



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

TIN HỌC ĐẠI CƯƠNG

Bài 11. Tập tin

Nội dung

1. Khái niệm cơ bản
2. Các thao tác với tập tin
3. Truy nhập tệp văn bản
4. Truy nhập trên tệp nhị phân

Nội dung

1. Khái niệm cơ bản
2. Các thao tác với tệp tin
3. Truy nhập tệp văn bản
4. Truy nhập tệp nhị phân

3

1. Khái niệm cơ bản

- Tệp tin là tập hợp các byte liên tục được lưu trữ và được gán tên gọi.
- Biến tệp tin: là một biến thuộc kiểu dữ liệu tệp tin dùng để đại diện cho một tệp tin.
- Con trỏ tệp tin: con trỏ chỉ đến vị trí của tệp tin mà tại đó việc đọc/ghi thông tin sẽ xảy ra
 - Sau khi đọc/ghi xong dữ liệu, con trỏ sẽ chuyển dịch thêm một phần tử về phía cuối tệp tin
- Ký tự kết thúc tệp tin (EOF – End Of File): ký tự cuối cùng báo kết thúc tệp tin

4

Phân loại

- Khi xử lý tệp tin chương trình có thể xem xét chuỗi byte với cách nhìn khác nhau, có những ứng xử khác nhau với dữ liệu
- Tệp văn bản:
 - Dữ liệu của tệp tin là các chuỗi ký tự được tổ chức thành dòng
 - Mỗi dòng được kết thúc bằng ký tự xuống dòng '\n'
 - Ký tự EOF: mã ASCII là 26
- Tệp nhị phân:
 - Dữ liệu tổ chức thành dãy byte liên tục
 - Ký tự EOF: byte có giá trị -1

5

Nội dung

1. Khái niệm cơ bản
2. Các thao tác với tệp tin
3. Truy nhập tệp văn bản
4. Truy nhập tệp nhị phân

6

2. Các thao tác với tệp tin

- Trong C truy nhập tệp phải thông qua con trỏ tệp.
- Các hàm thao tác với tệp tin được khai báo trong tệp tiêu đề `stdio.h`
- Một biến con trỏ tệp (*file pointer*) được khai báo như sau:
FILE *tenConTroTep;
- Ví dụ
FILE * f1, * f2;

7

Mở tệp tin

- Muốn làm việc với tệp trước hết ta phải mở tệp.
- Để mở một tệp ta dùng hàm `fopen()`
`tenConTroTep = fopen(tenTep, cheDoMo);`
- Trong đó:
 - `tenTep` là xâu ký tự chứa đường dẫn tới tệp tin
 - `cheDoMo` là xâu ký tự chỉ định chế độ mở tệp khi thao tác với tệp
 - `fopen()` trả về giá trị `NULL` nếu có lỗi mở tệp
- Ví dụ:
`FILE *fp;`
`fp = fopen ("C:\TDC\test.dat", "rb");`

8

Chế độ mở tệp

Chế độ	Mục đích sử dụng
"r"	Mở tệp đã có để đọc, không được ghi. Nếu tệp không tồn tại, hàm fopen() sẽ trả lại trạng thái lỗi.
"w"	Mở tệp đã có để đọc, không được ghi. Nếu tệp không tồn tại, hàm fopen() sẽ trả lại trạng thái lỗi.
"a"	Mở tệp để ghi thêm dữ liệu vào cuối tệp. Nếu tệp chưa tồn tại, nó sẽ được tạo mới
"r+"	Mở tệp để vừa đọc vừa ghi. Nếu tệp chưa tồn tại thì sẽ báo lỗi
"w+"	Mở tệp để vừa đọc vừa ghi. Nếu tệp đã tồn tại, nội dung của nó sẽ bị xóa hết.
"a+"	Mở tệp để ghi thêm dữ liệu vào cuối tệp. Tệp mới sẽ được tạo nếu nó chưa tồn tại.

9

Bản chất của tệp

- Khi chỉ định chế độ mở tệp, ta cũng có thể chỉ ra bản chất của tệp

Ký hiệu	Bản chất của tệp
"b"	Tệp nhị phân
"t"	Tệp văn bản

- Khi mở tệp, nếu không chỉ rõ bản chất dữ liệu của tệp thì C sẽ ngầm hiểu đó là tệp văn bản.

10

Ví dụ

- Mở một tệp văn bản để đọc
`fp = fopen("diem.txt", "rt");`
- Mở một tệp văn bản để ghi đè
`fp = fopen("diem.txt", "wt");`
- Mở một tệp văn bản để vừa đọc vừa ghi thêm
`fp = fopen("C:\TDC\diem.txt", "a+t");`
- Mở một tệp nhị phân để vừa đọc vừa ghi
`fp = fopen("D:\VD\diem.dat", "r+b");`

11

Đóng tệp tin

- Đóng tệp là đảm bảo những thay đổi dữ liệu được lưu lại trên tệp.
- Để đóng tệp ta dùng hàm `fclose()` có cú pháp khai báo
`fclose(tenConTroTep);`
- Kết quả trả về:
0 nếu đóng tệp thành công
Ngược lại nếu có lỗi

12

Các thao tác khác

- **int feof(FILE *fp)**
 - Kiểm tra vị trí con trỏ đã ở cuối tệp chưa
 - Trả về = 0 nếu sai, ≠ 0 nếu đúng
- **void rewind(FILE *fp)**
 - Di chuyển con trỏ vị trí về đầu tệp

