Visual Studio Debug

F5 正常调试，只有在断点时才能监视变量

F10 逐过程，一条条走，遇见过程全跳过

F11 逐语句，可以深入过程中

1字节=8个2进制

**整数类型**

|  |  |  |  |
| --- | --- | --- | --- |
| 16位机 | | | |
| 类型 | 最小值 | 最大值 | 长度（字节） |
| Short int | -32768 | 32767 | 2 |
| Unsigned short int | 0 | 65535 | 2 |
| int | -32768 | 32767 | 2 |
| Unsigned int | 0 | 65535 | 2 |
| long int | -2147483648 | 2147483647 | 4 |
| Unsigned long int | 0 | 4294967295 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 32位机 | | | |
| 类型 | 最小值 | 最大值 | 长度（字节） |
| Short int | -32768 | 32767 | 2 |
| Unsigned short int | 0 | 65535 | 2 |
| int | -2147483648 | 2147483647 | 4 |
| Unsigned int | 0 | 4294967295 | 4 |
| long int | -2147483648 | 2147483647 | 4 |
| Unsigned long int | 0 | 4294967295 | 4 |
| Long long int | -2e63 | 2e63-1 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
| 64位机 | | | |
| 类型 | 最小值 | 最大值 | 长度（字节） |
| Short int | -32768 | 32767 | 2 |
| Unsigned short int | 0 | 65535 | 2 |
| int | -2147483648 | 2147483647 | 4 |
| Unsigned int | 0 | 4294967295 | 4 |
| long int | -9223372036854775808 | 9223372036854775807 |  |
| Unsigned long int | 0 | 18446774073709551615 |  |
| 其他 | | | |
| 类型 | 最小值 | 最大值 | 长度（字节） |
| Float | 1.17549e-38 | 3.40282e-38 | 4 |
| Double |  |  | 15 |
| Long double |  |  | 31 |

NOTE 1

#line 2

//this line, next to #line will be line 2

NOTE 2

#define 运算符#，##的缺陷

#define A(x) #x，调用A(\_\_FILE\_\_)的结果是”\_\_FILE\_\_”

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define A(x,y) #x+y

宏替换运算次序为

NO1 查找标识符

NO2 替换为#x+y

NO3 先检查未参与#和##的运算符，是否有参数列表（x,y），然后替换呢

NO4 然后，最后整体再检查是否有宏需要替换

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

NOTE 3

(1) 为了产生行号（11）的字符串字面量

#define Show(x) #x

调用Show(\_\_FILE\_\_)结果为”\_\_FILE\_\_”

解决方法：

#define Show2(x) Show(x)

调用Show2(x)结果为Show(11)，再扩展为“11”

如果像goto break continue case 跳过了一段代码，如果同时跳过了局部变量的声明，再次在相同局部引用变量，效果？

{  
 的作用域没改变，但是禁止case跳过变量初始化，而且不管是否引用变量。

}

NOTE 4

Escape Sequence 字符转义

\a警报符 \b回退符 \f换页符 \n换行符 \r回车符 \t水平制表

\v垂直制表 \\反斜杠 \?问号 \’单引号 \”双引号

Scanf/Printf独有百分号表示转义 %%表示匹配一个百分号，但一般字符串一个%即可

NOTE 5

隐式转换

//C89

（1）任意操作数为浮点类型，对较狭小的类型提升（适用于? : 表达式的结果）

Float -> double -> long double

（2）两个数都不是浮点类型，两个数进行整值提升，再对狭小的提升

Int -> unsigned int -> long -> unsigned long

(例外 long + unsigned int -> unsigned long两个数都这样提升 x32)

//1整值提升，char/short转换为int (INFO: unsigned short 转化 unsigned int x16)

//C99

//整数等级排位

//1 long long, unsigned long long

//2 long, unsigned long

//3 int, unsigned int

//4 short, unsigned short

//5 char, signed char, unsigned char

//6 \_Bool

（1）任意操作数为浮点类型，对较狭小的类型提升（适用于? : 表达式的结果）

Float -> double -> long double

（2）两个数都不是浮点类型，

1. 如果两个都是 (有符号/无符号) ，将整数等级低的转化为等级高的。

2. 如果无符号数等级高于或等于有符号数等级，将有符号数转化为无符号类 型.

