

I. Chương 2: Hệ thống File

1. Giả sử bảng FAT đã được đọc vào bộ nhớ tại địa chỉ <<void *fat>>: liệt kê các cluster trống trong N cluster đầu tiên; cluster đầu là n, liệt kê các cluster thuộc file đó,...

```
//Liệt kê các cluster trống trong N cluster đầu:
for (int i = 2; i < N; i++) {
    if (fat[i] == 0) printf("%u ", i);
}
```

```
//File bắt đầu tại cluster n, liệt kê các cluster thuộc file đó:
int cur = n;
while (cur < 0xFFF8) {
    printf("->%u", cur);
    cur = fat[cur];
}
```

2. Giả sử thư mục gốc của hệ thống file FAT16 sử dụng tên file độ dài tối đa 8 ký tự đã được đọc vào bộ nhớ tại địa chỉ <<void *root>>: In tên, độ dài mỗi file trong thư mục gốc; tìm 1 file có tên cho trước và in ra số thứ tự các cluster thuộc file đó hoặc in ra số thứ tự cluster đầu tiên của file đó.

```
//In tên, độ dài các file trong thư mục gốc:
for (int i = 0; i < boot.ROOT_size; i++) {
    if (root[i].name[0] == 0 || root[i].name[0] == ' ') continue;
    for (int j = 0; j < 8 && root[i].name[j] != ' '; j++)
        printf("%c", root[i].name[j]);
    if (root[i].ext[0] != ' ') {
        printf(".");
        for (int k = 0; k < 3 && root[i].ext[k] != ' '; k++)
            printf("%c", root[i].ext[k]);
    }
    printf(" %ld\n", root[i].size);
}
```

```
//Tìm 1 file có tên cho trước và in ra số thứ tự cluster đầu tiên của file đó hoặc in ra số thứ tự các cluster thuộc file đó:
```

```
char str[9];
char filename[8];
```

```

printf("\n Enter a file name: ");
scanf("%s", filename);
int first_cluster = -1;
for (int i = 0; i < boot.ROOT_size; i++) {
    for (int k = 0; k < 8 && root[i].name[k] != ' '; k++)
        str[k] = root[i].name[k];
    str[k] = 0;
    if (strcmp(str, filename) == 0) {
        first_cluster = root[i].first_cluster;
        break;
    }
}
if (first_cluster != -1) {
    printf("Tim thay file!\n");
    unsigned int cur = first_cluster;
    //Số thứ tự cluster đầu tiên
    printf("Cluster dau tien: %u", cur);
    //Số thứ tự các cluster thuộc file đó
    while (cur < 0xFFFF8) {
        printf("->%u", cur);
        cur = fat[cur];
    }
}
else printf("Khong tim thay file!");

```

II. Chương 3: Quản lý bộ nhớ

Dạng 1: Sử dụng kỹ thuật kề cận để cấp phát không gian cho các tiến trình

Bộ nhớ kích thước 1MB. Sử dụng phương pháp kề cận để cấp phát cho các tiến trình lần lượt với kích thước như sau: A (120KB), B(210KB), C(150KB), D(40KB)

Sử dụng phương pháp ta có thứ tự cấp phát cho các tiến trình như sau:

1024KB trống	1024KB					
Cấp phát A: 120KB	A	128KB	256KB	512KB		
Cấp phát B: 210KB	A	128KB	B	512KB		
Cấp phát C: 150KB	A	128KB	B	C	256KB	
Cấp phát D: 40KB	A	D	64KB	B	C	256KB

Dạng 2: Tính toán số lượng bit cho địa chỉ logic, địa chỉ vật lý trong phân trang bộ nhớ

- Ánh xạ địa chỉ gồm 2 phần là số thứ tự trang (p) và độ dịch (địa chỉ lệch) của địa chỉ so với đầu trang (o).
- Nếu kích thước trang là 2^n , biểu diễn địa chỉ logic có (m + n) bit với m bit biểu diễn p và n bit biểu diễn o.
- Chuyển địa chỉ logic sang địa chỉ vật lý:
 - Lấy m bit cao của địa chỉ \Rightarrow được số thứ tự trang.
 - Dựa vào bảng trang, tìm được số thứ tự khung vật lý (k).
 - Địa chỉ vật lý bắt đầu của khung là $k * 2^n$.
 - Địa chỉ vật lý của byte được tham chiếu = địa chỉ bắt đầu của khung + Địa chỉ lệch (độ dịch).

1. Cho trang nhớ kích thước 1024byte, độ dài địa chỉ logic là 16bit. Tính địa chỉ logic 1500? Tính số thứ tự trang và độ dịch trong trang ?