13

Nội dung

1. Khái niệm cơ bản
2. Các thao tác với tệp tin
3. Truy nhập tệp văn bản
4. Truy nhập tệp nhị phân

14

3. Truy nhập tệp văn bản

- **int fscanf(FILE *fp, char[] xauDinhDang, danh_sach_dia_chi)**
 - Đọc dữ liệu từ tệp
 - Sử dụng tương tự scanf()
 - Trả về số lượng giá trị đọc được theo xâu định dạng nếu hàm thực hiện thành công, ngược lại trả về giá trị nhỏ hơn
 - Hạn chế: Cần biết định dạng dữ liệu trong tệp
- **int fflush(FILE *fp)**
 - Xóa bộ đệm đọc tệp
 - Nên thực hiện trước khi dùng hàm fscanf()

15

3. Truy nhập tệp văn bản(tiếp)

- **char* fgets(char[] str, int n, FILE* fp)**
 - Đọc tối đa $n-1$ ký tự, hoặc khi gặp dấu xuống dòng trong tệp fp và gán cho str
 - Tự động thêm ký tự “\n”
 - Trả về con trỏ tới xâu str nếu thành công, ngược lại trả về NULL
- **int fgetc(FILE *fp)**
 - Đọc 1 ký tự trên tệp
 - Trả về mã ASCII của ký tự nếu thành công, ngược lại trả về EOF

16

3. Truy nhập tệp văn bản(tiếp)

- **int fprintf(FILE *fp, char[]
xauDinhDang, danh_sach_tham_so)**
 - Ghi dữ liệu lên tệp
 - Tương tự printf()
 - Trả về số byte đã ghi nếu thành công, ngược lại trả về số < 0
- **int fputs(char[] str, FILE *fp)**
 - Ghi một xâu lên tệp
 - Trả về mã ASCII của ký tự cuối cùng đã ghi nếu thành công, ngược lại trả về số < 0

17

3. Truy nhập tệp văn bản(tiếp)

- **int fputc(int ch, FILE *fp)**
 - Ghi ký tự lên tệp
 - Trả về mã ASCII của ký tự đã ghi nếu thành công, ngược lại trả về EOF

18

Ví dụ - Mở và ghi lên tệp văn bản

```
fp = fopen("diem.txt", "w");
if(fp == NULL){
    printf("Error: Could not open file.");
    return 0;
}
for(i = 0; i < soSV; i++){
    nbytes = fprintf(fp, "%-4d%-25s%-4.1f%-4.1f\n",
                    bangDiem[i].stt, bangDiem[i].hoTen,
                    bangDiem[i].diemQT, bangDiem[i].diemCK);
    if (nbytes < 0){
        printf("Error: Could not write to file.");
        break;
    }
}
fclose(fp);
```

19

Ví dụ - Mở và đọc dữ liệu trên tệp văn bản

```
fp = fopen("diem.txt", "r");
if(fp == NULL){
    printf("Error: Could not open file.");
    return 0;
}

char line[1024];
while (!feof(fp))
    if(fgets(line, 1024, fp) != NULL)
        printf("%s", line);
fclose(fp)
```

20

Nội dung

1. Khái niệm cơ bản
2. Các thao tác với tệp tin
3. Truy nhập tệp văn bản
4. Truy nhập tệp nhị phân

21

4. Truy nhập tệp nhị phân

- **`int fread(void *ptr, int smem, int nmems, FILE *fp)`**
 - Đọc dữ liệu trên tệp
 - ptr trỏ tới vùng nhớ chứa dữ liệu đọc được. Vùng nhớ phải có kích thước $\geq smem * nmems$
 - smem: kích thước của 1 phần tử
 - nmems: số phần tử muốn đọc
 - Trả về tổng số phần tử đọc được. Nếu giá trị trả về nhỏ hơn nmems thì có lỗi hoặc gặp EOF

22

4. Truy nhập tệp nhị phân

- `int fwrite(void *ptr, int smem, int nmems, FILE *fp)`
 - Đọc dữ liệu trên tệp
 - ptr trỏ tới vùng nhớ chứa dữ liệu cần ghi
 - smem: kích thước của 1 phần tử muốn ghi
 - nmems: số phần tử muốn ghi
 - Trả về tổng số phần tử ghi được. Nếu giá trị trả về nhỏ hơn nmems thì có lỗi

23

Ví dụ - Mở và ghi lên tệp nhị phân

```
fp = fopen("diem.dat", "wb");
if (fp == NULL) {
    printf("Error: Could not open file.");
    return 0;
}

for (i = 0; i < soSV; i++) {
    ret = fwrite(&bangDiem[i], sizeof(DiemSV), 1, fp);
    if (ret < 1) {
        printf("Error: Could not write to file.");
        break;
    }
}
fclose(fp);
```

24

Ví dụ - Mở và dữ liệu trên tệp nhị phân

```
fp = fopen("diem.dat", "rb");
if(fp == NULL){
    printf("Error: Could not open file.");
    return 0;
}
DiemSV ketQua[MAX];
i = 0;
soSV = 0;
while (!feof(fp)){
    if(fread(&ketQua[i], sizeof(DiemSV), 1, fp) < 1)
        break;
    i++;
    soSV++;
}
fclose(fp);
```

25

Thảo luận



26