字符串截取及切割

定义一个变量phone，并确认其字符串长度：

1. **[**root@svr5 **~]**# phone**=**"13788768897"
2. **[**root@svr5 **~]**# echo $**{**#phone**}**
3. **11**

echo　${变量:开始:长度}

${}截取时，起始位置可以省略，省略时从第一个字符开始截，例：

1. **[**root@svr5 **~]**# echo $**{**phone**:0:6}**
2. **137887**

echo ${变量/旧/新}　替换字符

1. **[**root@svr5 **~]**# echo $**{**phone**/8/**X**}**
2. 137X8768897

替换全部子串

格式：${变量名//old/new}

1. **[**root@svr5 **~]**# echo $**{**phone//8/X}
2. 137XX76XX97

以处理系统默认的账户信息为例，定义变量A：

1. **[**root@svr5 **~]**# A**=**`head **-1** **/**etc**/**passwd`
2. **[**root@svr5 **~]**# echo $A
3. root**:**x**:0:0:**root**:**/root:/bin**/**bash

echo ${变量#\*:｝

掐头，从左开始匹配到最近的：全部删除（包括冒号）

1. **[**root@svr5 **~]**# echo $**{**A#**\*:}**
2. x**:0:0:**root**:**/root:/bin**/**bash

echo ${变量##\*:}

掐头，从左开始匹配到最远的：全部删除（包括冒号）

1. **[**root@svr5 **~]**# echo $**{**A##**\*:}**
2. /bin/bash

echo ${变量%:\*}

去尾，从右开始匹配到最近的：全部删除（包括冒号）

1. **[**root@svr5 **~]**# echo $**{**A**%:\*}**
2. root**:**x**:0:0:**root**:/**root

echo ${变量%%:\*}

去尾，从右开始匹配到最远的：全部删除（包括冒号）

1. **[**root@svr5 **~]**# echo $**{**A**%%:\*}**
2. root

echo ${变量：-word}

正则　基本

^　以．．开始

$　以．．结尾

［］　　　集合

［＾］　　对集合取反

．　任意单个字符

＊　匹配一个字符出现的任意次数包括０次

\{n,m\}　匹配前一个字符出现了n 次以上

\{n,\} 　匹配前一个字符出现了ｎ次以上

＼{n\} 　匹配前一个字符出现了ｎ次以上

扩展正则

+　　至少１次　　ａ＋

？　０次或１次　　ａ？

{n,m}

() 分组，保留

｜ 或者

＼b边界

例如：egrep “\bthe\b” a.txt

＂the＂前面加＼b表示它前面不能有别地字符 （以the开头）

＂the＂后面加＼b表示它后面不能有别地字符（以the结尾）

sed 基本用法

1. 用法**1**：前置命令 **|** sed **[**选项**]** '条件指令'
2. 用法**2**：sed **[**选项**]** '条件指令' 文件**..** **..**

sed命令的常用选项如下：

-n（屏蔽默认输出，默认sed会输出读取文档的全部内容）

-r（让sed支持扩展正则）

-i（sed直接修改源文件，默认sed只是通过内存临时修改文件，源文件无影响）

条件：

行号 4 4,5 4~2 4,+1　　　/正则/

指令：

p 打印　　　d 删除　　　　　s 替换s/旧/新/g

a 追加　　　i 插入　　　　　　c 替换行

1. sed命令的 -n 选项

行号可以是连续的行号，如打印passwd第3到第6行账户的信息：

1. **[**root@svr5 **~]**# sed **-**n '3,6p' /etc/passwd

sed命令的 -i 选项

正常情况下，sed命令所做的处理只是把操作结果（包括打印、删除等）输出到当前终端屏幕，而并不会对原始文件做任何更改：

1. **[**root@svr5 **~]**# sed 'd' /etc/passwd                     //删除所有行
2. **[**root@svr5 **~]**# cat **/**etc**/**passwd                         //查看原始文本，并未改动
3. **[**root@svr5 **~]**# sed **-**i '1,4d' test**.**txt         //删除操作
4. **[**root@svr5 **~]**# cat test**.**txt                     //确认删除结果

多个指令可以使用分号隔离用分号来隔离多个操作，比如：

1. **[**root@svr5 **~]**# sed **-**n '1p;4p' /etc/passwd
2. root**:**x**:0:0:**root**:**/root:/bin**/**bash
3. adm**:**x**:3:4:**adm**:**/var/adm**:**/sbin/nologin

1）行号案例

打印第3行：

1. **[**root@svr5 **~]**# sed **-**n '3p' /etc/passwd

打印第3到5行：

1. **[**root@svr5 **~]**# sed **-**n '3,5p' /etc/passwd

打印第3和5行：

1. **[**root@svr5 **~]**# sed **-**n '3p;5p' /etc/passwd

打印第3以及后面的10行：

1. **[**root@svr5 **~]**# sed **-**n '3,+10p' /etc/passwd

打印奇数行：

1. **[**root@svr5 **~]**# sed **-**n '1~2p' /etc/passwd

打印偶数行：

1. **[**root@svr5 **~]**# sed **-**n '2~2p' /etc/passwd

2）正则案例

打印包含root的行：

1. **[**root@svr5 **~]**# sed **-**n '/root/p' /etc/passwd

打印bash结尾的行：

1. **[**root@svr5 **~]**# sed **-**n '/bash$/p' /etc/passwd

打印bin开头的行：

1. **[**root@svr5 **~]**# sed **-**n '/^bin/p' a**.**txt        //输出以bin开头的行

3）没有条件，则表示匹配所有行

1. **[**root@svr5 **~]**# sed **-**n 'p' /etc/passwd
2. **[**root@svr5 **~]**# sed **-**n '$=' a**.**txt            //输出文件的行数

