

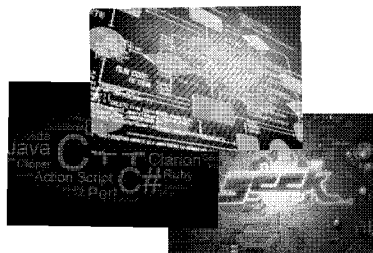


ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

NHẬP MÔN LẬP TRÌNH

CHƯƠNG VIII

LẬP TRÌNH VỚI TẬP TIN



Nguyễn Trọng Chính
chinhnt@uit.edu.vn

LẬP TRÌNH VỚI TẬP TIN

- ❖ **TẬP TIN**
- ❖ **LẬP TRÌNH VỚI TẬP TIN VĂN BẢN**
- ❖ **LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN**
- ❖ **ĐỊNH VỊ TRONG TẬP TIN (sinh viên tự đọc)**
- ❖ **CÁC HÀM QUẢN LÝ THƯ MỤC VÀ TẬP TIN (sinh viên tự đọc)**

LẬP TRÌNH VỚI TẬP TIN

❖TẬP TIN

- Tập tin là kiểu dữ liệu dùng để quản lý dữ liệu được lưu trữ trên bộ nhớ ngoài như đĩa cứng, flash drive, CD,...
- Trong C, tập tin được chia thành 2 dạng: tập tin văn bản và tập tin nhị phân.
- Tập tin văn bản chứa dữ liệu dưới dạng dãy ký tự, ký số, dấu câu, ... có thể gõ trực tiếp từ bàn phím.
- Tập tin nhị phân chứa dữ liệu có cùng dạng với dữ liệu trên bộ nhớ (dãy số nhị phân theo byte)

3

LẬP TRÌNH VỚI TẬP TIN

❖TẬP TIN

Ví dụ: để lưu số 97.

- Tập tin văn bản lưu hai ký số "97". Vì hai ký số này biểu diễn giá trị số 97.
- Tập tin nhị phân là lưu ký tự "a". Vì ký tự là "a" là biểu diễn bằng ký tự của một byte có giá trị thập phân là 97.

4

LẬP TRÌNH VỚI TẬP TIN

❖TẬP TIN

- Tập tin trong C được sử dụng dưới dạng con trỏ.

Cú pháp khai báo biến kiểu tập tin:

FILE *tên_biến;

- Quy trình xử lý dữ liệu trên bộ nhớ ngoài thông qua tập tin:

+ Mở vùng dữ liệu của tập tin.

+ Thao tác đọc/ghi dữ liệu.

+ Đẩy dữ liệu lên bộ nhớ ngoài (trường hợp ghi).

+ Đóng vùng dữ liệu của tập tin.

5

LẬP TRÌNH VỚI TẬP TIN

❖LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

* Mở vùng dữ liệu của tập tin bằng hàm fopen().

fopen trả về giá trị NULL nếu không mở được tập tin:

FILE *tên_biến;

tên_biến = fopen(tên_tập_tin, kiểu_truy_cập);

+ Tên_tập_tin: là chuỗi chứa đường dẫn và tên tập tin. Ví dụ "c:/a.txt". Lưu ý: ký tự phân cách thư mục là dấu '/' mặc dù có thể sử dụng ký tự '\\'.
+ Kiểu_truy_cập: là ký tự mô tả cách truy cập tập tin.

6

LẬP TRÌNH VỚI TẬP TIN

❖LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

+ Kiểu truy cập là chuỗi định dạng có các ý nghĩa sau:

- "wt": mở mới một tập tin để ghi (w - write) dưới dạng văn bản (t - text). Nếu tập tin chưa có thì tạo mới, nếu đã có thì xóa toàn bộ nội dung.
- "w+t": tương tự "wt" nhưng cho phép đọc.
- "rt": mở tập tin để đọc (r - read) dưới dạng văn bản (t - text), nếu chưa có tập tin thì hàm fopen trả về NULL.
- "r+t" tương tự như "rt" nhưng cho phép ghi.

7

LẬP TRÌNH VỚI TẬP TIN

❖LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

- "at" mở tập tin văn bản đã có để ghi tiếp vào phía cuối tập tin. nếu tập tin chưa có thì tạo mới.
- "a+t": tương tự như "at" nhưng cho phép đọc.

