第一题

(1)shutdown –r 10 回车

wall 回车

our machine will restart in 10 minutes later, please save your project 回车

键盘按下[ctrl+d]

(2) ls –l example | grep -c '^d'

(3) shuf -n5 file.txt –o file\_temp.txt

sort -k2n file\_temp.txt –o sample.txt

第二题

(1)

#打开cron

sudo service cron start

#编辑自己的crontab任务计划

crontab -e

#输入操作

20 2 \* \* \* rm -rf /confbackup/\*

#重启cron

sudo service cron restart

(2)

sudo service cron start

crontab -e

0 8-14/1 \* \* \* bash /home/noone/test.sh

sudo service cron restart

其中test.sh中的内容为:

#!/bin/sh

array=($(find /home/user/workspace -type f -name \*.log))

echo $array

for name in ${array[\*]};

do

cat $name >> /backup/bak.txt

done

第三题

3-1添加用户组

#!/bin/bash

sudo su

i=1

groupadd students

while [ $i -le 20 ]

do

if [ $i -le 9 ] ;then

USERNAME=stu0${i}

else

USERNAME=stu${i}

fi

useradd $USERNAME

mkdir /home/$USERNAME

chown -R $USERNAME /home/$USERNAME

chgrp -R students /home/$USERNAME

i=$(($i+1))

done

3-2打印ascii码表

#!/bin/bash

ascii=("!" "\"" "#" "$" "%" "&" "'" "(" ")" );

printf "%s %s %s\n" Decimal HEX Character

echo ---------------------

for i in ${ascii[@]};

do

printf "%4d%6x%5c\n" "'$i" "'$i" "$i";

done

printf "%4d%6x%5c\n" "'\*" "'\*" "\*";

ascii\_1=("+" "," "-" "." "/" "0" "1" "2" "3" "4" "5" "6" "7" "8" "9" ":" ";" "<" "=" ">" "?" "@" "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z" "[" "\\" "]" "^" "\_" "\`" "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "x" "y" "z" "{" "|" "}" "~")

for i in ${ascii\_1[@]};

do

printf "%4d%6x%5c\n" "'$i" "'$i" "$i";

done

第四题.

#!/bin/bash

points=(2 3 4 5 6 7 8 9 10 "J" "Q" "K" "A") #点数

spec=("红桃" "黑桃" "方块" "梅花") #花色

poker=()

## 将点数和花色组成一副扑克

function makepoker(){

v=0;

for i in ${points[@]};do

for j in ${spec[@]};do

v=$(($v+1));

if [[ $j == "红桃" || $j == "方块" ]];then

poker[$v]="\033[33m"$j"=>"$i"\033[0m";

else

poker[$v]=$j"=>"$i;

fi

done;

done;

}

## 使用递归函数，生成随机扑克索引

function seed\_random(){

if [[ ${#poker[@]} != 0 ]];then

if [[ ${#porker[@]} == 1 ]];then

return 0

else

k=$(rand 0 52);

#echo $k

if [[ ${poker[$k]} == "" ]] ;then # 当随机数不再数组时在生成一个随机数，直到随机数为数组的索引，且值不为空

seed\_random

else

echo $k;

return;

fi

fi

else

return 0;

fi

}

## 生成0-52的随机数

function rand(){

min=$1

max=$(($2-$min+1))

num=$(date +%s%N)

echo $(($num%$max+$min))

}

## 随机发牌

function deal\_cards(){

echo "开始随机发牌......"

for i in {1..4};do

printf "玩家（${i}）抓牌:\n\n"

for j in {1..13};do

kv=$(seed\_random)

printf "%10s${j} %s\033[03m"${poker[$kv]}"\033[0m\n"

unset poker[$kv] #将发出的牌删除

done;

printf "\n"

done;

}

echo "正在洗牌......"

makepoker

deal\_cards

第五题

(1)sed -n '23p' file.txt

head -n 23 file.txt|tail -n 1

awk '{if(NR==23) print }' file.txt

(2)cat file.txt |tr -cs "[a-z][A-Z]" "[\012\*]"|tr [A-Z] [a-z]|sort|uniq -c|sort -k1n -k2

(3)#!/bin/bash

while true;do

for i in 20001 20002 20003

do

count=`netstat -anp |grep "$i"|grep LISTEN|wc -l`#前面的参数是小写的L不是1

if [ "$count" == "0" ];then

./start "$i"

fi

sleep 5

done

done

第六题

#! /bin/bash

file\_names=($(ls ./models))

index=0

for file\_name in ${file\_names[\*]}

do

if [ "${file\_name##\*.}" = "model" ];then

model\_list[index]=$file\_name

temp=($(./bin deploy.prototxt models/$file\_name | tail -n 1))

model\_accuracy[index]=${temp[-1]}

index=$((index+1))

fi

done

max=${model\_accuracy[0]}

max\_index=0

length=${#model\_accuracy[\*]}

for((i=1;i<length;i++))

do

if [ ${model\_accuracy[i]} -gt $max ];then

max=${model\_accuracy[i]}

max\_index=$i

fi

done

echo The model with the best accuracy ${model\_accuracy[max\_index]} is\

${model\_list[max\_index]}

第七题

Android.mk:

LOCAL\_PATH:= $(call my-dir)

include $(CLEAR\_VARS)

OPENCV\_INSTALL\_MODULES:=on

OPENCV\_CAMERA\_MODULES:=off

include \home\user\OpenCV-android-sdk\sdk\native\jni\OpenCV.mk

LOCAL\_C\_INCLUDES:= ../OpenCV-android-sdk/sdk/native/jni/include

LOCAL\_MODULE := nonfree

LOCAL\_CFLAGS := -Werror -O3 -ffast-math

LOCAL\_LDLIBS += -llog

LOCAL\_SRC\_FILES := nonfree\_init.cpp \

sift.cpp \

surf.cpp

# 条件预编译，如果当前架构为ARMv7以下的架构，则定义MY\_NO\_CSWAP这个宏

ifeq ($(TARGET\_ARCH\_ABI),armeabi)

LOCAL\_CFLAGS := -DMY\_NO\_CSWAP=1

endif

include $(BUILD\_SHARED\_LIBRARY)

Application.mk:

APP\_ABI := armeabi armeabi-v7a mips

APP\_STL := stlport\_static

APP\_STL := gnustl\_static

#允许使用c++11的函数等功能

APP\_CPPFLAGS +=-std=c++11

# 启用Blocks语法

APP\_CFLAGS += -fblocks