

Linux基础&服务器介绍

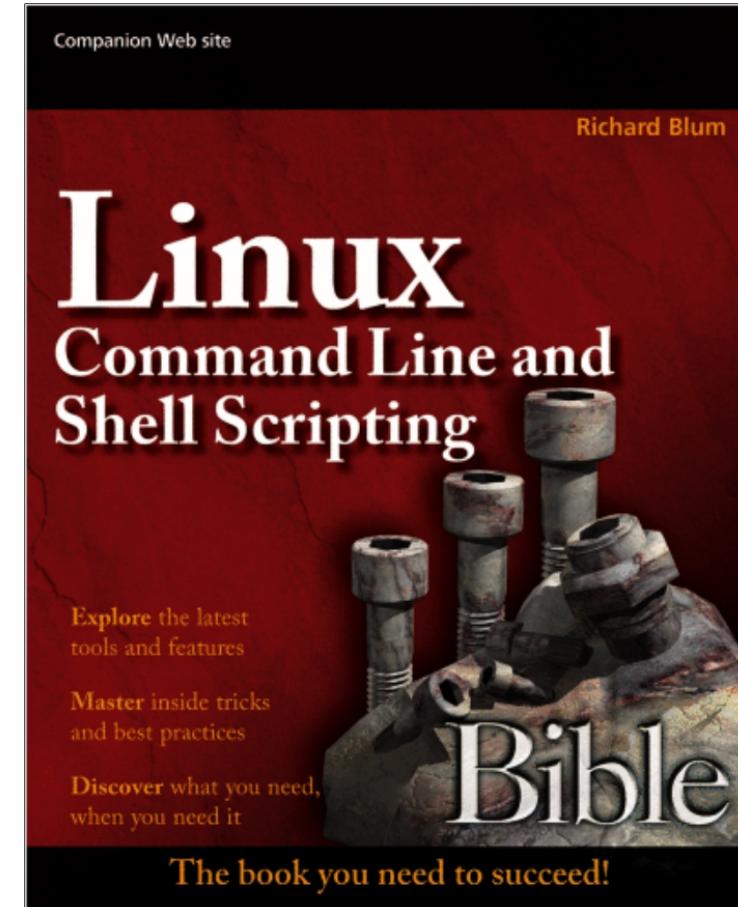
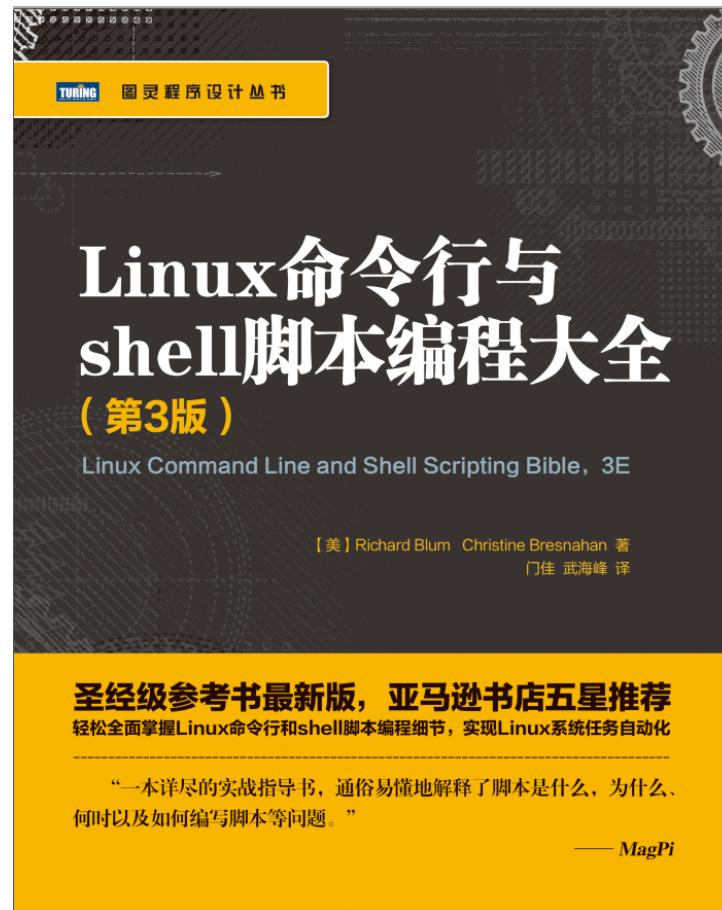
培训人：陈傲

导 师：程俊

2020-8-31

参考书籍

1. QQ群有PDF文件



2. 鸟哥的Linux私房菜 http://cn.linux.vbird.org/linux_basic/linux_basic.php

3. 常用bash命令简单用法查询 <https://devhints.io/bash>

Part1 登陆后台/终端

* 关于终端后面会有介绍，这一部分是便于培训的一些准备内容

后台/终端打开

利用jupyterhub登陆，网页打开：<http://139.9.84.172:8002>

已经给新生准备了测试账号可以用于登陆，相关账号可以询问培训组相关人员。

Sign in

Warning: JupyterHub seems to be served over an unsecured HTTP connection. We strongly recommend enabling HTTPS for JupyterHub.