8

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

* Đọc dữ liệu từ vùng lưu trữ của tập tin với hàm `fscanf()`. Hàm `fscanf()` trả về số biến nhập được:

int n;

FILE *f;

...

n = fscanf(f, chuỗi_định_dạng, đchỉ_1, đchỉ_2, ...);

+ chuỗi_định_dạng là chuỗi chỉ định kiểu dữ liệu cần nhập tương ứng với thứ tự các biến cần nhập tương tự như trong hàm `scanf()`.

+ đchỉ_1, đchỉ_2,.. là địa chỉ các biến cần nhập.

9

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

* Đọc dữ liệu chuỗi chứa khoảng trắng từ vùng lưu trữ của tập tin với hàm `fgets()`. Hàm `fgets()` đọc đến khi hết file hoặc gặp ký tự '\n' hoặc đủ n-1 số byte đã chỉ định. Hàm trả về số byte đọc được:

int n;

FILE *f;

...

n = fgets(chuỗi, n, f);

+ n là kích thước mảng để lưu chuỗi đọc được.

+ chuỗi là mảng hoặc con trỏ kiểu ký tự.

10

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

* Ghi dữ liệu vào vùng lưu trữ của tập tin với hàm `fprintf()`. Hàm `fprintf()` trả về số byte ghi được:

```
int n;
```

```
FILE *f;
```

```
...
```

```
n = fprintf(f, chuỗi_định_dạng, biến_1, biến_2, ...);
```

+ chuỗi_định_dạng là chuỗi chỉ định kiểu dữ liệu cần nhập tương ứng với thứ tự các biến cần nhập tương tự như trong hàm `printf()`.

+ biến_1, biến_2,... là các biến chứa dữ liệu cần ghi.¹¹

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

* Ghi dữ liệu chuỗi vào vùng lưu trữ của tập tin với hàm `fputs()`. Hàm `fputs()` không thêm `'\0'` và `'\n'` vào cuối dãy ký tự cần ghi lên file, trả về số byte đã ghi:

```
int n;
```

```
FILE *f;
```

```
...
```

```
n = fputs(chuỗi, f);
```

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

* Đẩy dữ liệu ra bộ nhớ ngoài với hàm `fflush()`. Hàm `fflush()` chỉ sử dụng cho việc ghi dữ liệu:

```
FILE *f;
```

```
...
```

```
fflush(f);
```

13

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN VĂN BẢN

* Kiểm tra hết tập tin với hàm `feof()`. Hàm `feof()` trả về giá trị khác 0 nếu đã hết tập tin.

```
int ktra;
```

```
FILE *f;
```

```
...
```

```
ktra = feof(f);
```

* Đóng vùng dữ liệu của tập tin với hàm `fclose()`:

```
FILE *f;
```

```
...
```

```
fclose(f);
```

14

LẬP TRÌNH VỚI TẬP TIN

Bài tập:

- 1) Viết chương trình nhập một dãy số nguyên A có kích thước n được nhập từ bàn phím và ghi vào một tập tin văn bản với thứ tự sau: n A[0] A[1] .. A[n-1]
- 2) Viết chương trình đọc vào một dãy số nguyên A có kích thước n từ một tập tin văn bản có cấu trúc như sau: n A[0] A[1] .. A[n-1]. Sắp xếp dãy số theo thứ tự giảm dần và in ra màn hình.

15

<pre>#include <stdio.h> // bai 1 #include <stdlib.h> void main() { int n, *a, i; FILE *f; scanf("%d", &n); a = (int *) malloc(n * sizeof(int)); if (a == NULL) return; for (i = 0; i < n; i++) scanf("%d", &a[i]); f = fopen("mang.txt", "wt"); if (f == NULL) { free(a); return; } fprintf(f, "%d ", n);</pre>	<pre> for (i = 0; i < n; i++) fprintf(f, "%d ", a[i]); fflush(f); fclose(f); free(a); }</pre>
--	---

16

<pre>#include <stdio.h> // bai 2 #include <stdlib.h> void main() { int n, *a, i, j, t; FILE *f; f = fopen("mang.txt", "rt"); if (f == NULL) return; fscanf(f, "%d ", &n); a = (int *) malloc(n * sizeof(int)); if (a == NULL) { fclose(f); return; } for (i = 0; i < n; i++) fscanf(f, "%d", &a[i]); fclose(f);</pre>	<pre> for (i = 0; i < n - 1; i++) for (j = i + 1; j < n; j++) if (a[i] < a[j]) { t = a[i]; a[i] = a[j]; a[j] = t; } for (i = 0; i < n; i++) printf("%d ", a[i]); free(a); }</pre>
--	--

17

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN

* Mở vùng dữ liệu của tập tin bằng hàm fopen().
fopen trả về giá trị NULL nếu không mở được tập tin:

FILE *tên_biến;

tên_biến = fopen(tên_tập_tin, kiểu_truy_cập);

+ Tên_tập_tin: là chuỗi chứa đường dẫn và tên tập tin.

