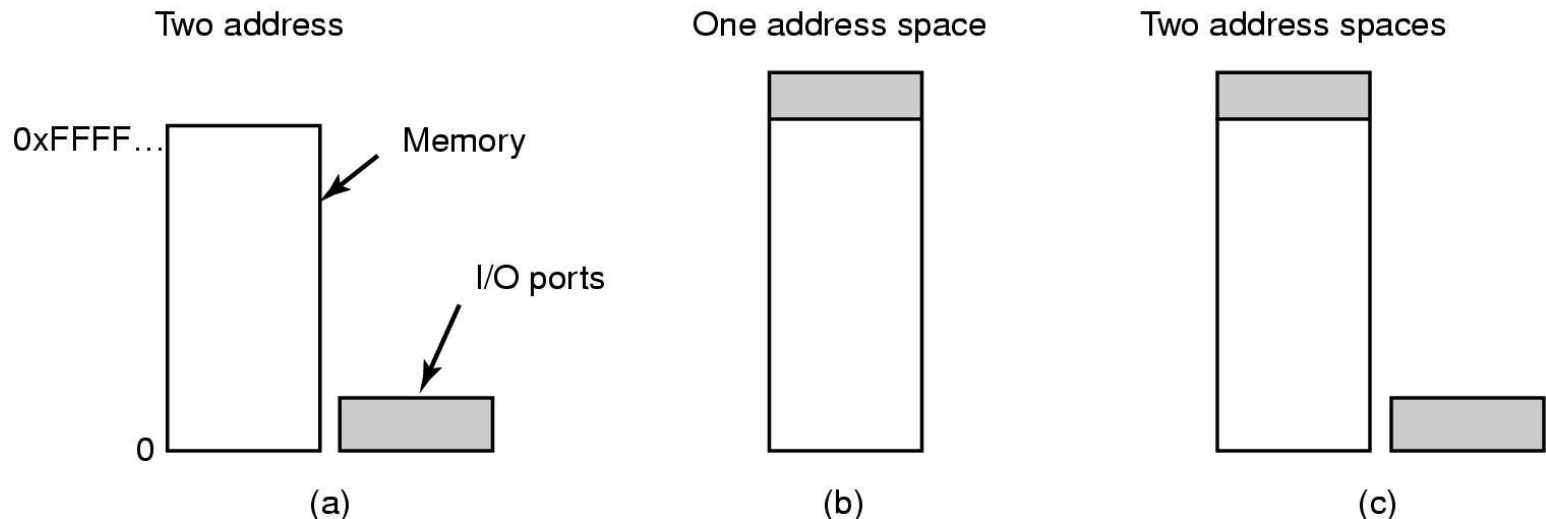


# Memory-Mapped I/O: “Ports”

---

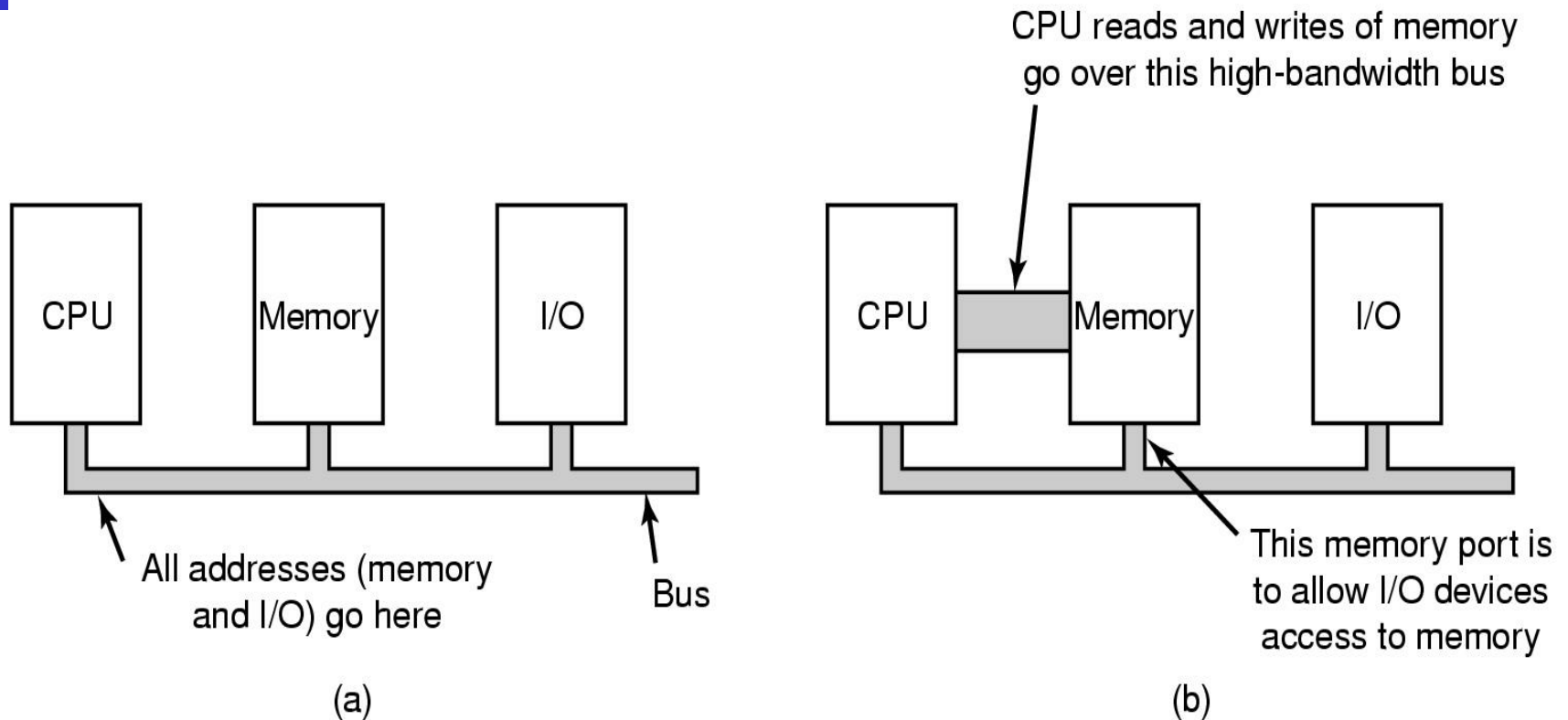
- 4 thanh ghi (các cổng) – trạng thái, điều khiển, dữ liệu vào, dữ liệu ra
  - **Trạng thái** - chỉ trạng thái mà lệnh có hoàn thành hay không hay là thiết bị có lỗi, ...
  - **Điều khiển** – đưa ra một lệnh hoặc thay đổi mode của thiết bị
  - **Dữ liệu vào** – để nhận dữ liệu vào
  - **Data-out** – dữ liệu để chuyển ra output
- Kích thước của các thanh ghi - 1 to 4 bytes
- CPU phải đợi cho thanh ghi trạng thái bằng 0 trước khi đưa ra chỉ thị lệnh tiếp theo

# Memory-Mapped I/O (1)



- (a) I/O và không gian bộ nhớ tách biệt nhau
- (b) Memory-mapped I/O
- (c) Lai giữa hai mô hình

# Memory-Mapped I/O (2)



(a) Kiến trúc bus đơn  
(b) Kiến trúc bus kép





# Cách tiếp cận khác: Các ngắt

---

- CPU được hỗ trợ lưu thông tin ngắt, sau khi thi hành một chỉ thị lệnh, nó sẽ đọc được thông tin này
  1. Các thiết bị gọi một ngắt
