Thao tác với file

- Mục đích sử dụng
- Các hàm thao tác với file
- Các chế độ mở file
- Đường dẫn file

Mục đích sử dụng

- File giúp ta lưu trữ dữ liệu để sử dụng lâu dài về sau
- Dữ liệu lưu trong file không bị mất đi khi tắt máy tính
- Các file dữ liệu dễ dàng quản lý, chia sẻ, truyền dẫn qua internet và qua các thiết bị nhớ ngoài
- Dữ liệu trong file sẽ bị hủy khi ta xóa file hoặc ghi đè nội dung của file

Các bước thực hiện

- Khi thao tác với file ta sử dụng con trỏ cấu trúc FILE.
- Đó là một cấu trúc được xây dựng sẵn trong thư viện <stdio.h> và chứa các tính năng cần thiết cho việc thao tác với file
- Nội dung về đọc ghi file sẽ tìm hiểu 2 phần:
  - Đọc ghi file text – đọc ghi tuần tự
  - Đọc ghi file nhị phân – đọc ghi ngẫu nhiên
- Các thao tác để ghi file: tạo file -> mở file -> ghi thông tin vào file -> đóng file
- Các thao tác để đọc file: mở file -> đọc dữ liệu trong file -> đóng file

Các bước thực hiện

- Nếu quá trình mở file hoặc tạo file thất bại ta sẽ nhận được con trỏ NULL. Ta thường kiểm tra con trỏ file trước khi thực hiện các hành động khác
- Luôn chủ động đóng file ngay khi hoàn tất việc đọc ghi file để tránh lãng phí tài nguyên máy tính và các lỗi không mong muốn
- Điều gì xảy ra nếu ta quên đóng file?
- Điều gì xảy ra nếu chương trình cần truy cập một file đang bị chiếm giữ quyền truy cập bởi một chương trình khác nhưng không sử dụng file đó nữa?

Các hàm thao tác với file

- Các chuẩn vào, ra, lỗi được quản lí qua các keyword: stdin, stdout, stderr
- Sau đây là các hàm thông dụng:

Hàm nguyên mẫu và mô tả
FILE* fopen(const char *fileName, const char *mode); Dùng để mở file có tên cho ở tham số thứ nhất, chế độ mở file cho ở tham số thứ hai, tham khảo bảng bên dưới. Trả về con trỏ file trỏ đến file mở thành công. Nếu không mở được file thì sẽ trả về NULL
int fclose(FILE *file); Dùng để đóng file đang tham chiếu tới bởi con trỏ file cho trong tham số. Trả về giá trị 0 nếu thực hiện thành công và EOF hay -1 nếu thực hiện thất bại
int fscanf(FILE *const file, const char *const format, params); Dùng để đọc dữ liệu từ file với các định dạng cho trong tham số thứ hai và đẩy dữ liệu vào lưu trữ ở các tham số thứ 3 trở đi. Trả về số lượng các đối số được điền giá trị thành công. Nếu có lỗi xảy ra thì số tham số được điền sẽ luôn nhỏ hơn số lượng đối số cần điền.

Các hàm thao tác với file

- Các hàm thường dùng(tiếp):

char* fgets(char *buffer, int maxCount, FILE *file); Dùng để đọc cả dòng, tối đa maxCount – 1 kí tự trong một file nào đó trỏ tới bởi tham số thứ ba và lưu kết quả đọc vào tham số thứ nhất. Trả về chuỗi buffer nếu đọc thành công và NULL nếu đọc thất bại
int fputs(const char *buffer, FILE *file); Dùng để ghi dữ liệu trong tham số thứ nhất ra file được trỏ tới bởi tham số thứ hai. Nội dung sau khi ghi file KHÔNG tự động được xuống dòng mới. Hàm trả về một giá trị nguyên không âm nếu thực hiện thành công và -1 trong trường hợp thực hiện thất bại
int fprintf(FILE *const stream, const char* const format, params); Dùng để ghi dữ liệu ra file trỏ tới bởi con trỏ file trong tham số thứ nhất với định dạng cho trong tham số thứ hai và các tham số tương ứng từng định dạng trong tham số thứ 3. Trả về tổng số kí tự đã được ghi ra file nếu thành công và nếu thất bại trả về số âm
size_t fread(void* buffer, size_t elementSize, size_t size, FILE *stream); Dùng để đọc dữ liệu ra ở dạng nhị phân gồm mảng size phần tử mỗi phần tử có kích thước elementSize từ luồng stream và lưu vào buffer. Trả về tổng số phần tử đọc thành công
size_t fwrite(const void* ptr, size_t size, size_t count, FILE * stream); Ghi file nhị phân: ghi mảng gồm count phần tử mỗi phần tử có kích thước size byte vào vị trí hiện thời trong file được trỏ tới bởi stream. Trả về tổng số phần tử ghi vào file thành công

Các chế độ mở file

- Bảng danh sách chế độ mở file:

Chế độ mở file và mô tả
r: mở file đã tồn tại để đọc
w: tạo một file để ghi. Nếu file đã tồn tại thì xóa bỏ nội dung đã có
a: mở file để thêm nội dung vào cuối file, nội dung gốc của file sẽ được bảo toàn. Nếu file chưa tồn tại thì tạo mới
r+: mở một file đã tồn tại để update - đọc và ghi
w+: tạo một file để đọc và ghi. Nếu file đã tồn tại thì xóa bỏ nội dung của file đó
a+: mở hoặc tạo mới file để cập nhật file – đọc và ghi. Mọi thao tác ghi sẽ thêm vào cuối file. Dữ liệu gốc của file sẽ được bảo toàn
rb: mở một file đã tồn tại để đọc ở chế độ đọc file nhị phân
wb: mở hoặc tạo mới một file để thực hiện việc ghi dữ liệu nhị phân. Nếu file đã tồn tại thì xóa bỏ nội dung đã có
ab: mở hoặc tạo mới một file để thực hiện ghi thêm dữ liệu vào cuối file theo kiểu ghi nhị phân. Dữ liệu gốc trong file không bị ảnh hưởng
rb+: mở một file đã tồn tại để cập nhật ở chế độ đọc ghi nhị phân
wb+: mở hoặc tạo mới file để cập nhật ở chế độ đọc ghi nhị phân, nếu file đã tồn tại sẽ bị xóa toàn bộ nội dung
ab+: mở hoặc tạo mới file để cập nhật ở chế độ đọc ghi nhị phân. Nếu file đã tồn tại thì bảo toàn nội dung và hành động ghi chỉ thêm dữ liệu vào cuối file

Đường dẫn file

- Khi ta cung cấp đường dẫn file trong chuỗi kí tự hằng, ta thêm 1 dấu \ vào sau mỗi dấu \ đã có hoặc đổi \ thành /
- Ví dụ cho đường dẫn: D:\Ctutorial\OUTPUT.txt ta viết thành:
  - “D:\\Ctutorial\\OUTPUT.txt” hoặc
  - “D:/Ctutorial/OUTPUT.txt”

Tiếp theo

Đọc ghi file text