

TỆP (File)

Kn: Tập là một tập hợp dữ liệu có liên quan với nhau được nhóm lại **lưu trữ trên bộ nhớ ngoài** hoặc **các thiết bị** và tồn tại ngay cả khi **chương trình kết thúc hoặc mất điện**.

- +Tập **có thể sử dụng nhiều lần**.

- +Tập được đặt bằng một tên gọi là tên tệp (DOS)

- +Có 2 loại tệp: tệp văn bản và tệp nhị phân

Con trỏ tệp:

- +Xác định **vị trí hiện tại** của tệp để đọc (hoặc ghi) trên tệp

- +Khi đọc (hoặc ghi) con trỏ tệp sẽ **tự động tăng** lên một khoảng theo **đúng số byte** vừa đọc (hoặc ghi)

- +Khi **mở tệp** thì **con trỏ tệp** luôn ở đầu tệp

TỆP (File)

Các bước xử lý trên tệp

- Khai báo biến tệp
- Mở tệp để đọc hoặc ghi
- Đọc hoặc Ghi trên tệp
- Đóng tệp

Khai báo biến tệp

```
FILE *biến_tệp;
```

Đóng tệp

```
fclose(biến_tệp);
```

Hàm kiểm tra con trỏ tệp đã ở vị trí kết thúc tệp

```
feof(biến_tệp)
```

Kết quả =1 nếu con trỏ tệp là ở vị trí cuối tệp

Kết quả =0 nếu con trỏ tệp chưa ở vị trí cuối tệp

- Mô hình của một quá trình xử lý tệp

FILE ***fptr**; // khai báo biến tệp là **fptr**

fptr = **fopen** (tên_tệp , “r”); text : w, r,
nhi phan : wb,rb

if (**fptr** != **NULL**)

{

// lệnh đọc hoặc ghi trên tệp

fclose(**fptr**); *//đóng tệp*

}

else

cout<<“ERROR: Khong mo duoc tep”;

TẬP VĂN BẢN

- Kn:
 - Là file text bao gồm các các dòng văn bản.
 - Mỗi dòng trong có ký tự kết thúc dòng là CR+LF. Kết thúc tập là ^Z
 - Có thể dùng các trình soạn thảo văn bản để đọc nội dung.
- Mở tệp
 - biến_tệp = fopen (tên_tệp, kiểu_xử_ly);**
 - tên_tệp phải đúng theo nguyên tắc đặt tên của DOS, có thể chứa cả đường dẫn
 - Nếu mở tệp thành công thì **biến_tệp ≠ NULL**, còn mở tệp không được thì **biến_tệp = NULL**
 - **kiểu_xử_ly** là kiểu xử lý tệp sau khi mở, nó là một trong các chuỗi thuộc bảng ở trang 128

TỆP VĂN BẢN

- Đọc tệp

getc (*biến_tệp*); *đọc 1 ký tự*

fgets (*str* , *n* , *biến_tệp*); *đọc nguyên 1 dòng*

*Đọc một xâu có ***n ký tự*** hoặc ***1 dòng văn bản*** từ tệp văn bản lưu vào biến *str*. Nếu chiều dài của dòng văn bản $\leq n$ thì chỉ đọc được nguyên dòng văn bản đó, nếu chiều dài dòng văn bản $> n$ thì chỉ đọc được *n* ký tự*

fscanf(*fptr*, “ *n KTĐD*”, *DSĐC n biến*); %d, %c, %f

Trong đó:

- *fptr*. biến tệp
- *DSĐC* (danh sách địa chỉ) : nếu là biến thì dùng dấu & trước biến
- *KTĐD*(*ký tự đại diện*): các ký tự đại diện cho các kiểu dữ liệu của các giá trị cần đọc từ tệp thông qua các biến

TỆP VĂN BẢN

- Ghi tệp

putc (*ch*, *biến_tệp*); // *ch* kiểu *char*

fputs (*str* , *biến_tệp*); // *str* kiểu *xâu ký tự*

fprintf(*biến_tệp*, “ *ND*+*[n KTĐD]*+ *[KTĐK]*”, *[bt₁, bt₂, ..., bt_n]*);

– **KTĐK**: (*ký tự điều khiển*): \t \a \n....

