

Unix中利用转义和引用来管理元字符Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/607/2021_2022_Unix_E4_B8_AD_E5_88_A9_c103_607069.htm 在Unix操作系统中有一类特殊的字符，叫做元字符。他们在系统中表示特殊的含义。如*号与?号，操作系统认为这些元字符是通配符。如果在路经、文件名或者命令参数中有这些元字符的话，就会引起操作系统的误解。为此在系统中要采用一些方法，能够让操作系统将这些元字符当作普通字符来对待。在Unix系统中，就是采用转义和引用这两种方式来处理。系统工程师需要明白的是，这两种解决方案的差异。并在特定的情况下，采用合适的解决方式。

一、转义功能的应用。其实转义机制不仅仅在Unix操作系统上有，在其他的编程语言中也有类似的处理机制。如果各位读者有程序开发经验的话，那么可能更加容易了解这个转义功能。简单的说，转义功能就是指在一些具有特殊含义的元字符前面加上一个转义字符\，目的是告诉操作系统这是一个普通的字符，取消元字符的特殊意义。如*本来代表通配符，如ls *则表示所有的文件与目录。而*就是将*号当作一个普通的字符来对待，系统不会再认为其是一个通配符。这个转义功能的普通应用是比较简单的，只需要在元字符前面加上\即可。不过笔者这里还需要强调转义字符的几个特殊的应用。这些应用虽然比较特殊，普通用户平时不怎么用的到。但是对于系统工程师来说，这些功能可能就特别的有用。

一是需要在文件名中插入空格。如在微软操作系统中有一个My Documents文件夹，其中间就有一个空格。这个空格在Unix系统中也属于一个特殊的原字符。如果在建立文件或

者目录时，直接加入空格，则Unix操作系统会提示错误信息。此时往往需要转移字符的帮助。如使用命令`mkdir My \ Documents`，就可以建立中间带有空格的目录名。如要建立带有空格的文件名，也是如此。不过需要注意的是，如果在文件或者目录中包含了元字符的话，那么在使用删除命令或者目录定位命令的时候，也需要采用转移字符来定义这些特殊的元字符，否则的话可能会出现一些莫名其妙的问题。二是利用转义字符对命令进行换行。在Unix操作系统中有些命令特别复杂，特别是那些部属在Unix操作系统上的命令。如`expdp`是Oracle数据库中的一个数据库对象导出命令。虽然这个命令的功能很强大，但是可惜的是只有通过复杂的参数才能够完成一些特定的功能。有时候一个命令需要好几行才能够写完。虽然Unix操作系统提供了命令自动换行的方法。但是这个自动换行的功能比较薄弱，如不会在系统工程想要换行地方进行换行。为此采用这个自动换行的话，会造成代码阅读的困难。所以不少系统工程师还是希望能够自己手工进行分行，在某些关键参数后面进行分行，以增加命令的可阅读性。但是，如果直接使用回车键分行的话，系统是不认可的。因为这个回车键系统认为是一个执行命令的键。即当系统工程师点击回车键的时候，系统会认为用户已经将命令输入完毕。键入回车键后系统就会自动运行这个命令。所以直接键入回车键往往不能够完成命令分行的需求。此时就需要转义字符的帮助。如现有一个命令比较长，工程师希望能够将其分为两行，主要是将参数`-name`后面的内容另起一行。此时就可以在这个参数前面加入一个转义字符`\`，然后再按回车键即可。由于有了这个转义字符，系统就会取消回车键的换行作用

。如此操作之后，系统会出现一个次级提示符，这个次级提示符表示命令还没有结束，下一行继续。这个功能对于系统工程师来说，可能是非常实用的。因为此时系统工程师可以轻松的将一长串的命令根据自己的需要进行分行，从而提高命令的可读性。另外需要注意的是，\这个转义字符本身就是一个比较特殊的元字符。如果用户想在命令或者文件名中使用这个\符号的话，也需要采用转移字符。如用户想利用echo或者printf命令显示网址。在网址中就有很多\符号，此时就需要采用转义字符来让系统将\符号当作普通字符来对待。

