**Lớp N8 Nguyễn Hoàng Tuấn Anh+B19DCCN025 (điền ngay trước khi làm bài !)**

**Lưu bài làm này với tên tệp là tên Lớp N8 tênSV+MSV.doc.**

**Gởi vào Email** [**htcuoc@gmail.com**](mailto:htcuoc@gmail.com)**.**

**E mail đặt tiêu đề N8-tênSV+MSV. Ai không gửi vào cuối giờ, không nhận bài sau thời điểm đó !**

**Ví dụ: tệp bài làm N8-Nguyễn Văn An-** **D19CQCN0001-B.docx**

**Bài kiểm tra cuối kỳ, môn Hệ Điều Hành.**

**Học phần: Hệ điều hành (Học kỳ 1 2021-2022)**

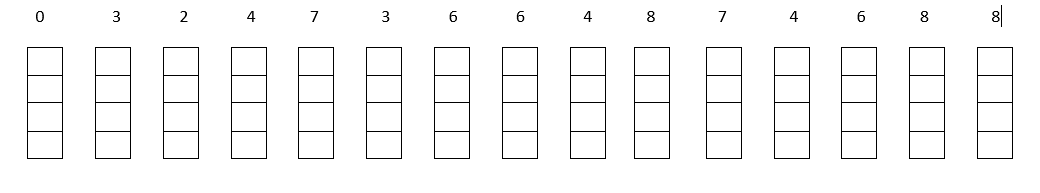
**Thời gian làm bài : 90 phút**

**Đề số 2**

**I. Điểm cho bài kiểm tra học kỳ (3 câu đầu):**

**Câu 1 (3 điểm):**

Một tiến trình được cấp 4 khung để nạp code. Thứ tự các trang nạp theo yêu cầu có trình tự như trong hình (0,3,2,4,7,….). Hãy điền số trang vào các ô khung khi cần nạp trang nếu xẩy ra tình trạng thiếu khung cho trang bằng phương pháp **Đổi trang gần đây ít sử dụng nhất tính từ lần sử dụng cuối cùng tới thời điểm đổi trang LRU**

****

**Câu 2 (3 điểm):**

1. Trình bày kỷ thuật phân chương cận kề (buddy).
2. Phân tích rõ điểm giống và khác nhau, ưu điểm của phương pháp cận kề so với phân chương cố định và phân chương động ?
3. Bộ nhớ có kích thước 2MB. Hãy dùng phương pháp cận kề để cấp bộ nhớ cho các tiến trình xuất hiện lần lượt (A,B,C,D) và có kích thước như sau: **TT A=220KB, TT B=610KB, TT C=720KB, TT D=80 KB.**

**Câu 3 (4 điểm):**

1. Trình bày khái niệm, tổ chức khi phân trang bộ nhớ.
2. Trình bày thuật ánh xạ địa chỉ khi phân trang bộ nhớ.
3. Không gian bộ nhớ logic của TT gồm có 4 trang, mỗi trang có kích thước 4096 bytes. Bộ nhớ vật lí có 64 khung. Để biểu diễn địa chỉ logic và địa chỉ vật lí cần bao nhiêu bits ?
4. **Cho một địa chỉ logic= 12046. Hãy tính địa chỉ vật lí của địa chỉ logic đó**

**dựa vào bảng trang sau:**

|  |  |
| --- | --- |
| Số trang | Số khung |
| 0 | 1 |
| 1 | 7 |
| 2 | 10 |
| 3 | 6 |

1. **Điểm cho thực hành:**

**Câu 4 (4 điểm): (Điểm tham gia thực hành có mặt 2 buổi: 6 đ và bài tập này cộng lại)**

**Tạo hai TT cùng chia sẻ code là hàm cpfiles( ). Thử nghiệm và điều chỉnh code để đảm bảo copy chính xác. Thử code với tệp nguồn có kích thước > 15 KB kiểu tệp.txt. Sau khi copy kiểm tra tại màn hình bằng lệnh *>- comp tep1 tep2*, tệp đích có giống tệp nguồn không, nếu không giống thì làm sao để copy xong, nội dung hai tệp giống nhau ?**

