经验与心得  
1.最好用双引号定义变量 " $HAT" 避免有空格或者变量名输错的错误.

Option1 ， 可以新建一个变量 USR = “/usr” ，然后`ls -a $USR`

2. Input a name, report whether it is a file, a user, a device, a command, or nobody

Device -c可以查字符设备。-b FILE FILE exists and is block special 块设备

Command -x 试试。

3.不可读 if !( –w )

4.

hexdump -C

-C 输出规范的十六进制和ASCII码。

 5.ftp自动登录批量下载文件。  
#####从ftp服务器上的/home/data 到 本地的/home/databackup####  
#!/bin/bash  
ftp -n<<!  
open 192.168.1.171  
user guest 123456  
binary  
cd /home/data  
lcd /home/databackup  
prompt  
mget \*  
close  
bye  
!

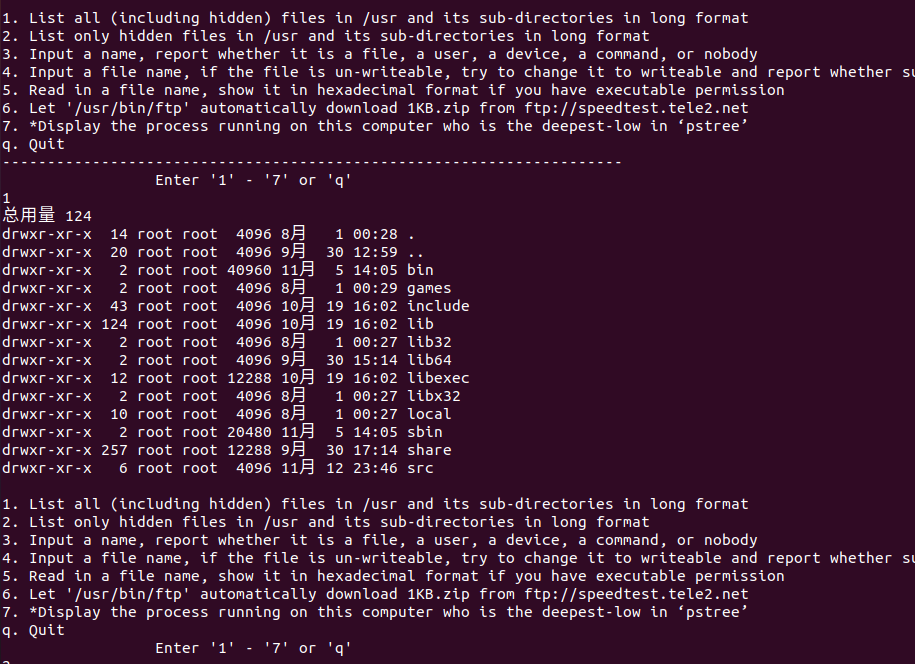
6. 你的程序不应该沉默, 应该很多地方有echo.

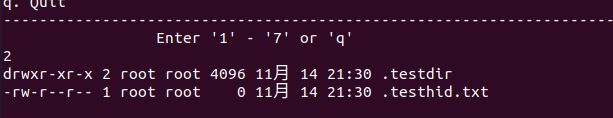
报错：

1未预期的符号“elif”附近有语法错误

Solution : 我都加了then,加了then好像就可以了

实现截图：

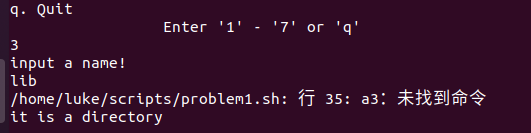




Option2 完成

## Option3

他只能看shell所在文件夹下的文件,文件夹等.



