**awk**

（1）统计词频

#! /bin/bash

# awk sort uniq

cat words.txt | awk '{i=1;while(i<=NF){print $i;i++}}' | sort | uniq -c \

| sort -k1nr | awk '{print $2" "$1}'

（2）文件旋转

#! /bin/bash

# awk

awk '{

for (i = 1; i <= NF; i++) {

if ((NR == 1)) {

num[i]=$i

} else {

num[i]=num[i] " " $i

}

}

}END {

for (i = 1; i <= NF; i++) {

print num[i]

}

}' file.txt

read

（3）打印指定列

#! /bin/bash

# print the first,third and fourth field of /etc/passwd

# awk

cat /etc/passwd| awk -F: '{print "user account="$1" uid="$3" gid="$4}'

（4）打印指定列满足条件的行

#! /bin/bash

# print the line which the third field is over 100 of /etc/passwd

# awk

cat /etc/passwd| awk -F: '{if($3>100) print $o}'

（5）输出开头和结尾

#! /bin/bash

# print the first field with a header and a footer

# awk

cat /etc/passwd| awk -F: 'BEGIN{print "========="} {print $1} END{print "========="}'

（6）获取最后一列

#! /bin/bash

# print line which line number > 15 and the last field matching bash

# print first field matching daemon

# awk regexp

# hint: $NF means the last field of current line; NF means the column of current line

cat /etc/passwd| awk -F: '{if(NR>15 && $NF~/bash/) print $o}'

cat /etc/passwd| awk -F: '{if($1~/daemon/) print $o}'

cat /etc/passwd| awk -F: '{if($1=="daemon") print $o}'

（7）输出第三列的行

#! /bin/bash

# print sum of thirf field

# awk varity

cat /etc/passwd| awk -F: '{sum = sum + $3} END{print sum}'

（8）gsub替换

#! /bin/bash

# replace root to yupi

# awk replace gsub

cat /etc/passwd| awk -F: '{gsub(/root/,"yupi");{print $o}}'

（9）~匹配

#! /bin/bash

# count line which the last field ends with bash

# awk wc

cat /etc/passwd| awk -F: '{if($NF~/bash$/) print NR}'| wc -l

cat /etc/passwd| awk -F: '{if($NF~/bash$/) print NR}'

（10）|正则匹配或

#! /bin/bash

# print line contains mail or bash

# awk regexp $o

# hint: $o will print by default

cat /etc/passwd| awk -F: '$o~/mail|bash/'

（11）电话文件处理

原文件

Mike Harrington:[510] 548-1278:250:100:175

Christian Dobbins:[408] 538-2358:155:90:201

Susan Dalsass:[206] 654-6279:250:60:50

Archie McNichol:[206] 548-1348:250:100:175

Jody Savage:[206] 548-1278:15:188:150

Guy Quigley:[916] 343-6410:250:100:175

Dan Savage:[406] 298-7744:450:300:275

Nancy McNeil:[206] 548-1278:250:80:75

John Goldenrod:[916] 348-4278:250:100:175

Chet Main:[510] 548-5258:50:95:135

Tom Savage:[408] 926-3456:250:168:200

Elizabeth Stachelin:[916] 440-1763:175:75:300

#! /bin/bash

# [tel.txt]

# awk

# print all phone numbers

echo "print all phone numbers"

cat tel.txt| awk -F: '{print $2}'

# print Dan's phone number

echo "print Dan's phone number"

cat tel.txt| awk -F: '($1~/^Dan/){print $2}'

# print Susan's name and phone number

echo "print Susan's name and phone number"

cat tel.txt| awk -F: '($1~/^Susan/){print $1 $2}'

# print name starts with 'D'

echo "print name starts with 'D'"

cat tel.txt| awk -F: '{if($1~/^D/) print $1}'

# print name starts with 'C' or 'E'

echo "print name starts with 'C' or 'E'"

cat tel.txt| awk -F: '($1~/^[C|E]/)'

# print name which length == 4

# hint: two delimeters; length()

echo "print name which length == 4"

cat tel.txt| awk -F '[: ]' '{if(length($1) == 4) print $1" "$2}'

# print name whose region number is 916

echo "print name whose region number is 916"

cat tel.txt| awk -F '[: ]' '($3~/916/) {print $1" "$2}'

