



# HỆ ĐIỀU HÀNH

# Operating Systems

1

## NỘI DUNG

---

- Chương 1: Tổng quan
- Chương 2: Quản lý tiến trình
- Chương 3: Deadlock
- Chương 4: Quản lý bộ nhớ
- Chương 5: Hệ thống file
- Chương 6: Quản lý nhập xuất**

---

2

2

## Chương 6

# QUẢN LÝ NHẬP XUẤT

[www.cunghoclaptrinh.com](http://www.cunghoclaptrinh.com)

3

## Nội dung

---

1. Thiết bị I/O
2. Nguyên lý phần cứng I/O
3. Nguyên lý phần mềm I/O
4. Đĩa từ

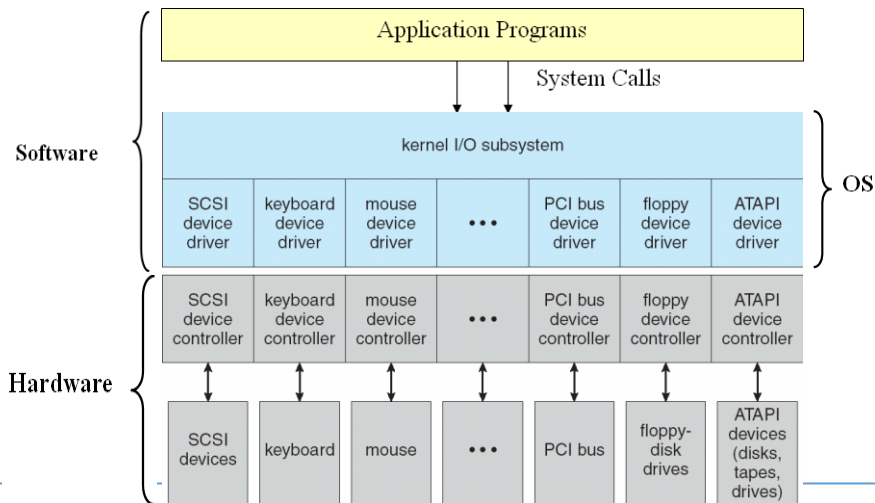
---

4

4

## 6.1. Thiết bị I/O

### • Cấu trúc tổng quát hệ thống nhập xuất



5

5

## Thiết bị I/O (tt)

### • Các loại thiết bị I/O:

- Giao tiếp với người dùng: màn hình, máy in, bàn phím, chuột
- Giao tiếp giữa các thiết bị điện tử: ổ đĩa, USB,...
- Truyền thông: giao tiếp với các thiết bị từ xa: modem

### • Đặc tính của các thiết bị I/O:

- Tốc độ truyền dữ liệu
- Dung lượng lưu trữ, tốc độ truy xuất
- Chức năng
- Đơn vị truyền dữ liệu: khối hay ký tự
- Biểu diễn dữ liệu
- Tình trạng lỗi: nguyên nhân, cách báo lỗi,...

6

6

## 6.2. Nguyên lý phần cứng I/O

- Giao tiếp với thiết bị I/O
- Các thanh ghi I/O
- Kỹ thuật DMA
- Kỹ thuật Interrupt

7

7

### 6.2.1 Giao tiếp với thiết bị I/O

- Cách thức giao tiếp dữ liệu giữa máy tính và thế giới thực: thiết bị I/O thuộc một trong các loại
  - Thiết bị khối: thông tin được đọc/ghi theo từng khối có kích thước cố định (512 - 65,536 bytes) và địa chỉ xác định (đĩa)
  - Thiết bị tuần tự: thông tin được gửi/nhận theo dãy tuần tự các bit, không có địa chỉ (màn hình, bàn phím, máy in, card mạng, chuột)
  - Ngoại lệ:
    - Clock: không có dữ liệu, chỉ kích hoạt tín hiệu ngắt quãng theo chu kỳ định trước
    - Màn hình có nội dung hiển thị nằm trong Ram. Máy tính chỉ cần dùng các lệnh máy truy xuất vùng Ram tương ứng, màn hình sẽ hiển thị ngay lập tức kết quả bị hiệu chỉnh

8

8

## Giao tiếp với thiết bị I/O

- Các thiết bị I/O có tốc độ giao tiếp rất khác nhau tùy vào tính chất sử dụng:

Device	Data rate	Device	Data rate
Keyboard	10 bytes/sec	FireWire 800	100 MB/sec
Mouse	100 bytes/sec	Gigabit Ethernet	125 MB/sec
56K modem	7 KB/sec	SATA 3 disk drive	600 MB/sec
Scanner at 300 dpi	1 MB/sec	USB 3.0	625 MB/sec
Digital camcorder	3.5 MB/sec	SCSI Ultra 5 bus	640 MB/sec
4x Blu-ray disc	18 MB/sec	Single-lane PCIe 3.0 bus	985 MB/sec
802.11n Wireless	37.5 MB/sec	Thunderbolt 2 bus	2.5 GB/sec
USB 2.0	60 MB/sec	SONET OC-768 network	5 GB/sec