**#include <stdio.h>**

**#include <stdlib.h> // For exit()**

**#include <pthread.h>**

**#include <semaphore.h>**

**#include <fcntl.h>**

**#include <unistd.h>**

**#include <stdio.h>**

**#include <sys/stat.h>**

**//Các hàm hppr trợ:**

**//giải quyết tranh chấp code tới hạn:**

**//pthread\_mutex\_lock(&mutex);**

**//pthread\_mutex\_unlock(&mutex);**

**FILE \*fptr1, \*fptr2;**

**char tepnguon[100]; //lưu tên tệp nguồn từ bàn phím**

**char tepdich[100]; //lưu tên tệp đích từ bàn phím**

**char c;**

**pthread\_mutex\_t a\_mutex = PTHREAD\_MUTEX\_INITIALIZER;**

**//hàm copy tệp:**

**void\* cpfiles(void\* )**

**{**

**while(1){**

**c=fgetc(fptr1);**

**if(c==EOF){**

**break;**

**}**

**fputc(c,fptr2);**

**}**

**printf("\nThread da copy vao tep dich %s", tepdich);**

**}**

**//**

**int main () {**

**pthread\_t one, two;**

**int i;**

**printf("Nhap ten tep nguon: \n");**

**scanf("%s", tepnguon);**

**// Mo tep nguon de doc**

**fptr1 = fopen(tepnguon, "r");**

**if (fptr1 == NULL)**

**{**

**printf("Khong thay tep nguon ???? %s \n", tepnguon);**

**exit(0);**

**}**

**printf("Nhao ten tep dich: \n");**

**scanf("%s", tepdich);**

**// tao tep dich de ghi**

**fptr2 = fopen(tepdich, "w");**

**if (fptr2 == NULL)**

**{**

**printf("Khong tao duoc tep dich %s \n", tepdich);**

**exit(0);**

**}**

**//Tại đây hãy tạo hai luồng thực thi copy tệp dùng chung hàm //cpfiles( ). Thử code với tệp nguồn có kích thước > 15 KB. Sau khi //copy bằng lệnh tại màn hình lệnh *>- comp tep1 tep2*, tệp đích có //giống tệp nguồn không, nếu không giống thì làm sao để copy xong, //nội dung hai tệp giống nhau ?**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

**}**

**BÀI LÀM**

**(Làm bài trong tệp này luôn)**

**Câu 1:**

Điền số trang vào các ô khung khi cần nạp trang nếu xẩy ra tình trạng thiếu khung cho trang bằng phương pháp **Đổi trang gần đây ít sử dụng nhất tính từ lần sử dụng cuối cùng tới thời điểm đổi trang LRU**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 3 | 2 | 4 | 7 | 3 | 6 | 6 | 4 | 8 | 7 | 4 | 6 | 8 | 8 |
| 0 | 0 | 0 | 0 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
|  | 3 | 3 | 3 | 3 | 3 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
|  |  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 8 | 8 | 8 | 8 | 8 | 8 |
|  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

F F F

**Câu 2:**

1. **Trình bày kĩ thuật phân chương cận kề (buddy).**

Kỹ Thuật phân chương sử dụng phương pháp cận kề là:

* Các chương và khối trống có kích thước là lũy thừa của 2^k ( L<=k<=H): 2^L: kích thước nhỏ nhất của chương; 2^H: kích thước MEM.
* Khởi đầu, toàn bộ nhớ là một khối trống duy nhất có kích thước 2^H .

Giả sử có yêu cầu cấp phát khối nhớ kích thước s. Nếu 2^H-1 < s ≤ 2 H thì toàn khối nhớ sẽ được cấp phát. Trong trường hợp s ≤ 2^ H-1 , khối nhớ được chia đôi thành hai khối kề cận, mỗi khối có kích thước 2H^1 . So sánh tiếp, nếu 2^H-2 < s ≤ 2 ^H-1 , một trong hai khối 2^H-2 sẽ được cấp phát. Ngược lại, một trong hai khối 2^H-2 được chia đôi tiếp thành các khối kề cận kích thước 2^H-3 . Quá trình này tiếp tục cho tới khi khối nhớ nhỏ nhất có kích thước lớn hơn hoặc bằng s được tạo ra hoặc ta đạt giới hạn dưới 2^L

1. **Phân tích rõ điểm giống và khác nhau, ưu điểm của phương pháp cận kề so với phân chương cố định và phân chương động ?**

Phương pháp này có thể sử dụng kết hợp với kỹ thuật phân trang để tận dụng ưu điểm của mình.

