

Linux 使用操作培训

国家超级计算天津中心 李青峰

liqf@nsc-tj.cn

Operating system Family share for 06/2008

Operating system Family	Count	Share %	Rmax Sum (GF)	Rpeak Sum (GF)	Processor Sum
Linux	427	85.40 %	8465100	13333004	1408054
Windows	5	1.00 %	159264	211320	25472
Unix	25	5.00 %	619912	874185	73174
BSD Based	1	0.20 %	35860	40960	5120
Mixed	40	8.00 %	2391451	2941095	897127
Mac OS	2	0.40 %	28430	44816	5272
Totals	500	100%	11700016.97	17445379.70	2414219

世界超级计算机TOP500操作系统统计

- Linux介绍
- Linux基础
- Linux常用命令
- Bash基本功能
- Vim文本编辑器
- 总结

- ❑ 操作系统是负责对计算机硬件直接控制及管理的系统软件。
- ❑ 操作系统的功能一般包括处理器管理、存储管理、文件管理、设备管理和作业管理等等。当多个程序同时运行时，操作系统负责规划以优化调度每个程序。
- ❑ 对于计算机系统而言：操作系统是对所有系统资源进行管理的程序的集合；
- ❑ 对于用户而言：操作系统提供了对系统资源进行有效利用的简单抽象的方法。

- **Linux**是一套**免费使用和自由传播**的类**Unix**操作系统，其目的是建立不受任何商品化软件的版权制约的、全世界都能自由使用的**Unix** 兼容产品。
- **Linux** 以它的高效性和灵活性著称。它能够在个人计算机上实现全部的**Unix** 特性，具有多任务、多用户的能力。
- **Linux** 支持多种硬件平台，包括低端的个人计算机和高端的超级并行计算机系统。
- **Linux** 系统具有强大的网络功能。

