

1. Linux基础指令

- 初始Linux操作系统
- 初识shell命令,了解若干背景知识。
- 初步使用40+个常用Linux命令
- 掌握Linux权限概念与思想,能深度理解“权限”
- 初步了解Linux 目录结构含义
- 了解Linux相关知识: 目录结构/路径/重定向/管道操作等

Linux背景

1. 发展史

本门课程学习Linux系统编程,你可能要问Linux从哪里来?它是怎么发展的?在这里简要介绍Linux的发展史。要说Linux,还得从UNIX说起。

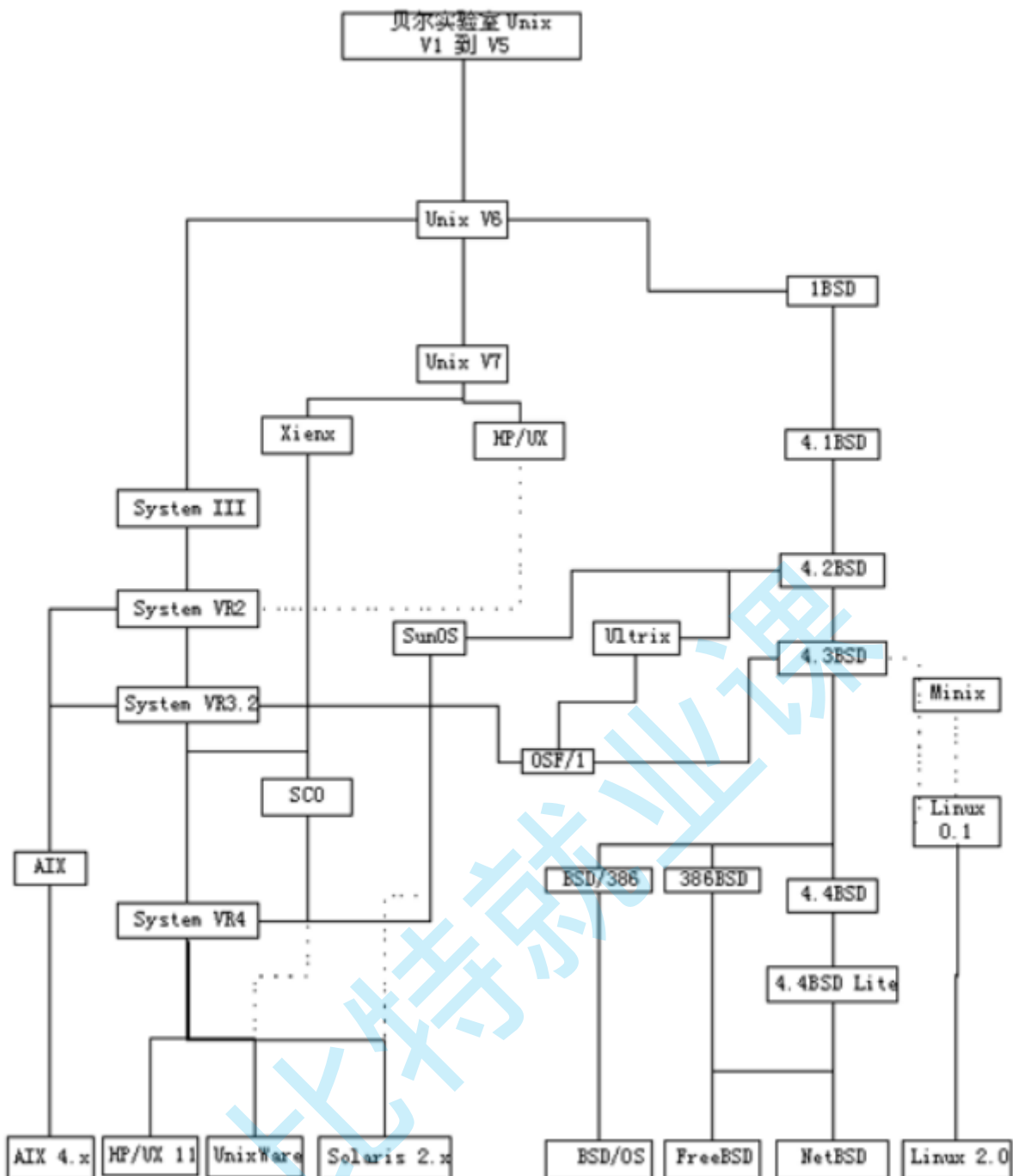
UNIX发展的历史

- 1968年,一些来自通用电器公司、贝尔实验室和麻省理工学院的研究人员开发了一个名叫Multics的特殊操作系统。Multics在多任务文件管理和用户连接中综合了许多新概念。
- 1969—1970年,AT&T的贝尔实验室研究人员Ken Thompson和Dennis Ritchie,在采用很多Multics特点的基础上开发了UNIX系统。它运行在小型机上,满足了系统对科研环境的要求。从产生开始,UNIX就是一个有价值的、高效的、多用户和多任务的操作系统。
- UNIX从满足个人的设计需求开始,逐步成长为由许多不同开发商所支持的标准软件产品。
- 第一个UNIX版本是免费给许多知名大学的计算机系使用的。
- 1972年,贝尔实验室开始发放商业版本,并且给不同的用户授权使用这个系统,使用者之一是加州大学伯克莱分校的计算机系。伯克莱给系统增加了许多新的特点,后来成为了标准。
- 1975年伯克莱由下属部门BSD发行了自己的UNIX版本。UNIX的BSD版本成为AT&T贝尔实验室版本的主要竞争者,而其它的独立开发出的UNIX版本也开始萌生。
- 1980年微软公司开发了叫做Xenix的UNIX PC版本。AT&T发行了第一个商业版本。名叫System III,后来被成为对商用软件产品良好支持的System V所替代。
- 同时UNIX的BSD版本不断发展,在70年代末期,BSD UNIX成为了国防部的高科技研究机构科研项目的基础。其结果,伯克莱发行了一个叫做BSD Release 4.2的有效版本。
- 它包括了高级的文件管理器和基于TCP/IP网络协议的网络特点。现在TCP/IP被Internet所使用。BSD Release 4.2被许多厂商所采用,例如SUN Microsystem。

- UNIX不同版本的出现导致了UNIX标准的需要，软件开发商不知道他们的程序运行在哪些版本上比较合适。
- 到80年代中期，两个竞争的标准出现了，一个是基于AT&T的UNIX版本，另一个是BSD版本。在今天的书店里你能发现分别适用于这两个版本的不同的UNIX的书，一些是System V，另一些集中在BSD UNIX。