Phân chương cố định hạn chế số lượng tiến trình trong bộ nhớ và gây phân mảnh trong.

Phân chương động, mặc dù tránh được các nhược điểm này, song tương đối phức tạp trong việc quản lý vùng trống và lựa chọn vùng trống phù hợp, đồng thời gây phân mảnh ngoài.

Phương pháp kề cận cho phép dung hoà các nhược điểm của hai phương pháp nói trên.

1. **Bộ nhớ có kích thước 2MB. Hãy dùng phương pháp cận kề để cấp bộ nhớ cho các tiến trình xuất hiện lần lượt (A,B,C,D) và có kích thước như sau: TT A=220KB, TT B=610KB, TT C=720KB, TT D=80 KB.**

**Bài làm**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2MB=2048KB | | | | | |
| A=220KB | 768KB | | | 1024KB | |
| A=220KB | B=610KB | 128KB | | 1024KB | |
| A=220KB | B=610KB | 128KB | | C=720KB | 256KB |
| A=220KB | B=610KB | D | 32KB | C=720KB | 256KB |

**Câu 3:**

**a.Trình bày khái niệm, tổ chức khi phân trang bộ nhớ.**

Khái niệm:

* Bộ nhớ vật lí được chia thành các khối nhỏ, có kích thước cố định và bằng nhau gọi là khung trang.
* Không gian địa chỉ logic tiến trình được chia thành những khối gọi là trang có kích thước bằng khung.
* Tiến trình được cấp các khung để chứa các trang của mảnh, các trang nhớ của một tiến trình không nhất thiết phải nằm liền nhau.
* Ưu điểm không phân mảnh ngoài.
* Nhược điểm bỏ phí trang do phân mảnh trong.

**b.Trình bày ánh xạ địa chỉ khi phân trang bộ nhớ:**

- để tính toán địa chỉ hiệu quả, kích thước khung được chọn là lũy thừa của 2.

- địa chỉ logic gồm 2 phần:

+ Số thứ tự trang (p)

+ Độ dịch (địa chỉ lệch) của địa chỉ so với đầu trang(o)

-Nếu kích thước trang là 2n. biễu diễn địa chỉ logic dưới dạng địa chỉ có độ dài (m+n) bit, m bit cao: biểu diễn số thứ tự trang, n bit thấp biểu diễn độ dịch trong trang nhớ

Địa chỉ logic

|  |  |
| --- | --- |
| Số thứ tự trang (p) | Độ dịch trong trang(0) |

Độ dài m N

Ví dụ: cho trang nhớ kích thước 1024B, độ dài địa chỉ lô gic 16 bit. Địa chỉ lô gic 1502, hay 0000010111011110 ở dạng nhị phân, sẽ có:

phần p = 1502/1024 = 1 tương ứng với 6 bit cao 000001

phần o = 1502 module 1024 = 478 tương ứng với 10 bit thấp 0111011110.

Như vậy ta có địa chỉ logic dưới dạng thập phân 1502 có hai phần hay dưới dạng nhị phân có hai phần là 000001|0111011110, trong đó 6 bit cao là số thứ tự trang và 10 bit thấp là độ dịch trong trang.