$m+n = 16$ do không gian nhớ logic là 16 bit.

Số lượng khung trong 1 trang là 1024Bytes = 2^{10} Bytes, nên cần 10 bit để biểu diễn cho khung trong trang $\Rightarrow n = 10$; $m = 6$

Ta có:

Phần p = $[1500 / 1024] = 1$

Phần o = $1500 \bmod 1024 = 476$

Vậy ta có địa chỉ logic dưới dạng thập phân 1500 có 2 phần (trang 1, độ dịch 476).

2. Cho bảng trang như sau với kích thước trang là 1KB. Hãy chuyển các địa chỉ logic sau sang địa chỉ vật lý: 1240, 3580

Page	Frame
0	3
1	2
2	6
3	4

Kích thước trang là 1024B

• 1240:

Ta có:

Phần p = $[1240 / 1024] = 1 \Rightarrow$ số thứ tự trang là 1 \Rightarrow số khung là 2.

Phần o = $1240 \bmod 1024 = 216 \Rightarrow$ độ dịch trong trang là 216.

Địa chỉ vật lý: $1024 * 2 + o = 1024 * 2 + 216 = 2264$.

• 3580:

Ta có:

Phần p = $[3580 / 1024] = 3 \Rightarrow$ số thứ tự trang là 3 \Rightarrow số khung là 4.

Phần o = $3580 \bmod 1024 = 508 \Rightarrow$ độ dịch trong trang là 508.

Địa chỉ vật lý: $1024 * 4 + o = 1024 * 4 + 508 = 4604$.

3. **Không gian nhớ logic gồm 512 trang, mỗi trang có kích thước 1024B. Bộ nhớ vật lý gồm 64 khung. Cho biết địa chỉ logic dài bao nhiêu bit, phần số thứ tự trang và phần độ dịch trong trang có độ dài lần lượt là bao nhiêu bit. Địa chỉ vật lý dài bao nhiêu bit.**

Kích thước không gian nhớ: $512 * 1024 = 2^{19}$ byte \Rightarrow Cần 19bit để biểu diễn địa chỉ logic.

Kích thước trang là 2^{10} byte \Rightarrow Độ dịch trong trang cần 10bit biểu diễn

Có $m + n = 19$ với $n = 10 \Rightarrow m = 9$ bit.

Vậy phần số thứ tự trang có độ dài 9bit và phần độ dịch trong trang độ dài 10bit.

Bộ nhớ vật lý gồm 64 khung \Rightarrow Không gian nhớ vật lý là $64 * 1024 = 2^{16}$ byte \Rightarrow Địa chỉ vật lý dài 16bit.

4. **Giả sử không gian nhớ logic gồm 52 trang, mỗi trang có kích thước 1048B. Bộ nhớ vật lý gồm 69 khung. Để biểu diễn địa chỉ vật lý trong trường hợp này cần bao nhiêu bit? Bao nhiêu bit dùng để biểu diễn số thứ tự các khung (frame) vật lý? Địa chỉ logic cần bao nhiêu bit?**

Kích thước trang $1048B > 2^{10}B \Rightarrow$ cần 11 bit để biểu diễn độ dịch trong trang

Bộ nhớ vật lý gồm 69 khung ($2^6 < 69 < 2^7$) \Rightarrow cần 7 bit để biểu diễn số thứ tự các khung vật lý.

\Rightarrow Để biểu diễn địa chỉ vật lý trong trường hợp này cần $11 + 7 = 18$ bit

Không gian nhớ logic gồm 52 trang ($2^5 < 52 < 2^6$) \Rightarrow cần 6 bit biểu diễn số thứ tự trang.

\Rightarrow Địa chỉ logic cần $6 + 11 = 17$ bit

Dạng 3: Tính địa chỉ logic của phân đoạn bộ nhớ

- Sử dụng bảng đoạn (SCB: Segment Control Block) cho mỗi tiến trình. Mỗi phần tử của bảng tương ứng với 1 đoạn, chứa: Dấu hiệu (Mark (0/1)); Địa chỉ cơ sở (Address); Giới hạn đoạn (Length)
- Địa chỉ logic gồm 2 phần: <Tên (STT) đoạn (s), độ dịch trong đoạn (o)>
- Từ chỉ số đoạn s, vào bảng đoạn, tìm địa chỉ vật lý bắt đầu của đoạn.
- So sánh độ dịch o với chiều dài đoạn, nếu lớn hơn \Rightarrow Địa chỉ sai. Lỗi truy cập.
- Địa chỉ vật lý mong muốn là tổng của địa chỉ vật lý bắt đầu đoạn và địa chỉ lệch.

