Bài tập Bộ nhớ ảo

Bài 1:

Một hệ thống máy tính sử dụng bộ nhớ ảo với cơ chế phân trang, cấu hình như sau : địa chỉ logic 32 bits, 512MB RAM; kích thước trang là 4096 byte.

Trong hệ thống trên, xét các tiến trình P1, P2, P3 với các bảng trang tương ứng:

	Frame	Valid bit
0	#300	V
1		I
2	#301	V
3		I

Bảng trang của P1

	Frame	Valid bit
0	#300	V
1		I
2	#302	V
3	#400	V

Bảng trang của P2

	Frame	Valid bit
0		I
1		I
2		I

Bảng trang của P3

Câu 1.a: Cho biết hệ thống có bao nhiêu khung trang ? Số lượng trang trong không gian địa chỉ của một tiến trình ?

Câu 1.b: Mô tả cách thức hệ thống thực hiện tuần tự các truy xuất bộ nhớ đến những địa chỉ sau, phân biệt vai trò của MMU (Memory Management Unit) và Hệ Điều Hành:

a) P1 truy cập đến địa chỉ 13000.
b) P2 truy cập đến địa chỉ 13000.
c) P1 truy cập đến địa chỉ 16383.
d) P3 truy cập đến địa chỉ 4096.
e) P3 truy cập đến địa chỉ 13000.
f) P2 truy cập đến địa chỉ 16383.
g) Cho biết tại sao có hai tiến trình cùng được cấp phát khung trang #300 ?

Biết rằng :

- Hiện tại hệ thống chỉ còn 1 khung trang tự do là #0.
- Nhằm đảm bảo tính hiệu quả khi thi hành, HĐH neo một số trang trong bộ nhớ (không được phép swap out các trang này), bao gồm : page 0, page 2 của P1; page 0, page 2 của P2
- Khi có lỗ trang, sử dụng chiến lược thay thế trang LRU (Least Recently Used).

Giải:

Câu 1.a:

+ Số khung trang = $\frac{512MB}{4096b} = \frac{2^9 \times 2^{20}}{2^2 \times 2^{10}} = 2^{17} = 128\text{kb}$

+ Kích thước không gian nhớ ảo: 2^{32}b

+ Số trang: $\frac{2^{32}b}{4096b} = 2^{20}$b (=1MB)

Câu 1.b:

a. P_1 truy cập địa chỉ 13000 => p = 13000 DIV 4096, d = 13000 MOD 4096

====> p =3, d= 712. Như vậy P_1 truy xuất trang 3. MMU dò bảng trang, trang 3 invalid -> phát sinh lỗi trang. HĐH cấp cho tiến trình P_1 khung trang #0 để lưu trang 3, nạp trang, cập nhật bảng trang P_1, trang 3 ứng với khung trang #0.

	Frame	Valid bit
0	#300	V
1		I
2	#301	V
3	#0	V

Bảng trang của P1

	Frame	Valid bit
0	#300	V
1		I
2	#302	V
3	#400	V

Bảng trang của P2

	Frame	Valid bit
0		I
1		I
2		I

Bảng trang của P3

<p = 3, d = 712> -----> <f = #0, d = 712>

MMU dò bảng trang, trang 3 ứng với khung trang 0, truy cập bộ nhớ tại đ/c 712 của khung trang #0.

b. P_2 truy cập địa chỉ 13000: <p = 3, d = 712>

MMU dò bảng trang P_2: <p = 3, d= 712> ----> <f = #400, d = 712>

MMU truy cập bộ nhớ tại địa chỉ: khung trang #400, offset 712.

c. P_1 truy cập đ/c 16383 : <p = 3, d = 4095>

MMU dò bảng trang P_1 : <p = 3, d = 4095> ----> <f = #0, d = 4095>

MMU truy cập bộ nhớ tại địa chỉ: khung trang #0, offset 4095

d. P_3 truy cập đ/c 4096: <p = 1, d = 0>

MMU dò bảng trang, trang 1 không hợp lệ, lỗi trang.

HĐH: Tìm trang trống: không còn, phải chọn 1 trang nạn nhân

Các trang: (P_1.0 #300 P_1.2#301 P_2.0#300 P_1.2#302) bị neo

Chỉ còn trang #0 và #400.

Theo chiến lược LRU, HDH chọn khung trang #400 vì trong các chu kỳ gần nhất sủ dụng #0 . Cấp cho trang 1 của P_3 khung trang #400, nạp trang cập nhật bảng trang.