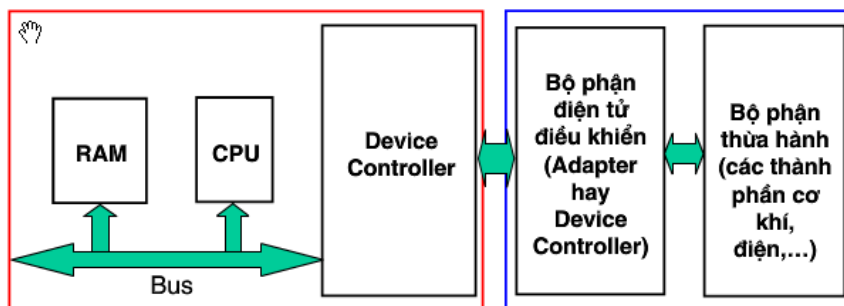
9

9

## Giao tiếp với thiết bị I/O (tt)

### • Device controller:

- Các thiết bị I/O có thể giao tiếp với thiết bị khác
- Để máy tính có thể giao tiếp với thiết bị I/O → sử dụng một bộ phận tương thích với bộ phận điều khiển của thiết bị, gọi là device controller



10

10

## Các thanh ghi I/O

### •Device controller (tt)

- Mỗi device controller đều có những ô nhớ chứa các thông tin hoạt động gọi là các thanh ghi (register). Có 4 loại thanh ghi:
  - **Thanh ghi lệnh:** thanh ghi chứa mã lệnh chức năng mà CPU ghi vào để bắt controller thực hiện. Chiều di chuyển thông tin của thanh ghi này là: CPU → controller (OUT).
  - **Thanh ghi trạng thái:** thanh ghi chứa các bit thông tin về trạng thái hiện hành của thiết bị I/O tương ứng (bận, rảnh,...). Chiều di chuyển thông tin của thanh ghi này là: controller → CPU (IN).
  - **Thanh ghi dữ liệu xuất:** chứa dữ liệu mà CPU muốn xuất ra thiết bị I/O. Chiều di chuyển thông tin của thanh ghi này là: CPU → controller (OUT).
  - **Thanh ghi dữ liệu nhập:** chứa dữ liệu mà thiết bị I/O gửi về máy tính. Chiều di chuyển thông tin của thanh ghi này là: controller → CPU (IN)

11

11

## Các thanh ghi I/O (tt)

### •Device controller (tt)

- Mỗi thanh ghi cần có địa chỉ truy xuất duy nhất.
- Gán địa chỉ cho các thanh ghi của controller:
  - **Địa chỉ I/O**
  - **Địa chỉ memory-mapped I/O**
  - **Địa chỉ hỗn hợp**
  - **Địa chỉ I/O:** có hai loại địa chỉ khác nhau:
    - địa chỉ ô nhớ dành để truy xuất các ô nhớ trong RAM
    - địa chỉ I/O dành truy xuất các thanh ghi của các mạch controller.
    - Ví dụ:
 

```
mov al, [f5]: đọc nội dung ô nhớ RAM ở địa chỉ F5H vào thanh ghi al của CPU
in al, f5: đọc nội dung thanh ghi của controller có địa chỉ (port) là F5H.
```

12

12

## Các thanh ghi I/O (tt)

### •Device controller (tt)

#### •Địa chỉ I/O: có hai loại:

- Địa chỉ ô nhớ dùng để truy xuất các ô nhớ trong RAM
- Địa chỉ I/O dùng để truy xuất các thanh ghi của các mạch controller.
- Ví dụ:

**mov al, [f5]:** đọc nội dung ô nhớ RAM ở địa chỉ F5H vào thanh ghi AL của CPU  
**in al, f5:** đọc nội dung thanh ghi của controller có địa chỉ (port) là F5H.

#### •Địa chỉ memory-mapped I/O:

- Mỗi thanh ghi I/O có một địa chỉ ô nhớ xác định, muốn truy xuất thanh ghi I/O, CPU sẽ dùng lệnh truy xuất ô nhớ như bình thường.
- Ví dụ:  
**mov al, [f5]:** đọc nội dung thanh ghi I/O ở địa chỉ F5H tùy thuộc vào địa chỉ này đang được dùng cho phần tử nào.

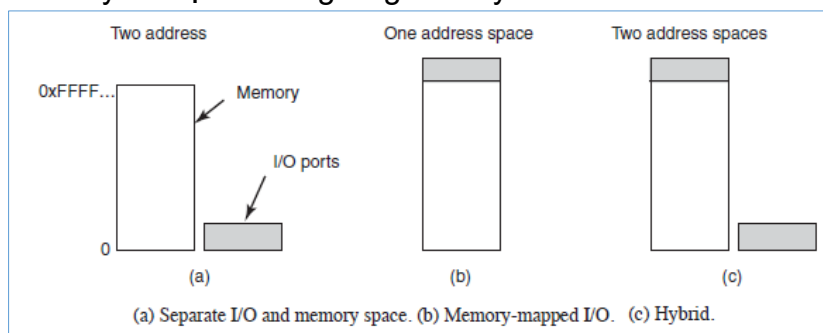
13

13

## Các thanh ghi I/O (tt)

### •Địa chỉ hỗn hợp (hybrid): có 2 loại địa chỉ:

- Địa chỉ ô nhớ dành để truy xuất các ô nhớ trong RAM
- Địa chỉ I/O dành truy xuất các thanh ghi của các mạch controller.
- Tùy theo tính chất sử dụng của từng thanh ghi → sử dụng địa chỉ I/O (port) hay địa chỉ memory với lệnh tương ứng để truy xuất



14

14

## Các cơ chế Nhập/Xuất

---

- Polling (kiểm soát vòng)
- DMA (Direct Memory Access)
- Interrupt Driven (ngắt)

