

Hướng dẫn Lab 4.3 – Thao tác tập tin

Mục tiêu	Lý thuyết liên quan	Tài nguyên
Truyền đối số vào thread	Ch4: Threads and Concurrency.	Sử dụng image Ubuntu 16 / 18
Truyền struct vào thread		
Thao tác tập tin	SV tự học	
Thao tác trên ma trận	Toán đại số.	

Yêu cầu sinh viên: Hiểu và thực thi các đoạn mã đã cung cấp. Hiểu các tham số được định nghĩa và các đối số được truyền vào khi tạo ra các tiểu trình. Áp dụng cho bài toán sinh ma trận ngẫu nhiên và thao tác trên ma trận.

Đánh giá sinh viên: Trả lời các vấn đề lý thuyết. Kỹ năng thực hành. Bài tập.

Yêu cầu nộp bài: các tập tin mã nguồn .c và tập tin khả thực thi .out của các Yêu cầu trong buổi thực hành và Bài tập cuối hướng dẫn trong thời gian cho phép của giảng viên.

Preferences

[1] Abraham Silberschatz, Peter B. Galvin, Greg Gagne, [2018], Operating System Concepts, 10th edition, John Wiley & Sons, New Jersey.

Programming Problems of Chapter 4.

[2] Basics of File Handling in C, geeksforgeeks.org

[3] Passing pointer as an argument to threads, [2021]

Access <https://stackoverflow.com/questions/11253025/pthread-create-not-working-passing-argument-3-warning>

Yêu cầu 1: Viết chương trình sinh ra ma trận kích thước $N \times M$ (là những hằng số cho trước) với giá trị các phần tử là một số ngẫu nhiên nguyên dương. Kết quả ghi vào tập tin text có tên gọi được truyền vào khi gọi thực thi chương trình, đồng thời được in ra màn hình.

Hướng dẫn: Tải về tập tin *LAB_IntroOS/LAB_4/task3_1.c*, biên dịch và thực thi.

```
>> gcc -o task3_1.out task3_1.c -lpthread
>> ./task3_1.out test.txt
```

- Dòng 12, 13 là định nghĩa kích thước dòng và cột của ma trận.
- Dòng 14 là giới hạn giá trị các số ngẫu nhiên sinh ra.
- Dòng 15 được khai báo để lưu trữ ma trận.
- Dòng 27 là quá trình sinh giá trị ngẫu nhiên cho ma trận.
- Dòng 30 tạo con trỏ quản lý tập tin ghi (tên được truyền qua argv[1])
- Dòng 35 tạo tiến trình nhận vào con trỏ tạo ra từ dòng 30.
- Dòng 38 đóng tập tin sau khi tiểu trình ghi hoàn tất.
- Dòng 42 là hiện thực hàm công việc của tiểu trình, có tham số là con trỏ ghi.
- Dòng 46 là ghi ra nơi con trỏ fp chỉ vào.

12	#define M 10
13	#define N 10
14	#define MAX_INT 10000
15	int a[M][N];
27	a[i][j] = rand() % MAX_INT;
29	FILE * fp;
30	fp = fopen(argv[1], "w")
35	pthread_create(& tid, & attr, writer, (void *) fp);
38	fclose(fp);
42	void * writer(void * fp) {
46	fprintf(fp, "%d ", a[i][j]);

- Tại dòng 35, thay vì truyền vào fp, ta truyền vào stdout thì kết quả chạy ra sao?
- Nếu tạo thêm 1 tiểu trình, tức là có 2 tiểu trình như sau:

```
pthread_create( & tid1, & attr, writer, (void * ) fp);
pthread_create( & tid2, & attr, writer, (void * ) stdout);
```

Kết quả sẽ như thế nào?

Yêu cầu 2: Viết chương trình đọc một hoặc nhiều ma trận kích thước $N \times M$ (là những hằng số cho trước) từ các tập tin text có tên gọi được truyền vào khi gọi thực thi chương trình.

Hướng dẫn: Tải về tập tin *LAB_IntroOS/LAB_4/task3_2.c*, biên dịch và thực thi. Giả sử rằng test.txt đã có sau khi thực hiện Yêu cầu 1 và ma trận không thay đổi kích thước.

```
>> gcc -o task3_2.out task3_2.c -lpthread
>> ./task3_2.out test.txt
```

- Dòng 17 đến 20 là khai báo cấu trúc trong đó chứa một ma trận (mảng 2 chiều) và tên tập tin lưu trữ nó trong đĩa cứng (nếu có).
- Dòng 32 là khai báo các ma trận theo nhu cầu.
- Dòng 34 cho phép người thực thi truyền tên các tập tin lưu trữ ma trận qua lời gọi thực thi.
- Dòng 36 tạo tiểu trình có nhiệm vụ đọc tập tin và lưu trữ vào ma trận thuộc về A (có địa chỉ làm tham số cuối trong lời gọi).
- Dòng 50, trong thân hàm tiểu trình, tạo ra một con trỏ fi trở đến đối tượng kiểu matrix. Lý do là param truyền đi chỉ là “một địa chỉ” chứ không định nghĩa đối tượng trở đến (đã ép thành void*).
- Dòng 51 tạo con trỏ quản lý tập tin.
- Dòng 52 mở tập tin để đọc với tên tập tin là thành phần filename của đối tượng mà fi trở đến.
- Dòng 66 gán giá trị mỗi phần tử đọc vào thành phần ma trận của đối tượng mà fi trở đến.

17	struct matrix {
18	int a[M][N];
19	char* filename;
20	};
32	struct matrix A, B;
34	A.filename = argv[1];
36	pthread_create(& tid, & attr, reader, (void *) &A);
50	struct matrix *fi = (struct matrix*) param;
51	FILE *fp;
52	fp = fopen(fi->filename, "r")
66	fi->a[i][0] = put1;

Bài tập lập trình

1. Sử dụng chương trình task3_1.out đã có sinh ra thêm vài ma trận.
2. Hoàn thiện chương trình task3_2.out thành SumMatrix.out mà cho phép người sử dụng truyền vào hai ma trận có sẵn từ hai tập tin và tính tổng (hay hiệu) rồi in ra màn hình, đồng thời ghi vào tập tin có tên được truyền vào khi gọi thực thi. Ví dụ muốn tính $C = A + B$.

```
| >> ./SumMatrix.out A.txt B.txt C.txt
```

3. * Viết chương trình để sao chép dữ liệu từ file nguồn vào file đích. Với tên file nguồn và file đích là đối số đầu vào. Đếm số kí tự đã sao chép. Giả sử chương trình chỉ chạy trên tập tin text.

```
| >./Copyfile.out source.txt target.txt  
Da sao chep thanh cong 129 ki tu.
```