  2. CPU nhận ngắt và ghi lại trạng thái của tiến trình đang chạy (nếu cần)
  3. CPU gửi tới bộ điều khiển ngắt
  4. Bộ điều khiển ngắt xác định nguồn gốc, phục vụ thiết bị rồi xóa thông tin ngắt đó trong CPU
- Tại sao phải ngắt?



# Bộ điều khiển ngắt

---

- Tại thời điểm khởi động, HĐH kiểm tra các bus để xác định xem có các thiết bị nào và cài đặt trình điều khiển ngắt tương ứng trong bảng vector ngắt.
- Bảng vector ngắt được dùng để tìm vị trí mã lệnh của trình điều khiển ngắt cho một ngắt tương ứng.
  - Một ngắt vào/ra được sử dụng bởi một bộ điều khiển thiết bị để báo hiệu rằng thiết bị đã sẵn sàng.



# Direct Memory Access (DMA)

## Truy cập bộ nhớ trực tiếp

---

- Vấn đề: đọc một khối dữ liệu (VD, một trang từ một thiết bị khối (VD, đĩa từ))
- Lập trình vào/ra (PIO)
  - Dùng CPU để đọc bit trạng thái và đưa dữ liệu vào thanh ghi điều khiển 1 byte một lần – TỐN KÉM để truyền dữ liệu lớn
- Direct memory access (DMA)
  - Dùng 1 bộ xử lý cho một mục đích chuyên dụng, được gọi là bộ điều khiển DMA

