# 网上资料

## 菜鸟网

[https://www.runoob.com/linux/](https://www.runoob.com/linux/linux-comm-grep.html)

## 博客园

https://www.cnblogs.com/

## b站shawn李双宏

https://space.bilibili.com/403317915?spm\_id\_from=333.788.b\_765f7570696e666f.1

# 常识

## 图形界面切换命令行界面

CentOS 8 图形界面如何切换到命令行界面

打开终端，输入init 3

输入密码 acwe3/8+

账号是admin 密码同上

登录的时候遇到login incorrect ，是由于小写键盘的num lock 的锁定状态不正确导致的，重新按两下就好了

## 按tab补全的时候会发出蜂鸣器的噪音

### 网上随便找的

找到/etc/inputrc的 #set bell-style none项，她是注释掉的。bell-style的取值范围是:none, visible, audible，想要把报警声去掉将/etc/inputrc中的set bell-style 的值改为 visible，并将注释去掉

输入 reboot 重启系统后即可

### 解决思路

尝试用chmod更改inptrc的权限，显示没有权限

尝试搜索后没有结果

尝试搜索vim权限不足时如何强制保存

输入 ：w !sudo tee %

确实保存上了，但是重启之后还是有声音和没改一样

## 我主观感受到的cat和vim的区别

cat不用进入到vim的那个界面，直接显示文件中的内容到界面上，下面可以继续敲命令

vim还需要退出，而退出这个步骤太麻烦

cat更像是把文件中的内容打印到屏幕上

vim则是使用vim这个软件打开这个文件

## 结合其他命令与grep组合

cat和grep命令

cat filename|grep e$

查询filename文件中以e结尾的所有行

## 简单的cat命令配合grep和直接grep的查询的结果一样

cat filename|grep e$

grep e$ filename

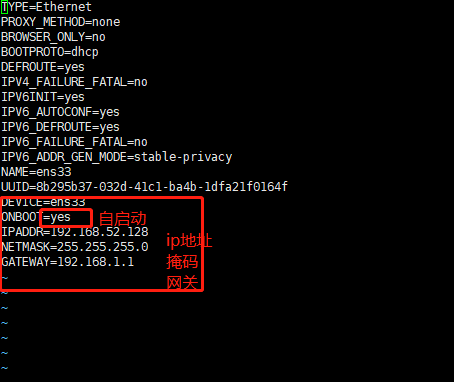
两个查询结果一致，应该是检索的逻辑不一致，再了解应该要去扣linux里面的命令的源码是具体怎么实现这个命令效果的了

## 虚拟机连不上网怎么办

一般都是虚拟机里面的配置文件中的自启动被关了，把no改成yes就行了，记得要用root用户登录才能更改，admin上不去

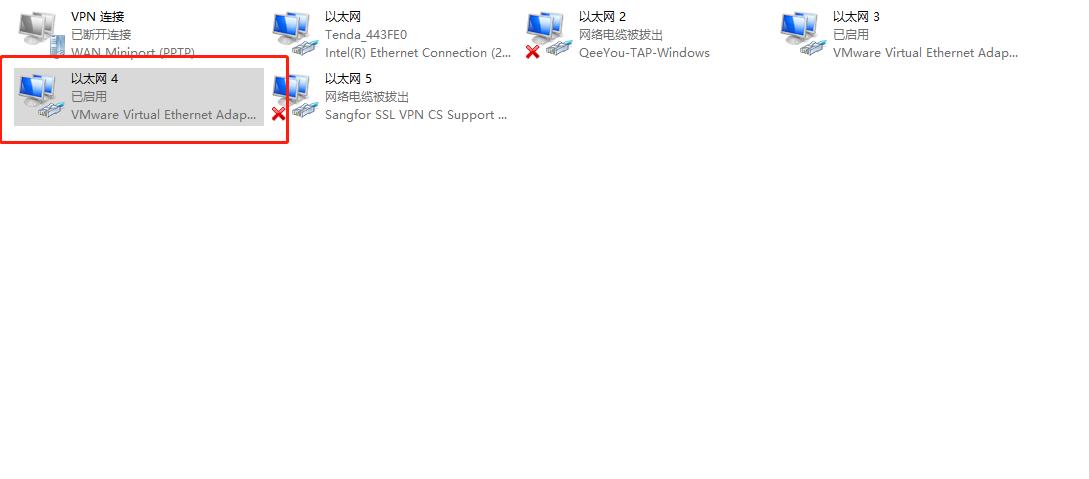
vi /etc/sysconfig/network-scripts/ifcfg-eth0

进去之后



如果没有ip地址那么就写上

我这里的虚拟机网络使用nat链接的，在主机的网络适配器里面vmware安装的时候是有一个用来nat方式连接的适配器，linux里面配置的ip要和这个适配器是同一个网段



# linux基础命令

## linux中翻页命令

用的centOS 8 & vmware虚拟机

Shift + PageUp 和 Shift + PageDown向上和向下翻页

比如用cat命令查询出来的内容过多，那么就应该使用该快捷键实现翻页

如果在xshell中的话直接PageUp PageDown 即可

## 快速将光标移动到行首或者行尾

ctrl + a 行首

ctrl + e 行尾

## 复制粘贴

ctrl + ins 复制

shift + ins 粘贴

## 剪切粘贴

ctrl + k 剪切光标到行尾

ctrl + u 剪切光标到行首

ctrl + y 将剪切的粘贴

## 移动光标

ctrl + f 往光标后移动一个字符

ctrl + b 往光标前移动一个字符

alt + f 向右移动到下一个空格位置

alt + b向右移动到下一个空格位置

ctrl + xx 会移动到当前行开头，再按一边会移动到之前所在的位置

如果在按一次之后在当前行前面插入字符，再按一次回到之前的位置，都是固定的

除了第一次会到行首以外，剩下的继续使用都是记录两次改变之后的位置

## 大小写转换

alt + u 将光标后至最近空格前所有字符变成大写，且光标移动到右最近空格处

alt + l 将光标后至最近空格前所有资助变成小写，且光标移动到左最近空格处

## 删除

ctrl + d 删除光标后一个字符 和delete一致

ctrl + h 删除光标前一个字符 和backspace一致

alt + d 删除当前光标以后的最近一个空格之前的所有字符

## 字符修正

alt + t 将光标前第一个空格处至第二个空格处所有字符与光标前至空格处字符进行替换

其实是将前面一个字符向后面搬运，使用前前面必须有空格或者.才可以使用

ctrl + t 将光标前第一个字符与第二个字符替换

使用前前面必须有一个字符才可以将前面的字符向后面搬运

ctrl + \_ 将最近的键盘输入取消输入

有点类似ctrl + z

## 剪切和粘贴

ctrl + w 剪切至前一个断点，长按就一直剪切到行首

ctrl + k 剪切光标后所有字符

ctrl + u 剪切光标前所有字符

ctrl + y 粘贴

## 命令历史记录控制

ctrl + p = up

ctrl + n = down

alt + r 在执行过一个命令之后，会将当前行字符与最近执行过的命令进行匹配复原

ctrl + r 会切换到命令历史记录的搜索栏中

比如之前输入过ll -a

那么按下 ctrl + r 后 输入 ll+空格，那么就会出现之前ll -a的命令

如果这个命令还是不符合的话，那么就再按一下ctrl + r 查看一下是否符合

如果符合，那么按下ctrl + o即可，和回车键效果一样

ctrl + g 就是推出这个搜索模式，不运行任何命令

**界面切换？**

alt + 方向键左（长按） 登出

# mv命令

## 给目录或者文件改名

mv 原名 新名

mv txt1 txt2

txt1就被改为txt2，更改目录名也是同理

# tree命令

## 打开当前目录下的树结构

tree ./

# vim命令

## vim复制粘贴

按v后通过光标可以选中文本，y复制p粘贴

按了y之后自动退出v的可视模式

## vim删除一行

esc后输入dd

# Grep命令

## 给查询出的结果标注颜色

--color

橘色的

全局命令，ls –color，ll –color 也可以

## 查询时不区分大小写

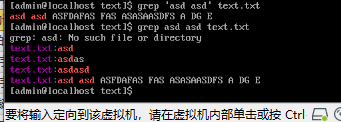
grep -i‘a’ filename

这样就会在filename文件中查询出‘A’和‘a’的那些行

## 查询的字符串包含空格需要用‘’括起来

grep‘a s d’ filename

### 不加上的话就会出现这样的效果



## 查询关键字的上下多行

grep –C5 ‘a’ filename

查询关键字a的上下五行

如果是开头和结尾，就只展示到开头和结尾，不会用空行表示

如果查询出了多个关键字，那么就会展示每个关键字的上下五行

## 查询以关键字开头的行

grep ^‘a’ filename

grep ‘^a’ filename

这两种方式都不影响查询结果

## 查询以关键字结尾的行

grep ‘a’$ filename

grep ‘a$’ filename

这两种方式都不影响查询结果

## 查询不以关键字为开头或结尾的行

找开头

grep –v ^‘key’ filename

找结尾

grep –v ‘key’$ filename

## 查询关键字之前或之后的行数

grep –B1 key filename

查询filename文件中关键字key前面的一行，查询结果中当然也包括要查询的内容

grep –A1 key filename

查询filename文件中关键字key后面的一行，查询结果中当然也包括要查询的内容

如果到了文件的顶部或者尾部，不会有多余的显示内容

## 配合ls命令查询当前目录下包含in的目录名

ls|grep in

## 配合ls命令查询当前目录下包含in的目录名，且显示权限

ls –l|grep in

-l是ls的参数，不是grep的

## 查询关键字所在的行,共有几行

grep –c key filename

菜鸟网上写的是列，实际是计算行数

## 查询关键字所在的行号

grep –n key filename

## 查询匹配字符所在的字节位置

grep –b key filename

就是从文档的左上角开始从零数，第一行第一个就是0回车也算一个字符，第二行就是这一行的字符加上一个回车字符，一共是多少，下一行第一个就是多少编号。

比如

0:acs

5:dv

a就是0，d就是5，就是一个计数器

## grep –d

把之前的grep -d命令又学习了一下，grep -d 下有三个参数，read,recurse,skip

-d 命令是用来读取目录的 ，在使用的时候一直显示 xx is directory 意思就是这是个目录

需要在命令的结尾加上 \* 就能检索这个目录下的所有文件

grep –d read key directory/\*

参数如果是recurse就是递归查询目录下的文件，也就是该目录下的所有目录下的所有文件都会被查询

grep –d recurse key directory/\*

skip网上说是跳过的意思

我使用的时候，区别好像就是skip加上了之后在不加上\*的时候不会出现xx is directory

# Shell

## 如何用一个脚本让服务器重启

### 这个脚本需要有如下操作

#### 杀掉进程

怎么使用shell脚本杀掉指定进程

#!/bin/bash

(/bin/ps -e|grep java)>>jincheng.log

pid=$(/bin/awk '{print $1}' jincheng.log)

/bin/kill -9 $pid

/bin/ps -ef|grep java

#!/bin/bash 用来声明有啥不知道就去/bin/bash目录下面去找

(/bin/ps -e|grep java)>>jincheng.log 查询出进程将字符串打印到日志文件，这里的ps –e比ps –ef显示的字符要少一点

pid=$(/bin/awk '{print $1}' jincheng.log)

/bin/kill -9 $pid

/bin/ps -ef|grep java

rm -rf ~/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/servers/\*

./Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/bin/startManagedWebLogic.sh ManagedServer\_2 http://10.5.57.4:7375 >weblogic.log 2>&1 &

#### 删除stage文件

rm -rf /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/servers/managedserver1/stage

#### 执行启动服务的命令

cd /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/bin

nohup sh /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/bin/startManagedWebLogic.sh managedserver1 http://10.0.19.1:7375 2>&1 &

#### 用tail打开日志文件

tail -f /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/bin/nohup.out

自动杀进程启动服务的脚本写完了，就是有2个地方耿耿于怀，我是将ps -ef|grep java 命令 显示的内容导出之后，用awk命令以空格分割，然后拿到的进程的pid，有没有办法直接能够获取到进程的pid，或者一条命令直接给进程删了第二个点就是，这个方法是生成了一个文本文件，我是读取这个文件中的ps -ef|grep java 的结果 获取的pid，我不想出现这么一个记录进程pid的日志文件，但是我没做到，目前知识储备只能干成这样好像

还有第三个点，就是我不能直接在其他目录执行 nohup 命令不知道为啥，我只能先cd进去才行

### 最后完成的结果

#!/bin/bash

(/bin/ps -e|grep java)>>jincheng.log

pid=$(/bin/awk '{print $1}' jincheng.log)

