* 为什么一些指令我在哪个目录下都可以执行啊，而不会告诉我说 找不到可执行文件 呢？这是因为系统环境变量 PATH 的帮助所致呀!
* 先执行下命令，看看你的系统环境变量 $PATH 长得什么样子啊：
  + **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)
    1. echo $PATH

echo $PATH

* + 可以看到这个变量的内容是由一堆目录所组成的
  + 每个目录中间用冒号(:)来隔开, 每个目录是有『顺序』之分的
* 当我们在执行一个指令的时候,就拿『ls』举例来说好了,系统会依照 PATH 的设定去每个 PATH 定义的目录下搜寻文件名为 ls 的可执行文件, 如果在 PATH 定义的目录中含有多个文件名为 ls 的可执行文件,那么先搜寻到的同名指令先被执行!
* 这里有个很好玩的东西，请看：
  + 假设你是 root,如果你将 ls 由/bin/ls 移动成为/root/ls(可用『mv /bin/ls /root』指令达成),然后你自己本身也在/root目录下, 请问：
    1. (1)你能不能直接输入 ls 来执行?
       - 接下来无论你在哪个目录底下输入任何与 ls 相关的指令,都没有办法顺利的执行 ls 了!也就是说,你不能直接输入 ls 来执行,因为/root 这个目录并不在 PATH 指定的目录中,所以,即使你在/root 目录下,也不能够搜寻到 ls 这个指令!
    2. (2)若不能,你该如何执行 ls 这个指令?
       - 因为这个 ls 确实存在于/root 底下,并不是被删除了!所以我们可以透过使用绝对路径或者是相对路径直接指定这个执行文件名, 底下癿两个方法都能够执行 ls 这个指令:
         * **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)

/root/ls

/root/ls

* + - * + **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)

./ls

./ls

* + 1. (3)若要直接输入 ls 即可执行,又该如何迚行?
       - 如果想要让 root 在任何目录下均可执行/root 底下的 ls ,那么就将/root 加入 PATH 当中即可。 加入的方法很简单,就像底下这样:
         * **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)

PATH="$PATH":/root

PATH="$PATH":/root

* + - * + 不过这样只是临时的，当你退出下次再登录的时候 $PATH 的值还是从前的样子
  + 如果确定这个问题进行得没有问题了,请将 ls 搬回/bin 底下,不然系统会挂掉的!
    1. **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)
       - mv /root/ls /bin

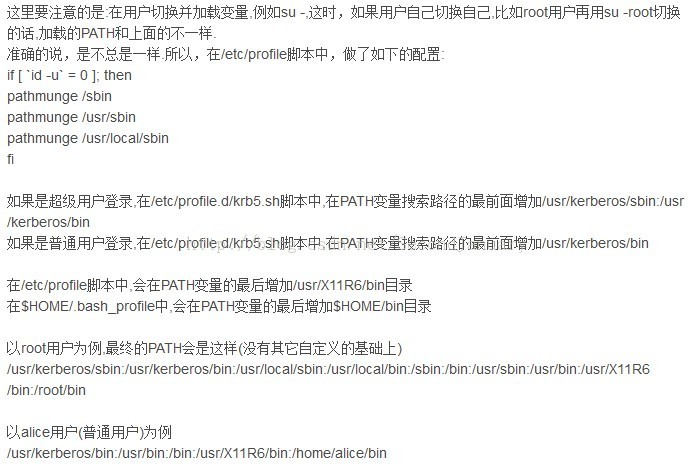
mv /root/ls /bin

* 那你就要问了，怎么不拿当前目录【.】加入系统环境变量 PATH 呢，这样上面的问题(2)，我输入 ls 就可以执行了啦，这个主要处于以下考虑啦：
  + 如果在 PATH 中加入本目录(.)后,确实我们就能够在指令所在目录中进行指令的执行了。 但是由于你的工作目录并非固定(常常会使用 cd 来切换到不同癿目录), 因此能够执行的指令会有变动(因为每个目录底下的可执行文件都不相同嘛!),这对使用者来说幵非好事
  + 另外,如果有个坏心使用者在/tmp 底下做了一个指令,因为/tmp 是大家都能够写入的环境,所以他当然可以这样做。 假设该指令可能会窃取用户的一些数据,如果你使用 root 的身份来执行这个指令,那不是很糟糕? 如果这个指令的名称又是经常会被用到的 ls 时,那『中标』癿机率就更高了!
  + 所以,为了安全起见,不建议将『.』加入 PATH 的搜寻目录中
* 如果是超级用户登录,在没有执行/etc/profile之前,PATH已经设定了下面的路径:
  + **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)
    1. /usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

* 如果是普通用户,PATH在/etc/profile执行之前设定了以下的路径:
  + **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)
    1. /usr/local/bin:/bin:/usr/bin

/usr/local/bin:/bin:/usr/bin

* 下面的东西还无法理解，理解了再来解释，先记着
  + 
  + **定义环境变量的方法：**
    1. 控制台：
       - **[plain]** [view plaincopyprint?](http://blog.csdn.net/SecondJanuary/article/details/8946807)
         1. PATH="$PATH":YOUR\_PATH

$PATH="$PATH":YOUR\_PATH

* + - * 这种方法是临时的，只针对当前shell，换个shell就失效了
    1. 修改文件 /ect/profile
       - 这种方法是永久的，针对所有的用户，所有的shell
    2. 修改文件 ~/.bashrc
       - 这种方法也是永久的，它可以拿环境变量的设置控制在用户级别

**vMware redhat linux中bash:fdisk:command not found**

root权限运行 fdisk -l 会出现

**bash:fdisk:command not found**。  
先查看你当前的命令搜索路径：

[root@localhost sbin]# echo $PATH  
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/home/vincent/bin

以上第二行显示的是命令搜索路径，再看看自己系统中fdisk所在路径。

用whereis fdisk 查询显示路径

[root@localhost home]# whereis fdisk  
fdisk: /sbin/fdisk /usr/share/man/man8/fdisk.8.gz

以上第二行显示的fdisk所在路径 **/sbin/**fdisk......

找到原因，**/sbin**的目录路径（系统用户目录）不在root的＄PATH变量中造成的。  
  
$su  
#cd /home/wangwei(你的用户名)  
#gedit .bash\_porfile  
  
在出现的文本编辑器中，找到PATH=$PATH:$HOME/bin。  
  
在该行后面加上“:/sbin“（不含引号），如下行：  
  
PATH=$PATH:$HOME/bin:/sbin  
  
保存退出。  
  
注销→勾选保存当前设置→重新启动。