

# 如何在Linux下寫一個自己的命令？

嵌入式大雜燴 昨天

以下文章來源於嵌入式情報局，作者情報小哥



**嵌入式情報局**

集單片機與linux於一身的技術號，深漂三年仍成功上岸研究生，這裡不僅有技術還有故事，只為分享全面的嵌入式知識和深度的職場趣事， ...

關注「**嵌入式大雜燴**」，星標公眾號，一起進步！



微信搜一搜



嵌

01

什麼是命令？

很多朋友對Linux的一個印象就是各種各樣的命令行，在圖形化界面不夠成熟的時期，Linux系統開發人員都是通過命令行直接操控OS，其實和DOS系統差不多，幾乎可以通過鍵盤搞定所有的事情，不要和鍵盤俠混談哦。

然而現在的ubuntu等等Linux的桌面版本僅僅只是為了讓更多的人去接觸Linux這款優秀的開源操作系統，像同期發布的Linux服務器版本還是一直以命令行為主，其實大多數的圖形操作僅僅只是一些命令的封裝罷了。

windows為什麼這麼廣泛使用，其圖形化界面操作是重要的因素，同時圖形傻瓜式操作來滿足更多不同層次的人使用該操作系統，而就因為傻瓜式的操作比較多，大部分工作丟給圖形交互和操作系統了，所以往往功能都比較單一且效率較低；而Linux操作系統更多的是發展命令行，命令行會把更多的功能選擇權交給用戶，且一次還可以同時完成多項事務，所以操作效率上Linux肯定是穩勝的，但是諸多的命令行往往叫很多人望而生畏。

命令並沒有那麼可怕，只是大部分人習慣了windows環境下的傻瓜式操作，而不太適應Linux下枯燥的命令罷了。

所谓的命令不是什么神器的东西，它仅仅也只是是一个应用程序罢了，你可以把命令行+回车看成windows下双击应用程序，但是通过简单的敲击命令并传递参数的就比图形化双击应用程序要灵活多了。

## 02 自己写一个命令

既然前面我们讲到命令就是程序，那这里写个程序编译成可执行程序看是否可以执行。

```
1  #include <stdio.h>
2
3  int main(int argc, char *argv[])
4  {
5      int i = 0;
6
7      printf("execute my command!\n");
8      for(i = 0 ;i < argc;i++)
9      {
10         printf("input param:%d - %s\n",i,argv[i]);
11     }
12     return 1;
13 }
```

上面是非常简单的一个C程序，主要的功能就是打印外部调用传入的参数。

然后采用GCC编译器，编译执行得到如下结果：

```
root@linux-eos: /home/Test6/Demo10
root@linux-eos:/home/Test6/Demo10# gcc mycommand.c -o mycommand
root@linux-eos:/home/Test6/Demo10# ls
mycommand mycommand.c
root@linux-eos:/home/Test6/Demo10# ./mycommand
execute my command!
input param:0 - ./mycommand
root@linux-eos:/home/Test6/Demo10# ./mycommand 123 456 789
execute my command!
input param:0 - ./mycommand
input param:1 - 123
input param:2 - 456
input param:3 - 789
root@linux-eos:/home/Test6/Demo10#
```