18

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN

+ Kiểu truy cập là chuỗi định dạng có các ý nghĩa sau:

- "wb": mở mới một tập tin để ghi (w - write) dưới dạng nhị phân (b - binary). Nếu tập tin chưa có thì tạo mới, nếu đã có thì xóa toàn bộ nội dung.
- "w+b": tương tự "wb" nhưng cho phép đọc.
- "rb": mở tập tin để đọc (r - read) dưới dạng nhị phân (b - binary), nếu chưa có tập tin thì hàm fopen trả về NULL.
- "r+b" tương tự như "rb" nhưng cho phép ghi.

19

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN

- "ab" mở tập tin nhị phân đã có để ghi tiếp vào phía cuối tập tin. nếu tập tin chưa có thì tạo mới.
- "a+b": tương tự như "ab" nhưng cho phép đọc.

20

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN

* Đọc dữ liệu từ vùng lưu trữ của tập tin với hàm fread(). Hàm fread() trả về byte đọc được nhập được:

int n;

FILE *f;

...

n = fread(đchỉ, kích_thước_pt, số_pt, f);

+ đchỉ là địa chỉ vùng nhớ sẽ chứa dữ liệu đọc được.

+ kích_thước_pt là kích thước theo byte của một phần tử dữ liệu. Kích thước này được xác định bằng sizeof()

21

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN

* Ghi dữ liệu vào vùng lưu trữ của tập tin với hàm fwrite(). Hàm fwrite() trả về số byte ghi được:

int n;

FILE *f;

...

n = fwrite(đchỉ, kích_thước_pt, số_pt, f);

+ đchỉ là địa chỉ vùng nhớ sẽ chứa dữ liệu đọc được.

+ kích_thước_pt là kích thước theo byte của một phần tử dữ liệu. Kích thước này được xác định bằng sizeof()

22

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN

* Đẩy dữ liệu ra bộ nhớ ngoài với hàm `fflush()`. Hàm `fflush()` chỉ sử dụng cho việc ghi dữ liệu:

```
FILE *f;
```

```
...
```

```
fflush(f);
```

23

LẬP TRÌNH VỚI TẬP TIN

❖ LẬP TRÌNH VỚI TẬP TIN NHỊ PHÂN

* Kiểm tra hết tập tin với hàm `feof()`. Hàm `feof()` trả về giá trị khác 0 nếu đã hết tập tin.

```
int ktra;
```

```
FILE *f;
```

```
...
```

```
ktra = feof(f);
```

* Đóng vùng dữ liệu của tập tin với hàm `fclose()`:

```
FILE *f;
```

```
...
```

```
fclose(f);
```

24

LẬP TRÌNH VỚI TẬP TIN

Bài tập:

3) Định nghĩa cấu trúc Nhanvien với các thông tin: tên nhân viên, năm sinh, lương tháng. Viết chương trình nhập một danh sách nhân viên có kích thước n được nhập từ bàn phím và ghi vào một tập tin nhị phân với cấu trúc: 1 int đầu tiên chứa số n nhân viên, n khối sau chứa dữ liệu của n nhân viên.

4) Viết chương trình đọc vào một danh sách nhân viên theo cấu trúc như bài 3, viết hàm sắp xếp danh sách nhân viên theo thứ tự giảm dần và in ra màn hình các thông tin nhân viên theo thứ tự đó.

25

```
#include <stdio.h> // bai 3
#include <stdlib.h>
typedef struct {
    char hoten[31];
    int namsinh;
    int luong;
} Nhanvien;
void main() {
    int n, i;
    Nhanvien *ds;
    FILE *f;
    scanf("%d", &n);scanf("%c");
    ds = (Nhanvien *) malloc(n*
        sizeof(Nhanvien));
    if (ds == NULL) return;
```

```
    for (i = 0; i < n; i++) {
        gets(ds[i].hoten);
        scanf("%d", &ds[i].namsinh);
        scanf("%d", &ds[i].luong);
        scanf("%c");scanf("%c");
    }
    f = fopen("NV.dat", "wb");
    if (f == NULL) {
        free(ds);
        return;
    }
    fwrite(&n, sizeof(int), 1, f);
    fwrite(ds, sizeof(Nhanvien),n,f);
    fflush(f); fclose(f);
    free(ds);
```

26

<pre>#include <stdio.h> //bai 4 #include <stdlib.h> typedef struct { char hoten[31]; int namsinh; int luong; } Nhanvien; void main() { int n, i, j; Nhanvien *ds, t; FILE *f; f = fopen("NV.dat", "rb"); if (f == NULL) return; fread(&n, sizeof(int), 1, f); if (n == 0) return;</pre>	<pre>ds = (Nhanvien *) malloc(n * sizeof(Nhanvien)); fread(ds, sizeof(Nhanvien), n, f); fclose(f); for (i = 0; i < n - 1; i++) for (j = i + 1; j < n; j++) if(ds[i].luong < ds[j].luong) { t = ds[i]; ds[i] = ds[j]; ds[j] = t; } for (i = 0; i < n; i++) printf("%s\t%d\t%d\n", ds[i].hoten, ds[i].namsinh, ds[i].luong); free(ds); }</pre>
---	--

27

LẬP TRÌNH VỚI TẬP TIN

Bài tập:

5) Viết chương trình cho phép lựa chọn chức năng nhập một bài thơ và lưu vào tập tin hoặc đọc bài thơ trên tập tin và in ra màn hình.

6) Viết chương trình copy file, người sử dụng sẽ nhập vào đường dẫn một file cần copy và file copy.

28

<pre>#include <stdio.h> //bai 6 #define MAX 1024 typedef char TenFile[200]; void main() { char buff[MAX]; int n; TenFile ten1, ten2; FILE *f1, *f2; gets(ten1); gets(ten2); f1 = fopen(ten1, "rb"); if(f1 == NULL) return; f2 = fopen(ten2, "wb"); if (f2 == NULL) { fclose(f1); return; } </pre>	<pre>while (!feof(f1)) { n = fread(buff, 1, MAX, f1); if (n > 0) fwrite(buff, 1, n, f2); } fflush(f2); fclose(f1); fclose(f2); }</pre>
---	---

29

ÔN TẬP

- Viết hàm tìm nghiệm nguyên của phương trình $a^5+b^5+c^5+d^5+e^5=f^5$ sao cho $0 < a \leq b \leq c \leq d \leq e \leq f \leq 75$.
- Viết chương cho phép nhập và cộng hai số nguyên có độ dài nhiều nhất 30 chữ số được truyền từ dòng lệnh.
- Viết hàm xóa chuỗi d trong chuỗi s.
- Viết hàm xoay một ma trận 5x5 theo chiều kim đồng hồ.
- Viết hàm đệ quy tìm số lớn nhất trong một mảng n số nguyên.

30

ÔN TẬP

6) Viết chương trình tìm phương án xếp 8 con hậu trên bàn cờ vua sao cho không có con hậu nào bị ăn.

7) Định nghĩa danh sách liên kết chứa thông tin sinh viên như sau: họ tên, năm sinh, điểm trung bình, thông tin sinh viên kế tiếp. Viết chương trình nhập danh sách sinh viên. Trước mỗi lần nhập thông tin sinh viên, yêu cầu người sử dụng lựa chọn 0-thoát, ngược lại tiếp tục nhập. In ra danh sách sinh viên vừa nhập.

31

```
#include <stdio.h> //bai 7
#include <stdlib.h>
typedef struct sv {
    char hoten[31];
    int namsinh;
    float tb;
    struct sv *tiep;
} SINHVIEN, *Sinhvien;
void main() {
    Sinhvien ds = NULL, t = NULL;
    Sinhvien cuoids = NULL;
    int chon;
    scanf("%d", &chon);
    scanf("%c", &ch);
    while (chon) {
```

```
        t = (Sinhvien) malloc(
            sizeof(SINHVIEN));
        gets(t->hoten);
        scanf("%d", &t->namsinh);
        scanf("%f", &t->tb);
        scanf("%c%c", &t->ch, &ch);
        t->tiep = NULL;
        if (ds == NULL)
            ds = t;
        else
            cuoids->tiep = t;
        cuoids = t;
        scanf("%d", &chon);
        scanf("%c", &ch);
    }
    t = ds;
```

32

<pre>while (t != NULL) { printf("%s\t%d\t%.1f\n", t->hoten, t->namsinh, t->tb); t = t->tiep; } t = ds->tiep; while (t != NULL) { free(ds); ds = t; t = t->tiep; } free(ds); }</pre>	33
---	----