week10 Experiment : C String and Functions

15331302 王翔宇 2015软工1班 教务一班

Deep thinking

Question 1

目的是方便程序中语句内联，使代码可以写得更加高效和简洁。
如： `int length = strlen(strcpy(s, t));`
当然，直接写`strcpy(s, t);`也可，只是没有用到其返回值。就像类似于`526;`这种（看起来没什么用的）语句，同样可以编译通过。

Question 2

Part 1

C语言最初的设计并非快速开发语言，其要求开发者对程序的每一处逻辑都了解的足够清楚。
检查越界的工作会额外占用很多时间，反之不检查越界，会使程序的运行效率大幅提升。
C语言是现存高级程序设计语言中程序执行效率最高的语言。不检查越界正是C语言被广泛应用于对性能要求苛刻的环境的原因之一。
优秀的程序员甚至可以利用这一特性来写出更加高效简洁的代码，完成安全编程中较为复杂的实现。（因为程序的操作不外乎修改内存中的数据、和以不同方式读取）

Part 2

不检查越界的特性，

1. 很可能导致初级C程序员犯很多危险错误
2. 更容易被黑客利用，且此类高危代码难以检测。

Question 3

编译错误。引用正确，错在修改常量。
以`char *s = "xxx"`定义字符串，在本质上依然是定义了一个指针。
字符串”xxx”被存储在常量存储区【特定的内存区域】而”xxx”实际上是返回了该字符串首个字符在常量存储区的地址。
显然，对常量进行修改会导致编译错误。

Question 4

1. 设计优秀的库函数耦合性一般极低，不免会出现库函数之间互相调用的情况。由于如果使用函数则必须其函数声明在发送call的函数体代码之前，所以函数原型直接全部放在前面，就几乎完全避免了这个问题。可以使开发者专心于代码设计。
2. 方便团队合作，以及有利于开发者对自己程序的了解（大幅度减少出现“我竟然还定义了这样一个函数”的情况）
3. 在编译的时候，编译器通常都会先处理一些特殊数据【宏定义，函数声明，变量声明】，此过程中编译器可以通过声明来预测整个定义需要的空间，并预留出空间，留给定义时使用。

4. 如果不事先声明，直接定义，在内存资源紧张的环境当中，很可能会造成系统崩溃，出现内存不足，无法分配。

Question 5

Part 1 : LeftRotate

1. 取模运算。左移/右移 (k + length * n)个字符位其实与移动k个字符等价。
2. 取模之后可以知道是字符串的哪一部分要移动到前面。将字符串看作`AB`,则要将其转

变成`BA`，可以通过倒转运算。设`A^T`为将字符串A逆序排列，则显然有

$$((A^T) * (B^T))^T = BA$$

则分别对两个子字符串原地倒置，最后将整个字符串倒置即可。

3. 应该还可以用递归，反复调用转置函数来实现，复杂度好像相比2的O(n)变成了O(NlogN)，得不偿失了，完成报告的时候我还没有详细想明白这个做法。
   我认为方案2是一个足够高效的解。

Part 2 : strContain

1. 最直接的方法就是开一个长度为128的数组，对str1进行桶排序。然后从str2开始每个字符向其中查找，若每个对应值都大于0，就可以返回1，否则不包含则可以返回0. 复杂度从O(n*m)降至O(n+m)。是最优复杂度。

2. （128的数组倒是不会爆）当然也可以充分利用开出内存的每一个（0,1）位，只要发现对应字符包含，该位置1即可。如一个int是32位，则128的桶排序只需要sizeof(int) * 4 的空间。位域同理。

反馈

在写报告的时候（还没做Deep thinking）还没有想法；
但这次的Deep thinking做完之后感觉获益巨大，如果可以的话，希望师兄可以多提供一些这方面的题目，非常感谢。

Add！

感觉我交了作业之后题目又变了的感觉，伤【我看错了吗】