于是我修复了一下， 可以少显示“未找到命令”。

elif [ "$name" != " " ]

                        a3 = `cat /etc/passwd |cut -d ':' -f 1 | grep $name`;

                        if [ "$a3" != " " ]

                             then

                                echo 'it is a user name.'

                else

                        echo 'nothing'

                fi;

把判断放在最后

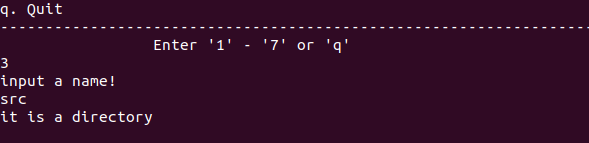
a3 = `cat /etc/passwd |cut -d ':' -f 1 | grep $name`;

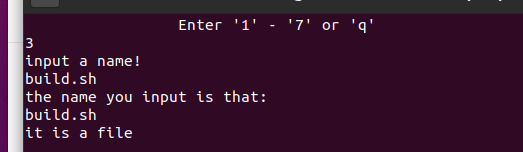
Option3:

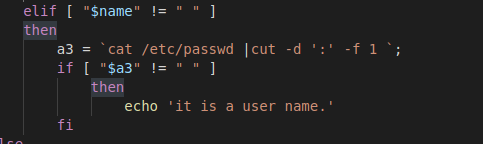
为什么

a3=`cat /etc/passwd |cut -d ':' -f 1 | grep $name`;

在shell里可以, 执行脚本就显示没有找到呢?







事实上这是个错误, 等号两边不能有空格.不然这句话就变成了，运行一个叫a3的命令，传给它第一个参数是“=” .

为什么他总是等于true呢? != “ ” 或者 ! $a3也可以

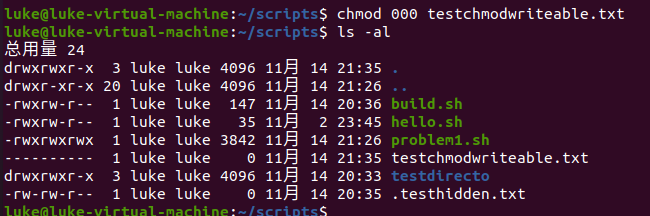
不, 不是这个的问题.

if [ "$uname" != "" ] 可以有效判断

! $uname不能有效判断 为空

## Option4:

我先建立一个testchmodwriteable.txt



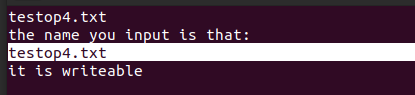
 if [ ! -w "$name"]

好像要加个空格不能[!紧挨着. “] 也不行, 都要有空格.

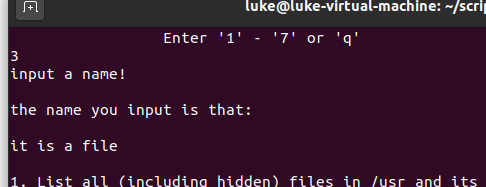
else

echo "it is writeable"

任何分支都应该有回应:



打一个空格进去也会判断为file:



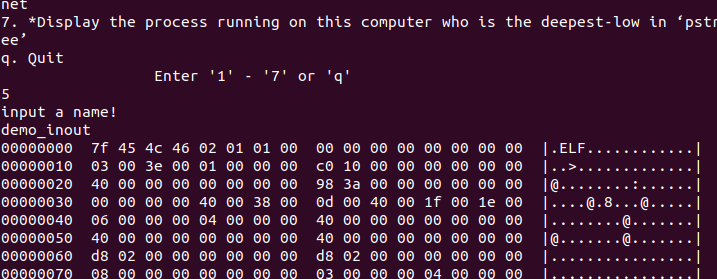
假如说你变量是个空格，

最好所有变量都加引号

shell其实做的就是文本替换，不加引号替换之后就是空的了

## Option5:

截图如下:



## Option 6:

425 Failed to establish connection.

因此，可能是需要客户端和服务器同时在防火墙中开放ftp服务。那么才能在命令行下进行数据传输吧（windows的cmd下面和一些FTP客

因此，可能是需要客户端和服务器同时在防火墙中开放ftp服务。那么才能在命令行下进行数据传输吧

ftp传输文件错误“421 Data timeout. Reconnect”，代表数据连接超时，连接被服务器强制断开了，只要我们找到vsftpd.conf文件，将文件里面的data\_connection\_timeout参数值改大一点的就可以解决问题了，特别是当我们网络环境不好的时候特别容易出现这个问题。

quote PASV

不行

**sudo ufw status命令**来查看当前防火墙的状态 如果是**inactive**说明我们的防火墙已经关闭掉了。

sudo ufw allow 20

sudo ufw default allow

还是不能下载. 425 Failed to establish connection.

