



TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

QUẢN TRỊ DỮ LIỆU - CT467

Chương 2: **LƯU TRỮ VÀ
CẤU TRÚC TẬP TIN**

Biên soạn:



Ths. Nguyễn Thị Kim Yến



Ntkyen@ctu.edu.vn



MỤC TIÊU CỦA CHƯƠNG 2

- Biết được các loại thiết bị lưu trữ và các đặc điểm cơ bản của chúng
- Hiểu được cơ chế hoạt động của đĩa từ, công nghệ RAID và các mức RAID cơ bản.
- Hiểu được phương pháp tổ chức tập tin và tổ chức các mẫu tin trong tập tin.
- Hiểu được các phương năng cao hiệu năng truy xuất dữ liệu như chỉ mục và băm



CANTHO UNIVERSITY

NỘI DUNG

1

Các loại thiết bị lưu trữ

2

Đĩa từ và RAID

3

Tổ chức tập tin

4

Tổ chức mẫu tin trong tập tin

5

Quản lý bộ đệm

6

Chỉ mục và băm

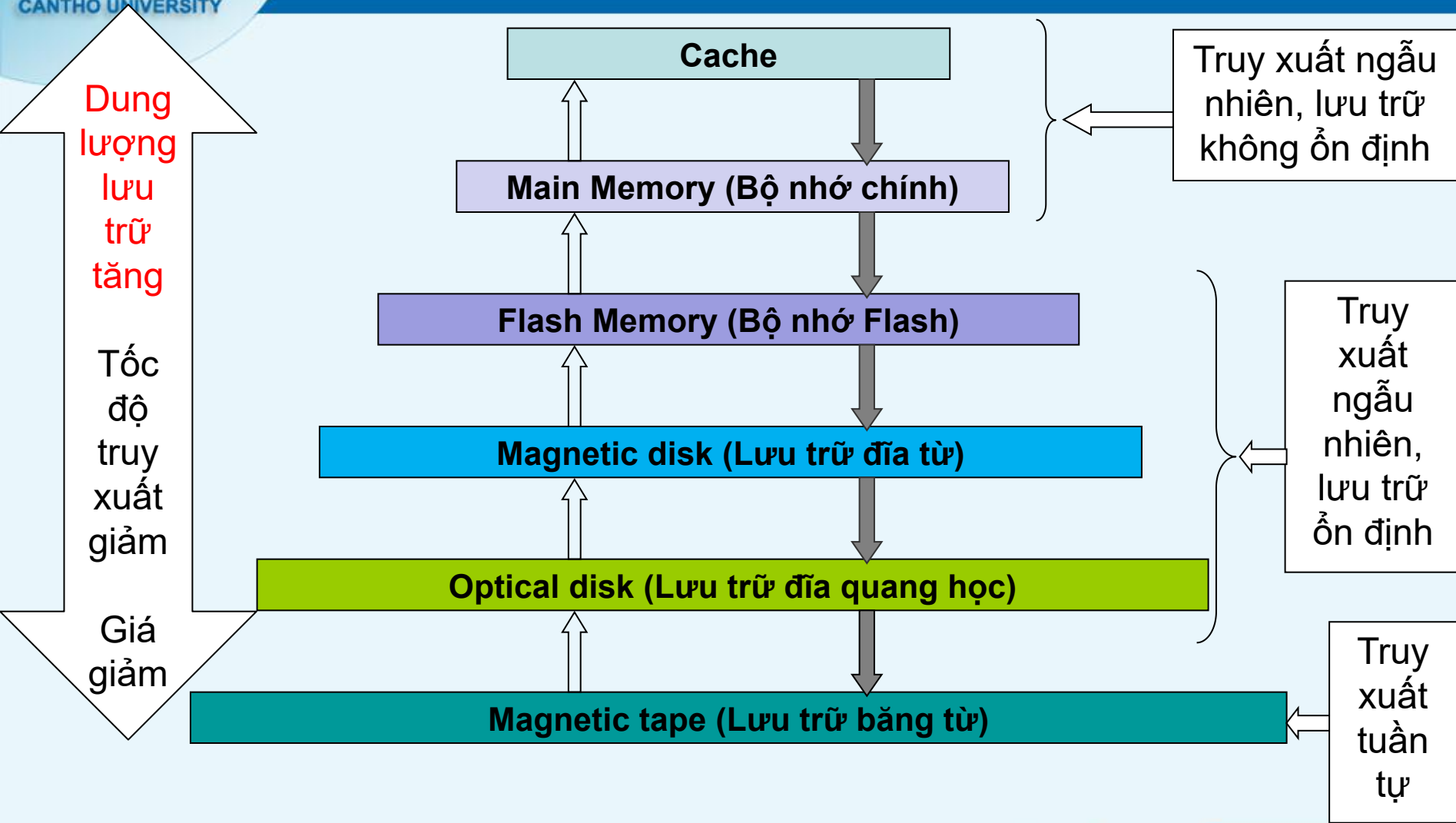
7

Xử lý vấn tin

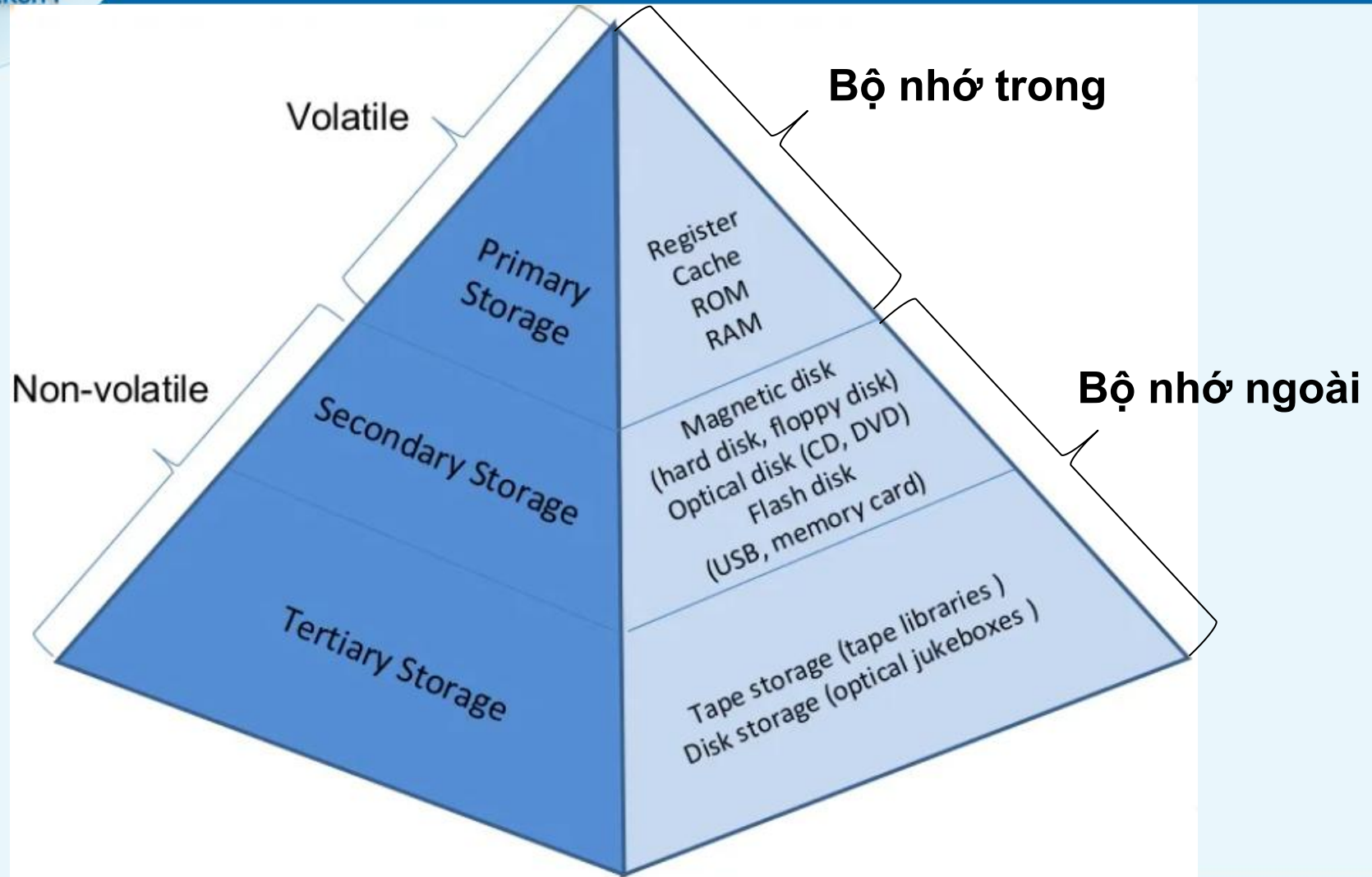


CANTHO UNIVERSITY

1. Các loại thiết bị lưu trữ



1. Các loại thiết bị lưu trữ (tt)





CANTHO UNIVERSITY

2. Đĩa từ và RAID

2.1 Đĩa từ

- **Đĩa từ:** có dạng hình tròn, được làm nhôm, hợp chất gốm và thủy tinh, 2 mặt đĩa được phủ lớp từ tính và lớp bảo vệ. Các đĩa được xếp chồng và gắn trên cùng 1 trục.



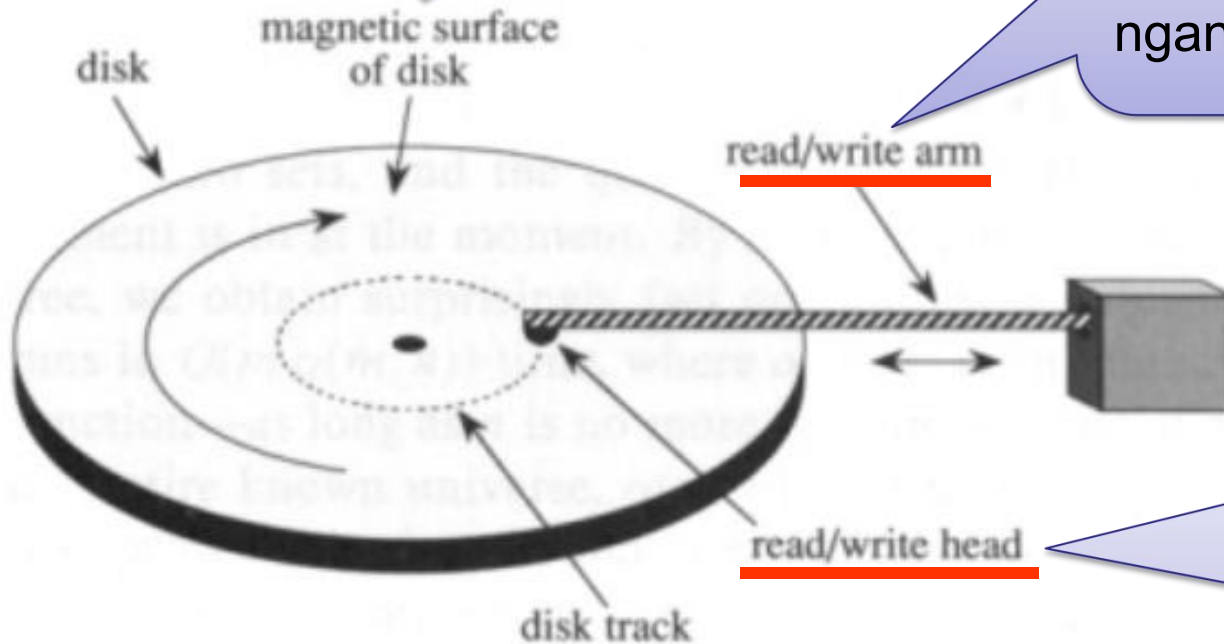


2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Cấu trúc vật lý

Bề mặt từ tính của đĩa

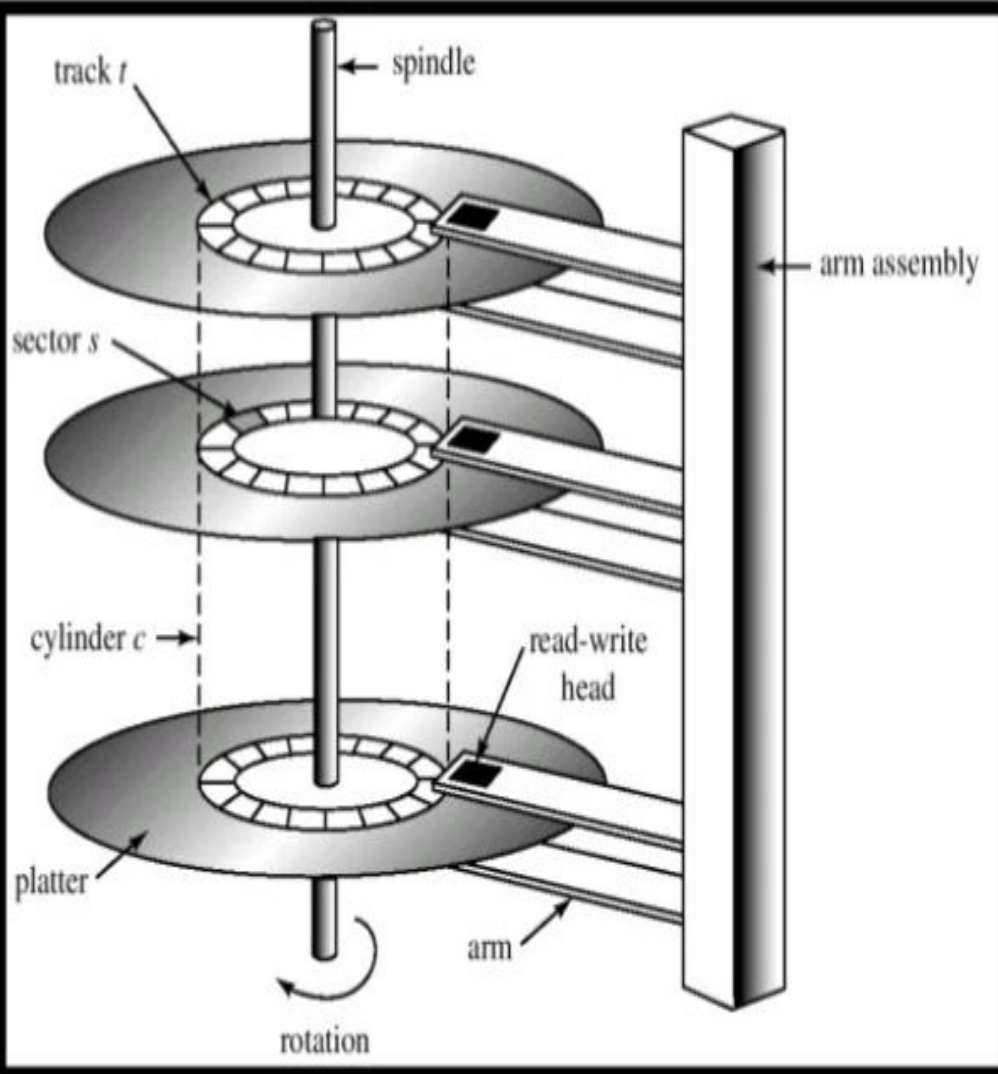
Cần di chuyển đầu đọc:
Giúp các đầu từ dịch chuyển ngang trên bề mặt đĩa



Đầu đọc/ghi dữ liệu:
Mỗi mặt đĩa có đầu đọc riêng

2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Cấu trúc vật lý

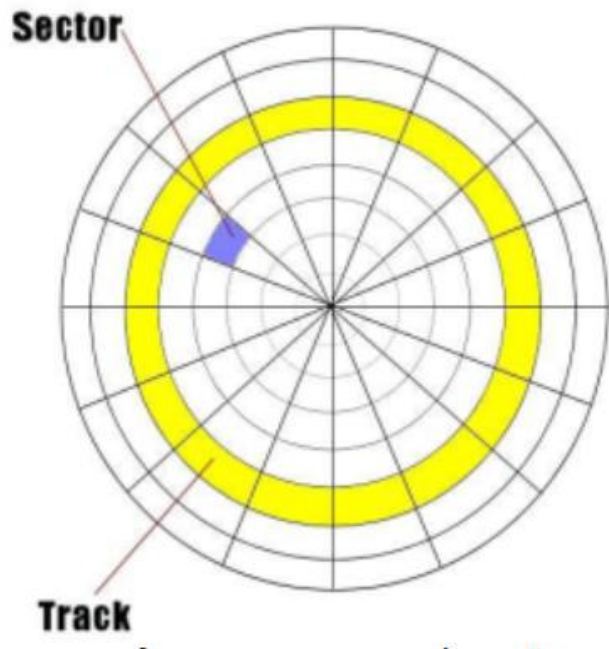
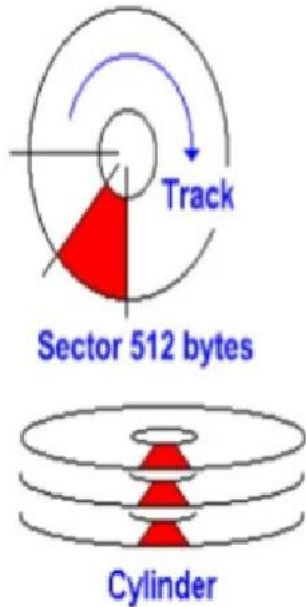


