HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

C PROGRAMMING LANGUAGE

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

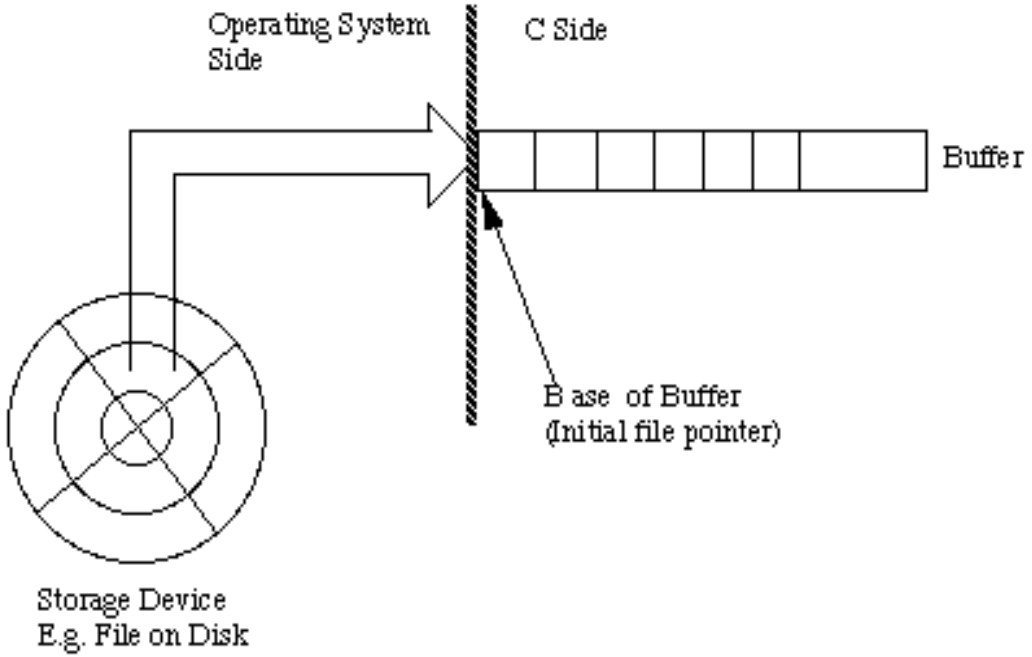
C PROGRAMMING LANGUAGE

TUẦN 14: VÀO/RA TỆP

ONE LOVE. ONE FUTURE.

Kênh xuất nhập

- Là một vùng đệm dùng cho việc nhập xuất dữ liệu ở mức cao
- Chương trình chỉ đọc và ghi dữ liệu trên vùng đệm, do đó các hàm vào ra độc lập với thiết bị đầu cuối
- Hệ điều hành đảm nhiệm việc đồng bộ hoá dữ liệu trên vùng đệm với thiết bị vào ra
- Có thể chuyển hướng vào ra của một kênh xuất nhập cho nhiều thiết bị khác nhau



Kênh xuất nhập chuẩn

- Luôn tồn tại 3 kênh xuất nhập chuẩn trong một chương trình:
  - **`stdin`** : kênh nhập chuẩn
  - **`stdout`**: kênh xuất chuẩn
  - **`stderr`**: kênh báo lỗi chuẩn
- Việc định hướng các kênh xuất nhập chuẩn này cho thiết bị nào phụ thuộc vào lúc chạy chương trình, ngầm định là bàn phím cho stdin, màn hình cho **`stdout`** và **`stderr`**
- **`scanf()`** và **`printf()`** là các hàm đọc và ghi trên các kênh **`stdin`** và **`stdout`** tương ứng
- **`perror()`** là hàm in thông báo lỗi ra kênh **`stderr`**

Ví dụ xuất nhập

Input.c

```
#include <stdio.h>
void main()
{
  int a;
  if ( scanf("%d", &a) != 1 )
    perror(“Khong phai so nguyen\n”);
  else
    printf(“Nhap so %d", a);
}
```

```
$input ↵
10 ↵
Nhap so 10
$input ↵
abc ↵
Khong phai so nguyen
$input >out.txt ↵
10 ↵
$input >out.txt ↵
abc ↵
Khong phai so nguyen
```

Chuyển hướng stdout
ra tệp out.txt

Vào ra tệp

- Tệp cần được mở trước khi sử dụng
- Mỗi tệp được gắn với một thẻ tệp khi mở
- Thao tác với tệp chỉ thông qua thẻ tệp mà không sử dụng tên tệp
- Thẻ tệp được dùng như là kênh xuất nhập cho các hàm vào ra tệp
- Cần phải đóng tệp khi kết thúc
- Các hàm thao tệp cơ bản: **`fopen(), fclose(), fscanf(), fprintf()`**.

