

史丹利部落格

倘若您認為「本網站」內容、圖案、超連結等侵害他人之著作權或不願意被連結，請留言通知站長，站長看到留言後會立即移除。

搜尋

2007年4月4日 星期三

C語言巨集定義技巧

轉載自：
http://miaozi.spaces.live.com/?_c11_blogpart_blogpart=blogview&_c=blogpart&_c02_owner=1&partqs=amonth%3d2%26ayear%3d2007
C語言巨集定義技巧
1．防止一個頭檔被重複包含

```
#ifndef COMDEF_H

#define COMDEF_H

//頭檔內容

#endif

2．重新定義一些類型，防止由於各種平臺和編譯器的不同，而產生的類型位元組數差異，方便移植。
```

```
typedef unsigned char boolean; /* Boolean value type. */

typedef unsigned long int uint32; /* Unsigned 32 bit value */

typedef unsigned short uint16; /* Unsigned 16 bit value */

typedef unsigned char uint8; /* Unsigned 8 bit value */

typedef signed long int int32; /* Signed 32 bit value */

typedef signed short int16; /* Signed 16 bit value */

typedef signed char int8; /* Signed 8 bit value */
//下面的不建議使用

typedef unsigned char byte; /* Unsigned 8 bit value type. */

typedef unsigned short word; /* Unsinged 16 bit value type. */

typedef unsigned long dword; /* Unsigned 32 bit value type. */

typedef unsigned char uint1; /* Unsigned 8 bit value type. */

typedef unsigned short uint2; /* Unsigned 16 bit value type. */

typedef unsigned long uint4; /* Unsigned 32 bit value type. */

typedef signed char int1; /* Signed 8 bit value type. */

typedef signed short int2; /* Signed 16 bit value type. */

typedef long int int4; /* Signed 32 bit value type. */

typedef signed long sint31; /* Signed 32 bit value */

typedef signed short sint15; /* Signed 16 bit value */

typedef signed char sint7; /* Signed 8 bit value */
```

3．得到指定位址上的一個位元組或字

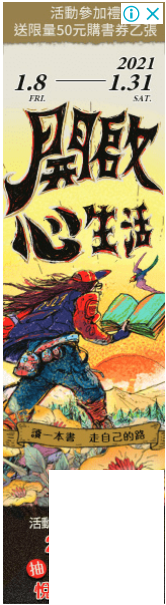
```
#define MEM_B( x ) ( *( (byte *) (x) ) )

#define MEM_W( x ) ( *( (word *) (x) ) )

4．求最大值和最小值
```

```
#define MAX( x, y ) ( ((x) > (y)) ? (x) : (y) )

#define MIN( x, y ) ( ((x) < (y)) ? (x) : (y) )
```



請選取語言
由「Google 翻譯」技術提

中原98碩專班

- 中原資工系
- 中原iLearning
- 中原e點靈(touch)
- 中原圖書館
- 張耀仁-寬頻無線通訊

文章分類

- 平行處理 (8)
- 生活 (7)
- 投資 (9)
- 定義 (6)
- 英文 (11)
- 計算機組織 (7)
- 專利 (1)
- 專案管理 (5)
- 理財 (8)
- 感測器 (2)

5 · 得到一個field在結構體(struct)中的偏移量

```
#define FPOS( type, field ) \

/*lint -e545 */ ( (dword) &(( type *) 0)-> field ) /*lint +e545 */
```

6,得到一個結構體中field所佔用的位元組數

```
#define FSIZ( type, field ) sizeof( ((type *) 0)->field )
```

7 · 按照LSB格式把兩個位元組轉化為一個Word

```
#define FLIPW( ray ) ( (((word) (ray)[0]) * 256) + (ray)[1] )
```

8 · 按照LSB格式把一個Word轉化為兩個位元組

```
#define FLOPW( ray, val ) \

(ray)[0] = ((val) / 256); \

(ray)[1] = ((val) & 0xFF)
```