- AT&T建立了一个叫UNIX系统实验室的新组织，它的作用就是综合UNIX的不同版本，集中开发一个标准系统。
- 1991年，UNIX综合实验室综合了System V Release3，SUN OS和Xenix的所有特点，发行了System V Release 4。为了与System V Release 4 竞争，一些其它公司，如IBM和惠普Open Software Foundation（OSF）去产生自己的UNIX标准版本，继而出现了两个标准商业版本OSF版本和System Release 4。
- 1993年，AT&T把它的UNIX转卖给Novell公司。UNIX系统实验室成为了Novell的UNIX系统小组的一部分。Novell发行了基于System V Release 4的自己的UNIX版本UNIXWare，它可以和Novell公司的Netware系统相联。SUN公司已经把System V Release 4 融进了它的SUN OS，发行了Solaris。两个相互竞争的UNIX使用的图形用户界面（一个叫Motif，另一个叫Openlook），已经合并为一个新的工作平台标准，叫做通用平台环境（CDE）。

Linux发展历史

- 1991年10月5日，赫尔辛基大学的一名研究生Linus Benedict Torvalds在一个Usenet新闻组（comp.os.minix）中宣布他编制出了一种类似UNIX的小操作系统，叫Linux。新的操作系统是受到另一个UNIX的小操作系统——Minix的启发，该系统是由一名叫Andrew S Tanenbaum的教师开发的。读者也许猜想所发布的这个系统应该是Linux的0.01版本，实际上不是这样。真正的Linux 0.01版本并没有被发布，原因是0.01版本不实用。Linus仅仅在第一个Linux的FTP站点（<ftp://nic.funet.fi>）上提供过这个版本的源代码。
- Torvalds于10月5日发布的这个Linux版本被称为0.02版，它能够运行GNU Bourne Again Shell(bash)和GNU的C编译程序（gcc）以及为数不多的其它语言。Torvalds绝对没有想到他设想的一种能够针对高级业余爱好者和黑客们的操作系统已经产生，这就是人们所称的Linux。；
- Linux发布时的版本是0.02，后来又有0.03版，然后又跳到0.10版。因为世界各地越来越多的程序员都开始开发Linux，它已经达到0.95版。这就意味着正式公布1.0版本的时间已经为期不远了。正式的1.0版本是在1994年公布的