Username:

Password:

Sign In

Home Page - Select or create a +

Not Secure | 139.9.84.172:8002/user/jfzhang/tree?

jupyterhub

Logout Control Panel

Files Running Clusters

Select items to perform actions on them.

0 /

Name ↓

The notebook list is empty.

Upload New ↗

Notebook: Python 3

Other: Text File Folder Terminal

Terminal

Home Page - Select or create a +

Not Secure | 139.9.84.172:8002/user/jfzhang/terminals/1

jupyterhub

Logout Control Panel

jfzhang@carbon:~\$

登陆后 → new → terminal

后台/终端打开

培训资料在 <https://github.com/chenggroup/new-comer-tutorial>

可以使用 git 命令下载到自己的账号查看：

jfzhang@carbon:~\$ git clone *web-URL*

The screenshot displays two windows side-by-side. On the left, a terminal window titled 'jfzhang@carbon: ~' shows the command `git clone https://github.com/chenggroup/new-comer-tutorial.git` being run, followed by the output of the cloning process. On the right, a browser window shows the GitHub repository `chenggroup / new-comer-tutorial`. The 'Code' tab is selected. A red oval highlights the 'Clone with HTTPS' button, which contains the URL `https://github.com/chenggroup/new-c`.

jfzhang@carbon:~\$ git clone https://github.com/chenggroup/new-comer-tutorial.git
Cloning into 'new-comer-tutorial'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 1), reused 6 (delta 1), pack-reused 0
Unpacking objects: 100% (10/10), done.
jfzhang@carbon:~\$ ls
new-comer-tutorial
jfzhang@carbon:~\$

GitHub - chenggroup/new-comer-tutorial

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search

Code Issues Pull requests Actions Projects Security Insights

Join GitHub today

over 50 million developers working together to host and review code, manage projects, and build software together.

Sign up

Go to file Code

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

`https://github.com/chenggroup/new-c`

LICENSE README.md

Initial commit Initial commit

README.md

new-comer-tutorial

Part2 Linux基础

- Linux终端就是控制台，是用户与内核交互的平台，通过输入指令来控制内核完成任务操作。
- 外形是一个方框，有光标在闪烁。
- 终端现在对于普通Linux来说，一般说的就是字符(或者模拟字符)的命令交互界面，通过终端命令实现对计算机的控制。

1. 启动shell

shell是Linux命令集的概称，是属于命令行的人机界面。

bash是borne again shell缩写，是shell一种，Linux默认采用bash，常用的还有zsh。

2. shell提示符

默认bash shell提示符是美元符号（\$），该符号表明shell在等待用户输入。

```
hub_test@carbon:~$ █
```

```
[aochen@mgt od-donor]$ |
```

```
windows@DESKTOP-L00F8NB MINGW64 ~/Desktop  
$ |  
# chenao @ chenaos-MacBook-Pro in ~ [9:54:55]  
$ |
```

提示符还能提供其他辅助信息，如上图中：用户ID@系统名。

shell提示符并非一成不变，可以根据需要修改。

3. bash手册

大多数Linux发行版自带用以查找shell命令及其他GNU工具信息的在线手册。

man命令用来访问存储在Linux系统上的手册页面。

Example: man xterm

读完手册页按q键退出。

Question: **man man**

学习技术行话

*手册用来**快速参考**，不是按部就班学习指南

*手册页不是唯一参考资料，还有一种info页面信息，大家通过info ls命令了解。

*大多数命令可以接受--help选项，例如

hostname --help ; ls --help

```

LS(1)                               User Commands                         LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries alphabetically if none of
    -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters

    --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes in units of 1,048,576
        bytes; see SIZE format below

