# linux基础

grep（管道） sed(流编辑) awk(报告文本生成器)

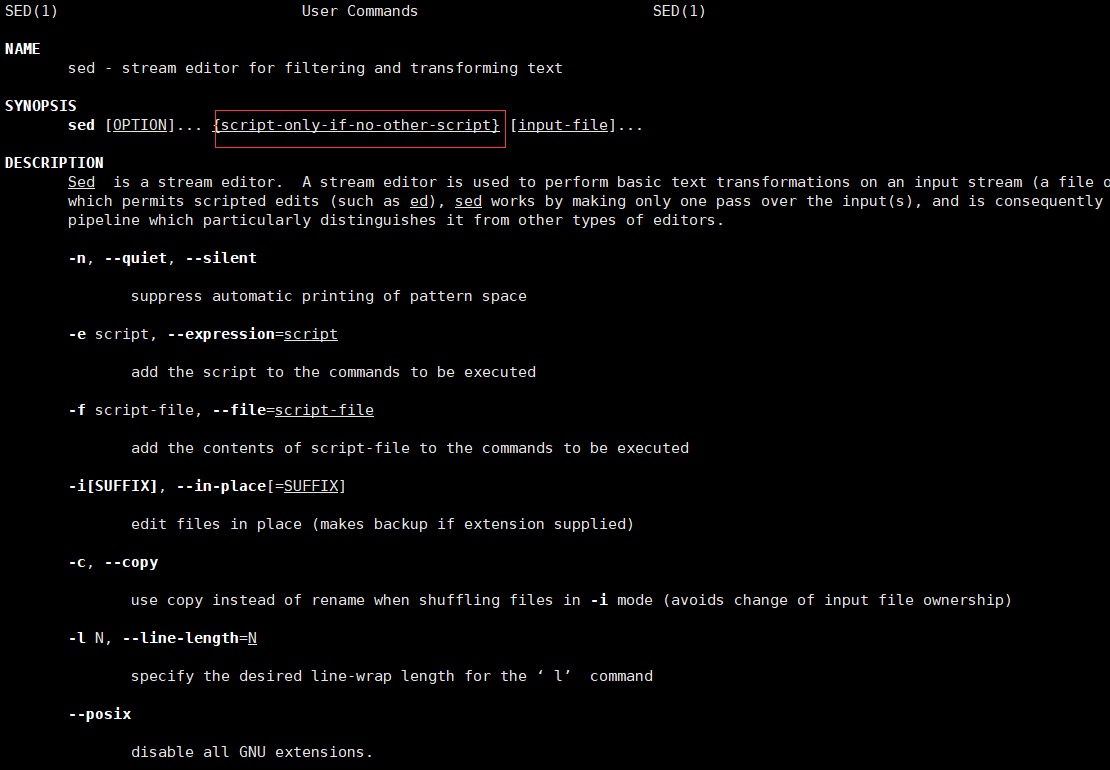
## sed

**sed的基本用法**

sed : stream editor

sed是一个行编辑器，当然还有（全屏编辑器：vi）

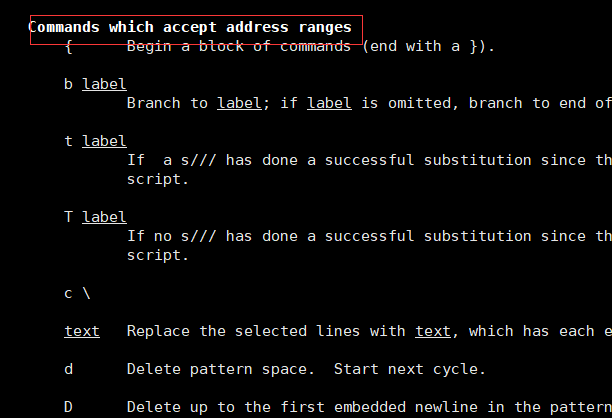
sed并不直接处理文本文件本身，而是把文件读取到内存，在内存完成编辑，显示到屏幕上来，这段内存空间称为**模式空间**。在这段内存中完成**模式匹配**。默认不编辑源文件，仅对内存中的副本处理。