Giả sử trong quá trình quản lý bộ nhớ dạng phân đoạn, hệ điều hành duy trì bảng phân đoạn (Segment Table) như sau:

Số thứ tự đoạn s (Segment)	Cơ sở (Base)	Giới hạn (Limit)
0	350	450
1	1200	950

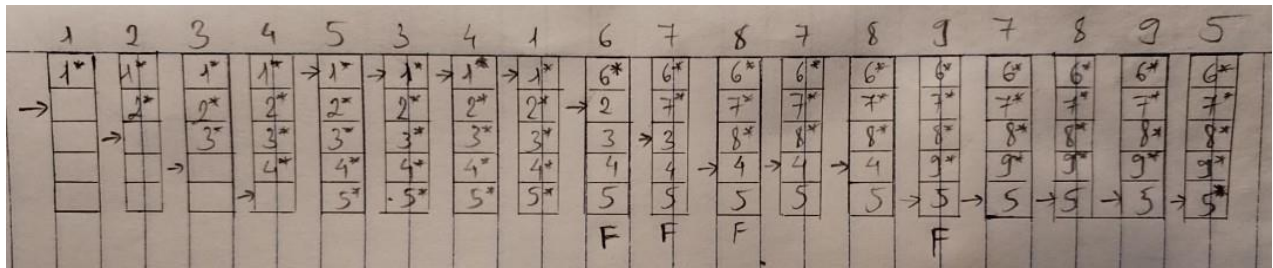
F F F

F

F

- Sử dụng CLOCK

Sơ đồ cấp phát bộ nhớ: \Rightarrow 4 lần đổi trang.



III. Chương 4: Quản lý tiến trình

Các thuật toán điều độ tiến trình (thời gian chờ đợi trung bình, thời gian vòng đời trung bình)

- Cho các tiến trình với thời gian (độ dài) chu kỳ CPU tiếp theo như trong bảng. Biết rằng các tiến trình cùng xuất hiện tại thời điểm 0 theo thứ tự P1, P2, P3, P4, P5. Vẽ biểu đồ thể hiện thứ tự và thời gian cấp phát CPU cho các tiến trình khi sử dụng thuật toán đến trước phục vụ trước; thuật toán điều độ quay vòng với thời gian lượng tử = 2. Tính thời gian chờ đợi trung bình và thời gian vòng đời trung bình cho từng trường hợp hợp.

Process	Burst Time
P1	5
P2	3
P3	1
P4	7
P5	4

- Đến trước phục vụ trước

0	5	8	9	16	20
5	3	1	7	4	
P1	P2	P3	P4	P5	

Thời gian chờ đợi:

P1: 0

P2: 5

P3: 8 \Rightarrow Thời gian chờ đợi trung bình: $(0 + 5 + 8 + 9 + 16) / 5 = 7.6$

P4: 9

P5: 16

Thời gian vòng đời:

P1: $5 - 0 = 5$

P2: $8 - 0 = 8$

P3: $9 - 0 = 9 \Rightarrow$ Thời gian vòng đời trung bình: 11.6

P4: $16 - 0 = 16$

P5: $20 - 0 = 20$

- Điều độ quay vòng với lượng tử thời gian = 2.

0	2	4	5	7	9	11	12	14	16	17	19	20
2	2	1	2	2	2	1	2	2	1	2	1	
P1	P2	P3	P4	P5	P1	P2	P4	P5	P1	P4	P4	

Thời gian chờ đợi:

$$P1: 9 - 2 + 16 - 11 = 12$$

$$P2: 2 + 11 - 4 = 9$$

$$P3: 4$$

=> Thời gian chờ đợi trung bình: 10

$$P4: 5 + 12 - 7 + 17 - 14 = 13$$

$$P5: 7 + 14 - 9 = 12$$

Thời gian vòng đời:

$$P1: 17$$

$$P2: 12$$

$$P3: 5 \Rightarrow \text{Thời gian vòng đời trung bình: } 14$$

$$P4: 20$$

$$P5: 16$$

2. Cho các tiến trình với thời điểm xuất hiện, thời gian (chu kỳ) CPU tiếp theo trong bảng sau. Vẽ biểu đồ thể hiện thứ tự và thời gian cấp phát CPU cho các tiến trình sử dụng thuật toán: Điều độ ưu tiên thời gian còn lại ngắn nhất.

Process	Burst	Arrival
P1	15	0
P2	20	0
P3	20	20
P4	20	25
P5	5	45
P6	15	55

Tính thời gian chờ đợi trung bình và thời gian vòng đời trung bình.

