

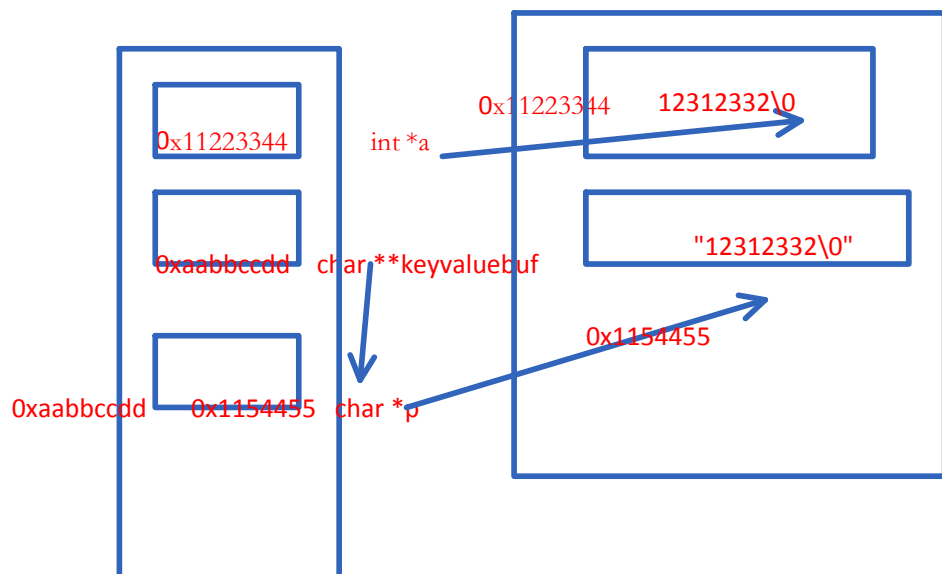
初始化要设为NULL

使用完之后，依然要设为NULL

2 错误模型

2015年11月15日 9:37

```
char *getKeyByValue(char **keyvaluebuf, char *keybuf)
{
    int i = 0;
    char *a = (char *)malloc(50);
    for (; **keyvaluebuf != '\0'; i++)
    {
        *a++ = *(p)++;
        *a a++
    }
    free(a);
}
```



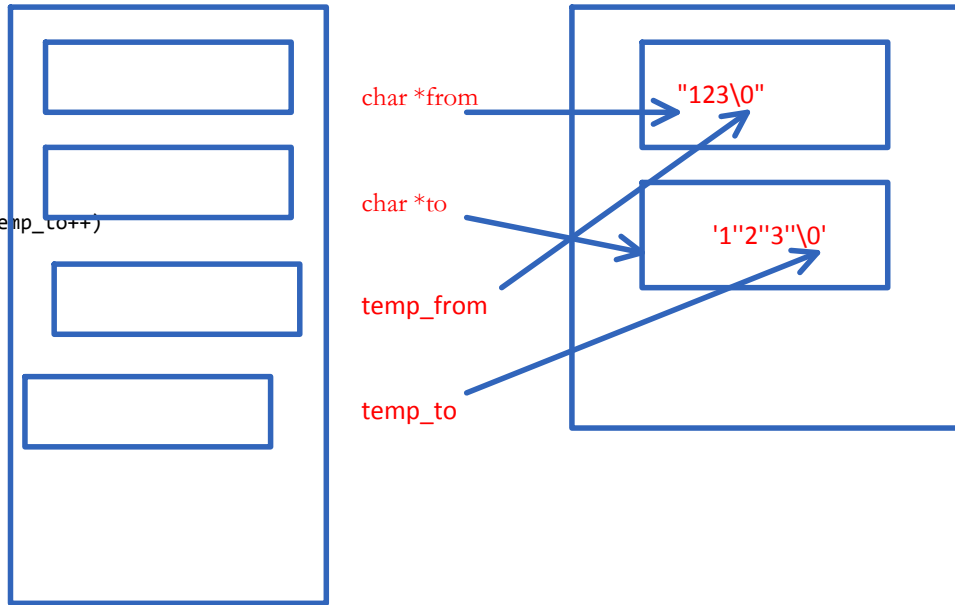
3 易错模型

2015年11月15日 9:44

```
void copy_str_err(char *from, char *to)
{
    char * temp_from = from;
    char *temp_to = to;

    for (; *temp_from!='\0'; temp_from++, temp_to++)
    {
        *temp_to= *temp_from;
    }
    *temp_to= '\0';

    printf("to:%s", to);
    printf("from:%s", from);
}
```



