    在前一篇文章中，我们介绍了如何在Ubuntu上为Android系统编写Linux内核驱动程序。在这个名为hello的Linux内核驱动程序中，创建三个不同的文件节点来供用户空间访问，分别是传统的设备文件/dev/hello、proc系统文件/proc/hello和devfs系统属性文件/sys/class/hello/hello/val。进一步，还通过cat命令来直接访问/proc/hello和/sys/class/hello/hello/val文件来，以验证驱动程序的正确性。在这一篇文章里，我们将通过自己编写的C可执行程序来访问设备文件/dev/hello。可能读者会觉得奇怪，怎么能在Android系统中用C语言来编写应用程序呢？Android系统上的应用程序不都是Java应用程序吗？其实是可以的，读者不妨用adb shell命令连上Android模拟器，在/system/bin目录下可以看到很多C可执行程序，如cat命令。今天，我们就来学习一下怎么在Android系统中添加用C语言编写的可执行程序吧。

      一. 参照[在Ubuntu上为Android系统编写Linux内核驱动程序](http://blog.csdn.net/luoshengyang/article/details/6568411)一文，准备好Linux驱动程序。使用Android模拟器加载包含这个Linux驱动程序的内核文件，并且使用adb shell命令连接上模拟，验证在/dev目录中存在设备文件hello。

      二. 进入到Android源代码工程的external目录，创建hello目录:

**USER-NAME@MACHINE-NAME:~/Android$ cd external**

**USER-NAME@MACHINE-NAME:~/Android/external$ mkdir hello**

**三.**在hello目录中新建hello.c文件：

[view plain](http://blog.csdn.net/luoshengyang/article/details/6571210)

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <fcntl.h>
4. #define DEVICE\_NAME "/dev/hello"
5. **int** main(**int** argc, **char**\*\* argv)
6. {
7. **int** fd = -1;
8. **int** val = 0;
9. fd = open(DEVICE\_NAME, O\_RDWR);
10. **if**(fd == -1) {
11. printf("Failed to open device %s./n", DEVICE\_NAME);
12. **return** -1;
13. }
15. printf("Read original value:/n");
16. read(fd, &val, **sizeof**(val));
17. printf("%d./n/n", val);
18. val = 5;
19. printf("Write value %d to %s./n/n", val, DEVICE\_NAME);
20. write(fd, &val, **sizeof**(val));
22. printf("Read the value again:/n");
23. read(fd, &val, **sizeof**(val));
24. printf("%d./n/n", val);
25. close(fd);
26. **return** 0;
27. }

这个程序的作用中，打开/dev/hello文件，然后先读出/dev/hello文件中的值，接着写入值5到/dev/hello中去，最后再次读出/dev/hello文件中的值，看看是否是我们刚才写入的值5。从/dev/hello文件读写的值实际上就是我们虚拟的硬件的寄存器val的值。

四. 在hello目录中新建Android.mk文件：

**LOCAL\_PATH := $(call my-dir)**

**include $(CLEAR\_VARS)**

**LOCAL\_MODULE\_TAGS := optional**

**LOCAL\_MODULE := hello**

**LOCAL\_SRC\_FILES := $(call all-subdir-c-files)**

**include $(BUILD\_EXECUTABLE)**

注意，BUILD\_EXECUTABLE表示我们要编译的是可执行程序。

五. 参照[如何单独编译Android源代码中的模块](http://blog.csdn.net/luoshengyang/article/details/6566662)一文，使用mmm命令进行编译：

**USER-NAME@MACHINE-NAME:~/Android$ mmm ./external/hello**

编译成功后，就可以在out/target/product/gernerc/system/bin目录下，看到可执行文件hello了。

      六. 重新打包Android系统文件system.img：

**USER-NAME@MACHINE-NAME:~/Android$ make snod**

这样，重新打包后的system.img文件就包含刚才编译好的hello可执行文件了。

      七. 运行模拟器，使用/system/bin/hello可执行程序来访问Linux内核驱动程序：

**USER-NAME@MACHINE-NAME:~/Android$ emulator -kernel ./kernel/common/arch/arm/boot/zImage &**

**USER-NAME@MACHINE-NAME:~/Android$ adb shell**

**root@android:/ # cd system/bin**

**root@android:/system/bin # ./hello**

**Read the original value:**

**0.**

**Write value 5 to /dev/hello.**

**Read the value again:**

**5.**

看到这个结果，就说我们编写的C可执行程序可以访问我们编写的Linux内核驱动程序了。

      介绍完了如何使用C语言编写的可执行程序来访问我们的Linux内核驱动程序，读者可能会问，能不能在Android的Application Frameworks提供Java接口来访问Linux内核驱动程序呢？可以的，接下来的几篇文章中，我们将介绍如何在Android的Application Frameworks中，增加Java接口来访问Linux内核驱动程序，敬请期待。