

Linux 常用命令

一、Linux的历史

(一)、Linux 诞生

主要基于Unix操作系统开发，系统诞生于 1991年10月5日。

(二)、Linux 特性

- 1.万物皆目录- Linux核心理念
- 2.Linux是内核不是操作系统。CentOS、Ubuntu、Fedora等才是操作系统的名字，它们都使用Linux内核。
- 3.完全免费。意味任何人都能通过网络或其它途径免费获得源代码；这也是Linux愈加强大的重要原因。
- 4.多用户、多任务。各个用户对于自己的文件设备有自己特殊的权利，保证了各用户之间互不影响。
- 5.支持多平台。Linux 内核90% C语言和10%汇编语言；可运行在多种硬件上，包括配置最小的树莓派。

二、Linux常用命令

Linux内的操作系统严格区分大小写！严格区分大小写！严格区分大小写！

(一)、目录

1.cd - 切换目录

```
# cd [PATH]
cd /ect/nginx # 切换到目录 /etc/nginx
cd ./conf     # 切换到当前目录下的 conf 目录中 "." 表示当前目录
cd ../nginx   # 切换到上层目录的 nginx 目录中 ".." 表示上层目录
```

2.pwd - 查看当前完整目录

```
# pwd [option]
pwd
/root

pwd -P # 显示目录实际路径 非软链 路径 (参考: ln 命令)

drwxr-xr-x   6 root root  4096 Jun 27 14:29 .
drwxr-xr-x. 92 root root  4096 Jun 27 14:41 ..
drwxr-xr-x   2 root root  4096 Jun 20 11:25 conf
drwxr-xr-x   2 root root  4096 Jun 27 14:08 conf.d
drwxr-xr-x   2 root root  4096 Jun 19 18:04 extra
-rw-r--r--   1 root root 13804 Jun 14 17:56 httpd.conf
lrwxrwxrwx   1 root root    19 Feb  3  2015 logs -> ../../var/log/httpd
-rw-r--r--   1 root root 12958 Feb  3  2015 magic
-rw-r--r--   1 root root 49815 Feb  3  2015 mime.types
lrwxrwxrwx   1 root root    29 Feb  3  2015 modules ->
../../usr/lib64/httpd/modules
drwxr-xr-x   3 root root  4096 Feb  3  2015 original
lrwxrwxrwx   1 root root    19 Feb  3  2015 run -> ../../var/run/httpd

[root@www httpd]# cd logs/
[root@www logs]# pwd
/etc/httpd/logs
[root@www logs]# pwd -P
/var/log/httpd
```

3.ls - 查看文件于目录的相关属性

```
# ls 相关参数
# ls [option]
# -l : 列出长数据串, 包含文件的属性与权限数据等
# -a : 列出全部的文件, 连同隐藏文件 (开头为.的文件) 一起列出来 (常用)
# -d : 仅列出目录本身, 而不是列出目录的文件数据
# -h : 将文件容量以较易读的方式 (GB, kB等) 列出来
# -R : 连同子目录的内容一起列出 (递归列出), 等于该目录下的所有文件都会显示出来
# 以上参数均可组合使用
ls -la
```

4.grep - 筛选列表结果

```
# grep 相关参数
-a : 将binary文件以text文件的方式查找数据
-c : 计算找到‘查找字符串’的次数
-i : 忽略大小写的区别，即把大小写视为相同
-v : 反向选择，即显示出没有‘查找字符串’内容的那一行
# 例如：
# 取出文件/etc/man.config中包含MANPATH的行，并把找到的关键字加上颜色
grep --color=auto 'MANPATH' /etc/man.config
# 把ls -l的输出中包含字母file（不区分大小写）的内容输出
ls -l | grep -i file
```

