

公告

昵称: gmh
园龄: 5年11个月
粉丝: 67
关注: 0
[+加关注](#)

<	2010年6月						>
日	一	二	三	四	五	六	
30	31	1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30	1	2	3	
4	5	6	7	8	9	10	

搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

随笔分类

- ARM11
- ARM7
- ARM9(3)
- BootLoader
- C语言(4)
- EVC
- FPGA(1)
- jade Z228
- LCD(4)
- linux 2.6.12(2)
- PCB制图(1)
- s3c2410(1)
- ucos(6)
- wince5.0电源管理(4)
- wince5.0调试方法(14)
- wince5.0函数介绍(9)
- wince5.0体系结构(4)
- wince5.0应用(3)

随笔-87 文章-0 评论-27

指针函数与函数指针的区别

一、

在学习arm过程中发现这“指针函数”与“函数指针”容易搞错，所以今天，我自己想一次把它搞清楚，找了一些资料，首先它们之间的定义：

1、指针函数是指带指针的函数，即本质是一个函数。函数返回类型是某一类型的指针

类型标识符 *函数名(参数表)

int *f(x, y);

首先它是一个函数，只不过这个函数的返回值是一个地址值。函数返回值必须用同类型的指针变量来接受，也就是说，指针函数一定有函数返回值，而且，在主调函数中，函数返回值必须赋给同类型的指针变量。

表示：

float *fun();

float *p;

p = fun(a);

注意指针函数与函数指针表示方法的不同，千万不要混淆。最简单的辨别方式就是看函数名前面的指针*号有没有被括号（）包含，如果被包含就是函数指针，反之则是指针函数。

来讲详细一些吧！请看下面

指针函数：

当一个函数声明其返回值为一个指针时，实际上就是返回一个地址给调用函数，以用于需要指针或地址的表达式中。

格式：

类型说明符 * 函数名(参数)

当然了，由于返回的是一个地址，所以类型说明符一般都是int。

例如：int *GetDate();

int * aaa(int,int);

函数返回的是一个地址值，经常使用在返回数组的某一元素地址上。

int * GetDate(int wk,int dy);

main()

wince5.0中断体系(3)
wince5.0子模块驱动(6)
Windows CE6.0调试方法(4)
Windows CE6.0函数介绍
Windows CE6.0体系结构
Windows CE6.0应用
Windows CE6.0中断体系
Windows CE6.0子模块驱动
电路调试(1)
电源设计(3)
电子类(3)
三极管，运放电路(4)
摄像头(4)
无线模块(1)
元器件选择对比(1)

随笔档案

2011年1月 (2)
2010年11月 (1)
2010年10月 (3)
2010年9月 (5)
2010年8月 (3)
2010年7月 (4)
2010年6月 (3)
2010年5月 (11)
2010年4月 (8)
2009年10月 (1)
2009年9月 (34)
2009年6月 (1)
2009年3月 (4)
2009年2月 (2)
2009年1月 (4)

最新评论

1. Re: 指针函数与函数指针的区别

讲解的很好，受益了

--领编程之美

2. Re: 指针函数与函数指针的区别

谢谢分享~~

--天之痕苏

3. Re: 三极管基极下拉电阻作用

学习了，谢谢

--几米™

4. Re: 指针函数与函数指针的区别

学习了，谢谢楼主！！！！

--Jacy_liu

5. Re: 指针函数与函数指针的区别

写的很好，有帮助

--xiashengwang

```

{
    int wk, dy;
    do
    {
        printf("Enter week (1-5) day (1-7) \n");
        scanf("%d%d", &wk, &dy);
    }
    while(wk<1||wk>5||dy<1||dy>7);
    printf("%d\n", *GetDate(wk, dy));
}

int * GetDate(int wk, int dy)
{
    static int calendar[5][7]=
    {
        {1, 2, 3, 4, 5, 6, 7},
        {8, 9, 10, 11, 12, 13, 14},
        {15, 16, 17, 18, 19, 20, 21},
        {22, 23, 24, 25, 26, 27, 28},
        {29, 30, 31, -1}
    };
    return &calendar[wk-1][dy-1];
}

```

程序应该是很好理解的，子函数返回的是数组某元素的地址。输出的是这个地址里的值。

2、函数指针是指向函数的指针变量，即本质是一个指针变量。

```

int (*f)(int x); /* 声明一个函数指针 */

f=func; /* 将func函数的首地址赋给指针f */

```

指向函数的指针包含了函数的地址，可以通过它来调用函数。声明格式如下：

类型说明符 (*函数名)(参数)