4 const

2015年11月15日 10:25

```
#define "abcdefghijklmn"STRING
const char string[]="ABCDEFGHIJK";
printf(STRING); //为STRING分配了第一次内存
printf(string); //为string一次分配了内存，以后不再分配
printf(STRING); //为STRING分配了第二次内存
printf(string);
```

常量区

"abcdefghijklmn"

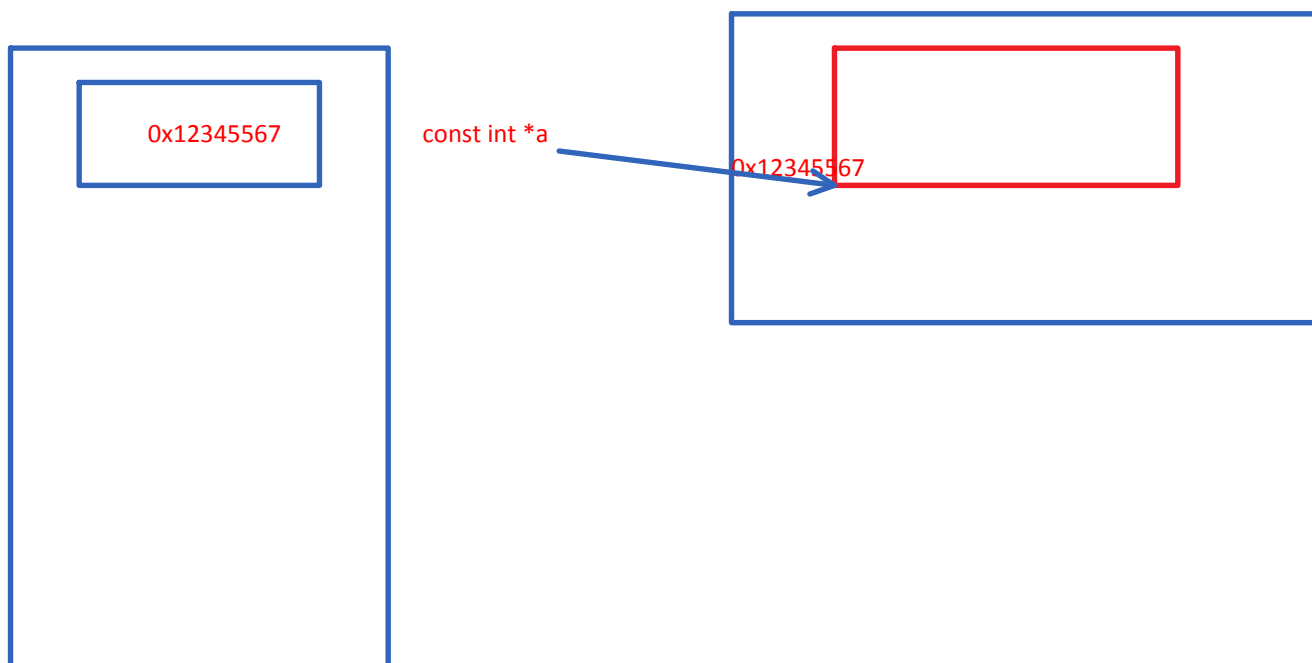
"abcdefghijklmn"

"ABCDEFGHIJK"