- 开放性

系统遵循世界标准规范，完全符合**POSIX** 标准。

- 多用户

系统资源可以被不同用户使用，每个用户对自己的资源有特定的权限，互不影响。

- 多任务

计算机可同时执行多个程序，且各个程序的运行互相独立。

- 良好的用户界面

Linux 同时向用户提供了字符界面和图形界面，给用户呈现一个直观、易操作、交互性强的友好的图形化界面。

- 设备独立性

Linux 把所有外部设备统一当作文件来看待，任何用户都可以象使用文件一样，操纵、使用这些设备。

- 丰富的网络功能

完善的内置网络是Linux一大特点。

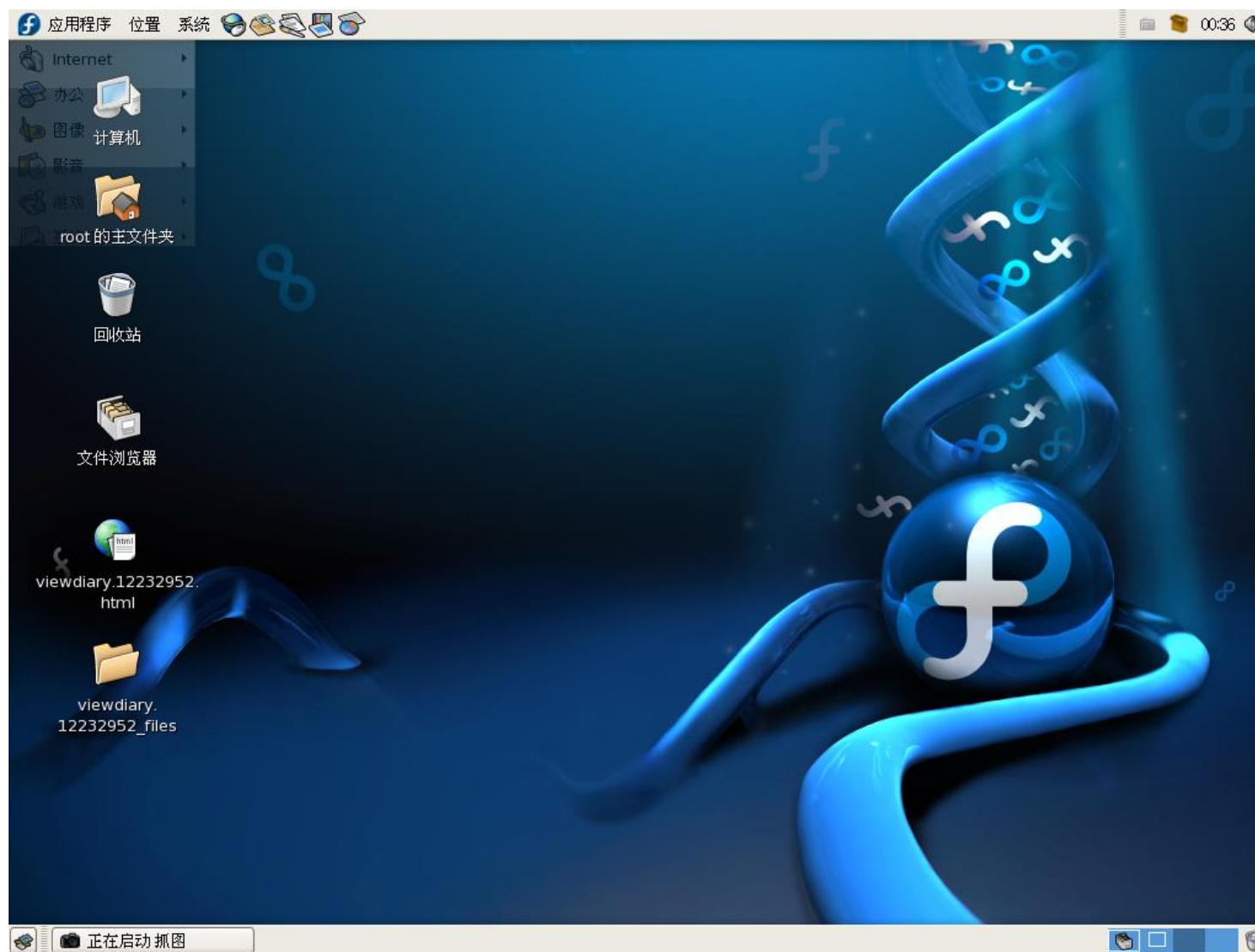
- 可靠的安全系统

Linux 采取多种安全技术措施，为网络多用户环境中的用户提供了必要的安全保障。

- 良好的可移植性

Linux 是一种可移植的操作系统，能够在从微型计算机到大型计算机的任何环境中和任何平台上运行。

- **内核：**内核是系统的核心，是运行程序和管理各种硬件设备的核心程序。
- **Shell：**Shell 是Linux 系统的用户界面，提供了用户与内核进行交互操作的一种接口，是命令语言解释器。
- **文件系统：**文件系统是文件存放在磁盘等存储设备上的组织方法。Linux 支持多种文件系统，如EXT2、EXT3、FAT、NTFS、ISO9660、NFS、SMB 等。
- **应用软件：**标准的Linux 发行版都有一整套的应用程序集，包括文本编辑器、编程语言、X Window、办公套件、Internet 工具、数据库等。



- Linux 提供了漂亮的图形界面操作环境，但命令行操作方式仍必不可少。
 - ✓ 场景所需
 - ✓ 操作便捷
- Linux 下的字符操作环境主要使用终端(Terminal);
- 终端是UNIX 的一个标准特性，用户可以通过终端访问系统资源。

```

? MobaXterm 10.9 ?
(SSSH client, X-server and networking tools)

> SSH session to liqf@192.168.2.103
? SSH compression : ✓
? SSH-browser      : ✓
? X11-forwarding   : ✓ (remote display is forwarded through SSH)
? DISPLAY          : ✓ (automatically set on remote server)

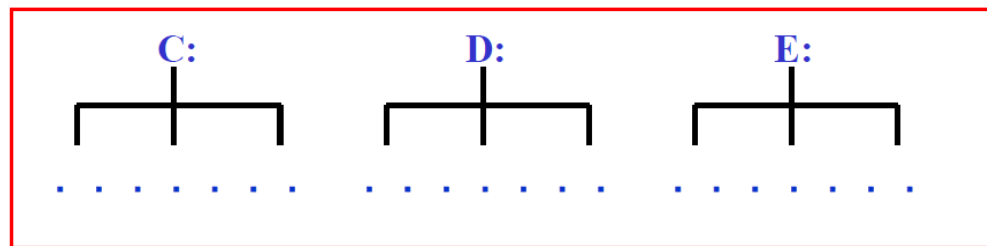
> For more info, ctrl+click on help or visit our website

Last login: Tue Nov  5 14:33:35 2019 from 10.200.200.67
*****
* 1.Welcome to TH-1A System of NSCC-TJ.                               *
* 2.If you have any problem, you can send mail to support@nsccl-tj.cn *
*****
Disk quotas for grp liqf (gid 40343):
Filesystem      used  quota  limit  grace  files  quota  limit  grace
/THL7  10.88G   500G   1T      -   172072 1000000 2000000  -

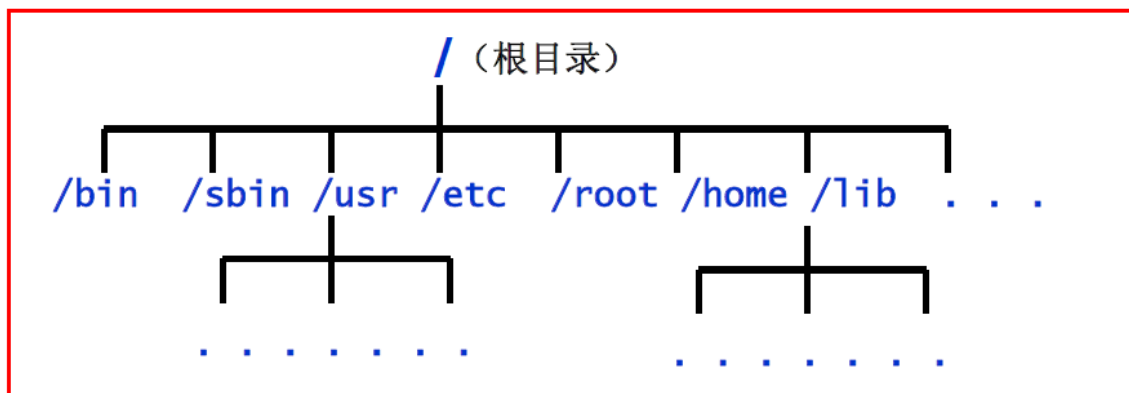
Important Notice
Dear user,
1. The domain name of NSCC-TJ will be closed on March 1, 2019.
2. The domain name of NSCC-TJ website will be changed from www.nsccl-tj.gov.cn to www.nsccl-tj.cn.
3. The domain name of NSCC-TJ mailbox will be changed from mail.nsccl-tj.gov.cn to mail.nsccl-tj.cn.
4. If you have any problem, you can send mail to support@nsccl-tj.cn.
We are sorry for any inconvenience this may cause and appreciate your understanding!
National SuperComputing Center in Tianjin

[liqf@th-hpc2-ln0 ~]$
```

