



TRƯỜNG ĐẠI HỌC THỦY LỢI
Khoa Công nghệ thông tin
Bộ môn Tin học và KTTT

NHẬP MÔN LẬP TRÌNH

Giảng viên: TS.GVC Bùi Thị Thanh Xuân

Email: xuanbtt@tlu.edu.vn

Điện thoại: 0902001581

6.1. Tập và phân loại tập

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

6.2. Các thao tác với tập

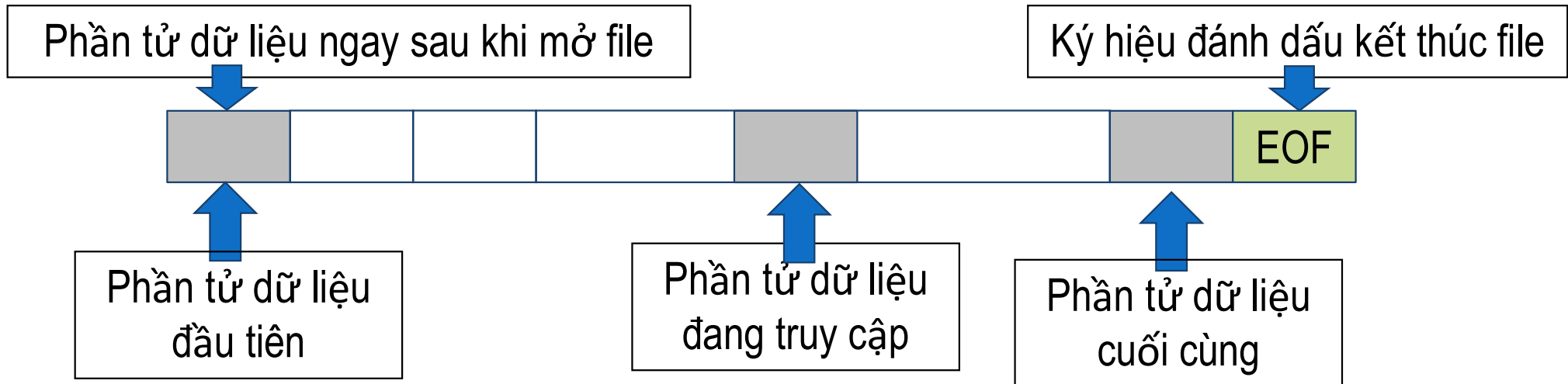
- Khai báo
- Mở tập
- Đóng tập
- Truy nhập tập văn bản
- Truy nhập tập nhị phân

- 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?

- 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

- Tập văn bản
 - Được tổ chức theo từng dòng
 - Trên mỗi dòng là các ký tự ASCII hiển 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 dùng mã hóa thông tin
 - Thông tin được mã hóa: số, cấu trúc dữ liệu,...



- 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 khoảng bằng số byte đã đọc/ ghi

6.1. Tập và phân loại tập

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

6.2. Các thao tác với tập

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

- 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


```
FILE *Con_Trở_Tệp
```

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

`Con_Trở_Tệp = fopen(Tên_Tệp, Chế_độ_mở)`

- Hàm fopen() khai 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

```
int fclose(FILE *Con_Trở_Tệp)
```

- Hàm **fclose()** khai 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

- 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()`

Đọc dữ liệu từ tệp: `fscanf()`, `fgets()` và `getc()`

1. Hàm `fscanf()` có cú pháp:

```
int fscanf(FILE *con_trò_tệp, Xâu_định_dạng, [DS_địa_chỉ]);
```

- Kết quả trả về: Nếu thực hiện thành công, hàm `fscanf()` trả về một giá trị nguyên là số byte đọc được từ tệp. Nếu thực hiện không thành công thì trả về giá trị EOF.
- Ví dụ: `fscanf(fp, "%d", &n);`

Chú ý: Trước khi dùng hàm `fscanf()` để nhập dữ liệu là kí tự, xâu kí tự từ tệp thì ta nên dùng lệnh `fflush()`. Hàm `fflush()` có cú pháp khai báo:

```
int fflush(FILE *con_trò_tệp)
```

2. Hàm `fgets()` có cú pháp:

```
char fgets(char *xâu_kí_tự, int n, FILE *con_trò_tệp);
```

- Việc đọc từ tệp sẽ dừng lại khi `fgets()` đọc đủ $n-1$ ký tự hoặc nó gặp ký tự xuống dòng, tùy gặp cái nào trước. Hàm `fgets()` tự động thêm ký tự xuống dòng “\n” và ký tự kết thúc xâu “\0” vào cuối xâu ký tự.