下面看看sed工具的d指令案例集锦（自己提前生成一个a.txt文件）

1. **[**root@svr5 **~]**# sed '3,5d' a**.**txt             //删除第3~5行
2. **[**root@svr5 **~]**# sed '/xml/d' a**.**txt            //删除所有包含xml的行
3. **[**root@svr5 **~]**# sed '/xml/!d' a**.**txt         //删除不包含xml的行，!符号表示取反
4. **[**root@svr5 **~]**# sed '/^install/d' a**.**txt    //删除以install开头的行
5. **[**root@svr5 **~]**# sed '$d' a**.**txt                //删除文件的最后一行
6. **[**root@svr5 **~]**# sed '/^$/d' a**.**txt             //删除所有空行

下面看看sed工具的s指令案例集锦（自己提前生成一个a.txt文件）

注意：替换操作的分隔“/”可改用其他字符，如#、&等，便于修改文件路径

1. **[**root@svr5 **~]**# sed 's/xml/XML/' a**.**txt        //将每行中第一个xml替换为XML
2. **[**root@svr5 **~]**# sed 's/xml/XML/3' a**.**txt     //将每行中的第3个xml替换为XML
3. **[**root@svr5 **~]**# sed 's/xml/XML/g' a**.**txt     //将所有的xml都替换为XML
4. **[**root@svr5 **~]**# sed 's/xml//g' a**.**txt     //将所有的xml都删除（替换为空串）
5. **[**root@svr5 **~]**# sed 's#/bin/bash#/sbin/sh#' a**.**txt //将/bin/bash替换为/sbin/sh
6. **[**root@svr5 **~]**# sed '4,7s/^/#/' a**.**txt         //将第4~7行注释掉（行首加#号）
7. **[**root@svr5 **~]**# sed 's/^#an/an/' a**.**txt         //解除以#an开头的行的注释（去除行首的#号）
8. **[**root@svr5 **~]**# sed **-**n 's/2017/xxxx/p' ａ**.**txt　　//显示替换的行

删除文件中每行的第二个、最后一个字符

分两次替换操作，第一次替换掉第2个字符，第二次替换掉最后一个字符：

1. **[**root@svr5 **~]**# sed 's/.//2 ; s/.$//' nssw**.**txt

将文件中每行的第一个、倒数第1个字符互换

每行文本拆分为“第1个字符”、“中间的所有字符”、“倒数第1个字符”三个部分，然后通过替换操作重排顺序为“3-2-1”：

1. **[**root@svr5 **~]**# sed **-**r 's/^(.)(.\*)(.)$/**\3\2\1**/' nssw**.**txt

删除文件中所有的数字

因原文件内没有数字，行首也没有空格，这里稍作做一点处理，生成一个新测试文件：

1. **[**root@svr5 **~]**# sed 's/[0-9]//' nssw**.**txt

以nssw2.txt文件为例，删除所有数字、行首空格的操作如下：

1. **[**root@svr5 **~]**# sed **-**r 's/[0-9]//g;s/^( )+//' nssw2**.**txt

为文件中每个大写字母添加括号

使用“（）”可实现保留功能，所以可参考下列操作解决：

1. **[**root@svr5 **~]**# sed **-**r 's/([A-Z])/[**\1**]/g' nssw**.**txt

基本语法格式案例：

1. **[**root@svr5 **~]**# sed '2a XX' a**.**txt            //在第二行后面，追加XX
2. **[**root@svr5 **~]**# sed '2i XX' a**.**txt            //在第二行前面，插入XX
3. **[**root@svr5 **~]**# sed '2c XX' a**.**txt            //将第二行替换为XX

　　　　　　　　awk 提取文本

基本操作方法

　格式：awk [选项] '[条件]{指令}' 文件其中，print 是最常用的编辑指令；若有多条编辑指令，可用分号分隔。

　Awk过滤数据时支持仅打印某一列，如第2列、第5列等。

　处理文本时，若未指定分隔符，则默认将空格、制表符(Tab)等作为分隔符。

选项 -F 可指定分隔符

输出passwd文件中以分号分隔的第1、7个字段，显示的不同字段之间以逗号隔开，操作如下：

1. **[**root@svr5 **~]**# awk **-**F**:** '{print $1,$7}' /etc/passwd

$1是指第１列　$7是指第７列

awk还识别多种单个的字符，比如以“:”或“/”分隔，输出第1、10个字段：

1. **[**root@svr5 **~]**# awk **-**F **[:**/] '{print $1,$10}' /etc**/**passwd

awk常用内置变量：

$0 文本当前行的全部内容

$1 文本的第1列

$2 文件的第2列

$3 文件的第3列，依此类推

NR 文件当前行的行号

NF 文件当前行的列数（有几列）

输出每次处理行的行号，以及当前行以“:”分隔的字段个数（有几列）：

1. **[**root@svr5 **~]**# awk **-**F**:** '{print NR,NF}' passwd**.**txt

awk的print指令不仅可以打印变量，还可以打印常量

1. **[**root@svr5 **~]**# awk **-**F**:** '{print $1,"的解释器:",$7}' /etc/passwd

通过df命令,直接awk中使用正则，查看以根结尾所在行的第四列内容．　　”\”表示取消＂/＂的特殊含义

1. **[**root@svr5 **~]**# df **-**h **|** awk '/**\ /**$/{print $4}'