15

15

## Polling

---

- Bộ xử lý liên tục kiểm tra bit busy cho đến khi bit này bị xóa, gọi là quá trình Waiting hay Polling
- Bộ xử lý bật bit write trong thanh ghi điều khiển và ghi một byte vào thanh ghi dữ liệu ra
- Bộ xử lý bật bit command-ready
- Khi bộ điều khiển nhận ra bit command-ready được bật, nó bật bit busy
- Bộ điều khiển đọc yêu cầu trong thanh ghi điều khiển, dữ liệu trong thanh ghi dữ liệu ra và thực thi yêu cầu nhập xuất
- Bộ điều khiển xóa bit command-ready, xóa bit error trong thanh ghi trạng thái (thực thi thành công), và xóa bit busy để kết thúc một yêu cầu nhập xuất

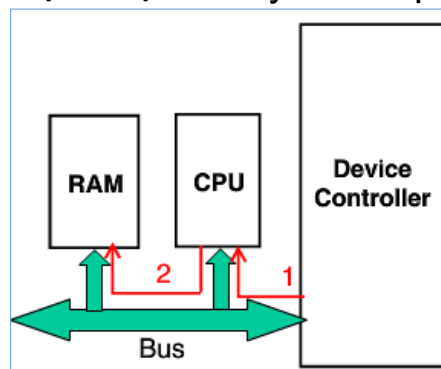
16

16



## Kỹ thuật DMA (Direct Memory Access)

- Việc chuyển thông tin giữa thiết bị I/O với bộ nhớ máy tính được thực hiện thông qua CPU theo cơ chế tuần tự.
- Mỗi lần cần di chuyển dữ liệu từ thiết bị I/O (đang nằm trong thanh ghi data in của controller), CPU phải thực hiện 2 lệnh máy liên tiếp:
  - Lệnh in<sub>2</sub> (hay mov) để di chuyển dữ liệu từ controller vào thanh ghi của CPU.
  - Lệnh mov để di chuyển dữ liệu từ thanh ghi CPU ra ô nhớ RAM xác định.
- Hạn chế: không hiệu quả
  - Tốn nhiều chu kỳ để di chuyển thông tin
  - Thông tin phải đi qua trung gian là CPU
- Kỹ thuật DMA



17

17

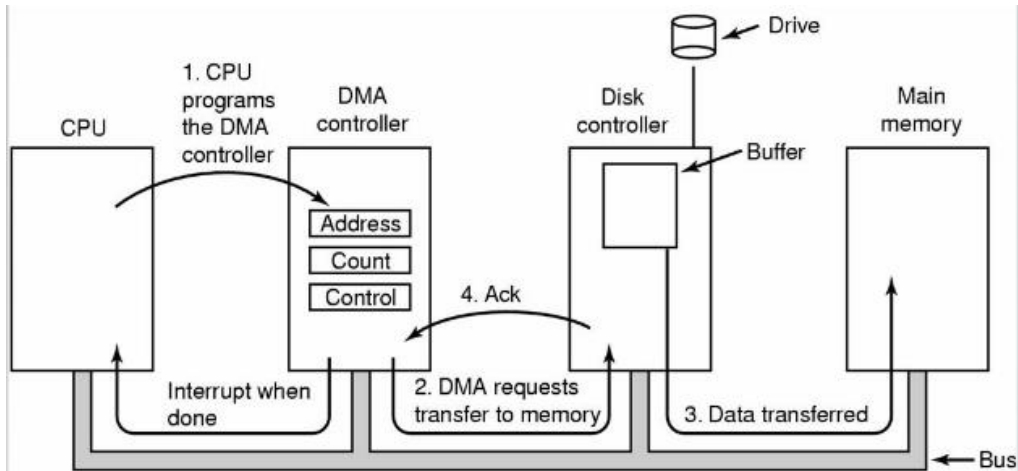
## Kỹ thuật DMA (tt)

- DMA là một vi mạch có chức năng đặc biệt, cho phép chuyển dữ liệu trực tiếp giữa I/O và RAM mà không cần đi ngang qua CPU.
- CPU khởi động các thanh ghi của DMA, các thanh ghi này chứa địa chỉ ô nhớ và số byte cần chuyển.
- DMA chuyển số liệu và khi chấm dứt thì trả quyền điều khiển cho CPU: Bộ điều khiển DMA báo (ngắt) CPU khi hoàn thành công việc theo từng block dữ liệu.

18

18

## Kỹ thuật DMA (tt)

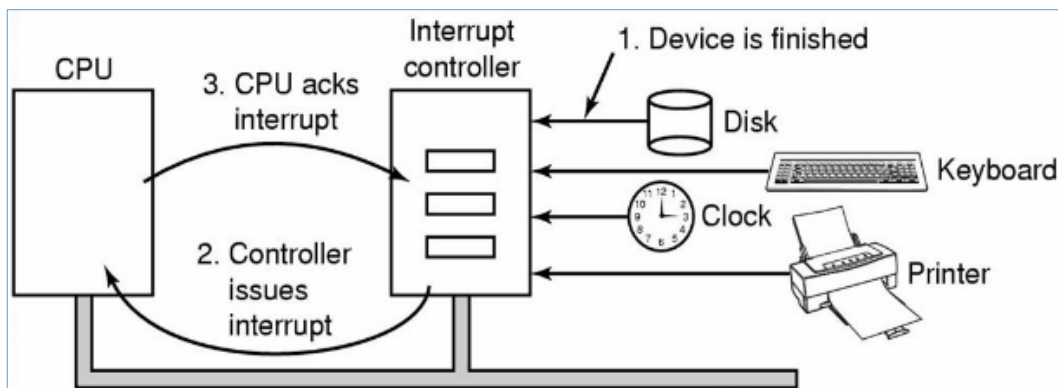


19

19

## Kỹ thuật Interrupt (ngắt quãng)

- Bộ điều khiển thiết bị sẽ báo (ngắt) CPU khi hoàn thành công việc theo từng đơn vị dữ liệu → CPU không phải chờ,