# print mike's donation with a '$' before each money

echo "print mike's donation with a '$' before each money"

cat tel.txt| awk -F: '($1~/Mike/) {print "$"$3"$"$4"$"$5}'

# print name(reverse)

echo "print name(reverse)"

cat tel.txt| awk -F '[: ]' '{print $2","$1}'

**grep**

（1）获得有效的电话号码

#! /bin/bash

# get valid phone number

# easy

# https://leetcode-cn.com/problems/valid-phone-numbers/

# grep

grep -P '^(\d{3}-|\(\d{3}\) )\d{3}-\d{4}$' file.txt

（2）grep操作

#! /bin/bash

# grep

忽略大小写

# hint: grep -i : ignore case

echo "print line starts with 's' or 'S'"

grep -i ^s /proc/meminfo

不匹配

# hint: grep -v : not matching

echo "print line not ends with /bin/bash"

grep -v /bin/bash$ /etc/passwd

正则表达式

# hint regexp

echo "print two or three digits"

cat /etc/passwd|grep "\<[0-9]\{2,3\}\>"

grep常用参数说明

grep [OPTIONS] PATTERN [FILE...]

grep [OPTIONS] [-e PATTERN]... [-f FILE]... [FILE...]

OPTIONS:

-e: 使用正则搜索

-i: 不区分大小写

-v: 查找不包含指定内容的行

-w: 按单词搜索

-c: 统计匹配到的次数

-n: 显示行号

-r: 逐层遍历目录查找

-A: 显示匹配行及前面多少行, 如: -A3, 则表示显示匹配行及前3行

-B: 显示匹配行及后面多少行, 如: -B3, 则表示显示匹配行及后3行

-C: 显示匹配行前后多少行, 如: -C3, 则表示显示批量行前后3行

--color: 匹配到的内容高亮显示

--include: 指定匹配的文件类型

--exclude: 过滤不需要匹配的文件类型

grep常用用法

#多文件查询

grep leo logs.log logs\_back.log

#查找即包含leo又包含li的行

grep leo logs.log | grep li

#查找匹配leo或者匹配li的行

grep leo | li logs.log

#显示匹配行前2行

grep leo logs.log -A2

#显示匹配行后2行

grep leo logs.log -B2

#显示匹配行前后2行

grep leo logs.log -C2

#不区分大小写

grep -i leo logs.log

#使用正则表达式

grep -e '[a-z]\{5\}' logs.log

#查找不包含leo的行

grep -v leo logs.log

#统计包含leo的行数

grep -c leo logs.log

#遍历当前目录及所有子目录查找匹配leo的行

grep -r leo .

#在当前目录及所有子目录查找所有java文件中查找leo

grep -r leo . --include "\*.java"

#查找并输出到指定文件

grep leo logs.log > result.log

#查找以leo开头的行

grep ^leo logs.log

#查找以leo结尾的行

grep leo$ logs.log

#查找空行

grep ^$ logs.log

**sed**

（1）打印2-4行

#! /bin/bash

# sed

# hint: sed -n '2,4p' : print 2 to 4 lines

echo "print 2-4 lines of /etc/passwd"

sed -n '2,4p' /etc/passwd

1. 把/etc/passwd 复制到/root/test.txt，用sed打印所有行

2. 打印test.txt的3到10行

3. 打印test.txt中包含“root”的行

4. 删除test.txt的15行以及以后所有行

5. 删除test.txt中包含“bash”的行

6. 替换test.txt中“root”为“toor”

7. 替换test.txt中"/sbin/nologin"为"\bin/login"

8. 删除test.txt中5到10行所有的数字

9. 删除test.txt中所有特殊字符（除了数字以大小写字母）

10. 把test.txt中第一个单词和最后一个单词调换位置

11. 把test.txt中出现的第一个数字和最后一个单词调换位置

12. 把test.txt中第一个数字移动到行末尾； 13在test.txt 20行到末行最前面加“aaa”

cp - a / etc / passwd / root / test . txt ; sed - n '1,$' p test . txt

sed - n '3,10' p test . txt

sed - n '/root/' p test . txt

sed '15,$' d test . txt

sed '/bash/' d test . txt

sed 's/root/toor/g' test . txt

sed 's#sbin/nologin#bin/login#g' test . txt

sed '5,10s/[0-9]//g' test . txt

sed 's/[^0-9a-zA-Z]//g' test . txt

sed 's/\(^[a-zA-Z][a-zA-Z]\*\)\([^a-zA-Z].\*\)\([^a-zA-Z]\)\([a-zA-Z][a-zA-Z]\*$\)/\4\2\3\1/' test . txt

sed 's#\([^0-9][^0-9]\*\)\([0-9][0-9]\*\)\([^0-9].\*\)\([^a-zA-Z]\)\([a-zA-Z][a-zA-Z]\*$\)#\1\5\3\4\2#' test . txt

sed 's#\([^0-9][^0-9]\*\)\([0-9][0-9]\*\)\([^0-9].\*$\)#\1\3\2#' test . txt

sed 's/^.\*$/&aaa/' test . txt

**其他练习**

1. 打印第十行

#! /bin/bash

# print 10th line

# easy

# https://leetcode-cn.com/problems/tenth-line/

# tail head sed

tail -n +10 file.txt | head -n 1

# sed -n '10p' file.txt

（2）判断字符还是数字

#! /bin/bash

# judge input is num or string

# easy

# https://wenku.baidu.com/view/deab4b7b2bf90242a8956bec0975f46527d3a7ad.html 1

# regexp

read -p "input: " val

if [[ -z ${val/[0-9]\*/""} ]]; then

echo "num"

else

echo "string"

fi

（3）判断数字奇偶

#! /bin/bash

# judge input num is odd or even

# easy

# https://wenku.baidu.com/view/deab4b7b2bf90242a8956bec0975f46527d3a7ad.html 2

# case operator

read -p "input: " val

let left=val%2

# left=$[$val%2]

case $left in

1)

echo "odd"

;;

0)

echo "even"

;;

\*)

echo "invalid num"

;;

esac

（4）求和

#! /bin/bash

# get sum of input range

# easy

# https://wenku.baidu.com/view/deab4b7b2bf90242a8956bec0975f46527d3a7ad.html 3

# for operator

read -p "input min: " min

read -p "input max: " max

sum=0

for ((i=min; i<=max; i++)) {

let sum=sum+i

# sum=$[$sum+$i]

# sum=$((sum+i))

}

echo $sum

（5）利用函数和输入参数求和

#! /bin/bash

# get sum of input range

# easy

# https://wenku.baidu.com/view/deab4b7b2bf90242a8956bec0975f46527d3a7ad.html 4

# while operator function $@

function sum() {

n=$1

i=1

sum=0

while [[ i -lt $n ]]; do

sum=$((sum+i))

i=$((i+1))

done

echo $sum

}

sum $@

（6）复制每个目录中第一个文件

#! /bin/bash

# file copy, only copy first file in dir

# medium

# https://wenku.baidu.com/view/deab4b7b2bf90242a8956bec0975f46527d3a7ad.html 5

# file cp

origin="origin\_dir"

target="target\_dir"

files=`ls $origin`

mkdir $target

for file in $files; do

if [ -d $origin/$file ]; then

mkdir $target/$file

first\_file=`ls $origin/$file| head -n 1`

if [[ ! -z $first\_file ]]; then

cp $origin/$file/$first\_file $target/$file

fi

fi

done

（7）对每个用户问好，cut的使用

#! /bin/bash

# view all users in /etc/passwd and greet to every user

# medium

# https://wenku.baidu.com/view/deab4b7b2bf90242a8956bec0975f46527d3a7ad.html 6

# cut

file=`cat /etc/passwd`

num=0

for line in $file; do

username=`echo $line| cut -d: -f1`

userid=`echo $line| cut -d: -f3`

echo "Hello, $username, your uid is $userid"

num=$((num+1))

done

echo "total num is $num"

（8）文件操作

#! /bin/bash

# some file operation

# easy

# book practice 6-6

# case

case $1 in

1)

read -p "input a dir name: " dirname

ls $dirname

;;

2)

read -p "input a dir name: " dirname

cd $dirname

;;

3)

read -p "input a filename: " filename

mkdir /home/$filename

;;

4)

read -p "input a filename: " filename

vi $filename

;;

5)

exit

;;

esac

（9）用户和组操作

#! /bin/bash

# user and group operation

# easy

# book practice 6-7

# user and group

groupadd TestGroup

if [[ ! -d /share ]]; then

mkdir /share

fi

cd /share

for ((i=1;i<=30;i++)); do

mkdir std$i

useradd -d /share/std$i -g TestGroup std$i

done

# `groups [user]` can view all groups the user belongs to