- **Spindle** (trục quay): truyền chuyển động của đĩa từ
- **Platter** (đĩa): Mỗi đĩa có thể sử dụng hai mặt
- **Động cơ**: Được gắn đồng trục với trục quay và các đĩa

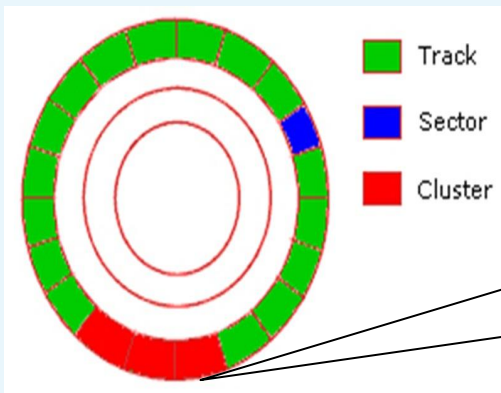


2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Cấu trúc luận lý



- **Track** (rãnh): các vòng tròn đồng tâm trên bề mặt đĩa
- **Sector** (cung): phần tử trên track, mỗi sector có 512 byte => chứa DL
- **Cylinder** (mặt trục): tập hợp những track đồng tâm của tất cả các lá đĩa

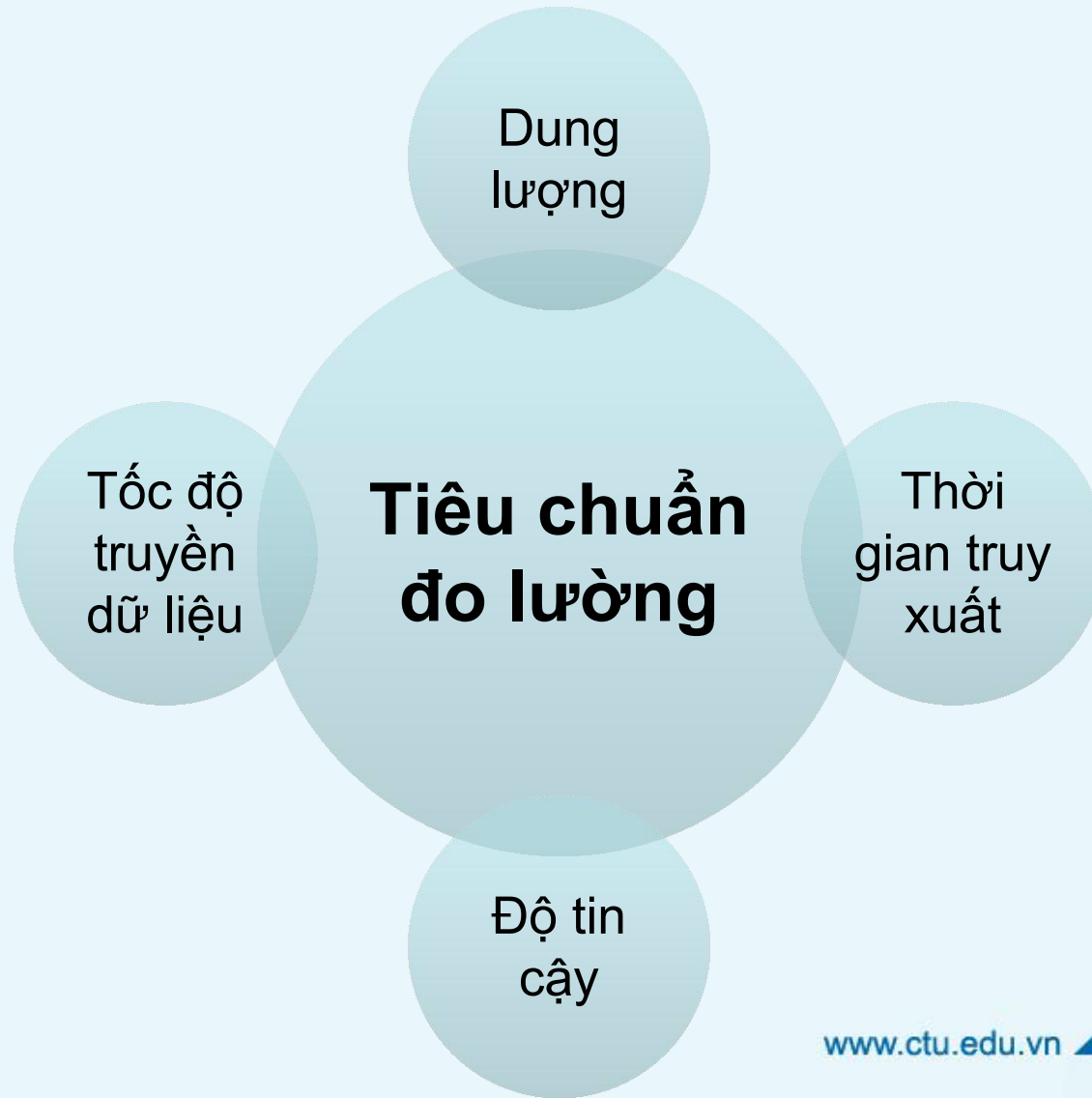


Cluster: Tập hợp 1 hoặc nhiều sector



2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Đo lường hiệu năng



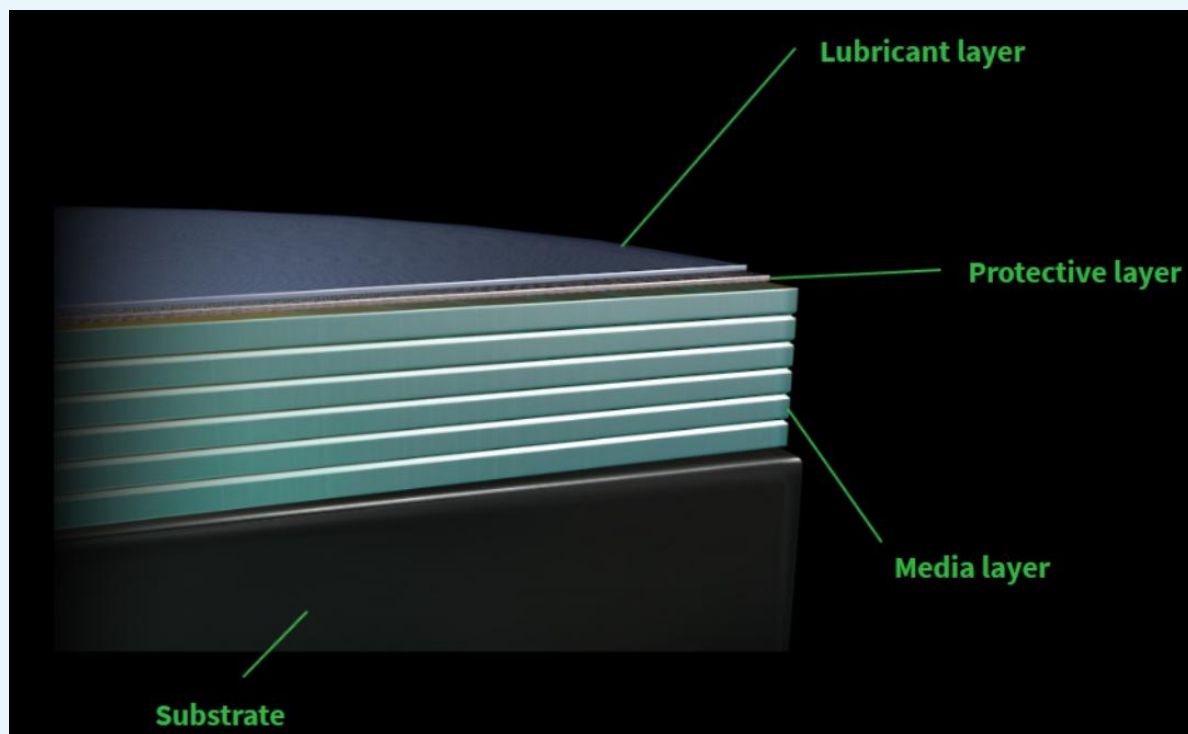


CANTHO UNIVERSITY

2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Đo lường hiệu năng

❖ **Dung lượng đĩa** từ phụ thuộc mật độ tích hợp của các phần tử từ tính trên bề mặt đĩa



- Lớp bôi trơn
- Lớp bảo vệ
- **Lớp lưu trữ**
- Lớp nền



2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Đo lường hiệu năng

❖ **Thời gian truy xuất:** khoảng thời gian từ khi yêu cầu đọc/viết được phát đi đến khi bắt đầu truyền dữ liệu.