3. Hàm `getc()` có cú pháp:

```
int getc(FILE *con_trò_tệp);
```

Hàm `getc()` đọc từ tệp một ký tự (1 byte dữ liệu), sau chuyển đổi ký tự đó sang số nguyên int rồi lấy giá trị số nguyên thu được làm giá trị trả về của hàm.

Ghi dữ liệu lên tệp: fprintf(), fputs(), putc()

1. Hàm fprintf() có cú pháp:

```
int fprintf(FILE *con_trở_tệp, Xâu_định_dạng [, DS tham số]);
```

Ví dụ: fprintf (fptr, "%d", n);

2. Hàm fputs() có cú pháp:

```
char fputs(char *xâu_kí_tự, FILE *con_trở_tệp);
```

3. Hàm putc() có cú pháp:

```
int putc(int ch, FILE *con_trở_tệp);
```

Ghi dữ liệu lên tệp: fprintf(), fputs(), putc()

1. Hàm fprintf() có cú pháp:

```
int fprintf(FILE *con_trở_tệp, Xâu_định_dạng [, DS tham số]);
```

Ví dụ: fprintf (fptr, "%d", n);

2. Hàm fputs() có cú pháp:

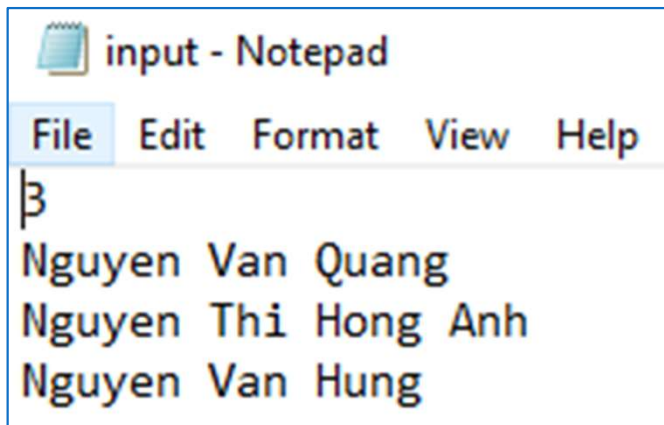
```
char fputs(char *xâu_kí_tự, FILE *con_trở_tệp);
```

3. Hàm putc() có cú pháp:

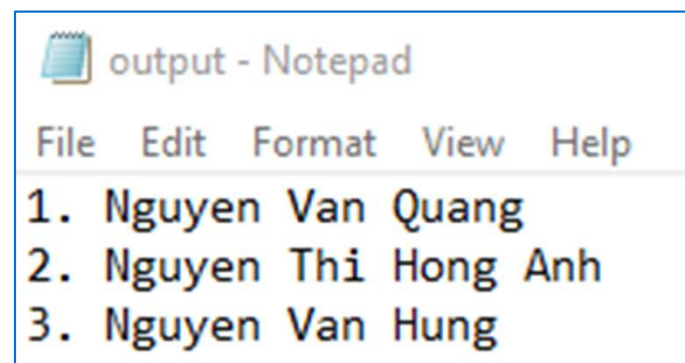
```
int putc(int ch, FILE *con_trở_tệp);
```

Truy nhập tệp văn bản

Ví dụ: Cho file input.txt, dòng đầu tiên gồm 1 số n là số lượng sinh viên, n dòng tiếp theo mỗi dòng là họ tên của 1 SV. Đọc và ghi ra file output.txt danh sách SV kèm theo số thứ tự



```
input - Notepad
File Edit Format View Help
|
Nguyen Van Quang
Nguyen Thi Hong Anh
Nguyen Van Hung
```



```
output - Notepad
File Edit Format View Help
1. Nguyen Van Quang
2. Nguyen Thi Hong Anh
3. Nguyen Van Hung
```


Truy nhập tệp văn bản

```
#include <stdio.h>
```

```
int main(){
```

```
    int n, i;
```

```
    char name[255];
```

```
    FILE *fi = fopen("input.txt", "rt");
```

```
    FILE *fo = fopen("output.txt", "wt");
```

```
    fscanf(fi, "%d", &n);
```

```
    fgets(name, 255, fi);
```

```
    for(i = 0; i < n; i++){
```

```
        fgets(name, 255, fi);
```

```
        fprintf(fo, "%d. %s", (i+1), name);
```

```
    }
```

```
    fclose(fi);
```

```
    fclose(fo);
```

```
    return 0;
```

```
}
```

```
// khai bao bien ten
```

```
// mo file de doc
```

```
// mo file de ghi
```

