

## 参数传递

笔记本： 数据结构

创建时间： 2018/5/3 20:41

更新时间： 2018/5/3 21:29

作者： zzshneug@163.com

- 一种是按值传递方式。在这种参数传递方式下，把实参的值传递给函数局部工作区相应的副本中。函数使用副本执行必要的计算。因此函数实际修改的是副本的值，实参的值不变
- 一种是按地址传递参数。在这种参数传递方式下，需将形参声明为指针类型，即在参数名前加上符号“\*”。当一个实参与一个指针类型结合时，被传递的不是实参的值，而是实参的地址。函数通过地址存取被引用的实参。执行函数调用后，实参的值将发生改变
- 在C语言中数组参数的传递属特殊情形，**数组作为形参可按值传递方式声明，但实际传递的是数组第一个元素的地址。因此在函数体内对于形参数组所作的任何改变都会在实参数组中反映出来。**
- **malloc(), free()** 分配与释放内存

```
/*
> File Name: mallocTest.cpp
> Author: Zisheng Zou
> Mail: zzshneug@163.com
> Created Time: Thu May 3 21:03:05 2018
*/

#include<iostream>
#include<string.h>
#include<stdlib.h>
#include<stdio.h>

using namespace std;

int main(){
    char *str;
    if((str=(char *)malloc(10*sizeof(char)))==0){
        printf("the memory is not enough\n");
        exit(1);
    }
    strcpy(str,"Hello");
    printf("String is %s\n",str);
    // free the space
    free(str);
}
```

### • 二维数组

在许多情况下，当形式参数是一个二维数组时，必须指定其第二维大小。例如，`a[][10]`是一个合法的形式，而`a[][]`则不是。故可使用动态分配的二维数组。Example：

```
/*
> File Name: malloc2d.cpp
> Author: Zisheng Zou
> Mail: zzshneug@163.com
> Created Time: Thu May 3 21:16:55 2018
*/

#include<iostream>
#include <stdlib.h>
#include <stdio.h>
using namespace std;
int **malloc2arr(int r,int c){
    int i;
    int **t = (int **)malloc( r*sizeof(int *));
    for(int i=0;i<r;++i){
        t[i] = (int *)malloc(c*sizeof(int));
    }
    return t;
}
int main(){
    int ** arr = malloc2arr(3,4);
    arr[0][1] = 9;
    printf("the arr num is %d",arr[0][1]);
    return 0;
}
```