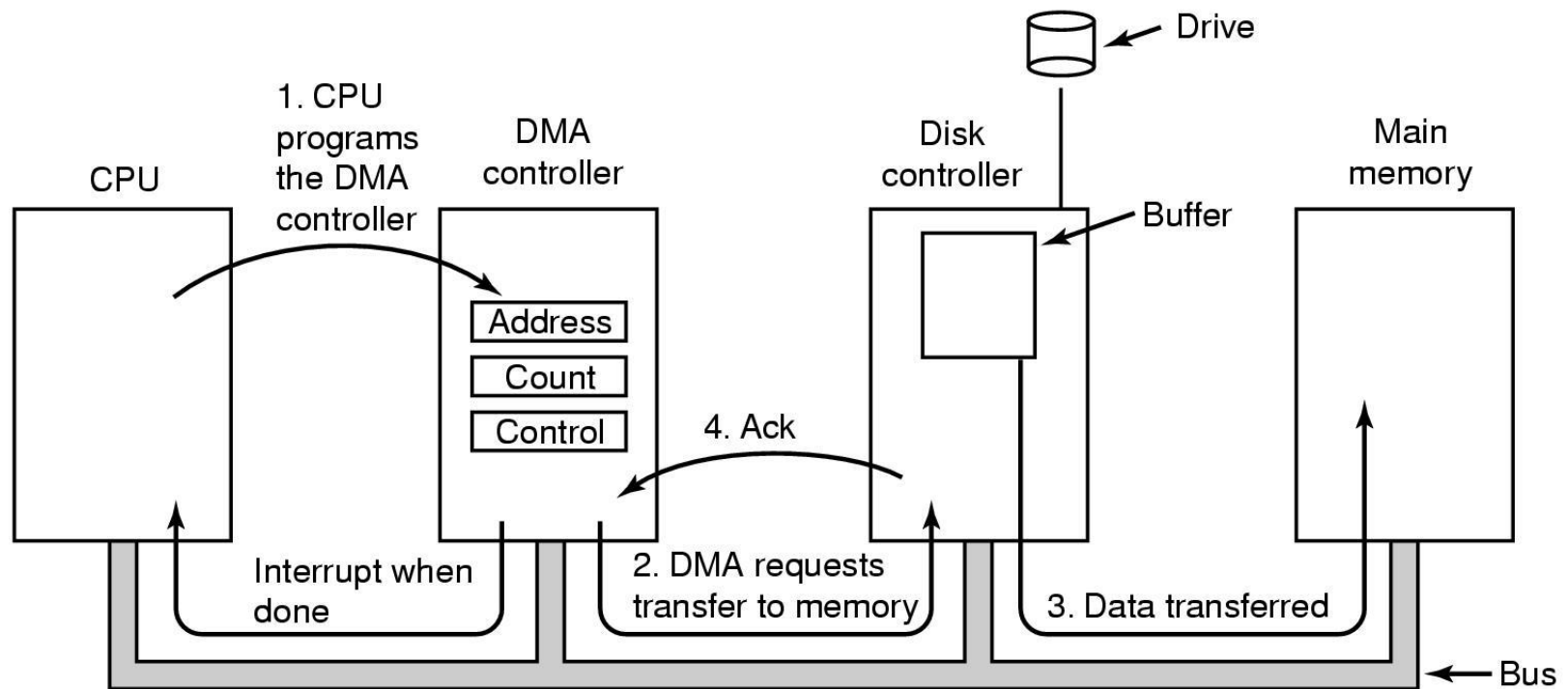


# Giao thức DMA-CPU

---

- CPU viết một khối lệnh DMA vào bộ nhớ
  - Trỏ tới nguồn, trỏ tới đích, số byte được truyền
- CPU viết địa chỉ của khối lệnh này vào bộ điều khiển DMA và tiếp tục công việc khác
- Bộ điều khiển DMA tiếp tục các thao tác với bus bộ nhớ và không cần sự hỗ trợ của CPU
  - CPU làm nhiệm vụ kích hoạt bộ điều khiển DMA

# Direct Memory Access (DMA)



## Operation of a DMA transfer

- Fly-by: thiết bị truyền trực tiếp tới bộ nhớ
- Burst mode: thiết bị truyền cả khối vào bộ nhớ mà không phải từng byte một



## 2. Phần mềm I/O

---

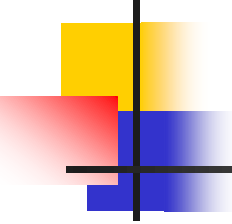
- Nguyên lý
- Cài đặt
- Các vấn đề liên quan



# Review

---

- I/O basic concepts
- Interrupts
- DMA
- Memory Mapped I/O



# I/O Software: Principles (nguyên tắc)

---

- Độc lập với thiết bị
  - Có thể viết chương trình truy cập tới các loại thiết bị vào ra khác nhau mà không cần phải có thêm đặc tả thiết bị.
  - Đọc các trang từ đĩa mềm hoặc đĩa cứng mà không cần phải sửa đổi chương trình
- Đặt tên thống nhất
  - Tên của một file hoặc một thiết bị là một chuỗi hoặc là một số nguyên mà không phụ thuộc vào thiết bị
  - Ví dụ: Trong Unix tên của tất cả các file và các thiết bị đều được đánh địa chỉ có dạng như `/dev/*` ...
- Đồng bộ (blocking) và không đồng bộ (interrupt-driven) transfers





# I/O Software Principles (cont.)

---

## ■ Bộ đệm

- Dữ liệu được lưu trữ tạm thời trước khi sử dụng
- Dữ liệu cần đưa ra bộ đệm vì có thể lúc đó nó chưa cần xử lý ngay
- Vd: Bộ điều khiển thiết bị đưa một dòng dữ liệu ra bộ đệm cho đến khi toàn bộ khối được đọc. Các gói tin mạng được truyền từ mạng, và cần đưa ra bộ đệm cho bộ xử lý giao thức.