```

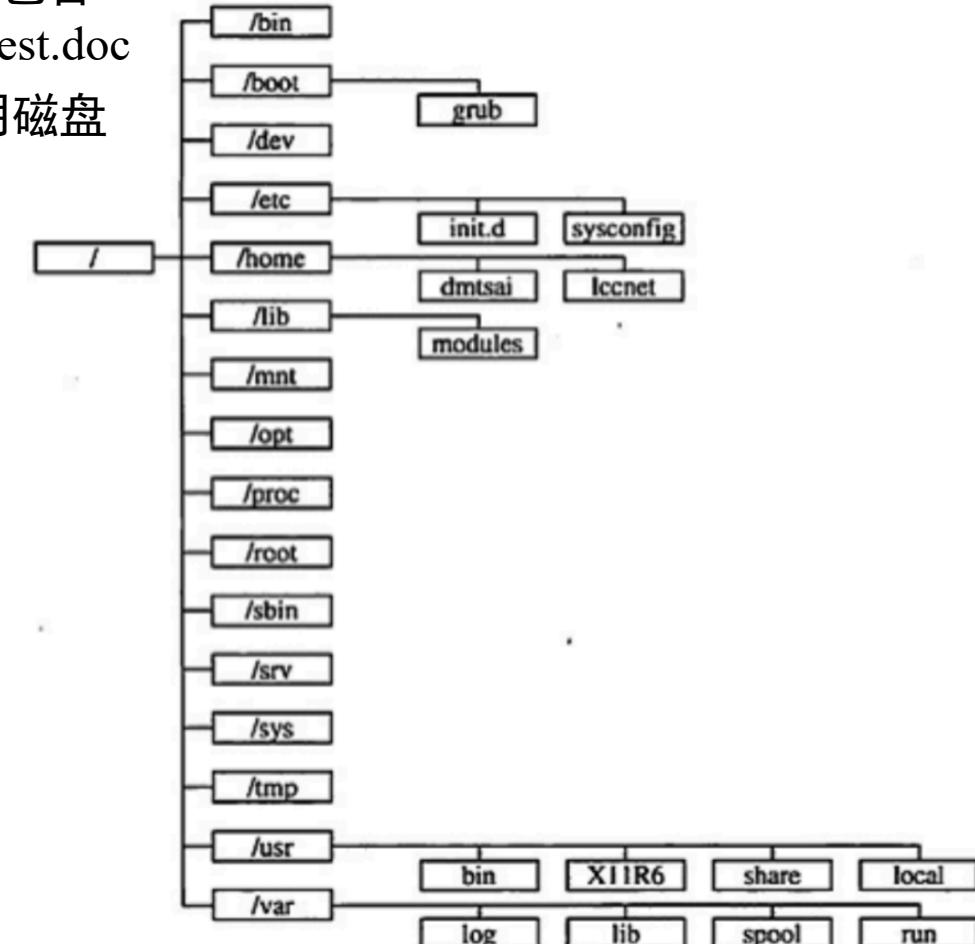
4. 浏览文件系统

Linux不使用驱动器盘符，文件存储在单个目录结构中，只包含一个根目录（/）作为基础目录，常见路径：/home/Rich/Documents/test.doc

Windows的文件路径：C:\Users\Rich\Documents\test.doc表明磁盘分区，若保存在闪存J上：J:\test.doc

表3-3 常见Linux目录名称

目 录	用 途
/	虚拟目录的根目录。通常不会在这里存储文件
/bin	二进制目录，存放许多用户级的GNU工具
/boot	启动目录，存放启动文件
/dev	设备目录，Linux在这里创建设备节点
/etc	系统配置文件目录
/home	主目录，Linux在这里创建用户目录
/lib	库目录，存放系统和应用程序的库文件
/media	媒体目录，可移动媒体设备的常用挂载点
/mnt	挂载目录，另一个可移动媒体设备的常用挂载点
/opt	可选目录，常用于存放第三方软件包和数据文件
/proc	进程目录，存放现有硬件及当前进程的相关信息
/root	root用户的主目录
/sbin	系统二进制目录，存放许多GNU管理员级工具
/run	运行目录，存放系统运作时的运行时数据
/srv	服务目录，存放本地服务的相关文件
/sys	系统目录，存放系统硬件信息的相关文件
/tmp	临时目录，可以在该目录中创建和删除临时工作文件
/usr	用户二进制目录，大量用户级的GNU工具和数据文件都存储在这里
/var	可变目录，用以存放经常变化的文件，比如日志文件



4.1. 遍历目录

切换目录命令 `cd` 将 shell 会话切换到另一个目录: `cd destination`

* 可以是绝对路径, 也可以是相对路径。

```
hub_test@carbon:~$ ls
new-comer-tutorial
hub_test@carbon:~$ cd new-comer-tutorial/
hub_test@carbon:~/new-comer-tutorial$ ls
LICENSE  linux  README.md
hub_test@carbon:~/new-comer-tutorial$ pwd
/home/hub_test/new-comer-tutorial
hub_test@carbon:~/new-comer-tutorial$ cd /home/hub_test/new-comer-tutorial/linux/
hub_test@carbon:~/new-comer-tutorial/linux$ ls
linux_and_hpc  linux_introduction_v1.2.pdf
hub_test@carbon:~/new-comer-tutorial/linux$ █
```

提示符中 “~” 信息被修改 (大家的账号登录系统后在 “/home/account” 目录下)

* `cd /data/test`

切换到目录 /data/test

* `cd ./path` 或 `cd path`

切换到当前目录下的 path 目录中, “.” 表示当前目录

* `cd ../path`

切换到上层目录中的 path 目录中, “..” 表示当前目录的父目录

`pwd` 该命令以绝对路径的方式显示当前目录

4.2. 文件和目录列表

可以使用**列表命令ls**，这是一个非常有用的查看文件与目录的命令，list之意。

```
[aochen@mgt aochen]$ pwd  
/data/aochen  
[aochen@mgt aochen]$ ls  
Ru02-Pt dingtalk_in_51.py langli nohup.out pt-co  
[aochen@mgt aochen]$ cd pt-co/  
[aochen@mgt pt-co]$ ls  
1Rb 1Rb-Cl c2h5 ch3 pzc  
[aochen@mgt pt-co]$ cd ../  
[aochen@mgt aochen]$ ls -a  
. .. Ru02-Pt dingtalk_in_51.py langli nohup.out pt-co  
[aochen@mgt aochen]$ ls -l  
total 40  
drwxrwxr-x 3 aochen aochen 4096 Jun 18 16:01 Ru02-Pt  
-rw-r--r-- 1 aochen aochen 1691 Jun 2 11:28 dingtalk_in_51.py  
drwxrwxr-x 3 aochen aochen 4096 Jun 9 14:54 langli  
-rw----- 1 aochen aochen 16608 Jul 10 00:33 nohup.out  
drwxrwxr-x 7 aochen aochen 4096 Apr 26 14:45 pt-co  
[aochen@mgt aochen]$ |
```