0	15	35	45	50	60	75	95
15	20	10	5	10	15	20	
P1	P2	P3	P5	P3	P6	P4	

Thời gian chờ đợi:

$$P1: 0$$

$$P2: 15 - 0 = 15$$

$$P3: 35 - 20 + 50 - 45 = 20 \Rightarrow \text{Thời gian chờ đợi trung bình: } 15$$

$$P4: 75 - 25 = 50$$

$$P5: 45 - 45 = 0$$

$$P6: 60 - 55 = 5$$

Thời gian vòng đời:

$$P1: 15 - 0 = 15$$

$$P2: 35 - 0 = 35$$

P3: $60 - 20 = 40 \Rightarrow$ Thời gian vòng đời trung bình: $185/6$

P4: $95 - 25 = 70$

P5: $50 - 45 = 5$

P6: $75 - 55 = 20$

3. Cho các tiến trình với thời điểm xuất hiện, thời gian (chu kỳ) CPU tiếp theo và số ưu tiên như trong bảng sau (số ưu tiên nhỏ ứng với độ ưu tiên cao). Vẽ biểu đồ thể hiện thứ tự và thời gian cấp phát CPU cho các tiến trình sử dụng thuật toán: Điều độ theo mức ưu tiên không có phân phối lại; Điều độ ưu tiên tiến trình ngắn nhất; Điều độ ưu tiên thời gian còn lại ngắn nhất.

Tính thời gian chờ đợi trung bình và thời gian vòng đời trung bình cho từng trường hợp.

Tiến trình	Thời điểm xuất hiện (c)	Thời gian (độ dài)	Số ưu tiên
P1	0	10	1
P2	2	8	3
P3	4	5	2
P4	5	3	2

- Mức ưu tiên không phân phối lại.

0	10	15	18	26
10	5	3	8	
P1	P3	P4	P2	

Thời gian chờ đợi:

P1: 0

P2: $18 - 2 = 16$

P3: $10 - 4 = 6 \Rightarrow$ Thời gian chờ đợi trung bình: 8

P4: $15 - 5 = 10$

Thời gian vòng đời:

P1: $10 - 0 = 10$

P2: $26 - 2 = 24$

P3: $15 - 4 = 11 \Rightarrow$ Thời gian vòng đời trung bình: 14.5

P4: $18 - 5 = 13$

- Ưu tiên tiến trình ngắn nhất:

0	10	13	18	26
10	3	5	8	
P1	P4	P3	P2	

Thời gian chờ đợi:

P1: 0

P2: $18 - 2 = 16$

P3: $13 - 4 = 9 \Rightarrow$ Thời gian chờ đợi trung bình: 7.5

$$P4: 10 - 5 = 5$$

Thời gian vòng đời:

$$P1: 10 - 0 = 10$$

$$P2: 26 - 2 = 24$$

$$P3: 18 - 4 = 14 \Rightarrow \text{Thời gian vòng đời trung bình: } 14$$

$$P4: 13 - 5 = 8$$

- Ưu tiên thời gian còn lại ngắn nhất:

0	4	5	8	12	18	26
4	1	3	4	6	8	
P1	P3	P4	P3	P1	P2	

Thời gian chờ đợi:

$$P1: 0 + 12 - 4 = 8$$

$$P2: 18 - 2 = 16$$

$$P3: 4 - 4 + 8 - 5 = 3 \Rightarrow \text{Thời gian chờ đợi trung bình: } 6.75$$

$$P4: 5 - 5 = 0$$

Thời gian vòng đời:

$$P1: 18 - 0 = 18$$

$$P2: 26 - 2 = 24$$

$$P3: 12 - 4 = 8 \Rightarrow \text{Thời gian vòng đời trung bình: } 14.75$$

$$P4: 8 - 5 = 3$$

4. Cho các tiến trình với độ dài và thời điểm xuất hiện như trong bảng:

Tiến trình	Thời gian xuất hiện	Số ưu tiên	Độ dài
P1	0	3	24
P2	10	1	13
P3	12	1	31

Vẽ biểu đồ thứ tự và thời gian cấp phát CPU cho các tiến trình và thời gian vòng đời trung bình, thời gian chờ đợi trung bình khi sử dụng thuật toán điều độ ưu tiên tiến trình theo mức độ ưu tiên có phân phối lại biết số ưu tiên lớn thì độ ưu tiên thấp hơn.

0	10	23	54	68
10	13	31	14	
P1	P2	P3	P1	

Thời gian chờ đợi:

$$P1: 54 - 10 = 44$$

$$P2: 0 \Rightarrow \text{Thời gian chờ đợi trung bình: } 55/3$$

$$P3: 23 - 12 = 11$$

Thời gian vòng đời:

$$P1: 68$$

P2: $23 - 10 = 13 \Rightarrow$ Thời gian vòng đời trung bình: 41

P3: $54 - 12 = 42$