20

20

## 6.3. Nguyên lý phần mềm I/O

- **Độc lập thiết bị:** ứng dụng có thể truy xuất bất kỳ thiết bị nào, qui trình truy xuất phải đồng nhất, không phụ thuộc vào tính chất vật lý hay loại thiết bị.
- **Đặt tên thiết bị đồng nhất:** tên thiết bị là một chuỗi hay một số nguyên giống như tên một file.
- **Che dấu việc xử lý lỗi:** nếu có lỗi trong khi truy xuất I/O, hệ thống phần mềm I/O phải xử lý tốt nhất và ở cấp thấp nhất rồi cố gắng che dấu lỗi càng nhiều càng tốt

21

21

## Nguyên lý phần mềm I/O (tt)

- **Chuyển đổi cung cách nhập/xuất dữ liệu dạng bất đồng bộ thành dạng đồng bộ:**
  - Ở cấp vật lý: mỗi lần cần đọc một sector đĩa, CPU sẽ xuất lệnh đọc ra controller, chờ ngắt khi có dữ liệu, đọc khối dữ liệu vào bộ nhớ.
  - Ở cấp ứng dụng: nên cung cấp một hàm chức năng, ứng dụng gọi hàm này → trả về kết quả → ứng dụng có dữ liệu để xử lý tiếp.
- **Đệm dữ liệu nhập/xuất:**
  - Dữ liệu từ một thiết bị I/O đưa vào máy tính cần được cache lại để cung cấp cho ứng dụng khi có yêu cầu

22

22

## Nguyên lý phần mềm I/O (tt)

---

- **Thiết bị dùng chung (shared)**

- Tại mỗi thời điểm có thể phục vụ cho nhiều ứng dụng đồng thời

- **Thiết bị dùng riêng (dedicated):**

- Chỉ phục vụ một ứng dụng tại mỗi thời điểm

23

23

## Thực hiện nhập/xuất

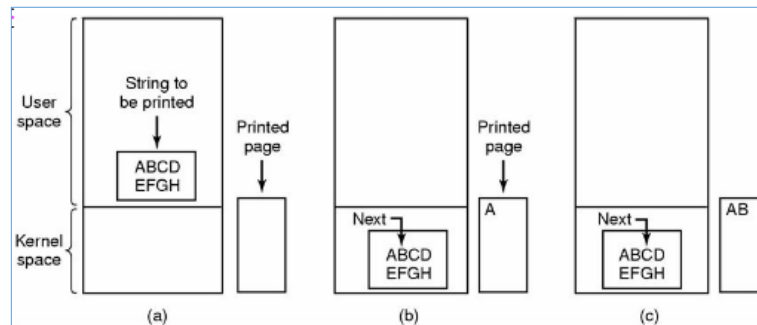
---

- Lập trình thực hiện I/O (programmed I/O)
- Dùng cơ chế ngắt (interrupt)
- Dùng kỹ thuật DMA (DMA)

24

24

## Lập trình thực hiện I/O



```

copy_from_user(buffer, p, count);          /* p is the kernel bufer */
for (i = 0; i < count; i++) {              /* loop on every character */
    while (*printer_status_reg != READY);   /* loop until ready */
    *printer_data_register = p[i];          /* output one character */
}
return_to_user();

```

25

25

## Dùng cơ chế ngắt

- Ghi một chuỗi lên máy in
  - (a) Code executed at the time the print system call is made.
  - (b) Interrupt service procedure for the printer.

```

copy_from_user(buffer, p, count);
enable_interrupts();
while (*printer_status_reg != READY);
*printer_data_register = p[0];
scheduler();

```

I

(a)

```

if (count == 0) {
    unblock_user();
} else {
    *printer_data_register = p[i];
    count = count - 1;
    i = i + 1;
}
acknowledge_interrupt();
return_from_interrupt();

```

(b)

26

26

## Dùng kỹ thuật DMA (DMA)

- Ngắt → lãng phí một lượng thời gian CPU → sử dụng DMA.
- Ý tưởng: bộ điều khiển DMA nạp các ký tự vào máy in tại một thời điểm, mà không cần CPU.
- DMA được lập trình I / O để thực hiện tất cả công việc

```
copy_from_user(buffer, p, count);
set_up_DMA_controller();
scheduler();
```

(a) Code executed when the print system call is made.

