## File trong C

### Giới thiệu về File trong C

File là một loại “Tài liệu” nhằm lưu dữ dữ liệu và thông tin. Có rất nhiều kiểu file khác nhau nhằm lưu dữ các kiểu dữ liệu và thông tin khác nhau, ví dụ như file text, file Excel, file Json, file XML, hay là file CSV v.v…

Để có thể thao tác với các loại file này, người dùng thông thường cần các phần mềm chuyên dùng để mở, đọc ghi và lưu chúng. Tuy nhiên thì với các lập trình viên như chúng ta thì bằng cách sử dụng sử dụng ngôn ngữ C, chúng ta cũng có thể dễ dàng thực hiện các thao tác tương tự với file như vậy.

Cũng giống như mảng, chuỗi hay biến thì file trong C là một loại dữ liệu trong chương trình, và một file khi nhập vào chương trình cũng sẽ được lưu giữ tại một vùng nào đó trên bộ nhớ máy tính. Tuy nhiên khác với các loại dữ liệu khác có thể gán vào một biến để xử lý, thì để thao tác với file, chúng ta cần tạo ra một thực thể của kiểu cấu trúc FILE để chứa thông tin của file cần thao tác, sau đó sử dụng tới một con trỏ để chỉ đến vị trí của thực thể FILE chứa thông tin file đó trên bộ nhớ, qua đó thực hiện các thao tác với file thông qua các hàm có sẵn, với các chức năng cụ thể như đóng, mở, ghi hay lưu file.

Có 2 loại file chính trong C:

1. File văn bản (text file) – lưu trữ dưới dạng ký tự có thể đọc được (.txt, .csv,...).
2. File nhị phân (binary file) – lưu trữ dữ liệu dưới dạng mã hóa (.bin, .dat,...).

#### File văn bản – text files

File văn bản là file thường có đuôi là .txt. Những file này bạn có thể dễ dàng tạo ra bằng cách dùng các text editer thông dụng như Notepad, Notepad++, Sublime Text,…

Khi bạn mở các file này bằng các text editer nói trên, bạn sẽ thấy được văn bản ngay và có thể dễ dàng thao tác sửa, xóa, thêm nội dung của file này.

Kiểu file này thuận tiện cho chúng ta trong việc sử dụng hàng ngày, nhưng nó sẽ kém bảo mật và cần nhiều bộ nhớ để lưu trữ hơn.

#### File nhị phân – Binary files

File nhị phân thường có đuôi mở rộng là .bin

Thay vì lưu trữ dưới dạng văn bản thuần thúy, các file này được lưu dưới dạng nhị phân, chỉ bao gồm các số 0 và 1. Bạn cũng sẽ thấy các con số này nếu cố mở nó bằng 1 text editer kể trên.

Loại file này giúp lưu trữ được dữ liệu với kích thước lớn hơn, không thể đọc bằng các text editer thông thường và thông tin lưu trữ ở loại file được bảo mật hơn so với file văn bản.

### Các bước làm việc với File

1. Mở file (fopen()).
2. Thao tác đọc hoặc ghi file (fprintf(), fscanf(), fwrite(), fread()...).
3. Đóng file (fclose()).

#### Mở File trong C

**Sử dụng fopen() để mở file.**

|  |
| --- |
| FILE \*fopen(const char \*filename, const char \*mode); |

Trong đó:

🔹 **filename**: Tên file cần mở.<br>

🔹 **mode**: Chế độ mở file (đọc, ghi, cập nhật...).

**Chế độ mở file (mode)**

| **Mode** | **Mô tả** |
| --- | --- |
| "r" | Mở file để đọc (phải tồn tại). |
| "w" | Mở file để ghi (tạo mới hoặc xóa nội dung cũ). |
| "a" | Mở file để ghi nối tiếp (không xóa nội dung cũ). |
| "r+" | Mở file để đọc và ghi (phải tồn tại). |
| "w+" | Mở file để đọc và ghi (tạo mới hoặc xóa nội dung cũ). |
| "a+" | Mở file để đọc và ghi nối tiếp (không xóa nội dung cũ). |
| "rb", "wb", "ab" | Tương tự như trên nhưng dành cho file nhị phân. |

#### Ghi File trong C

**1. Dùng fprintf() để ghi vào file văn bản.**

|  |
| --- |
| FILE \*f = fopen("data.txt", "w");  fprintf(f, "Xin chào!\n");  fclose(f); |

**2. Dùng fputs() và fputc()**

|  |
| --- |
| FILE \*f = fopen("data.txt", "w");  fputs("Hello World!\n", f);  fputc('A', f);  // Ghi 1 ký tự vào file  fclose(f); |

