# 绝对路径和相对路径

【参考：<https://blog.csdn.net/dengjin20104042056/article/details/95518444】>

## 概述：

绝对路径和相对路径在Shell环境中经常遇到，各有用处。有时候相对路径比较方便，有时候绝对路径比较方便。

## 绝对路径：

1. 绝对路径必须由“/”开头
2. 绝对路径是为档案 / 文件的所在位置做指向
3. 在任何时候，都可以用绝对路径来找到我们想要的文件

【举例说明】

|  |
| --- |
| 示例一：  /home/deng  示例二：  /usr/share/man |

## 相对路径：

1. 相对路径不由“/”开头
2. 相对路径是相对于现在目前所在的位置，为目的地指向
3. 通常相对路径比绝对路径短一点，可以当成我们迅速找到文件/档案的捷径

|  |
| --- |
| 示例一：  ../test/itcast  示例二：  ./test |

## 思维发散：

问题：在给出相对路径时，系统如何识别其对应的绝对路径呢？

事实上，相对路径时根据PATH环境变量来查找绝对路径的。

举例来说，我目前在 /home/itcast 这个itcast 使用者家目录中，我想要看一看里面的 .bashrc 这个档案的数据，使用的是 more 命令，而这个指令在 /bin/more 当中。通常的命令写法是：

|  |
| --- |
| [itcast@localhost ~]$ more .bashrc |

而由于PATH变量的值为：

|  |
| --- |
| [deng@localhost ~]$ echo $PATH  /usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/deng/.local/bin:/home/deng/bin  [deng@localhost ~]$ |

在指定 more 命令时，shell会根据PATH变量的路径中顺序查找，一旦查找匹配到（此例匹配到的应该是 /bin/more ），便执行 /bin/more .bashrc 。

这便是执行命令时由相对路径查找绝对路径的过程。

所以我们就可以理解为什么执行当前目录下的可执行程序应该用的格式是：

|  |
| --- |
| [deng@localhost ~]$ ./test.sh |

而不是直接用 test.sh 了。因为通常PATH变量不包含当前路径。如果不加 ./ 会导致相对路径格式的命令找不到绝对路径。