

# 实验九、C++指针

计算机学院 熊明 20305055

## 一、实验目的

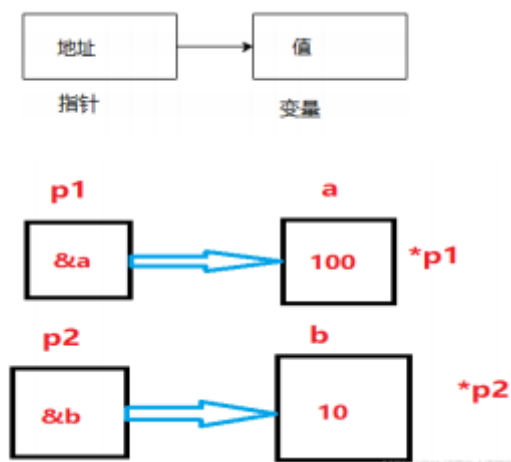
1. 掌握指针的概念和定义方法。
2. 掌握指针的操作符和指针的运算。
3. 掌握指针与数组的关系。
4. 掌握指针与字符串的关系。
5. 熟悉指针作为函数的参数及返回指针的函数。
6. 了解函数指针。
7. 掌握引用的概念，会定义和使用引用；
8. 掌握函数实参与形参的对应关系，熟悉“地址传递”和“引用传递”的方式。

## 二、实验原理

内存的问题是软件开发中最核心的问题之一

- 内存的基本和划分单位是字节，每个字节含有8位
- 内存的每个字节都有一个唯一确定的编号，这个编号叫地址
- CPU 只能通过地址来取得内存中的代码和数据，程序在执行过程中会告知 CPU 要执行的代码以及要读写的数据的地址。
- CPU 访问内存时需要的是地址，而不是变量名和函数名！变量名和函数名只是地址的一种助记符，当源文件被编译和链接成可执行程序后，它们都会被替换成地址。
- 编译和链接过程的一项重要任务就是找到这些名称所对应的地址。
- 指针即地址，地址是内存单元的编号，不可重复，但所存内容可以重复。
- 指针变量就是存放内存单元地址的变量。
- 指针能够对地址进行操作使C/C++能够执行更底层的操作。但是如果在程序设计中使用指针出错，可能造成严重的程序异常，甚至会导致死机。
- 指针可以使程序在速度或内存使用方面更有效率，并且可以用来构建复杂的数据结构。如链表、队列、堆栈和树。

由于通过地址能找到所需的变量单元，可以说，地址指向变量单元，打个比方，一个房间的门口挂了一个房间号2008，这个2008就是房间的地址，或者说，2008“指向”该房间。因此，将地址形象化地称为“指针”。意思是通过它能找到以它为地址的内存单元！



### 三、实验内容

1. 使用指针变量对一字符串按照字母、空格、数字和其他字符进行分类统计（提示：读一行字符包括空格用函数cin.getline(ch, 81))

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  int main(){
6      char str[100];
7      int count[4]={0};
8      char *p;
9      p = str;
10     cin.getline(str,100);
11     while(*p!='\0'){
12         if(*p>='A'&& *p<='Z' || *p>='a'&& *p<='z') count[0]++;
13         else if(*p==' ') count[1]++;
14         else if(*p>='0' && *p<='9') count[2]++;
15         else count[3]++;
16         p++;
17     }
18     cout<<"字母: "<<count[0]<<endl;
19     cout<<"空格: "<<count[1]<<endl;
20     cout<<"数字: "<<count[2]<<endl;
21     cout<<"其他字符: "<<count[3]<<endl;
22 }
```

运行结果：

```
abci 0 123 *&%
字母: 4
空格: 3
数字: 4
其他字符: 3
```

最开始的while循环写的是：

```
1  while(p){
2      if(*p>='A'&& *p<='Z' || *p>='a'&& *p<='z') count[0]++;
3      else if(*p==' ') count[1]++;
4      else if(*p>='0' && *p<='9') count[2]++;
5      else count[3]++;
6      p++;
7  }
```

报了段错误，原因是因为定义了数组为100大小，但是指针一直都有值，没有达到nullptr的条件，故会指向数组外，导致段错误。

2. 下面的程序中调用了findmax()函数，该函数寻找数组中的最大元素，将该元素的下标通过参数返回并返回其地址值，用指针编程实现findmax()函数。

```

1 #include <iostream>
2 using namespace std;
3 int * findmax(int * array, int size, int * index);
4 void main ( )
5 {
6     int a[10] = {33,91,54,67,82,37,85,63,19,68};
7     int * maxaddr;
8     int idx;
9     maxaddr = findmax(a, sizeof(a)/sizeof( * a), &idx);
10    cout<<idx<<endl<<maxaddr << endl<<a[idx] << endl;
11 }

```

代码入下：

```

1 #include <iostream>
2 using namespace std;
3 int * findmax(int * array, int size, int * index){
4     int max = array[0];//第一个元素赋值给max
5     for(int i=0;i<size;i++){//遍历数组，找到最大的数和其索引
6         if(max<array[i]) {max = array[i];*index = i;}
7     }
8     return array+*index;//返回array的第index个地址
9 }
10 int main()
11 {
12     int a[10] = {33,91,54,67,82,37,85,63,19,68};
13     int * maxaddr;
14     int idx;
15     maxaddr = findmax(a, sizeof(a)/sizeof( * a), &idx);
16     cout<<idx<<endl<<maxaddr << endl<<a[idx] << endl;
17 }

```

运行结果：

```

1
0x61fdf4
91

```

## 四、实验心得体会

归纳总结

1. 字符串处理在计算机中会经常遇到，因此，了解字符串的定义、初始化、输入输出、存储方式等至关重要。
2. C语言编译系统会自动的在字符串末尾添加'\0'标识符作为字符串结束的标志，因此，在循环读取字符串中的字符时可以使用这个标志作为循环跳出的判断。
3. 为了方便字符串的输入输出，C语言提供了除 scanf()之外的 getchar()、gets() 函数。可以方便我们对字符串进行输出输出。
4. 字符数组的数组名是字符串首地址，是一个常量，不可以对其进行运算，字符串指针是一个指针变量，可以进行运算，因此可以更灵活的操作字符串，提高字符串的处理效率