TG tìm kiếm

- Di chuyển đầu đọc để định vị đúng track
 - 2...30ms
- Xấu 1/3 TGTK

TG trễ quay

- Từ khi đầu đọc được định vị lên track cho đến khi tìm được sector cần đọc
 - 1/2 vòng quay

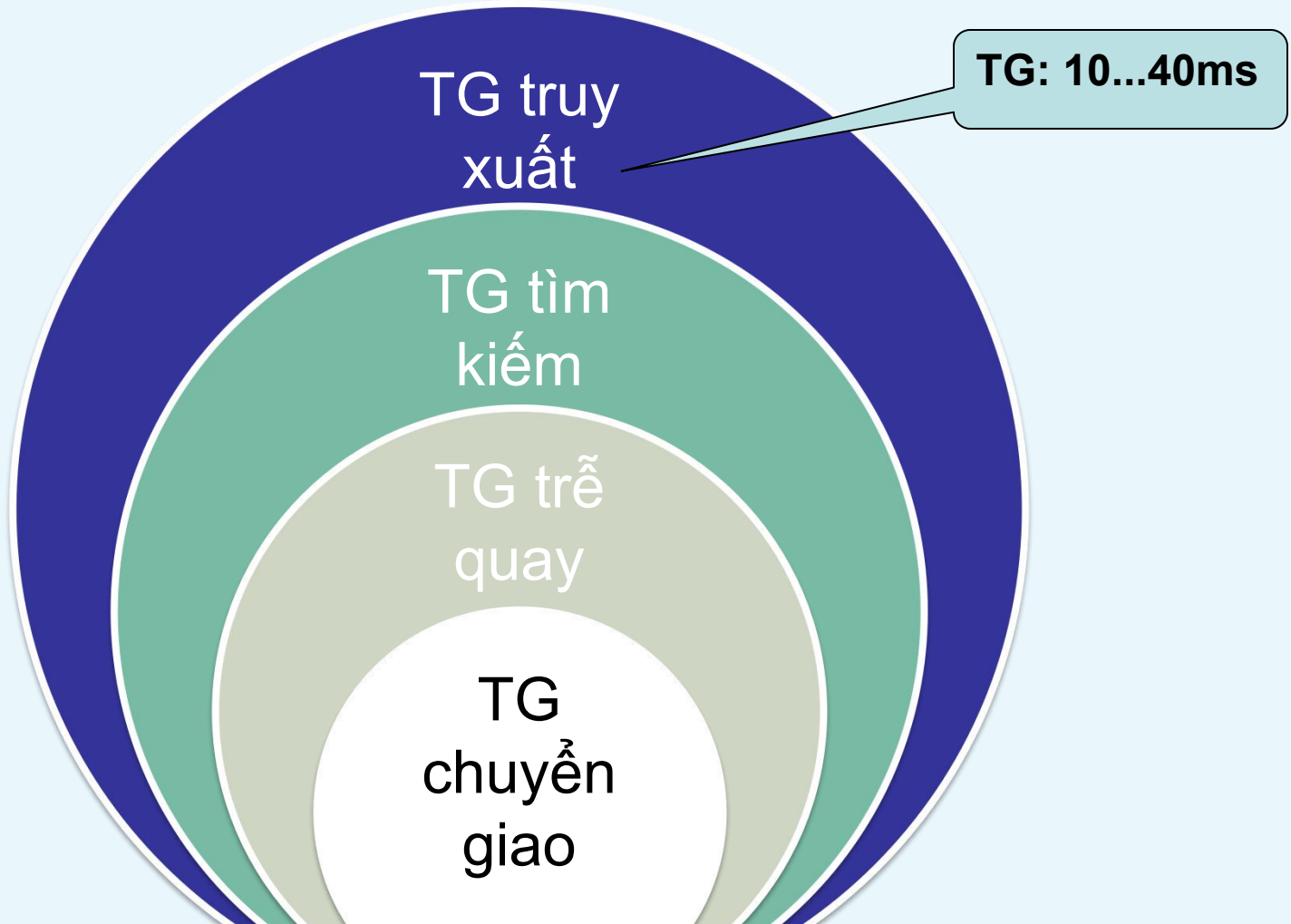
TG chuyển giao

- Chuyển DL từ đĩa vào bộ nhớ hoặc ngược lại
 - 60...120 vòng/giây



2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Đo lường hiệu năng



$$\text{TG truy xuất} = \text{TG tìm kiếm} + \text{TG trễ quay} + \text{TG chuyển giao}$$



2. Đĩa từ và RAID (tt)

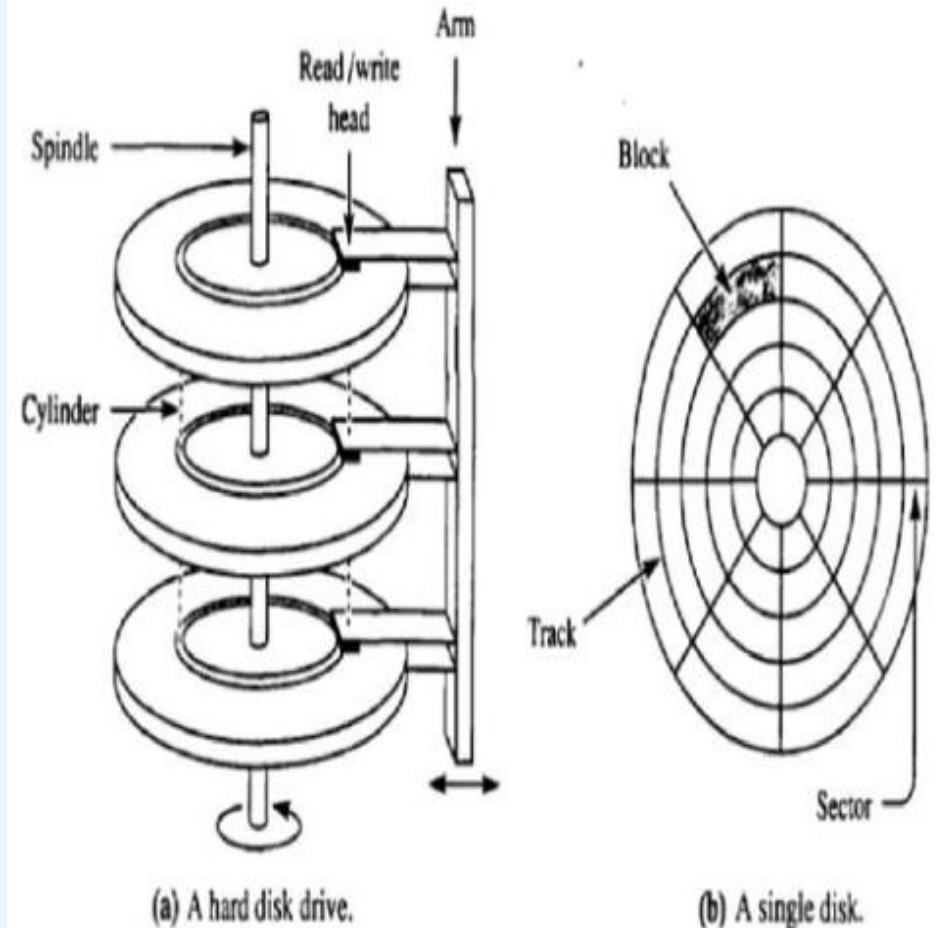
2.1 Đĩa từ - Đo lường hiệu năng

- ❖ **Tốc độ truyền dữ liệu:** tốc độ dữ liệu có thể được lấy ra từ đĩa hoặc được lưu trữ vào đĩa. Hiện nay tốc độ này vào khoảng 1..5 Mbps
- ❖ **Độ tin cậy** (TG trung bình không sự cố): lượng TG trung bình hệ thống chạy liên tục không có bất kỳ sự cố nào. Khoảng 30000 .. 800000 giờ (3,4 - 91 năm)

2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Tối ưu hóa truy xuất

- Yêu cầu I/O sinh ra bởi hệ thống file và bộ quản trị bộ nhớ ảo
- Đơn vị dữ liệu truyền giữa đĩa và bộ nhớ là **khối**, khối sẽ được tính ra số trụ của mặt + sectors.