Ví dụ

```
#include <stdio.h>

int main()
{
  FILE * out = fopen(“hello.txt”, “w”);

  if (out == NULL)
  {
    perror(“Khong the mo tep de ghi.\n”);
    return 1;
  }

  fprintf(out, “Hello world”);
  fclose(out);

  return 0;
}
```

Mở tệp để ghi

Ghi dữ liệu ra tệp

Đóng tệp khi kết thúc

Chế độ mở tệp

“r”	Đọc (tệp đã tồn tại)
“w”	Ghi đè (tệp đã tồn tại)
“a”	Ghi thêm (tệp đã tồn tại)
“r+”	Đọc và ghi (tạo mới nếu không có tệp)
“w+”	Ghi đè (tạo mới nếu không có tệp)
“a+”	Ghi thêm (tạo mới nếu không có tệp)

fprintf() và printf()

- **`fprintf(FILE * stream, …)`** hoạt động giống như **`printf(…)`** nhưng được áp dụng trên một kênh xuất dữ liệu bất kì (thường là tương ứng với một tệp)
- **`printf(…)`** là một trường hợp cụ thể tương đương với **`fprintf(stdout, …)`**
- Tương tự ta có các cặp xuất dữ liệu:
  - **`fputs(char*, FILE*)`** và **`puts(char*)`**
  - **`fputc(char, FILE*)`** và **`putchar(char)`**

fscanf() và scanf()

- **`fscanf(<stream>, …)`** dùng để đọc dữ liệu hoạt động trên một kênh nhập bất kì
- Hoạt động giống như **`scanf()`** dùng để đọc dữ liệu trên kênh stdin
- Kết quả của **`fscanf()`** và **`scanf()`** là số phần tử dữ liệu đọc được
- Tương tự ta có các cặp nhập dữ liệu:
  - **`char* fgets(char*, int maxlen, FILE*)`** và **`char*gets(char*)`**
  - **`int fgetc(FILE*)`** và **`int getchar(void)`**

Nhập dữ liệu

- Quá trình nhập liệu tương ứng với việc quét dữ liệu trên vùng đệm theo một đặc tả dữ liệu được chỉ ra
- Sau mỗi lần quét thành công con trỏ vùng đệm di chuyển để có thể quét dữ liệu kế tiếp cho những lần đọc sau
- Khi dữ liệu trong vùng đệm không còn, con trỏ vùng đệm trỏ đến vị trí EOF.
  - Để kiểm tra con trỏ có ở vị trí kết thúc hay không sử dụng hàm: **`int feof(FILE*)`**

Khuôn dạng nhập

- Nhập số theo các khuôn dạng như %d, %l, %x,..., sẽ được bỏ qua các kí tự trắng và ↵ nếu gặp tại vị trí quét
- %s cho phép quét một xâu kí tự không bao gồm kí tự trắng và ↵. Nó cũng bỏ qua kí tự trắng và ↵ nếu gặp phải ở đầu xâu khi quét
- %c cho phép nhập một kí tự bất kì tại vị trí con trỏ (kể cả kí tự trắng và ↵)
- Ví dụ nếu "`12 ab`↵" là dữ liệu trên kênh nhập
  - `"%d%s"` quét được số 12 và xâu "ab"
  - `"%d%c%s"` quét được số 12, kí tự trắng và xâu "ab"
  - `"%d %c%s"` quét được số 12, kí tự 'a' và xâu "b"
  - `"%s%s"` quét được xâu "12" và xâu "ab"
  - `"%d%s%c"` quét được số 12, xâu "ab" và kí tự ↵

fflush()

- `fflush(<stream>)` là hàm dùng để làm sạch một vùng đệm xuất hoặc nhập
- Khi một tệp được đóng, vùng đệm của nó sẽ được tự động làm sạch
- Nên sử dụng hàm `fflush()` trước khi nhập một kí tự hoặc một xâu với `gets()` hay `fgets()`
- Giống như nhập kí tự, `gets()` không bỏ qua bất cứ một kí tự nào khi quét. Hàm này quét tất cả kí tự trắng và dừng ở sau kí tự ↵ gặp đầu tiên. Tuy nhiên kí tự ↵ sẽ không nằm ở trong xâu đích.

