



Internet Security

Hãy làm việc hết mình, những điều tốt đẹp sẽ đến với bạn !



Bkiss R&D

Programming skills



Contents

- ✓ Làm quen với ngôn ngữ lập trình C
- ✓ Phong cách lập trình
- ✓ Kiểm soát lỗi, tìm lỗi, sửa lỗi

C Programming Language

1. Giới thiệu
2. Các khái niệm cơ bản
3. Các cấu trúc điều khiển
4. Con trỏ
5. File

C Programming Language

1. Giới thiệu

- Do Dennis Ritchie phát triển (1972)
- IDE : Turbo C, Eclipse, NetBeans, Microsoft Visual Studio, ...

C Programming Language

2. Các khái niệm cơ bản

- Cấu trúc 1 chương trình C

[Tiền xử lí]

[Các hàm]

Main()

{

[Khai báo biến;]

[Nhập dữ liệu;]

[Xử lí;]

[Xuất;]

}

```
#include <stdio.h>
int Add(int a, int b)
{
    return a+b;
}
void main()
{
    int a, b, sum;
    a = 1; b = 2;
    sum = Add(a, b);
    printf("Tong la : %d", sum);
}
```

C Programming Language

2. Các khái niệm cơ bản

- Các kí hiệu

STT	KÝ HIỆU	DIỄN GIẢI	VÍ DỤ
1	{ }	Bắt đầu và kết thúc hàm hay khối lệnh.	<pre>void main() { }</pre>
2	;	Kết thúc khai báo biến, một lệnh, một lời gọi hàm, hay khai báo nguyên mẫu hàm.	<pre>int x; void NhapMang(int a[], int &n);</pre>
3	//	Chú thích (ghi chú) cho một dòng. Chỉ có tác dụng đối với người đọc chương trình.	<pre>//Ham nay dung de nhap mang void NhapMang(int a[], int &n);</pre>
4	/* */	Tương tự như ký hiệu //, nhưng cho trường hợp nhiều dòng.	<pre>/* Dau tien nhap vao n. Sau do nhap cho tung phan tu */ void NhapMang(int a[], int &n);</pre>

C Programming Language

2. Các khái niệm cơ bản

- Các kiểu dữ liệu

STT	Kiểu	GHI CHÚ	KÍCH THƯỚC	ĐỊNH DẠNG
KIỂU LIÊN TỤC (SỐ THỰC)				
1	float		4 bytes	%f
2	double		8 bytes	%lf
3	long double		10 bytes	%lf
KIỂU RỜI RẠC (SỐ NGUYÊN)				
1	char	Ký tự	1 byte	%c
		Số nguyên	1 byte	%d
2	unsigned char	Số nguyên dương	1 byte	%d
3	int	Số nguyên	2 bytes	%d
4	unsigned int	Số nguyên dương	2 bytes	%u
5	long	Số nguyên	4 bytes	%ld
6	unsigned long	Số nguyên dương	4 bytes	%lu
7	char *	Chuỗi		%s

C Programming Language

2. Các khái niệm cơ bản

- Các phép toán

STT	PHÉP TOÁN	Ý NGHĨA
PHÉP TOÁN SỐ HỌC		
1	+	Cộng
2	-	Trừ
3	*	Nhân
4	/	Chia lấy phần nguyên
5	%	Chia lấy phần dư
PHÉP TOÁN QUAN HỆ		
1	>	Lớn hơn
2	<	Nhỏ hơn
3	>=	Lớn hơn hoặc bằng
4	<=	Nhỏ hơn hoặc bằng
5	==	Bằng nhau
6	!=	Khác nhau

STT	PHÉP TOÁN	Ý NGHĨA
PHÉP TOÁN LOGIC		
1	!	NOT
2	&&	AND
3		OR
TOÁN TỬ TĂNG GIẢM		
1	++	Tăng 1
2	--	Giảm 1
PHÉP TOÁN THAO TÁC TRÊN BIT		
1	&	AND
2		OR
3	^	XOR
4	<<	Dịch trái
5	>>	Dịch phải
6	~	Lấy phần bù theo bit

C Programming Language

2. Các khái niệm cơ bản

- Các hàm cơ bản

STT	TÊN HÀM	THƯ VIỆN	DIỄN GIẢI	VÍ DỤ
1	printf	#include<stdio.h>	Xuất ra màn hình.	<pre>#include<stdio.h> #include<conio.h> #include<dos.h> void main() { int c = 1, n; clrscr(); printf("Nhap n:"); scanf("%d", &n); do{ textcolor(c); gotoxy(20, 10); cprintf("%d", n); c++; if (c>15) c = 1; delay(200); } while(!kbhit()); }</pre>
2	scanf	#include<stdio.h>	Lấy dữ liệu từ bàn phím.	
3	gotoxy	#include<conio.h>	Di chuyển đầu nháy đến tọa độ (x, y) trên màn hình văn bản.	
4	textcolor	#include<conio.h>	Đặt màu cho chữ (có giá trị từ 0 đến 15).	
5	cprintf	#include<stdio.h>	Xuất ra màn hình với màu chữ đã định liền trước đó.	
6	delay	#include<dos.h>	Dừng thực hiện lệnh tiếp sau một khoảng thời gian.	
7	kbhit	#include<conio.h>	Kiểm tra xem có nhấn phím.	

C Programming Language

1. Giới thiệu
2. Các khái niệm cơ bản
3. Các cấu trúc điều khiển
4. Con trỏ
5. File

C Programming Language

3. Các cấu trúc điều khiển

- Cấu trúc rẽ nhánh

```
if (biểu thức điều kiện)
{
    <khối lệnh> ;
}
```

```
if (biểu thức điều kiện)
{
    <khối lệnh 1>;
}
else
{
    <khối lệnh 2>;
}
```

- Cấu trúc lựa chọn

```
switch (biểu thức)
{
    case n1:
        các câu lệnh ;
        break ;
    case n2:
        các câu lệnh ;
        break ;
    .....
    case nk:
        <các câu lệnh> ;
        break ;
    [default: các câu lệnh]
}
```

C Programming Language

3. Các cấu trúc điều khiển

- Cấu trúc lặp

```
for (<biểu thức khởi gán>; <biểu thức điều kiện>; <biểu thức tăng/giảm>)  
{  
    <khối lệnh>;  
}
```

Ví dụ: In ra màn hình bảng mã ASCII từ ký tự số 33 đến 255.

```
#include<conio.h>  
#include<stdio.h>  
void main()  
{  
    for (int i=33;i<=255;i++)  
        printf("Ma ASCII cua %c: %d\t", i, i) ;  
    getch () ;  
}
```

C Programming Language

3. Các cấu trúc điều khiển

- Cấu trúc lặp

```
< Khởi gán>  
while ( <biểu thức điều kiện>  
{  
    lệnh/ khối lệnh;  
    < tăng/giảm chỉ số lặp>;  
}
```

```
do  
{  
    < khối lệnh> ;  
} while (biểu thức điều kiện) ;
```

C Programming Language

3. Các cấu trúc điều khiển

- Break

```
while (1)
{
    printf("\nNhap n: ");
    scanf("%d", &n);
    if(n<0)
        break;
}

getch () ;
}
```

- Continue

Ví dụ: In ra màn hình giá trị từ 10 đến 20 trừ đi số 13 và số 17.

```
#include<stdio.h>
#include<conio.h>

void main()
{
    for(int i=10 ; i<=20; i++)
    {
        if(i==13||i==17)
            continue;
        printf("%d\t", i);
    }
    getch () ;
}
```

C Programming Language

1. Giới thiệu
2. Các khái niệm cơ bản
3. Các cấu trúc điều khiển
4. Con trỏ
5. File

C Programming Language

4. Con trỏ

- Định nghĩa ?
- Khai báo : `<Kiểu> *<Ten_Con_Tro>`
VD : `int *pA, char* pB`
- Kích thước ?
- Toán tử lấy địa chỉ của biến : `&` (VD : `pA = &C`)
- Truy cập giá trị của con trỏ : `*pA = 10`
- Cấp phát động : `calloc, malloc, free`

C Programming Language

1. Giới thiệu
2. Các khái niệm cơ bản
3. Các cấu trúc điều khiển
4. Con trỏ
5. File

C Programming Language

5. File

- File văn bản, File nhị phân
- Thao tác với tập tin

Quá trình thao tác trên tập tin thông qua 4 bước:

Bước 1: Khai báo con trỏ trỏ đến tập tin.

Bước 2: Mở tập tin.

Bước 3: Các xử lý trên tập tin.

Bước 4: Đóng tập tin.

C Programming Language

5. File

- Khai báo

FILE *< tên biến >;

Ví dụ : `FILE *f;` // Khai bao bien con tro file *f*

fopen (< đường dẫn tên tập tin> , < kiểu truy nhập >);

- Mở file

Ví dụ : `FILE *f;` // Khai bao bien con tro *f*
`f = fopen ("C:\\\\VD1.txt" , "rt");`

Các kiểu truy nhập tập tin thông dụng:

- | | |
|-----------|---|
| t | là kiểu truy nhập tập tin đối với dạng tập tin văn bản (text). |
| b | là kiểu truy nhập tập tin đối với dạng tập tin nhị phân (binary). |
| r | mở ra để đọc (ready only). |
| w | mở ra để ghi (create / write). |
| a | mở ra để thêm vào (append). |
| r+ | mở ra để đọc và ghi (modify). |

C Programming Language

5. File

- Các hàm xử lí
 - ❖ Đọc, ghi file : fscanf, fgets, fprintf, fputs, fread, fwrite, ...
 - ❖ Di chuyển con trỏ file : fseek, ftell
- Đóng file : fclose, fcloseall

Contents

- ✓ Làm quen với ngôn ngữ lập trình C
- ✓ Phong cách lập trình
- ✓ Kiểm soát lỗi, tìm lỗi, sửa lỗi

Programming style

1. Tổ chức chương trình

- Module hóa
 - Mỗi module thực hiện 1 công việc
 - Độc lập
 - Tham số thay biến toàn cục
- File
 - Header : *.h
 - Code : *.cpp

Programming style

2. Chuẩn tài liệu

- // : chú thích
- /* */ : tạm vô hiệu hóa 1 đoạn code (test, debug)
- Đầu mỗi chương trình, module, lớp, ... : có khối chú thích

```
////////////////////////////////////  
// CuongNC : 10/10/2009  
// Info : Code lai ham FindResource cua windows  
// Param :  
// - szResType (in) : Ten hoac id cua Resource Type (dang Unicode)  
// - szResName (in) : Ten hoac id cua Resource Name (dang Unicode)  
// - pDwResOff (out) : Dia chi cua resource neu tim thay  
// - pDwOffInGpcode (out) : Offset cua resource trong mang g_Pcode neu tim thay  
// - Return : 0 neu co loi hoac ko tim thay, resource size neu tim thay  
// Note : neu tham so truyen vao la id thi dung Macro : CNC_MAKEINTRESOURCE  
// Mac dinh lay theo language dau tien  
////////////////////////////////////  
DWORD CNC_GetResource(LPCWSTR szResType, LPCWSTR szResName, PDWORD pDwResOff, LOCALVAR* V)  
{
```

Programming style

3. Tên

- Giàu ý nghĩa
- Tên biến
 - ✓ i, j : chỉ số, p : con trỏ (`int *piBufferSize`)
 - ✓ Hằng : tất cả là chữ cái hoa (`NUMBER_NAME`)
 - ✓ Tiền tố : chữ cái thường (kiểu dữ liệu) :
`char chKitu; int iCount; string sThongBao;`
 - ✓ Nhất quán

```
class
    UserQueue
{
    int noOfItemsInQ, frontOfTheQueue, queueCapacity;
    public int noOfUsersInQueue() {...}
};
```

Programming style

3. Tên

- Tên hàm
 - ✓ Thể hiện chức năng
 - ✓ Động từ + danh từ : getTime(), isEmpty(), createStudentList (), ...

4. Định dạng

- Lùi đầu dòng các đoạn code
- Căn chỉnh

```
int songuyen    = 100;

double sothuc   = 3.14 ;
char kytu       = 'a' ;
char ten[]      = { "tam", "lan", "hiep", "bao", "yen", "tuan", "hoa" };
bool gt[]       = { 0 , 0 , 0 , 1 , 0 , 1 , 0 };
```

Programming style

4. Định dạng

- `{ }`

```
if (!done)
{
    doSomething();
    moreToDo();
}
else
{
    //...
}
```

```
if (!done) {
    doSomething();
    moreToDo();
} else {
    //...
}
```

- `()`

```
// no space between 'strcmp' and '(',
// but space between 'if' and '('
if ( strcmp( input_value, "done" ) == 0 )
return 0;
```

- Giữa các đoạn : chèn dòng trắng

Programming style

5. 0

```
File    *f;  
Float   fHight;  
Char    szName[10];
```

```
f           = 0;  
fHight      = 0;  
szName[9]   = 0;
```

```
f           = NULL;  
fHight      = 0.0;  
szName[9]   = '\0';
```

Programming style

6. Microsoft (MSDN)

```
// crt_toupper.c
/* This program uses toupper and tolower to
 * analyze all characters between 0x0 and 0x7F. It also
 * applies _toupper and _tolower to any code in this
 * range for which these functions make sense.
 */

#include <ctype.h>
#include <string.h>

char msg[] = "Some of THESE letters are Capitals.";
char *p;

int main( void )
{
    printf( "%s\n", msg );

    /* Reverse case of message. */
    for( p = msg; p < msg + strlen( msg ); p++ )
    {
        if( islower( *p ) )
            putchar( _toupper( *p ) );
        else if( isupper( *p ) )
            putchar( _tolower( *p ) );
        else
            putchar( *p );
    }
}
```