## ■ Thiết bị chia sẻ và thiết bị chuyên dụng

- Chia sẻ → màn hình, đĩa từ. Chuyên dụng → bàn phím, băng từ.
- Chia sẻ → tốc độ chậm. Chuyên dụng → có nhiều nguy cơ xảy ra deadlock.



# I/O được lập trình sẵn

---

- 3 cách thực hiện I/O:
  - IO được điều khiển bởi ngắt
  - I/O sử dụng DMA
  - **IO phải lập trình:**
    - Mọi thao tác I/O phải được lập trình và được điều khiển.
    - VD: in một file từ chương trình người dùng ra máy in có nghĩa là dữ liệu được chuyển tới nhân, và HĐH gửi lần lượt từng ký tự (tới cổng điều khiển).
    - Điều quan trọng ở đây là CPU liên tục phải xem liệu thiết bị đã sẵn sàng để nhận ký tự thứ 2 chưa. Nó dùng thao tác **polling** (or busy waiting) trên cổng trạng thái.
    - CPU không thể làm việc gì khác trong lúc đó.



# Giao tiếp với bộ điều khiển: Polling

---

- Bộ điều khiển và CPU có mối quan hệ tương tự quan hệ giữa nhà sản xuất và khách hàng...
- Bộ điều khiển thông qua busy bit (bit trạng thái) để chỉ ra đó đang bận hay không
- CPU gửi tín hiệu các yêu cầu của nó thông qua command-ready bit (command register)