- Windows 文件系统以分区为树根，若有多个分区，则形成多个树并列的情形：



- Linux 文件系统是一个目录树的结构，文件系统结构从一个根目录“/”开始，根目录下可以有任意多个文件和子目录，子目录中又可以有任意多个文件和子目录。





- 用户登录后，将会进入一个系统指定的专属目录，即用户的主目录，该目录名通常为用户的登录帐号。

例：用户**liqf**的主目录为**/vol6/home/liqf**

- 用户对自己主目录的文件拥有所有权，可以在自己的主目录下进行相关操作。
- 每个用户都被分配到一个指定的组(group) 中。
 - 主账号名和用户组名相同
 - 组内可以建立子账号

- **路径：**文件或目录在文件系统中所处的位置，分为“绝对路径”和“相对路径”
- **绝对路径：**是指一个文件或目录从根目录开始的完整的路径。
- **相对路径：**是指一个文件或目录相当于当前工作目录的路径。
 - 任何不以/ 和~ 开始的路径均为相对路径。

例：系统中linux.ppt 的绝对路径为：

`/home/liqf/course/linux.ppt`

在/home/liqf下的相对路径为：

`./linux/lect01.ppt`

- ❑ 用户登录后进入的目录通常是自己的主目录
- ❑ 用户可用**pwd**命令查看当前工作目录
- ❑ 可用**cd** 命令来切换目录
- ❑ 一些特殊字符的特殊含义：
 - ✓ “.” 表示当前目录
 - ✓ “..” 表示当前目录的上一级目录（父目录）
 - ✓ “-” 表示用**cd** 命令切换目录前所在的目录
 - ✓ “~” 表示用户主目录的绝对路径名

- **文件名：**用户可以创建文件名最长为**255** 个字符的文件，且可以使用几乎任何字符，但通常使用字母（大小写）、数字、下划线、句号等。

例： **This_is_my_First_file.txt**

□ Linux 下的文件类型

- ✓ 普通文件
- ✓ 目录文件
- ✓ 链接文件
- ✓ 设备文件

□ 普通文件

- 压缩与归档文件
.gz .tar .tgz .zip .rar 等
- 文字与多媒体文件
.gif .jpg .png .html .ps .pdf .txt 等
- 编程与脚本文件
.c .cpp .h .o .f .so .sh 等
- 系统文件
.cong .rpm 等

□ 目录文件

□ 链接文件

- 其作用类似于windows 下的快捷方式，它本身不包含内容，而是指向其它的文件或目录。
- 可用**ln** 命令创建链接文件。

□ 设备文件

存放在**/dev** 目录下。如：

hda、hdb、sda、sdb、fd0、lp1、...

□ 文件和目录的权限：

一个文件或目录可以有**读**、**写**和**执行**的权限。当一个文件或目录被创建时，系统会自动赋予文件所有者的读和写的权限，这样就允许文件所有者能够阅读和修改该文件。

□ 文件的访问权限：

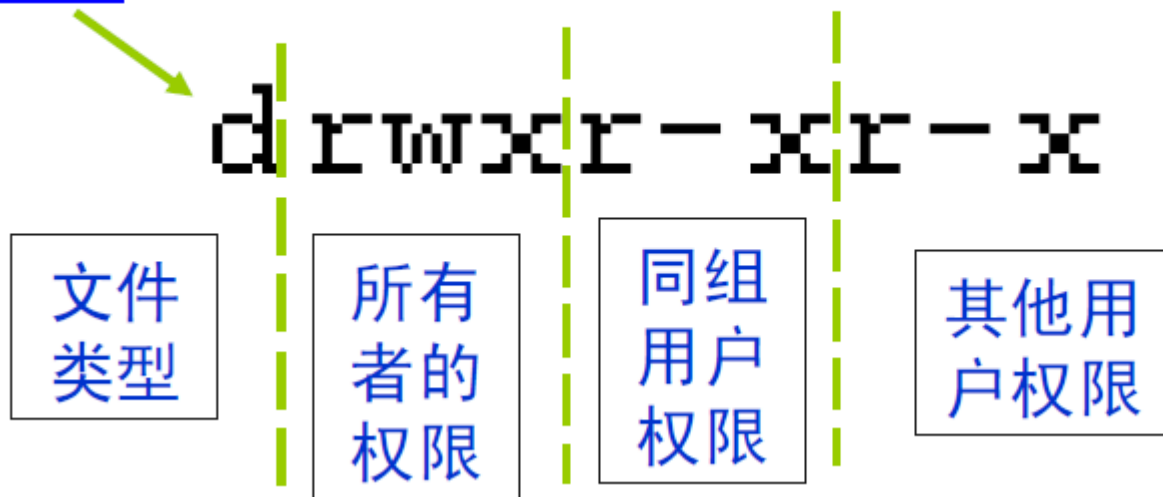
- ✓ 只允许用户自己访问；
- ✓ 允许一个预先指定的用户组的用户访问；
- ✓ 允许系统内的所有用户访问；

查看文件和目录的属性: **ls -l 或 ll**

-rw-r--r--	1	jypan	tch	702019	6月 7 13:29	2ndECpde2.zip
-rw-r--r--	1	jypan	tch	472956	3月 15 11:52	2ndECpde.zip
drwx-----	2	jypan	tch	4096	5月 13 12:22	backup
drwxr-xr-x	3	jypan	tch	4096	5月 3 01:35	bin
<u>lrwxrwxrwx</u>	1	root	root	11	<u>12月 27 2003</u>	Desktop -> .Desk