Programming style

6. Microsoft (MSDN)

```
BOOL ListProcessThreads( DWORD dwOwnerPID )
{
    HANDLE hThreadSnap = INVALID_HANDLE_VALUE;
    THREADENTRY32 te32;

    // Take a snapshot of all running threads
    hThreadSnap = CreateToolhelp32Snapshot( TH32CS_SNAPTHREAD, 0 );
    if( hThreadSnap == INVALID_HANDLE_VALUE )
        return( FALSE );

    // Fill in the size of the structure before using it.
    te32.dwSize = sizeof(THREADENTRY32);

    // Retrieve information about the first thread,
    // and exit if unsuccessful
    if( !Thread32First( hThreadSnap, &te32 ) )
    {
        printError( TEXT("Thread32First") ); // show cause of failure
        CloseHandle( hThreadSnap );          // clean the snapshot object
        return( FALSE );
    }
}
```


Programming style

6. Microsoft (MSDN)

```
#include <windows.h>
#include <stdio.h>

void main()
{
    HANDLE hFile;
    HANDLE hAppend;
    DWORD dwBytesRead, dwBytesWritten, dwPos;
    BYTE buff[4096];

    // Open the existing file.

    hFile = CreateFile(TEXT("one.txt"), // open One.txt
                      GENERIC_READ,    // open for reading
                      0,                // do not share
                      NULL,            // no security
                      OPEN_EXISTING,   // existing file only
                      FILE_ATTRIBUTE_NORMAL, // normal file
                      NULL);           // no attr. template

    if (hFile == INVALID_HANDLE_VALUE)
    {
        printf("Could not open One.txt.");
        return;
    }
}
```

Programming style

7. DucLM

```
////////////////////////////////////  
// Name:      IsDirectory  
// Description: check whether file is directory or not  
// Params:    szFileName: fullpath of the file being checked  
// Return:    TRUE if file is directory  
//           FALSE otherwise  
////////////////////////////////////  
BOOL IsDirectory(const TCHAR* szFileName)  
{  
    DWORD dwAttrib = GetFileAttributes(szFileName);  
    if (dwAttrib == INVALID_FILE_ATTRIBUTES) return FALSE;  
  
    return ((dwAttrib & FILE_ATTRIBUTE_DIRECTORY) == FILE_ATTRIBUTE_DIRECTORY);  
}
```

Programming style

8. Các bạn chế K53

```
int ng,th,n;  
int FALSE=0;  
clrscr();
```

```
int primenumber(int n)  
{  
    int p=1,i;  
    for (i=2;i<(n+1)/2;i++)  
        if (n%i==0) p=0;  
    return p;  
}
```

```
#include<stdio.h>  
#include<conio.h>  
//Ham tim UCLN cua 2 so  
int UCLN(int a,int b)  
{if(a>b)  
if(a%b==0) return b;  
else return UCLN(b,a%b);  
else if(b%a==0) return a;  
else return UCLN(a,b%a);}
```

Programming style

8. Các bạn chế K53

```
if(fpt1==NULL)
    printf("\n LOI-ko mo duoc tep");
else
{
    while(feof(fpt1)!=0)
    {
        i++;
        fread(p[i],sizeof(int),1,fpt1);
    }
    n=i;max=*p[0];
    for(i=0;i<=n;i++)
        if(*p[i]>max) max=*p[i];
    fpt2=fopen("C:\\K53\\TC\\BIN\\Max.out","wb");
    if(fpt2==NULL)
        printf("\n LOI-ko mo duoc tep");
    else
        fwrite(&max,sizeof(int),1,fpt2);
}
```

Programming style

8. Các bạn chế K53

```
void main()
{
    int n,a,b,c,a1,b1,c1;
    clrscr();
    printf("\t Nhap 1 so nguyen co 3 chu so.");
    printf("\n N = ");
    scanf("%d",&n);
    a=n/100;
    b=(n%100)/10;
    c=((n%100)%10);
    a1= (a>b) ? (c>a)?c:a : (c>b)?c:b;
    b1= (a>b) ? (c>a)?a:((c>b)?c:b) : (a>c)?a:((c>b)?b:c);
    c1= (a>b) ? (c>b)?b:c : (c>a)?a:c;
    printf("\n\t Ket qua : %d",a1*100+b1*10+c1);
    getch();
}
```

■ Programming skills

THE END !