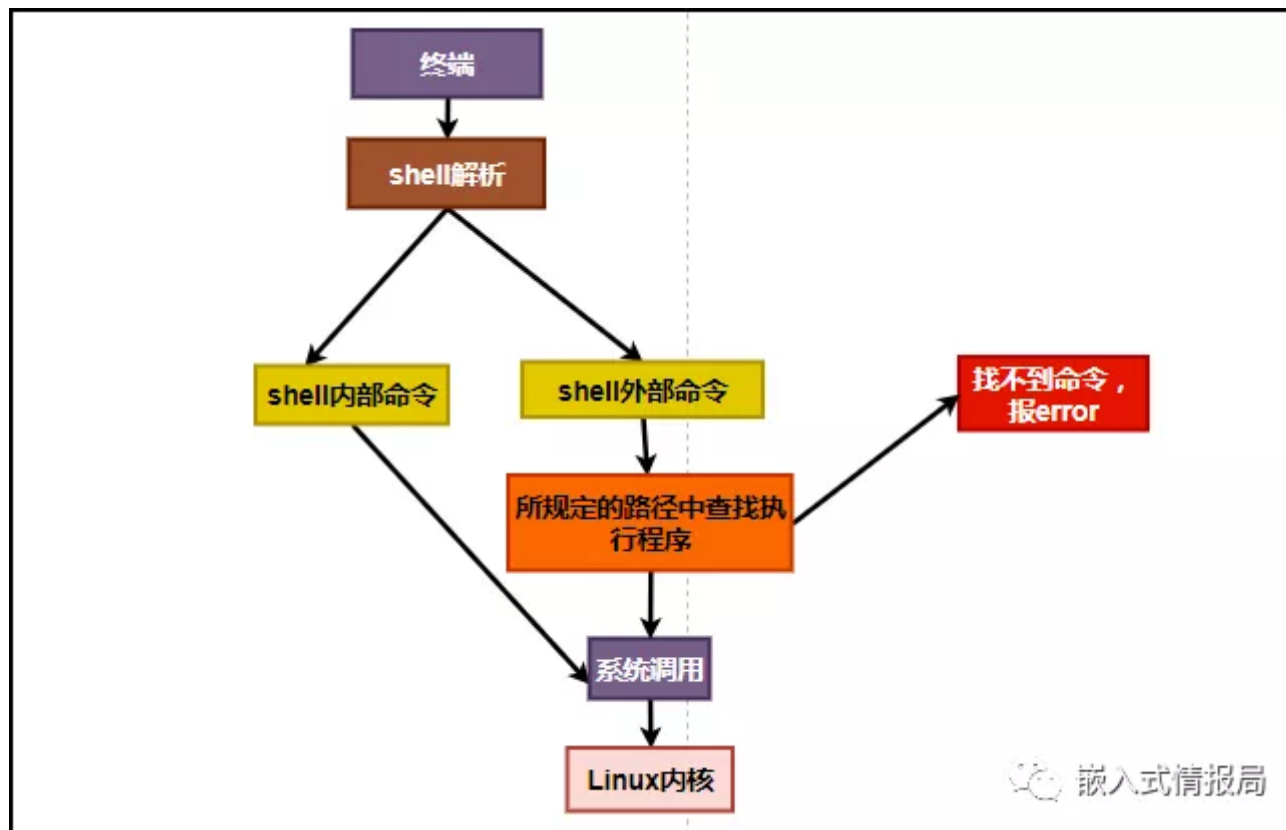
嵌入式情报局

我们采用"./应用程序"的形式来执行一切正常，但当我们直接输入执行程序命令，却没有得到满意的结果，如下图所示：

```
root@linux-eos: /home/Test6/Demo10
root@linux-eos:/home/Test6/Demo10# ls
mycommand mycommand.c
root@linux-eos:/home/Test6/Demo10# mycommand
mycommand: command not found
root@linux-eos:/home/Test6/Demo10#
```

嵌入式情报局

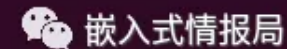
首先我们应该理解shell，Linux下常用shell就属于bash了，当然还有很多其他种类，shell你可以认为它就是一个命令解析器，当你输入这些命令行并回车的时候，首先其会经过shell进行解析以后，再由操作系统执行，大致过程如下：



从上面shell的大致运行情况可以了解到，当shell找不到对应的命令便会去环境变量中设置的路径中进行查找相应的可执行程序来执行。

那么是不是我们把当前路径加入到环境变量中就可以了呢？下面来简单验证一下：

```
root@linux-eos:/home/Test6/Demo10# ls
mycommand mycommand.c
root@linux-eos:/home/Test6/Demo10# mycommand
mycommand: command not found
root@linux-eos:/home/Test6/Demo10# pwd
/home/Test6/Demo10
root@linux-eos:/home/Test6/Demo10# export PATH=/home/Test6/Demo10:$PATH
root@linux-eos:/home/Test6/Demo10# echo $PATH
/home/Test6/Demo10:/home/qingbaoju:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr
/bin:/sbin:/bin:/usr/games:/usr/local/games:/work/tools/gcc-3.4.5-glibc-2.3.6/bi
n
root@linux-eos:/home/Test6/Demo10# mycommand
execute my command!
input param:0 - mycommand
root@linux-eos:/home/Test6/Demo10# mycommand 123 456
execute my command!
input param:0 - mycommand
input param:1 - 123
input param:2 - 456
root@linux-eos:/home/Test6/Demo10# cd ../
root@linux-eos:/home/Test6# m
Display all 132 possibilities? (y or n)
root@linux-eos:/home/Test6# mycommand
execute my command!
input param:0 - mycommand
root@linux-eos:/home/Test6# mycommand 123 456
execute my command!
input param:0 - mycommand
input param:1 - 123
input param:2 - 456
root@linux-eos:/home/Test6#
```



临时采用export命令把当前mycommand应用程序路径加入到环境变量PATH中，然后直接运行mycommand 参数，便可以获得对应的应用程序输出信息。

这样就实现了一个简单的命令~

## 最后

好了，这里小哥就介绍了一下Linux下命令的基本原理以及shell的相关知识，希望本文能够对你有帮助！

猜你喜欢：

[嵌入式大杂烩文章精选](#)

[分享一篇很好的C指针文章，查缺补漏！](#)

[一种省内存的MCU堆栈溢出检测方法](#)

[长文 | C语言从入门到精通保姆级教程（上）](#)

[长文 | C语言从入门到精通保姆级教程（下）](#)



嵌入式大杂烩

本公众号专注于嵌入式技术，包括但不限于C/C++、嵌入式、物联网、Linux等编程学习笔记，同时，公众号内包含大量的...  
255篇原创内容



公众号



Linux大陆

Hello Linux  
3篇原创内容



公众号

在公眾號聊天界面回复**1024**，可獲取嵌入式資源；回复**m**，可查看文章匯總。

喜歡此內容的人還喜歡

166 個最常用的Linux 命令匯總，總有你需要用到的！

軟件測試面試匯總



從0到1，實現自己的操作系統

ZYNQ



常用Linux 軟件匯總！

杰哥的IT之旅