5.find - 文件及文件夹查找

```
# find [PATH] [option] [action]
# 与时间有关的参数：
-mtime n : n为数字，意思为在n天之前的“一天内”被更改过的文件；
-mtime +n : 列出在n天之前（不含n天本身）被更改过的文件名；
-mtime -n : 列出在n天之内（含n天本身）被更改过的文件名；
-newer file : 列出比file还要新的文件名
# 例如：
find /root -mtime 0 # 在当前目录下查找今天之内有改动的文件

# 与用户或用户组名有关的参数：
-user name : 列出文件所有者为name的文件
-group name : 列出文件所属用户组为name的文件
-uid n : 列出文件所有者为用户ID为n的文件
-gid n : 列出文件所属用户组为用户组ID为n的文件
# 例如：
find /home/httpd -user apache # 在目录/home/ljianhui中找出所有者为ljianhui的文件

# 与文件权限及名称有关的参数：
-name filename : 找出文件名为filename的文件
-size [+ -]SIZE : 找出比SIZE还要大 (+) 或小 (-) 的文件
-type TYPE : 查找文件的类型为TYPE的文件，TYPE的值主要有：一般文件 (f)、设备文件 (b、c)、
                目录 (d)、连接文件 (l)、socket (s)、FIFO管道文件 (p)；
-perm mode : 查找文件权限刚好等于mode的文件，mode用数字表示，如0755；
-perm -mode : 查找文件权限必须要全部包括mode权限的文件，mode用数字表示
-perm +mode : 查找文件权限包含任一mode的权限的文件，mode用数字表示
# 例如：
find / -name passwd # 查找文件名为passwd的文件
find . -perm 0755 # 查找当前目录中文件权限的0755的文件
find . -size +12k # 查找当前目录中大于12KB的文件，注意c表示byte
```

(二)、文件

1.cp - 复制文件及文件夹

```
# cp [option] [SOURCE PATH] [TARGET PATH]
```

- a : 将文件的特性一起复制
- p : 连同文件的属性一起复制, 而非使用默认方式, 与-a相似, 常用于备份
- i : 若目标文件已经存在时, 在覆盖时会先询问操作的进行
- r : 递归持续复制, 用于目录的复制行为
- u : 目标文件与源文件有差异时才会复制

2.mv - 移动文件及文件夹

```
# mv [option] [SOURCE PATH] [TARGET PATH]
```

- f : force强制的意思, 如果目标文件已经存在, 不会询问而直接覆盖
- i : 若目标文件已经存在, 就会询问是否覆盖
- u : 若目标文件已经存在, 且比目标文件新, 才会更新

3.rm - 删除文件及文件夹

```
# rm [option] [PATH]
```

- f : 就是force的意思, 忽略不存在的文件, 不会出现警告消息
- i : 互动模式, 在删除前会询问用户是否操作
- r : 递归删除, 最常用于目录删除, 它是一个非常危险的参数

慎用删除! 更加慎用rm -rf! 建议 先写要删除的文件或文件夹 再写rm命令! 血的教训!

4.ln - 软链接文件及文件夹

```
# ln [option] [SOURCE PATH] [TARGET PATH]
```

- b或--backup : 删除, 覆盖目标文件之前的备份;
- d或-F或--directory : 建立目录的硬连接;
- f或—force : 强行建立文件或目录的连接, 不论文件或目录是否存在;
- i或--interactive : 覆盖既有文件之前先询问用户;
- n或--no-dereference : 把符号连接的目的目录视为一般文件;
- s或—symbolic : 对源文件建立符号连接, 而非硬连接;
- S<字尾备份字符串>或--suffix=<字尾备份字符串>: 用"-b"参数备份目标文件后, 备份文件的字尾会被加上一个备份字符串, 预设的备份字符串是符号"~", 用户可通过"-s"参数来改变它; -v或—verbose: 显示指令执行过程; -V<备份方式>或--version-control=<备份方式>: 用"-b"参数备份目标文件后, 备份文件的字尾会被加上一个备份字符串, 这个字符串不仅可用"-s"参数变更, 当使用"-v"参数<备份方式>指定不同备份方式时, 也会产生不同字尾的备份字符串;
- help : 在线帮助;
- version : 显示版本信息。