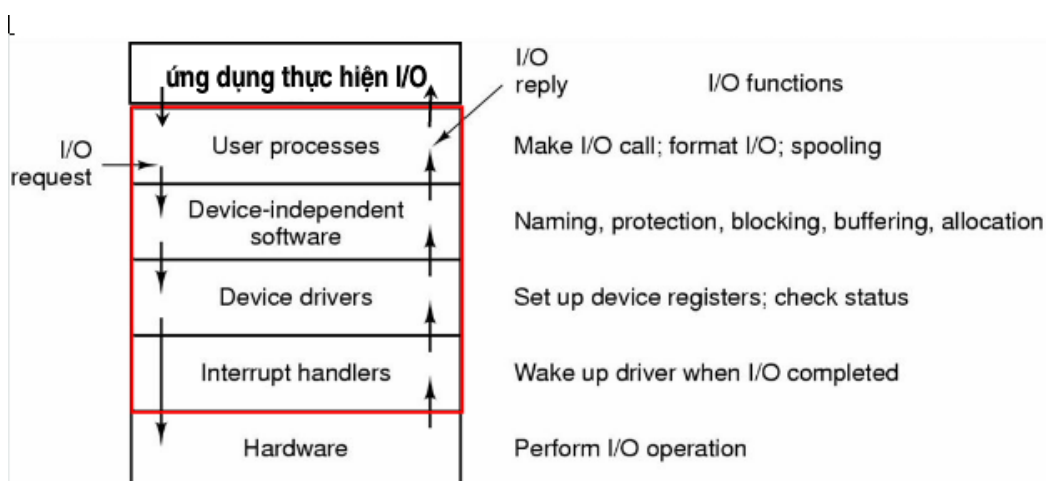
```
acknowledge_interrupt();
unblock_user();
return_from_interrupt();
```

(b) Interrupt-service procedure.

27

27

## Các cấp chức năng cơ bản của hệ thống phần mềm I/O

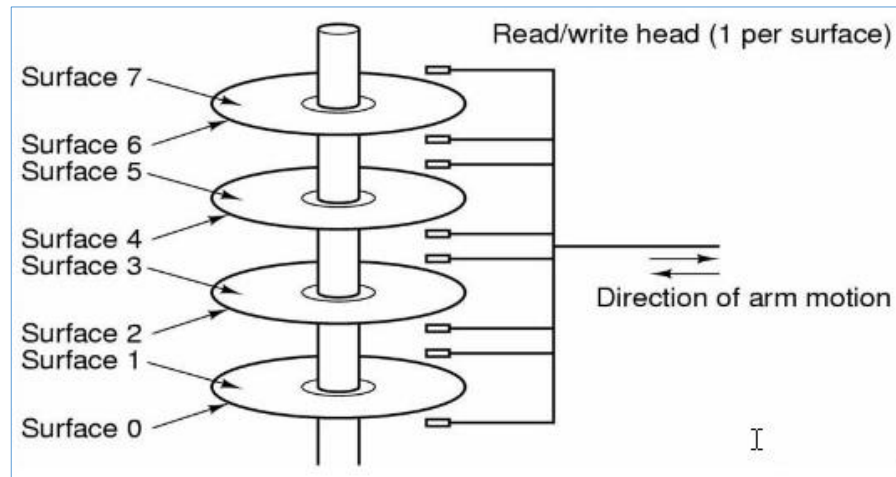


28

28

## 6.4. Đĩa từ

### • Cấu trúc đĩa

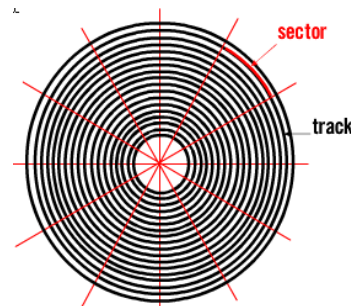


29

29

## Đĩa từ (tt)

- Đĩa vật lý là không gian dữ liệu 3 chiều, gồm nhiều cylinder
- Mỗi cylinder gồm nhiều track
- Mỗi track chứa nhiều sector.
- Sector là đơn vị truy xuất tin nhỏ nhất ở cấp vật lý (từ ngoài ta không thể truy xuất từng byte dữ liệu trên disk được).
- Muốn truy xuất một sector, cần xác định tọa độ 3 chiều của nó (C,H,S) → khó.



30

30

## RAID

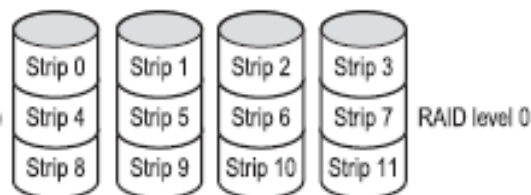
- **Đĩa SLED (Single Large Expensive Disk):**
  - Tốc độ truy xuất chậm
  - Độ tin cậy thấp
  - → khắc phục bằng kỹ thuật RAID (Hệ thống đĩa dự phòng - Redundant Array of Inexpensive Disks)
- **RAID:**
  - Kết nối một dãy các ổ cứng có chi phí thấp lại với nhau để hình thành một thiết bị nhớ đơn có dung lượng lớn.
  - Ưu điểm của RAID
    - Dự phòng
    - Hiệu quả cao
    - Giá thành thấp

31

31

## Đĩa cứng RAID 0

- Dùng kỹ thuật striping: gộp nhiều sector thành một strip, phân bổ trên nhiều đĩa → đĩa luận lý như sau:
  - phân bổ các strip theo kiểu round-robin.
  - Nếu cần truy xuất khối dữ liệu dài n strip, controller trên RAID0 sẽ biết được khối dữ liệu n strip này nằm trên n đĩa khác nhau → mỗi đĩa truy xuất 1 strip.
  - Dữ liệu truy xuất được sẽ được hợp lại/phân ra bởi controller