5 const

2015 年 11 月 15 日

10:29



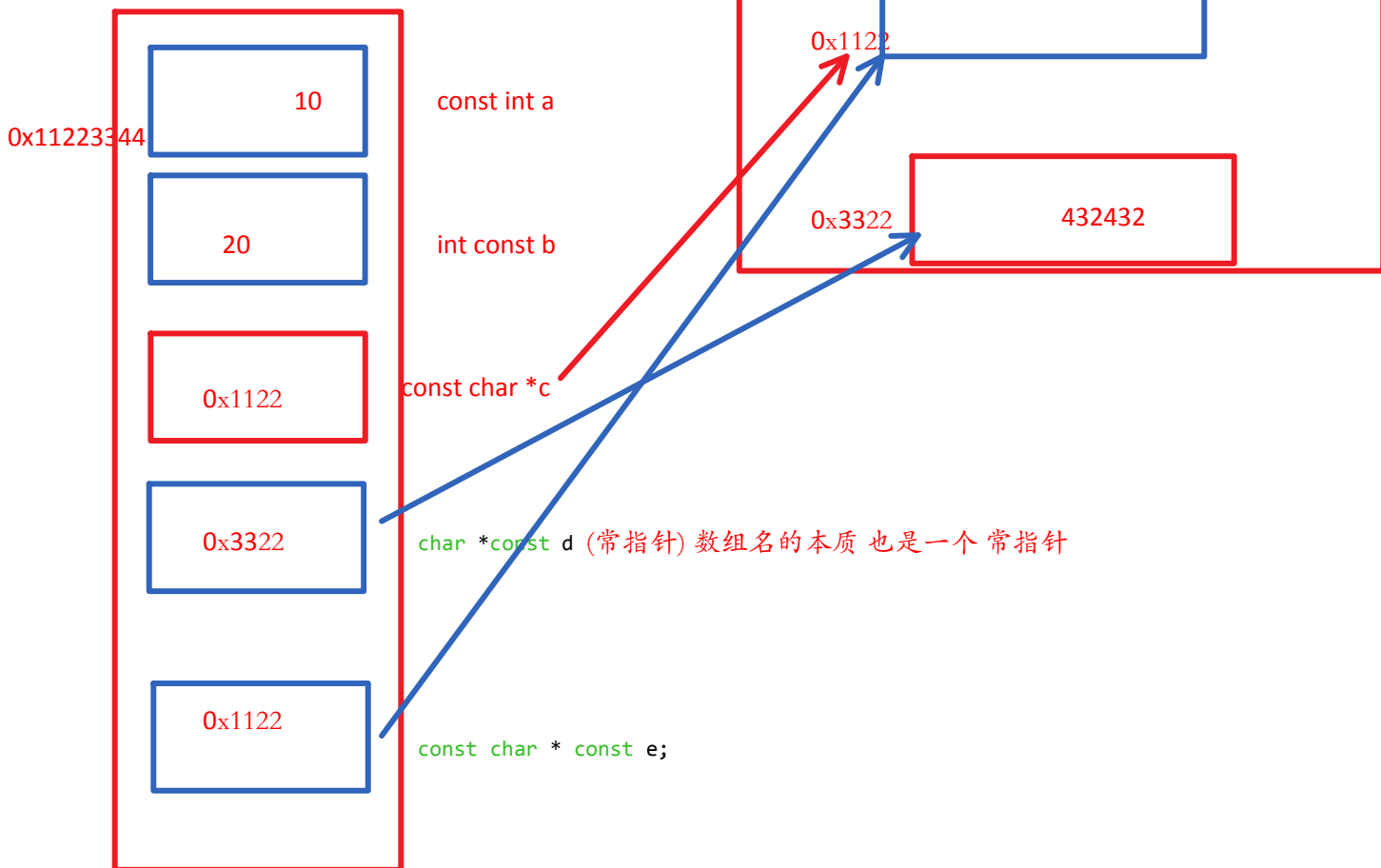
6 const的修饰种类

2015年11月15日 10:33

红色代表可以修改 蓝色不可以

```
int main(void)
{
    const int a; //代表一个常整型数
    int const b; //代表一个常整型数
    const char *c; // 是一个指向常整型数的指针
    // (所指向的内存数据不能修改,
    // 但是本身可以修改)

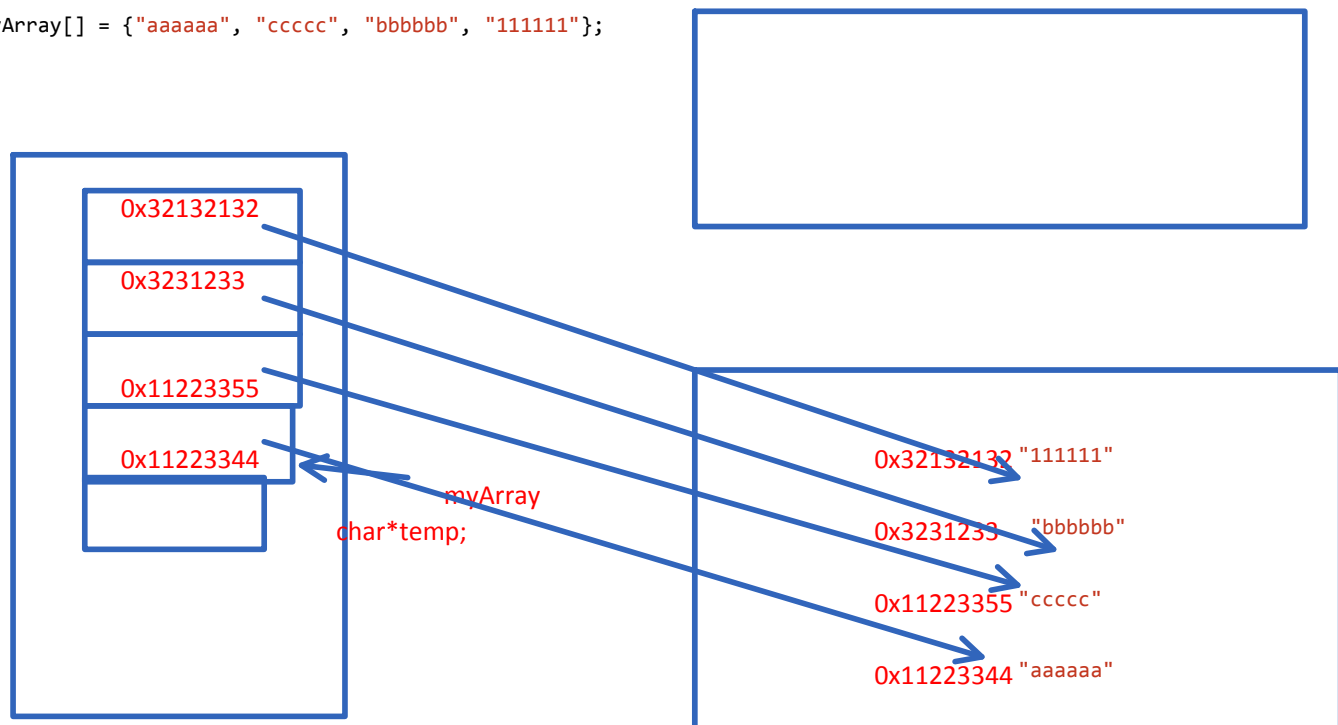
    char *const d; // 常指针(指针变量不能被修
    改,