↓	↓	↓	↓	↓	↓	↓
文件类型	文件权限	链接个数	文件所有者	文件所有者所在的用户组	文件长度	文件上次修改的时间和日期
						文件名

```
drwxr-xr-x  3 jypan  tch          4096  5月  3 01:35 bin
```



第1个字符代表的文件类型

- 普通文件
- d 目录
- l 链接文件

第2~10个字符是用来表示权限。

这9个字符每3个一组，组成3套权限控制，第一套控制文件所有者的访问权限，第二套控制所有者所在用户组的其他成员的访问权限，第三套是控制系统内其他用户的访问权限。

drwxr-xr-x

每套控制中三个字符按顺序排列，所代表的含义如下：

r	Read (读取): 对文件而言，具有阅读文件内容的权限，对目录而言，具有浏览目录内容的权限。
w	Write (写入): 对文件而言，具有修改文件内容的权限，对目录而言，具有移动、删除目录内文件的权限。
x	eXecute (执行): 对文件而言，具有执行文件的权限，对目录而言，具有进入目录的权限。
-	表示不具有该权限，或取消该权限

例：

-rwx------

普通文件

系统内其它用户没有读、写和执行权限

用户组其它成员没有读、写和执行权限

用户有读、写和执行权限

只有文件所有者具有读、写和执行权限

命令	说明	命令	说明
cat	查看文件内容	rm	删除文件
more		pwd	显示当前所在目录
less		mkdir	创建目录
ls	显示目录内容	chmod	改变文件/目录权限
cd	切换工作目录	tar	文件打包/解包
cp	拷贝文件	ln	创建文件链接
mv	移动文件	which	显示命令所在目录
passwd	修改账号密码		

□ **cd**: 改变工作目录

- 用法: **cd** 目录
- 进入指定的目录, 即将工作目录改为指定的目录

□ **pwd**: 显示当前目录

- 用法: **pwd**
- 告诉用户当前的工作目录, 即当前在哪个目录下

□ **ls**: 显示文件或目录信息

- 用法: **ls** [选项] [目录或是文件]

□ 常用选项:

- ✓ **-a** 列出目录下的所有文件，包括以. 开头的隐含文件。
- ✓ **-l** 列出文件的详细信息，可以用ll表示。
- ✓ **-t** 以时间排序。
- ✓ **-d** 输入参数是目录时，只显示该目录名，而不是列出该目录下的文件。
- ✓ **-x** 按列输出，横向排序。

□ **mkdir**: 创建目录

用法: **mkdir** 目录名

□ 例: **mkdir data**

在当前目录下创建一个名为**data** 的目录。

□ 例: **mkdir -p /vol6/home/liqf/test/data**

递归创建目录

□ **mv**: 移动或更名现有的文件或目录

用法: **mv** [选项] 源文件或目录 目标文件或目录

□ 主要用法

- 若目标文件已存在，则被覆盖。
- 把多个源文件移至目标目录中。
- 把源目录改名为指定的目标目录（目标目录不存在）。
- 把源目录移至指定的目标目录中。

□ 常用选项:

- **-f** 若目标文件与源文件同名，则直接覆盖而不提示；
- **-i** 和 **f** 选项相反，在覆盖目标文件之前要求用户确认，回答 **y** 时目标文件将被覆盖；
- **-u** 如果要被覆盖的文件或目录比源文件要新，则不覆盖

例:

- **mv file1 mydata/**
将文件**file1** 移至当前目录的子目录**mydata/** 中
- **mv -i file1 mydata/file2**
将文件**file1** 移至当前目录的子目录**mydata/** 中，并改名为**file2**
- **mv data1 /home/zhang/data**
 - ✓ 若目标目录**/home/zhang/data** 不存在，则将当前目录下的子目录**data1** 移至**/home/zhang/**目录中，并改名为**data**;
 - ✓ 若目标目录**/home/zhang/data** 已存在，则**data1** 将被移至**/home/zhang/data**目录下;
 - ✓ 若**/home/zhang/data** 中已存在目录**data1** ，则报错

□ **cp**: copy, 文件复制

- 用法: **cp** [选项] 源文件或目录 目标文件或目录

□ 常用选项:

- **-f** 若目标文件与源文件同名, 则直接覆盖而不提示。
- **-i** 和**f** 选项相反, 在覆盖目标文件之前将给出提示要求用户确认, 是交互式拷贝。
- **-p** 此时**cp** 除复制源文件的内容外, 同时保留其修改时间和访问权限。
- **-r** 若给出的源文件是目录文件, 此时**cp** 将递归复制该目录下所有子目录和文件。此时目标文件必须为一个目录名。复制目录时必须加此选项。

例:

✓ **cp file /home/jypan/data/**

将文件file拷贝到/home/jypan/data/目录下;

✓ **cp file /home/jypan/file.bk**

将文件file 拷贝到/home/jypan/ 这个目录下, 并改名为file.bk

✓ **cp -pr /home/jypan/bin/ /root/**

将/home/jypan/bin/ 目录中的所有文件及其子目录拷贝到/root/目录下, 并保留所有文件的属性。

□ **rm**: 删除文件或目录

用法: **rm** [选项] 文件或目录名

□ 常用选项:

- ✓ **-f** 强制删除，并且不给提示。
- ✓ **-i** 删除每个文件时给用户提示。
- ✓ **-r** 删除整个目录，包括文件和子目录，用**rm** 删除目录时必须使用该选项。

□cat: 查看文件内容

用法: **cat** [选项] 文件列表

- 功能一: 用来显示文件内容。它依次读取指定文件的内容并在标准输出中输出。
- 功能二: 用来将两个或多个文件连接起来。

例1: **cat -n file1**

将文件file1 的内容在屏幕上输出, 并显示行号。

例2: **cat file1 file2 > file3**

将文件file1 和文件file2 的内容合并起来, 放入文件file3中, 此时在屏幕上并不能直接看到该命令执行后的结果。

注: 当文件内容过长时, 就带来一个问题, 因为文本在屏幕上迅速地闪过, 用户来不及看清其内容。因此, 当文件内容较长时, 一般可用**more** 或 **less**等命令分屏显示。

□ **more / less**: 查看文档内容

用法: **more / less** 文件名

该命令一次显示一屏文本，显示满之后，停下来，并在终端底部打印出“-- More --”，若要继续显示，按回车（显示下一行）或空格键（显示下一屏）即可。若要退出，按**q**

more 显示文件内容时，只能往后翻

less 可以相前翻，也可以向后翻。向后翻时，除了用回车和空格外，还可以用字母**f**；翻看前面的内容，可以用字母**b**

□ ln: 创建链接(link)

- 用法: **ln** [选项] 目标[链接名]

创建链接，是给系统中已有的某个文件指定另外一个可用于访问它的名称。如果链接指向目录，用户就可以利用链接直接该目录。删除这个链接，不会破坏原来的文件和目录。

□ 常用选项:

- **-s** 建立符号链接(软链接)。

```
[liqf@th1a test]# ln file2 file2hand
[liqf@th1a test]# ln -s file1 file1soft
[liqf@th1a test]# ls -il
1491138 -rw-r--r-- 1 root root 48 07-14 14:17 file1
1491140 lrwxrwxrwx 1 root root 5 07-14 14:24 file1soft -> file1
1491139 -rw-r--r-- 2 root root 0 07-14 14:17 file2
1491139 -rw-r--r-- 2 root root 0 07-14 14:17 file2hand
```

□ tar: 文件打包和解包

用法: **tar** [主选项+辅选项] 文件或者目录

□ **tar** 可以把一大堆的文件和目录全部打包成一个文件, 这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。

□ **tar** 有主选项和辅选项, 主选项是必须要有的, 它告诉**tar** 要做什么事情, 辅选项是辅助使用的, 可以选用。

□ 常用主选项:

- ✓ **-c** 创建新的档案文件。
- ✓ **-r** 把要存档的文件追加到档案文件的末尾。
- ✓ **-t** 列出档案文件的内容, 查看已经备份了哪些文件。
- ✓ **-u** 更新档案中的已有文件, 如果备份文件中没有要更新的文件, 则将该文件追加到备份文件的最后。
- ✓ **-x** 从档案文件中释放文件。

□ 常用辅助选项：

- ✓ **f** 使用档案文件或设备，这个选项通常是必选的。
- ✓ **k** 保存已经存在的文件。例如我们把某个文件还原，在还原的过程中，遇到相同的文件，不会进行覆盖。
- ✓ **m** 在还原文件时，把所有文件的修改时间设定为现在。
- ✓ **M** 创建多卷的档案文件，以便在几个磁盘中存放。
- ✓ **v** 详细列出**tar** 处理的文件信息。
- ✓ **z** 用**gzip** 来压缩/解压缩文件，备份时加上该选项可以将档案文件进行压缩，但还原一个压缩的备份文件时必须使用该选项进行解压缩。
- ✓ **j** 用**bzip2** 来压缩/解压缩文件。

例:

- **tar -cvf data1.tar data1**

将子目录data1/ 下所有文件备份，包括该目录下的所有子目录中的文件，备份文件名为data1.tar

- **tar -czvf data1.tgz data1**

将子目录data1/ 下所有文件压缩，文件名为data1.tgz

- **tar -xzvf data1.tgz**

解压data.tgz中的文件

- **tar** 的选项前的符号“-” 可以不加

□ **which**: 查找指定命令所在的位置

用法: **which** command

□ 例: **which gcc**

/usr/local/bin/gcc

查找命令gcc的全路径名

□ **chmod** 修改文件或目录的访问权限

□ **chmod** 有两种用法:

- 文字设置法
- 数字设置法

□ 文字设定法：使用字母和操作符表达式来修改或设定文件的访问权限

□ 使用方法：**chmod [who] [+|-|=] [mode] 文件或目录名**

□ 操作对象**who** 可是下述字母中的任一个或者它们的组合：

- **u** 表示“用户（**user**）”，即文件或目录的所有者
- **g** 表示“同组（**group**）用户”
- **o** 表示“其他（**others**）用户”
- **a** 表示“所有（**all**）用户”（缺省值）

□ 访问权限可用下述字母的任意组合

r 可读

w 可写

x 可执行

□ 操作符号可以是：

- **+** 添加某个权限。
- **-** 取消某个权限。
- **=** 赋予给定权限并取消其他所有权限

例：**chmod +x myname.txt**

□ 数字设定法


chmod num 文件或目录名

□ 我们必须先了解如何用数字表示的属性：


- 0 表示没有权限
- 1 表示可执行权限
- 2 表示可写权限
- 4 表示可读权限
- 数字相加，即可得到一个表示访问权限的数字
- 一个文件的访问权限可以用三个数字来表示，分别代表用户、同组用户和其它用户的访问权限