- 对比一下 Windows 发展史



2. 开源

- Linux是一种自由和开放源代码的类UNIX操作系统，该操作系统的内核由林纳斯托瓦兹在1991年首次发布，之后，在加上用户空间的应用程序之后，就成为了Linux操作系统。严格来讲，Linux只是操作系统内核本身，但通常采用“Linux内核”来表达该意思。而Linux则常用来指基于Linux内核的完整操作系统，它包括GUI组件和许多其他实用工具。
- GNU通用公共许可协议（GNU General Public License，简称GNU GPL或GPL），是一个广泛被使用的自由软件许可协议条款，最初由理查德斯托曼为GNU计划而撰写，GPL给予了计算机程序自由软件的定义，任何基于GPL软件开发衍生的产品在发布时必须采用GPL许可证方式，且必须公开源代码
- Linux是自由软件和开放源代码软件发展中最著名的例子。只要遵循GNU通用公共许可证，任何个人和机构都可以自由地使用Linux的所有底层源代码，也可以自由地修改和再发布。随着Linux操作系统飞速发展，各种集成在Linux上的开源软件和实用工具也得到了应用和普及，因此，Linux也成为了开源软件的代名词。

3. 官网

- [kernel官网](#)

4. 企业应用现状

• Linux在服务器领域的发展

随着开源软件在世界范围内影响力日益增强，Linux服务器操作系统在整个服务器操作系统市场格局中占据了越来越多的市场份额，已经形成了大规模市场应用的局面。并且保持着快速的增长率。尤其在政府、金融、农业、交通、电信等国家关键领域。此外，考虑到Linux的快速成长性以及国家相关政策的扶持力度，Linux服务器产品一定能够冲击更大的服务器市场。

据权威部门统计，目前Linux在服务器领域已经占据75%的市场份额，同时，Linux在服务器市场的迅速崛起，已经引起全球IT产业的高度关注，并以强劲的势头成为服务器操作系统领域中的中坚力量。

• Linux在桌面领域的发展

近年来，特别在国内市场，Linux桌面操作系统的发展趋势非常迅猛。国内如中标麒麟Linux、红旗Linux、深度Linux等系统软件厂商都推出的Linux桌面操作系统，目前已经在政府、企业、OEM等领域得到了广泛应用。另外SUSE、Ubuntu也相继推出了基于Linux的桌面系统，特别是Ubuntu Linux，已经积累了大量社区用户。但是，从系统的整体功能、性能来看，Linux桌面系统与Windows系列相比还有一定的差距，主要表现在系统易用性、系统管理、软硬件兼容性、软件的丰富程度等方面。

- **Linux在移动嵌入式领域的发展**

Linux的低成本、强大的定制功能以及良好的移植性能，使得Linux在嵌入式系统方面也得到广泛应用，目前Linux以广泛应用于手机、平板电脑、路由器、电视和电子游戏机等领域。在移动设备上广泛使用的Android操作系统就是创建在Linux内核之上的。目前，Android已经成为全球最流行的智能手机操作系统，据2015年权威部门最新统计，Android操作系统的全球市场份额已达84.6%。

此外，思科在网络防火墙和路由器也使用了定制的Linux，阿里云也开发了一套基于Linux的操作系统“YunOS”，可用于智能手机、平板电脑和网络电视；常见的数字视频录像机、舞台灯光控制系统等都在逐渐采用定制版本的Linux来实现，而这一切均归功于Linux与开源的力量。

