

C语言中可变参数的使用计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_C\\_E8\\_AF\\_AD\\_E8\\_A8\\_80\\_E4\\_B8\\_AD\\_c97\\_645858.htm](https://www.100test.com/kao_ti2020/645/2021_2022_C_E8_AF_AD_E8_A8_80_E4_B8_AD_c97_645858.htm) 一、什么是可变参数  
我们在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()函数,其函数原型为: int printf( const char\* format, ...). 它除了有一个参数format固定以外,后面跟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("%s",s). printf("the number is %d ,string is:%s", i, s). 以上这些东西已为大家所熟悉。但是究竟如何写可变参数的C函数以及这些可变参数的函数编译器是如何实现,这个问题却一直困扰了我好久。本文就这个问题进行一些探讨,希望能对大家有些帮助. 二、写一个简单的可变参数的C函数 先看例子程序。该函数至少有一个整数参数,其后是占位符..., 表示后面参数的个数不定. 在这个例子里,所有的输入参数必须都是整数,函数的功能是打印所有参数的值. 函数代码如下: //示例代码1: 可变参数函数的使用 #include "stdio.h" #include "stdarg.h" void simple\_va\_fun(int start, ...) { va\_list arg\_ptr. int nArgValue =start. int nArgCout=0. //可变参数的数目 va\_start(arg\_ptr,start). //以固定参数的地址为起点确定变参的内存起始地址。 do { nArgCout. printf("the %d th arg: %d\n",nArgCout,nArgValue). //输出各参数的值 nArgValue = va\_arg(arg\_ptr,int). //得到下一个可变参数的值 } while(nArgValue != -1). return. } int main(int argc, char\* argv[]) { simple\_va\_fun(100,-1). simple\_va\_fun(100,200,-1). return 0. } 从这个函数的实现可以看到,我们使用可变参数应该有以下步骤:

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. 来源: [www.100test.com](http://www.100test.com) 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. 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. 然后用`va_start`宏初始化 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. 来源: [www.examda.com](http://www.examda.com) 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.

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. 编辑特别推荐: 重解几道华为经典C语言面试题 应该怎样学习C语言? C语言学习应该注意的地方 计算机二级C语言程序设计实战 100Test 下载频道开通,各类考试题目直接下载. 详细请访问 [www.100test.com](http://www.100test.com)