

## #Basic simple commands are not very useful -->linux基本命令

date #显示时间

cal #显示本月的日历

cal -y #显示本年的日历

history #显示所用过的历史命令

history -c #清空历史命令

pwd #显示当前路径

cat log.txt #查看文件内容。-->文件内容过大，只能显示最后一页。

#-n 或 -number :由1开始对所有输出行的行数编号。

#-b 或 -number-nonblank :对于空白行不编号。

#-s 或 -squeeze-blank 将连续两行及以上的空白变为一行空白

cat -n a.txt b.txt > c.txt #把a.txt和b.txt中的内容加上行号后输入c.txt中。

cat /dev/null > a.txt #清空a.txt文件内容。

#more log.txt 废弃

#翻页查看文件内容。-->只能向后翻页[enter下一行(向下键),q退出;翻页键]

less log.txt #翻页查看文件内容。-->前后翻页[enter下一行(向下键),向上键上一行,q退出;翻页键]

mkdir test1 test2 test3 #创建多个目录

mkdir -p test4/test5/test6 #递归创建多个空目录。

#-p 参数表示，父目录不存在时，先创建父目录。

#rmdir test 废弃,删除空目录

#rmdir -p test4/test5/test6 #-p 递归删除空目录，向上递归。

rm #删除文件或目录

rm -rf 1.txt #不提示删除1.txt文件

rm -rf test/ #不提示删除test/目录

#-r 递归删除目录

#-f 不提示用户

#-i 删除文件或目录时提示用户

touch 1.txt #新建普通文件

cp 用于文件或目录的复制

cp 1.txt 2.txt #文件复制

cp -f 1.txt 2.txt #不提示覆盖已存在文件。会发现-f参数不起作用: /usr/bin/cp -f 1.txt 2.txt 可以

cp -r 1.txt 2.txt #目录复制

cp -rf t1/ t2/ #不提示覆盖已存在的目录。-f参数不起作用: /usr/bin/cp -rf t1/ t2/ 可以

#-r:递归形式复制

#-f: 不提示 直接覆盖已存在文件。

alias #查看所有系统别名  
unalias cp #取消别名  
alias cp='cp -i' #设置别名

mv 重命名 移动

mv a.txt b.txt #将a.txt重命名为b.txt  
mv a.txt test/ #将a.txt移动到test/目录下  
mv a.txt test/aa.txt #将a.txt移动到test/目录下,且重命名为aa.txt

//目标不存在:移动且重命名。 目标存在:只移动

mv test1/ test2/ #将test1目录重命名为test2

mv test1/ test2/test1/ #如果test2/test1目录不存在,则是移动到test2/下面作为test1目录。如果test2/test1目录存在,则是移动到test2/test1/下面作为test1目录。

mv test1/ test2/test3/ #如果test3不存在,则是将test1移动到test2下面作为test3。如果test3存在,则是移动到test3下面作为test1。

