

Quản lý tệp tin - File

Chương 8

8.1. Khái niệm tệp tin

- Một tệp tin (file) là một đơn vị lưu trữ thông tin, dữ liệu ở bộ nhớ ngoài.
- Có hai loại tệp tin là tệp tin văn bản và tệp tin nhị phân.
- Khi một chương trình kết thúc bình thường, tất cả các tệp tin đều tự động đóng.
- Khi một chương trình kết thúc bất thường, các tệp tin vẫn còn mở.

8.2. Một số hàm thao tác với tệp tin

Tên	Chức năng
fopen()	Mở một tệp tin
fclose()	Đóng một tệp tin
fputc()	Ghi một ký tự vào một tệp tin
fgetc()	Đọc một ký tự từ một tệp tin
fread()	Đọc từ một tệp tin vào một vùng đệm
fwrite()	Ghi từ một vùng đệm vào tệp tin
fseek()	tìm một vị trí nào đó trong tệp tin
fprintf()	Hoạt động giống như printf(), nhưng trên một tệp tin
fscanf()	Hoạt động giống như scanf(), nhưng trên một tệp tin
feof()	Trả về true nếu đã đến cuối tệp tin
ferror()	Trả về true nếu xảy ra một lỗi
rewind()	Đặt lại con trỏ định vị trí bên trong tệp tin về đầu tệp tin
remove()	Xóa một tệp tin
fflush()	Ghi dữ liệu từ một vùng đệm bên trong vào một tệp tin xác định

8.3. Con trỏ tệp tin

- Một con trỏ tệp tin cần thiết cho việc đọc và ghi các tệp tin
- Nó là một con trỏ đến một cấu trúc chứa thông tin về tệp tin. Thông tin bao gồm tên tệp tin, vị trí hiện tại của tệp tin, liệu tệp tin có đang được đọc hay ghi, và liệu có bất kỳ lỗi nào xuất hiện hay đã đến cuối tệp tin
- Định nghĩa lấy từ stdio.h bao gồm một khai báo cấu trúc tên FILE
- Câu lệnh khai báo duy nhất cần thiết cho một con trỏ tệp tin là:

FILE *fp;

8.4. Tập tin Văn Bản

- Mở tệp văn bản
 - Hàm `fopen()` mở một tệp tin văn bản
 - Hàm `fopen()` trả về con trỏ kết hợp với tệp tin
- `FILE *fopen(const char *filename, const char *mode);`

mode	Ý nghĩa
"r"	Mở một tệp tin văn bản để đọc
"w"	Tạo một tệp tin văn bản để ghi
"a"	Nối vào một tệp tin văn bản
"r+"	Mở một tệp tin văn bản để đọc/ghi
"w+"	Tạo một tệp tin văn bản để đọc/ghi
"a+f"	Nối hoặc tạo một tệp tin văn bản để đọc/ghi

8.4. Tập tin Văn Bản

- Đóng tệp:
`int fclose(FILE *fp);`
- Ghi một ký tự vào tệp:
`int fputc(int ch, FILE *fp);`
- Đọc 1 ký tự từ tệp:
`int fgetc(FILE *fp);`
- Đọc / Ghi chuỗi ký tự:
`int fputs(const char *str, FILE *fp);`
`char *fgets(char *str, int length, FILE *fp);`

8.5. Tập tin nhị phân

- Mở tập nhị phân:

FILE *fopen(const char *filename, const char *mode);

- Đóng tập:

int fclose(FILE *fp);

mode	Ý nghĩa
"rb"	Mở một tập tin nhị phân để đọc
"wb"	Tạo một tập tin nhị phân để ghi
"ab"	Nối vào một tập tin nhị phân
"r+b"	Mở một tập tin nhị phân để đọc/ghi
"w+b"	Tạo một tập tin nhị phân để đọc/ghi
"a+b"	Nối vào một tập tin nhị phân để đọc/ghi

8.5. Tập tin nhị phân

- Hàm fread() và fwrite() là các hàm đọc hoặc ghi dữ liệu không định dạng.
- Chúng được dùng để đọc/ghi toàn bộ khối dữ liệu.
- Hầu hết các chương trình ứng dụng hữu ích đều đọc và ghi các kiểu dữ liệu do người dùng định nghĩa, đặc biệt là các cấu trúc.

**size_t fread(void *buffer, size_t num_bytes,
size_t count, FILE *fp);**

**size_t fwrite(const void *buffer, size_t num_bytes,
size_t count, FILE *fp);**

Hàm feof()

- Hàm feof() trả về true nếu đã đến cuối tệp tin, nếu không nó trả về false (0).
- Hàm này được dùng trong khi đọc dữ liệu nhị phân.
- Nguyên mẫu là:

```
int feof (FILE *fp) ;
```

Hàm fprintf() và fscanf() – 1

- Hệ thống nhập xuất có vùng đệm bao gồm các hàm **fprintf()** và **fscanf()** tương tự như hàm **printf()** và **scanf()** ngoại trừ rằng chúng thao tác trên tệp tin
- Nguyên mẫu của các hàm này là:

```
int fprintf(FILE * fp, const char *control_string, ...);
```

```
int fscanf(FILE *fp, const char *control_string, ...);
```

Hàm fprintf() và fscanf() – 2

- Mặc dù fprintf() và fscanf() là cách dễ nhất nhưng không phải luôn luôn là hiệu quả nhất
- Mỗi lời gọi phải mất thêm một khoảng thời gian overhead, vì dữ liệu được ghi theo dạng ASCII có định dạng chữ không phải theo định dạng nhị phân
- Vì vậy, nếu tốc độ và độ lớn của tệp tin là vấn đề đáng ngại, thì fread() và fwrite() sẽ là lựa chọn tốt hơn

Nhập xuất tệp văn bản với C++

- Thư viện chuẩn fstream định nghĩa 3 kiểu dữ liệu:
 - ostream: thao tác với output file
 - istream: thao tác với input file
 - fstream: thao tác với file

Nhập xuất tệp văn bản với C++

- Mở tệp

```
void open(const char* filename, ios::mode);
```

mode	Ý nghĩa
ios::in	Mở một tệp tin văn bản để đọc
ios::out	Tạo một tệp tin văn bản để ghi
ios::app	Nối vào một tệp tin văn bản

- Đóng tệp

```
void close();
```

Nhập xuất tệp văn bản với C++

- Ghi dữ liệu vào tệp – tệp mới được tạo ra trên đĩa

- Khai báo đối tượng lớp fstream:

```
fstream fp;
```

- Mở tệp để ghi:

```
fp.open(const char *filename, ios::out);
```

- Ghi dữ liệu: sử dụng toán tử xuất (<<)

```
fp<<dữ_liệu;
```

- Đóng tệp:

```
fp.close();
```

Nhập xuất tệp văn bản với C++

- Đọc dữ liệu từ tệp – tệp đã có sẵn trên đĩa

- Khai báo đối tượng lớp fstream:

```
fstream fp;
```

- Mở tệp để đọc:

```
fp.open(const char *filename, ios::in);
```

- Đọc dữ liệu: sử dụng toán tử nhập (>>)

```
fp>>biến_chứa_dữ_liệu;
```

- Đóng tệp:

```
fp.close();
```

Bài tập 01

- Tạo 1 tệp văn bản vanban.txt chứa 1 chuỗi không quá 100 ký tự.
- Viết chương trình thực hiện các yêu cầu:
 - Đọc chuỗi từ tệp vanban.txt và lưu vào biến str.
 - Hãy cho biết chuỗi str có bao nhiêu chữ cái in.
 - Hãy cho biết chuỗi str có bao nhiêu chữ cái thường.
 - Hãy cho biết chuỗi str có bao nhiêu chữ số.
 - Tạo một chuỗi mới str1 là đảo ngược của chuỗi str, hiển thị chuỗi str1 và ghi chuỗi str1 vào file văn bản chuoinguoc.txt.
 - Thay thế tất cả các chữ số trong chuỗi str bằng từ đọc nó (ví dụ 9 thay bằng chín), hiển thị lại chuỗi str.

Bài tập 02

- Viết chương trình thực hiện các yêu cầu:
 - Khởi tạo một danh sách 5 sinh viên, mỗi sinh viên gồm các thông tin: Mã SV, họ tên, ngày sinh, giới tính, điểm TK.
 - Hiển thị danh sách ra màn hình.
 - Hiển thị ra màn hình những sinh viên sinh năm 1999.
 - Chèn thêm một sinh viên mới vào vị trí thứ k trong danh sách, hiển thị lại danh sách.
 - Xóa sinh viên có mã 105, hiển thị lại danh sách.
 - Sắp xếp danh sách sinh viên theo tên với thứ tự từ điển. Hiển thị danh sách sau khi sắp xếp. Lưu danh sách đã sắp xếp vào file sorted.bnr.

Bài tập 03

- Tạo 1 tệp văn bản input.inp gồm 2 dòng: dòng 1 chứa 1 số nguyên dương n ($n \leq 20$), dòng 2 chứa n số thực mỗi số cách nhau 1 dấu cách.
- Viết chương trình thực hiện các yêu cầu:
 - Đọc dữ liệu từ tệp input.inp: dòng 1 lưu vào biến n, dòng 2 lưu vào mảng a (cấp phát động).
 - Hiển thị mảng a.
 - Cho biết mảng a có hợp lệ không? Biết mảng hợp lệ là mảng có ít nhất 5 số dương.
 - Xóa tất cả các số âm trong mảng a, hiển thị lại mảng.
 - Sắp xếp mảng theo chiều giảm dần và lưu mảng vào file output.out.