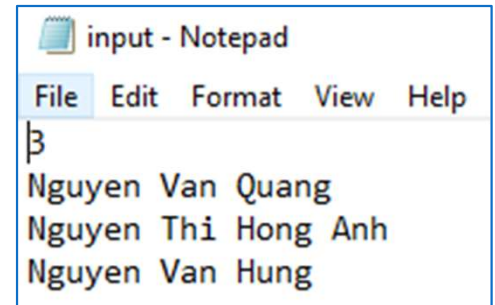
```
// doc so n tu file fi
```

```
//doc chuoai gom 255 ky tu tu file fi
```

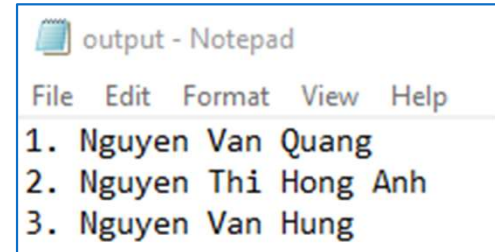
```
//ghi chuoai ra file fo
```

```
//dong file fi
```

```
//dong file fo
```



```
input - Notepad
File Edit Format View Help
|
Nguyen Van Quang
Nguyen Thi Hong Anh
Nguyen Van Hung
```



```
output - Notepad
File Edit Format View Help
1. Nguyen Van Quang
2. Nguyen Thi Hong Anh
3. Nguyen Van Hung
```

Đọc dữ liệu

```
int fread(void *Địa_Chỉ_Đích, int Kích_thước,  
          int Số_Phần_Tử, FILE *Con_trỏ_tệp)
```

- Đọc từ file xác định bởi biến `Con_trỏ_tệp` 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);
```

Ghi dữ liệu

```
int fwrite(void *Địa_Chỉ_Nguồn, int Kích_thước,  
           int số_phần_tử, FILE * Con_trỏ_tệp)
```

- Ghi từ vùng nhớ xác định bởi Địa_Chỉ_nguồn một khối dữ liệu có kích thước Số_Phần_Tử x Kích_Thước ra file được xác định bởi biến Con_trỏ_tệp.
 - 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);
```

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

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;
}
```

Đọc file SoNguyen.Dat, đưa ra màn hình các số lẻ của file

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

```
 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35
37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71
73 75 77 79 81 83 85 87 89 91 93 95 97 99 101 103 105 107
109 111 113 115 117 119 121 123 125 127 129 131 133 135 137 139 141 143
145 147 149 151 153 155 157 159 161 163 165 167 169 171 173 175 177 179
181 183 185 187 189 191 193 195 197 199 199
-----
Process exited after 0.4859 seconds with return value 0
Press any key to continue . . .
```

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;
}
```

```
101 103 105 107 109 111 113 115 117 119 121 123 125 127 129
131 133 135 137 139 141 143 145 147 149 151 153 155 157 159
161 163 165 167 169 171 173 175 177 179 181 183 185 187 189
191 193 195 197 199 199
-----
Process exited after 0.4991 seconds with return value 0
Press any key to continue . . .
```

- Nhập vào từ bàn phím 10 số thực và ghi file nhị phân SOTHUC.DAT các số thực đã nhập.
- Đọc từ file SOTHUC.DAT và đưa ra màn hình số thực có giá trị lớn nhất

```
1  #include <stdio.h>
2  int main() {
3      FILE *f = fopen("SoThuc.Dat", "wb");
4      int i;
5      float x, max;
6      for(i = 0; i < 10; i++){
7          printf("Cho x = "); scanf("%f", &x);
8          fwrite(&x, sizeof(float), 1, f);
9      }
10     fclose(f);
11 }
```

```
Cho x = 3
Cho x = 7
Cho x = 6
Cho x = 8
Cho x = 9
Cho x = 1
Cho x = 3
Cho x = 4
Cho x = 6
5.500000 3.000000 7.000000 6.000000 8.000000 9.000000 1.000000 3.000000 4.000000
6.000000 6.000000
So thuc max: 9.000000
```

```
12 FILE *fo = fopen("SoThuc.Dat", "rb");
13 fseek(fo, 0, SEEK_SET);
14 fread(&x, sizeof(float), 1, fo);
15 printf("%f ", x);
16 max = x;
17 while(!feof(fo)){
18     fread(&x, sizeof(float), 1, fo);
19     printf("%f ", x);
20     if (max < x)
21         max = x;
22 }
23 printf("\nSo thuc max: %f", max);
24 fclose(fo);
25 return 0;
26 }
```


Ví dụ 3b

- 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

- 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 "Khong tim thay!"