## option 7 ：

for processid in $(ps -ax | awk 'NR>1 {print $1}' ); do

depth=$(pstree -sA $processid | head -n1 | sed -e 's#-+-.\*#---foobar#' -e 's#---\*#\n#g' -eq | wc -l)

echo "$depth: $processid"

done

sed -e<script>或--expression=<script> 以选项中指定的script来处理输入的文本文件。

s ：取代，可以直接进行取代的工作哩！通常这个 s 的动作可以搭配正规表示法！例如 1,20s/old/new/g 就是啦！

foo-bar，其中bar是beyond all recognition的缩写，通俗点就是无法识别，一塌糊涂的意思。

wc 计算字数

使用>指令覆盖文件原内容并重新输入内容，若文件不存在则创建文件。

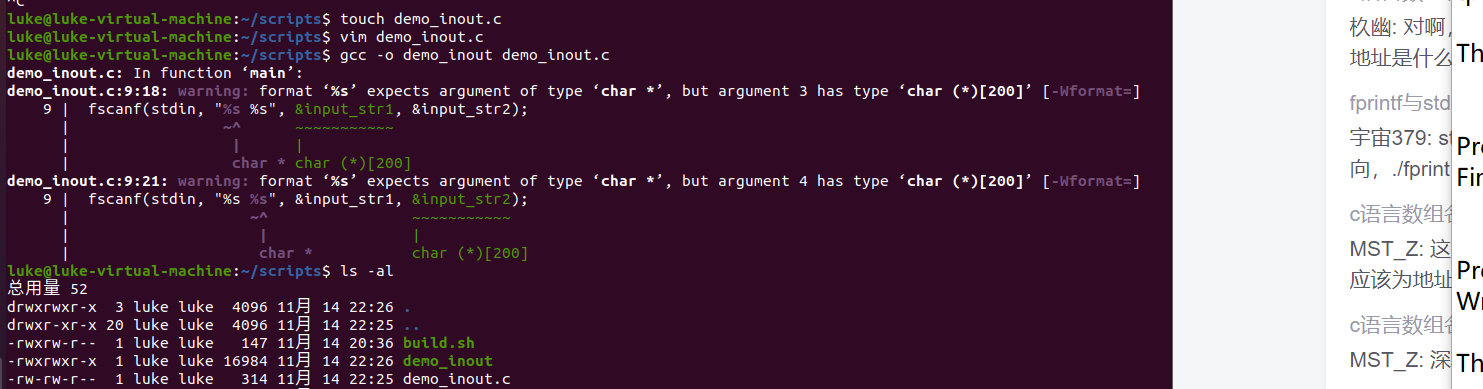
#!/bin/bash

echo "Raspberry" > test.txt

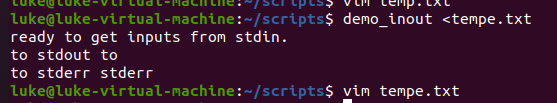
sort 排序,参考这篇: <https://www.jianshu.com/p/c4d159a98dd8>

# Problem2

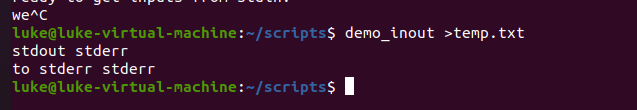
read input from a file, and in the same time, send two outputs to two separate files. Figure out the mechanism for stdin, stdout and stderr redirection.



输入重定向



:

stdout -- 标准输出设备 (printf("..")) 同 stdout。  
stderr -- 标准错误输出设备 两者默认向屏幕输出。  
但如果用转向标准输出到磁盘文件，则可看出两者区别。stdout输出到磁盘文件，stderr在屏幕。  
  
strerr是作为程序运行过程中的错误显示出来的，若想将它重写向到某文件中，需要运行如下命令：  
./fprint 2>temp.txt



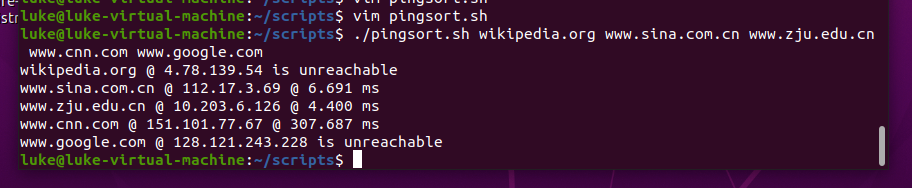
# Problem3

b：未找到命令

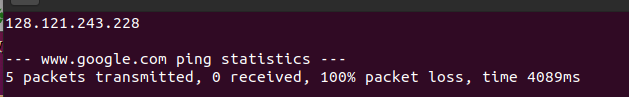
这里不能加空格.

a=`sed -n 1p pingsort\_1.txt`

加了空格会出错



没有排序.