- **Linux在云计算/大数据领域的发展**

互联网产业的迅猛发展，促使云计算、大数据产业的形成并快速发展，云计算、大数据作为一个基于开源软件的平台，Linux占据了核心优势；据Linux基金会的研究，86%的企业已经使用Linux操作系统进行云计算、大数据平台的构建，目前，Linux已开始取代Unix成为最受青睐的云计算、大数据平台操作系统。

5. 发行版本

- **Debian**

Debian运行起来极其稳定，这使得它非常适合用于服务器。debian这款操作系统无疑并不适合新手用户，而是适合系统管理员和高级用户。

- **Ubuntu**

Ubuntu是Debian的一款衍生版，也是当今最受欢迎的免费操作系统。Ubuntu侧重于它在这个市场的应用，在服务器、云计算、甚至一些运行Ubuntu Linux的移动设备上很常见。Ubuntu是新手用户肯定爱不释手的一款操作系统。

- **红帽企业级Linux**

这是第一款面向商业市场的Linux发行版。它有服务器版本，支持众多处理器架构，包括x86和x86_64。红帽公司通过课程红帽认证系统管理员/红帽认证工程师（RHCSA/RHCE），对系统管理员进行培训和认证。

- **CentOS**

CentOS是一款企业级Linux发行版，它使用红帽企业级Linux中的免费源代码重新构建而成。这款重构版完全去掉了注册商标以及Binary程序包方面一个非常细微的变化。有些人不想支付一大笔钱，又能领略红帽企业级Linux；对他们来说，CentOS值得一试。此外，CentOS的外观和行为似乎与母发行版红帽企业级Linux如出一辙。CentOS使用YUM来管理软件包。

- **Fedora**

小巧的Fedora适合那些人：想尝试最先进的技术，等不及程序的稳定版出来。其实，Fedora就是红帽公司的一个测试平台；产品在成为企业级发行版之前，在该平台上进行开发和测试。Fedora是一款非常好的发行版，有庞大的用户论坛，软件库中还有为数不少的软件包。

- Kali Linux

Kali Linux是Debian的一款衍生版。Kali旨在用于渗透测试。Kali的前身是Backtrack。用于Debian的所有Binary软件包都可以安装到Kali Linux上，而Kali的魅力或威力就来自于此。此外，支持Debian的用户论坛为Kali加分不少。Kali随带许多的渗透测试工具，无论是Wifi、数据库还是其他任何工具，都设计成立马可以使用。Kali使用APT来管理软件包。

