



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TIN HỌC ĐẠI CƯƠNG

Phần II: LẬP TRÌNH C

Nội dung chính

- Chương 1: Tổng quan về ngôn ngữ C
- Chương 2: Kiểu dữ liệu và biểu thức trong C
- Chương 3: Vào ra dữ liệu
- Chương 4: Cấu trúc điều khiển
- Chương 5: Mảng, con trỏ và chuỗi ký tự
- Chương 6: Cấu trúc
- Chương 7: Hàm
- Chương 8: Tập dữ liệu

Chương 8: Tập dữ liệu

8.1. Tập và phân loại tập

- Khái niệm và phân loại
- Tập và mảng
- Tổ chức tập

8.2. Các thao tác với tập

- Khai báo
- Mở tập
- Đóng tập
- Truy nhập tập văn bản
- Truy nhập tập nhị phân

Chương 8: Tập dữ liệu

8.1. Tập và phân loại tập

- Khái niệm và phân loại
- Tập và mảng
- Tổ chức tập

8.2. Các thao tác với tập

- Khai báo
- Mở tập
- Đóng tập
- Truy nhập tập văn bản
- Truy nhập tập nhị phân

Khái niệm

- Tập (Tập tin/File):
 - Tập hợp các dữ liệu cùng kiểu
 - Có liên quan tới nhau
- Lưu trữ tập
 - Lưu trữ trên thiết bị lưu trữ ngoài
 - Có tên riêng để phân biệt
- Phân thành 2 loại
 - Tập văn bản (text file)
 - Tập nhị phân (binary file)

} Kiểu mảng?

Phân loại

- Tập văn bản
 - Được tổ chức theo từng dòng
 - Trên mỗi dòng là các ký tự ASCII hiện thị được như chữ cái, chữ số, dấu câu,...
 - Cuối mỗi dòng là các ký tự điều khiển
 - CR: Carriage Return - mã ASCII 13
 - LF: Line Feed- Mã ASCII 10
- Tập nhị phân
 - Các phần tử của tập là các số nhị phân dung mã hóa thông tin
 - Thông tin được mã hóa: số, cấu trúc dữ liệu,...

Tập dữ liệu và mảng

- Mảng

- Được lưu trong bộ nhớ → dữ liệu bị mất đi khi tắt máy
- Truy nhập trực tiếp tới một phần tử qua số hiệu
- Kích thước mảng xác định trước

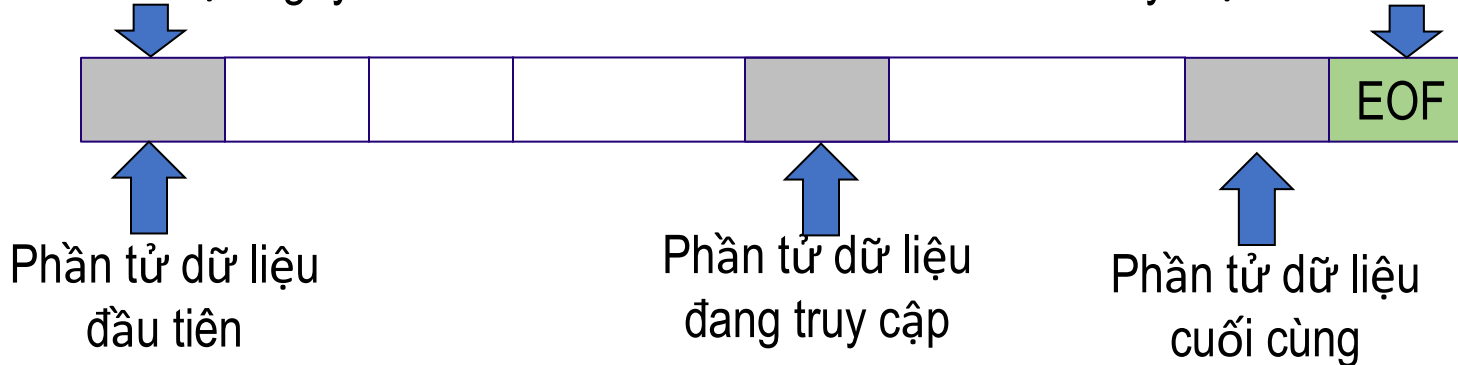
- Tập

- Lưu trữ trên thiết bị lưu trữ ngoài → dữ liệu được lưu trữ lâu dài, không bị mất đi khi tắt máy
- Không truy nhập trực tiếp qua số hiệu phần tử
- Kích thước có thể rất lớn và không cần xác định trước