(三)、软件

1.ps - 查看程序进程

```
# ps [option]
```

-A : 显示所有进程 (等价于**-e**) (utility)
-a : 显示一个终端的所有进程, 除了会话引线
-N : 忽略选择。
-d : 显示所有进程, 但省略所有的会话引线(utility)
-x : 显示没有控制终端的进程, 同时显示各个命令的具体路径。dx不可合用。 (utility)
-p pid : 进程使用cpu的时间
-u uid or username : 选择有效的用户id或者是用户名
-g gid or groupname : 显示组的所有进程。
U username : 显示该用户下的所有进程, 且显示各个命令的详细路径。如:ps U zhang;
(utility)
-f : 全部列出, 通常和其他选项联用。如: ps **-fa** or ps **-fx** and so on.
-l : 长格式 (有F,wchan,C 等字段)
-j : 作业格式
-o : 用户自定义格式。
v : 以虚拟存储器格式显示
s : 以信号格式显示
-m : 显示所有的线程
-H : 显示进程的层次(和其它的命令合用, 如: ps **-Ha**) (utility)
e : 命令之后显示环境 (如: ps **-d** e; ps **-a** e) (utility)
h : 不显示第一行

2.kill - 关闭指定进程

```
# kill [option] pid
```

-s : 指定发送的信号
-p : 模拟发送信号
-l : 指定信号的名称列表
-9 : 强制关闭
pid : 要中止进程的ID号
Signal : 表示信号

3.killall - 关闭所有指定进程

```
# killall [-iIe] [command name]
```

-i : 交互式的意思, 若需要删除时, 会询问用户
-e : 表示后面接的command name要一致, 但command name不能超过15个字符
-I : 命令名称忽略大小写

例如:

```
killall httpd # 关闭所有 apache 进程
```

4.file - 查看文件属性

```
# file [PATH]
```

```
file test.php
```

```
test.php: PHP script text, ASCII text
```

5.tar - 压缩、解压缩文件及文件夹

```
# tar [option] [SOURCE PATH] [option]
```

- c : 新建打包文件
- t : 查看打包文件的内容含有哪些文件名
- x : 解打包或解压缩的功能, 可以搭配 -C (大写) 指定解压的目录, 注意 -c, -t, -x 不能同时出现在同一条命令中
- j : 通过 bzip2 的支持进行压缩/解压缩
- z : 通过 gzip 的支持进行压缩/解压缩
- v : 在压缩/解压缩过程中, 将正在处理的文件名显示出来
- f filename : filename 为要处理的文件
- C dir : 指定压缩/解压缩的目录 dir

压缩: tar -jcv -f filename.tar.bz2 要被处理的文件或目录名称
查询: tar -jtv -f filename.tar.bz2
解压: tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

-jcv -f 可以合并写 -jcvf

6.cat - 查看文件内容

```
# cat [option] [PATH]
```

- n : 由 1 开始对所有输出的行数编号
- b : 和 -n 相似, 只不过对于空白行不编号
- s : 当遇到有连续两行以上的空白行, 就代换为一行的空白行
- v : 显示非打印字符

7.tail - 查看文件

```
# tail [option] [PATH]
```

- b : Number 从 Number 变量表示的 512 字节块位置开始读取指定文件。
- c : Number 从 Number 变量表示的字节位置开始读取指定文件。
- f : 如果输入文件是常规文件或如果 File 参数指定 FIFO (先进先出), 那么 tail 命令不会在复制了输入文件的最后的指定单元后终止, 而是继续从输入文件读取和复制额外的单元 (当这些单元可用时)。如果没有指定 File 参数, 并且标准输入是管道, 则会忽略 -f 标志。tail -f 命令可用于监视另一个进程正在写入的文件的增长。
- k : Number 从 Number 变量表示的 1KB 块位置开始读取指定文件。
- m : Number 从 Number 变量表示的多字节字符位置开始读取指定文件。使用该标志提供在单字节和双字节字符代码集环境中的一致结果。
- n : Number 从首行或末行位置来读取指定文件, 位置由 Number 变量的符号 (+ 或 - 或无) 表示, 并通过行号 Number 进行位移。
- r : 从文件末尾以逆序方式显示输出。-r 标志的缺省值是以逆序方式显示整个文件。

