

3.5 指针的设计规则_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.5 指针的设计规则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。

本质:所有的指针都是用来保存地址的,只不过因为保存地址的数据类型不同,从而拥有多种指类型。

规则:一级指针变量是用来保存普通变量的地址。
二级指针变量是用来保存一级指针变量本身自己的地址。
三级指针变量是用来保存二级指针变量本身自己的地址。
.....

以二级指针为例:
定义方式 如下:
数据类型 ** 变量名;

int a = 10;
int *p = &a;

int **q = &p;
=====

p = &a; /

在 32bit 的系统中，所有的指针变量都是 4bytes

示例代码:

```
#include <stdio.h>

int main()
{
    int *p = NULL;

    char **p_char = (char **)&p;
    short **p_short = (short **)&p;
    int **p_int = (int **)&p;

    return 0;
}
```

运行结果:

```
sizeof(p_char) = 4
sizeof(p_short) = 4
sizeof(p_int) = 4
```

在 32bit 的系统中，多级指针（二级和二级以上）在移动的每次移动都是 4bytes. 因为一个指针大小是 4bytes

示例代码:

```
#include <stdio.h>

int main()
{
    int *p = NULL;

    char **p_char = (char **)&p;
    short **p_short = (short **)&p;
```

```

int **p_int = (int **)&p;

printf("p_char = %p\n", p_char);
printf("p_short = %p\n", p_short);
printf("p_int = %p\n", p_int);
printf("=====\n");
p_char++;
p_short++;
p_int ++;
printf("p_char = %p\n", p_char);
printf("p_short = %p\n", p_short);
printf("p_int = %p\n", p_int);
return 0;
}

```

运行结果:

```

p_char = 0xffff0fab
p_short = 0xffff0fab
p_int = 0xffff0fab
=====
p_char = 0xffff0fac
p_short = 0xffff0fac
p_int = 0xffff0fac

```

二级指针和一维数组的转换

```

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

而根据一维数组得出结论:
a[i] <====>*(a + i)<====>*(p + i)<====>p[i]

且当前 *q<====>p, 故

*(p + i)<====>p[i]<====>*( *q + i) <====>(*q)[i]

```

示例代码:

```

#include <stdio.h>

int main()
{
    int a[5] = {10, 11, 12, 13, 14};
    int *p = a;
    int **q = &p;

    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]);

    printf("=====\n");

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

```

运行结果:

```

a[3] = 13
*(a + 3) = 13
*(p + 3) = 13
p[3] = 13
=====

```

```
*(*q + 3) = 13  
(*q)[3] = 13
```

```
int a[5] = {1,3,5,7,9};  
int *p = NULL;  
int **q = NULL;
```

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