	Frame	Valid bit
0	#300	V
1		I
2	#301	V
3	#0	V

Bảng trang của P1

	Frame	Valid bit
0	#300	V
1		I
2	#302	V
3		I

Bảng trang của P2

	Frame	Valid bit
0		I
1	#400	V
2		I

Bảng trang của P3

MMU dò bảng trang P_1 <p = 1, d = 0> -> <f = #400, d = 0>

MMU truy cập bộ nhớ đ/c 0 cua khung trang #400.

g. Đó là cơ chế chia sẻ bộ nhớ.

Trang 0 của P1 và trang 0 của P2 đều được ánh xạ vào #300.

Nếu P1 sửa trang 0, thì trang 0 của P2 cũng bị sửa đổi và ngược lại.

<u>Câu 2</u>:

Một hệ thống máy tính giả lập sử dụng bộ nhớ ảo áp dụng cơ chế cấp phát trang toàn cục (nghĩa là khi chọn trang nạn nhân, hệ thống có thể chọn trang của một tiến trình khác). Hệ thống có 4 khung trang, kích thước mỗi trang là 200 byte.

Trong hệ thống hiện có 2 tiến trình vừa được nạp vào (nghĩa là hệ thống chưa cấp phát khung trang nào cho tiến trình). Quá trình truy xuất bộ nhớ của 2 tiến trình lần lượt là P1.200, P1.399, P2.400, P1.2000, P2.199, P1.350, P2.499, P1.2000, P2.0.

a. Tính chuỗi truy xuất trang.
b. Mô tả quá trình cấp phát trang của hệ thống biết chiến lược được sử dụng là chiến lược tối ưu (sử dụng thông tin tương lai). Cho biết số lỗi trang.
c. Diễn giải chi tiết cách thức hệ điều hành & MMU thực hiên truy cập bộ nhớ đến 2 trường hợp truy xuất cuối cùng P1.2000, P2.0 trong câu b, biết rằng bảng trang của P1 có 10 phần tử, bảng trang của P2 có 11 phần tử.

Giải:

a) Chuỗi truy xuất trang

Chia cho địa chỉ cho 200

200	399	400	2000	199	350	499	2000	0
P_1.1	P_1.1	P_2.2	P_1.10	P_2.0	P_1.1	P_2.2	P_1.10	P_2.0

b) Chiến lược tối ưu: trang lâu dùng nhất trong tương lai.

	*		*	*	*				
	P_1.1	P_1.1	P_2.2	P_1.10	P_2.0	P_1.1	P_2.2	P_1.10	P_2.0
#0	P_1.1	P_1.1				P_1.1			
#1			P_2.2				P_2.2		
#2				P_1.10				P_1.10	
#3					P_2.0				P_2.0

*: lỗi trang ➔ 4 lỗi

c)

*) *Trường hợp P_1.2000 kế cuối*

MMU đổi P_1.2000 ➔ P_1 < p =10, d = 0 >

MMU dò bảng trong trang p = P_1.10 ứng với f = #2

MNU truy cập địa chỉ ⟨f = #2, d= 0⟩

} Coi như sai: nếu ghi là lỗi trang do trang không thuộc tiến trình mới dừng

*) *Trường hợp P_2.0 cuối*

MMU đổi P_2.0 ➔ P_1 <p = 0, d = 0>

MMU dò bảng trang P = P_1 .0 ứng với f = #3

MMU truy cập địa chỉ <f = #3, d = 0>

Câu 3:
Cho một chương trình sau:

```
Double A[10] = { 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 };
double x = 10;
void main()
{
      int l = 0, r = 9;
      while (l<=r) {
            int mid = a[(l+r)/2];
            if (x > a[mid])
                  l = mid+1;
            else if (x < a[mid])
                  r = mid-1;
            else
                  break;
      }
}
```

được thi hành trên một máy tỉnh giả lập sử dụng bộ nhớ ảo có kích thước trang là 24 byte.
Các biến l, r, mid được sử dụng là biến thanh ghi (không chiếm bộ nhớ).
Biến x được cấp phát tại địa chỉ 216, A được cấp phát tại địa chỉ 240.
Yêu cầu xác định chuỗi truy xuất trang của tiến trình (không xét các trang chứa mã thi hành).
Ghi chú: sizeof(double) = 8.