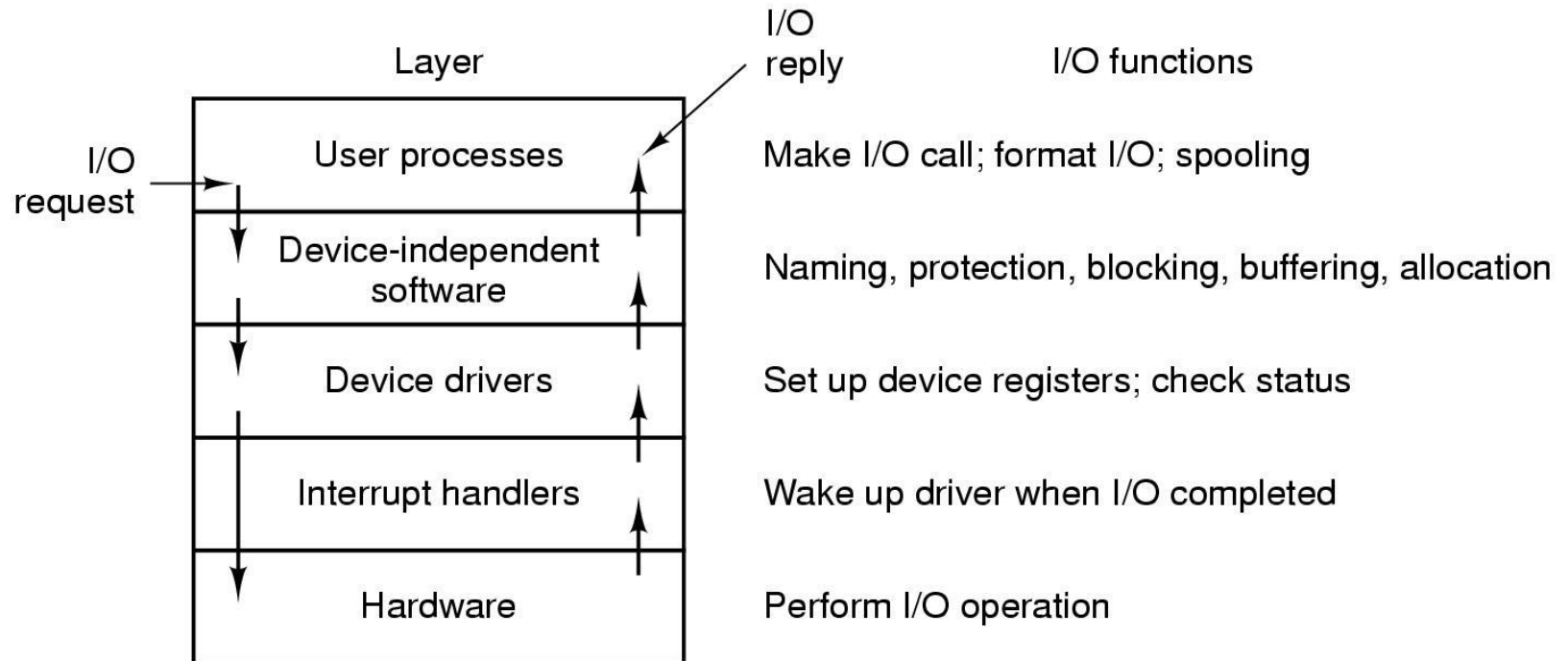


# So sánh

---

- Interrupt-driven I/O:
  - Ưu: giúp CPU không bị busy waiting
  - Nhược: Cần tính chi phí khởi tạo và điều khiển các ngắt
- Programmed I/O:
  - Ưu: Không cần hỗ trợ ngắt hay DMA
  - Nhược: tốn thời gian CPU
- I/O sử dụng DMA:
  - Ưu: giải phóng CPU khỏi thao tác I/O
  - Nhược: Bộ xử lý DMA chậm hơn CPU

# I/O Software



Layers of the I/O system and the main functions of each layer (typical)



# Trình điều khiển thiết bị

- Mã đặc tả thiết bị để điều khiển một thiết bị vào/ra, *luôn được viết bởi nhà cung cấp thiết bị (vendor)*
  - Mỗi bộ điều khiển có một số thanh ghi thiết bị dùng để ra lệnh cho thiết bị. Số các thanh ghi thiết bị và các lệnh khác nhau với các thiết bị khác nhau (VD, trình điều khiển chuột nhận các thông tin di chuyển của chuột, trình điều khiển đĩa từ nhận các thông tin về cung, rãnh,...).
- Một driver thiết bị là một phần của nhân HĐH
  - được biên dịch với HĐH
  - Tự động được tải vào nhân OS
- Mỗi driver thiết bị điều khiển
  - một loại thiết bị (VD, chuột)
  - một lớp các thiết bị tương tự nhau (VD, driver của đĩa từ chuẩn SCSI điều khiển nhiều đĩa từ có kích thước khác nhau và tốc độ khác nhau.)

# Chức năng của trình điều khiển thiết bị

- Nhận các yêu cầu đọc, ghi triu tượng từ một tầng trên độc lập với thiết bị.
- Khởi tạo thiết bị
- Quản lý các yêu cầu sử dụng và ghi lại các sự kiện
- Kiểm tra các tham số đầu vào hợp lệ hay không
- Chuyển từ đầu vào triu tượng sang các thông số cụ thể
  - Vd, chuyển từ các khối liên tiếp sang thông tin về đầu đọc, rãnh, cung để truy cập đĩa
- Kiểm tra các thiết bị xem có đang rồi hay không
- Điều khiển thiết bị bằng cách đưa ra một chuỗi các lệnh
  - trình điều khiển phải quyết định xem đưa ra lệnh nào



# Giao thức trình điều khiển thiết bị

---

- Sau khi trình điều khiển biết cần đưa ra lệnh nào, nó viết các lệnh này vào các thanh ghi thiết bị của bộ điều khiển
- Sau khi viết mỗi lệnh, nó kiểm tra xem nếu bộ điều khiển chấp nhận lệnh không và có đang sẵn sàng để nhận lệnh tiếp theo hay không.
- Sau khi lệnh được gửi đi, hoặc (a) trình điều khiển thiết bị đợi cho thiết bị gọi ngắt lại; hoặc (b) hoặc là dạng thiết bị không cần đợi vì lệnh đã được thi hành ngay lập tức.





# Tầng phần mềm vào/ra độc lập với thiết bị

---

- Chức năng:
  - giao tiếp thống nhất với các driver thiết bị
  - Tạo bộ đệm
  - thông báo lỗi
  - Cấp phát và giải phóng các thiết bị chuyên dụng
  - Cung cấp các kích cỡ khối độc lập với thiết bị

# Device-Independent I/O Software Layer

- Để cho HĐH biết càng ít về ngắt càng tốt, ngắt nên được ẩn đi.
- Cách tốt nhất để ẩn các ngắt là buộc trình điều khiển khởi tạo một thao tác vào/ra bị block cho tới khi thao tác vào/ra kết thúc và xảy ra ngắt.
- Khi ngắt xảy ra, bộ điều khiển ngắt điều khiển ngắt.
- khi điều khiển ngắt được thực hiện, bộ điều khiển ngắt unblocks trình điều khiển thiết bị và cho nó hoạt động lại.
- Mô hình này hoạt động nếu các trình điều khiển có cấu trúc như **các tiến trình nhân có trạng thái của riêng chúng**, có các ngăn xếp và bộ đếm chương trình.