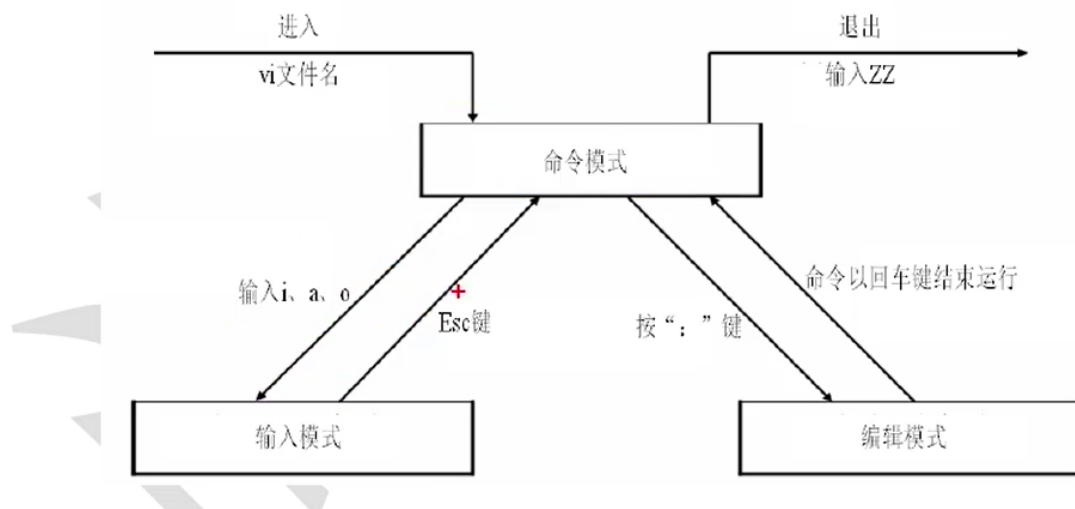
cd

. 表示当前目录  
.. 表示当前目录的父目录  
~ 表示当前用户的家目录  
/ 表示根目录  
- 上一次的目录

cd ../ntp #表示切换到当前目录的父目录的子目录ntp

## vim编辑器：

vim 工作在三种模式之下：



## 命令模式-输入模式-编辑模式:

命令模式:

进入输入模式:

```
a #在光标所在字符后插入(追加)
A #在光标所在字符的行尾插入(行尾追加)
i #在光标所在字符前插入(前插入)
I #在光标所在字符的行首插入(行首插入)
o #在光标下插入新行,向下插入一行(向下打开新行)
O #(向上打开新行)
```

输入模式: 主要用于文本编辑,类似windows的记事本

末行模式(编辑模式): 执行常用操作

```
:w #保存(不退出)
:w abc #另存为abc文件
:q #退出(不保存)。在文件没有修改的条件下
:wq #保存退出。等于:x。等于 ZZ(大)
:!  
:q! #强制退出(不保存)。主要用于修改文件后,不保存数据退出。  
:wq! #强制保存退出。当文件的所有者或者root用户,对文件没有写权限的  
时候,强制写入数据。
```

## .vimrc文件常用设置

vim /etc/vimrc 对整个系统生效

~/.vimrc 只对单个用户生效

```
vim ~/.bashrc  
set nu #显示行号  
alias vi='vim' #别名
```

## 末行模式(编辑模式), 临时常用设置:

```
:set nu #设置行号  
:set nonu #取消行号  
:noh #取消高亮  
  
:set list #显示隐藏字符。tab键用^I显示,回车符用$显示,^M$表示windows下的回车符  
(shell脚本无法执行)。作用等同于cat -A test.txt  
:set nolist  
#windows下创建一个shell脚本或者普通文件,上传到linux上,使用cat -A test.txt查看  
回车符用的是什么。:set list看不出来。  
#使用dos2linux进行转换  
yum install dos2unix  
dos2unix test.sh #将windows下的回车符转换为linux下的回车符  
unix2dos test.sh #将linux转为windows
```

```
echo $? #显示0表示成功。

#-----不常用-----
:syntax on #开启语法高亮显示
:syntax off

:set ruler #开启右下角的状态栏
:set noruler
```

## 1.光标定位:

```
gg #文档首行
G #文档末行
3gg 或 3G #定位到第3行
:3 #定位到第3行
w #移动一个单词
W #跳至下一个单词的词首。 7W 移动7个单词
e #跳至当前或下一个单词的词尾
b #跳至当前或下一个单词的词尾。

0 #绝对行首。是0不是shift+0
^ #行首的第一个非空白字符
$ #绝对行尾

h 左 j 下
k 上 l 右
#7h 移动7个字符

vim +40 t1.txt #打开文件, 光标定位到40行
```