9 · 得到一個變數的位址 (word寬度)

```
#define B_PTR( var ) ( (byte *) (void *) &(var) )

#define W_PTR( var ) ( (word *) (void *) &(var) )
```

10 · 得到一個字的高位和低位元組

```
#define WORD_LO(xxx) ((byte) ((word)(xxx) & 255))

#define WORD_HI(xxx) ((byte) ((word)(xxx) >> 8))
```

11 · 返回一個比X大的最接近的8的倍數

```
#define RND8( x ) (((x) + 7) / 8 ) * 8 )
```

12 · 將一個字母轉換為大寫

```
#define UPCASE( c ) ( ((c) >= 'a' && (c) <= 'z') ? ((c) - 0x20) : (c) )
```

13 · 判斷字元是不是10進值的數字

```
#define DECCHK( c ) ((c) >= '0' && (c) <= '9')
```

14 · 判斷字元是不是16進值的數字

```
#define HEXCHK( c ) ( ((c) >= '0' && (c) <= '9') || \

((c) >= 'A' && (c) <= 'F') || \

((c) >= 'a' && (c) <= 'f') )
```

15 · 防止溢出的方法

```
#define INC_SAT( val ) (val = ((val)+1 > (val)) ? (val)+1 : (val))
```

16 · 返回陣列元素的個數

```
#define ARR_SIZE( a ) ( sizeof( a ) / sizeof( a[0] ) )
```

17 · 返回一個無符號數n尾的值MOD_BY_POWER_OF_TWO(X,n)=X%(2^n)

```
#define MOD_BY_POWER_OF_TWO( val, mod_by ) \

( (dword)(val) & (dword)(mod_by)-1 )
```

18 · 對於IO空間映射在存儲空間的結構 · 輸入輸出處理

```
#define inp(port) (*((volatile byte *) (port)))

#define inpw(port) (*((volatile word *) (port)))

#define inpdw(port) (*((volatile dword *) (port)))

#define outp(port, val) (*((volatile byte *) (port)) = ((byte) (val)))

#define outpw(port, val) (*((volatile word *) (port)) = ((word) (val)))

#define outpdw(port, val) (*((volatile dword *) (port)) = ((dword) (val)))
```

19,使用一些宏跟蹤調試

A N S I標準說明了五個預定義的宏名。它們是：

_ L I N E _

經典c程式100例 (10)

影音 (25)

數學 (1)

論文 (1)

雜項 (10)

霹靂布袋戲 (9)

Android (17)

ARM (12)

Audio (2)

benchmark (1)

BigData (1)

BIOS (10)

bluespec (1)

Bootloader (5)

Brillo (2)

C 語言 (18)

C# (1)

CSS (2)

DoCoMo (5)

ESL (3)

Google (1)

GPL (1)

Graphic (2)

jpeg (1)

kernel (16)

Linux (8)

LLVM (2)

mangos (8)

MID (1)

MPEG (2)

mysql (1)

OpenSourceTool (1)

Protocol (3)

socket (9)

SystemC (13)

tensorflow (1)

ubuntu (6)

USB (3)

verilog (11)

VirttualPlatform (3)

WindowsTool (2)

x86 (1)

我的最愛

AustraliaNetwork

ABC Radio Australia

ESLBits

EnglishCentral

可汗學院

Highbrow

_ F I L E _

_ D A T E _

_ T I M E _

_ S T D C _

如果編譯不是標準的，則可能僅支援以上宏名中的幾個，或根本不支援。記住編譯程序

也許還提供其他預定義的宏名。

_ L I N E _及 _ F I L E _巨集指令在有關# l i n e的部分中已討論，這裏討論其餘的宏名。

_ D A T E _巨集指令含有形式為月/日/年的串，表示原始檔案被翻譯到代碼時的日期。

源代碼翻譯到目標代碼的時間作為串包含在 _ T I M E _中。串形式為時：分：秒。

如果實現是標準的，則宏 _ S T D C _含有十進位常量1。如果它含有任何其他數，則實現是非標準的。

可以定義宏，例如：

當定義了 _ D E B U G _，輸出資料資訊和所在檔所在行

```
#ifdef _DEBUG

#define DEBUGMSG(msg,date) printf(msg);printf(“%d%d”,date,_LINE_,_FILE_)

#else

#define DEBUGMSG(msg,date)

#endif
```