Cơ chế ánh xạ địa chỉ:

* Quá trình biến đổi từ địa chỉ logic sang địc chỉ vật lí được thực hiện bằng phần cứng. Cần lưu ý là việc ánh xạ địa chỉ đều phải dựa vào phần cứng vì thao tác ánh xạ cần thực hiện thường xuyên và ảnh hưởng lớn tới tốc độ toàn hệ thống
* Do việc phân trang phải dựa vào sự hỗ trợ của phần cứng khi biến đổi địa chỉ, kích thước trang và khung do phần cứng quyết định. Để thuận tiện cho việc ánh xạ địa chỉ kích thước trang luôn được chọn bằng lũy thừa của 2, và nằm trong khoảng từ 512B đến 1GB. Lưu ý rằng với kích thước trang bằng lũy thừa của 2, việc tách phần p và o trong địa chỉ lô gic được thực hiện dễ dàng bằng phép dịch bit thay vì thực hiện phép chia và chia mô đun thông thường

1. **Không gian bộ nhớ logic của TT gồm có 4 trang, mỗi trang có kích thước 4096 bytes. Bộ nhớ vật lí có 64 khung. Để biểu diễn địa chỉ logic và địa chỉ vật lí cần bao nhiêu bits ?**

Kích thước trang là 4096=2^12=> độ dịch o là 12 bit

Bộ nhớ vật lý 64 khung =2^6=> cần 6 bít để biểu diễn

Do đó cần 12+6=18 bit để biểu diễn dđịa chỉ vật lí

Không gian nhớ logic =4\*4096=2^14

Do đó cần 14 bit để biểu diễn địa chỉ logic

1. **Cho một địa chỉ logic= 15050. Hãy tính địa chỉ vật lí của địa chỉ logic đó**

**dựa vào bảng trang sau:**

|  |  |
| --- | --- |
| Số trang | Số khung |
| 0 | 1 |
| 1 | 7 |
| 2 | 10 |
| 3 | 6 |

Địa chỉ logic 15050

p =15050/4096=3-> số thứ tự trang là 3, sô khung trang là 6

o = 15050mod 4096 =2762

khi đó địa chỉ vật lí =6\*4096+2762=27338

**Câu 4:**

#include <stdio.h>

#include <stdlib.h> // For exit()

#include <pthread.h>

#include <semaphore.h>

#include <fcntl.h>

#include <unistd.h>

#include <stdio.h>

#include <sys/stat.h>

//Các hàm hppr tr?:

//gi?i quy?t tranh ch?p code t?i h?n:

//pthread\_mutex\_lock(&mutex);

//pthread\_mutex\_unlock(&mutex);

FILE \*fptr1, \*fptr2;

char tepnguon[100]; //l?u tên t?p ngu?n t? bàn phím

char tepdich[100]; //l?u tên t?p ?ích t? bàn phím

char c;

pthread\_mutex\_t a\_mutex = PTHREAD\_MUTEX\_INITIALIZER;

//hàm copy t?p:

void\* cpfiles(void\*arg )

{

while(1){

c=fgetc(fptr1);

if(c==EOF){

break;

}

fputc(c,fptr2);

}

printf("\nThread da copy vao tep dich %s", tepdich);

}

//

void thread1(void \*arg) {

pthread\_mutex\_lock(&a\_mutex);

cpfiles(arg);

pthread\_mutex\_unlock(&a\_mutex);

}

void thread2(void \*arg) {

pthread\_mutex\_lock(&a\_mutex);

cpfiles(arg);

pthread\_mutex\_unlock(&a\_mutex);

}

int main () {

pthread\_t one, two;

int i;

printf("Nhap ten tep nguon: \n");

scanf("%s", tepnguon);

// Mo tep nguon de doc

fptr1 = fopen(tepnguon, "r");

if (fptr1 == NULL)

{

printf("Khong thay tep nguon ???? %s \n", tepnguon);

exit(0);

}

printf("Nhao ten tep dich: \n");

scanf("%s", tepdich);

// tao tep dich de ghi

fptr2 = fopen(tepdich, "w");

if (fptr2 == NULL)

{

printf("Khong tao duoc tep dich %s \n", tepdich);

exit(0);

}

//T?i ?ây hãy t?o hai lu?ng th?c thi copy t?p dùng chung hàm

pthread\_create(&one, NULL, (void\*)&thread1, NULL);

pthread\_create(&two, NULL, (void\*)&thread2, NULL);

pthread\_join(one, NULL);

pthread\_join(two, NULL);

}