    // 但是它所指向的内存空间可以被修改)
    const char * const e; // 一个指向常量整型的常
    指针
    // (指针和它所指向的内存空间,
    // 均不可以修改)
    return 0;
}
```



7 二级指针作为传输参数

2015年11月15日 15:14

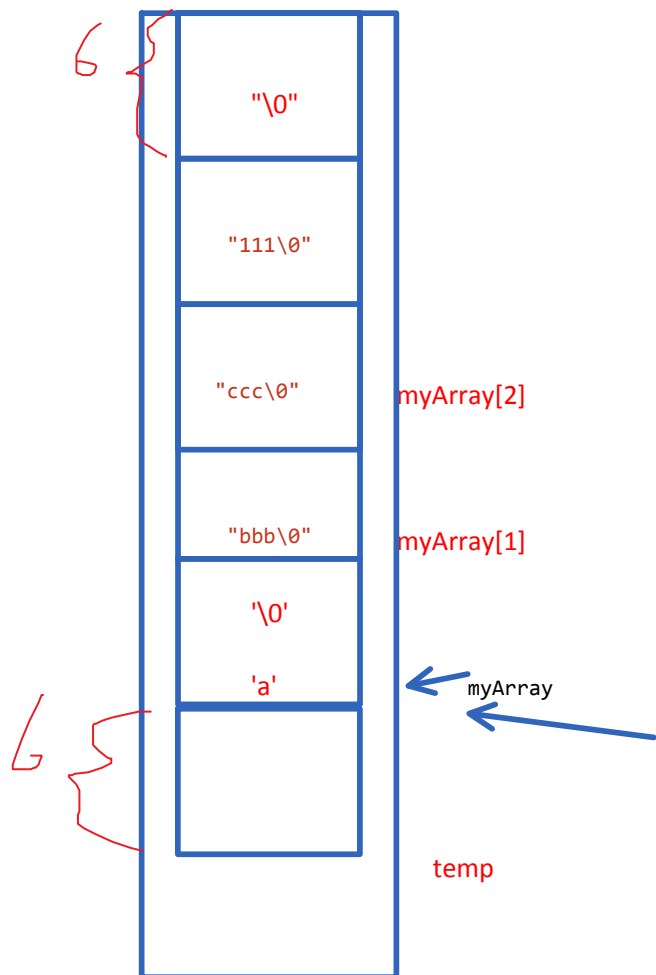
```
char *myArray[] = {"aaaaaa", "cccc", "bbbbbb", "111111"};
```



8 二级指针

2015年11月15日 15:58

```
char myArray[5][6] = {"aaa", "ccc", "bbb", "111", ""};
```