32

32



## Đĩa cứng RAID 0 (tt)

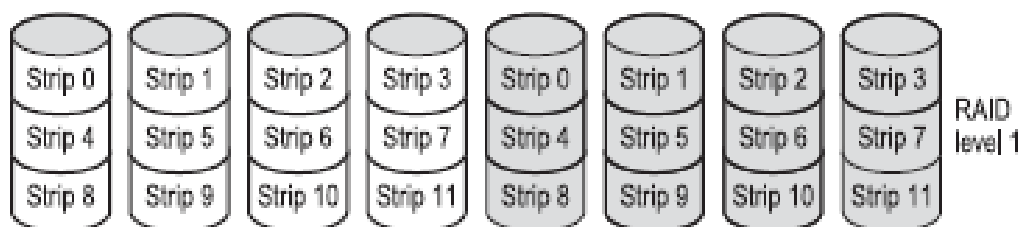
- Ưu điểm:
  - Tăng tốc độ truy xuất lên  $n$  lần
  - Tăng hiệu quả lưu trữ.
  - Không làm mất dung lượng dữ liệu.
- Nhược điểm
  - Không có ổ dự phòng
  - Giảm độ tin cậy  $n$  lần.

33

33

## Đĩa cứng RAID 1

- Dùng kỹ thuật Master/Mirror: nhân bản dự phòng
- Mỗi đĩa có sẵn (master), cần thêm một đĩa dự phòng (slave).
- Ghi dữ liệu: controller sẽ ghi dữ liệu đồng thời lên cả master lẫn slave.



34

34

## Đĩa cứng RAID 1 (tt)

- Đọc dữ liệu: controller yêu cầu master thực hiện.
- Trường hợp master trục trặc, controller đổi vai trò của master và slave → thực hiện lại việc đọc dữ liệu.
- Phục hồi: copy lại dữ liệu.
- Ưu điểm: khả năng dự phòng dữ liệu.
- Nhược điểm:
  - Sử dụng nhiều đĩa cứng → hao phí
  - Không tăng hiệu suất thực thi.
  - Tốn thời gian thay đổi ổ hoạt động khi có sự cố.

35

35

## Đĩa cứng RAID 2

- Để lưu 32 bit dữ liệu, controller sẽ đổi 32 bit dữ liệu thành mã sửa sai Hamming 39 bit, từng bit sẽ được ghi lên một đĩa khác nhau.
- Tốc độ: tăng lên 32 lần.
- Độ tin cậy: nếu có lỗi ở một disk, nhờ mã Hamming, controller sẽ tự sửa sai để xác định 32 bit dữ liệu gốc



36

36

## Đĩa cứng RAID 3

- Tương tự RAID 0, dùng kỹ thuật stripping theo từng byte: chia file thành từng byte ghi trên từng đĩa
- Sử dụng thêm 1 đĩa chứa bit phát hiện lỗi (parity).
- Tốc độ truy xuất tập tin kích thước nhỏ rất nhanh
- Độ tin cậy: nếu có lỗi ở một đĩa, nhờ bit parity phát hiện lỗi + tín hiệu báo lỗi ở đĩa thứ i, controller sẽ tự sửa sai bit i để xác định 32 bit dữ liệu gốc.

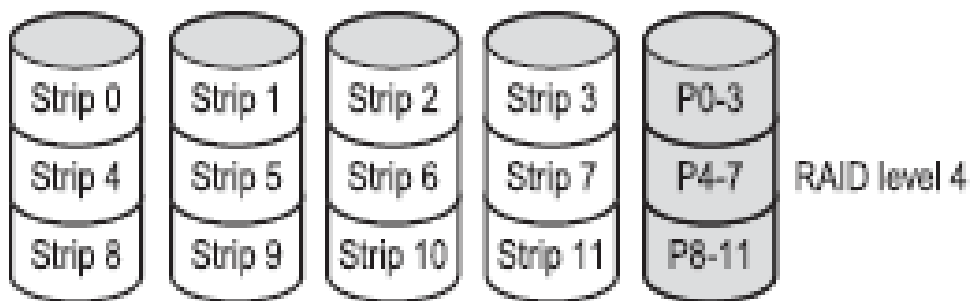


37

37

## Đĩa cứng RAID 4

- Dữ liệu ghi trên n đĩa + 1 đĩa chứa bit phát hiện lỗi (parity).
- Độ tin cậy: nếu có lỗi ở một đĩa, nhờ bit parity phát hiện lỗi + tín hiệu báo lỗi ở đĩa thứ i.

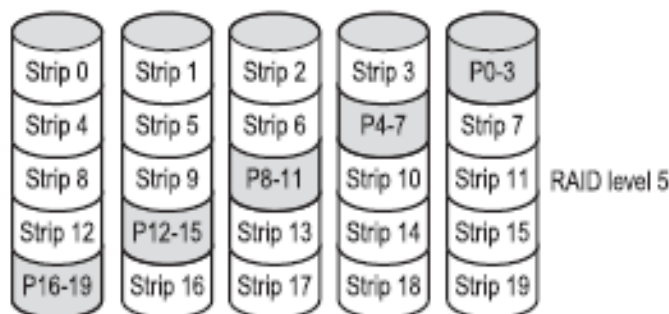


38

38

## Đĩa cứng RAID 5

- Sử dụng kỹ thuật striping theo từng Block và phân chia thông tin “parity” (chẵn lẻ)
- Cần ít nhất 3 đĩa
- Dữ liệu ghi vào  $n$  đĩa + 1 đĩa ghi bit parity dự phòng theo kiểu luân phiên

