doi: 10.7690/bgzdh.2014.06.025

一个 C 语言的命名问题

陈伟[1]，董勇[2]，刘彦艳[2]，郭磊[3]
（1. 中国兵器工业第五八研究所军品部，四川 绵阳 621000；2. 总装备部炮兵防空兵装备技术研究所，北京 100012；
3. 总装备部北京第三离职干部休养所，北京 100101）

摘要：针对一次全局函数和全局变量命名相同引起的软件故障问题，对 C 语言的命名问题进行研究。研究验证结果表明：当全局变量名与全局函数名相同时，变量指向了函数代码或函数指向了变量地址，这 2 种情况都会引起程序异常的发生。文中给出了避免该问题的方法。

关键词：软件；故障；异常；命名
中图分类号：TP311.11　　**文献标志码：**A

A Naming Problem of C Language

Chen Wei[1], Dong Yong[2], Liu Yanyan[2], Guo Lei[3]
(1. *Department of Military Products*, *No*. 58 *Research Institute of China Ordnance Industries*, *Mianyang* 621000, *China*;
2. *Institute of Equipment Technology of Artillery & Air Defense*, *The General Equipment Department*, *Beijing* 100012, *China*;
3. *No*. 3 *Rest of Beijing Cadres Leaving*, *The General Equipment Department*, *Beijing* 100101, *China*)

Abstract: Aiming at the software failure problem which caused by the same name of global function and global variable, research on C language naming problem. The research verification results show: when a global function name is same as a global variable name in a C program, the variable will point to the function code or the function will point to the variable space, and will result in software failure. The paper gives some suggestions.

Keywords: software; failure; exception; naming

0　引言

在软件开发中，由于疏忽、不了解等各种原因，程序难免会出现一些异常故障[1]。针对一个全局函数和一个全局变量命名相同引起的软件故障，笔者进行了探讨验证。

1　问题说明

一个嵌入式计算机软件在检测中出现了程序异常，异常名为 page fault。其检验布局示意如图 1。

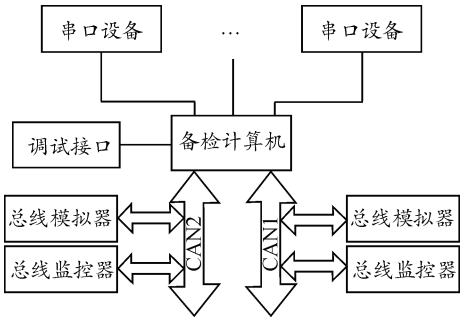

图 1　检测布局示意

2　命名问题

笔者通过对大量程序源代码和执行文件的分析盘查，确认了 2 个故障原因，其中 1 个问题就是全局变量和全局函数的命名相同所致。

在程序中要用到汉字的显示，而汉字显示程序中定义使用了一个全局整型变量 index，显示汉字时会对该变量赋值。在检测过程中，检测人员会访问存储设备，导致使用了函数 index。编程使用的 GNU 工具对 index 全局变量和 index 函数进行了合并处理，即被看成了同一个元素，运行时问题就出现了。通过跟踪发现，程序中对 index 全局变量的赋值变成了对 index 函数代码的修改，使 index 函数的调用出现异常。

在保证 index 变量设计功能的情况下，将其声明为静态变量后，相应的问题得到解决。

3　验证情况

3.1　文件准备

2 文件 test.c、test2.c 组成一个程序源文件，内容如下：

Test.c:

```
#include <string.h>
#include <stdio.h>
```

收稿日期：2014-02-18；修回日期：2014-03-05
作者简介：陈伟(1969—)，男，四川人，本科，研高工，从事软件研究。

```
void test2(int);
void main()
{
    char *str="sfdsjafdsfsdfG";
    char *pos;
    pos=strchr(str,'G');
    printf("%08x\n",pos);
    test2(1234);
    pos=strchr(str,'G');
    printf("%08x\n",pos);
    }
    Test2.c:
int strchr;
void test2(int val)
{
    strchr=val;
    }
```

3.2 验证情况 1

在库中有 strchr 函数，其声明在 string.h 文件中。

在 turbo C3.0 中建立 project，包含 2 个文件。编译时，编译器链接失败。

在 Visual C++ 6.0 中建立 project 编译，编译器链接成功，但运行时异常。

单步调试到 test.c 中的第一个 pos=strchr(str,'G') 处就异常。也就是说，在 VC 中，strchr 已经作为变量来看待了。

在 VS2008 中编译链接通过，运行时现象与 VC6.0 相同。

在 tornado 中编译连接通过，调试运行时访问 strchr 就会异常。

在 tornado2.2 中选择“Create a bootable VxWorks image(custom con…”，TOOL 选择 diab 建立 project，加入 test.c 和 test2.c，然后 build，仍未提示错误。

这样看来，因编译链接工具不一样，使结果有所不同。

3.3 验证情况 2

在 test2.c 文件加入 strchr 函数的声明文件如下：

```
    Test2.c:
#include <string.h>
int strchr;
void test2(int val)
{
    strchr=val;
    }
```

在 turbo C3.0 中建立 project，包含 2 个文件，编译器链接失败。

在 VC6.0 中编译链接失败，在 tornado 中也编译失败。

也就是说，如果声明充分，这种命名冲突是可以被编译链接识别的。如果熟悉系统，知道变量要出错，就可以更改变量名；如果对系统不熟悉，怎么指定哪些声明呢。

3.4 验证说明

验证中发现，C 语言中当全局变量名与全局函数名相同时，有时是变量指向了函数代码，有时是函数指向了变量地址，这 2 种情况都会引起异常的发生。

4 结论

1) 充分了解开发系统的关键字，避免作为自己的全局变量名使用；

2) 尽量使用静态说明，减小变量作用域；

3) 建立有效的私有的命名规则，避免与开发系统内部命名冲突；

4) 采用面向对象语言，减小命名冲突；

5) 改进编译工具，尽量识别命名冲突；

6) 避免自己的全局函数和全局变量命名冲突。

参考文献：

[1] 张萍. 嵌入式软件实时动态仿真系统设计[J]. 四川兵工学报, 2013, 34(5): 102-104.

[2] Windriver 公司. VxWorks Programmer's Guide[S]. 1999.

[3] Microsoft 公司. MSDN Visual Studio 6.0 [S].

[4] Borland 公司. Turbo C++6.0 help[S].