### sed模式空间

默认不编辑源文件，仅对模式空间中的内容做处理，而后，处理结束后，将模式空间中的内容显示到屏幕中来。

指定地址范围



**sed [opiton] addresscommond file…**

**-n 静默模式，不在默认显示模式空间中的内容。**

**-i 直接修改源文件，危险**

**-e script -e script 可以同时执行多个脚本**

**-f 指定读取文件 这个文件中可以有多个脚本。**

**-r 表示使用扩展正则表达式**

Address:

1. startlines,endlines 1,100

2. /Regexp/ /^root/ 正则表达式指定模式 找root开头的字符串

3. /pattern1/ ,/pattern2/ 第一次被模式1匹配到的行开始至第一次模式2匹配到的行结束

4. lineNumber 指定行 $表示最后一行，$-1 倒数第二行

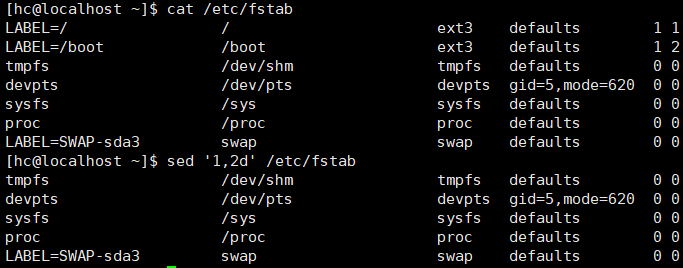
5. startline,+N 从startline开始，向后的N行

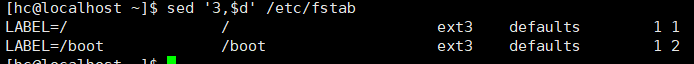
commond:

### d

删除符合条件的行

sed ‘1,2d’ /etc/fstab 删除前两行 地址加命令



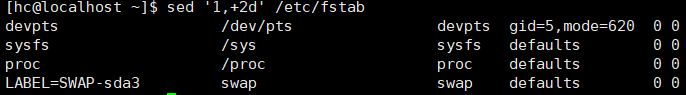


sed ‘**/oot/**d’ /etc/fstab

模式匹配这些行中有oot字符的就删除



sed ‘1,+2d’ /etc/fstab 从第一行开始删除3行



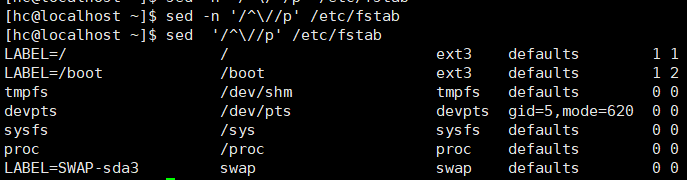
### P

P(大写)显示符合条件的行

-n 只显示符合条件的行

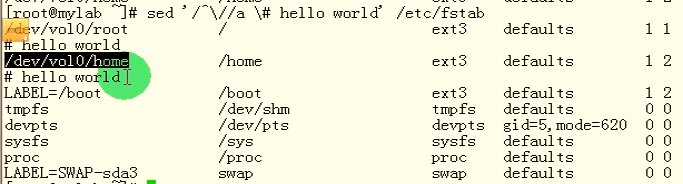
以斜线开头的行

sed ‘/^\//p’ /etc/fstab **\/** 对斜线转义

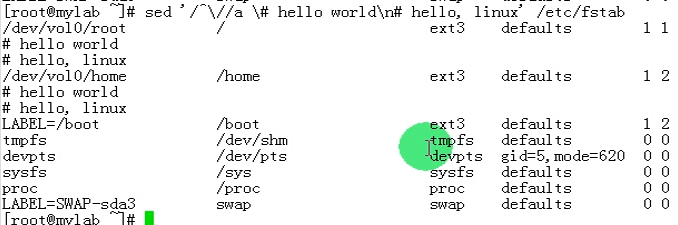


### a

a \string :在指定的行后面追加新行，内容为string



追加两行

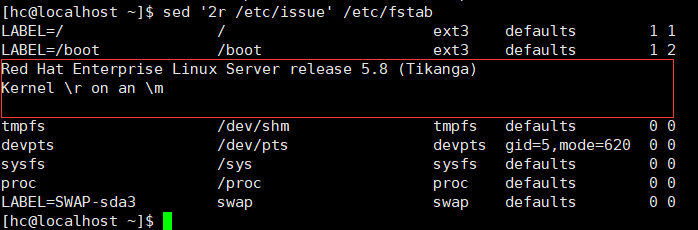


### i

i \string 在指定的行的前面添加新行，内容为string

### r

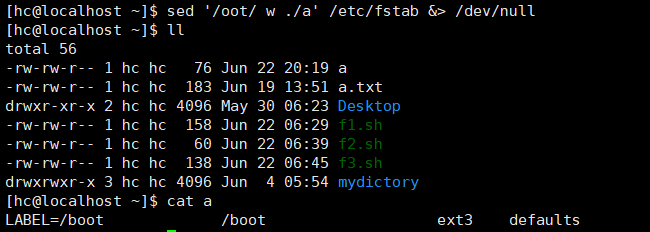
r file 将指定的文件的内容添加至符合条件的行的后面



### w

w file 将指定范围的内容另存至指定的文件中去。

sed ‘/oot/ w ./a’ /etc/fstab &> /dev/null



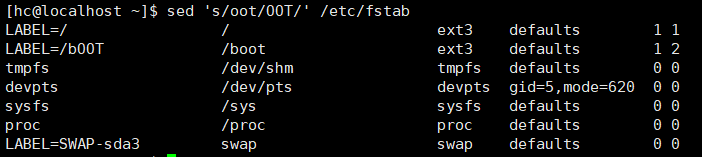


### s

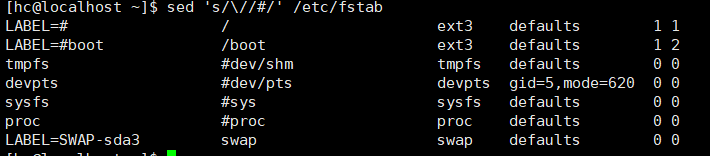
查找并替换

s \pattern\ string\ 表示将pattern匹配到的全部替换为string

sed 's/oot/OOT/' /etc/fstab oot 替换成 OOT



sed 's/\//#/' /etc/fstab 将/开头的替换为#



加修饰符

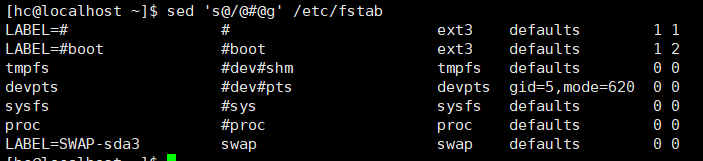
**g** : 全局替换

**i**： 忽略字符的大小写

当然也可以使用其他符号作为分隔符。但是匹配字符串是改符号时，要避开使用。

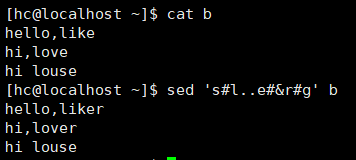
**s ###** **s@@@**  替换的字符串不能是分隔符才能使用

sed ‘s@/@#@g’ /etc/fstab 将/全部替换成#号



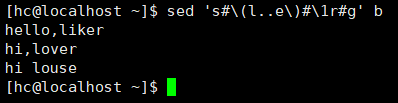
**&** 表示引用模式匹配的整个串

sed 's#l..e#&r#g' b 全文替换，有l开头并且e结尾的都在其尾部追加r



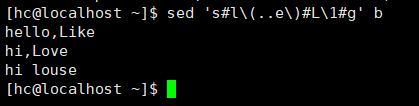
**\( \) \1 \2 后项引用**

sed 's#\(l..e\)#\1r#g' b



有些时候只能使用后项引用，比如部分替换

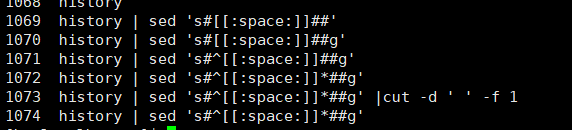
sed 's#l\(..e\)#L\1#g' b



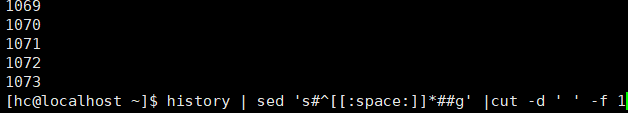
history | sed 's#[[:space:]]##g' 替换掉history所有空白字符

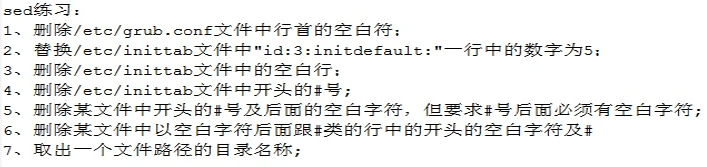


history | sed 's#^[[:space:]]\*##g' 替换掉history行首的空白字符



history | sed 's#^[[:space:]]\*##g' | cut -d ' ' -f 1 只要行数





1 sed 's#^[[:space:]]\*##g' /etc/grub.conf

2 sed 's#^\(id:3:initdefault:\)#id:5:initdefault: #g' /etc/inittab

sed 's#\(id:\)[0-9]\(:initdefault:\)#\15\2#g' /etc/inittab

3 sed '/^$/d' /etc/inittab

4 sed 's@^#@@' /etc/inittab

5 sed 's@^\(#[[:space:]]\)@@g' /etc/inittab

6 sed 's@^[[:space:]]+@@g' /etc/inittab

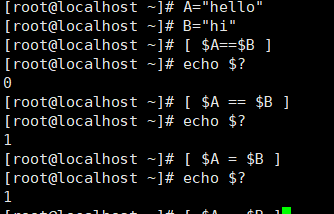
7 echo "/etc/rc.d" | sed -r 's@^(/.\*/)[^/]+/?@\1@g'

## 字符串测试

### 相等

== 或者 =

**== 或者 = 两边一定要有空格**



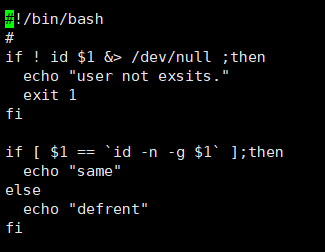
### 不等

**！=**

**< > <= >= 一般不用**

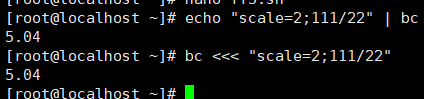
**-n string** 测试指定的字符串是否为空，空为真，不空为假

**-s string** 测试指定的字符串是否为空，空为假，不空为真



**echo "scale=2;111/22" | bc**

**bc <<< "scale=2;111/22"**



## 循环

**for while until 三种循环**

循环，有**进入条件**和**退出条件**。

for 变量 in 列表；do

循环体

done

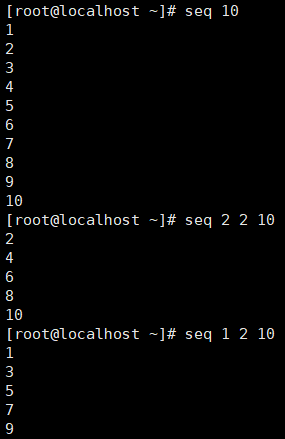
当列表遍历完成后就退出。

如何生成列表

1 {1..100}

2 seq 生成列表

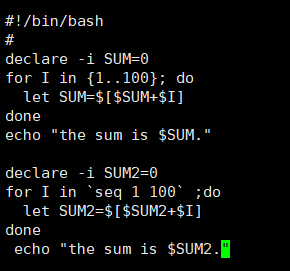
seq 1 2 10 第一个数字为**起始数**，第二个为**步长**，第三个为**结束数**



declare -i SUM 声明一个整型变量

declare -x toto 申明一个环境变量

1+2+3+…+100 = ? 写一个脚本



LINES=`wc -l /etc/passwd | cut -d ' ' -f 1`

echo $LINES

for I in `seq 1 $LINES` ; do echo "hello, ` head -n $I /etc/passwd | tail -1 | cut -d : -f 1`" ; done