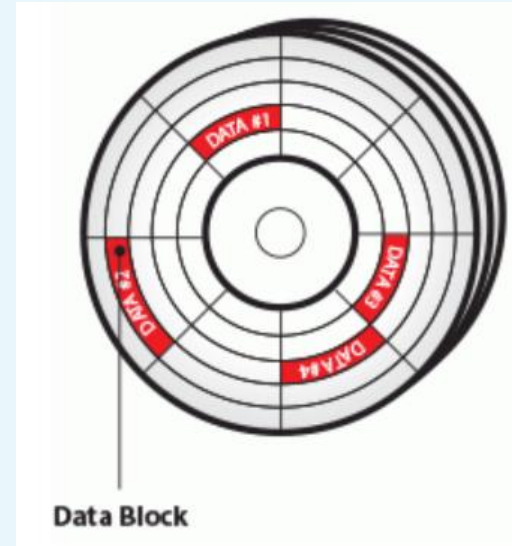




2. Đĩa từ và RAID (tt)

2.1 Đĩa từ - Tối ưu hóa truy xuất

- Cần có chiến lược nâng cao **tốc độ truy xuất khối đĩa**:
 - Định thời các tác vụ truy xuất đĩa (lập lịch biểu) để hạn chế di chuyển đầu đọc
 - Bố trí các file thường sử dụng vào vị trí thích hợp
 - Tăng tốc độ quay của đĩa
 - Giảm kích thước đĩa
 - Đĩa log





2. Đĩa từ và RAID

2.2 RAID - Định nghĩa

- **RAID** - **R**edundant **A**rray of **I**nexpensive **D**isks
- Là hình thức gộp nhiều ổ đĩa cứng vật lý thành một hệ thống ổ đĩa cứng
- Giúp gia tăng khả năng đọc/ghi và truy xuất dữ liệu từ ổ đĩa cứng



2. Đĩa từ và RAID (tt)

2.2 RAID - Mục đích và Lợi ích

❖ Mục đích

- Nâng cao hiệu suất vận hành của toàn bộ hệ thống
- Cải tiến hiệu năng thông qua quản lý song song
- Tăng độ tin cậy nhờ tính lưu trữ dư thừa dữ liệu

❖ Lợi ích:

- An toàn vì có cơ chế dự phòng
- Hiệu quả cao, sử dụng bộ nhớ dung lượng lớn
- Giá thành thấp



2. Đĩa từ và RAID (tt)

2.2 RAID - Các loại mức độ

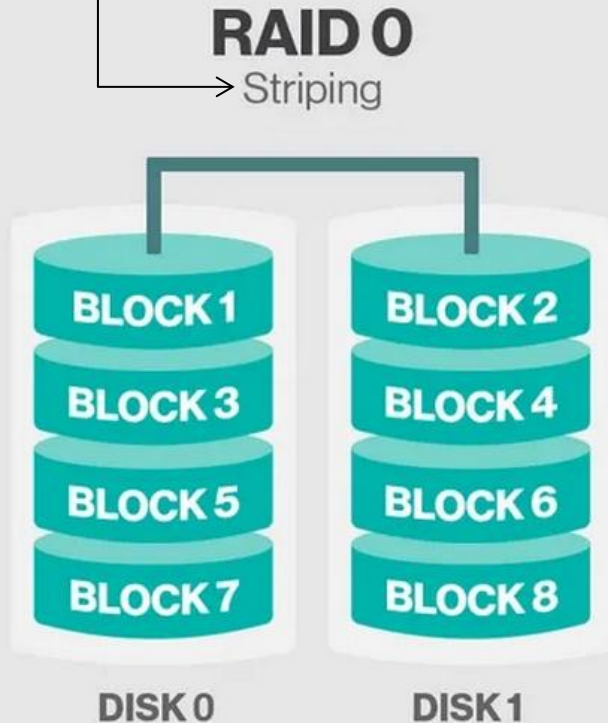
- Các nhân tố ảnh hưởng đến việc lựa chọn mức RAID:
 - Chi phí
 - Hiệu suất về số lượng hoạt động I/O
 - Hiệu năng khi hệ thống bị sự cố
 - Hiệu năng khi phục hồi
- Raid phổ biến, được sử dụng nhiều đó là 4 loại RAID:
Raid 0 – Raid 1 – Raid 5 – Raid 10



2. Đĩa từ và RAID (tt)

2.2.1 RAID 0 - Striping (chia sọc)

- Tách luồng dữ liệu thành các **khối** có kích thước nhất định
=> viết từng khối qua từng RAID



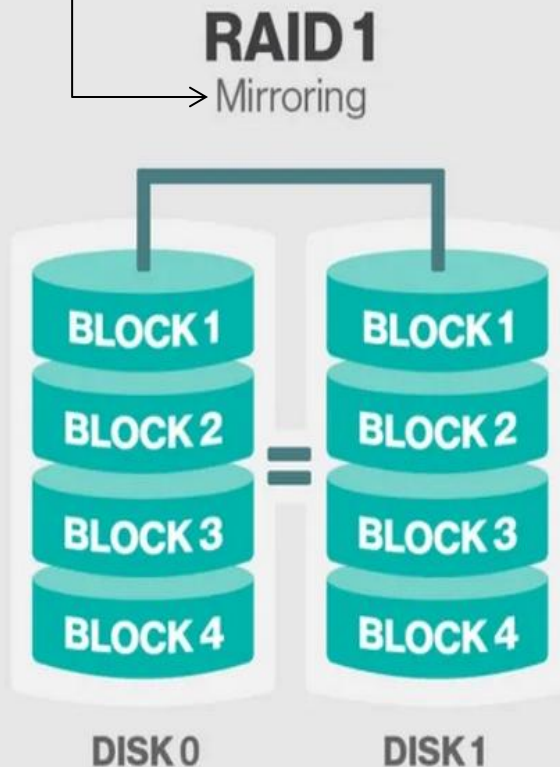
- Lưu trữ dữ liệu không quan trọng cần được đọc/ghi ở tốc độ cao
- Chỉnh sửa ảnh hoặc video



2. Đĩa từ và RAID (tt)

2.2.2 RAID 1 - Mirroring

- Là một kỹ thuật lưu trữ các bản sao dữ liệu giống hệt nhau được lưu trữ trên các thành viên RAID cùng một lúc



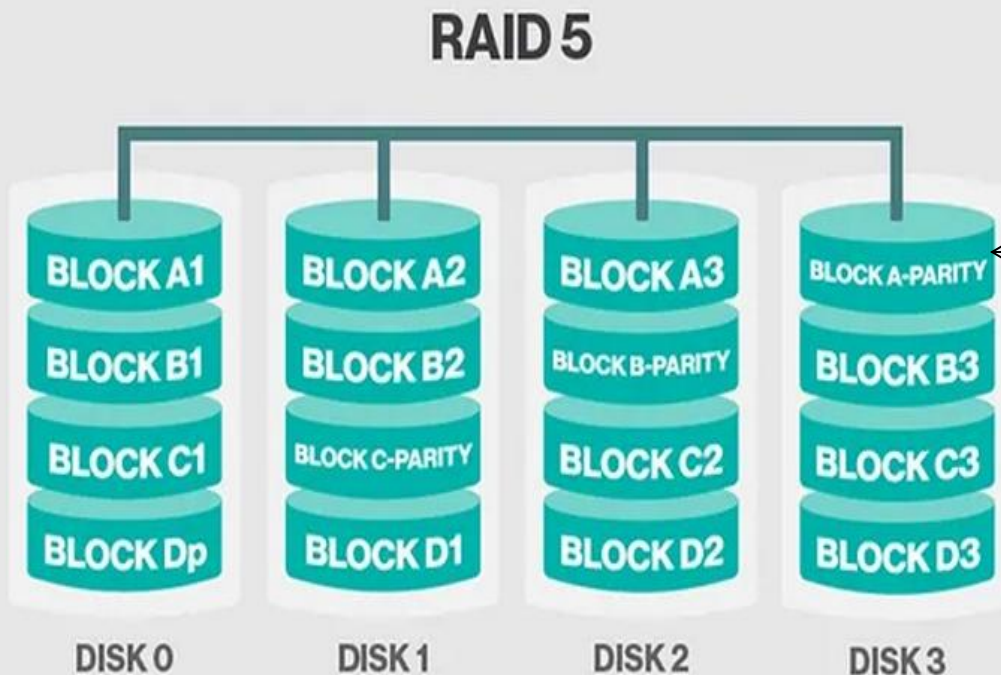
- Các dịch vụ yêu cầu về an toàn dữ liệu, không yêu cầu tốc độ đọc/ghi



2. Đĩa từ và RAID (tt)

2.2.3 RAID 5 - Parity

- Là chẵn và lẻ. Nếu tổng của 2 khối dữ liệu là chẵn thì số bit là chẵn, nếu tổng của 2 khối dữ liệu là lẻ thì số bit là lẻ

