

3.7 指针和二维数组_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.7 指针和二维数组涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

数据类型 变量名[行数][列数];

例如:
int a[3][2];

特点:
(1)数组名 a
(2)数组元素: a[0][0],a[0][1]... a[2][1]
(3)实际内存中的排列方式

低地址
a[0][0]
a[0][1]
a[1][0]
a[1][1]
a[2][0]
a[2][1]
高地址
认为理解为:

a
a[0] a[0][0] a[0][1]
a[1] a[1][0] a[1][1]
a[2] a[2][0] a[2][1]

=====

经验: 看后面，可以确定它是什么

指针数组是一个数组

数组指针是一个指针

本质上是一个数组，这个数组中存放都是地址数据

```
int a = 10,b = 20,c = 30;  
  
int *p[3] = {&a,&b,&c};
```

简介

本质上是一个指针变量，这个变量可以用来记录一个数组类型的地址

```
int a[3]; => 去掉名字，剩下的是类型，int [3]  
&a => 表示获取数组类型的地址，此时这个地址类型是 int (*)[3];
```

```
int a[3][2];  
a表示的地址中存放了三个一维数组类型的数据，此时这个地址的类型是 int (*)[2]
```

a

a[0]	a[0][0]	a[0][1]
a[1]	a[1][0]	a[1][1]
a[2]	a[2][0]	a[2][1]

```
a[0]:{a[0][0],a[0][1]} => int [2]  
a[1]:{a[1][0],a[1][1]} => int [2]  
a[2]:{a[2][0],a[2][1]} => int [2]
```

```
a<====>&a[0]  
a[0]<====> int [2]  
&a[0]<====> int (*)[2]  
a<====>&a[0]<====>int (*)[2]  
*a<====>*(&a[0]) <====> *a<====>a[0]
```

```
*a<====>a[0]<====>&a[0][0]<====>int *
a[0][0]<====>int
```

总结:

```
int a[3]; int b[3][2];
a : int *      => a + 1 移动1个元素, 4byte
&a : int (*)[3] => &a + 1 移动3个元素, 12byte

b : int (*)[2] => b + 1 移动2个元素, 8byte
b[0]:int *      => b[0]+1 移动1个元素, 4byte
&b :int (*)[3][2]==>&b + 1 移动6个元素, 24byte
```

定义方式

数据类型 (*变量名)[元素的个数];
例如:
int (*p)[3];

移动

```
int (*p)[3];
数组指针每次移动的是一个数组的大小
p ++;
```

转换

数组类型指针---加*---->普通类型指针
int (*p)[3];
p的类型int (*)[3]-----加 * ----->*p的类型是int *

示例代码:

```
#include <stdio.h>

int main()
{
    int a[3];
    int b[3][2];

    printf("a = %x\n",a);
    printf("&a = %x\n",&a);
    printf("b = %x\n",b);
    printf("b[0] = %x\n",b[0]);

    printf("-----\n");

    printf("a + 1 = %x\n",a + 1);
    printf("&a + 1 = %x\n",&a + 1);
    printf("b + 1 = %x\n",b + 1);
    printf("b[0] + 1 = %x\n",b[0] + 1);

    return 0;
}
```

全文完

本文由 简悦 SimpRead 优化, 用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