8.top - linux 版本的任务管理器

```
# top [option]
-d: 指定每两次屏幕信息刷新之间的时间间隔。当然用户可以使用s交互命令来改变之。
-p: 通过指定监控进程ID来仅仅监控某个进程的状态。
-q: 该选项将使top没有任何延迟的进行刷新。如果调用程序有超级用户权限，那么top将以尽可能高的优先级运行。
-S: 指定累计模式
-s : 使top命令在安全模式中运行。这将去除交互命令所带来的潜在危险。
-i: 使top不显示任何闲置或者僵死进程。
-c: 显示整个命令行而不只是显示命令名
```

9.scp - 主机之间文件复制

```
# scp [option] [SOURCE] [TARGET]
# scp /local/file remote_user@remote_ip:/remote/file_path
-v : 和大多数 linux 命令中的 -v 意思一样，用来显示进度。可以用来查看连接，认证，或是配置错误
-C 使能压缩选项
-P : 选择端口。注意 -p 已经被 rcp 使用
-4 : 强行使用 IPV4 地址
-6 : 强行使用 IPV6 地址
```

10.netstat - 显示各种网络相关信息

```
# netstat [option]
-a : (all)显示所有选项，默认不显示LISTEN相关
-t : (tcp)仅显示tcp相关选项
-u : (udp)仅显示udp相关选项
-n : 拒绝显示别名，能显示数字的全部转化成数字。
-l : 仅列出有在 Listen (监听) 的服务状态
-p : 显示建立相关链接的程序名
-r : 显示路由信息，路由表
-e : 显示扩展信息，例如uid等
-s : 按各个协议进行统计
-c : 每隔一个固定时间，执行该netstat命令。
```

11.ssh - 远程登陆

```
# ssh [option]
# ssh -l user -p 22 host/ip
-l : 登陆的用户名
-c : 登陆时使用的加密方式
-p : 登陆时使用的端口
```

三、vim简介

(一) vim 历史

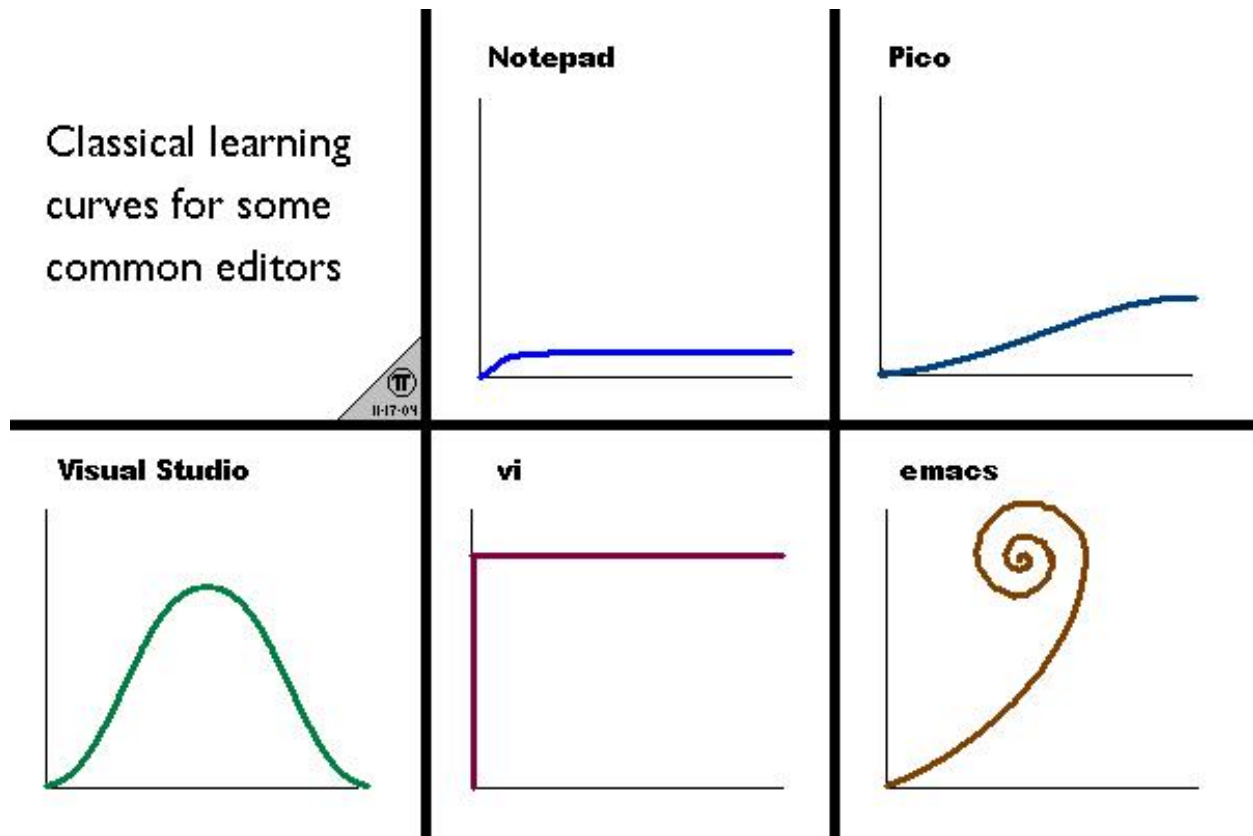
1.vim是继承于vi的高度可定制的文本编辑器，在vi的基础上改进和增强了很多特性

