1. C语言const char\*、char const\*、char \*const的区别

（1）const char\*buf和char const\*buf表示的是指针变量buf所指向的内容是常量，任何对其进行赋值的操作都是非法的。

//任何对指针所指向的字符串进行的操作都是不允许的

const char \*buf1 = "abc";

char const \*buf2 = "abc";

buf1[0] = ‘y’; //非法操作

strcpy(buf2,”bnm”); //非法操作

（2）char \*const表示的是buf是常量，任何对指针进行的操作都是非法的。

//任何对指针进行的操作都是不允许的

char \*const buf3 = "abc";

buf3++; //非法操作

// strcpy(buf3,"abcd");

printf("out buf3:%s addr:0x%x\n",buf3,buf3);

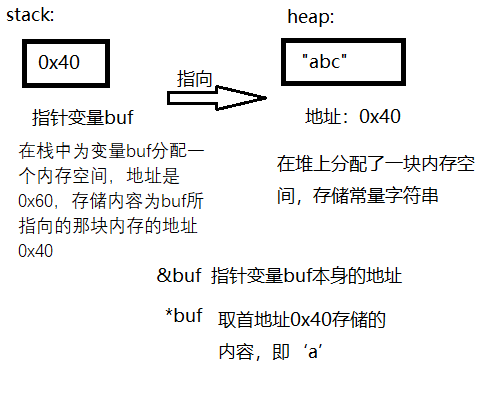
const char \*const buf4 = "abc";

表示指针buf是常量，所指向的内容也是常量，都不可以进行修改。

1. 指针与字符串

char \*buf = "abc";

在内存中的分配方式是下面这样的：



指针练习题：

char \*p;

char \*p[20];

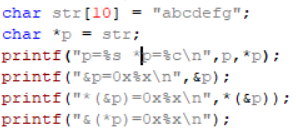
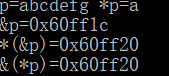
char p[30][20];

char \*\*p;

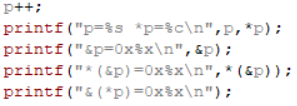
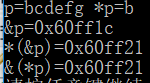
上面这些如何赋初值？

进行\*p, &p, \*(&p), &(\*p)操作后，结果是什么含义？

代码分析1：

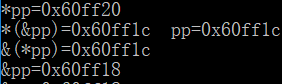
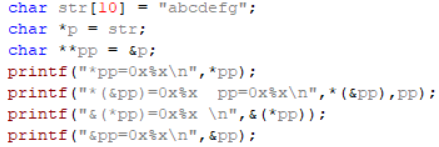
运行结果： 

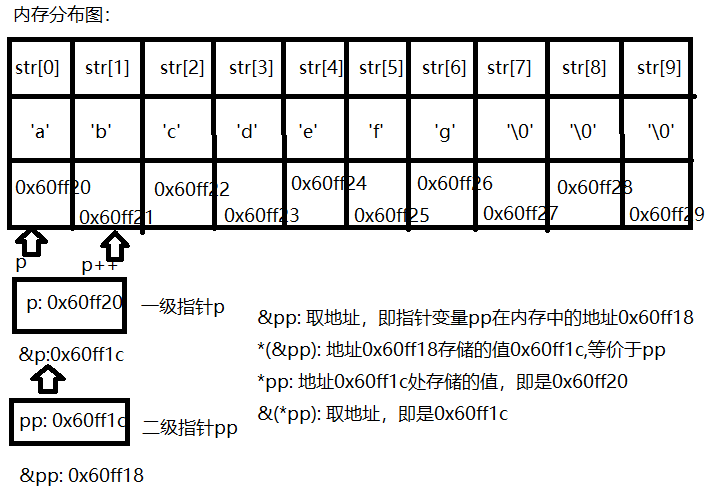
P++操作：

运行结果： 



代码分析二：

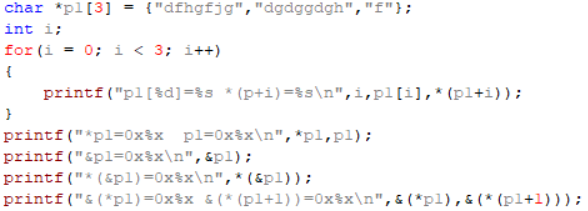




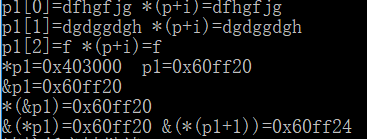
说明：pp++; 操作很危险，因为一次性向高地址方向偏移4字节（sizeof(char \*)），此时0x60ff18+4=0x60ff1c,如果再写入数据，就会引起死机。

代码分析三：

//说明：不能进行p1++操作，因为p1是数组名，是一个常量，不是指针变量

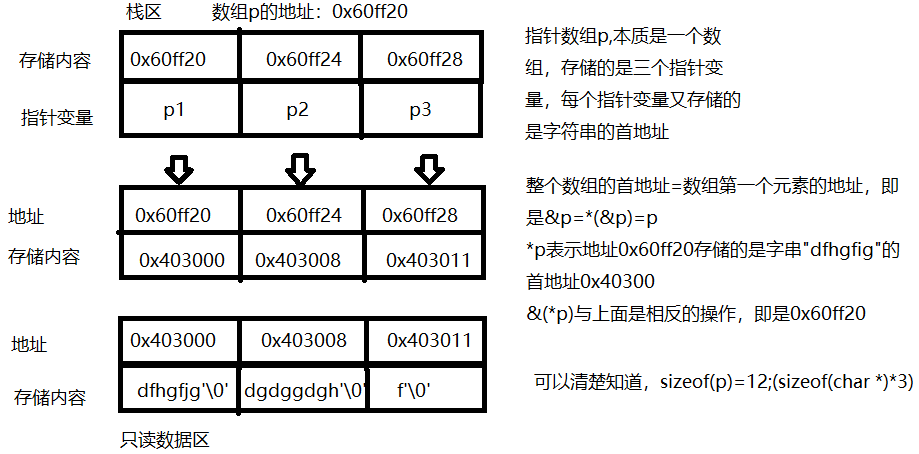


运行结果：

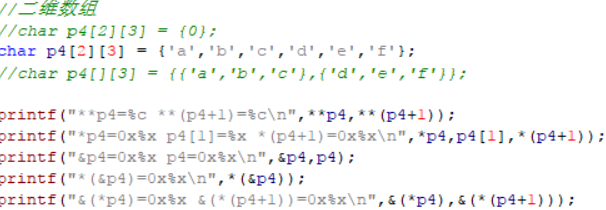


说明：不能进行p1++操作，因为p1是数组名，是一个常量，不是指针变量

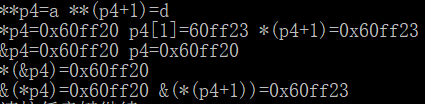
内存分析：



代码分析四：



输出结果：



内存分析：