# Giao tiếp thống nhất với các trình điều khiển thiết bị

---

- Mục đích: làm thế nào để làm cho tất cả các thiết bị I/O và các driver tương tự nhau
- Vấn đề: mỗi driver có một giao diện khác nhau với HĐH
- Giải pháp: không phải tất cả các thiết bị giống hệt nhau, nhưng chỉ có một số loại thiết bị.
  - VD: thiết bị khối và thiết bị ký tự
- các phần mềm độc lập thiết bị chỉ quan tâm tới việc ánh xạ tới tên của các thiết bị (VD, /dev/kbd or /dev/dsk0) với thiết bị thích hợp.



# Bộ đệm

---

- bộ đệm là một vùng nhớ lưu trữ dữ liệu trong khi nó được truyền giữa 2 thiết bị hoặc giữa một thiết bị và một ứng dụng.
- nguyên nhân sinh ra bộ đệm:
  - Giải quyết vấn đề **không hợp tốc độ** giữa dòng dữ liệu tới và nhận - dùng *đệm kép*
  - Thích hợp giữa các thiết bị có **kích thước truyền khác nhau**
  - Hỗ trợ **copy** cho các ứng dụng I/O - ứng dụng ghi vào một bộ đệm và HĐH copies tới một bộ đệm nhân và ghi nó vào đĩa từ
  - Nếu không có bộ đệm thì sẽ không hiệu quả, vì tiến trình người dùng phải làm việc với từng ký tự đầu vào.
- **Caching**
  - Cache là vùng bộ nhớ có tốc độ truy cập nhanh giữ các bản sao dữ liệu và cho phép truy cập hiệu quả hơn.
- khác nhau giữa cache và buffer?



# Content

---

- Phần mềm vào/ra
- Đồng hồ và bộ tính thời gian
- Quản lý công suất

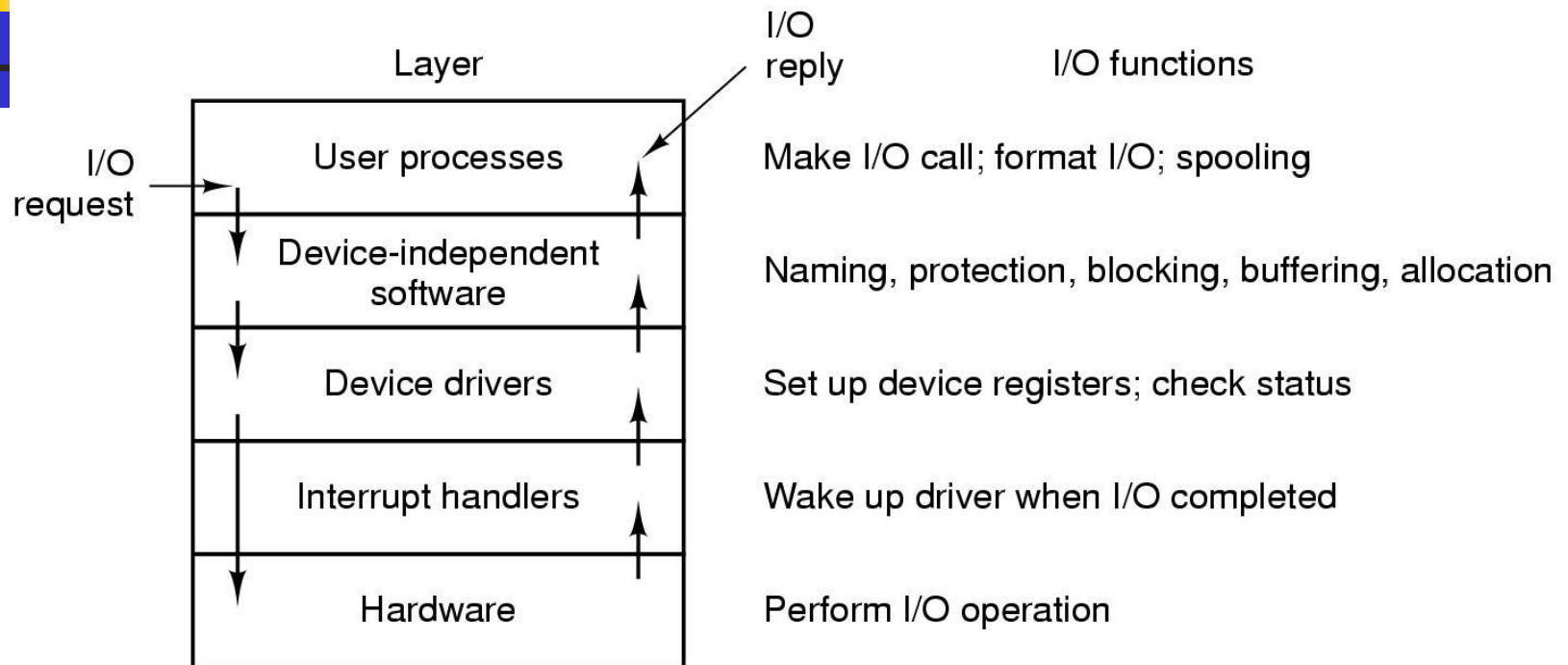


# Review

---

- Phần mềm vào/ra
- 3 cách thực hiện vào/ra
  - Vào ra được lập trình
  - Vào ra dựa trên các ngắt
  - Vào/ra sử dụng DMA
- Trình điều khiển thiết bị
- Lời gọi hệ thống vào/ra không gian người dùng