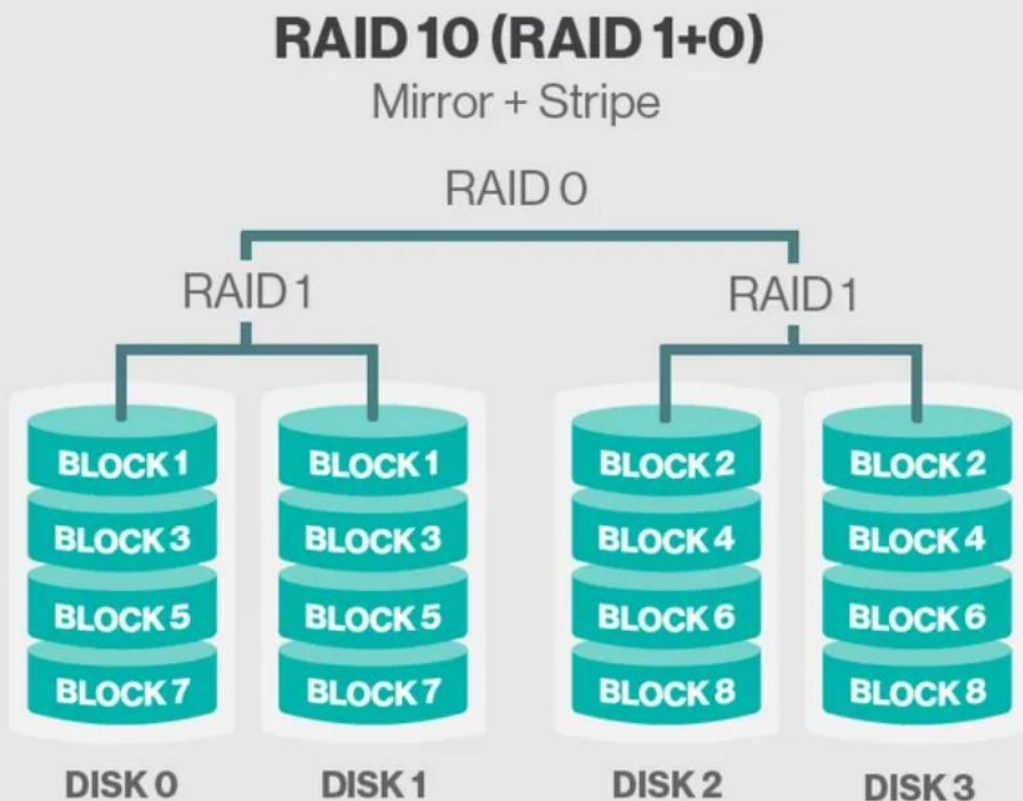


- Dịch vụ có số lượng truy cập và yêu cầu tài nguyên từ nhỏ đến vừa và lớn

2. Đĩa từ và RAID (tt)

2.2.4 RAID 10 - Mirror + Stripe

- Là sự kết hợp của RAID 1 và RAID 0. Với một mảng stripe (RAID 0) được sao chép qua một mảng mirror (RAID 1)



- Phù hợp mọi đối tượng, dịch vụ
- Yêu cầu về hiệu suất và đảm bảo an toàn dữ liệu



So sánh RAID 0, 1, 5, 10

	RAID 0	RAID 1	RAID 5	RAID 10
Giống nhau	Dùng để lưu trữ dữ liệu			
Ưu điểm	Tốc độ ghi nhanh	An toàn về dữ liệu Dễ phục hồi	Nâng cao hiệu suất, an toàn dữ liệu, chi phí nhỏ hơn RAID 10	Lưu trữ nhanh, an toàn, nâng cao hiệu suất
Nhược điểm	Tiềm ẩn nguy cơ mất dữ liệu	Dung lượng lưu trữ giảm	Làm giảm lưu trữ thực tế	<ul style="list-style-type: none">- Chi phí cao- Lưu trữ chỉ hiệu quả khoảng 50%



CANTHO UNIVERSITY

NỘI DUNG

1

Các loại thiết bị lưu trữ

2

Đĩa từ và RAID

3

Quản lý bộ đệm

4

Tổ chức tập tin

5

Tổ chức mẫu tin trong tập tin

6

Chỉ mục và băm

7

Xử lý vấn tin



3. Quản lý bộ đệm

- Mục tiêu nổi trội của hệ CSDL là **tối thiểu hóa** số khối truyền giữa đĩa và bộ nhớ. Một cách thực hiện là trữ sẵn trong RAM (bộ nhớ chính)
- Tuy nhiên, RAM không đủ để chứa tất cả DB → **buffer**.



Là một phần của bộ nhớ chính sẵn có để lưu trữ bản sao khối đĩa.



3. Quản lý bộ đệm (tt)

- Hệ thống con đảm trách cấp phát không gian buffer được gọi là **bộ quản trị buffer**
 - Khối **đã sẵn sàng** trong buffer → địa chỉ khối trong bộ nhớ chính được chuyển cho người yêu cầu.
 - Khối **chưa có** trong buffer → **bộ QT buffer** cấp phát không gian trong buffer cho khối
- Kích cỡ CSDL có thể rất lớn không đủ chứa toàn bộ trong bộ nhớ chính → sử dụng các kỹ thuật:



3. Quản lý bộ đệm (tt)

Các kỹ thuật:

1. Chiến lược thay thế: Khi không có chỗ trong buffer → một khối phải xóa khỏi buffer → khối mới được đọc vào.
2. **Khối chết**: Khối không cho phép viết lại lên đĩa
3. **Xuất ra bắt buộc các khối**: nội dung bộ nhớ chính bị mất khi có sự cố, DL trên đĩa còn tồn tại sau sự cố



3. Quản lý bộ đệm (tt)

📖 Chiến lược thay thế khối trong buffer: tối thiểu hóa các truy xuất đĩa

- **LRU:** thay thế khối đã được dùng **lâu nhất**.
- **MRU:** thay thế khối được dùng gần **đây nhất**.
- **Các thông tin khác:** xác suất sử dụng (tự điển DL, index), thông tin từ các thành phần khác (bộ điều khiển cạnh tranh, hệ thống khôi phục)...



Lưu trữ từ điển dữ liệu

- Duy trì *dữ liệu về các quan hệ*. Lưu thông tin:
 - Các tên của các quan hệ
 - Các tên của các thuộc tính của mỗi quan hệ
 - Các miền (giá trị) và các độ dài của các thuộc tính
 - Các ràng buộc toàn vẹn....
- ➔ Biểu diễn DL hệ thống sử dụng các quan hệ như thế nào là do *người thiết kế hệ thống* quyết định.



4. Tổ chức tập tin (file)

- Một file được tổ chức logic như một dãy các **mẫu tin** (record) → ánh xạ lên các khối đĩa.
- Có 2 loại mẫu tin:
 - Các mẫu tin có độ dài **cố định**
 - Các mẫu tin có độ dài **thay đổi**
- Ta sẽ xét các phương pháp lưu trữ các mẫu tin vào trong file cho từng loại mẫu tin.



4. Tổ chức tập tin (file) (tt)

4.1 Mẫu tin có độ dài cố định

- Xét các mẫu tin trong file instructor, được định nghĩa như:

```
type instructor = record
```

```
    ID varchar (5);
```

```
    name varchar(20);
```

```
    dept_name varchar (20);
```

```
    salary numeric (8,2);
```

```
end
```

53 byte

Vấn đề nảy sinh:

Nếu kích thước của một khối không chia hết cho 53 thì một số mẫu tin sẽ vượt quá một khối đĩa (một mẫu tin có thể nằm trong hai khối đĩa. Phải truy xuất tới hai khối đĩa để đọc hay ghi một bản ghi.

4. Tổ chức tập tin (file) (tt)

4.1 Mẫu tin có độ dài cố định

- **Cách giải quyết:** Lưu mỗi bản ghi trong một khối. Phần khoảng trống dư lại ở cuối khối sẽ bị bỏ qua.

→ Khó khăn khi xóa một mẫu tin

Xóa mẫu
tin thứ 3

Hình 1: File chứa các
record Instructor

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000



4. Tổ chức tập tin (file) (tt)

4.1 Mẫu tin có độ dài cố định

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Hình 2: File Instructor
sau khi đã xóa record
3 và di chuyển các
record sau về trước

- Khoảng không gian còn trống luôn ở cuối khối
- Phải di chuyển một lượng lớn các record



4. Tổ chức tập tin (file) (tt)

4.1 Mẫu tin có độ dài cố định

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 11	98345	Kim	Elec. Eng.	80000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000

Hình 3: File Instructor sau khi đã xóa record 3 và di chuyển record cuối cùng lên

- Phải truy xuất khối bổ xung → hoạt động xen xảy ra thường xuyên hơn hoạt động xóa.



4. Tổ chức tập tin (file) (tt)

4.1 Mẫu tin có độ dài cố định

header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Hình 4: Sau khi xóa các record 1, 4 và 6 với chiến lược **bỏ ngỗ** và **danh sách tự do**

Header chứa địa chỉ của các record bị xóa ==> danh sách tự do

- Cấp phát một số byte nhất định làm header, chứa thông tin về file.