ls -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 (输出可能会很长)

ls -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)

ls -l:列出长数据串，包含文件的属性与权限数据等。

Question: ls -al

长列表格式

```
[aochen@mgt aochen]$ ls -l
total 40
drwxrwxr-x 3 aochen aochen 4096 Jun 18 16:01 Ru02-Pt
-rw-r--r-- 1 aochen aochen 1691 Jun  2 11:28 dingtalk_in_51.py
drwxrwxr-x 3 aochen aochen 4096 Jun  9 14:54 langli
-rw----- 1 aochen aochen 16608 Jul 10 00:33 nohup.out
drwxrwxr-x 7 aochen aochen 4096 Apr 26 14:45 pt-co
[aochen@mgt aochen]$ |
```

每一行包含文件（或目录）的下述信息：

文件类型： **目录(d)、文件(-)、字符型文件(c)或块设备(b)**

文件的权限： 三个一组， 分别代表文件主、组用户和其他用户权限， r可读、w可写、x可执行
文件的硬链接总数

文件属主的用户名

文件属组的组名

文件的大小（以字节为单位）

文件的上次修改时间

文件名或目录名

过滤输出列表

```
[aochen@mgt aochen]$ ls -l nohup.out
-rw----- 1 aochen aochen 16608 Jul 10 00:33 nohup.out
[aochen@mgt aochen]$ ls -l noh?p.out
-rw----- 1 aochen aochen 16608 Jul 10 00:33 nohup.out
[aochen@mgt aochen]$ ls -l no*
-rw----- 1 aochen aochen 16608 Jul 10 00:33 nohup.out
[aochen@mgt aochen]$ ls -l no*t
-rw----- 1 aochen aochen 16608 Jul 10 00:33 nohup.out
[aochen@mgt aochen]$ |
```

- 问号 (?) 代表一个字符
- 星号 (*) 代表零个或多个字符

```
[aochen@mgt run]$ ls
BASIS_MOLOPT      Pt-1.restart.bak-1  Pt-frc-1.xyz  ch3.xyz      input.inp
GTH_POTENTIALS    Pt-1.restart.bak-2  Pt-pos-1.xyz  coord.xyz   output.out
Pt-1.ener          Pt-1.restart.bak-3  Pt-vel-1.xyz  cube        pre-opt.wfn
Pt-1.restart       Pt-1.tregion           calculation  dftd3.dat  script
[aochen@mgt run]$ ls -l Pt-*
-rw-r--r-- 1 aochen aochen 4586441 Jan  8  2020 Pt-1.ener
-rw-rw-r-- 1 aochen aochen  244098 Jan  8  2020 Pt-1.restart
-rw-rw-r-- 1 aochen aochen  244098 Jan  8  2020 Pt-1.restart.bak-1
-rw-rw-r-- 1 aochen aochen  244098 Jan  8  2020 Pt-1.restart.bak-2
-rw-rw-r-- 1 aochen aochen  244098 Jan  8  2020 Pt-1.restart.bak-3
-rw-r--r-- 1 aochen aochen 3185984 Jan  8  2020 Pt-1.tregion
-rw-r--r-- 1 aochen aochen 894505500 Jan  8  2020 Pt-frc-1.xyz
-rw-r--r-- 1 aochen aochen 894505500 Jan  8  2020 Pt-pos-1.xyz
-rw-r--r-- 1 aochen aochen 894505500 Jan  8  2020 Pt-vel-1.xyz
[aochen@mgt run]$ |
```

可以有选择地选中需要的文件或目录

5. 处理文件

5.1. 创建文件

```
$touch test_one
```

5.2. 复制文件

cp source destination

```
$cp test_one test_two
```

Tip: 看看 man cp 有哪些参数

* **-r :递归持续复制** , 用于文件夹的复制行为 , 即命令为

```
cp -r /data/test/dir1 /data/test/dir2
```

* **制表键自动补全** , 避免长目录名或文件名输错。

创建新文件

复制文件 , 可以用绝对路径或相对路径

-i 强制shell询问是否需要覆盖已有文件 , 不回答y , 文件复制不会进行

```
[aochen@mgt aochen]$ cd test/  
[aochen@mgt test]$ ls  
[aochen@mgt test]$ touch test_one  
[aochen@mgt test]$ ls -l test_one  
-rw-rw-r-- 1 aochen aochen 0 Jul 15 15:53 test_one  
[aochen@mgt test]$ cp test_one test_two  
[aochen@mgt test]$ ls -l test_*
```

```
-rw-rw-r-- 1 aochen aochen 0 Jul 15 15:53 test_one  
-rw-rw-r-- 1 aochen aochen 0 Jul 15 15:54 test_two  
[aochen@mgt test]$ cp -i test_one test_two  
cp: overwrite 'test_two'? n  
[aochen@mgt test]$ |
```