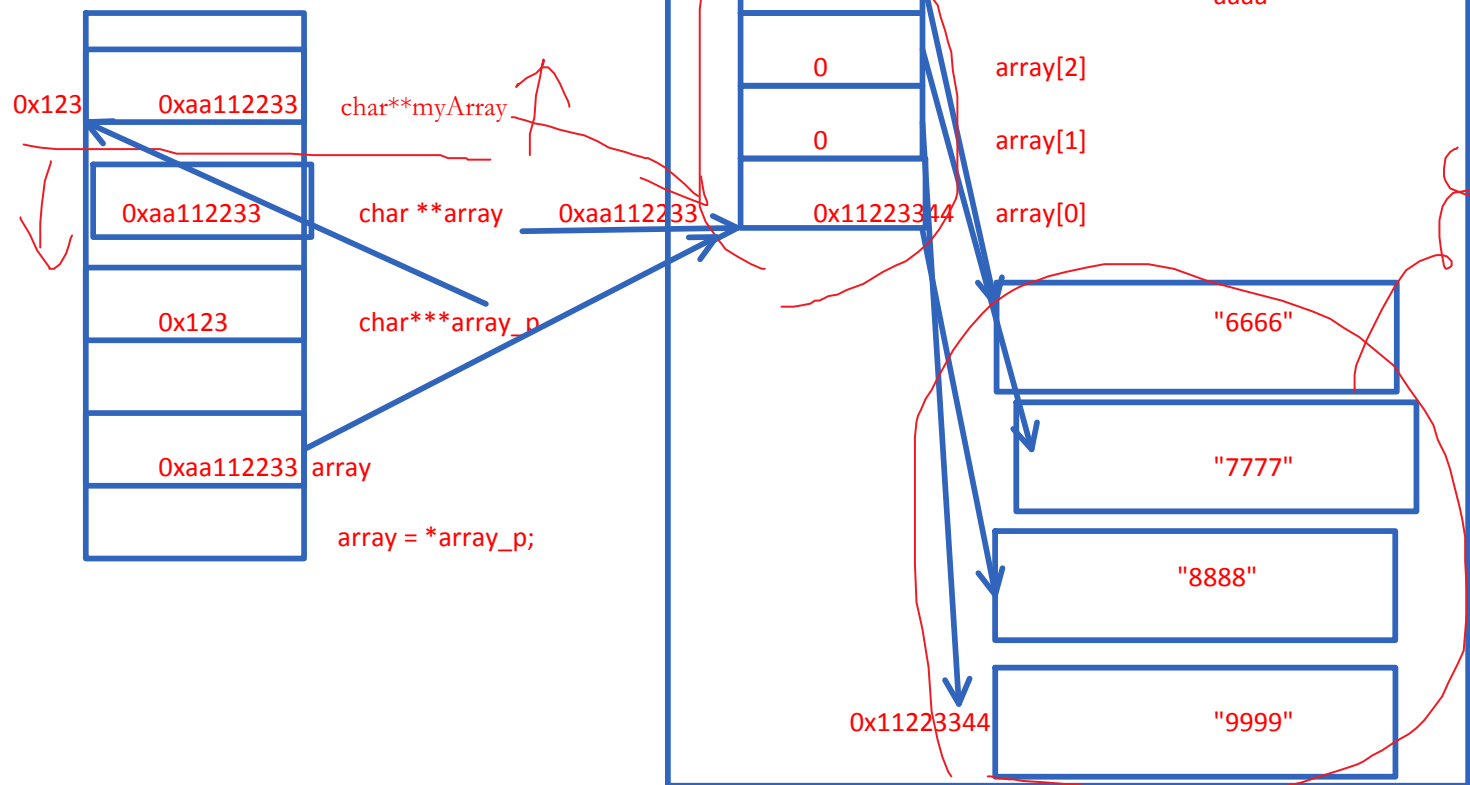
`char * *array == array+1 == unsigned int(array)+sizeof(char*)`