原来是覆盖了.

>               输出重定向到一个文件或设备 覆盖原来的文件

>!              输出重定向到一个文件或设备 强制覆盖原来的文件

>>             输出重定向到一个文件或设备 追加原来的文件

ping -c 5 $args | cut -d '(' -f2 | cut -d ')' -f1 | cut -d '/' -f 5 >> pingsort\_1.txt

截取出ip地址来,

-d：指定字段的分隔符，默认的字段分隔符为“TAB”；

-f：显示指定字段的内容；

ping [www.baidu.com](http://www.baidu.com) 得到的格式是这样的

PING www.a.shifen.com (36.152.44.95) 56(84) bytes of data.

64 bytes from 36.152.44.95 (36.152.44.95): icmp\_seq=1 ttl=128 time=19.6 ms

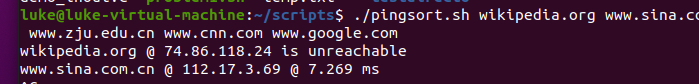
64 bytes from 36.152.44.95 (36.152.44.95): icmp\_seq=2 ttl=128 time=12.5 ms

64 bytes from 36.152.44.95 (36.152.44.95): icmp\_seq=3 ttl=128 time=14.6 ms

64 bytes from 36.152.44.95 (36.152.44.95): icmp\_seq=4 ttl=128 time=13.7 ms

64 bytes from 36.152.44.95 (36.152.44.95): icmp\_seq=5 ttl=128 time=14.5 ms

64 bytes from 36.152.44.95 (36.152.44.95): icmp\_seq=6 ttl=128 time=15.0 ms

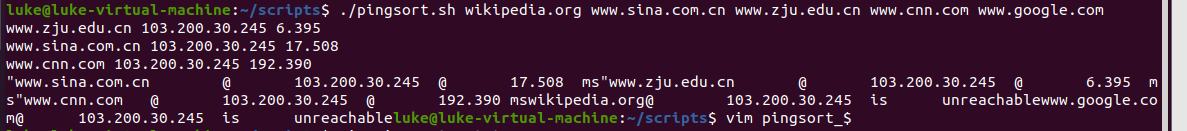


我需要每一个判断完成, 记录下来,然后排序

**SHELL 读取文件的每一行内容并输出**

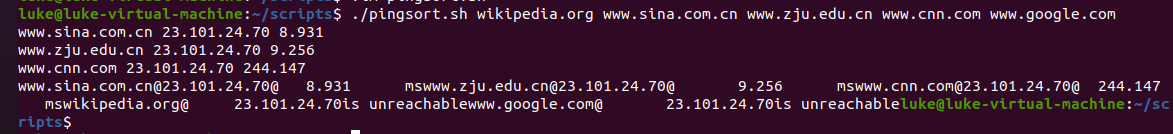
我想要加读取的时候,加上文字. 不知道怎么做.

printf命令并不适用于管道传输（也就是说它并不支持标准输入），而且在命令行应用中它也不常见（多应用于脚本文件）。



排序已经完成了,不过似乎还是有一点点问题.

%-10s 指一个宽度为10个字符（-表示左对齐，没有则表示右对齐），任何字符都会被显示在10个字符宽的字符内，如果不足则自动以空格填充，超过也会将内容全部显示出来。



少了 \n 还有就是sort了之后会输出一次.

