warning: function returns address of local variable

char \*ip\_proc(char \*p2)

{

int i;

long int num\_res[4];

char result[16];

for(i=0;i<4;i++)

{

char temp[3];

memcpy(temp,p2+i\*2,2);

temp[2]='\0';

num\_res[i]=strtol(temp,NULL,16);

}

sprintf(result,"%d.%d.%d.%d",num\_res[0],num\_res[1],num\_res[2],num\_res[3]);

return result;

}

在编译调用这个函数的程序时，产生一个warning，这个warning也正是自己想看到的，符合自己的预想：

我们知道一个函数的局部变量都是存在stack中的，当这个函数调用过程结束时，这个局部变量都是要释放掉的，所以就会产生这样的warning，这个是和变量的lifetime相关的，所以解决方法有：

1.将char result[16]改为static型，就是变为static char result[16];

2.使用malloc向heap申请，这些是需要caller用free去释放的

这里其实我想到了string.h里很多函数都是返回char \*的，比如strcat，strcpy，strchr等，看了看标准库里的实现，发现都是根据传进来的char \*来返回的，比如strcat.c

#include <string.h>

char \*(strcat)(char \*s1, const char \*s2)

{

char \*s;

for(s=s1;\*s!='\0';++s);

for(;(\*s=\*s2)!='\0';++s,++s2);

return (s1);

}

原文链接：https://blog.csdn.net/ricky\_hust/article/details/17226931