`(char[6]) *array ==> array+1`

9 二级指针第三种模型

2015年11月15日 16:25

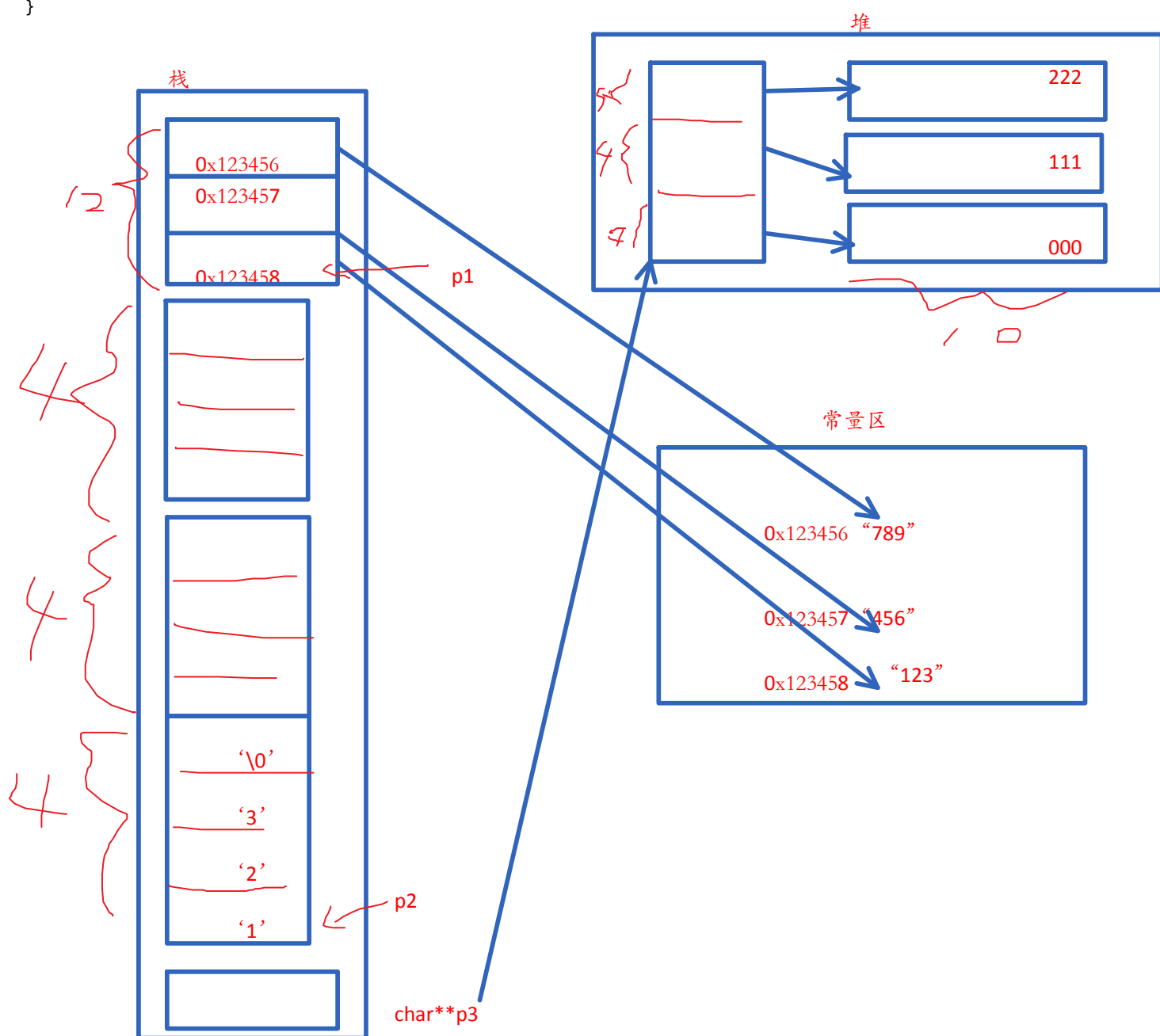
```
char **myArray = NULL;
```



10 四区图的练习

2015年11月15日 17:07

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
int main(void)
{
    int i = 0;
    //指针数组
    char *p1[] = {"123", "456", "789"};
    //二维数组
    char p2[3][4] = {"123", "456", "789"};
    //手工二维内存
    char **p3 = (char **)malloc(3 * sizeof(char *)); //int array[3];
    for (i=0; i<3; i++)
    {
        p3[i] = (char *)malloc(10*sizeof(char)); //char buf[10]
        sprintf(p3[i], "%d%d%d", i, i, i);
    }
    return 0;
}
```



11 替换子字符串

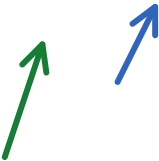
2015 年 11 月 15 日 17:18

ITCAST

strstr

5 2

12321321abcd12321edabcd213213abcd123



dst

12321321ITCAST12321edITCAST213213ITCAST