写一个脚本

1.向系统中的每个用户问声好，并显示每个用户使用的shell

#!/bin/bash

#

declare -i T=`wc -l /etc/passwd | cut -d' ' -f 1`

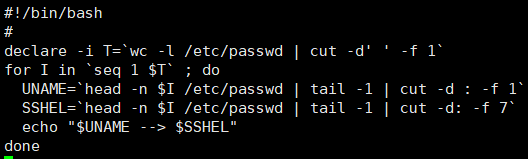
for I in `seq 1 $T` ; do

UNAME=`head -n $I /etc/passwd | tail -1 | cut -d : -f 1`

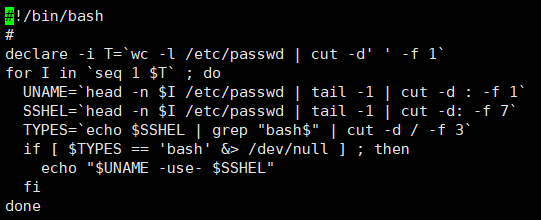
SSHEL=`head -n $I /etc/passwd | tail -1 | cut -d: -f 7`

echo "$UNAME --> $SSHEL"

done



2.只向shell默认为bash的用户问声好



#!/bin/bash

#

declare -i T=`wc -l /etc/passwd | cut -d' ' -f 1`

for I in `seq 1 $T` ; do

UNAME=`head -n $I /etc/passwd | tail -1 | cut -d : -f 1`

SSHEL=`head -n $I /etc/passwd | tail -1 | cut -d: -f 7`

TYPES=`echo $SSHEL | grep "bash$" | cut -d / -f 3`

if [ $TYPES == 'bash' &> /dev/null ] ; then

echo "$UNAME -use- $SSHEL"

fi

done

3.添加10个用户，usr1 - usr10,密码同用户名，要求存在就不添加

有问题的

#!/bin/bash

#

declare -i TOTA=`cat /etc/passwd | wc -l`

for I in `seq 1 10` ; do

ADDUNAME=user$I

for J in `seq 1 $TOTA` ; do

UNAME=`head -n $J /etc/passwd | tail -1 | cut -d : -f 1`

if [ $UNAME == $ADDUNAME ] ; then

echo "$ADDUNAME exists."

else

useradd $ADDUNMAE && echo "$ADDUNAME" | passwd --stdin $ADDUNAME &> /dev/null

echo "$ADDUNAME add successfully."

fi

done

done

还是有问题

#!/bin/bash

#

declare -i TOTA=`cat /etc/passwd | wc -l`

for I in `seq 1 10` ; do

ADDUNAME=user$I

UNAME=`grep "^user$I" /etc/passwd | cut -d : -f 1` 分不清user1 user10

if [ ${UNAME} == ${ADDUNAME} ] ; then

echo "$ADDUNAME exists."

else

useradd $ADDUNAME &> /dev/null && echo "$ADDUNAME" | passwd --stdin $ADDUNAME &> /dev/null

echo "$ADDUNAME add successfully."

fi

done

#!/bin/bash

#

declare -i TOTA=`cat /etc/passwd | wc -l`

for I in `seq 1 10` ; do

ADDUNAME=user$I

UNAME=`grep "^user" /etc/passwd | cut -d : -f 1 | grep "user$I$"` 这样才正确

if [ $UNAME == $ADDUNAME ] ; then

echo "$ADDUNAME exists."

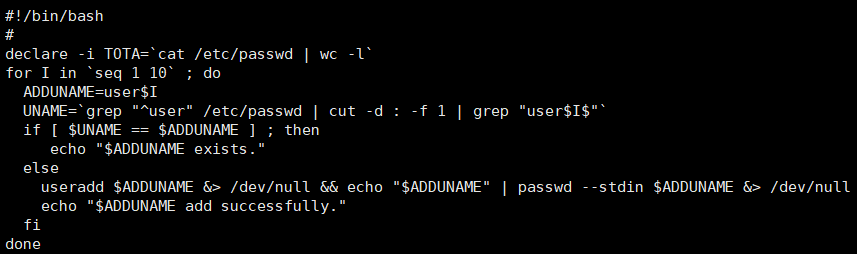
else

useradd $ADDUNAME &> /dev/null && echo "$ADDUNAME" | passwd --stdin $ADDUNAME &> /dev/null

echo "$ADDUNAME add successfully."

