**[深入C语言可变参数(va\_arg,va\_list,va\_start,va\_end,\_INTSIZEOF)](http://www.cnblogs.com/haoyuanyuan/p/3221463.html)**

一、什么是可变参数

         在C语言编程中有时会遇到一些参数个数可变的函数，例如printf(),scanf()函数，其函数原型为：   
int printf(const char\* format,…)，int scanf(const char \*format,…);它除了有一个参数format固定以外，后面跟着的参数的个数和类型是可变的(用三个点“…”做参数占位符)，实际调用时可以有以下的形式:   
           printf(“%d”,i);  printf(“%d,%c”,i,j);

二、C语言---简单的可变参数例子(实现思想)

**1、整型数据的输出**

[复制代码](javascript:void(0);)

#include <stdio.h>

#include <stdarg.h>

void ar\_cnt(int cnt,...);

void ar\_cst(char const \*s,...);

int main(int argc, char\* argv[])

{

int in\_size =\_INTSIZEOF(int);

printf("int\_size=%d\n",in\_size);

ar\_cnt(5,1,2,3,4);

return 0;

}

void ar\_cnt(int cnt,...)

{

int value1=0;

int i=0;

int arg\_cnt = cnt;

va\_list arg\_ptr;

va\_start(arg\_ptr,cnt);

for(i=0;i<cnt;i++)

{

value1=va\_arg(arg\_ptr,int);

printf("posation %d=%d\n",value1,i+1);

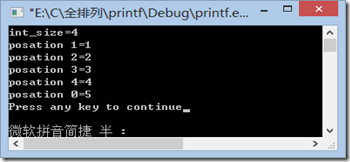
}

va\_end(arg\_ptr);

}

[复制代码](javascript:void(0);)

运行结果:

[](http://images.cnitblog.com/blog/490879/201307/28193443-9fc534881d184229b61586e29e8e3085.png)

**2、字符串的输出**

[复制代码](javascript:void(0);)

#include <stdio.h>

#include <stdarg.h>

void PrintLines(char \*first,...)

{

char \*str;

va\_list v1;

str = first;

va\_start(v1,first);

do

{

printf("%s\n",str);

str=va\_arg(v1,char\*);

} while (str != NULL );

va\_end(v1);

}

int main(int argc, char\* argv[])

{

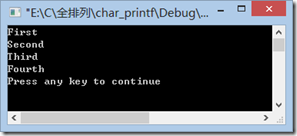
PrintLines("First","Second","Third","Fourth",NULL);

return 0;

}

[复制代码](javascript:void(0);)

运行结果：

[](http://images.cnitblog.com/blog/490879/201307/28193446-bac2ddcba215488db42815e2b87b2f1d.png)

**3、找出最大数**

[复制代码](javascript:void(0);)

#include <stdio.h>

#include <stdarg.h>

int FindMax(int amount,...)

{

int i,val,great;

va\_list v1;

va\_start(v1,amount);

great=va\_arg(v1,int);

for(i=1;i<amount;i++)

{

val=va\_arg(v1,int);

great=(great>val)?great:val;

}

va\_end(v1);

return great;

}

int main(int argc, char\* argv[])

{

int max=FindMax(5,100,20,456,102,4,300);

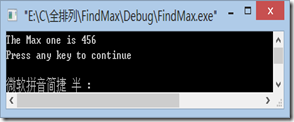
printf("The Max one is %d\n",max);

return 0;

}

[复制代码](javascript:void(0);)

运行结果：

[](http://images.cnitblog.com/blog/490879/201307/28193448-acb9e4c289a742fa9e4a8536bb0e25f6.png)

三、对va\_arg,va\_list,va\_start,va\_end,\_INTSIZEOF剖析

* 内存对齐  #define \_INTSIZEOF(n) ((sizeof(n)+sizeof(int)-1)&~(sizeof(int) - 1) )   
               具体请参考：[http://blog.csdn.net/swell624/article/details/3210779](http://blog.csdn.net/swell624/article/details/3210779" \o "http://blog.csdn.net/swell624/article/details/3210779)
* 可变参数用到以下宏函数    
      <1>原型： void va\_start(va\_list arg\_ptr,prev\_param);

              功能：以固定参数的地址为起点确定变参的内存起始地址，获取第一个参数的首地址

              返回值：无

        <2>原型：va\_list 类型的变量，va\_list arg\_ptr ,这个变量是指向参数地址的指针，因为得到参数的地址之后，再结合参数的类型，才能得到参数的值。

        <3>原型：type va\_arg(va\_list arg\_ptr,type);

              功能：获取下一个参数的地址

            返回值：根据传入参数类型决定返回值类型

        <4>原型：void  va\_end(va\_list arg\_ptr);

              功能：将arg\_ptr指针置0

             返回值：无

* 使用可变参数应该有以下步骤：

     ⑴在程序中将用到以下这些宏:

    void va\_start( va\_list arg\_ptr, prev\_param );

    type va\_arg( va\_list arg\_ptr, type );

    void va\_end( va\_list arg\_ptr );

    va在这里是variable-argument(可变参数)的意思.

    这些宏定义在stdarg.h中,所以用到可变参数的程序应该包含这个头文件.

    ⑵函数里首先定义一个va\_list型的变量,这里是arg\_ptr,这个变量是指向参数地址的指针.因为得到参数的地址之后，再结合参数的类型，才能得到参数的值。

    ⑶然后用va\_start宏初始化⑵中定义的变量arg\_ptr,这个宏的第二个参数是可变参数列表的前一个参数,也就是最后一个固定参数。

    ⑷然后依次用va\_arg宏使arg\_ptr返回可变参数的地址,得到这个地址之后，结合参数的类型，就可以得到参数的值。然后进行输出。

    ⑸设定结束条件，这里的条件就是判断参数值是否为-1。注意被调的函数在调用时是不知道可变参数的正确数目的，程序员必须自己在代码中指明结束条件。至于为什么它不会知道参数的数目，读者在看完下面这几个宏的内部实现机制后，自然就会明白。

三、参考文献：   
       <http://www.chineselinuxuniversity.net/articles/26262.shtml>

<http://www.cnblogs.com/wangyonghui/archive/2010/07/12/1776068.html>

<http://www.2cto.com/kf/201204/129038.html>