权限	对应数字
r	4
w	2
x	1
-	0

-rw-r--r--  **644**

drwx--x--x  **711**

drwx-----  **700**

-rwxr-xr-x  **755**

□ 数字设定法举例

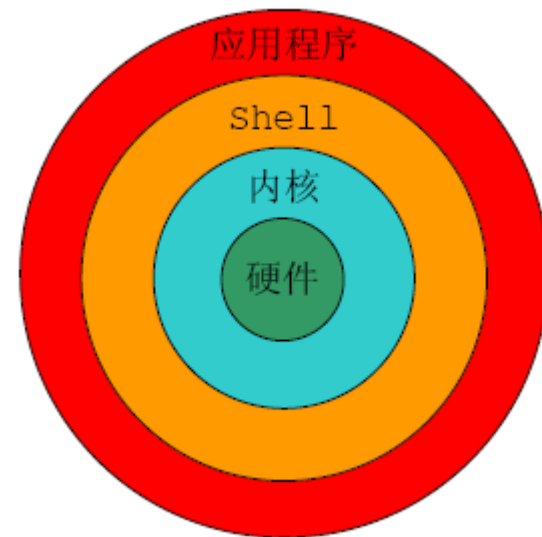
- **chmod 644 myname.txt**
设定文件**myname.txt** 的属性为: **-rw-r--r--**
- **chmod 750 myname.txt**
设定文件**myname.txt** 的属性为: **-rwxr-x---**
- **chmod 700 mydata/**
设定目录**mydata** 的属性为: **-rwx-----**

▣ **passwd**:更改口令

- 直接在命令行中输入**passwd**，然后根据提示，先输入原来的口令，回车后，再输入新口令，并确认一次。
- 为确保高安全性，Linux 对口令有一定的要求，而且通常需要在口令中加入一些特殊符号，如加号、减号等。

提醒：输入密码时，屏幕上不会有任何输出或提示！

- **Shell** 是系统的用户界面，它提供了用户和**Linux**（内核）之间进行交互操作的一种接口。用户在命令行中输入的每个命令都由**shell** 先解释，然后传给**Linux** 内核去执行。
- 如果把**Linux** 内核想象成一个球体的中心，**shell** 就是围绕内核的外层，从**shell** 向**Linux** 操作系统传递命令时，内核就会做出相应的反应。
- **Bash**基本特点：
 - **bash**是现在大多数**Linux** 系统的缺省**shell**
 - **bash**包含了很多**cs****h** 和**ksh** 里的优点。
 - **bash**有很灵活和强大的编程接口
 - **bash**有很友好的用户界面。



□ 命令行

当用户打开一个(虚拟) 终端时，可以看到一个**shell** 提示符，标识了命令行的开始。用户可以在提示符后面输入任何命令

command [选项] [参数]

例: **ls -l /home/jypan/linux/**

□ 在一个命令行中可以输入多个命令，用分号将各个命令隔开。

ls -F; cp -i mydata newdata

□ 如果一个命令太长，无法在一行中显示，可以使用反斜杠\ 来续行，在多个命令行上输入一个命令或多个命令。

**ls -F; \
cp -i mydata newdata**

常用的快捷键和组合键：

左/右箭头键	向左/向右移动一个字符
Ctrl+a	移动到当前行的行首
Ctrl+e	移动到当前行的行尾
Ctrl+f	向前移动一个字符
Ctrl+b	向后移动一个字符
Ctrl+k	从光标处删除到本行的行尾
Ctrl+u	从光标处删除到本行的行首
Ctrl+l	清屏
Alt+f	向前移动一个单词
Alt+b	向后移动一个单词

□通配符

bash 提供许多功能用来帮助用户节省输入命令的时间，其中最常用的一种方法就是使用通配符，可以用来在引用文件名时简化命令的书写。

□ 在**bash** 中可以使用三种通配符：*****、**?**、**[]**。

*	匹配任意长度的字符串（包括零个字符）
?	匹配任何单个字符
[]	<p>创建一个字符表列，方括号中的字符用来匹配或不匹配单个字符。如：[xyz] 匹配x、y 或z，但不能匹配xx，xy 或者其它任意组合。无论列表中有多少个字符，它只匹配一个字符。</p> <p>[abcde] 可以简写为[a-e]。</p> <p>另外，用感叹号作为列表的第一个字符可以起到反意作用，如：[!xyz] 表示匹配x、y、z 以外的任意一个字符。</p>

- 通配符 “*” 的常用方法就是查找具有相同扩展名的文件

ls *.tar.gz

rm *.old

- 问号通配符 “?” 必须匹配一个且只能匹配一个字符，通常用来查找比* 更为精确的匹配。

ls *.???

- 方括号通配符使用括号内的字符作为被匹配的字符，且只能匹配其中的一个字符。如列出以 **a**、**b**、**c** 开头，且以 **.dat** 为扩展名的所有文件：

ls [abc]*.dat

- 可以在方括号中使用连字符- 来指定一个范围，如列出以字母开头，数字结尾的所有文件：

ls [a-zA-Z]*[0-9]

□ 命令行自动补齐功能

- 通常用户在**bash**下输入命令时不必把命令输全，**shell** 就能判断出你所要输入的命令。
- 该功能的**核心思想**是：**bash**根据用户已输入的信息来查找以这些信息开头的命令，从而试图完成当前命令的输入工作。用来执行这项功能的键是**Tab** 键，
- 按下一次**Tab** 键后，**bash** 就试图完成整个命令的输入
- 按下两次**Tab** 键后，**bash** 将列出所有能够与当前输入字符相匹配的命令列表。

□ 管道

- UNIX 系统的一个基本哲学是：一连串的小命令能够解决大问题。其中每个小命令都能够很好地完成一项单一的工作。现在需要有一些东西能够将这些简单的命令连接起来，这样管道就应运而生。
- 管道 “|” 的基本含义是：将前一个命令的输出作为后一个命令的输入
- **ls /local | du -sh ***
- 利用管道可以实现一些很强的功能。

□ 管道举例

一个较复杂的例子：输出系统中用户名的一个排序列表。这里需要用到三个命令：**cat**、**awk**、**sort**，其中**cat**用来显示文件 **/etc/passwd** 的内容，**awk** 用来提取用户名，**sort** 用来排序。

```
cat /etc/passwd | awk -F: '{print $1}' | sort
```

在bash 中，有些字符具有特殊含义，通常称为特殊字符。

字符	含义	字符	含义
'	强引用	*, ?, !	通配符
"	弱引用	<, >, >>	重定向
\	转义字符	-	选项标志
\$	变量引用	#	注释符
;	命令分离符	空格、换行符	命令分隔符
`	命令替换：反引号中的字符串被 shell 解释为命令，在执行时， shell 首先执行该命令，并以它的标准输出取代整个反引号 (包括两个反引号) 部分		

□ **Shell** 变量大致可以分为三类：内部变量、用户变量和环境变量。

- 内部变量：由系统提供，用户不能修改。
- 用户变量：由用户建立和修改，在**shell** 脚本编写中会经常用到。
- 环境变量：这些变量决定了用户工作的环境，它们不需要用户去定义，可以直接在**shell** 中使用，其中某些变量用户可以修改。超算系统中常用的环境变量如下：

- PATH
- LD_LIBRARY_PATH
- C_LIBRARY_PATH

加载方式：

```
export PATH=/path/to/bin:$PATH
```

□ bash 配置文件

在命令行中设置和修改的变量值，只在当前的**shell** 中有效。一旦用户退出**bash**，所做的一切改变都会丢失。在启动交互式会话过程中，在出现提示符前，系统会读取几个配置文件，并执行这些文件中的命令。所以这些文件可以用来定制**bash** 环境。如：设置**shell** 变量值或建立别名等。

`/etc/profile`

`~/.bash_profile`
`~/.bash_login`
`~/.profile`

`~/.bashrc`

通常，个人**bash** 环境设置都定义在`~/.bashrc` 文件里

□ nohup 命令

用户退出系统后能继续运行的进程通常当用户退出系统后，所有属于该用户的进程将全部被终止。如果希望程序在退出系统后仍然能够继续运行，需使用 **nohup** 命令后台启动该进程。

- 用法： **nohup** 命令[选项] [参数] &

□ 查看正在运行的程序： **ps**

- **ps** [选项]
- 常用： **ps -ef**

□ 终止前台进程使用： **Ctrl+c**

□ 终止后台进程： **kill**

- 正常结束：**kill pid**
- 强制结束：**kill -9 pid**

□ Vim 编辑器

- 目前最流行的**Vi** 克隆版本，功能强大，扩展性强
- 支持目前绝大部分平台：**Windows, Linux, Unix, MacOS ...**
- **Vim** 没有菜单，只有命令

□ Vim 的工作模式

- **Normal** 模式(正常模式、编辑模式)
- **Insert** 模式（插入模式）
- **Command** 模式（命令模式、末行模式）

□ Normal 模式

- 启动**vim** 编辑器时，首先进入**Normal** 模式
- 在该模式下，用户可以输入各种合法的**vim** 命令来管理自己的文档
- 在该模式下，从键盘上输入的任何字符都被当做编辑命令来解释

□ 注意：

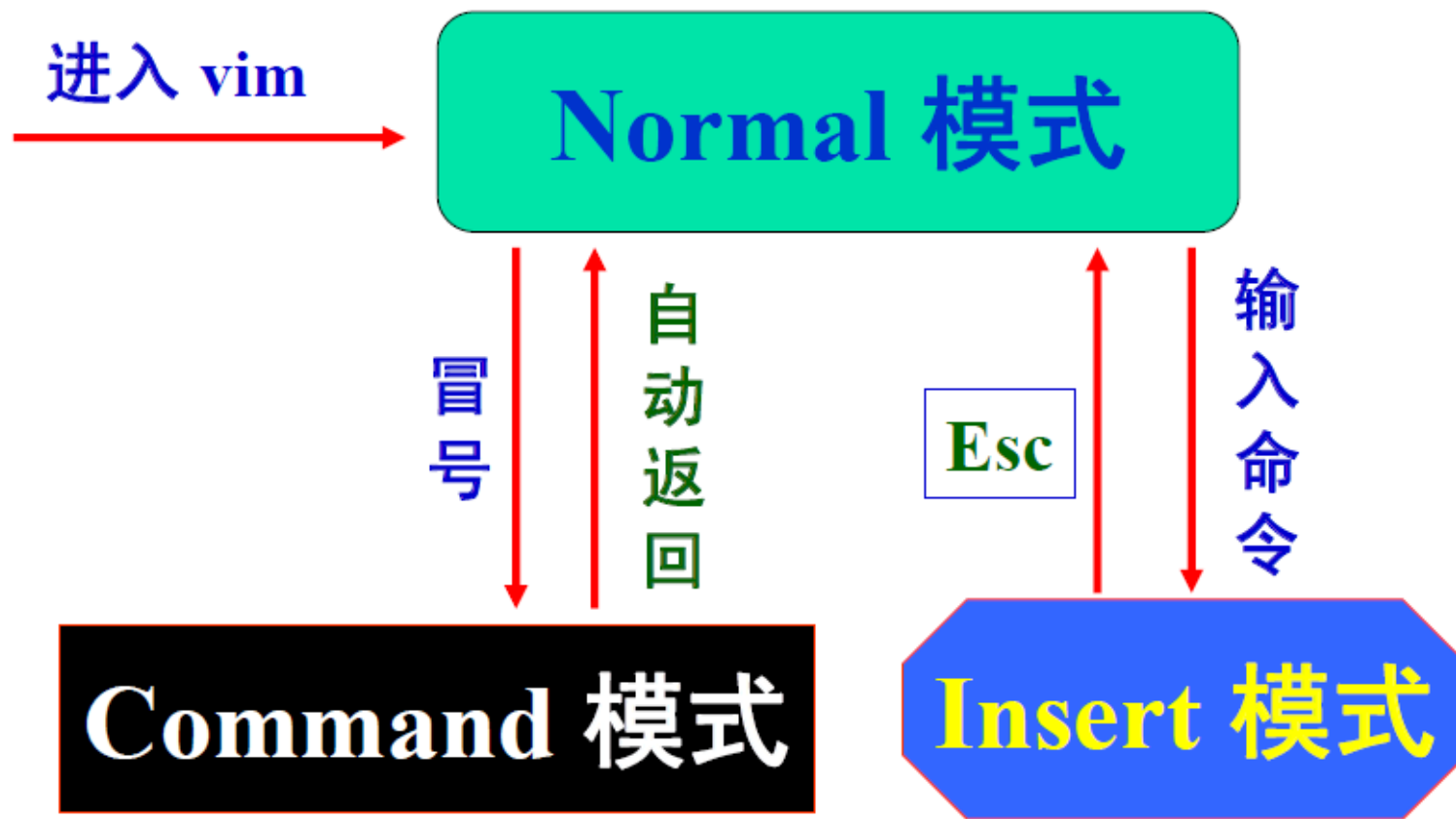
- 在**Normal** 模式下输入的命令不会在屏幕上显示
- 从其它模式进入**Normal** 模式：**Esc** 键
- 为确保进入**Normal** 模式，可多按几次**Esc** 键

□ Insert 模式

- 在该模式下，用户输入的任何字符都被vim 当做文件内容保存起来，并将其显示在屏幕上从**Normal** 模式进入**Insert** 模式：
 - 插入命令**i** 或**I**
 - 附加命令**a** 或**A**
 - 打开命令**o** 或**O**

□ Command 模式

- 从**Normal** 模式进入**Command** 模式： 冒号：
- 在该模式下，**vim** 会在窗口的最后一行显示一个冒号，作为**Command** 模式的提示符，等待用户输入命令
- 多数文件管理都是在此模式下执行的(如保存文件等)
- **Command** 模式中所有的命令都必须按<回车>后执行
- 命令执行完后，**vim** 自动回到**Normal** 模式
- 若在**Command** 模式下输入命令过程中改变了主意，可按**Esc** 键回到**Normal** 模式下



无法从**Command** 模式直接进入**Insert** 模式，也无法从**Insert** 模式直接进入**Command** 模式

□ 退出vim

- 连接两次大写字母**Z**，若文件被修改过，则保存后退出
- **:w** 保存当前编辑文件，但并不退出
- **:q** 系统退出**vim** 返回到**shell**
- **:wq** 表示存盘并退出
- **:q!** 放弃所作修改而直接退到**shell** 下
- **:x** 同**Normal** 模式下的**ZZ** 命令功能相同
- **:w fname** 另存为，新文件名为**fname**

- Vim文本编辑器中，光标的移动操作无疑是最经常使用的操作了。用户只有熟练地使用移动光标的这些命令，才能迅速准确地到达所期望的位置处进行编辑。
 - **Insert** 模式下光标的移动：键盘上的四个方向键
 - **Normal** 模式下光标的移动：四个方向键+各种命令
- Normal** 模式移动光标更快捷方便

- **Normal** 模式下的光标的快速移动方式

^	光标移到所在行的开始
\$	光标移到所在行的末尾
nG	光标移到第 n 行，若不指定 n ，则移到最后一行
gg	光标移到第一行
-	光标移到上面一行的开始
+	光标移到下面一行的开始
w	光标右移一个单词
b	光标左移一个单词
/word	光标移动到向下第一个搜索到的word处

❑ 取消上一命令，也称复原命令，是非常有用的命令，它可以取消前一次的误操作或不合适的操作对文件造成的影响，使之回复到这种误操作或不合适操作被执行之前的状态。

- 命令模式下使用 **u**

❑ 重复命令也是一个非常常用的命令。在文本编辑中经常会碰到需要机械地重复一些操作，这时就需要用到重复命令。它可以让用户方便地再执行一次前面刚完成的某个复杂的命令。

- 命令模式下使用 **Ctrl+r**

1. Linux是一款功能强大、安全性高、操作便捷、潜力巨大的操作系统；
2. 实践是学习Linux最好的方法；
3. 使用超算系统可能并不需要太强的Linux技术，但熟练掌握Linux技术能体会到无限可能性；