Tổ chức tệp

Phần tử dữ liệu ngay sau khi mở file

Ký hiệu đánh dấu kết thúc file



- Tệp là dãy các phần tử kế tiếp nhau
 - Sử dụng phần tử đặc biệt (EOF) để đánh dấu kết thúc tệp
- Con trỏ tệp:
 - Xác định vị trí phần tử hiện có thể truy cập
 - Khi mở file, con trỏ tệp luôn ở vị trí phần tử đầu
 - Sau các thao tác đọc/ghi tệp, con trỏ file dịch chuyển về cuối tệp một không bằng số byte đã đọc/ ghi

Chương 8: Tập dữ liệu

8.1. Tập và phân loại tập

- Khái niệm và phân loại
- Vai trò của tập
- Tổ chức tập

8.2. Các thao tác với tập

- Khai báo
- Mở tập
- Đóng tập
- Truy nhập tập văn bản
- Truy nhập tập nhị phân

Quy trình

- Khai báo biến tệp
- Mở tệp để làm việc
 - Phân biệt các loại tệp và các mục đích mở tệp
- Truy nhập tệp
 - Truy nhập để đọc/ ghi/thêm mới
 - Phân biệt giữa các loại tệp
- Đóng tệp

Khai báo biến tệp

FILE * Con_Trở_Tệp

- Tệp được truy nhập qua con_trở_tệp
- Ví dụ
 - FILE * f1, *f2;

Mở tệp

`Con_Trở_Tệp = fopen(Tên_Tệp, Chế_độ_mở)`

- Hàm `fopen()` khi báo trong thư viện `stdio.h`
- `Tên_Tệp`: Kiểu chuỗi, xác định tên tệp trên đĩa
 - Tên đầy đủ của tệp hoặc tệp trên thư mục hiện thời
- `Chế độ mở`: Hằng xâu, gồm các ký tự `r/w/a/+t/b`
 - Tùy thuộc kiểu tệp và mục đích sử dụng
 - Kiểu tệp: `t`: text file; `b`: binary file
 - Chế độ: `+` Vừa đọc/ vừa ghi
 - `r` : Mở để đọc; Báo lỗi nếu tệp chưa tồn tại
 - `w` : Mở mới để ghi; Xóa nội dung tệp cũ nếu đã có
 - `a` : Mở để ghi vào cuối; Tạo tệp mới nếu chưa tồn tại
- Trả về `NULL` nếu có lỗi mở tệp

Đóng tệp

```
int fclose(FILE * Con_Trở_Tệp)
```

- Hàm fclose() khi báo trong thư viện stdio.h
- Con_trở_tệp: Tên biến tệp
- Kết quả trả về
 - 0: Nếu đóng tệp thành công
 - EOF: Nếu có lỗi

Truy nhập tệp văn bản

- Tương tự như với bàn phím/ màn hình
- Yêu cầu chỉ rõ nguồn/đích thông tin
- Các thao tác
 - Đọc dữ liệu từ tệp : `fscanf()` / `fgets()` / `getc()`
 - Ghi dữ liệu ra tệp : `fprintf()` / `fputs()` / `putc()`
 - Dịch chuyển con trỏ tệp : `fseek()` / `rewind()`
 - Kiểm tra kết thúc tệp : `feof()`
- Ví dụ:
 - `fprintf(FILE *fptr, Xâu_định_dạng [,DS giá trị])`
 - `fgets(char * Xâu_ký_tự, int n, FILE *fptr)`

Truy nhập tệp nhị phân

Đọc dữ liệu

```
int fread(void * Địa_Chỉ_Đích, int Kích_thước,  
          int số_phần_tử, FILE *fptr)
```

- Đọc từ file xác định bởi biến fptr một khối dữ liệu kích thước Số_Phần_Tử x Kích_Thước vào vùng nhớ xác định bởi Địa_Chỉ_Đích
- Nếu đọc thành công: Trả về số phần tử đọc được
- Nếu không thành công: Trả về giá trị 0

• Ví dụ:

```
int Buf[100];  
FILE * fptr = fopen("so.dat","rb");  
fread( Buf, sizeof(int), 100, fptr);
```

Truy nhập tệp nhị phân

Ghi dữ liệu

```
int fwrite(void * Đ_Chỉ_Nguồn, int Kích_thước,  
           int số_phần_tử, FILE *fptr)
```

- Ghi từ vùng nhớ xác định bởi Địa_Chỉ_nguồn một khối dữ liệu có kích thước $Số_Phần_Tử \times Kích_Thước$ ra file được xác định bởi biến `fptr`
 - Nếu ghi thành công: Trả về số phần tử đã ghi
 - Nếu không thành công: Trả về giá trị 0
- Ví dụ:

```
int Buf[100];  
FILE * fptr = fopen("so.dat","wb")  
fwrite(Buf, sizeof(int), 100, fptr);
```


Truy nhập tệp nhị phân

Dịch chuyển con trỏ file


`int fseek(FILE *fptr, long int N, int Vị_Trí_Đầu)`

– Dịch chuyển con trỏ file của file `fptr` đi một khoảng `N` so với `Vị_Trí_Đầu`

- `SEEK_SET / 0`: Vị trí đầu là đầu tệp
- `SEEK_CUR / 1`: Vị trí đầu là vị trí con trỏ file hiện thời
- `SEEK_END / 2`: Vị trí đầu là cuối tệp

– `void rewind(FILE *fptr)`: Đưa con trỏ về đầu tệp

Kiểm tra kết thúc file

`int feof(FILE *fptr)`  **Chú ý khi dùng, có thể lỗi**

– Trả về 0 nếu con trỏ file vẫn còn trỏ tới một phần tử dữ liệu, 1 nếu con trỏ file đang trỏ tới EOF

Ví dụ 1

Tạo file Songuyen.dat ghi 100 số lẻ đầu tiên.

```
#include <stdio.h>
int main() {
    FILE * f = fopen("SoNguyen.Dat", "wb");
    int i, n;
    for(i = 0; i <100; i++) {
        n = 2*i+1;
        fwrite(&n, sizeof(int), 1, f);
    }
    fclose(f);
    return 0;
}
```

Ví dụ 2

Đọc file Songuyen.dat, đưa ra màn hình các số lẻ từ vị trí số thứ 50 của file

```
#include <stdio.h>
int main() {
    FILE * f = fopen("SoNguyen.Dat", "rb");
    int n;
    fseek(f, 50*sizeof(int), SEEK_SET);
    while(!feof(f)){
        fread(&n, sizeof(int), 1, f);
        printf("%4d", n);
    }
    fclose(f);
    return 0;
}
```

Ví dụ 3

- Nhập vào từ bàn phím 100 số thực và ghi file nhị phân SO_DUONG.DAT các số thực dương trong các số đã nhập
- Đọc từ file SO_DUONG.DAT và đưa ra màn hình số thực có giá trị lớn nhất