# I/O Software



Các tầng hệ thống vào/ra và chức năng chính của mỗi tầng



# Tầng độc lập với thiết bị: Báo lỗi

---

- Lỗi vào/ra do lập trình
  - Lỗi này xảy ra khi tiến trình yêu cầu những việc không thể làm được (VD, ghi vào một thiết bị chỉ truyền dữ liệu vào như bàn phím, hoặc đọc từ một thiết bị chỉ truyền dữ liệu ra như máy in).
- Các lỗi vào/ra
  - Lỗi này xảy ra tại tầng thiết bị (VD, đọc một khối dữ liệu của đĩa tại vùng đĩa đã bị hỏng, hoặc đọc từ một card video đang chưa có điện)
- Phần mềm vào/ra độc lập thiết bị phát hiện các lỗi như vậy và báo lên tầng không gian người dùng.





# Cấp phát và giải phóng các thiết bị chuyên dụng

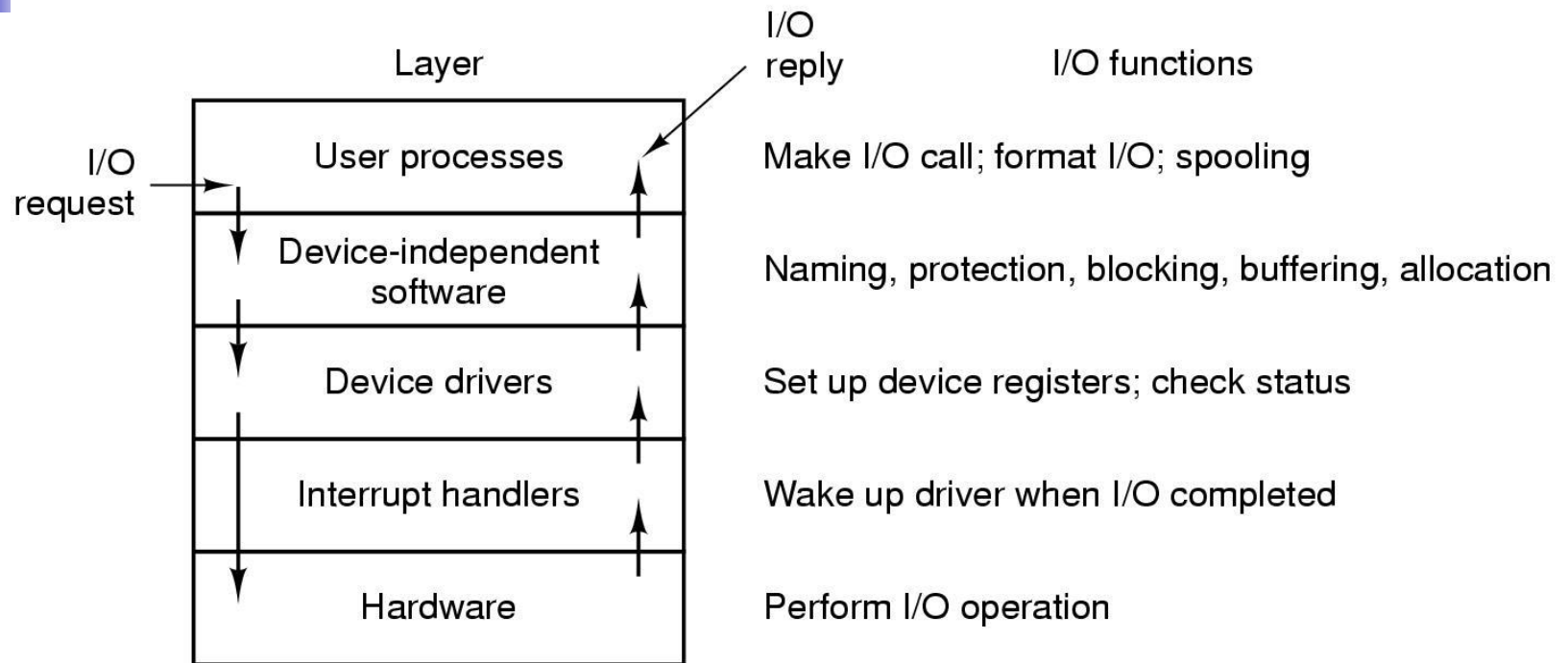
---

- Có một số thiết bị tại một thời điểm chỉ được dùng bởi một tiến trình
  - VD, CD-ROM, một số thiết bị audio
- 2 cách tiếp cận:
  - Thực thi hàm *open()* trực tiếp trên file thiết bị. nếu thiết bị không sẵn sàng, *open()* sẽ trả lại mã lỗi.
  - Một tiến trình yêu cầu một thiết bị. Nếu nó không sẵn sàng, tiến trình bị block, đợi cho đến khi thiết bị sẵn sàng (VD, dựa trên biến semaphore).