/bin/kill -9 $pid

/bin/ps -ef|grep java

rm -rf /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/servers/managedserver1/stage

cd /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/bin

nohup sh /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/bin/startManagedWebLogic.sh managedserver1 http://10.0.19.1:7375 2>&1 &

tail -f /home/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/xszz\_domain/bin/nohup.out

## 自动替换文件脚本

beitihuanwenjianmulu=./common

xiangmumulu=./202019

find $beitihuanwenjianmulu -type f|sed 's#.\*/##'>>yuanwenjianming.log

find $xiangmumulu -type f>>mubiaowenjianmulu.log

c=$(grep -c '' yuanwenjianming.log)

for ((d=1;d<=$c;d++))

do

a=$(cat yuanwenjianming.log|head -n$d|tail -n1)

b=$(grep -w $a mubiaowenjianmulu.log)

mv $beitihuanwenjianmulu/$a $b

ls -l|grep $b

echo $b

done

#要做的事情是：

#替换文件，被替换文件，要将替换文件覆盖到被替换文件

#被替换文件目录和项目目录，地址要写全，绝对路径可以，记得加上目录名

beitihuanwenjianmulu=./common

xiangmumulu=./202019

#在被替换文件的目录查询，type f 指的是一般文件，sed 's#.\*/##'是将文件进行了分割

#例如：-type f就是下面的结果

#./20210119/zzxt/login.jsp

#加上sed 's#.\*/##'后就是下面的结果

#login.jsp

#这条命令就是将指定目录的所有文件名一行一行的放到日志文件中

find $beitihuanwenjianmulu -type f|sed 's#.\*/##'>>yuanwenjianming.log

#将指定目录的一般文件及路径包括文件名和后缀放到日志文件中

find $xiangmumulu -type f>>mubiaowenjianmulu.log

#查询yuanwenjianming.log日志文件中有多少行，也就是有多少个文件需要被替换，作为计#数器，用来限制循环次数

c=$(grep -c '' yuanwenjianming.log)

#执行循环，用变量d和c来控制循环次数

for ((d=1;d<=$c;d++))

do

#将被替换文件日志中的第一行，也就是需要被替换的第一个文件放到a变量中里面的$d用

#来控制，head –n$d 是用来显示前$d行，tail –n1是用来显示最后一行，合在一起就是将指

#定行数的字符显示出来

a=$(cat yuanwenjianming.log|head -n$d|tail -n1)

#-w参数是完全匹配的意思

#问题：报错目标目录的日志文件中，有包含查询文件名的其他文件名

#例子：abc，查询的是abc，还有其他的文件名类似abcasd

#在查询abc的时候，不仅是abc的路径查询出来了，abcasd的路径也查询出来了

#导致mv命令无法正确执行

#mv命令执行的是：mv abc abc abcasd 肯定是报错了。

#处理办法

#grep 加上-w参数

#grep -w

b=$(grep -w $a mubiaowenjianmulu.log)

#执行替换

mv $beitihuanwenjianmulu/$a $b

#展示替换完成后文件的权限和时间

ls -l|grep $b

#展示替换完成后的文件路径

echo $b

done

## 如何将命令赋值给变量

两种方法

### 1.

用` `,（尖号）把命令括起来，然后赋值给变量

dir=`pwd`，这个尖号是esc下面的那个，不是引号

### 2.

变量=$(pwd)

dir=$(pwd)

将命令用小括号括起来，前面加上$然后赋值给变量

# Awk命令

## 输出第一个冒号前的所有字符

### 样本

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

### 命令

awk 'BEGIN{ FS=":" } {print $1}' filename

这里的$后面的1就是指的第一个冒号前的区间的字符串，如果这里的数字是2，那么就指的是第一个冒号和第二个冒号之间的字符串，以此类推

begin要大写，否则就是以空格去分割而不是：冒号

通过此命令可以将字符串分割

## 分割空格或者tab

Awk ‘{print $1} filename.log

将每一行的第一个空格前第一项输出

# chmod命令

更改文件或者目录的权限

chmod –R 777 filename

# chown命令

更改文件或者目录的属主

chown –R group filename

# 报错

输入grep –B key filename后出现invalid context length argument

意思是参数无效

也就是说，-B后没加参数

-B是用来查询关键字上面多少行的命令

# 没解决的问题

## -d没看懂

-d <动作> 或 --directories=<动作> : 当指定要查找的是目录而非文件时，必须使用这项参数，否则grep指令将回报信息并停止动作。