Ví dụ 4

- Nhập danh sách từ bàn phím các thí sinh dự thi, mỗi thí sinh gồm họ tên, số báo danh, khoa dự thi và điểm thi. Dữ liệu nhập được ghi vào file ThiSinh.dat. Kết thúc nhập khi gặp một thí sinh có tên là « *** »
- Đọc từ file ThiSinh.Dat, đưa ra màn hình danh sách các thí sinh thi vào ngành CNTT có điểm thi lớn hơn 21 theo quy cách
STT Số Báo Danh Họ Tên Điểm Thi
- Từ file ThiSinh.Dat, tạo file CNTT.Dat chỉ chứa danh sách các thí sinh thi vào khoa CNTT
- Nhập vào một số báo danh, tìm trong file ThiSinh.Dat và in ra họ tên, điểm thi và khoa đăng ký của thí sinh nếu tìm thấy. Nếu không tìm thấy thí sinh thì đưa ra thông báo « không tìm thấy »

Ví dụ 4

```
#include <stdio.h>
#include <string.h>
typedef struct {
    char Ten[30];
    long SBD;
    char Khoa[10];
    float Diem;
} SinhVien;

int main() {
    FILE *f1,*f2;
    SinhVien SV;
    int i, SBD;
    //Nhap thong tin cho file ThiSinh.Dat
    //Tao file CNTT.Dat
    return 0;
}
```

Ví dụ 4

```
//Nhap thong tin cho file ThiSinh.Dat
f1 = fopen("ThiSinh.Dat", "wb");
i = 1;
do {
    printf("Thi sinh %d :\n", i);
    printf("  Ho Ten : "); fflush(stdin); fgets(SV.Ten, sizeof(SV.Ten), stdin);
    SV.Ten[strlen(SV.Ten) - 1] = '\0'; // Bỏ ký tự xuống dòng
    if(strcmp(SV.Ten, "***") == 0) break;
    printf("  So Bao Danh: "); scanf("%d", &SV.SBD);
    printf("  Khoa : "); fflush(stdin); fgets(SV.Khoa, sizeof(SV.Khoa), stdin);
    SV.Khoa[strlen(SV.Khoa) - 1] = '\0'; // Bỏ ký tự xuống dòng
    printf("  Diem : "); scanf("%f", &SV.Diem);
    fwrite(&SV, sizeof(SinhVien), 1, f1);
    i++;
} while(1);
fclose(f1);
```

Ví dụ 4

```
printf("\n\n DANH SACH BAN DAU \n");
f1 = fopen("ThiSinh.Dat", "rb");
i = 0;
while (fread(&SV, sizeof(SinhVien), 1, f1) > 0)
    printf("%-3d %-5d %-20s %-20s %-5.1f\n", ++i, SV.SBD,
        SV.Ten, SV.Khoa, SV.Diem);

printf("\n\n Thi Sinh thi CNTT tren 21.0\n");
i = 0;
rewind(f1);

while (fread(&SV, sizeof(SinhVien), 1, f1) > 0)
    if (strcmp(SV.Khoa, "CNTT") == 0 && SV.Diem > 21.0)
        printf("%-3d %-5d %-20s %-5.1f\n", ++i, SV.SBD,
            SV.Ten, SV.Diem);
```


Ví dụ 4

```
printf("\n\n Tao file CNTT.Dat\n");  
i = 0;  
rewind(f1);  
f2 = fopen("CNTT.Dat", "wb");  
while (fread(&SV, sizeof(SinhVien), 1, f1) > 0)  
    if (strcmp(SV.Khoa, "CNTT") == 0 )  
        fwrite(&SV, sizeof(SinhVien), 1, f2);  
fclose(f2);  
  
f2 = fopen("CNTT.Dat", "rb");    //doc lai file  
while (fread(&SV, sizeof(SinhVien), 1, f2) > 0)  
    printf("%-3d %-5d %-20s  %-5.1f\n", ++i, SV.SBD, SV.Ten,  
        SV.Diem);  
fclose(f2);
```

Ví dụ 4

```
printf("\n\nTim Sinh Vien\n");
printf(" So Bao Danh "); scanf("%d", &SBD);
rewind(f1);
while (fread(&SV, sizeof(SinhVien), 1, f1))
    if (SV.SBD == SBD){
        printf("Tim thay sinh vien %s", SV.Ten);
        break;
    }

if (feof(f1)) printf("Khong thay");
fclose(f1);
```