5.3. 重命名文件

Linux中重命名文件称为移动(moving), **mv命令**可以将文件和目录**移动到另一个位置或重命名**。

test_two 重命名为test_three

test_three从test文件夹
移动到test_1文件夹

```
[aochen@mgt aochen]$ ls
Ru02-Pt dingtalk_in_51.py langli nohup.out pt-co test test_1
[aochen@mgt aochen]$ cd test
[aochen@mgt test]$ ls
test_one test_two
[aochen@mgt test]$ mv test_two test_three
[aochen@mgt test]$ ls
test_one test_three
[aochen@mgt test]$ cd ../test_1/
[aochen@mgt test_1]$ ls
[aochen@mgt test_1]$ mv ../test/test_three ./
[aochen@mgt test_1]$ ls
test_three
[aochen@mgt test_1]$ cd ../test/
[aochen@mgt test]$ ls
test_one
[aochen@mgt test]$ |
```

`mv file1 file2 file3 dir` #把文件file1、file2、file3移动到目录dir中

`mv file1 file2` #在file2不存在的情况下，把文件file1重命名为file2

`-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

*该命令可以把一个文件或多个文件一次移动一个文件夹中，但是最后一个目标文件一定要是“目录”

5.4. 删除文件

Linux中删除文件称为移除(removing), bash shell中**删除文件命令是rm**。

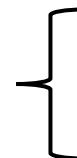
```
rm -i file      #删除文件file，在删除之前会询问是否进行该操作(推荐使用)  
rm -fr dir      #强制删除目录dir中的所有文件(慎用)
```

-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

比较一下加和不加 -r 的区别



```
[aochen@mgt aochen]$ ls  
Ru02-Pt dingtalk_in_51.py langli nohup.out pt-co test test_1  
[aochen@mgt aochen]$ rm test_1/  
rm: cannot remove 'test_1/': Is a directory  
[aochen@mgt aochen]$ rm -r test_1/  
[aochen@mgt aochen]$ ls  
Ru02-Pt dingtalk_in_51.py langli nohup.out pt-co test  
[aochen@mgt aochen]$ |
```

6. 处理目录

6.1. 创建目录

```
$mkdir New_dir
```

6.2. 删除目录

```
$rmdir New_dir #只能删除空目录
```

```
$rm -ri My_dir #删除大量目录文件, -ri 依次提醒非常琐碎, -rf 强制删除但很危险
```

```
[aochen@mgt aochen]$ rm -ri test/
rm: descend into directory 'test/'? y
rm: remove regular empty file 'test/test_one'? y
rm: remove directory 'test/'? y
[aochen@mgt aochen]$ ls
Ru02-Pt  dingtalk_in_51.py  langli  nohup.out  pt-co
[aochen@mgt aochen]$ |
```

7. 编辑器 vi / vim

Ubuntu中安装基础vim包: sudo apt-get install vim (一般不用装, 自带)

该命令主要用于文本编辑, 它接一个或多个文件名作为参数, 如果文件存在就打开, 如果文件不存在就以该文件名创建一个文件用vi命令(**vim filename**)打开文档之后, 按*i(insert之意)*进行文档编辑, 当编辑完成之后, 按ESC退出, 退出的命令有如下几个

:q 退出

:q! 强制退出

:wq 保存并退出

/fermi 在文档中查找fermi, 按n跳到下一个, shift+n上一个

一些常用到的操作:

- Ctrl+u:向文件首翻半屏;
- Ctrl+d:向文件尾翻半屏;
- Ctrl+f:向文件尾翻一屏;
- Ctrl+b:向文件首翻一屏;
- :行号:光标跳转到指定行的行首;

- :\$:光标跳转到最后一行的行首;
- D:删除从当前光标到光标所在行尾的全部字符;
- dd:删除光标行正行内容;
- ndd:删除当前行及其后n-1行;
- gg: 文档首 ; shift+g:文档末

编辑器 vi / vim 帮助

\$ vim

进入手册页输入 “:help X”

```
VIM - Vi IMproved
version 8.0.1453
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable

      Help poor children in Uganda!
type  :help iccf<Enter>      for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version8<Enter>   for version info
```

```
help.txt      For Vim version 8.0. Last change: 2017 Oct 28

VIM - main help file

Move around: Use the cursor keys, or "h" to go left,           h     i
             "j" to go down, "k" to go up, "l" to go right.      j
Close this window: Use ":q<Enter>".                         q
Get out of Vim: Use ":qa!<Enter>" (careful, all changes are lost!). q

Jump to a subject: Position the cursor on a tag (e.g. bars) and hit CTRL-T.
With the mouse:   ":set mouse=a" to enable the mouse (in xterm or GUI).
                 Double-click the left mouse button on a tag, e.g. bars.
Jump back: Type CTRL-T or CTRL-O. Repeat to go further back.

Get specific help: It is possible to go directly to whatever you want help
on, by giving an argument to the :help command.
Prepend something to specify the context: help-context

      WHAT          PREPEND      EXAMPLE
Normal mode command          :help x

help.txt [Help][RO]          1,1        Top

[ No Name ]
"help.txt" [readonly] 228L, 8578C
```

