VIỆN ĐIỆN TỬ - VIỄN THÔNG

School of Electronics and telecommunications

KỸ THUẬT LẬP TRÌNH C/C++

CÁC THAO TÁC NHẬP XUẤT TRONG C

Giảng viên: TS. Nguyễn Thị Kim Thoa

Học kỳ: 20201

Nội dung

- Giới thiệu
- Các thao tác nhập dữ liệu trong C
- Các thao tác xuất dữ liệu trong C

Giới thiệu

Tầm quan trọng của các thao tác nhập/xuất dữ liệu.

- Cho phép chương trình nhập dữ liệu/xuất dữ liệu từ/đến các thiết bị vào/ra chuẩn
- Các thao tác nhập dữ liệu phải đáp ứng một số yêu cầu sau:
  - Cho phép NSD có thể nhập đầy đủ các loại dữ liệu, tối thiểu là các kiểu dữ liệu cơ bản như kiểu số, kí tự, chuỗi, logic.
  - Cho phép NSD nhập dữ liệu từ nhiều nguồn khác nhau như từ bàn phím, chuột, các cổng vào,…
  - Hỗ trợ NSD nhập dữ liệu một cách chính xác.
  - Cho phép NSD nhập dữ liệu một cách nhanh chóng và dễ dàng.

Giới thiệu

- Các thao tác xuất dữ liệu phải đáp ứng một số yêu cầu:
  - Cho phép đưa ra đầy đủ các kiểu DL mà NSD yêu cầu, tối thiểu là các kiểu DL cơ bản.
  - Cho phép dữ liệu có thể được đưa ra các thiết bị ngoại vi khác nhau, như màn hình, máy in hay đưa ra tệp để lưu trữ.
  - Dữ liệu đưa ra phải chính xác, đầy đủ và rõ ràng.

Giới thiệu

- Các thao tác vào/ra có thể chia làm hai nhóm:
  - Các thao tác vào/ra từng kí tự
  - Các thao tác vào/ra nhiều kí tự (từng chuỗi kí tự).
- Các thao tác vào/ra còn có thể được chia làm hai loại:
  - Các thao tác có định dạng
  - Các thao tác không có định dạng.
- Trong C, mỗi thao tác vào/ra được thực hiện bởi một hàm. Các hàm vào/ra cơ bản đều được khai báo và định nghĩa trong hai tệp thư viện <stdio.h> và <conio.h>.

Dòng vào stdin và các hàm scanf, gets, getchar

- stdin là dòng vào chuẩn (bàn phím).
- Các hàm scanf, gets, getchar đều nhận dữ liệu từ stdin
- Nếu trên stdin có đủ dữ liệu thì các hàm trên sẽ nhận một phần dữ liệu mà chúng yêu cầu. Phần dữ liệu nếu còn sẽ vẫn ở trên stdin.
- Nếu trên stdin không đủ dữ liệu theo yêu cầu của các hàm thì máy tạm dừng để người dùng đưa thêm dữ liệu từ bàn phím lên stdin (cho đến khi bấm phím Enter)
- Làm sạch stdin bằng lệnh **fflush(stdin)**

Bộ đệm (buffer) bàn phím

- Là một vùng bộ nhớ dành riêng để lưu giữ các kí tự nhập vào từ bàn phím trước khi chúng được các chương trình lấy ra để xử lý.
- Được sinh ra để tăng tính độc lập và đồng bộ giữa các thao tác xử lý và các thao tác vào/ra dữ liệu, vì tốc độ giữa các thao tác này khác nhau rất nhiều.
- Làm việc theo nguyên tắc hàng đợi.
- Nếu bấm một phím thì ký tự tương ứng có thể được gửi vào stdin hoặc bộ đệm bàn phím. Cách phân biệt như sau:
  - Nếu gõ phím khi máy dừng chờ trong các hàm scanf, gets, getchar thì ký tự được gửi vào stdin
  - Gõ phím trong các trường hợp khác thì ký tự gửi vào bộ đệm bàn phím