## 2.复制粘贴

```
yy #复制一行 ---> 4yy 复制4行
y #只复制单个字符
p #粘贴在光标所在下一行
P #粘贴在光标所在上一行

dd #删除一整行
---> 4dd #删除4行
:3,5d #删除第三行到第五行的内容

x #先后删除一个字符。如果后面删完了, 则向前删除。[一行内]
D #从光标处删除到行尾。
```

```
#[剪贴：dd 。如果dd之后，不用就是删除；p(粘贴)一下，就是剪贴。]
```

```
dG #从光标所在行删除到文件尾部,光标放到文件头部，清空整篇文档
```

```
#[windows:ctrl+z撤销，ctrl+y反撤销]
```

## 2.删除修改操作：

```
r #替换单个字符
```

```
r #替换修改一个字符。-->光标定位，按r键，随后输入想要替换后的字符。
```

```
R #进入替换模式，替换一整行，替换完成按<ESC>键结束
```

```
x #删除光标所在位置的字符
```

```
3x #删除3个字母
```

```
D #从光标所在处删除到行尾
```

```
dd #删除一行
```

```
3dd #从光标所在行开始删除3行
```

```
u #撤销
```

```
ctrl + r #还原撤销。
```

vim 常用操作:

### 1.在文档中查找关键字

```
/aa #从光标所在行向下查找aa关键字
```

```
n #下一个，
```

```
N #上一个
```

```
:noh #取消高亮
```

### 2.替换 :替换行数范围s/old/new/g

```
#替换行数范围：1,4行；%整个文本
```

```
#最后一个g表示每行要替换的次数，默认只替换一次，g表示每行全部替换。
```

```
:1,4s/p/-/ #表示替换1，4行的p为-，每行只替换一次。
```

```
:1,4s/p/-/g #替换1,4行的p为-，每行全部替换。
```

```
:%s/p/-/g
```

```
:1,10s/^/#/ #第一行到第十行进行#号注释
```

```
:1,10s/#// #取消注释
```

```
:1,10s/^\/\// #进行//注释.转移字符\使得/取消了特殊含义
```

```
:1,10s\/\// #取消//注释
```

vim使用技巧:

## 1.导入其他文件内容

```
#导入passwd文件内容到当前文件中
vim 2.txt
:r /etc/passwd
```

## 2.在vim中执行系统命令

```
vim 2.txt
:!!ls /usr/local/
```

## 3.导入命令执行结果

```
vim 2.txt
:r !ls /usr/local/
```

## 4.设定快捷键. :map 快捷键 执行快捷键的命令

```
#按住ctrl+p进行行首#号注释
vim 2.sh
:map ^p I#<ESC> #^p是ctrl+v+p进行输入的
#按住ctrl+o取消行首#号注释

vim 2.sh
:map ^o ^x
```

## 5.字符替换。:ab 源自符 替换为字符

```
#当输入mainf时，替换为int main(){ return 0;}
vim main.c
:ab mainf int main(){ return 0;}
```

## 6.多文件打开/分屏显示.

```
#上下打开。ctrl+w,上下键移动光标到窗口
vim -o 1.txt 2.txt

#左右打开。ctrl+w,左右方向键移动光标到窗口。
vim -O 1.txt 2.txt
```

## 7.比较两个文件的内容

```
#diff t1.txt t2.txt 废弃  
vimdiff t1.txt t2.txt #不同的内容会红色显示
```

## 8.批量/多行注释

```
#注释:  
ctrl+v ---> 向下选中 ---> 按I(大写的i) --->插入注释符(比如://) ---> 按esc  
(全部注释了)  
  
#取消注释  
ctrl+v ---> 向下选中 --->按d (删除全部注释了)
```

---

### 4.[演示]替换

:1,3 s/t1/t2 #把第一行到第三行中,每行出现的第一个t1替换为t2

:1,3 s/t1/t2/g #把一行到第三行中,每行出现的所有的t1替换为t2

:% s/t1/t2 #把整个文件中, 每一行出现的第一个t1替换为t2

:% s/t1/t2/g #把整个文件中,每一行出现的所有的t1替换为t2

mount命令: