**过滤mac地址**

ip a s | egrep '([0-9a-fA-F][0-9a-fA-F]:){5}[0-9a-fA-F]{2}' egrep不能换成grep -e

ip a s |awk '/([0-9a-f]{2}:){5}[0-9a-f]{2}/{print $2}'

**生成随机密码(字串截取版本)**

#!/bin/bash

key="0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

num=${#key}

for i in {1..8}

do

index=$[RANDOM%num]

onekey=${key:$index:1}

pass=$pass$onekey

done

echo $pass

**非交互自动生成 SSH 密钥文件**

rm -rf ~/.ssh/{known\_hosts,id\_rsa\*}

ssh-keygen -t RSA -N '' -f ~/.ssh/id\_rsa

**切割 Nginx 日志文件**

#mkdir /data/scripts

#vim /data/scripts/nginx\_log.sh

#!/bin/bash

logs\_path="/usr/local/nginx/logs/"

mv ${logs\_path}access.log ${logs\_path}access\_$(date -d "yesterday" +"%Y%m%d").log

**kill -USR1 `cat /usr/local/nginx/logs/nginx.pid`**

# chmod +x /data/scripts/nginx\_log.sh

#crontab -e

0 1 \* \* \* /data/scripts/nginx\_log.sh

**统计/etc/passwd 中 root 出现的次数**

#!/bin/bash

#每读取一行文件内容,即从第 1 列循环到最后 1 列,依次判断是否包含 root 关键词,如果包含则 x++

awk -F: '{i=1;while(i<=NF){if($i~/root/){x++};i++}} END{print "root 出现次数为"x}' /etc/passwd

**统计当前 Linux 系统中可以登录计算机的账户有多少个**

#!/bin/bash

#方法 1:

grep "bash$" /etc/passwd | wc -l

#方法 2:

awk -F: '/bash$/{x++}end{print x}' /etc/passwd

**统计每个远程 IP 访问了本机 apache 几次?**

#!/bin/bash

awk '{ip[$1]++}END{for(i in ip){print ip[i],i}}' /var/log/httpd/access\_log

**查看有多少远程的 IP 在连接本机(不管是通过 ssh 还是 web 还是 ftp 都统计)**

**#sort 可以按数字大小排序,最后使用 uniq 将多余重复的删除,并统计重复的次数**

netstat -atu | awk '{print $5}' | awk '{print $1}' | sort ‐nr | uniq ‐c

**Shell 脚本的 fork 炸弹**

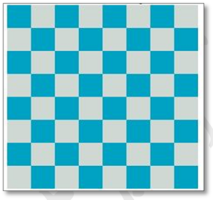
#!/bin/bash

#快速消耗计算机资源,致使计算机死机

#定义函数名为.(点), 函数中递归调用自己并放入后台执行

.() { .|.& };.

打印国际象棋棋盘,效果如下图:



#!/bin/bash

#设置两个变量,i 和 j,一个代表行,一个代表列,国际象棋为 8\*8 棋盘

#i=1 是代表准备打印第一行棋盘,第 1 行棋盘有灰色和蓝色间隔输出,总共为 8 列

#i=1,j=1 代表第 1 行的第 1 列;i=2,j=3 代表第 2 行的第 3 列

#棋盘的规律是 i+j 如果是偶数,就打印蓝色色块,如果是奇数就打印灰色色块

#使用 echo -ne 打印色块,并且打印完成色块后不自动换行,在同一行继续输出其他色块

for i in {1..8} echo -n (不换行输出) -e(扩展功能)

do

for j in {1..8}

do

sum=$[i+j]

if [ $[sum%2] -eq 0 ];then

echo ‐ne "\033[46m \033[0m"

**对 100 以内的所有正整数相加求和(1+2+3+4...+100)**

#!/bin/bash

#seq 100 可以快速自动生成 100 个整数

for i in `seq 100`

do

sum=$[sum+i]

done

echo "总和是:$sum"

**编写一个点名器脚本**

#!/bin/bash

#该文件中需要包含所有姓名的信息,一行一个姓名,脚本每次随机显示一个姓名

while :

do

line=`cat user.txt |wc ‐l`

num=$[RANDOM%line+1]

sed ‐n "${num}p" user.txt

sleep 0.2

clear

done

**9\*9 乘法表(编写 shell 脚本,打印 9\*9 乘法表)**

#!/bin/bash

for i in `seq 9`

do

for j in `seq $i`

do

echo -n "$i\*$j=$[i\*j] "

done

echo

done

**进度条,动态时针版本**

#!/bin/bash

#定义一个显示进度的函数,屏幕快速显示| / ‐ \

rotate\_line(){

INTERVAL=0.1 #设置间隔时间

COUNT="0" #设置 4 个形状的编号,默认编号为 0(不代表任何图像)

while :

do

COUNT=`expr $COUNT + 1` #执行循环,COUNT 每次循环加 1,(分别代表 4 中不同的形状)

case $COUNT in #判断 COUNT 的值,值不一样显示的形状就不一样

"1") #值为 1 显示‐

echo ‐e '‐'"\b\c"

sleep $INTERVAL

;;

"2") #值为 2 显示\\,第一个\是转义

echo ‐e '\\'"\b\c"

sleep $INTERVAL

;;

"3") #值为 3 显示|

echo ‐e "|\b\c"

sleep $INTERVAL

;;

"4") #值为 4 显示/

echo ‐e "/\b\c"

sleep $INTERVAL

;;

\*) #值为其他时,将 COUNT 重置为 0

COUNT="0";;

esac

done

}

rotate\_line

#使用 stty ‐echo 关闭 shell 的回显功能

#使用 stty echo 打开 shell 的回显功能

**统计双色球各个数字的中奖概率**

#1.txt 往期双色球中奖号码如下:

01 04 11 28 31 32 16

04 07 08 18 23 24 02

02 05 06 16 28 29 04

04 19 22 27 30 33 01

05 10 18 19 30 31 03

02 06 11 12 19 29 06

awk '{print $1"\n"$2"\n"$3"\n"$4"\n"$5"\n"$6}' 1.txt |sort| uniq -c |sort //蓝球各数字出现次数排序

awk '{print $7}' 1.txt |sort |uniq -c | sort //红球各数字出现次数排序

// **uniq -c** 1.txt 检查文件并删除文件中重复出现的行，并在行首显示该行重复出现的次数

echo $(( 2#101011 )) 数值转换,输出43 echo $(( 2#10 )) 输出2

**case** "$variable" **in**

**abc)** echo "\$variable = abc" **;;**

**xyz)** echo "\$variable = xyz" **;;**

**esac**

**${#a} 获取a变量的长度**

**${var} 限定变量var**

**如果一个变量名A是另一个变量名AB的前部分，那么，如果要在AB相连时，要得到A的值，就必须使用${var}限定。**

**如果变量名不会产生歧义时（即独立的变量名），${var}与$var没有任何区别。**

**var="hello"**

**var\_01="hello01"**

**echo  $var $var\_01 ${var}\_01**

**##输出 hello hello01 hello\_01**

**echo $(uuidgen) 产生16位的UUID 38f74307-019a-45e5-b067-4d7ea64fdfdf**

**条件测试 test 或者 [ ]**

**-a 逻辑与 -o 逻辑或**