Hàm gets()

- Cú pháp: char *gets(char *s) ;
- Ý nghĩa: hàm này đọc một chuỗi kí tự từ **stdin** cho đến khi gặp '\n'. Ký tự '\n' bị loại khỏi stdin nhưng không được đặt vào chuỗi. Chuỗi kí tự đọc vào sẽ được gán cho biến s. Kí tự kết thúc chuỗi ('\0') cũng được tự động bổ sung vào cuối chuỗi s.

```
#include <stdio.h>
void main() {
    char  name[80] ;
    printf("Nhap ten :");
    gets(name) ;
    printf("Ten nhap vao la :
    %s",name);
}
```

Kết quả thực hiện chương trình:

Nhap ten : Lan ↵

Ten nhap vao la : Lan

Hàm scanf()

- Nhận dữ liệu từ **stdin**, để lại stdin ký tự '\n'
- Chức năng: Là hàm nhập dữ liệu đa năng, có định dạng.
- Cú pháp:

scanf(Hằng chuỗi định dạng, danh sách các biến);

- Trong đó :
  - Hằng chuỗi định dạng bao gồm 1 hoặc nhiều mã định dạng.
  - Danh sách các biến bao gồm một hoặc nhiều biến mà ta muốn nhập giá trị. Số lượng mã định dạng đúng bằng số lượng các biến. Mỗi mã định dạng tương ứng cho một biến vào để xác định kiểu dữ liệu và khuôn dạng dữ liệu nhập vào cho biến đó. Mỗi mã định dạng là một hằng chuỗi kí tự đặc biệt với kí tự đầu tiên luôn là %

Hàm scanf()

- Nguyên tắc hoạt động
  - Khi ấn Enter, hàm sẽ kiểm tra số lượng DL đã nhập và số tham số cần nhập DL.
  - Nếu khớp nhau thì thực hiện chuyển đổi kiểu tương ứng cho các DL đã nhập rồi gán cho các tham số cần nhập.
  - Nếu không khớp thì sẽ tiếp tục đợi nhập tiếp.

Hàm getchar()

- Dạng hàm: int getchar (void)
- Nhận một ký tự từ **stdin**, hàm trả về ký tự nhận được
- Ví dụ:

```
    int ch;
    ch=getchar();
}
```

- Nếu bấm phím
  A ↵ thì: ch='A', ký tự '\n' vẫn còn lưu trong stdin
- Nếu bấm phím
  ↵ thì: ch='\n' và ký tự '\n' bị loại khỏi stdin

Hàm getch()

- Dạng hàm: int getch (void)
- Nhận một ký tự từ bộ đệm bàn phím, **không** cho hiện lên màn hình
- Nếu có sẵn ký tự trong bộ đệm bàn phím thì hàm nhận 1 ký tự trong đó
- Nếu bộ đệm rỗng thì máy tạm dừng. Khi gõ một ký tự thì hàm nhận ngay được ký tự đó (không cần bấm thêm Enter như các hàm nhập từ stdin). Ký tự vừa gõ không được hiện lên màn hình.

Hàm getche()

- Dạng hàm: int getche (void)
- Nhận một ký tự từ bộ đệm bàn phím, cho hiện lên màn hình
- Hoạt động tương tự như hàm getch, chỉ có một điểm khác là cho ký tự được gõ hiện lên màn hình

Các thao tác xuất dữ liệu

- putchar() : hàm này để in một kí tự ra màn hình.
  - Cú pháp : putchar(char c) ;
  - Ý nghĩa : Hàm này in kí tự c ra màn hình. Đây là hàm xuất dữ liệu không định dạng.
- puts() : hàm này để in một chuỗi kí tự ra màn hình
  - Cú pháp : puts(const char *s) ;
  - Ý nghĩa : Hàm này in chuỗi kí tự trỏ bởi con trỏ s ra màn hình, sau đó tự động đưa con trỏ xuống đầu dòng tiếp theo.
- printf() : in các loại dữ liệu ra màn hình và có định dạng.

VD: sử dụng hàm putchar() để in các kí tự từ a-z

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i='a';
    while (i<='z')
    {
        putchar(i++);
        putchar(' ');
    }
    getch();
}
```

Các thao tác xuất dữ liệu

- Hàm printf()
  - Xuất dữ liệu ra màn hình phổ biến nhất
  - Cho phép in ra một hay nhiều biểu thức ở tất cả các kiểu dữ liệu cơ bản như kí tự, chuỗi, số
  - Cho phép định dạng lại dữ liệu khi in ra: căn lề (trái hoặc phải), định kích thước vùng dữ liệu sẽ in ra (tính bằng số kí tự sẽ in ra, kích thước này có thể khác với kích thước thật của dữ liệu),…
  - Cú pháp: printf(Hằng chuỗi định dạng[, danh sách các biểu thức]);
  - Hằng chuỗi định dạng có thể gồm 2 phần:
    - Phần nội dung tĩnh: cố định, được xác định bởi một hay một số hằng chuỗi cho trước.
    - Phần nội dung động: không cố định, giá trị của nó được lấy từ giá trị các biểu thức trong danh sách các biểu thức, được quy định bởi các chuỗi kí tự đặc biệt được gọi là các **mã định dạng** (format code), với kí tự bắt đầu luôn là %.

Các thao tác xuất dữ liệu

- VD: sử dụng hàm printf();

```
char  c = 'X' ;
int  i =10 ;
const PI=3.1415 ;
float  r=11.2345678 ;
printf("Kí tự %c có mã ASCII là %d \n", c, c);
printf("%d bình phương", i) ;
printf(" bằng %d ", i*i) ;
printf("Bán kính hình tròn là %f ",r) ;
printf(" có diện tích là %.3f \n",PI*r*r) ;
```

- Kết quả chạy chương trình:

Kí tự X có mã ASCII là 88

10 bình phương bằng 100

Bán kính hình tròn là 11.234568
có diện tích là 378.647

Các thao tác xuất dữ liệu

- Mã định dạng
  - Có nhiệm vụ xác định kiểu dữ liệu và khuôn dạng dữ liệu sẽ được in ra. Mã định dạng là một dãy kí tự đặc biệt luôn bắt đầu bởi kí tự %. Mã định dạng có cấu tạo gồm hai phần như sau:

    % [mã khuôn in] mã chuyển đổi
  - Mã khuôn in: quy định khuôn dạng của dữ liệu sẽ được in ra như kích thước dữ liệu (tính theo số kí tự in ra), căn lề (trái hay phải),... Phần này không bắt buộc phải có. Khi không có phần này, hệ thống sẽ sử dụng khuôn in mặc định cho từng kiểu dữ liệu.
  - Mã chuyển đổi : Đây là phần bắt buộc phải có. Phần này quy định kiểu dữ liệu sẽ được in ra.

Các thao tác xuất dữ liệu

- Mã khuôn in
  - Cấu tạo mã khuôn in: [-][+][0][m | *][.n | .*]
  - Với m, n là các số nguyên không âm. Trong đó:
    - Dấu '-' : xác định căn lề bên trái. Không có dấu này là căn lề bên phải
    - Dấu '+' : áp dụng cho kiểu dữ liệu số (số nguyên hoặc thực), quy định số dương in ra sẽ có dấu +, số âm sẽ có dấu - .

Các thao tác xuất dữ liệu

- Mã khuôn in
  - Giá trị m: là một hằng số nguyên xác định kích thước tối thiểu của dữ liệu sẽ in ra (tính theo số kí tự).
  - Nếu m lớn hơn kích thước thật sự của dữ liệu thì các kí tự trắng (hoặc số 0) sẽ được tự động thêm vào bên trái hoặc bên phải.
  - Nếu m nhỏ hơn hoặc bằng kích thước dữ liệu thật thì kích thước thật của dữ liệu sẽ được in ra.
  - Có thể thay hằng số m bởi dấu *. Khi đó, giá trị kích thước tối thiểu sẽ được tính bởi giá trị của một biểu thức thay thế tương ứng nằm trong danh sách các biểu thức.

Các thao tác xuất dữ liệu

- Mã khuôn in
  - Số 0 : lựa chọn này áp dụng cho kiểu dữ liệu số (số nguyên, số thực), khi căn lề bên phải và kích thước dữ liệu in ra lớn hơn kích thước dữ liệu thật.
  - Khi đó, các số 0 sẽ được bổ sung vào phía trước số sẽ được in ra cho đủ kích thước tối thiểu của dữ liệu in ra.
  - Chế độ mặc định là chèn các kí tự trắng.
  - Việc chèn số 0 chỉ chèn vào phía trước của các kiểu dữ liệu số.

Các thao tác xuất dữ liệu

- Mã khuôn in
  - Giá trị n: luôn đi sau một dấu '.'
  - Lựa chọn này áp dụng cho các kiểu dữ liệu khác nhau với các vai trò khác nhau.
  - Nếu áp dụng cho số thực, giá trị này xác định độ chính xác (số chữ số sau dấu phẩy) của số thực đưa ra. Độ chính xác mặc định cho kiểu số thực là 6 chữ số thập phân.
  - Nếu áp dụng cho số nguyên, nó xác định số chữ số tối thiểu của số nguyên sẽ được in ra. Khi n lớn hơn số chữ số thực của số nguyên thì các số 0 sẽ được tự động chèn vào đầu cho đủ n chữ số.
  - Nếu áp dụng cho kiểu chuỗi, giá trị này xác định số kí tự tối đa của chuỗi sẽ được in ra. Khi n nhỏ hơn hoặc bằng độ dài thực của chuỗi kí tự thì chỉ n kí tự đầu tiên của chuỗi được in ra.
  - Giá trị hằng số n cũng có thể được thay thế bởi kí hiệu '*'.

Hàm printf() – Ví dụ áp dụng