20．巨集定義防止使用是錯誤

用小括弧包含。

例如：`#define ADD(a,b) (a+b)`

用`do{}while(0)`語句包含多語句防止錯誤

例如：`#difne DO(a,b) a+b;\`

`a++;`

應用時：`if(...)`

`DO(a,b);` //產生錯誤

`else`

解決方法：`#difne DO(a,b) do{a+b;\`

`a++;}while(0)`

宏中“#”和“##”的用法

一、一般用法

我們使用#把巨集引數變為一個字串,用##把兩個巨集引數貼合在一起.

用法:

`# i nclude`

`# i nclude`

`using namespace std;`

`#define STR(s) #s`

`#define CONS(a,b) int(a##e##b)`

```
int main()
{
printf(STR(vck)); // 輸出字串"vck"
printf("%d\n", CONS(2,3)); // 2e3 輸出:2000
return 0;
}
```

二、當巨集引數是另一個宏的時候

需要注意的是凡巨集定義裏有用#或##的地方巨集引數是不會再展開。

1, 非#和##的情況

`#define TOW (2)`

多鄰國

Udemy

何圓圓的媽咪

Jack Homework Blog

討論KGDB網站

查爾斯的天空

內核之旅

菠蘿麵包的家

空中英語教室

本益比排行榜

ESL-LAB

ELLLO免費英文聽力網

MaoYang-有話要說

南華大學-課程影音系

民視英文新聞

圖解英文字典[only IE]

JServ 黃敬群

Jollen網路學院

小吳實實家

CSDN中國IT社區

The Code Project

Codeguru

The News for Linux

LinuxJournal

Acrobat

Format My Source Cc

網誌封存

► 2020 (2)

► 2018 (2)

► 2017 (4)

► 2016 (14)

► 2015 (23)

► 2013 (13)

► 2012 (11)

► 2011 (19)

► 2010 (14)

► 2009 (30)

► 2008 (124)

▼ 2007 (45)

► 六月 (1)

► 五月 (5)

▼ 四月 (39)

C#面試題

C/C++語言struc

NTT DoCoMo V
School Frienc

楊宗緯 人質@起
2007.4.6

mangos ZG技能

女兒學會了"嘆"