fi

done



扩展，接受一个参数

add : 添加用户

del ：删除用户

其他 ： 退出

#!/bin/bash

#

if [ $# -eq 0 ] ;then

echo "you must use a arguments."

exit 7

fi

if [ $1 == add ] ;then

for I in `seq 1 10` ; do

ADDUNAME=user$I

UNAME=`grep "^user" /etc/passwd | cut -d : -f 1 | grep "user$I$"`

if [ $UNAME == $ADDUNAME ] ; then

echo "$ADDUNAME exists."

else

useradd $ADDUNAME &> /dev/null && echo "$ADDUNAME" | passwd --stdin $ADDUNAME &> /dev/null

echo "$ADDUNAME add successfully."

fi

done

elif [ $1 == del ] ;then

for I in `seq 1 10` ; do

ADDUNAME=user$I

UNAME=`grep "^user" /etc/passwd | cut -d : -f 1 | grep "user$I$"`

if [ $UNAME == $ADDUNAME ] ; then

userdel -r $ADDUNAME &> /dev/null

echo "delete $ADDUNAME successfully."

fi

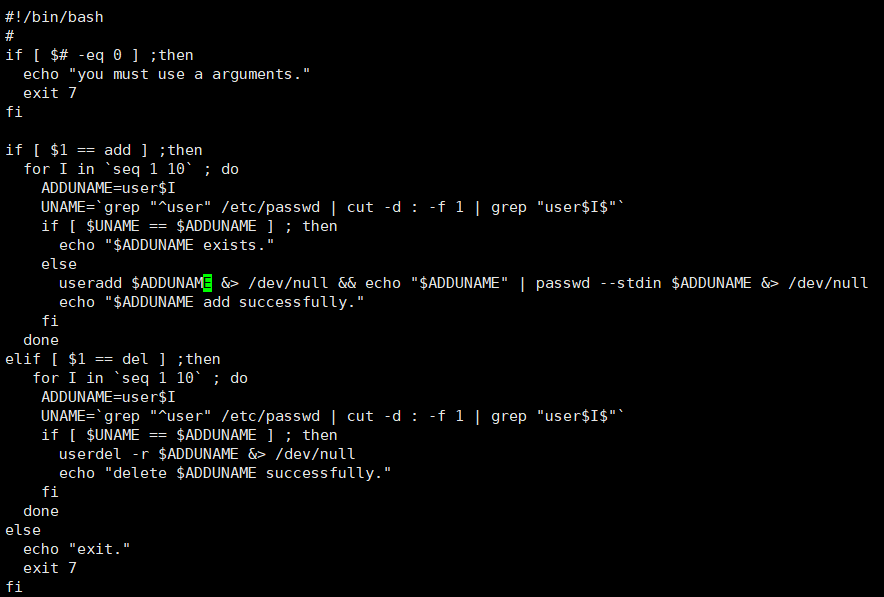
done

else

echo "exit."

exit 7

fi



4.计算100以内能被3整除的正整数的和

#!/bin/bash

#

declare -i TOTA=0

for I in `seq 1 100` ;do

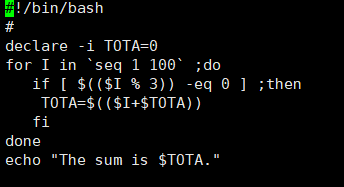
if [ $(($I % 3)) -eq 0 ] ;then

TOTA=$(($I+$TOTA))

fi

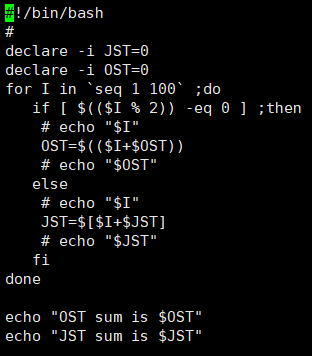
done

echo "The sum is $TOTA."





5.计算100以内奇数的和和偶数的和



#!/bin/bash

#

declare -i JST=0

declare -i OST=0

for I in `seq 1 100` ;do

if [ $(($I % 2)) -eq 0 ] ;then

# echo "$I"

OST=$(($I+$OST))

# echo "$OST"

else

# echo "$I"

JST=$[$I+$JST]

# echo "$JST"

fi

done

echo "OST sum is $OST"

echo "JST sum is $JST"