```
main(){
  char ch='A';
  int i=1234, j=-1234;
  float x=11.23456789;
  printf("%-5c,%5c\n", ch, ch);
  printf("%d,%-10d,%+10d,
    %10.6d, %010d\n",i,i,i,i,j);
  printf("%f,%12f,%-
    12.4f,%+012.4f\n",x,x,x,x);
  printf("%*f\n",12,x);
  printf("%*.*f",12,3,x);
  getch();
  return;
}
```

Kết quả chạy chương trình:

```
A^^^^,^^^^A
1234,1234^^^^^^,^^^^^+1234,
  ^^^^001234,-0000001234
11.234568,
  ^^^11.234568,11.2346^^^^^,+000011.2
  346
^^^11.234568
^^^^^^11.235
```

Quy ước các dấu '^' biểu diễn các kí tự trắng để dễ theo dõi.

Bảng mã định dạng

Mã chuyển đổi	Ý nghĩa	Mã chuyển đổi	Ý nghĩa
%c	in ra một kí tự	%ld	in ra một số nguyên dài (long) ở hệ thập phân
%s	in ra một chuỗi kí tự	%lu	in ra một số nguyên dài không dấu ở hệ thập phân
%d,%i,%hd	in ra một số nguyên ở hệ thập phân	%f,%lf	in ra một số thực ở dạng dấu phẩy động
%o	in ra một số nguyên ở hệ cơ số 8	%e,%E	in ra một số thực ở dạng khoa học
%x,%X	in ra một số nguyên ở hệ cơ số 16	%g,%G	in ra một số thực ở dạng dấu phẩy động hoặc dạng khoa học
%u	in ra một số nguyên không dấu ở hệ thập phân		