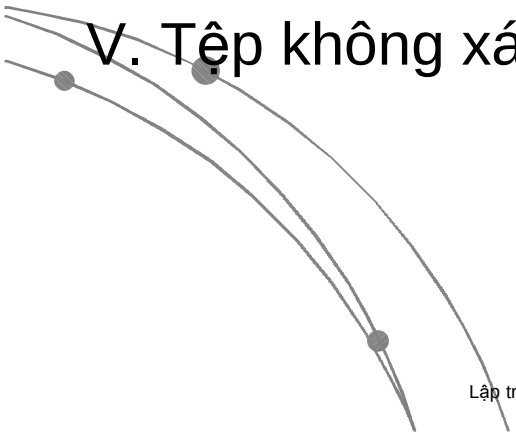
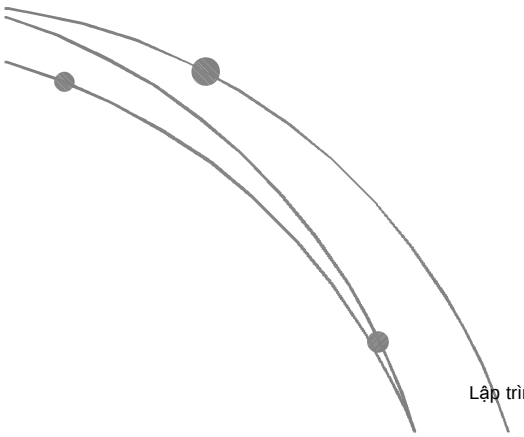
Chương 9. Kiểu dữ liệu tệp



I. Giới thiệu về tệp

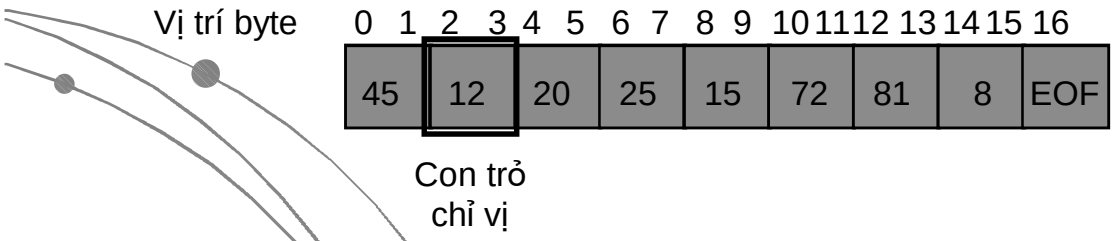


1. Khái niệm về tệp

- Kiểu tệp bao gồm một tập hữu hạn các phần tử có cùng kiểu dữ liệu được lưu trữ trên bộ nhớ ngoài.
- Số phần tử của tệp không cần xác định khi khai báo biến tệp.
- Các phần tử của tệp được lưu trữ trên bộ nhớ ngoài. Đây là đặc điểm khác với tất cả các kiểu dữ liệu khác.

2. Cấu trúc của tệp

- Các phần tử của tệp được sắp xếp thành một dãy các byte liên tiếp nhau. Sau phần tử dữ liệu cuối cùng là phần tử EOF. Phần tử này không phải là dữ liệu mà là mã kết thúc tệp.



3. Phân loại tệp

- Dựa vào cách lưu trữ dữ liệu trên tệp ta có các loại tệp sau:
  - Tệp nhị phân (binary): Dữ liệu ghi ra tệp nhị phân có dạng các byte nhị phân giống như trong bộ nhớ.
  - Tệp văn bản (text): Dữ liệu được ghi ra tệp thành các ký tự trong bảng mã ASCII. Trên tệp văn bản có mã xuống dòng gồm 2 ký tự LF (mã 10) và CR (mã 13).

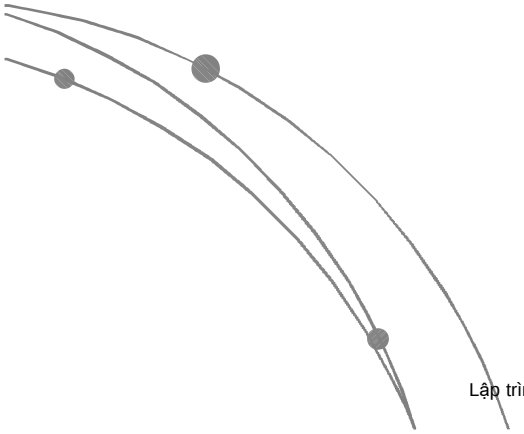
4. Khai báo tệp

- Kiểu tệp đã được trình biên dịch định nghĩa với tên chuẩn là FILE.
- Khai báo tệp ta khai báo biến con trỏ trỏ tới kiểu FILE.

  Ví dụ: FILE *f;
- Con trỏ tệp sẽ trỏ tới vùng nhớ chứa các thông tin về tệp trên bộ nhớ ngoài.

II. Tệp nhị phân

1. Ghi dữ liệu ra tệp nhị phân
2. Đọc dữ liệu từ tệp nhị phân



1. Ghi dữ liệu ra tệp nhị phân

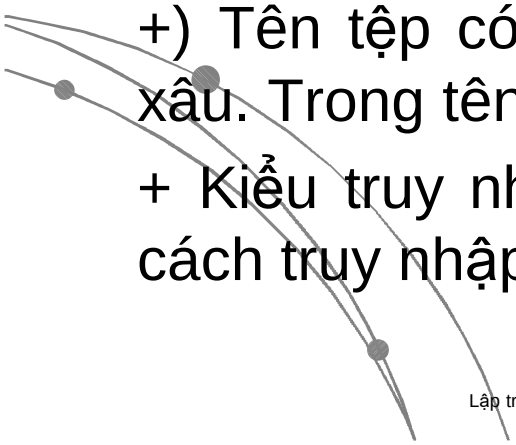
- B1: Mở tệp để ghi bằng hàm fopen()

  fp = fopen(Tên tệp, Kiểu truy nhập);

  trong đó: +) fp là con trỏ tệp được khai báo trỏ tới kiểu FILE;

  +) Tên tệp có thể là hằng xâu hoặc biến xâu. Trong tên tệp có thể có đường dẫn.

  + Kiểu truy nhập tệp là hằng xâu diễn tả cách truy nhập vào tệp.



Các kiểu truy nhập tệp nhị phân

Kiểu	Ý nghĩa
“wb”	Mở tệp mới để ghi theo kiểu nhị phân. Nếu tệp đã có nó sẽ bị xóa.
“rb”	Mở tệp mới để đọc theo kiểu nhị phân. Nếu tệp không có sẽ sinh ra lỗi.
“ab”	Mở tệp theo kiểu nhị phân để ghi bổ sung vào cuối tệp. Nếu tệp chưa có sẽ tạo tệp mới.
“r+b”	Mở tệp mới để đọc/ghi theo kiểu nhị phân. Nếu tệp không có sẽ sinh ra lỗi.
“w+b”	Mở tệp mới để đọc/ghi theo kiểu nhị phân. Nếu tệp đã có nó sẽ bị xóa.
“a+b”	Mở tệp theo kiểu nhị phân để đọc/ghi bổ sung vào cuối tệp. Nếu tệp chưa có sẽ tạo tệp mới.

1. Ghi dữ liệu ra tệp nhị phân (*tiếp*)

- B2: Ghi dữ liệu ra tệp bằng hàm fwrite()

```
fwrite(ptr, size, n, fp);
```

trong đó: +) ptr là con trỏ trỏ tới vùng nhớ chứa các phần tử dữ liệu cần ghi.

+) size là kích thước phần tử theo byte.

+) n là số phần tử cần ghi.

+) fp là con trỏ tệp.

Nếu có lỗi không ghi được, hàm trả về 0. Nếu không có lỗi hàm trả về số phần tử ghi được.

Ví dụ:

```
FILE *fp = fopen(“songuyen.dat”,”wb”);
int a=200;
fwrite(&a,sizeof(a),1,fp);
```

1. Ghi dữ liệu ra tệp nhị phân (*tiếp*)

- B3: Đóng tệp

```
fclose(fp);
```

trong đó fp là con trỏ tệp.

Ví dụ:

```
fclose(fp);
```


2. Đọc dữ liệu từ tệp nhị phân

- B1: Mở tệp để đọc bằng hàm fopen()

```
fp = fopen(Tên tệp, Kiểu truy nhập);
```

trong đó: +) fp là con trỏ tệp được khai báo trỏ tới kiểu FILE;

+) Tên tệp có thể là hằng xâu hoặc biến xâu. Trong tên tệp có thể có đường dẫn.

+ Kiểu truy nhập tệp là hằng xâu diễn tả cách truy nhập vào tệp.

2. Đọc dữ liệu từ tệp nhị phân (*tiếp*)

- B2: Đọc dữ liệu từ tệp bằng hàm fread()

fread(ptr, size, n, fp);

trong đó: +) ptr là con trỏ trỏ tới vùng nhớ chứa các phần tử dữ liệu đọc được.

+) size là kích thước phần tử theo byte.

+) n là số phần tử cần đọc.

+) fp là con trỏ tệp.

Hàm fread đọc n phần tử của tệp kể từ vị trí con trỏ chỉ vị. Hàm trả về số phần tử đọc được.

2. Đọc dữ liệu từ tệp nhị phân (*tiếp*)

- B2: Đọc dữ liệu từ tệp bằng hàm fread()

fread(ptr, size, n, fp);

Nếu con trỏ chỉ vị đã ở cuối tệp (EOF) mà vẫn đọc sẽ sinh lỗi.

Trước khi đọc tệp cần kiểm tra con trỏ chỉ vị đã ở cuối tệp chưa => dùng hàm feof(con trỏ tệp)

Nên đọc từng phần tử tệp, trước khi đọc cần kiểm tra vị trí con trỏ chỉ vị.

2. Đọc dữ liệu từ tệp nhị phân (*tiếp*)

- B3: Đóng tệp

```
fclose(fp);
```

trong đó fp là con trỏ tệp.

Ví dụ:

```
fclose(fp);
```

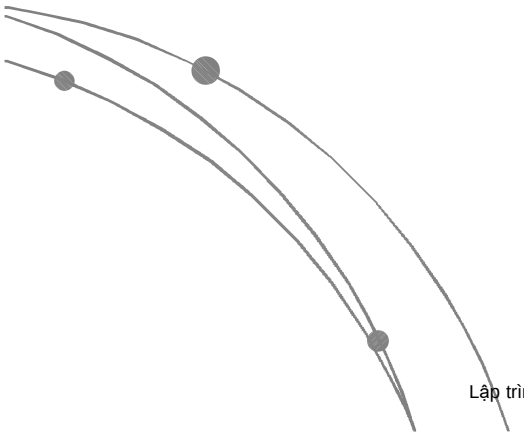

Ví dụ

Viết chương trình tạo tệp "sonuyen.dat" chứa n số nguyên nhập vào từ bàn phím. Đọc lại các số nguyên từ tệp và đưa ra màn hình.

BTVN: Tạo tệp nhị phân mathang.bin chứa thông tin về n mặt hàng, mỗi mặt hàng có thông tin về tên hàng, số lượng, đơn giá. Đọc tệp và tính tổng tiền của n mặt hàng.

III. Tệp văn bản

1. Ghi dữ liệu ra tệp văn bản
2. Đọc lại dữ liệu từ tệp văn bản



1. Ghi dữ liệu ra tệp văn bản

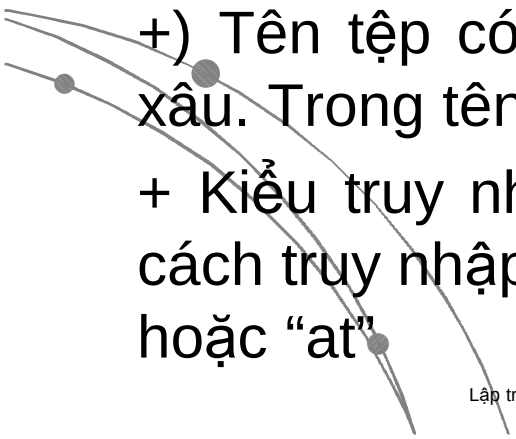
B1: Mở tệp để ghi bằng hàm fopen()

fp = fopen(Tên tệp, Kiểu truy nhập);

trong đó: +) fp là con trỏ tệp được khai báo trỏ tới kiểu FILE;

+) Tên tệp có thể là hằng xâu hoặc biến xâu. Trong tên tệp có thể có đường dẫn.

+ Kiểu truy nhập tệp là hằng xâu diễn tả cách truy nhập vào tệp. Ở đây ta dùng “wt” hoặc “at”



Các kiểu truy nhập tệp văn bản

Kiểu	Ý nghĩa
“wt”	Mở tệp mới để ghi theo kiểu văn bản. Nếu tệp đã có nó sẽ bị xóa.
“rt”	Mở tệp mới để đọc theo kiểu văn bản. Nếu tệp không có sẽ sinh ra lỗi.
“at”	Mở tệp theo kiểu văn bản để ghi bổ sung vào cuối tệp. Nếu tệp chưa có sẽ tạo tệp mới.
“r+t”	Mở tệp mới để đọc/ghi theo kiểu văn bản. Nếu tệp không có sẽ sinh ra lỗi.
“w+t”	Mở tệp mới để đọc/ghi theo kiểu văn bản. Nếu tệp đã có nó sẽ bị xóa.

1. Ghi dữ liệu ra tệp văn bản (*tiếp*)

B2: Ghi dữ liệu ra tệp văn bản

- Ghi dữ liệu theo định dạng ra tệp văn bản giống như đưa dữ liệu ra màn hình.

  fprintf(fp,dk,…);

  trong đó: +) fp là con trỏ tệp

  +) dk là hằng xâu ký tự có chứa đặc tả chuyển dạng dữ liệu.

  +) … là các đối mà giá trị của chúng cần ghi tệp.

  Ví dụ: fprintf(fp,“x= %d y= %d”,x,y);

1. Ghi dữ liệu ra tệp văn bản (*tiếp*)

B2: Ghi dữ liệu ra tệp văn bản

- Ghi cả xâu ký tự ra tệp văn bản.

fputs(s,fp);

trong đó: +) fp là con trỏ tệp

+) s là hằng xâu hoặc biến xâu.

Hàm fputs() không ghi ký tự '\0' ra tệp.

Ví dụ: fputs("Hung Yen",fp);

1. Ghi dữ liệu ra tệp văn bản (*tiếp*)

B3: Đóng tệp

fclose(fp);

trong đó fp là con trỏ tệp.

2. Đọc lại dữ liệu từ tệp văn bản

B1: Mở tệp để đọc bằng hàm fopen()

fp = fopen(Tên tệp, Kiểu truy nhập);

trong đó: +) fp là con trỏ tệp được khai báo trỏ tới kiểu FILE;

+) Tên tệp có thể là hằng xâu hoặc biến xâu. Trong tên tệp có thể có đường dẫn.

+) Kiểu truy nhập tệp là hằng xâu diễn tả cách truy nhập vào tệp. Ở đây ta dùng “rt”.

2. Đọc lại dữ liệu từ tệp văn bản (*tiếp*)

B2: Đọc lại dữ liệu từ tệp

- Đọc dữ liệu có định dạng

fscanf(fp, dk, …)

trong đó: +) fp là con trỏ tệp

+) dk là hằng xâu ký tự có chứa đặc tả chuyển dạng dữ liệu.

+) … là các địa chỉ vùng nhớ chứa dữ liệu đọc được.

Ví dụ: fscanf(fp,“%d”,&b);

2. Đọc lại dữ liệu từ tệp văn bản (*tiếp*)

B2: Đọc lại dữ liệu từ tệp

- Đọc một xâu ký tự từ tệp

fgets(s, n, fp);

trong đó: +) s là biến xâu ký tự.

+) n kích thước biến xâu, hàm này đọc tối đa n-1 ký tự từ tệp, thêm ký tự '\0' vào xâu.

+) fp là con trỏ tệp

Ví dụ: fgets(s, sizeof(s), fp);

2. Đọc lại dữ liệu từ tệp văn bản (*tiếp*)

B3: Đóng tệp

fclose(fp);

trong đó fp là con trỏ tệp.

Ví dụ

1) Viết chương trình tạo tệp văn bản 'baitho.txt' chứa n câu thơ. Đọc lại bài thơ từ tệp và đưa ra màn hình

2) Cho tệp văn bản input.txt chứa tọa độ của 2 điểm A và B trên mặt phẳng. Đọc tọa độ của 2 điểm và tính khoảng cách AB. Ghi kết quả ra tệp kq.txt

IV. Truy nhập trực tiếp các phần tử của tệp

1. Các hàm di chuyển con trỏ chỉ vị
2. Truy nhập một phần tử bất kỳ của tệp
3. Một số hàm thao tác trên tệp

1. Các hàm di chuyển con trỏ chỉ vị

- Hàm rewind(fp) chuyển con trỏ chỉ vị về phần tử đầu tiên của tệp, fp là con trỏ tệp.
- Hàm fseek(fp, sb, xp)

  trong đó: +) fp là con trỏ tệp

  +) sb là số byte cần di chuyển

  +) xp là vị trí xuất phát. xp chỉ có thể nhận một trong 3 giá trị sau:

  SEEK_SET hoặc 0: xuất phát từ đầu tệp

  SEEK_CUR hoặc 1: xuất phát từ vị trí hiện tại của con trỏ chỉ vị

  SEEK_END hoặc 2: Xuất phát từ cuối tệp.