8. 查看文件内容

Linux中几个命令可以查看文件内容，而不需要调用其他文本编辑器。

8.1. 查看整个文件：cat命令

cat命令的一些功能：

- 查看全部内容 cat filename
- 创建一个文件 cat > filename
- 几个文件合并 cat file1 file2 >> newfilename

➤ cat filename 在终端print所有内容

➤ less filename 只读

查看文本内容

给有文本行加上行号

```
[aochen@mgt test]$ pwd  
/data/aochen/test  
[aochen@mgt test]$ ls -al  
total 12  
drwxrwxr-x 2 aochen aochen 4096 Jul 15 16:47 .  
drwx----- 6 aochen aochen 4096 Jul 15 16:46 ..  
-rw-rw-r-- 1 aochen aochen 38 Jul 15 16:47 test_one  
[aochen@mgt test]$ cat test_one  
hello  
this is a test file  
have a try  
  
[aochen@mgt test]$ cat -b test_one  
1 hello  
2 this is a test file  
3 have a try  
  
[aochen@mgt test]$ |
```

8.2. 查看部分内容：tail命令和head命令

tail默认显示后十行

head默认显示前十行

tail [参数] [filename]

tail -n 2 filename #只显示文件的最后两行

head -n 5 filename #只显示文件最前五行

该两个命令允许破折号后面输入想显示行数

tail -2 filename

head -5 filename

head -n 5 filename > newfile 将文档中的前5行输出到新文件newfile中

tail -f filename #随着文件的增长输出附加数据

```
[aochen@mgt test]$ cat test_one
```

```
hello  
this is a test file  
have a try  
line4  
line5  
l6  
l7  
l8  
l9
```

```
i'm fine  
line11  
thanks  
pardon  
line13  
l14  
please  
l16
```

```
[aochen@mgt test]$ tail test_one
```

```
l8  
l9  
i'm fine  
line11  
thanks  
pardon  
line13  
l14  
please  
l16
```

```
[aochen@mgt test]$ |
```

9. 搜索数据

从大文件中找一行数据，且该数据藏在文件中间，并不需要手动翻看，借助grep命令帮助查找。

\$grep sth filename #在filename文件中搜索 ‘sth’

\$grep -n sth filename #显示匹配行所在行号

*常与管道符 ‘|’ 连用

10. 查找属于账号的文件

find [PATH] [option] [action]

find ./ -name passwd #查找文件名为passwd的文件

find ./ -name *.wfn -delete #查找并删除

```
[aochen@mgt test]$ ls -al
total 20
drwxrwxr-x 2 aochen aochen 4096 Jul 15 17:18 .
drwx----- 6 aochen aochen 4096 Jul 15 16:46 ..
-rw-rw-r-- 1 aochen aochen     3 Jul 15 17:12 test_2
-rw-rw-r-- 1 aochen aochen    9 Jul 15 17:12 test_3
-rw-rw-r-- 1 aochen aochen  113 Jul 15 16:55 test_one
[aochen@mgt test]$ ls -al | grep 'dr'
drwxrwxr-x 2 aochen aochen 4096 Jul 15 17:18 .
drwx----- 6 aochen aochen 4096 Jul 15 16:46 ..
[aochen@mgt test]$ |
```

```
[aochen@mgt test]$ pwd
/data/aochen/test
[aochen@mgt test]$ ls
test_2  test_3  test_one
[aochen@mgt test]$ grep 'line' test_one
line4
line5
line11
line13
[aochen@mgt test]$ grep -n 'line' test_one
4:line4
5:line5
11:line11
14:line13
[aochen@mgt test]$ find /data/aochen/ -name nohup.out
[data/aochen/nohup.out
[aochen@mgt test]$ |
```

11. 文本处理常用命令

11.1. awk命令

awk是一个强大的文本分析工具，一般awk是把文件进行逐行的读入，以空格为默认分隔符将每行切片，切开的部分再进行各种分析处理。

使用方法：`awk '{pattern + action}' {filenames}`

其中pattern表示awk在数据中查找的内容，而action是在找到匹配内容时所执行的一系列命令。花括号（{}）不需要在程序中始终出现，但它们用于根据特定的模式对一系列指令进行分组。pattern就是要表示的正则表达式，用斜杠括起来。

使用实例：对某个文本文件第5列的数字输出并做平均之后覆盖到data.txt文本文件中

```
awk '{i++;s=s+($5);print ($5) " " s/i}' filename > data.txt
```

```
awk '{i++;s=s+($5);printf "%8f %8f\n", i, s/i}' filename > data.txt
```