# Kích thước khối độc lập với thiết bị

- Các đĩa từ khác nhau có kích thước các cung (sector) khác nhau.
- Phần mềm độc lập với thiết bị ẩn yếu tố này và cung cấp một **kích thước khối thống nhất** cho các tầng cao hơn.
  - Có thể là có cùng kích thước với trang ở trong hệ thống bộ nhớ ảo.

# I/O Software



Layers of the I/O system and the main functions of each layer (typical)



# Tầng phần mềm vào/ra không gian người dùng

---

- Các ứng dụng trong không gian người dùng tạo ra các **lời gọi hệ thống** vào/ra để truy cập các thiết bị vào/ra.
- Lời gọi hệ thống vào ra thường được thực hiện bởi các hàm thư viện.
- Các bước thực thi lời gọi hệ thống vào/ra:
  - Thư viện không gian người dùng gọi phần mềm vào/ra độc lập với thiết bị
  - Phần mềm vào/ra độc lập với thiết bị gọi tới trình điều khiển thiết bị tương ứng
  - Trình đk thiết bị truy cập tới bộ điều khiển thiết bị để điều khiển thiết bị tương ứng.



# Lời gọi hệ thống

---

- Là giao diện giao tiếp giữa ứng dụng đang chạy và HĐH
- Là các macro ngôn ngữ máy hoặc các thủ tục con
- Truyền các tham số qua các thanh ghi, vào bảng nhớ, hoặc vào ngăn xếp
- Unix có khoảng 32 lời gọi hệ thống:
  - `read()`, `write()`, `open()`, `close()`, `fork()`, `exec()`,  
`ioctl()`,



# Đồng hồ và bộ tính thời gian

---

- Đồng hồ (hay còn gọi là bộ tính thời gian)
  - cho biết thời gian hiện tại
  - cho biết thời gian đã qua
  - khởi tạo một đồng hồ bằng cách gọi phép toán X tại thời điểm T
- Bộ tính thời gian có thể lập trình – phần cứng để đo thời gian đã qua và khởi tạo các phép toán
- Đồng hồ được dùng bởi bộ lập lịch (cho việc lập lịch ưu tiên), thời gian trong ngày, đưa ra các cảnh báo, điều khiển



# Đồng hồ

---

- Những đồng hồ thời trước
  - Cần hiệu điện thế 110 hoặc 220 vôn và tạo ra các ngắt mỗi một chu kỳ điện thế, tức là với tần số 50-60 Hz.
- Đồng hồ ngày nay bao gồm 3 thành phần:
  - Bộ tạo dao động
  - bộ đếm
  - thanh ghi
  - Mô tả:
    - Tinh thể thạch anh (trong bộ tạo dao động) sinh ra một tín hiệu định kỳ. Tín hiệu này được đưa vào bộ đếm để đếm cho tới 0. Khi bộ đếm = 0, nó gọi một ngắt CPU. Thanh ghi được dùng để tải lại bộ đếm.



# Đồng hồ có thể lập trình

---

- Các Mode thời gian có thể lập trình:
  - **one-shot mode**: Khi đồng hồ được khởi tạo (bởi phần mềm), nó copy giá trị thanh ghi vào bộ đếm, giảm bộ đếm, tạo ra một ngắt khi bộ đếm = 0, và dừng lại (và nó cần được khởi tạo lại)
  - **square-wave mode**: Khi đồng hồ được khởi tạo, nó copy giá trị thanh ghi vào trong bộ đếm, giảm dần bộ đếm, tạo ra một ngắt khi bộ đếm = 0, và tự động lại copy giá trị thanh ghi vào trong bộ đếm, và toàn bộ tiến trình được **lặp lại**
- Các ngắt định kỳ được gọi là **các nhịp đồng hồ**





# Đồng hồ

---

- Xét đến năng lực của pin đồng hồ
- Một số đồng hồ bị **sai lệch** và phải đồng bộ hóa lại.
  - Đồng bộ hóa đồng hồ “bằng tay” (chỉnh lại).
  - Tự động theo định kỳ đọc thời gian từ một máy tính từ xa được gọi là UTC (Universal Coordinated Time ).
- **NTP** (Network Time Protocol) là một giao thức chuẩn để đồng bộ hóa lại các đồng hồ giữa hai hệ thống mạng



# Q & A

---

- List câu hỏi