

C语言辅导:系统调用 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/285/2021_2022_C_E8_AF_AD_E8_A8_80_E8_BE_85_c97_285291.htm PC中最主要的难题之一，也是最容易引起误解的，就是系统调用。系统调用所代表的那些函数实际上是计算机的所有底层操作屏幕和磁盘的控制，键盘和鼠标的控制，文件系统的管理，时间，打印，这些只不过是系统调用所实现的一部分功能。总的来说，系统调用往往涉及到BIOS(基本输入输出系统)。实际中有好几种不同的BIOS，例如主板的BIOS负责初始硬件检测和系统引导，VGA BIOS(如果有VGA卡的话)处理所有的屏幕处理函数，固定磁盘BIOS管理硬盘驱动器，等等。DOS是位于这些低级BIOS之上的一个软件层，并且提供了进入这些低级BIOS的基本接口。一般说来，这意味着有一个DOS系统调用可以调用几乎所有你想使用的系统功能。实际上，DOS将调用相应的一种低级BIOS来完成所要求的任务。在本章中，你将会发现你既可以调用DOS来完成一项任务，也可以直接调用低级BIOS来完成相同的任务。

14.1 怎样检索环境变量

(environment variables)的值? ANSI C标准提供了一个名为getenv()的函数来完成这项任务。getenv()函数很简单一把指向要查找的环境串的指针传递给它，它就返回一个指向该变量值的指针。下面的程序说明了如何从C中获得环境变量PATH的值：

```
#include <stdio.h>
main(int argc, char * * argv) { char
envValue[129]. /* buffer to store PATH */ char * envPtr = envValue
. /* pointer to this buffer */ envPtr = getenv("PATH"). /* get the
PATH */ printf ("PATH= %s\n", envPtr) . /* print the PATH */ }
```

如果你编译并运行了这个程序，你就会看到与在DOS提示符下输入PATH命令完全相同的结果。事实上，你可以用getenv()检索AUTOEXEC . BAT文件中的或者系统引导后在DOS提示符下输入的所有环境变量的值。这里有一个小技巧。当运行Windows时，Windows设置了一个名为WINDIR的新的环境变量，它包含了Windows目录的路径全名。下面这段简单的程序用来检索这个串：

```
#include main(int argc, char * * argv) { char envValue[129]. char * envPtr = envValue . envPtr = getenv("windir"). /* print the Windows directory */ printf("The Windows Directory is %s\n", envPtr). }
```

这个程序还可以用来判断当前是否正在运行Windows，以及DOS程序是否运行在一个DOS shell下，而不是运行在“真正的”DOS下。注意，程序中的windir字符串是小写这一点很重要，因为它对大小写是敏感的。如果你使用WINDIR，getenv()就会返回一个NULL串(表示变量未找到错误)。用一putenv()函数也可以设置环境变量。但要注意，该函数不是一个ANSI标准函数，在某些编译程序中它可能不以这个名字出现，或者根本就不存在。你可以用一putenv()函数做许多事情。实际上，在上面那个例子中，Windows正是用这个函数创建了windir环境变量。请参：
14 . 2 怎样在程序中调用DOS函数? 14 . 3 怎样在程序中调用BIOS函数? 14 . 2 怎样在程序中调用DOS函数? 其实，当调用printf()，fopen()，fclose()，名字以一dos开始的函数以及很多其它函数时，都将调用DOS函数。Microsoft和Borland还提供了一对名为int86()和int86x()的函数，使你不仅可以调用DOS函数，还可以调用其它低级函数。用这些函数可以跳过标准的C函数而直接调用DOS函数，这常常可以节省你的

时间。下面的例子说明了如何通过调用DOS函数，而不是getch()和printf()函数，从键盘上得到一个字符并将其打印出来(该程序需要在大存储模式下编译)。

```
#include #include
char GetAKey(void). void OutputString(char * ). main(int argc,
char * * argv) { char str[128]. union REGS regs. int ch. /* copy
argument string. if none, use "Hello World" */ strcpy(str,
(argv[1]== NULL ? "Hello World": argv[1])), while ((ch =
GetAKey()) !=27){ OutputString(str). } } char GetAKeyO { union
REGS regs. regs.h. ah = 1. /* function 1 is "get keyboard character" */
/int86(0x21, amp.reg). return( (char)regs. h. al) . } void
OutputString(char * string) { union REGS regs. struct SREGS
segregs. /* terminate string for DOS function */ * (string
strlen(string)) = ' $ ' . regs.h. ah = 9. /* function 9 is "print a
string" */ regs.x. dx = FP_OFF(string) . segregs. ds= FP_SEG(string)
. int86x(0x21, amp.reg, &amp.segregs). }
```

上例创建了两个函数来代替getch()和printf()，它们是GetAKey()和OutputString()。实际上，函数GetAKey()与标准c函数getche()更为相似，因为它与getche()一样，都把键入的字符打印在屏幕上。这两个函数中分别通过int86()(在GetAKey()中)和int86x()(在OutputString()中)调用DOS函数来完成所要求的任务。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com