11.2. sed命令

sed编辑器是一种流编辑器，可以根据命令来处理数据流中的数据，这些命令可以从命令行中输入也可以存储在一个命令文本文件中来使用，sed常用的命令参数有：

p==print

d: delete

-n: 取消默认的完整输出，只要需要的

-e: 是编辑命令，用于多个编辑任务

-i: 修改文件内容

使用实例(p d -n -e -i):

```
sed -n '3p' /data/test/filename #查看文件filename的第3行
```

```
sed -n '3,9p' /data/test/filename #查看文件filename的第3-9行
```

```
sed -i -e '1i a' -e '$b' /data/test/filename #在文件filename的第1行添加 'a'，在文件末尾添加 'b'
```

```
sed -i '1,3i c' /data/test/filename #在文件filename的第1行和第3行的每行下面添加 'c'
```

```
sed -i '1,3d' /data/test/filename #删除文件filename的第1行到第3行
```

11.3. if判断语句

最基本的结构化命令是if-then，有如下格式：

```
if command  
then  
    commands  
fi
```

if语句之后对象是一个等式，求值结果为TURE或FALSE。

使用示例：#!/bin/bash

```
SYSTEM=`uname -s` # 获取操作系统类型  
if [ $SYSTEM = "Linux" ] ; then # 如果是linux话输出linux字符串  
    echo "Linux"  
elif [ $SYSTEM = “FreeBSD” ] ; then # 如果是FreeBSD话输出FreeBSD字符串  
    echo "FreeBSD"  
else  
    echo “What?” # 如果两者都不是就输出What字符串  
fi # 判断结束，以fi结尾
```

11.4. for循环

重复执行一系列命令，可做批量处理。bash shell中for命令基本格式；

```
for var in list  
do  
    commands  
done
```

在list参数中，需要提供迭代中使用到的一系列值。

使用示例：#!/bin/bash

```
for dir in $(seq 10 10 100) #对名为10到100的文件夹，每隔10步取一个文件夹  
do #进行下述操作  
    cd $dir #分别进入文件  
    tail -n +3 filename > tmp.pdos #分别进入文件之后对文件里的文本文件进行需要的操作  
    cd .. #上述操作结束后返回到上层目录  
done #结束for循环
```

所有涉及到的命令：

cd	ls	pwd	
touch	cp	mv	rm
mkdir	rmdir		
vi/vim			
cat	tail	head	
grep	find		
awk	sed	if	for

Part3 服务器介绍

高性能计算集群（ HPC ）是一个计算机集群系统，它通过各种互联技术将多个计算机系统连接在一起，利用所有被连接系统的综合计算能力来处理大型计算问题，所以又通常被称为高性能计算集群，在某种意义上，又可以被看作是一台计算机。高性能计算方法的基本原理就是将问题分为若干部分，而相连的每台计算机(称为节点)均可同时参与问题的解决，从而显著缩短了解决整个问题所需的计算时间。

目前课题组服务器共有三套，IP分别为121.192.191.51，121.192.191.52，210.34.15.205

访问集群统，需要用户登录任意登录节点或管理节点

集群访问分为密码登录和密钥登录，目前**课题组服务器采用密钥登录，国重采用密码登录**，下面为不同系统登录服务器的具体介绍

SSH使用入门：https://chenggroup.github.io/wiki/ssh_note

课题组集群账号申请：

ssh-keygen #生成密钥

根据终端提示操作（实际可能只要不停按enter键），默认情况在 ~/.ssh 目录中得到 id_rsa 和 id_rsa.pub 文件，它们分别是私钥和公钥。创建好后，将公钥文件发给服务器管理员。**请务必保存好私钥，防止电脑被入侵。**

1. 登陆系统

ssh常用命令:-i, -p, -X(或-Y)

- -i为密钥登录
- -p为指定端口，一般服务器端口为22，此时可省略，当端口不是22时，需加-p，我们的服务器端口是6666，则输入ssh -p 6666 test@121.192.191.51
- 若需开启X11转发功能，-X 或 -Y
- 注意:这些命令可同时使用

Linux & MacOS —— 密码登录

在Linux & MacOS 系统下， 用户可以使用任何支持ssh2的客户端工具，
用ssh命令访问集群系统, 命令为\$ ssh username@ip。

打开终端， 假设该用户用户名为test， 服务器ip为121.192.191.51，
则输入ssh test@121.192.191.51, 根据提示输入密码即可登录

Linux & MacOS —— 密钥登录

密钥登录与密码登录类似， 打开终端， 输入ssh -i /Users/test/.ssh/id_rsa test@121.192.191.51即可登录，
其中/Users/test/.ssh为私钥所在路径， id_rsa为私钥名， test为用户名。

当私钥路径和名称分别为/Users/test/.ssh和id_rsa时可省略， 即上述情况也可以用ssh test@121.192.191.52 替代

例： ssh -p 6666 -i ~/.ssh/id_rsa test@121.192.191.52

或者 **ssh -p 6666 -i ~/.ssh/id_rsa -Y test@121.192.191.52**

2. 课题组服务器介绍

➤ \$HOME目录

每个账户登陆后，都存在一个\$HOME目录，如果在创建用户时未特别指定用户的\$HOME目录，则默认在`/home/username`。默认的，**用户在使用ssh客户端登陆时，会自动切换到用户的\$HOME，即**`/home/username`，**用户对此目录拥有完全权限**