二、利用引用来解决元字符的问题。除了可以利用上面谈到的引用功能来处理这些元字符以外，还可以采用引用这个功能来解决。简单的说，就是将一个命令参数放置在一对引号之内，则这个引号之内内容如果有元字符的话，则这些元字符将不起作用。那么既然转义功能已经可以解决元字符的问题，系统还提出了一个引用的解决方案，是否多次一举呢？其实不然。当一个命令行中含有多个元字符的时候，需要在每个元字符前面加一个转移字符。为此此时如果采用转义字符来解决这个元字符的问题时，会非常的繁琐。此时如果采用引用机制来解决元字符的问题时，可能会更加的理想。如现在系统工程师希望能够在屏幕上打出如下信息(这是一个共享文件的路径)：192.128.11.3\share\IT\software\pdf。如果采用转移字符的话，该如何写呢？由于这个输出结果中有四个元字符(转义字符\)，为此管理员不得不用四个转移字符来处理。即要写成echo 192.128.11.3\\share\\IT\\software\\pdf。这显然非常的麻烦。遇到这种情况时，如果采用引用的话，显然会更加的合适。如采用引用机制，只需要将这个命令写为即可：

echo ' 192.128.11.3\share\IT\software\pdf ' 将一大串命令采用单引号因起来。则系统会将命令行中的一些元字符都当作普通字符来对待。即不需要为每一个元字符采用转移字符。显然这个引用解决方案要比采用转义字符来解决要方便的多。在采用引用机制来处理元字符时，需要注意跟双引号的区别。如现在有如下三个命令echo \$JAVA_HOME、echo ' \$JAVA_HOME '、 “ echo \$JAVA_HOME ”。其中\$JAVA_HOME表示应用程序Java的环境变量。此时如果系统工程师依次运行上面这三个命令，会有什么结果呢?第一个命令会正常显示Java的环境变量.第二个命令直接显示\$JAVA_HOME，即已经将元字符\$当作普通字符来对待。第三个命令的话仍然显示Java的环境变量。可见双引号与单引号在引用机制上，还是有所不同的。那么他们到底有什么差异呢?总的来说，系统工程师需要注意如下内容。单引号会保护里面所有的元字符，即遇到元字符时会告诉系统当作普通字符来对待。但是如果采用了双引号，则系统会将单引号之间的内容解释为命令行。如会将\$当作环境变量的前缀等等。其实单引号与双引号还有一种相互保护的作用。即双引号保护其中的单引号，而单引号保护其中的双引号。因为无论是双引号还是单引号本身就都是元字符，可以通过引用机制来进行保护。不过在遇到单引号与双引号(都当作普通字符)都存在的情况下，笔者建议还是采用转移字符来保护单引号或者双引号为好。这可以避免误解，也容易提高代码的可读性。同理，如果要将转移字符\当作普通字符来实用的话，那么最好也是利用单引号来保护这个转移字符，而不是利用转移字符来保护转移字符。这些虽然不影响实际的应用，但是却

是提高代码可读性的有效手段。从以上的分析中可以看出，虽然转移字符与引用机制等等都可以将元字符当作普通字符来对待。但是，两者在实现的方法上还是有一定的差异。为此最好能够根据应用场合的不同，在选择合适的解决方案。一般来说，两者的差异只是实现方法上的差异，在具体的功能上没有区别。不过为了代码的易读性上考虑，选择具体的解决方案是还需要慎重。不过总的来说，系统工程师必须同时掌握这两种方法。然后根据不同的情况现在合适的解决方法。如果光掌握一种解决措施，可能不能够解决所有跟元字符相关的问题。更多优质资料尽在百考试题论坛 百考试题在线题库 linux认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com