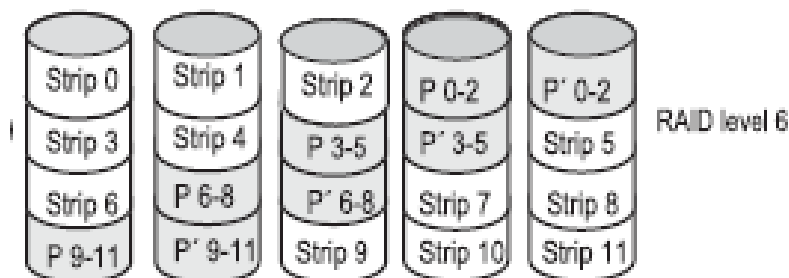


39

39

## Đĩa cứng RAID 6

- Phát triển từ RAID 5
- Cần ít nhất 4 đĩa
- Dữ liệu ghi vào  $n$  đĩa + 2 đĩa ghi bit parity dự phòng chẵn/lẻ độc lập với nhau theo kiểu luân phiên

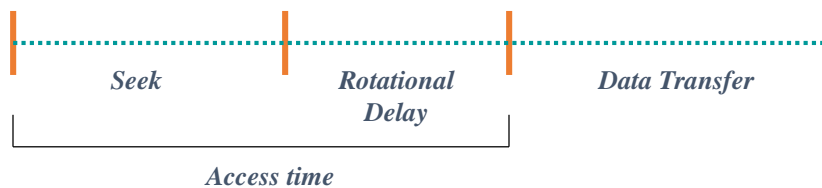


40

40

## Các tham số hiệu suất đĩa

- **Seek time**: thời gian di chuyển đầu đọc tới track hay cylinder thích hợp
- **Rotational delay** (rotational latency): thời gian quay đĩa đến sector cần đọc



41

## Công thức tính hiệu suất đĩa

- **Seek time** ( $T_s$ ) =  $m * n + s$   
 trong đó:  $m$  = hệ số đặc trưng của đĩa  
 $n$  = số track đi qua  
 $s$  = thời gian khởi động đầu từ
- **Rotational delay** ( $T_r$ ) =  $1 / (2 * r)$   
 trong đó:  $r$  là tốc độ quay (vòng /s)  
 • Thời gian quay của các từ 3600 rpm đến 15000 rpm (có thời gian rotational delay trung bình là 2 ms)
- **Transfer time** ( $T_t$ ) =  $b / (r * N)$   
 trong đó:  $b$  = số byte dữ liệu cần chuyển  
 $N$  = số byte trên 1 track
- **Average access time** ( $T_a$ ) =  $T_s + T_r + T_t$

42

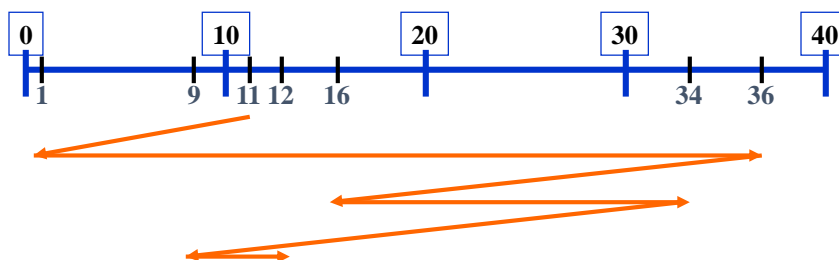
## Thuật toán định thời đĩa

- Xác định thứ tự di chuyển giữa các track trên đĩa một cách tối ưu
- Các thuật toán điển hình:
  - FIFO
  - SSF (Shortest seek first)
  - SCAN (Elevator algorithm)
  - C-SCAN (One-way elevator)
  - ...

43

## First in, First out (FIFO)

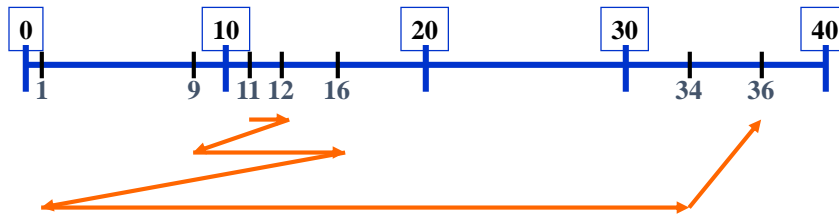
- Di chuyển theo thứ tự truy xuất, mỗi lần nhận 1 thao tác riêng lẻ
- Ví dụ: truy xuất 1, 36, 16, 34, 9, 12 và vị trí hiện tại ở Cylinder 11 (Di chuyển trung bình = 18.5 cylinders)



44

## Shortest Seek First (SSF)

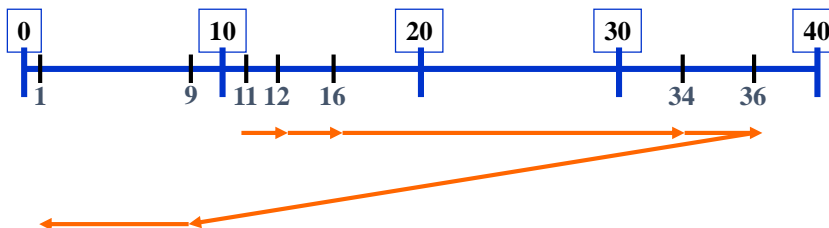
- Truy xuất nào có di chuyển ít nhất thì thực hiện trước
- Ví dụ: 1, 36, 16, 34, 9, 12 và vị trí hiện tại ở Cylinder 11 : Di chuyển trung bình = 10.2 cylinders