：systemctl  disable  firewalld.service >> /dev/null 2>&1

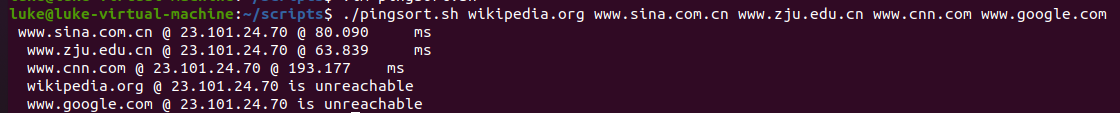
解读：

1. /dev/null  看作"黑洞"，它非常等价于一个只写文件，所有写入它的内容都会永远丢失。而尝试从它那儿读取内容则什么也读不到。
2. 2>&1 是将错误输出到标准输出。

延伸：

1> 指标准信息输出路径（也就是默认的输出方式）  
2> 指错误信息输出路径  
2>&1 指将标准信息输出路径指定为错误信息输出路径（也就是都输出在一起）﻿﻿

Ip全都一样了,这是为啥.





格式有一点点问题, pingsort3 unreachable部分已经对了

因为我追加了,所以之前的错误留下来了.

Linux把变量输入到 txt中

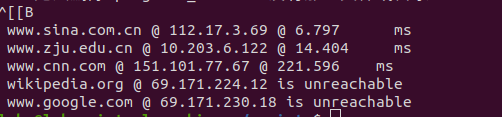
echo "$args $a $b" >> pingsort\_4.txt

错误二: 不能盲目追加,最好是 >到一个新文件, 然后rm掉新文件.

pingsort\_4.txt 在循环里, 必须要>>追加.

# tail -n 3 pingsort\_4.txt >> pingsort5.txt 选择最后3行

错误三: \n" 之后不要 \n “,不然换行后再空格, 会越来越不整齐.

最终截图: 

Problem 4.

**wget命令**用来从指定的URL下载文件。wget非常稳定，它在带宽很窄的情况下和不稳定网络中有很强的适应性，如果是由于网络的原因下载失败，wget会不断的尝试，直到整个文件下载完毕。如果是服务器打断下载过程，它会再次联到服务器上从停止的地方继续下载。这对从那些限定了链接时间的服务器上下载大文件非常有用。

grep test test\* #查找前缀有“test”的文件包含“test”字符串的文件

testfile1:This a Linux testfile! #列出testfile1 文件中包含test字符的行

testfile\_2:This is a linux testfile! #列出testfile\_2 文件中包含test字符的行

testfile\_2:Linux test #列出testfile\_2 文件中包含test字符的行

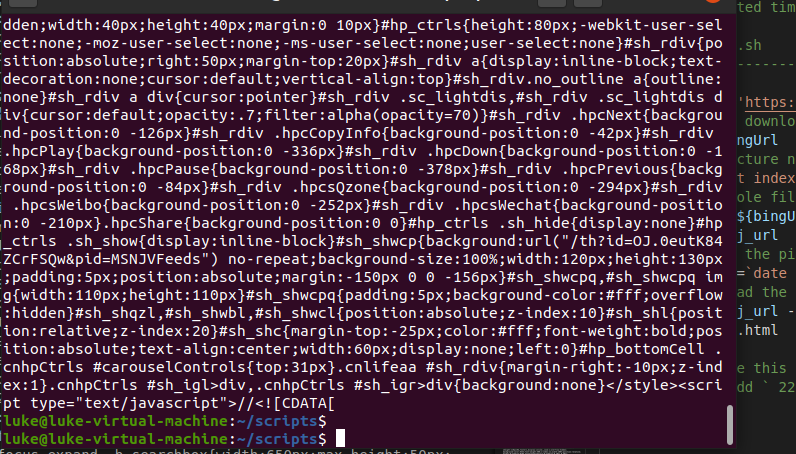
分析了一下index.html

下面这两个或许有图.

file=`cat index.html | grep ' href=' | cut -d '”' -f2 | cut -d '”' -f1`

file=`cat index.html | grep 'background-image' | cut -d '(' -f2 | cut -d ')' -f1`

这样会提取不正确. 这个html 一行非常的长.



这个不能用列处理,因为空格是不规则的.

