

ubuntu

2019年5月1日 22:05

命令行操作

ls 当前位置文件

pwd 当前位置

cd Desktop 切换到桌面

touch 1.txt创建文件

mkdir 1 创建文件夹

clear 清屏

command [-options] [partmeterl]

例如: ls / 根目录 ls -a选项显示全部 ls -l文件信息

ls -l -h信息 ls -a -l -h

中l列表h大小a全部可以放在一起ls /alh

ls --help全部命令

man ls 用q退出 f翻页d上翻页 manual-手册

man中各个section意义如下:

1. Standard commands (标准命令)
2. System calls (系统调用, 如open,write)
3. Library functions (库函数, 如printf,fopen)
4. Special devices (设备文件的说明, /dev下各种设备)
5. File formats (文件格式, 如passwd)
6. Games and toys (游戏和娱乐)
7. Miscellaneous (杂项、惯例与协定等, 例如Linux档案系统、网络协定、ASCII 码; environ全局变量)
8. Administrative Commands (管理员命令, 如ifconfig)

用于查找命令

TAB自动补全文件名字

gedit 1.txt 呼出1进行编辑

vi 1.txt 编辑文本

cat 1.txt 查看1内容

history 历史命令

!2074 历史命令里进行第2074行代码操作

rm 1.txt 删除1.txt

通配符 例如 ls 2 列出2开头文件

?通配符 单个符号

ls 1[23]4.txt 呼出124.txt 134.txt

ls /alh > 1.txt 信息写入1.txt 重定向

ls /alh >> 1.txt 信息追加1.txt末尾 重定向

more 1.txt 分页查看 f翻页 d上翻 q退出

ls -alh /bin | more 竖杠为管道 用分页方式查看bin内的文件信息

ls;ls -alh 两个命令用分号隔开 同时命令

cd ./A 跳到当天路径下的A文件夹内

cd .. 跳到当前路径的上一个路径

cd ../.. 跳到上上一个路径 发现cd .;cd ..也可以...

cd - 回到上一次操作路径 只能操作一次

cd~ 加目录 进入home目录

ctrl键盘+c 清除当前命令

rm *.txt 删除所有txt

tree 目录树

mkdir A/B/C/D/E -p 选项之前目录文件夹 创建

rm B -r 删除B文件夹

mv 1-softlink.txg 1-softlink.txt 把名字改成名字

ln -s 1.txt 1-softlink.txt 设置快捷方式软链接

ln 1.txt 1-softlink.txt 设置快捷方式硬链接 即别名

grep "ntfs" xxx.txt 搜索文件内容下包括引号内容的文件名字

-v 显示不包含匹配文本的所有行 -n显示匹配行及行号 -i忽略大小写

mv 111.txt laowang/ 111.txt文件移动到laowang/路径

mv laowang/ gebilaowang 文件夹重命名

cp 2.txt A 复制2.txt并粘贴到A文件夹里

-a	该选项通常在复制目录时使用，它保留链接、文件属性，并递归地复制目录，简单而言，保持文件原有属性。
-f	已经存在的目标文件而不提示
-i	交互式复制，在覆盖目标文件之前将给出提示要求用户确认
-r	若给出的源文件是目录文件，则cp将递归复制该目录下的所有子目录和文件，目标文件必须为一个目录名。
-v	显示拷贝进度

cd A gebilaowang/ -r 将文件夹拷贝到路径

find查找文件

find ./ -name test.sh	查找当前目录下所有名为test.sh的文件
find ./ -name '*.sh'	查找当前目录下所有后缀为.sh的文件
find ./ -name "[A-Z]*"	查找当前目录下所有以大写字母开头的文件
find /tmp -size 2M	查找在/tmp 目录下等于2M的文件
find /tmp -size +2M	查找在/tmp 目录下大于2M的文件
find /tmp -size -2M	查找在/tmp 目录下小于2M的文件
find ./ -size +4k -size -5M	查找当前目录下大于4k，小于5M的文件
find ./ -perm 777	查找当前目录下权限为 777 的文件或目录

sudo find / -name "*name*" 啥啥啥 给权限

tar -cvf test.tar *.py 所有的py文件打包成tar文件

-c	生成档案文件，创建打包文件
-v	列出归档解档的详细过程，显示进度
-f	指定档案文件名称，f后面一定是.tar文件，所以必须放选项最后
-t	列出档案中包含的文件
-x	解开档案文件

tar -xvf text.tar 在当前路径解压tar压缩包 压缩包压缩大小不变

tar -zcvf xxx.tar.gz *.py 以gz压缩

ll 相当于 ls -la 显示当前路径文本信息

tar -zxvf xxx.tar.gz 当前路径解压gz压缩包

tar -jcvf yyy.tar.bz2 *.py 第二种压缩方式

tar -jxvf yyy.tar.bz2 解压第二种压缩方式

zip zzz.zip *.py 以zip压缩方式

unzip zzz.zip解压zip

tar -zxvf text.tar.gz -C dongge/ 解压到指定路径-C

unzip -d ./text myzip.zip 解压到指定路径 -d

which ls ???????

