训练稿——Linux常用重点命令 tr&join&cut&paste&split

目录

[命令：tr 1](#_Toc323572749)

[命令：join 3](#_Toc323572750)

[命令：cut 4](#_Toc323572751)

[命令：paste 4](#_Toc323572752)

[命令：split 5](#_Toc323572753)

## 命令：tr

|  |
| --- |
| tr |
| 用来从标准输入中通过替换/删除进行字符转换【不接受指定的文件参数，而只是对标准输入进行翻译】  主要用于删除文件中的控制字符或进行字符转换  使用时，提供两个字符串，串1：用于查询，串2：用于处理各种转换；  串1的字符被映射到串2上，然后转换开始 |

|  |
| --- |
| 主要用途： |
| 1.大小写转换  2.去除控制字符  3.删除字符 |

|  |
| --- |
| 格式： |
| tr –c –d –s [“str\_from”] [“str\_to”] file  tr –选项 源 目标 输入文件 【即读取文件，将 源->目标】 |

|  |
| --- |
| 选项 |
| -c，用字符串1中字符集的补集替换此字符集，要求字符集为ASCII  -d，删除字符串中所有指定字符串  tr –d “:” < testfile  -s，删除所有str\_from重复出现字符序列，只保留一个，即重复字符串压缩为一个  【一个很经典的应用就是把不规律的空格缩减为一个空格】  字符范围——tr，可以指定字符串列表或范围作为形成字符串的模式，似正则，但不是正则。  [a-z] [A-Z] [0-9] /octal一个三位八进制数，对应有效ＡＳＣＩＩ字符  [s\*n]字符s出现n次 |

|  |  |  |
| --- | --- | --- |
| tr 中特定字符的不同表达方式 | | |
| \a | Ctrl-g铃声 | \007 |
| \b | Ctrl-h退格 | \010 |
| \f | Ctrl-l走纸模式 | \014 |
| \n | Ctrl-J新行 | \012 |
| \r | Ctrl-M回车 | \015 |
| \t | Ctrl-I tab键 | \011 |
| \v | Ctrl-x | \030 |

|  |  |
| --- | --- |
| tr中表示集合的符号 | |
| [:alnum:] | 所有字母字符与数字 |
| [:alpha:] | 所有字母字符 |
| [:blank:] | 所有水平空格 |
| [:cntrl:] | 所有控制字符 |
| [:digit:] | 所有数字 |
| [:graph:] | 所有可打印的字符(不包含空格符) |
| [:lower:] | 所有小写字母 |
| [:print:] | 所有可打印的字符(包含空格符) |
| [:punct:] | 所有标点字符 |
| [:space:] | 所有水平与垂直空格符 |
| [:upper:] | 所有大写字母 |
| [:xdigit:] | 所有16 进位制的数字 |

|  |
| --- |
| 常用示例： |
| 1. 去除所有重复字符【只保留一个】   $tr –s “[a-z]” < oops.txt   1. 去除空行   $tr –s “[\012]” <oops.txt  $tr –s “[\n]” <oops.txt  $tr –s “\n” <oops.txt   1. 小写转大写   $echo “AbcdefG” | tr “[a-z]” “[A-Z]”  $echo “AbcdefG” | tr “[:lower:]” “[:upper:]”   1. 删除指定字符串   $tr –cs “[a-z][A-Z]” “[\012\*]” < data.txt  将非字母字符转为新行 -s压缩重复的字符   1. 转换控制字符   $tr –s “[\136]” “[\011\*]” < start.txt   1. 快速转换   $tr –s “[\r]” “[\n]” < input.txt   1. 匹配多于一个字符   $tr “[0\*4]” “\*”< input.txt  8.  tr SET1 SET2  **即SET1中的字符都会被替换为SET2中相应位置上的字符。**  tr里面包含SET1和SET2，那如果出现两个集合的大小不同的情况，tr如何处理呢？  如果SET1>SET2, SET1中多出来的字符都会和SET2中最后一个字符相对应。  如果SET1<SET2，SET2中多余的部分将被抛弃\ |

## 命令：join

|  |
| --- |
| join【这个需要找习题练下】 |
| 将来自两个分类文本文件的行连在一起  前提 ：file1,file2已分类  每个文件都有一些元素与另一文件相关——连  有点像求并集  注：joio时文本域要少于20= |

|  |
| --- |
| 格式： |
| join [options] input-file input-file2 |

|  |
| --- |
| 选项： |
| an n为数字，用于连接时从文件n中显示不匹配行,-a1表示第一个文件的不匹配行  o n,m n为文件号，m为域号， 1,3 表示只显示文件1的第三域  j n m n为文件号，m为域 使用其他域做连接域  t 域分隔符 用来设置非空格/非tab分隔符 |

|  |
| --- |
| 常用示例: |
| 1. 连接两个文件[默认连接域为域0]   $Join name.txt. town.txt   1. 显示第一个文件的不匹配行   $join –a1 name.txt town.txt   1. 设置显示连接结果   $join –o 1.1,2.2 name.txt town.txt  第一个文件第一个域和第二个文件第二个域作为显示结果   1. 设置连接域   $join –j 1 3 –j 2 2 file1 file2 |

## 命令：cut

|  |
| --- |
| Cut: |
| 用来从标准输入或文本文件中剪切列或域【提取文本列】  cut命令可以从一个文本文件或者文本流中提取文本列  可将剪切到的文本贴到另一个文件中 |

|  |
| --- |
| 格式: |
| cut [options] file1 file2 |

|  |
| --- |
| 选项： |
| -b 字节  -c list 指定剪切字符数  -d 指定与空格/tab不同的分隔符  -c 指定剪切的字符范围，字符，如-c 1,5,7 第1,5,6,7字符 -c 1-50 前50个字符  -f 指定剪切的域范围 -f 1,5 剪切1,5两个域 -f 1,10-12 剪切1,10,11,12四个域  【一般组合为 –d ‘分隔符’ –f 域NO】  【另一组合-c】 |

|  |
| --- |
| 常用示例： |
| -使用域分隔符  $cut –d : -f3 data   1. 剪切指定域   $cut –d : -f1,3 data   1. 剪切字符   $who –u | cut –c 1-8 |

## 命令：paste

|  |
| --- |
| paste |
| 该命令主要用来将多个文件的内容合并，与cut命令完成的功能刚好相反，Paste将数据贴到相关文件中  存在两个不同来源 数据应先分类，确保文件行数同  功能说明：合并文件的列  补充说明：paste指令会把每个文件以列对列的方式，一列列地加以合并 |

|  |
| --- |
| 格式: |
| paste [-s][-d <间隔字符>][--help][--version][文件...] |

|  |
| --- |
| 选项： |
| -d 指定不同分隔符，默认分隔符为tab  -s 或--serial将每个文件合并成行而不是按行黏贴，文件A一行，文件B一行  --help 　在线帮助。 --version 　显示帮助信息。 |

|  |
| --- |
| 常用示例 |
| File1  1  2  File2  A  B   1. 合并之 【默认分隔符是tab】   $paste file1 file2   1. A 2. B 3. 指定分隔符   $paste –d: file2 file1  A:1  B:2   1. 合并两行，而非按行黏贴   $paste –s file1 file2  1 2  A B |

## 命令：split

|  |
| --- |
| split |
| 用来将文件切分为小文件  split（按大小分割）和csplit（按匹配分割） |

|  |
| --- |
| 格式： |
| split –output\_file\_size input\_filename output\_filename  其中out\_file\_size为被分割行数，默认1000 |