Ví dụ

Input.c

```
#include <stdio.h>

void main()
{
  int a;
  char s[20];
  printf("Nhap so: ");
  scanf("%d", &a);

  fflush(stdin);
  printf("Nhap xau: ");
  gets(s);

  printf("So %d, xau %s", a, s);
}
```

C:\>input ↵
Nhap so: 12↵
Nhap xau: ab↵
So 12, xau ab

%d chỉ lấy hai kí tự '12' để chuyển thành số, còn dư kí tự ↵ được làm sạch bằng fflush() trước khi nhập xâu bằng gets()

Đếm số từ của một tệp

```
#include <stdio.h>

int main()
{
  int dem = 0;
  char s[80];
  FILE * f = fopen("vanban.txt", "r");
  if (f == NULL)
  {
    perror("Loi mo tep vanban.txt\n");
    return 1;
  }
  while (!feof(f))
     dem += fscanf(f, "%s", s);
  fclose(f);
  printf("Tong so tu: %d", dem);
  return 0;
}
```

Mở tệp để đọc

Mỗi lần chỉ đọc 1 từ

fgetc() và fputc()

```
FILE *input, *output;
input = fopen( "tmp.c", "r" );
output = fopen( "tmpCopy.c", "w+" );

ch = fgetc( input );
while( ch != EOF ) {
  fputc( ch, output );
  ch = fgetc( input );
}

fclose(input);
fclose(output);
```

fgets()

```
#include <stdio.h>
#define LINE_LENGTH 80

main()
{
  FILE* fp;
  char line[LINE_LENGTH];
  int count=0;
  fp=fopen("input.txt","r");
  while ( fgets(line, LINE_LENGTH, fp) != NULL)
        count++;
  printf("File contains %d lines.\n", count);
  fclose(fp);
}
```

Tệp văn bản vs. tệp nhị phân

- Không có sự khác biệt giữa các byte dữ liệu trong tệp nhị phân
- Trong tệp văn bản, các byte dữ liệu được phân biệt là kí tự hiển thị hay kí tự điều khiển
- Một tệp văn bản có thể được đánh dấu kết thúc bằng một kí tự điều khiển (ví dụ kí tự mã 26 ở hệ điều hành DOS)
- Muốn mở ở chế độ văn bản ta chỉ cần thêm kí tự ‘t’ vào cho chế độ mở, kí tự ‘b’ nếu mở ở dạng nhị phân ("r+t", "wt", "r+b",...)

Vào ra dạng nhị phân

```
size_t fread(void* buf, size_t size,
             size_t num, FILE* f);
size_t fwrite(void* buf, size_t size,
              size_t num, FILE* f);
```

- Đọc và ghi dữ liệu trên bộ nhớ có địa chỉ trỏ bởi buf, có số phần tử là num, kích thước mỗi phần tử là size

- Ví dụ:

```
int a[10];
f=fopen("songuyen.dat", "r+b");
fread(a, 10, sizeof(int), f);
```

Bài tập

Viết chương trình tạo ra tệp văn bản F3 từ việc ghép nội dung hai tệp văn bản F1 và F2.

Viết một chương trình cho phép cắt hết chú thích của một chương trình C được lưu trữ trong một tệp. Tên tệp chương trình được nhập vào từ bàn phím. Giả thiết rằng chương trình không có lỗi cú pháp.

Giả thiết một tệp dữ liệu thu thập về thời tiết trong một năm có định dạng theo mỗi dòng là:
<ngày>/<tháng> <nhiệt độ thấp nhất>-<nhiệt độ cao nhất> <độ ẩm>
1/1 11-17 70
2/1 12-17 75
...
Hãy viết chương trình đọc dữ liệu của tệp này và in ra nhiệt độ trung bình của các tháng trong năm, tháng khô hanh nhất và tháng ẩm ướt nhất.

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !