各种特殊符号

1. 通配符

（1）

\*——匹配任意个任意的字符（0，1，无限个）

（2）

？——匹配任意的一个字符

（2）

[abcd]——匹配a,b,c,d中任意的一个字符

（3）

[!abcd]或[^abcd]——匹配非abcd的任意一个字符

整个[!abcd]只能表示一个字符，不能表示多个

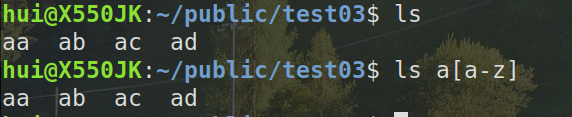


原本文件夹中有文件a01，a02。ls a[!2222]匹配ax（x不是222且x是单个字符），但是当前文件夹中的文件a后都接了两个数字，故会显示“没有那个文件或目录”。

ls a[!1]2 匹配ax2（其中x部位1且x为单个字符），自然匹配到了a02。

（4）

[a-z],[0-9]——匹配a到z中任意多个字符 可能等同于{a..z}?



1. 重定向符号
2. 文件有三种状态：0-标准输入（从键盘输入），

1-标准输出（输出需要的信息到屏幕上），

2-标准错误输出（输出错误信息到屏幕上）

（2）输入重定向 <

数据不从键盘输入，改为从指定的文件输入

（3）输出重定向 >

数据（文件中的数据，命令执行结果或报错）不输出到屏幕，而是输出到文件。

1> （>） 将一般的数据重定向输出到文件

2> 将错误信息重定向输出至文件

&> 将一般信息，错误信息都重定向输出至文件

>和>>的区别：前者直接将内容覆盖原文件，后者在末尾追加

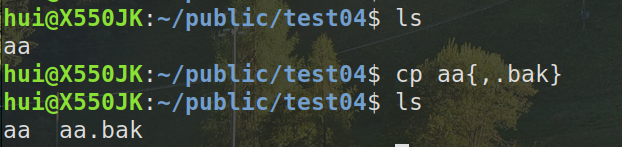
例子：现在有两个新创建的文件test1，和test2，需求:将test1中的内容重定向到test2中，怎么操作呢?

     cat <test1> test2

三、其他符号

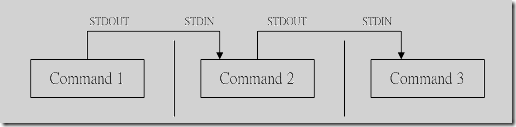
（1）{}：可进行快速备份，cp filename{,.bak}





（2）管道符（|）：

命令格式：命令A|命令B，即命令1的正确输出作为命令B的操作对象



如：ls |grep cad——在ls的输出结果中找到与cad相关的项。

（3）&：使当前命令在后台运行，一般用来打开应用

（4）&&：表示并且，按顺序依次执行命令，

只有当前面的命令执行成功才会执行后面的命令

（5）； 分割多个命令，没有因果关系，只是一步步执行

（6）\ 屏蔽别名，不转义，也即原样输出

(7):`command`:执行反引号中的命令 等同于$(command)

(8):echo $?:返回上个命令的执行状态

shell编程

（6）单引号，双引号，不加引号？

1.单引号 (' ') :所见即所得，不具有变量置换的功能，即单引号会告诉shell忽略所有的特殊字符。

IMG_256

2.双引号(" ") 双引号，具有变量置换的功能，即双引号之要求忽略大多数特殊字符，除了$（使用变量前导符）, \(转义符), `(反向引号)。

IMG_257

3.反向引号(` `): 位于键盘Tab键上面一行的键，用作命令替换（相当于$(...)）

IMG_258

注：命令替换是指shell能够将一个命令的标准输出插在一个命令行中任何位置。

（7） ^r：以r开头

^(rrr)：以rrr开头

r$：以r结尾

(rrr)$：以rrr结尾

注：多余一个字符便视为一个组，要加括号。