1. Linux基础命令
2. date
3. uname –a
4. shutdown –r 20 回车

wall 回车

our machine will restart in 20 minutes later, please save your project 回车

键盘按下[ctrl+d]

1. Contab

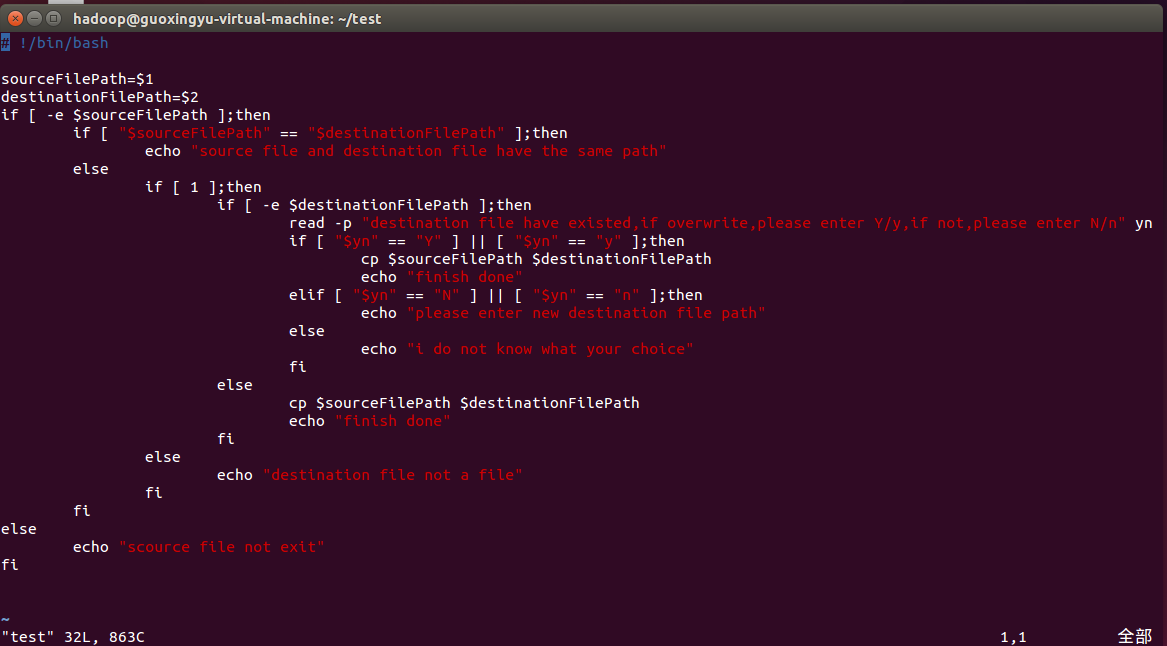
<1>crontab –e 回车，进入vi编辑

输入10 2 \* \* \* rm /cdf/\*

<2> crontab –e 回车

0 0 \* \* 0 cp /user/backup /tmp

1. Shell编程



问题：判断格式怎么写

if[1];then 这行改为：

if [ed $destinationFilePath];then

4、Ssh非交互式

Step1：首先在A服务器上执行ssh-keygen -t rsa 生成本机的公钥和私钥文件，上述命令执行后，隐藏目录~/.ssh下会出现两个文件：id\_rsa和id\_rsa.pub。其中，id\_rsa.pub为公钥文件。

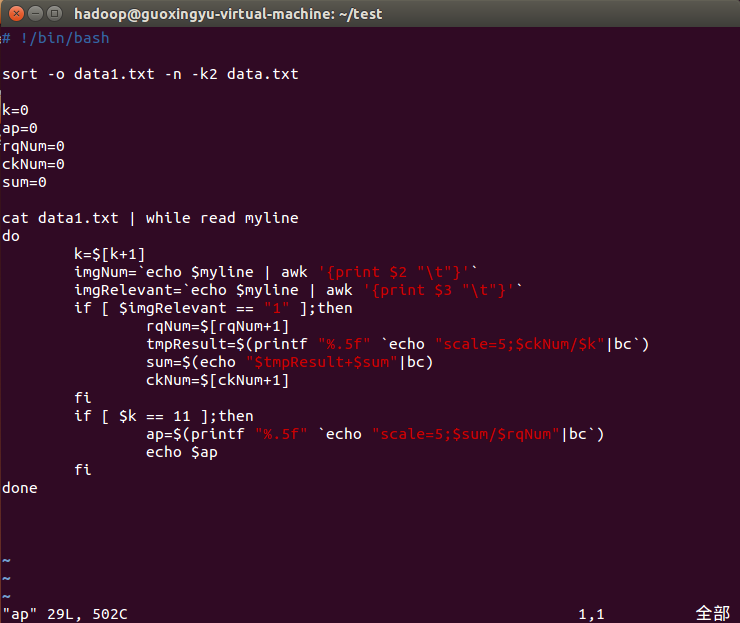
Step2：将id\_rsa.pub的内容追加到服务器B上~/.ssh目录下的authorized\_keys文件中。若该文件不存在，可自行创建之。

scp /root/.ssh/id\_rsa.pub [root@10.193.251.192:~/.ssh/authorized\_keys](mailto:root@10.193.251.192:~/.ssh/authorized_keys)

Step3:  接下来，就可以在不输入密码的情况下在远程主机私执行命令了。比如在服务器Ａ上，输入以下命令

 ssh root@10.193.251.192

1. shell 编程



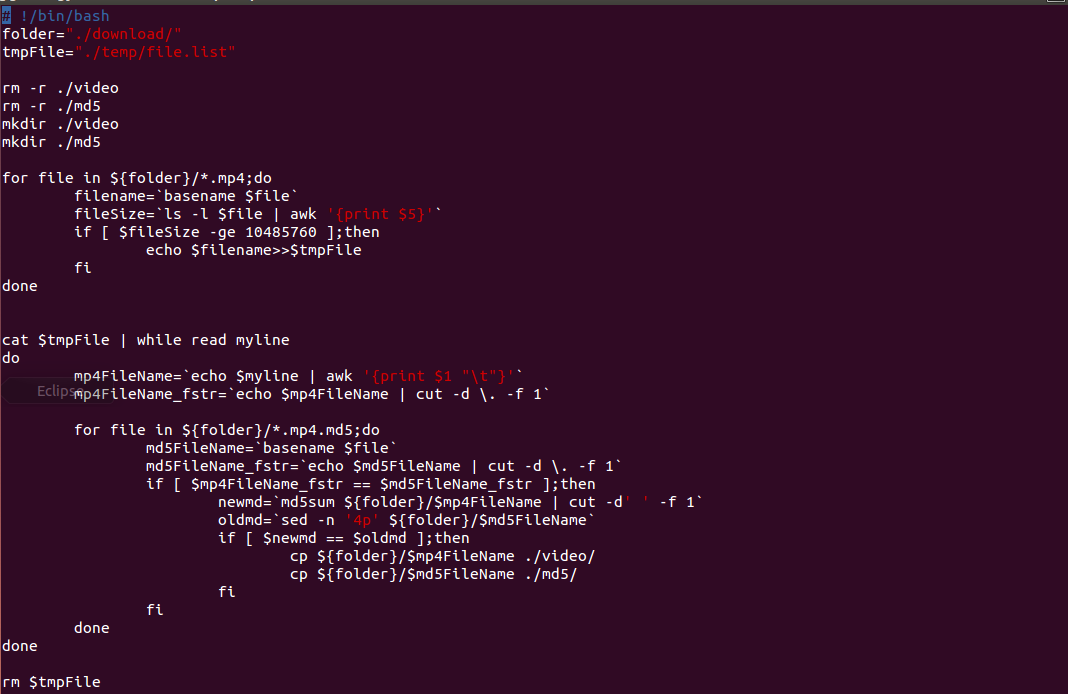
# 读取data1.txt文件中的每一行

# 取出图片序号

# 取图片相关度

if[ $k ==11 ];then 这行改为：

if[ $k ==1000 ];then

6、

# 读取所有/download/目录下的.mp4文件，判断是否大于10M，并写入file.list中

# 读取file.list内容

# 读取mp4格式文件的文件名

# 读取/download/目录下的.mp4.md5文件名

# 比较两个文件名是否一致

# 获取.mp4文件的md5码

# 获取.mp4.md5的md5码

# 拷贝文件

删除rm –r ./video这行

删除rm –r ./mds这行

删除mkdir ./video这行

删除mkdir ./mds这行

删除rm $tmpFile这行

1. gcc

//main.cpp

#include<function.h>

const int N = 100;

void main(int argc, char \*\*argv)

{

int a = char\_to\_int(argv[1]);

int b = char\_to\_int(argv[2]);

char result[N];

add\_char(a,b,result);

cout<<result;

}

//function.h

#include<iostream>

Using namespace std;

void add\_char(int a,int b,char \*c);

int char\_to\_int(char \*a);

void rotate(char \*c,int n);

//function.cpp

#include<function.h>

int char\_to\_int(char \*a)

{

int result = 0;

while(\*a != ‘/0’)

{

result = result \* 2 + (\*a - ‘0’);

a++;

}

return result;

}

void add\_char(int a,int b,char \*c)

{

int tmp = a + b;

int i = 0;

while(tmp)

{

c[i] = tmp % 2;

i++;

tmp = tmp / 2;

}

Rotate(c,i);

c[i] = ‘/0’;

}

void rotate(char \*c,int n)

{

int i = 0;

while(i != n/2)

{

Char tmp = c[i]; c[i] = c[n-1-i]; c[n-1-i] = tmp;

i++;

}

}

//Makefile

main: main.o function.o

gcc -o main main.o function.o

main.o:main.cpp function.h

gcc -c main.cpp

function.o:function.cpp function.h

gcc -c function.cpp

8、nfs mount

在服务器（10.193.251.191）上以root用户执行：

1. setfacl –m u:\*:rwx /home/data1 data2 data3

怎么给客户端用户赋予修改权限

（2）在/etc/exports文件中，添加：

/home/data1 10.193.251.192(ro,all\_squash)

/home/data2 10.193.251.192(rw,root\_squash,sync)

/home/data3 10.193.251.192(rw,no\_root\_squash,async)

在客户端（10.193.251.192）上以root用户执行：

（1）创建挂载点目录

mkdir –p /mnt/{data1,data2,data3}

（2）挂载

Mount –t nfs 10.193.251.191:/home/data1 /mnt/data1

Mount –t nfs 10.193.251.191:/home/data2 /mnt/data2

Mount –t nfs 10.193.251.191:/home/data3 /mnt/data3