45

## SCAN (Elevator Algorithm)

- Đầu đọc di chuyển theo 1 chiều Khi đến cylinder cuối cùng, mới quay lại
- Ví dụ: truy xuất 1, 36, 16, 34, 9, 12 và vị trí hiện tại ở Cylinder 11 (Di chuyển trung bình = cylinders)
  - → 11, 12, 34, 36 end 9, 1



46

46

## C-SCAN (One-way Elevator)

- Giống SCAN nhưng chỉ đi theo 1 hướng
- Khi đến cylinder cuối cùng, đầu đọc quay về vị trí xuất phát
- Ví dụ: truy xuất 1, 36, 16, 34, 9, 12 và vị trí hiện tại ở Cylinder 11 (Di chuyển trung bình = cylinders)
  - → 11, 12, 34, 36 end 0, 1, 9

47

## LOOK

- Kết hợp giữa SCAN và C-SCAN
- Quét đến khối cuối rồi quay lại
- Ví dụ: truy xuất 1, 36, 16, 34, 9, 12 và vị trí hiện tại ở Cylinder 11 (Di chuyển trung bình = cylinders)
  - → 11, 12, 34, 36 , 9, 1

48

48



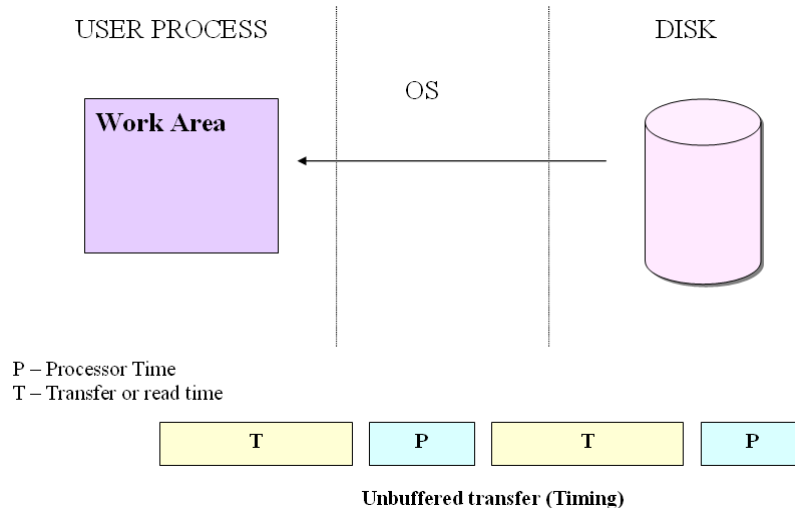
## Đệm dữ liệu (Buffering)

- Lưu tạm dữ liệu trong bộ nhớ khi truyền giữa các thiết bị để tối ưu:
  - Tốc độ truyền giữa các thiết bị
  - Ích thước dữ liệu truyền giữa các thiết bị

49

49

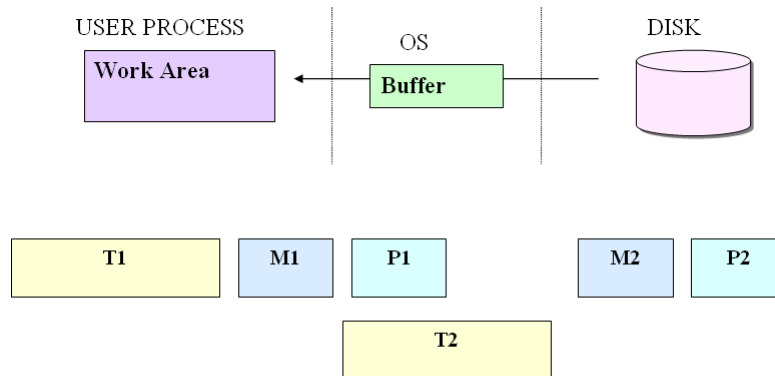
## Không sử dụng vùng đệm dữ liệu



50

50

## Sử dụng vùng đệm dữ liệu



Assuming M (Move time) to be less than P, this is an improvement over unbuffered transfer.

-Buffered transfer (Timing)

51

51

## Đệm dữ liệu truy xuất nhanh (Caching)

- Lưu trữ bản sao của dữ liệu tại một vùng nhớ truy xuất nhanh
- Cần phân biệt cache và buffer
  - Buffer lưu tạm bản sao của dữ liệu
  - Cache lưu tạm bản sao của dữ liệu tại nơi có tốc độ truy xuất nhanh hơn nơi lưu dữ liệu

52

52

## Spooling

- Đặc điểm:
  - Lưu trữ tạm dữ liệu đầu ra cho một thiết bị nếu thiết bị chỉ có thể phục vụ một yêu cầu tại một thời điểm, như máy in
  - Chỉ chấp nhận dòng dữ liệu liên tục
- Thao tác in:
  - Dữ liệu cần in của mỗi ứng dụng được đưa vào một tập tin riêng
  - Hệ thống spooling sẽ chuyển lần lượt các tập tin này cho máy in

53

53

## Quản lý lỗi (Error Handling)

- Lỗi thường ở 2 dạng:
  - Tạm thời
  - Lâu dài
- I/O Subsystem có thể phục hồi hiệu quả lỗi tạm thời
- Khi yêu cầu nhập xuất xảy ra lỗi → trả về mã lỗi

54

54