**3. Ghi file nhị phân (fwrite())**

|  |
| --- |
| typedef struct {      int id;      char name[20];  } Student;  FILE \*f = fopen("data.bin", "wb");  Student sv = {1, "Nguyen Van A"};  fwrite(&sv, sizeof(Student), 1, f);  fclose(f); |

#### Đọc File trong C

**1. Dùng fscanf() để đọc file văn bản**

|  |
| --- |
| FILE \*f = fopen("data.txt", "r");  char name[20];  fscanf(f, "%s", name);  printf("%s\n", name);  fclose(f); |

**2. Dùng fgets() và fgetc()**

|  |
| --- |
| FILE \*f = fopen("data.txt", "r");  char buffer[100];  fgets(buffer, sizeof(buffer), f); // Đọc một dòng  printf("%s", buffer);  char ch = fgetc(f);  // Đọc một ký tự  printf("%c", ch);  fclose(f); |

**3. Đọc file nhị phân (fread())**

|  |
| --- |
| FILE \*f = fopen("data.bin", "rb");  Student sv;  fread(&sv, sizeof(Student), 1, f);  printf("%d - %s\n", sv.id, sv.name);  fclose(f); |

### Kiểm tra file có tồn tại không

**Dùng fopen() và kiểm tra NULL**

|  |
| --- |
| FILE \*f = fopen("data.txt", "r");  if (f == NULL) {      printf("File không tồn tại!\n");  } else {      printf("File tồn tại!\n");      fclose(f);  } |

### Ghi và Đọc danh sách Sinh viên vào File

**1. Ghi danh sách sinh viên vào file**

|  |
| --- |
| #include <stdio.h>  #include <string.h>  typedef struct {      int maSV;      char HoTen[50];      float diem;  } SinhVien;  int main() {      FILE \*f = fopen("sinhvien.txt", "w");      if (f == NULL) {          printf("Lỗi mở file!\n");          return 1;      }      SinhVien dsSV[3] = {          {1, "Nguyen Van A", 8.5},          {2, "Tran Van B", 7.0},          {3, "Le Thi C", 9.2}      };      for (int i = 0; i < 3; i++) {          fprintf(f, "%d %s %.2f\n", dsSV[i].maSV, dsSV[i].HoTen, dsSV[i].diem);      }      fclose(f);      printf("Ghi file thành công!\n");      return 0;  } |

**2.** **Đọc danh sách sinh viên từ file**

|  |
| --- |
| #include <stdio.h>  #include <string.h>  typedef struct {      int maSV;      char HoTen[50];      float diem;  } SinhVien;  int main() {      FILE \*f = fopen("sinhvien.txt", "r");      if (f == NULL) {          printf("Lỗi mở file!\n");          return 1;      }      SinhVien sv;      printf("Danh sách sinh viên:\n");      while (fscanf(f, "%d %s %f", &sv.maSV, sv.HoTen, &sv.diem) != EOF) {          printf("%d - %s - %.2f\n", sv.maSV, sv.HoTen, sv.diem);      }      fclose(f);      return 0;  } |

### Ghi và Đọc File Nhị Phân

**1. Ghi danh sách sinh viên vào file nhị phân**

|  |
| --- |
| #include <stdio.h>  #include <string.h>  typedef struct {      int maSV;      char HoTen[50];      float diem;  } SinhVien;  int main() {      FILE \*f = fopen("sinhvien.bin", "wb");      SinhVien dsSV[3] = {          {1, "Nguyen Van A", 8.5},          {2, "Tran Van B", 7.0},          {3, "Le Thi C", 9.2}      };      fwrite(dsSV, sizeof(SinhVien), 3, f);      fclose(f);      printf("Ghi file thành công!\n");      return 0;  } |

**2. Đọc danh sách sinh viên từ file nhị phân**

|  |
| --- |
| #include <stdio.h>  typedef struct {      int maSV;      char HoTen[50];      float diem;  } SinhVien;  int main() {      FILE \*f = fopen("sinhvien.bin", "rb");      SinhVien dsSV[3];      fread(dsSV, sizeof(SinhVien), 3, f);      fclose(f);      printf("Danh sách sinh viên:\n");      for (int i = 0; i < 3; i++) {          printf("%d - %s - %.2f\n", dsSV[i].maSV, dsSV[i].HoTen, dsSV[i].diem);      }      return 0;  } |

### Xóa File

**1. Dùng remove() để xóa file**

|  |
| --- |
| if (remove("data.txt") == 0) {      printf("File đã bị xóa!\n");  } else {      printf("Không thể xóa file!\n");  } |