which ps ???????

cal 显示日历

cal -y 2008 显示2008日历

date 当前时间

date > test.txt存在文本内 cal同理

date "+%Y===%m===%d" 显示方式 Y年M月D日

<2>显示或设置时间：date

设置时间格式（需要管理员权限）：

```
date [MMDDhhmm[[CC]YY][.ss]] +format
```

CC为年前两位yy为年的后两位，前两位的mm为月，后两位的mm为分钟，dd为天，hh为小时，ss为秒。

如：date 010203042016.55。

显示时间格式（date '+%y,%m,%d,%H,%M,%S'）：

format格式	含义
%Y, %y	年
%m	月
%d	日
%H	时
%S	秒

<3>查看进程信息：ps

进程是一个具有一定独立功能的程序，它是操作系统动态执行的基本单元。

ps命令可以查看进程的详细状况，常用选项(选项可以不加“-”)如下：

选项	含义
-a	显示终端上的所有进程，包括其他用户的进程
-u	显示进程的详细状态
-x	显示没有控制终端的进程
-w	显示加宽，以便显示更多的信息
-r	只显示正在运行的进程

相当于任务管理器

ps -aux 全信息 q退出

top 按照使用资源大的运行进程

htop 更详细电脑信息 运行情况

kill 指定名字后面的数字

```
python 9822 97.4 0.3 67612 3936 pts/1 R+ 13:21 110:42 vi
python 9833 0.0 2.8 999332 27980 ? Ssl 13:22 0:05 /o
python 9846 0.0 0.7 1390484 7628 ? Sl 13:22 0:03 /o
python 9881 0.0 0.4 30688 4388 pts/4 Ss 13:22 0:00 ba
root 15711 0.0 0.0 0 0 ? S< 14:16 0:00 [k
root 15713 0.0 0.0 0 0 ? S< 14:16 0:00 [h
root 15714 0.0 0.0 0 0 ? S< 14:16 0:00 [h
root 15715 0.0 0.0 0 0 ? S< 14:16 0:00 [k
root 15731 0.0 0.2 16132 2464 ? S 14:16 0:00 /s
root 16037 0.0 0.3 16132 3244 ? S 14:17 0:00 /s
root 16325 0.0 0.0 0 0 ? S 14:34 0:00 [k
root 16565 0.0 0.0 0 0 ? S 14:54 0:00 [k
root 16642 0.0 0.0 0 0 ? S 15:02 0:00 [k
root 16711 0.0 0.0 0 0 ? S 15:10 0:00 [k
python 16736 0.0 0.3 44432 3136 pts/4 R+ 15:14 0:00 ps
python@ubuntu:~/Desktop/python06期/test2/test3/test4$ kill
t4$ kill 9822
```

kill -9 9822 强制关闭

<6>关机重启: ~~reboot~~、shutdown、init

命令	含义
reboot	重新启动操作系统
shutdown -r now	重新启动操作系统, shutdown会给别的用户提示
shutdown -h now	立刻关机, 其中now相当于时间为0的状态
shutdown -h 20:25	系统在今天的20:25 会关机
shutdown -h +10	系统再过十分钟后自动关机
init 0	关机
init 6	重启

df -h 显示硬盘空间

du -h显示每个路径的文件夹占用大小

ifconfig 查看当前电脑信息 ens33 ens38 lo网卡名

sudo ifconfig ens33 172.16.7.139修改ip

ping 172,16,7,1 查网络与另一个ip的延迟

ifconfig | grep 172 用于查找ip

useradd shuaige -m建立账户 并且创建文件夹 加目录

useradd shuaig 只建立账户

cat /etc/passwd 看末尾确认是否创建用户 用户名: 号 加目录

sudo useradd shuaige -m 开权限

ls /home/ 查看是否存在账户

su shuaige 切换用户到shuaige

sudo passwd shuaige 设置密码

whoami 当前账户是?

