指针—3.指针变量

如何定义一个指针变量？

定义指针变量的时候，**首先必须明确这是一个指针，所以必须保证“\*”运算符最先和变量名结合。**

定义指针变量的过程分如下三步：

|  |  |
| --- | --- |
| 描述 | 示例 |
| 1. 指定变量名，假定是p | p |
| 2. 明确p是个指针，所以前面加“\*” | \*p |
| 3. 明确这个指针指向什么类型 | 类型 \*p |

**注意：因为“\*运算符”的优先级比“[]运算符”和“（）运算符”低，所以在指向的类型是数组或者函数的时候，为了保证\*优先和p结合，必须要用（）括起来，如（\*p）。**

例如：

|  |  |
| --- | --- |
| 变量声明语句 | 说明 |
| int \*p | p是一个指向int的指针 |
| int \*\*p | p是一个指向int \*的指针 |
| int \*\*\*p | p是一个指向int \*\*的指针 |
| int \*p[10] | 此处[]优先级高于\*，所以p首先是个数组，然后int \*就表示数组成员的类型。 |
| int (\*p)[10] | 因为括号的原因，p首先和\*结合，所以p是指针，指向括号外的内容，p指向一个长10，每个成员是整型的数组。 |
| int \*p(void) | 此处()优先级高于\*，所以p首先是个函数，然后int \*表示函数的返回值 |
| int (\*p)(void) | 因为括号的原因，p首先和\*结合，所以p是指针，指向括号外的内容，p指向一个参数是void，返回值是int的函数。 |

当我们对指针进行强制类型转换的时候，前面括号中的类型也可以用上面的方法分析，区别只是少了变量名而已。

|  |  |
| --- | --- |
| 强制类型转换语句 | 说明 |
| (int \*)p | 把p转换为指向整型的指针 |
| (int \*\*)p | 把p转换为指向整形指针的指针 |
| (int \*[N])p | 错误，因为括号中是数组，不是指针类型 |
| (int (\*)[N])p | 因为括号的存在，这是个指针，指向长度为N，成员为int的数组。 |
| (int \*())p | 错误，因为括号中是函数，不是指针类型 |
| (int (\*)(void))p | 因为括号存在，这是个指针，指向参数为void，返回值为int的函数。 |

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

例1：请分析下列语句：

//语句一

const RIL\_RadioFunctions \*(\*rilInit)(const struct RIL\_Env \*, int, char \*\*);

……

//语句二

rilInit = (const RIL\_RadioFunctions \*(\*)(const struct RIL\_Env \*, int, char \*\*))dlsym(dlHandle, "RIL\_Init");

……

//语句三

funcs = rilInit(&s\_rilEnv, argc, rilArgv);

分析：

**语句一**定义了一个变量rilInit;

因为括号存在，所以rilInit先和\*号结合，所以rilInit是个指针；

指向参数为(const struct RIL\_Env \*, int, char \*\*)，返回值为const RIL\_RadioFunctions \*的一个函数。

综上，rilInit是一个指针变量，指向一个函数的指针变量。

**语句二**是个赋值语句，将右侧表达式的值赋给左侧的rilInit变量。

右侧表达式前部分是个类型的描述：(const RIL\_RadioFunctions \*(\*)(const struct RIL\_Env \*, int, char \*\*))，后半部分是个函数的调用：dlsym(dlHandle, "RIL\_Init")

所以右侧表达式的含义是：以dlHandle和 "RIL\_Init"为参数，调用函数lsym()，得到函数的返回值；

将函数的返回值强转为(const RIL\_RadioFunctions \*(\*)(const struct RIL\_Env \*, int, char \*\*))类型后赋给左侧的rilInit。

分析(const RIL\_RadioFunctions \*(\*)(const struct RIL\_Env \*, int, char \*\*))：注意最内侧的括号，首先扩住“\*运算符”，所以这首先是个指针类型，指向参数为(const struct RIL\_Env \*, int, char \*\*)，返回值为const RIL\_RadioFunctions \*的函数。

综上，语句二含义为将右侧函数调用的结果进行类型强转之后，赋给左侧的指针变量。

**语句三**通过rilInit调用rilInit指向的函数，将&s\_rilEnv和argc, rilArgv作为参数传入，将返回值赋给funcs。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

练习一：

上述例题中的rilInit是常量指针还是变量指针？

若是变量指针，怎样修改语句一，可以使其成为常量指针？

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

第24题（判断/4分）： (void \*)ptr 和 (\*(void\*\*)ptr)的值相同  
对 错

分析：

此题参考答案是“对”，但是就我的理解，应该是“错”。

假设如下，ptr在空间a，ptr内存放空间b的地址。

b

c

ptr

a

(void \*)ptr是把ptr强转成了void \*类型，但是不管类型怎么改变，ptr内的值没有变化，所以不管指向什么类型，ptr本身的值都是一样的，都是空间b的地址，即a空间的内容。

(\*(void \*\*)ptr)是先把ptr强转成void\*\*指针，然后有个指向运算，我们不管ptr被强转成了什么类型，因为ptr内是b空间首字节的地址，所以\*ptr都是b空间中所存储的内容，因为被强转成了void \*\*类型，所以需要把b空间中的内容当做一个void \*类型来看到，所以，(\*(void \*\*)ptr)得到的是b空间的内容。

综上，(void \*)ptr是a空间的内容，(\*(void\*\*)ptr)是b空间的内容，二者不同。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

练习二：

请编写程序验证以上结论。