3. 如果有符号数可以表示无符号数类型的所有值，将无符号数转化为有符号 数类型。

4. 否则，将两个数转化为有符号类型对应的无符号类型

//2 整数提升，任何等级低于int/unsigned int类型转化为int或unsigned int

NOTE 6

常量储存方式

//整数

//C89

//十进制常量 int -> long -> unsigned long

//八，十六常量 int -> unsigned int -> long -> unsigned long

//C99

//十进制常量 int -> long -> long long

//八，十六常量 int -> unsigned int -> long -> unsigned long -> long long -> unsigned long long

// u l(L) ll(LL) 标记改变储存方式

//浮点数

//默认为double, 除非 f(F) l(L) 标记

NOTE 7

算术溢出

//整数溢出

//有符号数 未定义

//无符号数 对n^2取模，（n^2等于相应无符号最大整数）丢掉超出

//浮点溢出

//未定义

NOTE 8

Signed/unsigned类型转换

//signed -> unsigned

//加上n^2，（n^2等于相应无符号最大整数）

//unsigned -> signed

//如果越界，参考算术溢出

NOTE 9

实际参数提升

//编译器调用前遇见原型，隐式转换换成相应形式参数。

//可变参数数目的函数，未指明的参数类型（ … 里面的）执行下面操作。

//没有遇见原型（仅C89）

Float -> double

Char/short -> int （C89 整值提升， C99整数提升）

NOTE 10

不完整类型

//definition: 描述了对象但缺少定义对象大小所需的信息

//struct t; typedef struct t \*T;

//不完整结构声明

//struct hack

{

Int length;

Char h[]

};

//malloc( sizeof(struct hack)+N );

//extern int a[];

//int a[]={1,2,3};

//void

NOTE 11

Standard input and output header file

//错误指示器清空函数

clearerr (feof ferror)

fopen (feof ferror)

freopen (feof ferror)

ungetc (eof)

rewind (eof ferror)

fseek (eof)

fsetpos (eof)

NOTE 12

String类型函数，参数n的限制 (以留下空字符位为标准)

strncat() n=sizeof(s1)-sizeof(s2)-1

//函数不加入空字符

strncpy() n=sizeof(s)-1

//函数不加入空字符

fgets() n=sizeof(s)

//函数一定加入空字符，可加入\n进字符串

scanf(“%ns”) n=sizeof(s)-1

//scanf(“%ns”) 匹配n个字符，一定加入空字符

//scanf(“%nc”) 匹配n个字符，但不加入空字符

snprintf() n=sizeof(s)

其他类型的n限制

setvbuf(size\_t size) n=sizeof(s)

NOTE 13

文件缓冲清洗条件

fclose

fflush

buffer is full/line/none

NOTE 14

文件函数

//open

FILE \***fopen**(const char \*restrict filename, const char \*restrict mode);

FILE \***freopen**(const char \*restrict filename, const char \*restrict mode, FILE \*restrict stream);

//close

int **fclose**(FILE \*stream);

//temp

FILE \***tmpfile**(void);

char \***tmpnam**(char \*s);

//buffer

Int **fflush**(FILE \*stream);

void **setbuf**(FILE \*restrict stream, char \*restrict buf);

int **setvbuf**(FILE \*restrict stream, char \*restrict buf, int mode, size\_t size);

//change

int **remove**(const char \*filename);

int **rename**(const char \*old, const char \*new);

int **ungetc**(int c, FILE \*stream);

//read

int **fscanf**(FILE \*restrict stream, const char \* restrict format, …);

int **scanf**(const char \* restrict format, …);

int **fgetc**(FILE \*stream);

int **getc**(FILE \*stream);

int **getchar**(void);

char \***fgets**(char \*restrict s, int n, FILE \*restrict stream);

char \***gets**(char \*s);

size\_t **fread**(void \* restrict ptr, size\_t size, size\_t nmemb, FILE \*restrict stream);

//write

int **fprintf**(FILE \*restrict stream, const char \* restrict format, …);

int **printf**(const char \* restrict format, …);

int **fputc**(int c, FILE \*stream);

int **putc**(int c, FILE \*stream);

int **putchar**(int c);

int **fputs**(const char \*restrict s, FILE \*restrict stream);

int **puts**(const char \*restrict s);

size\_t **fwrite**(void \* restrict ptr, size\_t size, size\_t nmemb, FILE \*restrict stream);

//error

void **clearerr**(FILE \*stream);

int **feof**(FILE \*stream);

int **ferror**(FILE \*stream);

//position

int **fgetpos**(FILE \*restrict stream, fpos\_t \*restrict pos);

int **fseek**(FILE \*stream, long int offset, int whence);

int **fsetpos**(FILE \*restrict stream, const fpos\_t \*pos);

long int **ftell**(FILE \*stream);

void **rewind**(FILE \*stream);

//string

int **sprintf**(char \*restrict s, const char \*restrict format, …);

int **snprintf**(char \*restrict s, size\_t n, const char \*restrict format, …);

int **sscanf**(cosnt char \*restrict s, const char \*restrict format, …);

NOTE 15



NOTE 16

字符串字面量，储存期限，与变量相同

NOTE 17

神奇的界限

//标识符识别长度

//内部链接

C89 31

C99 63

//外部链接

C89 6 （没错只有6个，而且不区分大小写）

C99 31 （区分大小写了）

// 转换说明 %g 的科学计数法表示界限

表示为科学记数法后的10^n中的n

if ( n<-4 || n >p ) 表示为科学计数法

NOTE 18

弱智VS2012，不支持C99的%hhu 读写unsigned char