exit 退出用户/退出终端

ssh python@172.16.7.139 远程登陆python的@ip地址 登录

alt+1 切换到第一终端 alt+2切换到第二终端

who 查看登陆该账户位置

userdel shuaige 删除账户 但加目录还存在

sudo userdel -r gebilaowang 删除并且删除加目录

rm / -rf 删库

su - xxxx 切换到账户同时 起始为根目录

sudo -s 切到超级管理员 root 设置权限可以不需要"sudo"

```
python@ubuntu:~/Desktop/python06期$ ll
总用量 12
drwxr-xr-x 3 python python 4096 3月 30 14:43 ./
drwxr-xr-x 6 python python 4096 3月 30 15:58 ../
-rw-rw-r-- 1 python python 0 3月 30 14:37 1.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 2.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 3.py
drwxrwxr-x 3 python python 4096 3月 30 14:44 test2/
python@ubuntu:~/Desktop/python06期$
```


cat /etc/group查看当前所有用户的组

groupadd YYY 创建一个组

groupdel YYY 删除组

groupmod 之后按两次TAB显示所有组名

```
xxxx@ubuntu:/home/python/Desktop/python06期$ cat /etc/group | grep sudo
sudo:x:27:python
xxxx@ubuntu:/home/python/Desktop/python06期$ cat /etc/group | grep adm
adm:x:4:syslog,python
lpadmin:x:113:python
xxxx@ubuntu:/home/python/Desktop/python06期$
```

可以使用超级管理员权限 root的账户

<12>为创建的普通用户添加sudo权限

新创建的用户，默认不能sudo，需要进行一下操作

```
sudo usermod -a -G adm 用户名
```

```
sudo usermod -a -G sudo 用户名
```

<13>usermod -g 与 -G的区别

-g 用来制定这个用户默认的用户组

-G 一般配合'-a'来完成向其它组添加

```
root@ubuntu:/home/python/Desktop/python06期# ll
总用量 12
drwxr-xr-x 3 python python 4096 3月 30 14:43 ./
drwxr-xr-x 6 python python 4096 3月 30 15:58 ../
-rw-rw-r-- 1 python python 0 3月 30 14:37 1.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 2.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 3.py
drwxrwxr-x 3 python python 4096 3月 30 14:44 test2/
root@ubuntu:/home/python/Desktop/python06期# groupadd YYY
root@ubuntu:/home/python/Desktop/python06期# chgrp YYY 1.py
root@ubuntu:/home/python/Desktop/python06期# ll
总用量 12
drwxr-xr-x 3 python python 4096 3月 30 14:43 ./
drwxr-xr-x 6 python python 4096 3月 30 15:58 ../
-rw-rw-r-- 1 python YYY 0 3月 30 14:37 1.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 2.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 3.py
drwxrwxr-x 3 python python 4096 3月 30 14:44 test2/
```

chgrp 修改文件所属组

```

root@ubuntu:/home/python/Desktop/python06期# ll
总用量 12
drwxr-xr-x 3 python python 4096 3月 30 14:43 ./
drwxr-xr-x 6 python python 4096 3月 30 15:58 ../
-rw-rw-r-- 1 python YYY 0 3月 30 14:37 1.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 2.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 3.py
drwxrwxr-x 3 python python 4096 3月 30 14:44 test2/
root@ubuntu:/home/python/Desktop/python06期# chown xxxx 1.py
root@ubuntu:/home/python/Desktop/python06期# ll
总用量 12
drwxr-xr-x 3 python python 4096 3月 30 14:43 ./
drwxr-xr-x 6 python python 4096 3月 30 15:58 ../
-rw-rw-r-- 1 xxxx YYY 0 3月 30 14:37 1.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 2.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 3.py
drwxrwxr-x 3 python python 4096 3月 30 14:44 test2/

```

chown 修改文件所有者

权限说明

```

root@ubuntu:/home/python/Desktop/python06期# ll
总用量 12
drwxr-xr-x 3 python python 4096 3月 30 14:43 ./
drwxr-xr-x 6 python python 4096 3月 30 15:58 ../
-rw-rw-r-- 1 xxxx YYY 0 3月 30 14:37 1.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 2.py
-rw-rw-r-- 1 python python 0 3月 30 14:37 3.py
drwxrwxr-x 3 python python 4096 3月 30 14:44 test2/
root@ubuntu:/home/python/Desktop/python06期#

```

同组者的权限
文件拥有者的权限
其他人的权限

开头的"-"表示普通文件 "d"文件夹

r读 w写 x可执行

u文件拥有者 g同组 o其他人

chmod u=rwx 2.py 把目标文件的u权限更改rwx

chmod u=g,o= 2.py 可以用逗号隔开 可以修改无权限

```

r——>4
w——>2
x——>1

```

1
2
3
4
5
6
7

chmod 137 2.py 相当于u=x g=xw,o=rwx 数字法修改权限

gedit和

跨平台编辑器sublime 可以分平台 我选择 vim (滑稽)



vi操作

```

Vi有三种基本工作模式：
+ 命令模式
+ 文本输入模式
+ 末行模式。

```

```

E325: 注意
发现交换文件 ".4.py.swp"
所有者: python    日期: Thu Mar 30 17:26:28 2017
文件名: ~python/Desktop/python06期/4.py
修改过: 是
用户名: python    主机名: ubuntu
进程 ID: 22840
正在打开文件 "4.py"
日期: Thu Mar 30 17:22:56 2017

(1) Another program may be editing the same file. If this is the case,
be careful not to end up with two different instances of the same
file when making changes. Quit, or continue with caution.
(2) An edit session for this file crashed.
    如果是这样, 请用 ":recover" 或 "vim -r 4.py"
    恢复修改的内容 (请见 ":help recovery")。
    如果你已经进行了恢复, 请删除交换文件 ".4.py.swp"
    以避免再看到此消息。

交换文件 ".4.py.swp" 已存在!
以只读方式打开([O]), 直接编辑((E)), 恢复((R)), 删除交换文件((D)), 退出((Q)), 中止((A)):

```

左下角显示"缓冲区无内容"按i进入编辑模式

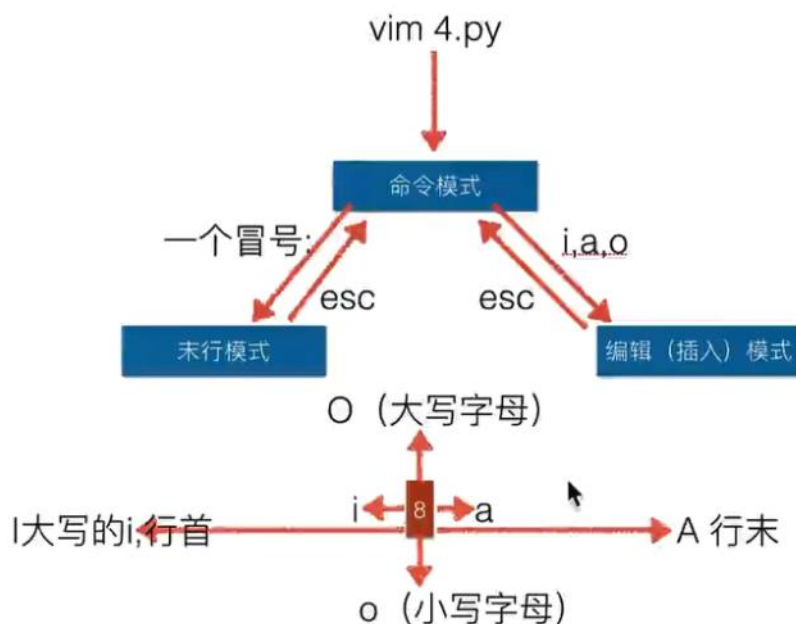
shift+zz:相当于wq

末行模式

w: 保存

q: 退出

wq: 保存并且推出



vi命令

yy:复制 光标所在的这一行
4yy:复制 光标所在行开始向下的4行

p: 粘贴

dd:剪切 光标所在的这一行
2dd:剪切 光标所在行 向下 2行
D:从当前的光标开始剪切,一直到行末

h左 j下 k上 l右

H:当前屏幕的上方
M:当前屏幕的中间
L:当前屏幕的下方

ctrl+f--->向下翻一页代码
ctrl+b--->向上翻一页代码

ctrl+d--->向下翻半页代码
ctrl+u--->向上翻半页代码

20G:快速的定位到第20行代码
G:快速的回到 整个代码的最后一行
gg:快速回到 整个代码的第1行

d0:从当前的光标开始剪切,一直到行首

订:20G:快速定位到20行

w:向后跳一个单词的长度,即调到下一个单词的开始出
b:向前跳一个单词的长度,即调到上一个单词的开始出

u:撤销刚刚的操作
ctrl+r:反撤销

选中一片代码

v:
V:

>>:向右移动代码
<<:向左移动代码

.:重复执行上一次的命令

可视模式:

v: 按字符移动,选中文本

V: 按行移动,选中文本可视模式可以配合 d, y, >>, << 实现对文本块的删除,复制,左右移动

r: 替换一个字符
R: 替换光标以及后面的字符

查找命令:

```
/: str查找  
n: 下一个  
N: 上一个
```

替换命令:

把abc全部替换成123

末行模式下, 将当前文件中的所有abc替换成123

```
:%s/abc/123/g
```



末行模式下, 将第一行至第10行之间的abc替换成123

```
:1, 10s/abc/123/g
```

附录 vim分屏操作

