高能物理计算暑期学校
IHEP School of Computing

LINUX基础及高能物理计算集群环境

主讲老师：郑伟

2022-08-19

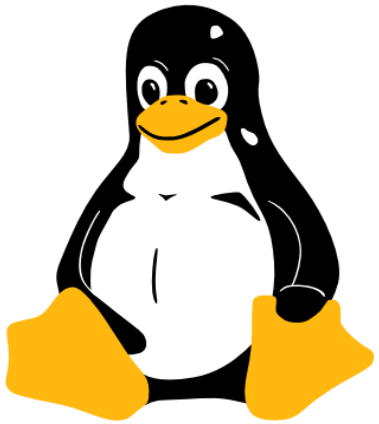
- 主讲内容
  - Linux操作系统的发展及其应用、Linux的主要特点和版本，Linux的启动、停止、Linux命令大全等介绍
  - 高能物理集群环境介绍
- 学完后你将收获
  - 了解Linux操作系统发展历史、应用现状和主要特点
  - 熟悉Linux主要操作命令，对计算集群有初步的认识

目录
CONTENTS

1 LINUX系统发展及应用
2 LINUX特点与版本
3 LINUX目录结构
4 LINUX命令大全

LINUX的发展史

- LINUX的历史最早要追溯到1991年，LINUX的核心原型是由芬兰赫尔辛基大学的一名叫Linus Torvalds的学生开发的
- LINUX是一套免费试用和自由传播的类UNIX操作系统
  - 自由软件基金会（Free Software Foundation，FSF）
  - GNU计划
  - 通用公共许可证（General Public License, GPL）原则



GNU/Linux

- 1991年Linus在comp.os.minix

新闻组中发布了Linux 0.02版本(kernel)

- 为了兼容已有的Unix软件，Linux的开发参照了POSIX（Portable Operating System Interface）标准
- 1994年Linux 1.0版本发布，并正式使用GPL协议

LINUX的应用现状

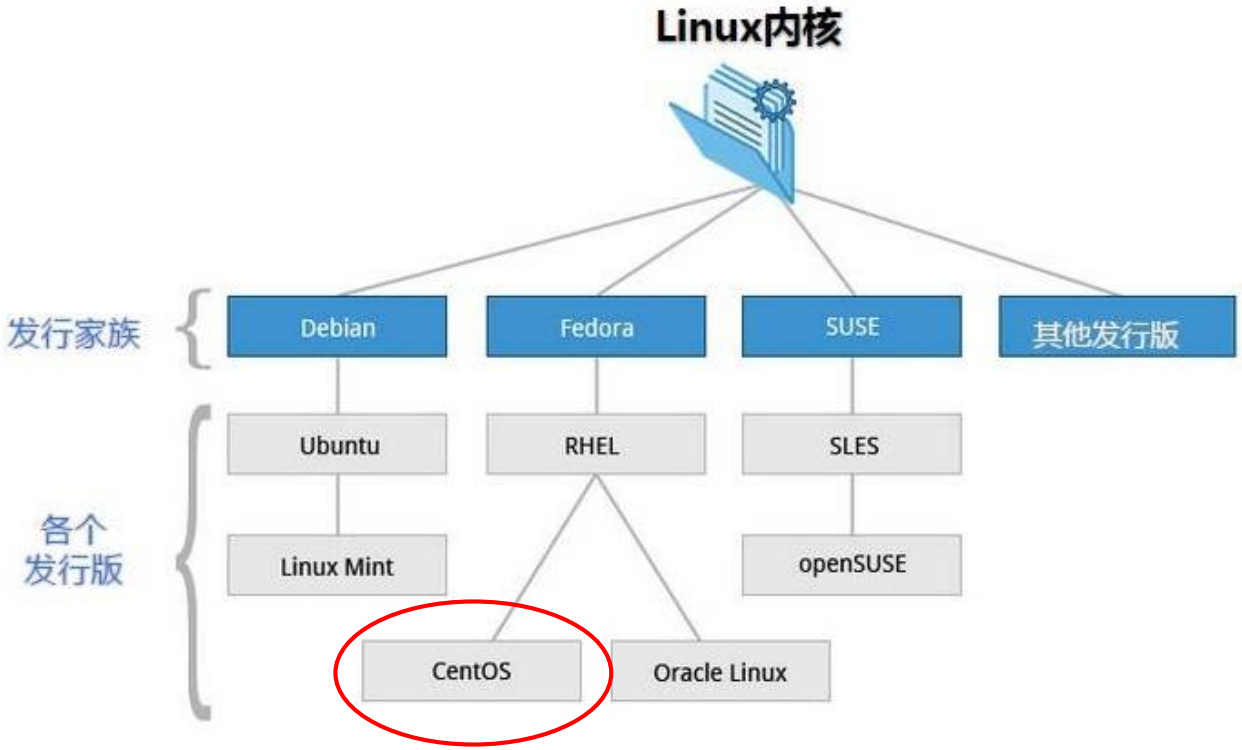
- LINUX 的应用范围主要包括桌面、工作站、服务器、嵌入式系统等方面
  - 主流的服务器产品
    - RHEL（Redhat Enterprise Linux）
    - SLES (Suse Linux Enterprise Server)
    - Red Flag Asianux Server
    - 中标普华服务器
  - 主流的桌面产品
    - Fedora
    - Ubuntu
    - Suse
    - RedFlag
    - Linpus
  - 嵌入式领域
    - 机顶盒
    - 数字电视
    - 网络电话
    - 程控交换机
    - 手机
    - PDA

Linux特点

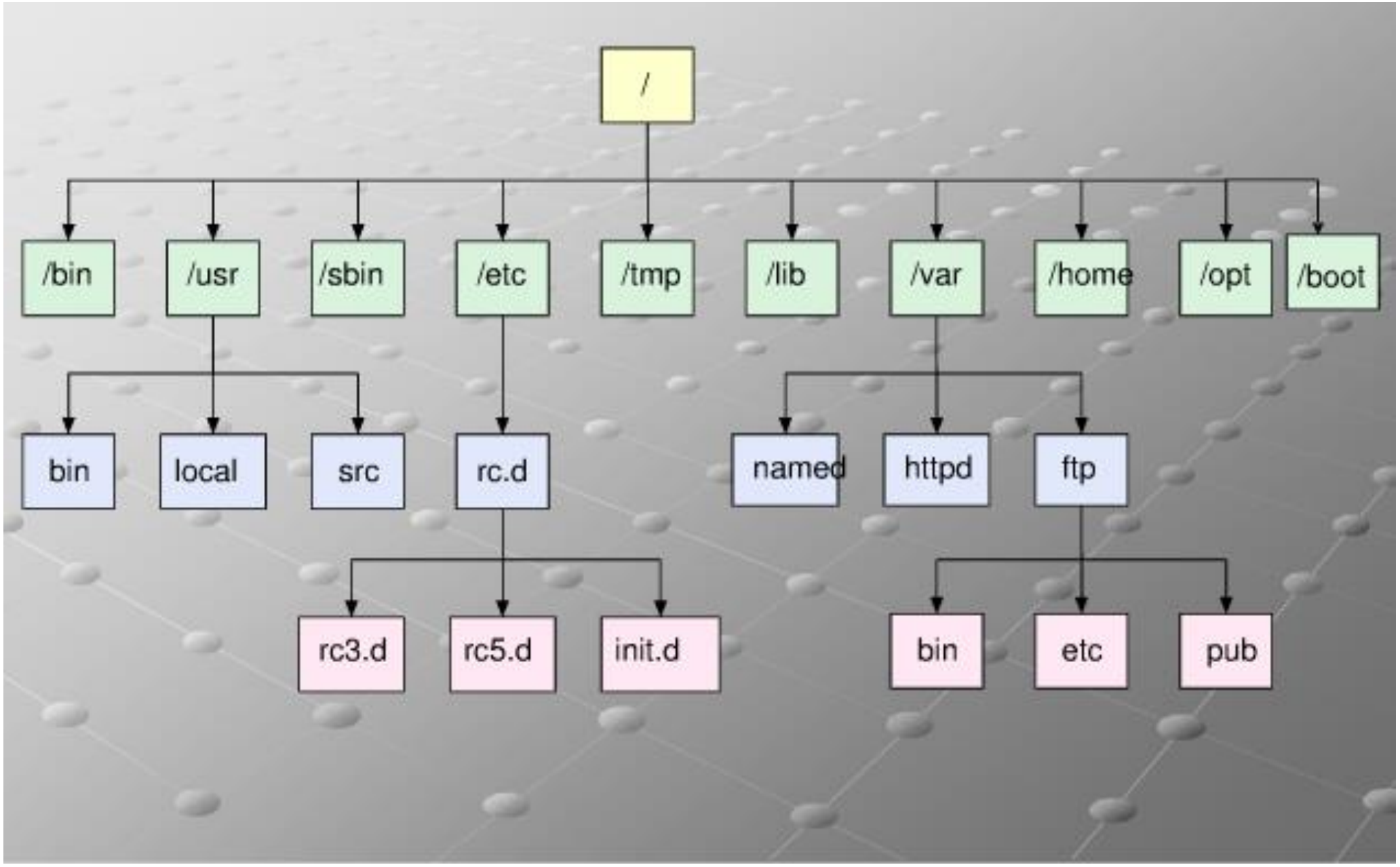
- 免费或少许费用的自由软件（基于GPL协议，自由获得Linux）
- 良好的硬件平台可移植性
- 稳定的系统（Linux继承了UNIX稳定并且有效率的特点）
- 完全符合POSIX标准
- 多用户多任务
- 丰富的应用程序和开发工具
- 良好的安全想和稳定性（Linux社区参与者众多，及时更新）
- 强大的网络功能
- 图形化界面

Linux的发行版本

- 知名的发行版本：Ubuntu、RedHat、CentOS、Debian、Fedora、SuSE、OpenSUSE、Arch Linux、SolusOS



LINUX目录结构


/
/bin
/usr
/sbin
/etc
/tmp
/lib
/var
/home
/opt
/boot
bin
local
src
rc.d
named
httpd
ftp
rc3.d
rc5.d
init.d
bin
etc
pub

常见的目录表示方法

- / 代表根目录
- . 代表当前目录或者本目录
- .. 代表当前目录的上级目录或者父目录
- ~ 代表当前用户的主目录
- 目录名 代表当前目录下的目录
- /目录名 代表根目录下的目录
- /目录名1/目录名2 代表根目录下目录名1下的目录名2

Linux 主要系统命令

- 用户和组
- 文件操作
- 存储命令
- 网络命令
- 软件管理
- 进程管理
- 系统服务
- ……

用户和组-ID

每个用户账号都拥有一个唯一的用户名和各自的口令。用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。

用户组是具有相同特征用户的逻辑集合。简单的理解，有时我们需要让多个用户具有相同的权限，比如查看、修改某一个文件、文件夹的权限，一个用户可以存在多个用户组

- whoami：命令用于显示自身用户名称

格式：whoami

```
[zhengw@lxslc703 ~]$ whoami
zhengw
```

- id： id命令用于显示用户的UID，以及所属群组的GID

格式：id

```
[zhengw@lxslc703 ~]$ id
uid=60020(zhengw) gid=600(u07) groups=600(u07),1010(buaa),1020(pku),1040(manager)

[zhengw@lxslc703 ~]$ ll /bes3fs
total 12
drwxr-xr-x 3 mdccal  physics 4096 Feb 27  2019 cgemCosmic
drwxr-xr-x 7 offline physics 4096 Dec 31  2018 offline
```

用户和组-HOME

- home目录：用户的主目录,也是登陆系统后默认目录

```
[zhengw@lxslc703 ~]$ echo $HOME
/afs/ihep.ac.cn/users/z/zhengw
```

- passwd： passwd用于修改用户口令（对于IHEP平台，需要在login.ihep.ac.cn页面进行修改）

```
[zhengw@lxslc703 ~]$ id
uid=60020(zhengw) gid=600(u07)
groups=600(u07),1010(buaa),1020(pku),1040(manager)
```

用户SHELL

➢LINUX 中的SHELL

一般用户，不能直接使用 LINUX kernel 。而是通过 kernel 的 “ 外壳 ” 程序，也就是所谓的 shell ，来与 kernel 沟通。

Shell 的最简单定义：命令行解释器（ command Interpreter ）主要包含：将使用者的命令翻译给核心（ kernel ）处理。同时，将核心的处理结果翻译给使用者。Shel解释用户输入的命令，将它传递给内核，调用其他程序，给其他程序传递数据或参数，并获取程序的处理结果。Shell本身也可以被其他程序调用。

目前Shell的版本有很多种，如sh(BourneShell)、csh(Cshell)、Bash、ksh、tcsh等，各有特点。

Shell的两种主要语法类型有Bourne和C，这两种语法彼此不兼容。Bourne家族主要包括sh、ksh、Bash、psh、zsh；C家族主要包括：csh、tcsh

➢查看用户SHELL

```
[zhengw@lxslc703 ~]$ echo $SHELL
/bin/bash
```

➢切换SHELL

```
[zhengw@dws147 ~]$ sh
sh-4.2$ csh
Good luck , I am tcsh
[zhengw@dws147 ~]$ zsh
[zhengw@dws147]~%
```

➢修改用户默认SHELL

```
[zhengw@dws147]~% chsh
Changing shell for zhengw.
New shell [/bin/bash]: /bin/tcsh
Password:
Shell changed.
```

Linux 主要系统命令

- 用户和组
- 文件操作
- 存储命令
- 网络命令
- 软件管理
- 进程管理
- 系统服务
- ……

Linux 文件操作命令

- pwd 查看当前所在目录
- mkdir 创建一个目录
- touch 创建一个文件
- chmod 修改文件属性
- chown 设置文件属主
- ls 查看文件
- cp 复制文件
- scp 文件传输
- mv 移动文件
- rm 移除文件
- cd 转到某个目录
- cat 查看文件内容
- more 详细信息
- find 文件查找
- grep 字符串查找
- tar 文件压缩
- vi/vim 文件编辑
- ln 文件链接

显示或查看目录的内容

用法1：ls 显示当前目录下的内容

```
[root@lxslc716 ~]# ls
anaconda-ks.cfg  bin  check_puppet.sh  install.post.log  install.postnochroot.log  original-ks.cfg  restart_puppet.sh  sen
[root@lxslc716 ~]#
```

用法2: ls 目录名 显示指定目录下的内容

```
[root@lxslc716 ~]# ls /usr/
bin  etc  games  include  java  lib  lib64  libexec  local  lxslc  sbin  share  spool  src  tmp  vice
[root@lxslc716 ~]#
```

Linux系统的超级用户一般命名为root。
root是系统中唯一的超级用户，具有系统中所有的权限，如启动或停止一个进程，删除或增加用户，增加或者禁用硬件等等。

显示或查看目录的内容

用法3：ls –l 以详细资料显示当前目录下的内容

```
[root@lxslc716 ~]# ls -l /usr/
total 548
dr-xr-xr-x.   2 root root 118784 Aug  2 04:47 bin
drwxr-xr-x.   2 root root      6 Apr 11  2018 etc
drwxr-xr-x.   2 root root      6 Apr 11  2018 games
drwxr-xr-x. 229 root root  20480 Feb 23 16:40 include
drwxr-xr-x    4 root root     81 Apr  7 17:02 java
dr-xr-xr-x.  57 root root   4096 Aug  5  2020 lib
dr-xr-xr-x. 220 root root 176128 Aug  2 04:47 lib64
drwxr-xr-x.  67 root root  12288 Aug  1 12:17 libexec
drwxr-xr-x.  17 root root    218 Nov 13  2020 local
drwxr-xr-x    4 root root     29 Aug  4  2020 lxslc
dr-xr-xr-x.   2 root root  28672 Aug  2 04:47 sbin
drwxr-xr-x. 361 root root  12288 Jan 12  2021 share
drwxr-xr-x    4 root root     31 Nov 13  2020 spool
drwxr-xr-x.   6 root root    199 Aug  5  2020 src
```

用法4：ls –a 显示当前目录下的隐藏内容

```
[root@lxslc716 ~]# ls -a
.                .bash_history  .bashrc  check_puppet.sh  install.post.log           .local          restart_puppet.sh  .ssh
..               .bash_logout   bin       .config          install.postnochroot.log  original-ks.cfg  .rpmmacros         .tcs
anaconda-ks.cfg  .bash_profile  .cache    .cshrc           .lesshst                  .pki             sent               .vim
[root@lxslc716 ~]#
```

显示或查看目录的内容

用法5：ls –ld 目录名　　显示指定目录的属性

```
[root@lxslc716 ~]# ls -ld /usr
drwxr-xr-x. 17 root root 205 Dec  9  2020 /usr
[root@lxslc716 ~]#
```

用法6：ll=ls -l

```
[root@lxslc716 ~]# ll /usr
total 548
dr-xr-xr-x.   2 root root 118784 Aug  2 04:47 bin
drwxr-xr-x.   2 root root      6 Apr 11  2018 etc
drwxr-xr-x.   2 root root      6 Apr 11  2018 games
drwxr-xr-x. 229 root root  20480 Feb 23 16:40 include
drwxr-xr-x    4 root root     81 Apr  7 17:02 java
dr-xr-xr-x.  57 root root   4096 Aug  5  2020 lib
dr-xr-xr-x. 220 root root 176128 Aug  2 04:47 lib64
drwxr-xr-x.  67 root root  12288 Aug  1 12:17 libexec
drwxr-xr-x.  17 root root    218 Nov 13  2020 local
drwxr-xr-x    4 root root     29 Aug  4  2020 lxslc
dr-xr-xr-x.   2 root root  28672 Aug  2 04:47 sbin
drwxr-xr-x. 361 root root  12288 Jan 12  2021 share
drwxr-xr-x    4 root root     31 Nov 13  2020 spool
drwxr-xr-x.   6 root root    199 Aug  5  2020 src
```

cd：更改目录或者进入目录

用法1：cd /　进入根目录

```
[root@lxslc716 /]# cd
[root@lxslc716 ~]# cd /
[root@lxslc716 /]#
```

用法2：cd 目录 进入指定目录

```
[root@lxslc716 /]# cd /tmp/
[root@lxslc716 tmp]#
```

用法3：cd 进入当前用户的主目录

```
[root@lxslc716 tmp]# cd
You have mail in /var/spool/mail/root
[root@lxslc716 ~]#
```

用法4：cd ~　当前用户的主目录

```
[root@lxslc716 tmp]# cd ~
[root@lxslc716 ~]#
```

mkdir：创建目录

格式：　mkdir [- p] fileName

用法1：mkdir 目录名 在当前目录下创建目录
用法2：mkdir 目录名/目录名1 在指定目录下创建目录
用法3：mkdir 目录名1 目录名2 目录名3 在指定目录下创建多个并列的目录
用法4：mkdir -p 目录名5/目录名6 在没有父目录的情况下同时创建父目录和子目录

cp：复制

- cp(copy)命令可以将文件从一处复制到另一处
- 一般在使用cp命令时将一个文件复制成另一个文件或复制到某目录时，需要指定源文件名与目标文件名或目录

格式：cp[参数]<源文件路径><目标文件路径>
用例1： 将test1.text复制成test2.txt
[test@linux test]$ cp test1.txt test2.txt

用例2：将test3.txt复制到/tmp目录中
[test@linux test]$ cp test3.txt /tmp

用例3： 加-r参数，拷贝目录
[test@linux test]$ cp –r test1(目录) test2(目录)

scp 文件远程拷贝

scp 命令用于 Linux 之间复制文件和目录。scp 是 secure copy 的缩写, scp 是 linux 系统下基于 ssh 登陆进行安全的远程文件拷贝命令。scp 是加密的，rcp 是不加密的，scp 是 rcp 的加强版。

scp [-1246BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port] [-S program][[user@]host1:]file1 [...] [[user@]host2:]file2

-1/2： 强制scp命令使用协议ssh1/ssh2

-4/6： 强制scp命令只使用IPv4/IPv6寻址

-C： 允许压缩。（将-C标志传递给ssh，从而打开压缩功能）

-p：保留原文件的修改时间，访问时间和访问权限。

-q： 不显示传输进度条。

-r： 递归复制整个目录。

-v：详细方式显示输出

-P port：注意是大写的P, port是指定数据传输用到的端口号

实例：

scp /home/space/music/1.mp3 www.runoob.com:/home/root/others/music/

scp -r /home/space/music/ username@www.runoob.com:/home/root/others/

scp -P 4588 username@www.runoob.com:/home/sin.sh /home/administrator/

rm：删除

功能：删除文件或目录 格式：rm[参数] <目标文件路径>
用例1：删除一个文件
[test@linux test]$ rm myfiles
用例2：删除当前目录下的所有文件
[test@linux test]$ rm *
用例3： -f参数， 强迫删除所有以后缀名为txt文件
[test@linux test]$ rm –f *.txt
用例4：-r参数
-r参数：删除目录，等同于rmdir命令
[test@linux test]$ rm –r disk1
用例5：-i参数：删除文件时询问，删除当前目录下的所有文件
[test@linux test]$ rm –i *

注意：在系统的默认状态下，rm命令会对每个删除的文件一一询问。如果用户确定要删除这些文件，则可以使用参数-f来避免询问

cat：查看文件

功能：用于显示文件的内容,也可以将数个文件合并成一个文件
格式：cat[参数]<文件名>

用例1：显示test.txt文件内容
[test@linux test]$cat test.txt

用例2：逐页显示test.txt文件中的内容
[test@linux test]$cat test.txt | more

用例3：将test.txt的内容附加到test1.txt文件之后
[test@linux test]$cat test.txt >>test1.txt

用例4：将test.txt和test2.txt文件合并成readme.txt文件
[test@linux test]$cat test.txt test2.txt >readme.txt

more：查看文件内容

- more命令是一般用于要显示的内容会超过一个画面长度的情况
- 为了避免画面显示时瞬间就闪过去，用户可以使用more命令，让画面在显示满一页时暂停，此时可按空格健继续显示下一个画面，或按Q键停止显示

用例1：显示 etc/named.conf文本文件的内容
[test@linux test]$ more /etc/named.conf

用例2：当用ls命令查看文件列表时，如文件太多，则可配合more命令使用
[test@linux etc]$ ls -al | more

PS：以长格形式显示etc目录下的文件列表，显示满一个画面便暂停，可按空格键继续显示下一画面，或按Q键跳离

less：查看文件内容

- less命令的用法与more命令类似，也可以用来浏览超过一页的文件
- 所不同的是less命令除了可以按空格键向下显示文件外，还可以利用上下键来卷动文
- 当要结束浏览时，只要按Q键即可

用例1：显示/etc/named.conf的文本文件内容

```
[test@linux etc]$less  named.conf
```

用例2：以长格形式列出/etc目录中所有的内容，用户可按上下键浏览或按Q键结束

```
[test@linux etc]$ls  -al | less
```

head、tail(尾)命令

- head功能：用于显示文件前几行的内容
  格式：head[参数]<文件名>

用例： 显示/etc/passwd/文件的前10行内容
[root@linux root]# head -10 /etc/passwd

- tail功能：用于显示文件后几行的内容
  格式：tail[参数]<文件名>

用例1： 显示文件的倒数10行内容
[root@linux root]# tail -10 /etc/passwd
用例2：显示文件的从第10行开始到末尾的内容
[root@linux root]# tail +10 /etc/passwd
用例3：实时输出文件内容
[root@linux root]# tail -f nohup.out

mv：移动或更改目录、文件名称

- 功能：可以将文件及目录移到另一目录下，或更改文件及目录的名称

  格式：[参数]<源文件或目录> <目标文件或目录>

用例1：将a.txt文件移动上层目录

[test@linux dir1]$ mv a.txt ../

用例2： 将a.txt改名为b.txt

[test@linux dir1]$ mv a.txt b.txt

用例3： 将dir2目录上移一层

[test@linux dir1]$ mv dir2 ../

pwd：显示当前所在的目录

功能：显示用户正在工作或当前所在的目录

格式：pwd

用例：显示用户test所在的当前目录是/home/test

```
[test@linux test]$ pwd
  /home/test
```

touch：创建文件

touch 修改文件时间或创建文件

- 格式：touch [- a c d t] fileName
- 参数说明
  - -a 修改访问时间
  - -c 如果没有文件不创建不加此参数在没有文件时创建文件
  - -d 将一组字符串转换成时间
  - -t 时间格式，[YY]MMDDhhmm[.ss]

用例1：修改filename的时间为当前时间，如果没有文件则自动创建
[test@linux dir1]$ touch filename

用例2：修改filename的时间为当前时间，如果没有文件则不创建
[test@linux dir1]$ touch -c filename

用例3：按MMDDhhmm的格式转换字符串并修改filename的时间
[test@linux dir1]$ touch –d 121212 filename

chmod：修改文件权限

chmod 修改文件或目录权限

- 用法

  chmod [0-7][0-7][0-7] [u g o] [+ xwr] filename

- 说明

  当我们通过 ll filename查看文件名的时候，系统会在最前面显示出这个文件的十个属性。这十个属性分为四部分；如drwxrwxrwx;这个十个属性

  - 第一个表示文件类型，如d表示这个文件是一个目录
  - 第一组rwx表示所有者的权限
  - 第二组rwx表示所有者所在组的权限
  - 第三组rwx表示其他所有人的权限

  PS：rwx则分别表示读、写和执行权限

chown命令

- LINUX是多用户多任务操作系统，所有文件都有属主
- chown命令可以指定文件的属主（用户和组）
- chown需要超级用户root或是应用系统的管理员的权限才能执行此命令
- 语法

  chown [-cfhvR] [--help] [--version] user[:group] file...

  参数:
  - user : 新的文件拥有者的使用者 ID
  - group : 新的文件拥有者的使用者组(group)
  - -c : 显示更改的部分的信息
  - -f : 忽略错误信息
  - -h :修复符号链接
  - -v : 显示详细的处理信息
  - -R : 处理指定目录以及其子目录下的所有文件
  - --help : 显示辅助说明
  - --version : 显示版本

find：文件查找

作用：find命令的作用是在目录中搜索文件，它的使用权限是所有用户

格式：find [path][options][expression]

path指定目录路径，系统从这里开始沿着目录树向下查找文件。它是一个路径列表，相互用空格分离，如果不写path，那么默认为当前目录。

[options]参数：

- depth：使用深度级别的查找过程方式，在某层指定目录中优先查找文件内容。

[expression]是匹配表达式，是find命令接受的表达式，find命令的所有操作都是针对表达式的。

- name：支持统配符*和?
- atime n：搜索在过去n天读取过的文件
- ctime n：搜索在过去n天修改过的文件
- print：输出搜索结果，并且打印

用例：搜索系统根目录下名为test.conf的文件

```
[root@linux root]# find / -name test.conf
/etc/httpd/conf/test.conf
```

显示搜索结果

grep：在文件中搜索匹配的字符并进行输出

格式：grep[参数] <要找的字串> <要寻找字 串的源文件>

用例1：搜索test.txt文件中字符串linux并输出

```
[root@linux root]# grep linux test.txt
```

用例2：搜索rpm包中含有httpd包的文件名

```
[root@linux root]# rpm  -qa | grep puppet
```

tar：Linux的压缩解压缩命令

tar

- -c: 建立压缩档案
- -x：解压
- -t：查看内容
- -r：向压缩归档文件末尾追加文件
- -u：更新原压缩包中的文件
- 这五个是独立的命令，压缩解压都要用到其中一个，可以和别的命令连用但只能用其中一个，下面的参数是根据需要在压缩或解压档案时可选的
- -z：有gzip属性的
- -j：有bz2属性的
- -Z：有compress属性的
- -v：显示所有过程
- -O：将文件解开到标准输出

tar：Linux的压缩解压缩示例

压缩示例

用例1：将目录里所有jpg文件打包成tar.jpg
tar –cvf jpg.tar *.jpg

用例2：将目录里所有jpg文件打包成jpg.tar后，并且将其用bzip2压缩，生成一个bzip2压缩过的包，命名为jpg.tar.bz2
tar –czf jpg.tar.gz *.jpg
tar –cjf jpg.tar.bz2 *.jpg

用例3：将目录里所有jpg文件打包成jpg.tar后，并且将其用compress压缩，生成一个umcompress压缩过的包，命名为jpg.tar.Z
tar –cZf jpg.tar.Z *.jpg

用例4： rar格式的压缩，需要先下载rar for linux
rar a jpg.rar *.jpg

用例5： zip格式的压缩，需要先下载zip for linux
zip jpg.zip *.jpg

tar：Linux的压缩解压缩命令

解压示例

用例1：解压 tar包
tar –xvf file.tar
用例2：解压tar.gz包
tar -xzvf file.tar.gz
用例3：解压 tar.bz2
tar -xjvf file.tar.bz2
用例4：解压tar.Z
tar –xZvf file.tar.Z
用例5：解压rar
unrar e file.rar
用例6：解压zip
unzip file.zip

tar：Linux的压缩解压缩命令

总结

*.tar 用 tar –xvf 解压

*.gz 用 gzip -d或者gunzip 解压

.tar.gz和.tgz 用 tar –xzf 解压

*.bz2 用 bzip2 -d或者用bunzip2 解压

*.tar.bz2用tar –xjf 解压

*.Z 用 uncompress 解压

*.tar.Z 用tar –xZf 解压

*.rar 用 unrar e解压

*.zip 用 unzip 解压

文本编辑器vi/vim

- vi，vim是Linux下的全屏幕文本编辑程序

  优点：编辑功能强大

  缺点：操作复杂

- vi的工作模式

  输入(编辑)模式

  命令(末行)模式

- vi的工作模式转换

  命令模式(默认模式)转换为输入模式

  Insert键，a(A),i(I),o(O)键

  输入模式转换为命令模式

  Esc键

vi的基本操作

- vi文件名 打开文件 保存文件
  - 命令模式下：

    :w　　保存
    :w!　　强制保存
    :w 文件名　　另存为

- 退出vi
  - 命令模式下：

    :q　　退出
    :q!　　强制退出
    :wq　保存并退出
    :wq!　强制保存并退出

vi的基本操作

- 移动光标

  上(J)下(K)左(h)右(l)方向键

- 修改文本文件内容

  输入模式

  输入：键盘

  删除：del或退格键

- 列出/取消文本内容的行号

  命令模式

  :set nu

  :set nonu

vi的基本操作

- vi显示颜色

  alias vi=vim

  - 使用命令

```
[root@RHEL5 ~]# alias vi=vim
[root@RHEL5 ~]#
```

  - 更改用户环境变量
    - 在当前用户主目录下修改.bashrc,添加一行
      alias vi=vim
    - 重启计算机

```
[root@RHEL5 ~]# vi .bashrc
[root@RHEL5 ~]# tail -1 .bashrc
alias vi=vim
[root@RHEL5 ~]#
```

vi的编辑命令

dd 删除当前行
nx 删除由光标位置起的n个字符（含光标位置）
nX 删除由光标位置起的n个字符（不含光标位置）
d0 删除到某一行的开始位置
dw 删除到某个单词的结尾位置
db 删除到某个单词的开始位置
d} 删除到段落的结尾位置
d{ 删除到段落的开始位置
5d d 删除由当前行开始的5行内容
dL 删除直到屏幕上的最后一行内容
yy或Y 把一行文本复制到寄存器中
dd剪切一行
P或p粘贴一行
u撤消当前的操作

ln：在文件和目录之间建立链接

格式：ln [参数] <源文件或目录> <目标文件或目录>

链接分“软链接”和“硬链接”

1.软链接:

[root@linux test]# ln –s /usr/share/do doc

创建一个链接文件doc,并指向目录/usr/share/do

2.硬链接:

[root@linux test]# ln /usr/share/test hard

创建一个硬链接文件hard，这时对于test文件对应的存储区域来说，又多了一个文件指向它

软链接与硬链接的区别：

- 软链接:为同一个文件或目录创建2、3个名字，如同在Windows下创建的快捷方式
- 硬链接:让多个不在或者同在一个目录下的文件，同时能够修改同一个文件，其中一个修改后，所有与其有硬链接的文件都一起修改了

Linux 主要系统命令

- 用户和组
- 文件操作
- 存储命令
- 网络命令
- 软件管理
- 进程管理
- 系统服务
- ……

df命令

功能：用于查看文件系统的各个分区的占用情况

```
[root@localhost RPMS]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3             6.9G  1.5G  5.1G  22% /
/dev/sda1              99M  8.4M   86M   9% /boot
none                  252M     0  252M   0% /dev/shm
/dev/hdc              161M  161M     0 100% /media/cdrom
```

du：查看磁盘使用情况命令

功能：查看某个目录中各级子目录所使用的硬盘空间数

格式：du [参数] <目录名>

```
[root@ccopt ~]# du -h /tmp
0       /tmp/.X11-unix
0       /tmp/.XIM-unix
0       /tmp/.ICE-unix
0       /tmp/.Test-unix
0       /tmp/.font-unix
0       /tmp/tmux-13109
1.7G    /tmp/log/omat/topinfo
1.7G    /tmp/log/omat
1.7G    /tmp/log
```

Linux 主要系统命令

- 用户和组
- 文件操作
- 存储命令
- 网络命令
- 软件管理
- 进程管理
- 系统服务
- ……

netstat 命令

1.作用 :检查整个Linux网络状态。
2.格式 :netstat [-acCeFghilMnNoprstuvVwx][-A][--ip]
3.主要参数
-a--all：显示所有连线中的Socket。
-c--continuous：持续列出网络状态。
-e--extend：显示网络其它相关信息。
-g--groups：显示多重广播功能群组组员名单。
-i--interfaces：显示网络界面信息表单。
-l--listening：显示监控中的服务器的Socket。
-n--numeric：直接使用IP地址，而不通过域名服务器。
-r--route：显示Routing Table。

netstat 命令

-s--statistice：显示网络工作信息统计表

-t--tcp：显示TCP传输协议的连线状况

-u--udp：显示UDP传输协议的连线状况

-v--verbose：显示指令执行过程

-w--raw：显示RAW传输协议的连线状况

netstat –tnlp | grep 7010 ： 通过应用端口号查看进程ID

ifconfig：网络配置

网卡配置文件的目录

/etc/sysconfig/network-scripts

网卡的配置文件

ifconfig-ethX，ifconfig-ethX:X，ifconfig-wlanX, ifconfig-enoX

```
[root@ccopt ~]# cat /etc/sysconfig/network-scripts/ifcfg-eno1
# Generated by dracut initrd
NAME="eno1"
DEVICE="eno1"
ONBOOT="yes"
NETBOOT="yes"
UUID="54ea1d06-c646-48d9-9b20-428e753ecf7e"
IPV6INIT="yes"
BOOTPROTO="dhcp"
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
```

DEVICE——设备名
BOOTPROTO ——IP地址的获取方式(静态或者dhcp)
HWADDR ——MAC地址
ONBOOT ——开机启动
IPADDR ——IP地址

ifconfig：网络配置

配置网卡的IP地址：重启网卡命令

ifdown ethX；ifup ethX 重启指定网卡

service network restart 重启所有网卡

```
[root@RHEL5 /]# ifdown eth0
[root@RHEL5 /]# ifconfig eth0 192.168.10.238 netmask 255.255.255.0
[root@RHEL5 /]# ifup eth0
[root@RHEL5 /]# ifconfig eth0 |grep 'inet addr'
          inet addr:192.168.10.238  Bcast:192.168.10.255  Mask:255.255
.255.0
[root@RHEL5 /]#
```

```
[root@localhost ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Setting network parameters:                                [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
```

配置网卡的IP地址：修改配置文件

vi　网卡配置文件，修改完成后，重启网卡，修改才能生效

```
[root@RHEL5 /]# cd /etc/sysconfig/network-scripts/
[root@RHEL5 network-scripts]# ifdown eth0
[root@RHEL5 network-scripts]# vi ifcfg-eth0
[root@RHEL5 network-scripts]# cat ifcfg-eth0
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
BOOTPROTO=static
HWADDR=00:0C:29:F4:A1:6B
ONBOOT=yes
IPADDR=192.168.10.239
[root@RHEL5 network-scripts]# ifup eth0
[root@RHEL5 network-scripts]#
```

Linux 主要系统命令

- 用户和组
- 文件操作
- 存储命令
- 网络命令
- 软件管理
- 进程管理
- 系统服务
- ……

软件包管理命令(rpm)

- RPM的全名是Red Hat Package Manager,它是由Red Hat 公司所开发的一个开放性软件包管理系统。其优点：
  - 易于安装、升级与删除软件包
  - 有强大的查询功能
  - 能够进行软件包验证
  - 支持代码形式的软件包

安装rpm包

- 使用RPM命令的安装模式可以将软件包内所有的组件放到系统中的正确路径，安装软件包的命令是:

[root@linux root]#rpm –ivh vsftpd-2.6.2-x86-64.rpm

i：作用rpm的安装模式

v: 校验文件信息

h: 以＃号显示安装进度

删除rpm包

- 删除模式会将指定软件包的内容全部删除，但并不包括已更改过的配置文件，删除RPM软件包的命令如下：

  [root@linux /]# rpm –e vsftpd

  删除参数　软件包名称

注意：这里必须使用软件名“vsftpd”或”vstpd-2.6.2-x86-64而不是使用当初安装时的软件包名vsftpd-2.6.2-x86-64.rpm

升级rpm包

- 升级模式会安装用户所指定的更新版本，并删除已安装在系统中的相同软件包，升级软件包命令如下：

```
[root@linux /]# rpm –Uvh vsftpd-2.6.2-x86-64.rpm
```

更新rpm包

- 更新模式下，rpm命令会检查在命令行中所指定的软件包是否比系统中原有的软件包更新如果更新，rpm命令会自动更新指定的软件包；反之，若系统中并没有指定软件包的较旧版本，rpm命令并不会安装此软件包

```
[root@linux /]# rpm –Fvh vsftpd-2.6.2-x86-64.rpm
```

查询rpm包

- 若要获取RPM软件包的相关信息，可以使用查询模式
- 使用-q参数可查询一个已安装的软件包的内容

[root@linux /]# rpm –q vsftpd
vsftpd-2.6.2-x86-64 显示软件包的名称、版本及次版本

- 查询软件包所安装的文件
  rpm –ql package-name 命令可得知
  [root@linux /]# rpm –ql vsftpd (l参数：显示文件列表)

Linux 主要系统命令

- 用户和组
- 文件操作
- 存储命令
- 网络命令
- 软件管理
- 进程管理
- 系统服务
- ……

程序和进程

- 程序是一组指令的有序集合，是一个静态的实体，是存放在磁盘文件中的可执行文件
- 进程是一个动态的实体，他有自己的生命周期
- 进程和程序并不是一一对应的，一个程序执行在不同的数据集上就成为不同的进程
- 一个运行着的程序，可能有多个进程
  - 比如WWW服务器是apache服务器，当管理员启动服务后，可能会有好多人来访问，也就是说许多用户来同时请求httpd服务，apache服务器将会创建有多个httpd进程来对其进行服务
- LINUX的进程结构

栈：局部变量
堆：动态变量

- 代码段：存放程序代码
- 数据段：存放全局变量、常数及动态数据分配的空间
- 堆栈段：存放子程序的返回地址、子程序的参数记忆程序的局部变量

进程分类

- 进程一般分为交互进程、批处理进程和监控进程三类
- 交互进程：由一个Shell启动的进程，交互进程既可以在前台运行，也可以在后台运行
- 批处理进程：是一个进程序列
- 监控进程：也称系统守护进程，是Linux系统启动时运行的进程，并常驻后台
  - 比如在Fedora或Redhat中，我们可以定义httpd 服务器的启动脚本的运行级别，此文件位于/etc/init.d目录下，文件名是httpd，/etc/init.d/httpd 就是httpd服务器的守护程序，当把它的运行级别设置为3和5时，当系统启动时，它会跟着启动

ps 监视进程工具

ps 的参数说明;

l 长格式输出

u 按用户名和启动时间的顺序来显示进程

j 用任务格式来显示进程

f 用树形格式来显示进程

a 显示所有用户的所有进程（包括其它用户）

x 显示无控制终端的进程

r 显示运行中的进程

ww 避免详细参数被截断

ps aux 输出的解释

USER 进程的属主;
PID 进程的ID;
%CPU 进程占用的CPU百分比;
%MEM 占用内存的百分比;
VSZ 进程虚拟大小;
RSS 驻留中页的数量;
TTY 终端ID
STAT 状态
START 启动进程的时间;
TIME 进程消耗CPU的时间;
COMMAND 命令的名称和参数;

```
[root@localhost ~]# ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   2704   512 ?        S    09:22   0:01 init [5]
root         2  0.0  0.0      0     0 ?        SN   09:22   0:00 [ksoftirqd/0]
root         3  0.0  0.0      0     0 ?        S<   09:22   0:00 [events/0]
```

Kill和killall命令

kill的应用是和ps 或pgrep 命令结合在一起使用的

- kill [信号代码] 进程ID

注：信号代码可以省略；我们常用的信号代码是 -9 ， 表示强制终止
对于僵尸进程，可以用kill -9 来强制终止退出

例：

```
[root@localhost~]# pgrep -l test
5031 test
[root@localhost ~]# kill -9 5031
```

- killall 通过程序的名字，直接杀死所有进程

用法：killall 正在运行的程序名
killall 也和ps或pgrep 结合使用，比较方便
通过ps或pgrep 来查看哪些程序在运行

```
[root@localhost]#pgrep-l gaim
2979 gaim
[root@localhost]# killall gaim
```

Linux 主要系统命令

- 用户和组
- 文件操作
- 存储命令
- 网络命令
- 软件管理
- 进程管理
- 系统服务
- ……

常用的系统命令

top：显示执行中的进程

ps：显示瞬间进程的动态

kill：通过PID中止一个进程

pkill：通过进程名中止一个进程

shutdown：关闭计算机

reboot：重启计算机

exit：退出登录

top：显示执行中的进程

作用：top命令用来显示执行中的程序进程，使用权限是所有用户。

格式：top [-] [d delay] [q] [c] [S] [s] [i] [n]

主要参数

-d：指定更新的间隔，以秒计算

-q：没有任何延迟的更新。如果使用者有超级用户，则top命令将会以最高的优先序执行

-c：显示进程完整的路径与名称

-S：累积模式，会将已完成或消失的子行程的CPU时间累积起来

-s：安全模式

-i：不显示任何闲置(Idle)或无用(Zombie)的行程

-n：显示更新的次数，完成后将会退出top

ps：显示瞬间进程的动态

作用：ps显示瞬间进程 (process) 的动态，使用权限是所有使用者

格式：ps [options] [--help]

主要参数：ps的参数非常多, 此出仅列出几个常用的参数

-A：列出所有的进程

-l：显示长列表

-m：显示内存信息

-w：显示加宽可以显示较多的信息

-e：显示所有进程

-a：显示终端上的所有进程,包括其它用户的进程

-au：显示较详细的信息

-aux：显示所有包含其它使用者的进程

kill：通过PID中止一个进程

作用：kill命令用来中止一个进程

格式：kill [- s signal | - p] [- a] pid ... 或者 kill - l [signal]

参数

- 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

pid：要中止进程的ID号

Signal：表示信号

用例：强行中止（经常使用杀掉）一个进程标识号为324的进程

＃kill - 9 324

pkill：通过进程名中止一个进程

作用：功能同killall命令，可以直接使用进程名字而不是进程标识号来中止一个进程
格式：pkill 正在运行的进程名

用例：

```
[root@localhost test]# pgrep -l test
2979 test
[root@localhost test]# pkill test
[root@localhost test]# pkill –u zhangshan
```

shutdown：关闭计算机

shutdown命令的作用是关闭计算机，它的使用权限是超级用户

格式 shutdown [- h][- i][- k][- m][- t]

重要参数

- t：在改变到其它运行级别之前，告诉init程序多久以后关机
- k：并不真正关机，只是送警告信号给每位登录者
- h：关机后关闭电源
- c：cancel current process取消目前正在执行的关机程序。所以这个选项当然没有时间参数，但是可以输入一个用来解释的讯息，而这信息将会送到每位使用者
- -F：在重启计算机时强迫fsck
- time：设定关机前的时间
- m: 将系统改为单用户模式
- i：关机时显示系统信息

reboot：重启计算机

reboot命令的作用是重新启动计算机，它的使用权限是系统管理者

格式 reboot [- n] [- w] [- d] [- f] [- i]

主要参数

- n: 在重开机前不做将记忆体资料写回硬盘的动作
- w: 并不会真的重开机，只是把记录写到/var/log/wtmp文件里
- d: 不把记录写到/var/log/wtmp文件里（ - n这个参数包含了 - d）
- i: 在重开机之前先把所有与网络相关的装置停止

exit：退出登录

作用

exit命令的作用是退出系统，它的使用权限是所有用户

格式

exit

参数

exit命令没有参数，运行后退出系统进入登录界面

top 命令用法及参数

- top是动态监视系统任务的工具，top 输出的结果是连续的

参数:

-b 以批量模式运行，但不能接受命令行输入

-c 显示命令行，而不仅仅是命令名

-d N 显示两次刷新时间的间隔，比如 -d 5，表示两次刷新间隔为5秒

-i 禁止显示空闲进程或僵尸进程

-n NUM 显示更新次数，然后退出。比如 -n 5，表示top更新5次数据就退出

-p PID 仅监视指定进程的ID；PID是一个数值

-q 不经任何延时就刷新

-s 安全模式运行，禁用一些效互指令

-S 累积模式，输出每个进程的总的CPU时间，包括已死的子进程

系统服务

#命令：systemctl [command] [unit]

#command主要包含：

start：启动unit

stop：关闭unit

restart：重启unit

reload：重载unit的配置

enable：设置下次开机时，unit会被启动

disable：设置下次开机时，unit不会被启动

is-active ：目前有没有正在运行中

is-enable ：开机时有没有默认要启用这个unit

show:显示某个 Unit 的所有底层参数

服务命令

systemctl status 服务名
systemctl start 服务名
systemctl stop 服务名
systemctl restart 服务名

、服务状态和依赖

systemctl list-units # 列出正在运行的 Unit
systemctl list-units --all # 列出所有Unit，包括没有找到配置文件的或者启动失败的
systemctl list-units --all --state=inactive # 列出所有没有运行的 Unit
systemctl list-units --failed # 列出所有加载失败的 Unit
systemctl list-units --type=service # 列出所有正在运行的、类型为 service 的 Unit

systemctl list-dependencies [unit] #列出unit依赖
systemctl list-dependencies --all [unit] #展开显示依赖关系
systemctl list-dependencies [unit] [--reverse] #反向查询依赖，unit被谁依赖

dmesg命令

功能：显示系统诊断信息、操作系统版本号、物理内存的大小以及其它信息

```
[root@ccopt ~]# dmesg|more
[    0.000000] microcode: microcode updated early to revision 0x2000064, date = 2019-07-31
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Initializing cgroup subsys cpuacct
[    0.000000] Linux version 3.10.0-1127.8.2.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
) #1 SMP Tue May 12 16:57:42 UTC 2020
[    0.000000] Command line: BOOT_IMAGE=/vmlinuz-3.10.0-1127.8.2.el7.x86_64 root=UUID=32eaf7c5-21a7-4518-a12a-2f3c69d9d065 ro nofb splash=quiet cr
ashkernel=auto spectre_v2=retpoline rhgb quiet LANG=en_US.UTF-8
[    0.000000] e820: BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000008dfff] usable
[    0.000000] BIOS-e820: [mem 0x000000000008e000-0x000000000008ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000090000-0x0000000000009ffff] usable
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x0000000085160fff] usable
[    0.000000] BIOS-e820: [mem 0x0000000085161000-0x0000000085163fff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000085164000-0x000000008df43fff] usable
[    0.000000] BIOS-e820: [mem 0x000000008df44000-0x000000008efe0fff] reserved
[    0.000000] BIOS-e820: [mem 0x000000008efe1000-0x000000008efe1fff] usable
[    0.000000] BIOS-e820: [mem 0x000000008efe2000-0x0000000090227fff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000090228000-0x0000000099a27fff] usable
[    0.000000] BIOS-e820: [mem 0x0000000099a28000-0x000000009a727fff] reserved
[    0.000000] BIOS-e820: [mem 0x000000009a728000-0x000000009cb27fff] ACPI NVS
[    0.000000] BIOS-e820: [mem 0x000000009cb28000-0x000000009cd27fff] ACPI data
[    0.000000] BIOS-e820: [mem 0x000000009cd28000-0x000000009cd29fff] usable
[    0.000000] BIOS-e820: [mem 0x000000009cd2a000-0x000000009cdaafff] reserved
[    0.000000] BIOS-e820: [mem 0x000000009cdab000-0x00000000af7fffff] usable
[    0.000000] BIOS-e820: [mem 0x00000000af800000-0x00000000cfffffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fe000000-0x00000000fe00ffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000100000000-0x000000103fffefff] usable
[    0.000000] BIOS-e820: [mem 0x000000103ffff000-0x000000103fffffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] e820: update [mem 0x984a7018-0x984d7e57] usable ==> usable
[    0.000000] e820: update [mem 0x98476018-0x984a6e57] usable ==> usable
[    0.000000] e820: update [mem 0x98445018-0x98475e57] usable ==> usable
[    0.000000] e820: update [mem 0x98414018-0x98444e57] usable ==> usable
[    0.000000] e820: update [mem 0x9840b018-0x98413057] usable ==> usable
[    0.000000] e820: update [mem 0x98400018-0x9840a857] usable ==> usable
[    0.000000] e820: update [mem 0x84f23018-0x8510de57] usable ==> usable
```

帮助命令

help：提供内部命令的帮助

```
[root@localhost ~]# ls --help | more
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuSUX nor --sort.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not hide entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               print the author of each file
  -b, --escape               print octal escapes for nongraphic characters
      --block-size=SIZE      use SIZE-byte blocks
```

man：提供所有命令的详细帮助；是 UNIX 系统手册的电子版本。根据习惯，UNIX 系统手册通常分为不同的部分（或小节，即 section），每个小节阐述不同的系统内容

info：提供所有命令的详细帮助；Linux 中的大多数软件开发工具都是来自自由软件基金会的 GNU 项目，这些工具软件件的在线文档都以 info 文件的形式存在

高能物理计算集群介绍

- 什么是集群?

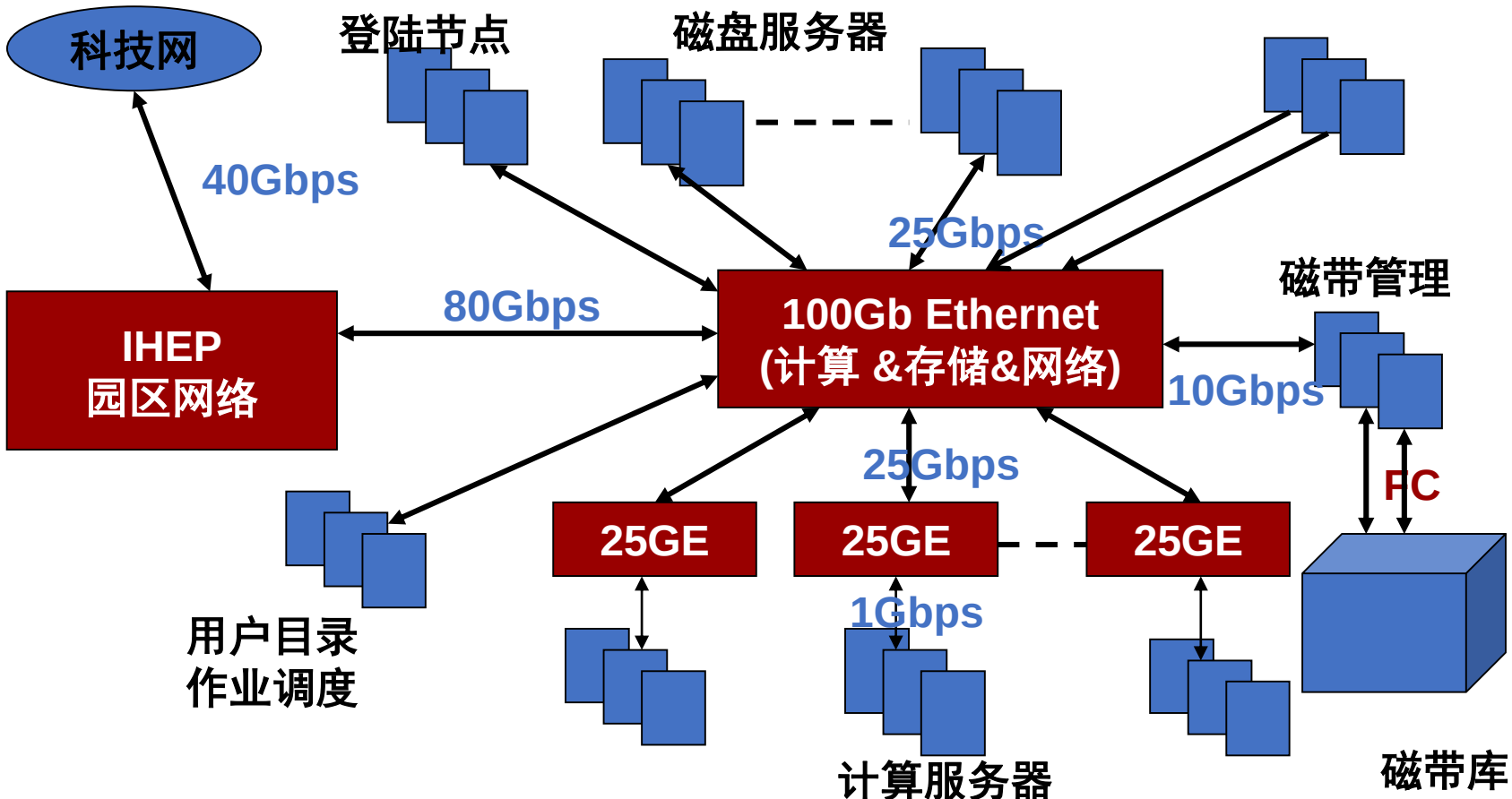
计算集群是将上千台计算机，利用高速通信网络组成的一个较大的计算机服务系统，形成一个大的超级计算机。这个超级计算机由用户管理、计算调度、数据存储、软件运行管理等多个子系统组成。

- 计算中心集群的具体运行情况

共有1500~台服务器，5.5万~CPU 核，200~块 GPU 卡。磁盘存储70PB，磁带存储39PB。计算核心网络100Gb/s,出口总带宽是40Gb/s

Architecture

- 什么是集群?
- 本地集群提供的服务主要包含三个部分，用户管理、计算服务、存储服务



本地集群使用

登录到登录结点

在登录结点上编辑文件，调试程序，并将程序保存在自己的文件目录中

提交作业到指定队列

作业被调度到计算结点上执行，执行完毕后结果将被拷贝回用户指定的输出目录

用户在登录结点上检查作业执行的结果

本地集群作业

- 针对不同的实验计算需求，计算平台提供两类作业的运行方式
- HTCondor集群支持高通量计算（High Throughput Computing），采用HTCondor作为负载管理系统，绝大多数作业为单核串行作业
- Slurm集群支持高性能作业（High Performance Computing），采用Slurm作为负载管理系统，绝大多数作业为多核并行作业。Hadoop集群是一个针对高能物理的特性、基于Hadoop的新型计算平台，目前支持LHAASO实验计算。本节将介绍各集群的作业提交、查看等基本操作

文件存储服务

- AFS：可通过广域网和局域网访问的分布式文件系统。 安全性高、速读慢、小文件
- NFS：网络文件系统，主要作为用户HOME目录，用于保存软件、脚本和一些实验数据。NFS适合于小文件可扩展性差、访问速度慢、维护复杂，将在以后被逐步淘汰
- LUSTER：计算中心最主要的磁盘文件系统，面向对象的并行文件系统，管理数十PB存储空间，聚合带宽高达每秒几十甚至上百GB，在全球的超算超过70%使用。由于元数据服务器的限制，在管理海量小文件时不具有优势，不支持纠删码、副本等功能，因此主要用于海量的大文件实验数据存储

文件存储服务

- EOS存储：EOS文件系统是一套基于xrootd框架的面向EB级数据存储的分布式文件系统。目前提供LHAASO、HXMT等多个实验的海量数据存储服务。
- 磁带库存储：通过部署分级存储，将磁带虚拟成可随机访问的目录，通过命令或API实现数据的导入（上传）或者拷出（下载）。优点是容量大、价格便宜、保存时间长、绿色节能等优势。但是磁带库是机械设备，访问速度慢，故障率高。因此目前主要用于原始数据长期保存、 在线存储，备份

集群用户手册

- 详情参考· GitBookhttp://afsapply.ihep.ac.cn/cchelp/zh/

谢谢