➤ 工作目录

在当前集群系统中，每个账户会被分配一个数据目录`/data/username`，**用户对此目录拥有完全权限，可以在此目录进行:创建、修改、删除、运行等操作。****用户可以在此目录运行程序，安装私有程序。****用户使用ssh客户端登陆集群系统后，请执行如下命令切换到工作目录:**`cd /data/username`**可用**`pwd`**命令查看当前所在目录**

3. 作业提交管理

目前使用的服务器采用的都是lsf作业管理系统， 用户登录集群管理节点或者是登陆节点后可进行作业提交和修改作业提交

登录服务器后可进入工作目录下进行作业提交。课题组服务器工作目录为`/data/username`， 登录服务器后执行命令`cd /data/username`进入工作目录在lsf系统中， 用户使用`bsub`命令提交用户作业

例如:`bsub < cp2k.lsf`， 其中`cp2k.lsf`为调用`cp2k`软件的作业提交脚本， 其他软件脚本存放在服务器`/share/base/scripts`目录下

提交作业的脚本介绍

```
#!/bin/bash (声明shell)
#
#BSUB -q medium (说明队列, 具体队列可参考课题组服务器介绍)
#BSUB -W 12:00 (运行时间, 可修改, 12:00表示12小时)
#BSUB -J name (作业名字)
#BSUB -o cp2k.%J.stdout (作业标准输出, %J为作业ID)
#BSUB -e cp2k.%J.stderr (作业标准错误, %J为作业ID)
#BSUB -n 28 (申请的CPU核心数)
#BSUB -R "span[ptile=28]" (要求每个节点申请24核心, 可修改, 对51服务器不应超过24, 52服务器不应超过28, 否则作业无法运行。)
# add modulefiles (此处需要加载的程序)
module load intel/15.0.6 mpi/intel/5.0.3.049
module load cp2k/5.0
mpiexec.hydra cp2k.popt input.inp >& output (运行程序)
```

➤ 作业状态查询

提交作业后，可以通过bjobs命令查看所交作业的状态及相关信息，
作业状态包括RUN(正在运行)和PEND(排队中)

bjobs 查看用户正在运行的作业

bjobs -a 查看作业(包括短时间内完成的)

bjobs -u all 查看所有作业

bjobs -p 查看作业等待原因

bacct 查看使用资源情况

➤ 删除作业

提交作业后，如果发现作业错误可通过**bkill+作业号**来删除作业

先用bjobs命令查看自己的作业号，再用bkill+作业号来删除作业

bkill+作业号 删除作业

bkill 0 删除所有作业

➤ 查询队列信息

在管理节点或登录节点上执行bqueues命令，
可查询当前系统的队列
bqueues 查看队列信息
bqueues -l 查看队列详细信息

➤ 查询队列负载

用bhosts命令查询各节点的作业情况。
节点状态说明：

- 1.closed:所有CPU资源已被占用
- 2.ok:CPU空闲

4. scp文件传输

➤ 从本地复制到远程

- 复制文件

```
scp local_file remote_username@remote_ip:remote_folder
```

```
scp local_file remote_username@remote_ip:remote_file
```

```
scp local_file remote_ip:remote_folder
```

```
scp local_file remote_ip:remote_file
```

第1，2个指定了用户名，命令执行后需要再输入密码，第1个仅指定了远程的目录，文件名字不变，第2个指定了文件名；

第3，4个没有指定用户名，命令执行后需要输入用户名和密码，第3个仅指定了远程的目录，文件名字不变，第4个指定了文件名

- 复制目录

```
scp -r local_folder remote_username@remote_ip:remote_folder
```

```
scp -r local_folder remote_ip:remote_folder
```

第1个指定了用户名，命令执行后需要再输入密码；第2个没有指定用户名，命令执行后需要输入用户名和密码

4. scp文件传输

➤ 从本地复制到远程

➤ 从远程复制到本地

从远程复制到本地，只要将从本地复制到远程的命令的后2个参数调换顺序即可

```
scp test@121.192.191.51:/data/test/file /home
```

```
scp -r test@121.192.191.51:/data/test/dir /home
```

➤ 运用实例

□ 从本地复制到远程（从桌面复制文件到服务器）

命令格式:

```
scp -r -i .ssh/id_rsa -P 6666 /Users/username/Desktop/test username@121.192.191.52:/data/test
```

□ 从远程复制到本地（从服务器复制文件到桌面）

命令格式:

```
scp -r -i .ssh/id_rsa -P 6666 username@121.192.191.52:/data/test ~/Desktop/
```

所有涉及到的内容：

准备内容	ssh-keygen	id_rsa	id_rsa.pub	
登陆集群	ssh	-i	-p	-X (or -Y)
作业处理	bsub	bjobs	bkill	
文件传输	scp	-r	-i	-P

Thank you for your attention!