2.vim设计理念是组合：命令组合和模式组合

3.vim和emacs的传说：emacs是神的编辑器，而vim是编辑器之神

4.vim和emacs的另一个传说：emacs是伪装成编辑的操作系统，而vim是伪装成编辑器的编辑器

4.各编辑器学习曲线



(二) vim 模式

1.普通模式 normal-mode

普通模式主要用来浏览和修改文本内容

2.插入模式 insert-mode

i : 在光标所在字符前开始输入文字并进入插入模式

a : 在光标所在字符后开始输入文字并进入插入模式

o : (字母o) 在光标所在行的下面单独开一新行来输入文字并进入插入模式

s : 删除光标所在的字符并进入插入模式

I : 在行首开始输入文字并进入插入模式。此行首指第一个非空白字符处。如果行首有空格，则在空格之后输入文字并进入插入模式

A : 在行尾开始输入文字并进入插入模式。这个好用，您不必管光标在此行的什么地方，只要按 A 就会在行尾等着您输入文字。

O : (大写字母o) 在光标所在行的上面单独开一新行来输入文字并进入插入模式。

S : 删除光标所在行并进入插入模式

3.可视模式 visual-mode

可视模式相当于高亮选取文本后的普通模式。

可视模式具有子模式，以行为单位进行选取的可视行模式，使用“V”键进入（也就是Shift+v）；

和以块为单位

进行选取的可视块模式，使用“Ctrl+v”键进入。

4.命令模式 command-mode

命令模式用于操作文件或执行外部命令 Shift+: 进入命令模式

（三）、vim 常用命令

1.文件命令

#打开单个文件

```
vim file
```

#同时打开多个文件

```
vim file1 file2 file3 ...
```

#在vim窗口中打开一个新文件

```
:open file
```

#在新窗口中打开文件

```
:split file
```

#切换到下一个文件

```
:bn
```

#切换到上一个文件

```
:bp
```

#查看当前打开的文件列表，当前正在编辑的文件会用[]括起来。

```
:args
```

#打开远程文件，比如ftp或者share folder

```
:e ftp://192.168.10.76/abc.txt
```

```
:e \\qadrive\test\1.txt
```

