

3.4 一级指针和一维数组_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.4 一级指针和一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。

4. 一级指针和一维数组

```
int a[5];

(1)数组名代表数组首元素的地址.
a <====>&a[0]<====>)0xdf30

int a[5];

数组共有5个元素，分别是a[0]~a[4];

a[0] <====>int

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

=====
int a[5];

int *p = &a[0];

而a和&a[0]是等价的，故我们也可以如下写法.

int *p = a;
```

示例代码：

```
#include <stdio.h>

int main()
{
    int a[5] = {10,20,30,40,50};

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

    return 0;
}
```

输出结果：

```
a = 0xffb2e6b8
&a[0] = 0xffb2e6b8

int buf[5] = {10,20,30,40,50};
int *p = &buf[0];

buf[0]<====>int
buf<====>&buf[0]<====>int *;

结论：
buf[i]<====>*(buf + i)<====>*(p + i)<====>p[i]
```

示例代码：

```
#include <stdio.h>
```

```
int main()
{
    int a[5] = {10,20,30,40,50};
    int *p = a;

    printf("a[3] = %d\n",a[3]);
    printf("*(a + 3) = %d\n",*(a + 3));
    printf("*(p + 3) = %d\n",*(p + 3));
    printf("p[3] = %d\n",p[3]);
    return 0;
}
```

运行结果：

```
a[3] = 40
*(a + 3) = 40
*(p + 3) = 40
p[3] = 40
```

(1)一级指针主要保存地址，占4bytes，而数组有多个内存空间用来存放不同的数据，空间较大。

```
p的类型  int *
a的类型  int [5]
```

(2)数组名是一个地址常量，表示数组首元素的地址，常量不能更改。

```
int a[5] = {10,20,30,40,50};
int *p = a;

p++;
a++;
```

示例代码：

```
#include <stdio.h>

int main()
{
    int a[5] = {10,20,30,40,50};
    int *p = a;

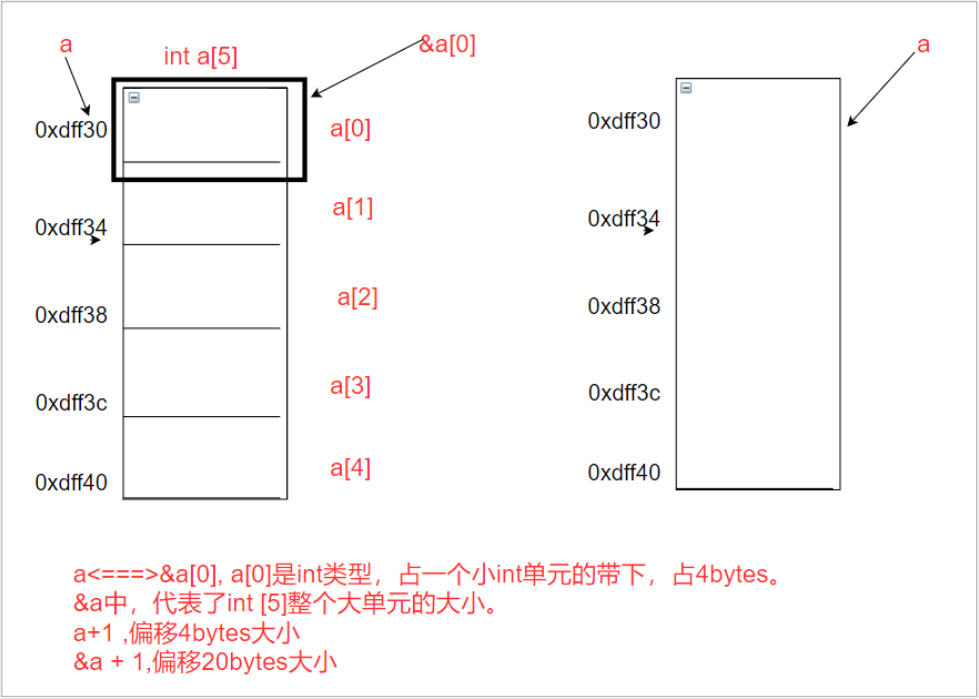
    p++;
    printf("*p = %d\n",*p);

    p++;
    printf("*p = %d\n",*p);

    return 0;
}
```

运行结果：

内存图解



结论

```
int a[5];
总结:
a <====>&a[0]

a + 1
&a[0] + 1

&a + 1

a[0]为int类型, 故&a[0] + 1一次偏移一个int类型的大小,4 bytes.
a为int [5]类型, 故&a + 1 一次偏移一个int [5]类型的大小,20 bytes.
=====

笔试题:

char a[5] = {10,20,31,32,40};

*(((char *)&a + 1)) - 3) - 2 =====>输出多少?

int a[5] = {10, 20, 30, 40, 50};
int *p = a;

要求通过a,a[i]和p,p[i]四种不同的形式输出数组的内容
```

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