4. Tổ chức tập tin (file) (tt)

4.1 Mẫu tin có độ dài cố định

- **Ưu điểm:** không gian được giải phóng bởi mẫu tin bị xoá **đúng bằng** không gian cho mẫu tin thêm vào.
- **Nhược điểm:** độ dài mẫu tin thay đổi ==> phức tạp

4. Tổ chức tập tin (file)

4.2 Mẫu tin có độ dài thay đổi

- Mẫu tin độ dài thay đổi trong CSDL do bởi:
 - Việc lưu nhiều kiểu mẫu tin trong cùng một file
 - Kiểu mẫu tin cho phép độ dài trường thay đổi
 - Kiểu mẫu tin cho phép lặp lại các trường

Ví dụ:

```
type account-list=record
    branch-name: char(22);
    account-info: array[1..∞] of
        record
            account_number: char(10);
            balance: real;
        end;
end;
```

end;

4. Tổ chức tập tin (file) (tt)

4.2 Mẫu tin có độ dài thay đổi

❖ Biểu diễn chuỗi byte

0	Perryridge	A-102	400	A-201	900	A-210	700	⊥
1	Round Hill	A-310	350	⊥				
2	Mianus	A-110	800	⊥				
3	Downtown	A-211	500	A-222	600	⊥		
4	Redwood	A-300	650	A-200	1200	A-255	950	⊥
5	Brighton	A-111	750	⊥				

End of record

- Khó sử dụng lại không gian trống => lưu trữ bị lãng phí
- Không có không gian phát triển cho mẫu tin.

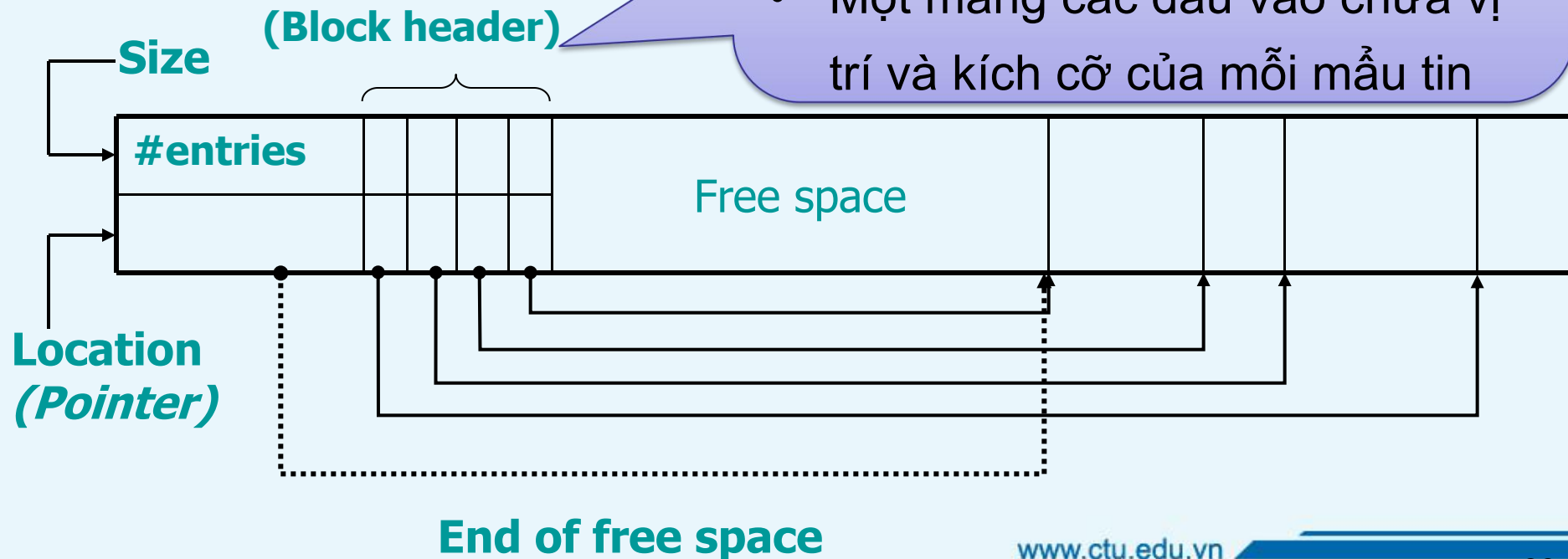


4. Tổ chức tập tin (file) (tt)

4.2 Mẫu tin có độ dài thay đổi

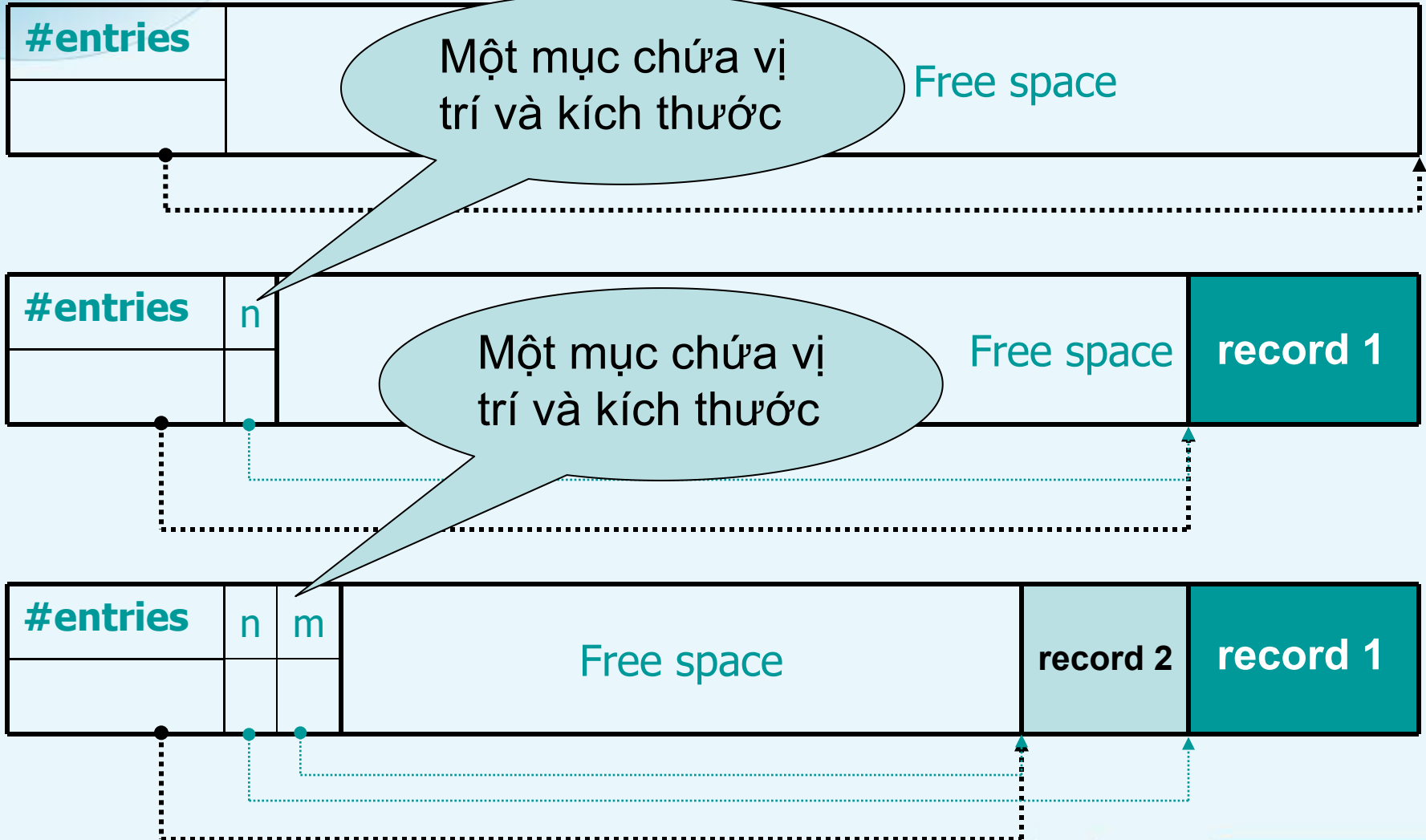
❖ Dùng cấu trúc khe trang

- Số các đầu vào mẫu tin trong head
- Điểm cuối khoảng không gian trống trong khối.
- Một mảng các đầu vào chứa vị trí và kích cỡ của mỗi mẫu tin