mangos 奧妮克

mangos 2.0.10 |

時間=金錢,智識:

```
#define MUL(a,b) (a*b)

printf("%d*%d=%d\n", TOW, TOW, MUL(TOW,TOW));
這行的宏會被展開為：
printf("%d*%d=%d\n", (2), (2), ((2)*(2)));
MUL裏的參數TOW會被展開為(2).

2, 當有'#或'##'的時候
#define A (2)
#define STR(s) #s
#define CONS(a,b) int(a##e##b)

printf("int max: %s\n", STR(INT_MAX)); // INT_MAX #include
這行會被展開為：
printf("int max: %s\n", "INT_MAX");

printf("%s\n", CONS(A, A)); // compile error
這一行則是：
printf("%s\n", int(AeA));

INT_MAX和A都不會被再展開, 然而解決這個問題的方法很簡單. 加多一層中間轉換宏.
加這層巨集的用意是把所有宏的參數在這層裏全部展開, 那麼在轉換巨集裏的那一個巨集(_STR)就能得到正確的巨集引數.

#define A (2)
#define _STR(s) #s
#define STR(s) _STR(s) // 轉換宏
#define _CONS(a,b) int(a##e##b)
#define CONS(a,b) _CONS(a,b) // 轉換宏

printf("int max: %s\n", STR(INT_MAX)); // INT_MAX,int型的最大值, 為一個變數 #include
輸出為: int max: 0x7fffffff
STR(INT_MAX) --> _STR(0x7fffffff) 然後再轉換成字串；

printf("%d\n", CONS(A, A));
輸出為：200
CONS(A, A) --> _CONS((2), (2)) --> int((2)e(2))

三、'#和'##'的一些應用特例
1、合併匿名變數名
#define __ANONYMOUS1(type, var, line) type var##line
#define __ANONYMOUS0(type, line) __ANONYMOUS1(type, _anonymous, line)
#define ANONYMOUS(type) __ANONYMOUS0(type, __LINE__)
例：ANONYMOUS(static int); 即: static int _anonymous70; 70表示該行行號；
第一層：ANONYMOUS(static int); --> __ANONYMOUS0(static int, __LINE__);
第二層：--> __ANONYMOUS1(static int, _anonymous, 70);
第三層：--> static int _anonymous70;
即每次只能解開當前層的宏，所以__LINE__在第二層才能被解開；

2、填充結構
#define FILL(a) {a, #a}

enum IDD{OPEN, CLOSE};
typedef struct MSG{
    IDD id;
    const char * msg;
}MSG;

MSG _msg[] = {FILL(OPEN), FILL(CLOSE)};
相當於：
MSG _msg[] = {{OPEN, "OPEN"},
{CLOSE, "CLOSE"}};

3、記錄檔案名
#define _GET_FILE_NAME(f) #f
#define GET_FILE_NAME(f) _GET_FILE_NAME(f)
static char FILE_NAME[] = GET_FILE_NAME(__FILE__);

4、得到一個數值類型所對應的字串緩衝大小
#define _TYPE_BUF_SIZE(type) sizeof #type
#define TYPE_BUF_SIZE(type) _TYPE_BUF_SIZE(type)
char buf[TYPE_BUF_SIZE(INT_MAX)];
--> char buf[_TYPE_BUF_SIZE(0x7fffffff)];
--> char buf[sizeof "0x7fffffff"];
這裏相當於：
char buf[11];

張貼者： 史丹利 於 下午4:45
標籤： C 語言
```

2 則留言:

 **jjwang** 提到...

您的文章寫到:

7 · 按照LSB格式把兩個位元組轉化為一個Word

有趣的計算

SUN公司的考試

有趣的C程式

電腦開機流程

變數的資料型態

深入淺出 Hello World

快速修復資料庫

C語言巨集定義技巧

Linux下的即時通訊

RTSP RTP RTC

mangos DBC File

編程秘笈篇：C語言

絕招

A*尋徑演算法

C語言嵌入式系統

操作

C語言嵌入式系統

優化

經典c程式100例

經典c程式100例

經典c程式100例

經典c程式100例

經典c程式100例

經典c程式100例

經典c程式100例

經典c程式100例

經典c程式100例

Embedded C Language

mangos 恐懼的代價

mangos 解決怪異問題

mangos 傳送資料

Mangos 簡介

關於我自己

史丹利

檢視我的完整簡介

Vis. today	105
Visits	1 214 498
Page today	115

```
#define FLIPW( ray ) ( (((word) (ray)[0]) * 256) + (ray)[1] )
8 · 按照LSB格式把一個Word轉化為兩個位元組
#define FLOPW( ray, val ) \
(ray)[0] = ((val) / 256); \
(ray)[1] = ((val) & 0xFF)
```

不過這兩個macro似乎是MSB的格式？
(ray)[0]=>most significant byte first ？

2008年4月6日 下午7:08



史丹利 提到...

Hi jjwan:
不好意思@@我也不知道你有問問題@@所以....這麼久才回文

其實它文章應該必需要說明這是給什麼平台用的,不同平台的Big Endian或Little Endian會影響LSB/MSB的定義.

它原本文章寫給Big Endian的話,它是對的
如果寫給Little Endian,你是對的.

不過我當初看到這段時,是直接把它想成它的LSB指的是低位址的ray[0]是存高位元資料,高位址的ray[1]是低位元資料@@

2008年8月6日 下午11:16

張貼留言

較新的文章

首頁

較舊的文章

訂閱： 張貼留言 (Atom)