**第十六章 指针和数组**

1.**指针**：一个存储对象的**地址**。

2.**数组**：存储器中被连续排列的一列数据。

3.**变元总是以值的形式从主调函数传递到被调用函数。**

4.声明指针变量：**类型 \*变量名;** 。星号\*前后空格可有可无，指针变量类型视为类型\*。指向指针的指针使用多个星号\*。

5.**地址运算符&**：生成它操作数的存储地址。

6.**间接运算符\***：间接操作存储对象里的值。

7.赋值运算符右侧的间接运算符\*生成lw指令，取出存储的值；赋值运算符左侧的间接运算符\*生成sw指令，将右值存储到存储单元中。

8.**scanf函数格式用字符串后的参数必须使用指针。**

9.**主调函数的局部变量可以在被调用函数中通过指针间接修改。**

10.指针变量可以赋值**NULL**，表示不指向任何变量的**空指针**。NULL是特别定义的预处理宏，等于0，因为没有一个有效的存储对象可以存储在单元0中，所以空指针不指向任何变量。

11.声明数组：**类型 数组名[数组大小];** 。数组名[0]被分配到最低存储地址，**数组名[n-1]**被分配到最高存储地址。

12.访问数组通过**基址+偏移量**访问。**数组的基址是数组名[0]的地址。**

13.**数组的大小使用预处理宏有时比较恰当。**

14.**格式%Nd表示十进制补码整数占N位，如果不足N位，则左边补空格。**

15.**格式%-Nd表示十进制补码整数占N位，如果不足N位，则右边补空格。**

16.原因：不会在调用函数时把每个元素从一个活动记录复制到另一个活动记录中而花费大量时间。

17.**一维数组的名字就是该数组的基址，可以直接使用但不可以被修改，类型是声明时的类型。**例如：int x[10];，则x，&x[0]，&x三者等价，类型都为int\*。

18.**函数声明和函数定义时的参数若为一维数组，数组大小可以留空，形如“类型 数组名[]”；也可以使用指针，形如“类型 \*数组名”。**例如：void function(**int x[]**); 或void function(**int \*x**);。

19.**函数引用时，参数若为数组必须使用地址（一般直接写数组名）。参数传递后被调函数对应的数组形式参数名被赋值为传递的数组参数基址。**

例如：int main(){

int x[10];

function(**x**);

}

void function(int cox[]){

cox[1]=0;

}

则cox被赋值为x（&x[0]）。

20.**主调函数的数组会在被调用函数中被直接修改并可见，包括使用“形式参数数组”或者引用“基址+偏移量指针”。偏移量是数组号而不用考虑所占存储单元数。**

例如：void function(int cox[]){

**cox[1]=0;** 或**\*(cox+1)=0;**

}

21.**字符串**：表示文本的字符序列，字符数组。

22.字符串用双引号” “包括。

23.字符串赋初值时编译器在末尾自动加上0（**’\0’**），不足位会补0。

24.**以空结尾的字符串’\0’占用一个数组元素。**

25.格式序列%s在printf函数中，打印以参数表示的地址开始的字符串，以’\0’结尾。

26.格式序列%s在scanf函数中，在输入流中读入从第一个非空字符开始，到下一个空白字符之间的字符，并在末尾自动加上一个’\0’，存储于以参数表示的地址为首地址的存储器中。最后的空白字符以及其后的内容保留在输入流中。

27.**scanf函数不检查数组大小，当输入超过数组大小时，将强制修改数组后面单元中的内容。**

28.字符串处理函数使用头文件string.h。

29.**C语言不提供防止超出数组大小（或边界）的保护措施，访问超出数组大小的存储单元会修改其中的内容，因此需要防御性编程。**

30.**C89规定编译程序必须知道数组大小，即声明时数组大小不能使用变量。**

31.二维数组声明：**类型 数组名[行数][列数];** 。分配空间时先按行后按列分配。

32.**二维数组名是该数组的基址，数组名[行数]是数组中该行的首地址。**例如：char a[5][5]; ，则a，&a[0][0]，a[0]，&a，&a[0]五者等价；a[1]，&a[1][0]，&a[1]三者等价。

33.**strcmp函数**比较两个字符串，原型为**int strcmp(char \*string1,char \*string2);** 。若string1大于string2，则返回1；若string1小于string2则返回-1；若string1等于string2，则返回0。比较方法为ASCII码字典序。

34.**strcpy函数**复制字符串，**strcpy(char \*string1,const char \*string2);** 。将string2复制到string1中。

35.**函数声明和定义时若参数为二维数组，一定要注意参数使用指针或地址。**

**①若在被调函数内对整个数组操作，则参数是整个数组，可以省略行数，函数引用参数是数组基址。**

例如：有二维数组int x[5][4]，则void function(int [5][4])或void function(int [][4])或void function(int (\*x)[4])中可以对整个数组x操作，函数引用为function(x);。

**②若在被调函数内对某行操作，则参数是指针。被调函数的形式参数名已经表示原数组的该行地址。**

例如：有二维数组int x[5][4]，若函数引用为function(x[2]);，则void function(int \*x)中只能操作第三行，x[3]表示x[2][3]。