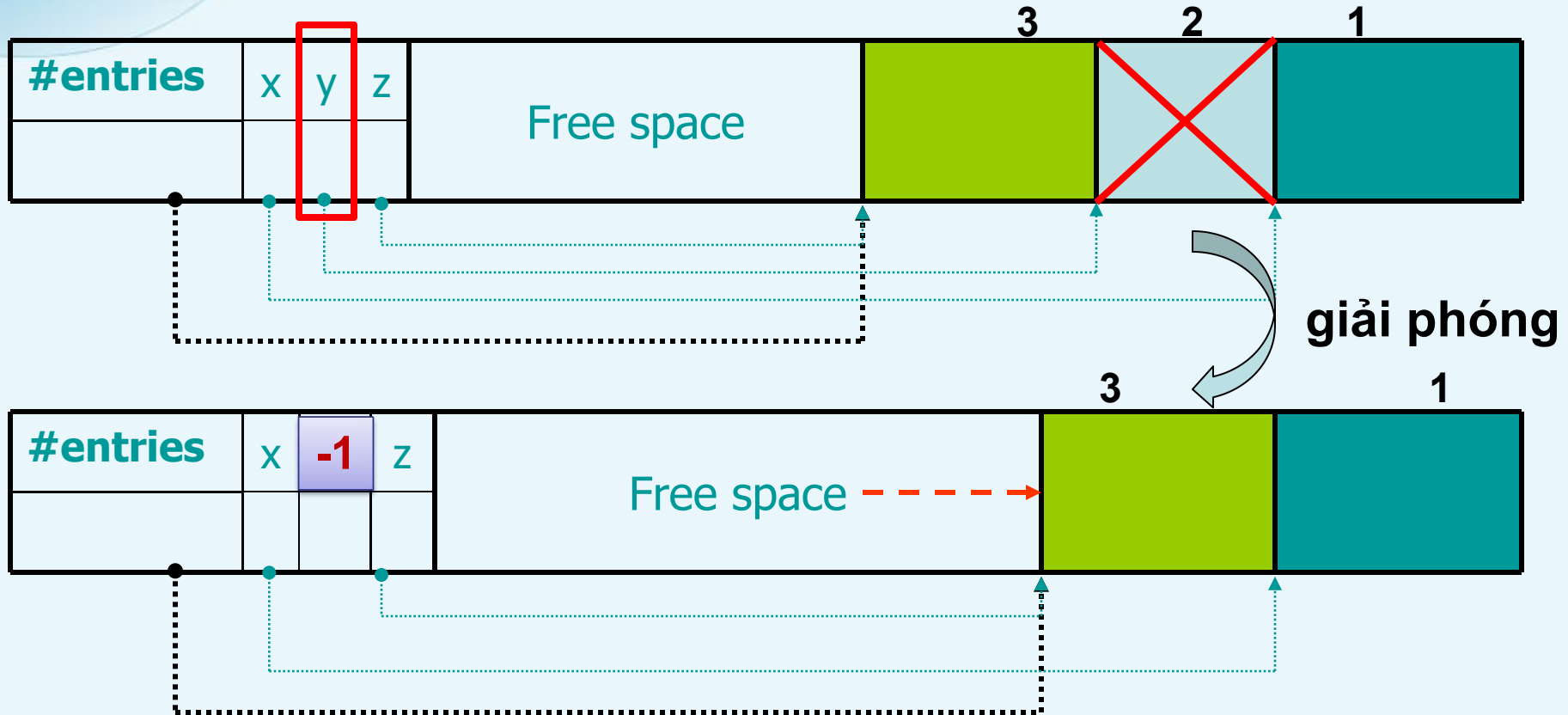


Khi mẫu tin được thêm vào khối





Khi xóa 1 mẫu tin trong khối





Biểu diễn độ dài cố định

➤ Dừng không gian dự trữ:

0	Perryridge	A-102	400	A-201	900	A-210	700	NULL
1	Round Hill	A-310	350	NULL	NULL	NULL	NULL	NULL
2	Mianus	A-110	800	NULL	NULL	NULL	NULL	NULL
3	Downtown	A-211	500	A-222	600	NULL	NULL	NULL
4	Redwood	A-300	650	A-200	1200	A-255	950	NULL
5	Brighton	A-111	750	NULL	NULL	NULL	NULL	NULL

- Sử dụng mẫu tin độ dài tối đa
- Phần không gian chưa dùng đến được lấp đầy ký tự **null**



CANTHO UNIVERSITY

Biểu diễn độ dài cố định

➤ Dùng phương pháp con trỏ:

0	Perryridge	A-102	400		
1		A-201	900		
2		A-210	700	●	
3	Round Hill	A-310	350		
4	Mianus	A-110	800		
5	Downtown	A-211	500		
6		A-222	600	●	
7	Redwood	A-300	650		
8	Brighton	A-111	750		
9		A-200	1200		
10		A-255	950	●	

Được "móc xích" với nhau bởi các con trỏ

Lãng phí không gian trong tất cả các mẫu tin (trừ mẫu tin đầu tiên)



Biểu diễn độ dài cố định

➤ Dùng phương pháp khối neo và khối tràn

Khối neo: chứa
chỉ các mẫu tin
đầu tiên trong
danh sách

Perryridge	A-102	400	
Round Hill	A-310	350	
Mianus	A-110	800	
Downtown	A-211	500	
Redwood	A-300	650	
Brighton	A-111	750	

Khối tràn: còn lại
của danh sách

A-201	900	
A-210	700	•
A-222	600	•
A-200	1200	
A-255	950	•



5. Tổ chức mẫu tin trong tập tin (File)

Tổ chức file
tuần tự
(sequential)

Tổ chức file
đống
(heap)

Tổ chức file
băm
(hashing)

Tổ chức file
cụm
(clustering)



5.1 Tổ chức file tuần tự (Sequential)

- Các mẫu tin được lưu trữ tuần tự dựa trên giá trị của khoá tìm kiếm của mỗi mẫu tin.

Brighton	A-217	750	
Downtown	A-101	500	
Downtown	A-110	600	
Mianus	A-215	700	
Perryridge	A-102	400	
Perryridge	A-201	900	
Perryridge	A-218	700	
Redwood	A-222	700	
Round Hill	A-305	350	

"xích" các mẫu tin lại bởi các con trỏ



File tuần tự thêm record vào khối tràn

Brighton	A-217	750	●
Downtown	A-101	500	●
Downtown	A-110	600	●
Mianus	A-215	700	●
Perryridge	A-102	400	●
Perryridge	A-201	900	●
Perryridge	A-218	700	●
Redwood	A-222	850	●
Round Hill	A-301	550	●

Khối tràn

North Town	A_777	1100	●
------------	-------	------	---

Áp dụng quy tắc:

1. Định vị mẫu tin trong file theo thứ tự khóa tìm kiếm
2. Nếu có mẫu tin tự do, thì xen vào khối. Nếu không, xen vào 1 khối tràn



5.2 Tổ chức file đồng (Heap)

- Là một mẫu tin có thể được lưu ở bất kỳ vị trí nào trong file, không có thứ tự nào cho các mẫu tin.
- Các thao tác:
 - Tìm kiếm: có giá trị khóa cho trước → quét toàn bộ tệp
 - Thêm: được thêm vào sau mẫu tin cuối cùng
 - Xóa: bao hàm thao tác tìm kiếm, nếu có đánh dấu xóa
 - Sửa: Tìm 1 mẫu tin sửa một hay nhiều trường



5.3 Tổ chức file băm (Hashing)

- Quy tắc:
 - Phân chia các mẫu tin vào các cụm
 - Mỗi cụm gồm 1 hoặc nhiều khối
 - Mỗi khối chứa số lượng mẫu tin cố định
- ➔ Tiêu chí chọn hàm băm: phân bổ các mẫu tin tương đối đồng đều theo các cụm



5.4 Tổ chức file cụm (clustering)

- Các mẫu tin của một vài quan hệ được lưu trữ trong cùng một file
- Các mẫu tin có liên hệ của vài quan hệ khác nhau được lưu trên cùng một khối sao cho hoạt động I/O đem lại các mẫu tin có liên hệ từ tất cả các quan hệ



5.4 Tổ chức file cụm (tt)

◆ Quan hệ *depositor* (người gửi tiền)

<i>Customer-name</i>	<i>Account-number</i>
Hayes	A-102
Hayes	A-220
Hayes	A-503
Turner	A-305

◆ Quan hệ *customer* (khách hàng)

<i>Customer-name</i>	<i>Customer-street</i>	<i>Customer-city</i>
Hayes	Main	Brooklyn
Turner	Putnam	Standford



5.4 Tổ chức file cụm (tt)

- ◆ Giả sử người dùng đặt ra câu vấn tin:
**SELECT account-number, customer-name,
customer-street, customer-city
FROM depositor d, customer c
WHERE d.customer-name = c.customer-name**
- ◆ Câu vấn tin này chính là phép nối của các quan hệ ***customer và depositor***
- ◆ Ta sẽ trình bày một cấu trúc file được thiết kế để thực hiện hiệu quả các câu vấn tin trên



5.4 Tổ chức file cụm (tt)

❖ Cấu trúc file cụm cho hai quan hệ ***customer*** và ***depositor***

Hayes	Main	Brooklyn
Hayes	A-102	
Hayes	A-220	
Hayes	A-503	
Turner	Putnam	Standford
Turner	A-305	