1. Các hàm di chuyển con trỏ chỉ vị (*tiếp*)

Hàm fseek() di chuyển con trỏ chỉ vị của tệp fp từ vị trí xác định bởi xp qua số byte sb. Chiều di chuyển về cuối tệp nếu sb dương, về đầu tệp nếu sb âm.

Hàm này trả về 0 nếu di chuyển thành công, trả về giá trị khác không nếu di chuyển không thành công.

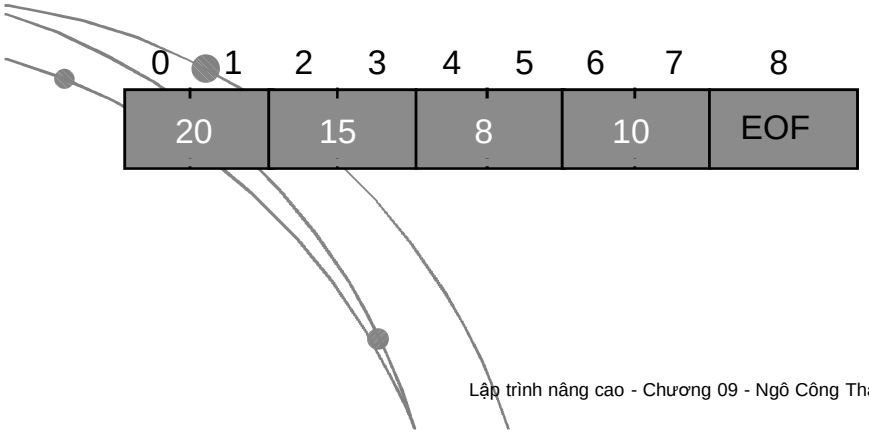
Chú ý: Không nên dùng fseek cho tệp văn bản.

1. Các hàm di chuyển con trỏ chỉ vị (*tiếp*)

- Hàm ftell(fp) cho biết vị trí hiện tại của con trỏ chỉ vị.

Ứng dụng: 1) Xác định kích thước tệp

2) Xác định số phần tử tệp



2. Truy nhập một phần tử bất kỳ của tệp

B1: Mở tệp với kiểu truy nhập là “r+b”

B2: Di chuyển con trỏ chỉ vị tới phần tử cần đọc/ghi

B3: Đọc/ghi phần tử

B4: Đóng tệp

3. Một số hàm thao tác trên tệp

- Hàm fcloseall() đóng tất cả các tệp đang mở.
- Hàm fflush(con trỏ tệp) làm sạch vùng đệm tệp.
- Hàm fflushall() làm sạch vùng đệm của tất cả các tệp đang mở.
- Hàm ferror(con trỏ tệp) kiểm tra lỗi thao tác tệp, nếu không có lỗi trả về 0, có lỗi trả về giá trị khác 0.
- Hàm remove(Tên tệp) xóa tệp
- Hàm rename(Tên tệp cũ, Tên mới);

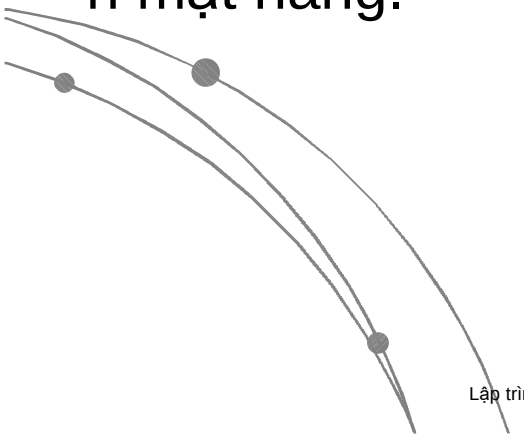
Ví dụ

1) Cho một tệp nhị phân chứa các phần tử là số nguyên. Thay thế phần tử thứ k bằng giá trị x nhập vào từ bàn phím. (1 <= k <= n, trong đó n là số phần tử trên tệp)

2) Cho một tệp nhị phân chứa các phần tử là số nguyên. Xóa phần tử thứ k (1 <= k <= n, trong đó n là số phần tử trên tệp).

Bài tập

1) Nhập vào n mặt hàng, mỗi mặt hàng có thông tin về tên hàng, số lượng, đơn giá. Lưu n mặt hàng ra tệp nhị phân mathang.dat. Đọc lại tệp, tính tổng tiền của n mặt hàng.



V-Tệp không xác định kiểu

- Coi tất cả dữ liệu như các byte nhị phân.
- Sử dụng các hàm thao tác tệp cấp thấp