

# 函数指针

## 1 基本概念

程序运行期间，每个函数都会占用一段连续的内存空间。而函数名就是该函数所占内存区域的起始地址（也称“入口地址”）。我们可以将函数的入口地址赋给一个指针变量，使该指针变量指向该函数。然后通过指针变量就可以调用这个函数。这种指向函数的指针变量称为“函数指针”。

**定义形式** 类型名 (\*指针变量名)(参数类型1, 参数类型2, ...);

**例如：** int(\*pf)(int,char);

表示pf是一个函数指针，它所指向的函数，返回值类型应该是int,该函数应有两个参数，第一个是int类型，第二个是char类型。

**使用方法** 可以用一个原型匹配的函数的名字给一个函数指针赋值。要通过函数指针调用它所指向的函数，写法为：

函数指针名（实参表）；

```
#include <iostream>
void PrintMin(int a,int b)
{
    if(a<b)
        printf("%d",a);
    else
        printf("%d",b);
}
int main()
{
    void(*pf)(int,int);
    int x=4,y=5;
    pf=PrintMin;
    pf(x,y);
    return 0;
}
```

## 2 函数指针和qsort库函数

c语言快速排序库函数:

```
void qsort(void*base,int nelem,unsigned int width,
int(*pfCompare)(const void*,const void*));
```

可以对任意类型的数组进行排序

base:待排序数组的起始地址

nelem:待排序数组的元素个数

width:待排序数组的每个元素的大小 (以字节为单位)

pfcompare:是一个函数指针, 指向一个比较函数, 作用就是告诉qsort该怎么排序。

## 3 实例

```
#include<stdio.h>
#include<stdlib.h>
int MyCompare(const void * elem1,const void * elem 2)
{
    unsigned int * p1,* p2;
    p1=(unsigned int *)elem1;///elem1非法,所以进行强制类型转换
    p2=(unsigned int *)elem2;///elem2非法
    return (* p1%10)-(* p2%10);
}
#define NUM 5
int main()
{
    unsigned int an[NUM]={8,123,11,10,4};
    qsort(an,NUM,sizeof(unsigned int),MyCompare);
    for(int i=0;i<NUM;i++)
        printf("%d",an[i]);
    return 0;
}
```