2.移动命令

h : 左移一个字符

l : 右移一个字符, 这个命令很少用, 一般用**w**代替。

k : 上移一个字符

j : 下移一个字符

以上四个命令可以配合数字使用, 比如20**j**就是向下移动20行, 5**h**就是向左移动5个字符, 在Vim中, 很多命令都可以配合数字使用, 比如删除10个字符10**x**, 在当前位置后插入3个!, 3**a!** <Esc>, 这里的Esc是必须的, 否则命令不生效。

w : 向前移动一个单词 (光标停在单词首部), 如果已到行尾, 则转至下一行行首。此命令快, 可以代替**l**命令。

b : 向后移动一个单词 2**b** 向后移动2个单词

e : 同**w**, 只不过是光标停在单词尾部

ge: 同**b**, 光标停在单词尾部。

^ : 移动到本行第一个非空白字符上。

0 : (数字0) 移动到本行第一个字符上,

\$: 移动到行尾 3**\$** 移动到下面3行的行尾

gg : 移动到文件头。 = [[

G (shift + **g**) : 移动到文件尾。 =]]

f (find) : 命令也可以用于移动, **fx**将找到光标后第一个为**x**的字符, 3**fd**将找到第三个为**d**的字符。

跳到指定行, 冒号+行号, 回车, 比如跳到240行就是 :240回车。另一个方法是行号+**G**, 比如230**G**跳到230行。

Ctrl + **e** : 向下滚动一行

Ctrl + **y** : 向上滚动一行

Ctrl + **d** : 向下滚动半屏

Ctrl + **u** : 向上滚动半屏

Ctrl + **f** : 向下滚动一屏

Ctrl + **b** : 向上滚动一屏

3.撤销和重做

u : 撤销 (Undo)
U : 撤销对整行的操作
Ctrl + r : 重做 (Redo) , 即撤销的撤销

4.删除

x : 删除当前字符

3x : 删除当前光标开始向后三个字符

x : 删除当前字符的前一个字符。X=dh

dl : 删除当前字符, dl=x

dh : 删除前一个字符

dd : 删除当前行

dj : 删除上一行

dk : 删除下一行

10d : 删除当前行开始的10行。

D : 删除当前字符至行尾。D=d\$

d\$: 删除当前字符之后的所有字符 (本行)

kdgg : 删除当前行之前所有行 (不包括当前行)

jdG (jd shift + g) : 删除当前行之后所有行 (不包括当前行)

:1,10d : 删除1-10行

:11,\$d : 删除11行及以后所有的行

:1,\$d : 删除所有行

J(shift + j) : 删除两行之间的空行, 实际上是合并两行。

5.拷贝和粘贴

`yy` : 拷贝当前行

`nyy` : 拷贝当前后开始的n行, 比如`2yy`拷贝当前行及其下一行。

`p` : 在当前光标后粘贴, 如果之前使用了`yy`命令来复制一行, 那么就在当前行的下一行粘贴。

`shift+p` : 在当前行前粘贴

`:1,10 co 20` : 将1-10行插入到第20行之后。

`:1,$ co $` : 将整个文件复制一份并添加到文件尾部。

正常模式下按`v` (逐字) 或`V` (逐行) 进入可视模式, 然后用`jklh`命令移动即可选择某些行或字符, 再按`y`即可复制

`ddp` : 交换当前行和其下一行

`xp` : 交换当前字符和其后一个字符

6. 剪切

正常模式下按`v` (逐字) 或`V` (逐行) 进入可视模式, 然后用`jklh`命令移动即可选择某些行或字符, 再按`d`即可剪切

`ndd` : 剪切当前行之后的n行。利用`p`命令可以对剪切的内容进行粘贴

`:1,10d` : 将1-10行剪切。利用`p`命令可将剪切后的内容进行粘贴。

`:1, 10 m 20` : 将第1-10行移动到第20行之后。

7. 退出保存

`:wq` : 保存并退出

`zz` : 保存并退出

`:q!` : 强制退出并忽略所有更改

`:e!` : 放弃所有修改, 并打开原来文件。

8. 其它命令

`:! command` : 执行外部命令

`:help` : 显示全部帮助

`:help command` : 显示command的帮助