提取name.txt 文件中的最后一列中以：分割的第二列 awk（符截取命令）

awk -F '\t' '{print $NF}' name.txt|awk -F ':' '{print $2}'>res.txt

其中-F 表示分割符，$NF表示最后一列，$(NF-1)表示倒数第二列 $2表示第二列 ，$0表示全部，$1表示第一列

　打开linux终端>进入cat（我自己创建的）目录>创建crontab文件夹 并进入> 执行命令 crontab -e进入编辑状态编写crontab表达式 1 \* \* \* \* /cat/crontab/sbin.sh(每天凌晨1点执行total.sh脚本)>

Ctrl+X 建退出>Yes(保存)>回车> crontab -l （如果能看到你刚才保存的crontab，就成功了~~）

Crontab -e后产生文件

# Edit this file to introduce tasks to be run by cron.

#

# Each task to run has to be defined through a single line

# indicating with different fields when the task will be run

# and what command to run for the task

#

# To define the time you can provide concrete values for

# minute (m), hour (h), day of month (dom), month (mon),

# and day of week (dow) or use '\*' in these fields (for 'any').

#

# Notice that tasks will be started based on the cron's system

# daemon's notion of time and timezones.

#

# Output of the crontab jobs (including errors) is sent through

# email to the user the crontab file belongs to (unless redirected).

#

# For example, you can run a backup of all your user accounts

# at 5 a.m every week with:

# 0 5 \* \* 1 tar -zcf /var/backups/home.tgz /home/

#

# For more information see the manual pages of crontab(5) and cron(8)

#

# m h dom mon dow command

0 12 \* \* \* ~/scripts/pro4.sh 每天中午

Crontab –e 修改

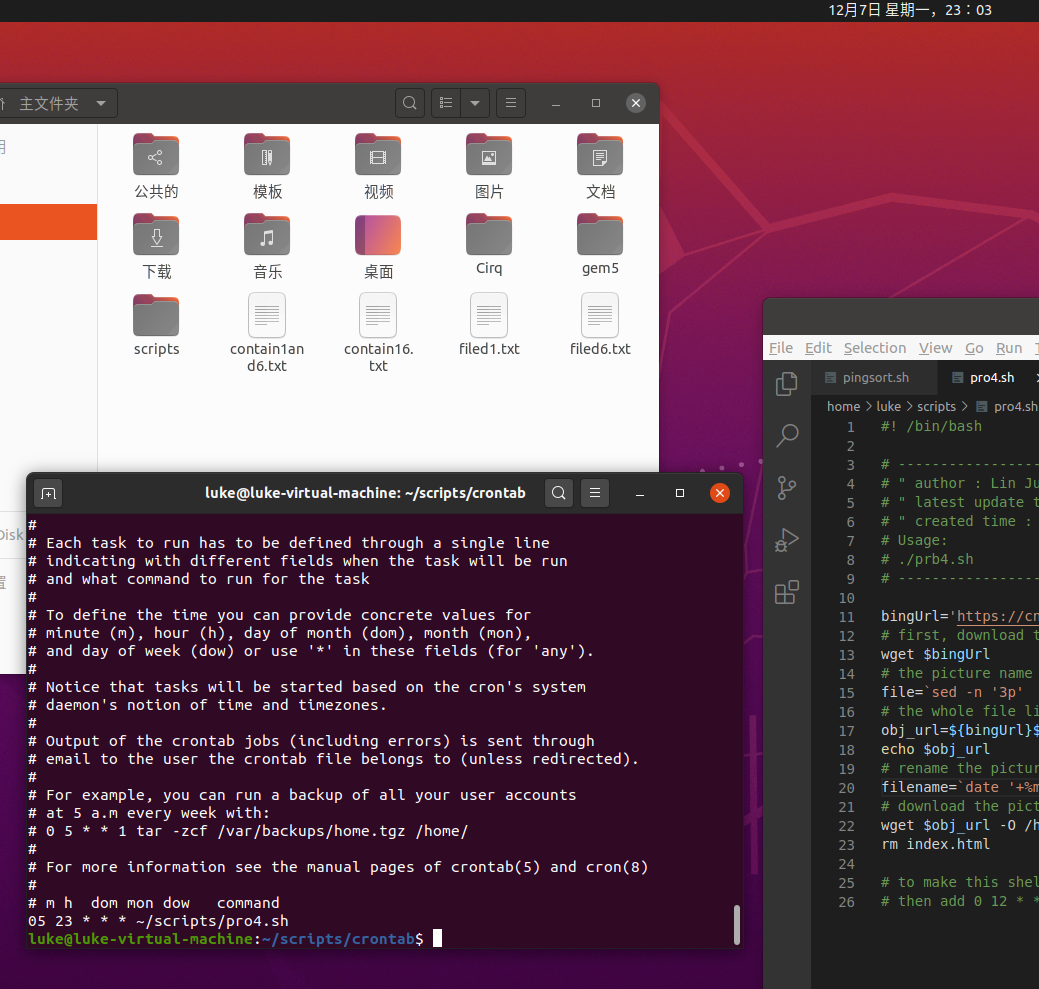
Crontab –l 查看

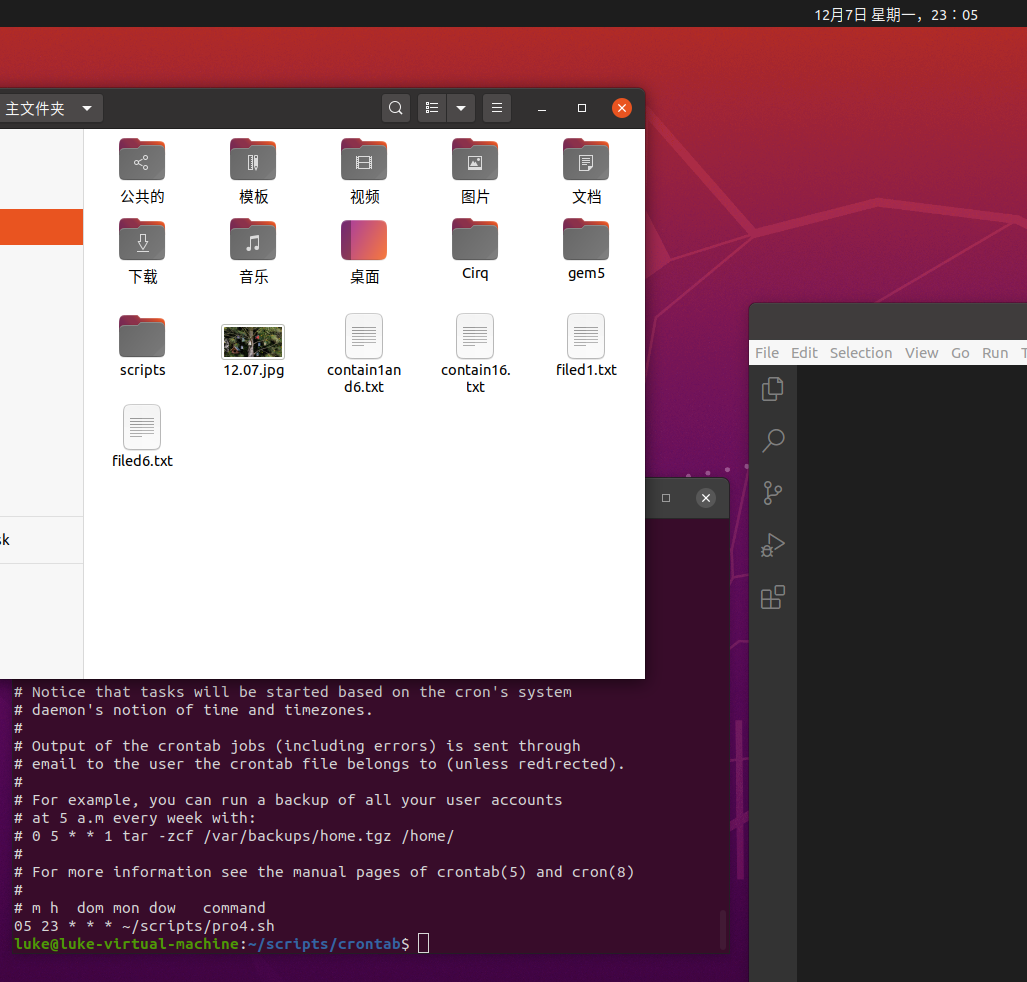
/sbin/service crond start

Failed to start crond.service: Unit crond.service not found.

A: 这是因为Ubuntu上的服务名称是cron不是crond。所以你的命令应该是：

sudo service cron start





自动下载

让她输出到txt来.