– **ND** (*nội dung*) là nội dung văn bản cần ghi lên tệp

– **KTĐD**(*ký tự đại diện*): các ký tự đại diện cho các kiểu dữ liệu của các giá trị cần ghi lên tệp thông qua các biểu thức *bt₁, bt₂, ..., bt_n*. Các ký tự đại diện được cho trong bảng các ký tự đại diện (tài liệu)

TỆP NHỊ PHÂN

Kn:

- +Tập nhị phân là tệp mà các phần tử của nó chỉ biểu diễn 1 kiểu dữ liệu. Mỗi phần tử trong tệp đúng bằng kích thước của kiểu dữ liệu mà nó lưu trữ.
- +Tập nhị phân không xem được bằng lệnh Type của DOS hoặc các trình soạn thảo văn bản

Mở tệp

biến_tệp = fopen (tên_tệp, kiểu_xử_lý);

Trong đó:

tên_tệp phải đúng theo nguyên tắc đặt Tên của DOS

Nếu mở tệp thành công thì **biến_tệp** \neq **NULL**, còn mở tệp không được thì **biến_tệp**=**NULL**

kiểu_xử_lý là kiểu xử lý tệp sau khi mở, nó là một trong các chuỗi thuộc bảng

TẬP NHỊ PHẦN

Đọc tệp

fread(&biến , kích_thước, số_phần_tử, biến_tệp);

Nếu mỗi lần **đọc tệp thành công** thì kết quả trả về là $\neq 0$

Trong đó:

&biến : địa chỉ của biến cần đọc dữ liệu từ tệp, trong trường hợp đọc một khối (mảng) thì không có dấu &

Kích_thước: kích thước của 1 phần tử cần đọc từ tệp.
Thông thường sử dụng toán tử ***sizeof(biến or kiểu)*** để xác định kích thước.

số_phần_tử: số phần tử cần đọc từ tệp. Như vậy kích thước mỗi lần đọc từ tệp là ***kích_thước*số_phần_tử***

Chú ý: Trong quá trình đọc có thể xác định xem con trỏ tệp đã ở cuối tệp chưa (dùng hàm ***feof(biến_tệp)***)

TỆP NHỊ PHÂN

Ghi tệp

***fwrite** (&biến, kích_thước, số_phần_tử, biến_tệp);*

Trong đó:

&*biến* : địa chỉ của biến cần ghi dữ liệu lên tệp, trong trường hợp ghi một khối (mảng) thì không có dấu &

Kích_thước: kích thước của 1 phần tử cần ghi vào tệp. Thông thường sử dụng toán tử **sizeof(biến or kiểu)** để xác định kích thước.

số_phần_tử: số phần tử cần ghi lên tệp. Như vậy kích thước mỗi lần ghi lên tệp là *kích_thước * số_phần_tử*

TỆP NHỊ PHÂN

Một số thao tác trên con trỏ tệp

- **rewind(biến_tệp)** : fseek(bientep,0,0)

Chuyển con trỏ về đầu tệp

- **fseek(biến_tệp, số_byte, hướng)** : Di chuyển con trỏ đến một vị trí nào đó trong tệp

trong đó:

số_byte là số byte con trỏ tệp sẽ di chuyển tính từ ***hướng*** xác định

hướng là một trong 3 giá trị sau: 0 đầu tệp, 1 tại vị trí con trỏ hiện tại, 2 cuối tệp.

- **ftell(biến_tệp)** Biết vị trí hiện tại của con trỏ file: hàm này trả về một số nguyên dài là vị trí con trỏ tệp tính từ đầu tệp.

- fseek(bientep,0,2)

- ftell(bientep)/sizeof(kieu)