其实这里不能称为函数名，应该叫做指针的变量名。这个特殊的指针指向一个返回整型值的函数。指针的声明笔削和它指向函数的声明保持一致。

指针名和指针运算符外面的括号改变了默认的运算符优先级。如果没有圆括号，就变成了一个返回整型指针的函数的原型声明。

例如：

```
void (*fptr)();
```

把函数的地址赋值给函数指针，可以采用下面两种形式：

阅读排行榜

1. 指针函数与函数指针的区别(95902)
2. 集成运放组成的电压比较器(11213)
3. sscanf()的用法和例子(10618)
4. s3c2440—lcd控制器参数如何设定(转) (6808)
5. 函数WaitForMultipleObjects整理(6600)

评论排行榜

1. 指针函数与函数指针的区别(20)
2. 电视信号——行场同步(2)
3. 三极管基极下拉电阻作用(1)
4. 函数WaitForMultipleObjects整理(1)
5. [转]SDRAM的原理和时序(1)

推荐排行榜

1. 指针函数与函数指针的区别(40)
2. SD卡驱动相关内容(4)
3. sscanf()的用法和例子(3)
4. 集成运放组成的电压比较器(3)
5. wince流驱动开发流程 (转载)(2)

```
fptr=&Function;
```

```
fptr=Function;
```

取地址运算符&不是必需的，因为单单一个函数标识符就标号表示了它的地址，如果是函数调用，还必须包含一个圆括号括起来的参数表。

可以采用如下两种方式来通过指针调用函数：

```
x=(*fptr)();
```

```
x=fptr();
```

第二种格式看上去和函数调用无异。但是有些程序员倾向于使用第一种格式，因为它明确指出是通过指针而非函数名来调用函数的。下面举一个例子：

```
void (*funcp)();
```

```
void FileFunc(),EditFunc();
```

```
main()
```

```
{
```

```
    funcp=FileFunc;
```

```
    (*funcp)();
```

```
    funcp=EditFunc;
```

```
    (*funcp)();
```

```
}
```

```
void FileFunc()
```

```
{
```

```
    printf("FileFunc\n");
```

```
}
```

```
void EditFunc()
```

```
{
```

```
    printf("EditFunc\n");
```

```
}
```

程序输出为：

```
FileFunc
```

```
EditFunc
```

主要的区别是一个是指针变量，一个是函数。在使用是必要要搞清楚才能正确使用

二、指针的指针

指针的指针看上去有些令人费解。它们的声明有两个星号。

例如：

```
char ** cp;
```

如果有三个星号，那就是指针的指针的指针，四个星号就是指针的指针的指针的指针，依次类推。当你熟悉了简单的例子以后，就可以应付复杂的情况了。当然，实际程序中，一般也只用到了二

级指针，三个星号不常见，更别说四个星号了。

指针的指针需要用到指针的地址。

```
char  c=' A' ;
char  *p=&c;
char  **cp=&p;
```

通过指针的指针，不仅可以访问它指向的指针，还可以访问它指向的指针所指向的数据。下面就是几个这样的例子：

```
char  *p1=*cp;
char  c1=**cp;
```

你可能想知道这样的结构有什么用。利用指针的指针可以允许被调用函数修改局部指针变量和处理指针数组。

```
void FindCredit(int  **);

main()
{
    int  vals[]={7, 6, 5, -4, 3, 2, 1, 0};
    int  *fp=vals;
    FindCredit(&fp);
    printf("%d\n", *fp);
}

void FindCredit(int  ** fpp)
{
    while(**fpp!=0)
        if(**fpp<0) break;
    else  (*fpp)++;
}
```

首先用一个数组的地址初始化指针fp，然后把该指针的地址作为实参传递给函数FindCredit()。FindCredit()函数通过表达式**fpp间接地得到数组中的数据。为遍历数组以找到一个负值，FindCredit()函数进行自增运算的对象是调用者的指向数组的指针，而不是它自己的指向调用者指针的指针。语句(*fpp)++就是对形参指针指向的指针进行自增运算的。但是因为*运算符高于++运算符，所以圆括号在这里是必须的，如果没有圆括号，那么++运算符将作用于二重指针fpp上。

三、指向指针数组的指针

指针的指针另一用法旧处理指针数组。有些程序员喜欢用指针数组来代替多维数组，一个常见的用法就是处理字符串。

```
char  *Names[]=
{
    Bill,
    Sam,
    Jim,
```

```
        Paul,  
        Charles,  
        0  
    };  
  
    main()  
    {  
        char **nm=Names;  
        while(*nm!=0) printf("%s\n",*nm++);  
    }
```

先用字符型指针数组Names的地址来初始化指针nm。每次printf()的调用都首先传递指针nm指向的字符型指针，然后对nm进行自增运算使其指向数组的下一个元素(还是指针)。注意完成上述认为的语法为*nm++，它首先取得指针指向的内容，然后使指针自增。

注意数组中的最后一个元素被初始化为0，while循环以次来判断是否到了数组末尾。具有零值的指针常常被用做循环数组的终止符。程序员称零值指针为空指针(NULL)。采用空指针作为终止符，在树种增删元素时，就不必改动遍历数组的代码，因为此时数组仍然以空指针作为结束。

分类: [C语言](#)

绿色通道: [好文要顶](#) [关注我](#) [收藏该文](#) [与我联系](#)



gmh

关注 - 0

粉丝 - 67

[+加关注](#)

40

0

(请您对文章做出评价)

« [上一篇: 怎样使用Map文件进行查错](#)

» [下一篇: WinCE CEDDK之DMA相关函数](#)

posted @ 2010-06-11 09:25 gmh 阅读(95902) 评论(20) 编辑 收藏

评论列表

#1楼 2012-03-11 10:36 ITMelody

送人玫瑰，手有遗香，谢谢分享。。

支持(1) 反对(0)

#2楼 2012-03-17 10:10 %冷雨...

FindCredit()函数进行自增运算的对象是调用者的指向数组的指针，而不

是它自己的指向调用者指针的指针
这句有点难理解

支持(0) 反对(0)

#3楼 2012-09-21 09:09 岁月漫步

写的太好了，收益不少

支持(0) 反对(0)

#4楼 2012-10-29 18:53 juefeideng

好

支持(0) 反对(0)

#5楼 2012-12-11 19:40 菜鸟晕

好

支持(0) 反对(0)

#6楼 2013-02-22 11:49 cpoint

还行吧

支持(0) 反对(0)

#7楼 2013-04-20 20:59 hero2009nj

学习了，谢谢博主。

支持(0) 反对(0)

#8楼 2013-06-20 16:11 qin2013

写的很用心，受教了

支持(0) 反对(0)

#9楼 2013-09-26 19:07 wj704

上述的代码有些有问题，大家没发现？

支持(0) 反对(0)

#10楼 2014-03-19 22:31 fgh15975300

最后一个程序有问题。
char *Names[6]=
{
"Bil",
"Sam",
"Jim",
"Pau",
"Oha",
"O"
};
main()
{
char **nm=Names;
while(*nm!="O") printf("%s\n",*nm++);
}

支持(0) 反对(0)

#11楼 2014-07-24 15:03 任心愿

@fgh15975300
没有问题啊！！
是不是你复制的有问题啊；我已经试过了 没有问题的

支持(0) 反对(0)

#12楼 2014-08-20 09:57 唐 杰

说得很透彻

支持(0) 反对(0)

#13楼 2014-08-24 23:58 log_a

由于返回的是一个地址，所以类型说明符一般都是int。（这句不太懂，void、float等类型的变量也有地址啊。）

支持(2) 反对(0)

#14楼 2014-09-30 13:56 林特斯9527

好文当顶!!!

支持(0) 反对(0)

#15楼 2014-10-17 10:20 落雪成冰

好东西，有些东西不用就忘了，看 下回顾下。

支持(0) 反对(0)

#16楼 2014-12-03 21:15 小蘑菇00

@log_a

他说的有点问题，数据类型是由地址所指向的数据类型所决定的 你说的是对的

支持(0) 反对(0)

#17楼 2015-01-08 11:55 xiashengwang

写的很好，有帮助

支持(0) 反对(0)

#18楼 2015-03-18 14:46 Jacy_liu

学习了，谢谢楼主!!!!

支持(0) 反对(0)

#19楼 2015-03-26 22:28 天之痕苏

谢谢分享~~

支持(0) 反对(0)

#20楼 2015-04-27 10:57 领编程之美

讲解的很好，受益了

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

融云，免费为你的App加入IM功能——让你的App“聊”起来!!



最新IT新闻:

- **【答分歧】**程序员鼓励师到底是个什么鬼？
 - SpaceX猎户火箭通过NASA批准认证
 - 独立开发者：我是如何管理日常开发工作的
 - 易凯资本王冉：说完“C轮死”，再来说说“A股活”
 - 女科学家用砒霜治愈白血病
- » 更多新闻...



最新知识库文章:

- 大师是怎样炼成的
 - 元数据驱动设计 —— 设计一套用于API数据检索的灵活引擎
 - 持续部署，并不简单！
 - 如果你做的事情毫不费力，就是在浪费时间
 - 解析微服务架构（一）单块架构系统以及其面临的挑战
- » 更多知识库文章...