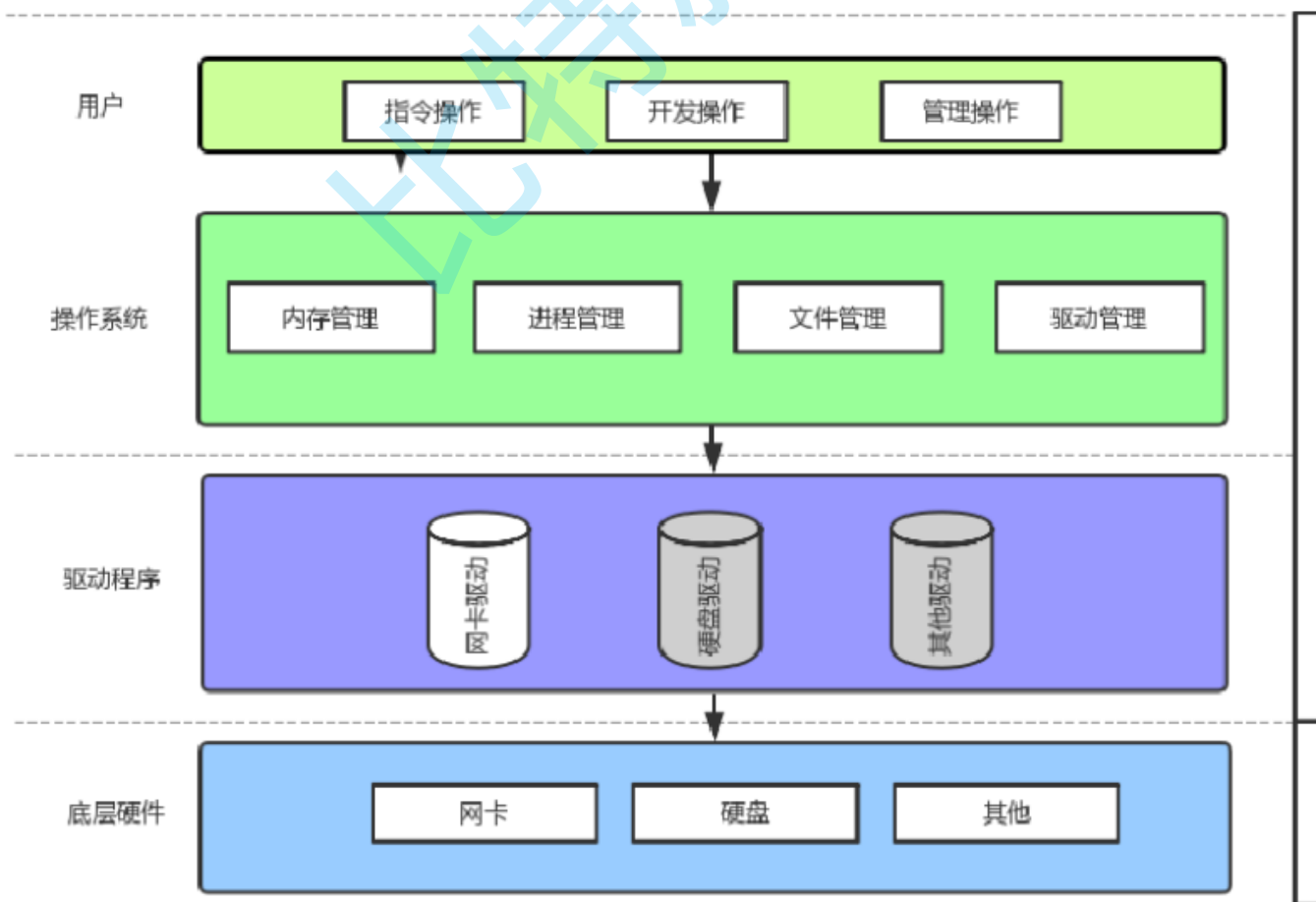
毫无疑问，Kali Linux是一款渗透测试工具，或者是文明黑客（我不想谈论恶意黑客）青睐的操作系统。

-



6. os概念，定位

使计算机更好用! 这是操作系统的根本要义!!



使用 XShell 远程登录 Linux

关于 Linux 桌面

很多同学的 Linux 启动进入图形化的桌面. 这个东西大家以后就可以忘记了. 以后的工作中 **没有机会** 使用图形界面.

思考: 为什么不使用图形界面?

下载安装 XShell

XShell 是一个远程终端软件. 下载官网: https://www.netsarang.com/products/xsh_overview.html

下载安装的时候选择 "home/school" 则为免费版本.

查看 Linux 主机 ip

在终端下敲 `ifconfig` 指令, 查看到 ip 地址.

使用 XShell 登陆主机

在 XShell 终端下敲 `ssh [ip]`, ip 为刚才看到的 ifconfig 结果.

如果网络畅通, 将会提示输入用户名密码. 输入即可正确登陆.

XShell 下的复制粘贴

复制: `ctrl + insert` (有些同学的 insert 需要配合 fn 来按)

粘贴: `shift + insert`

`ctrl + c` / `ctrl + v` 是不行的.

Linux下基本指令

ls 指令

语法: `ls [选项] [目录或文件]`

功能: 对于目录, 该命令列出该目录下的所有子目录与文件。对于文件, 将列出文件名以及其他信息。

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。
- `-d` 将目录像文件一样显示, 而不是显示其下的文件。如: `ls -d` 指定目录
- `-i` 输出文件的 i 节点的索引信息。如 `ls -ai` 指定文件
- `-k` 以 k 字节的形式表示文件的大小。 `ls -alk` 指定文件
- `-l` 列出文件的详细信息

- -n 用数字的 UID,GID 代替名称。（介绍 UID， GID）
- -F 在每个文件名后附上一个字符以说明该文件的类型，“*” 表示可执行的普通文件；“/” 表示目录；“@” 表示符号链接；“|” 表示FIFOs；“=” 表示套接字(sockets)。（目录类型识别）
- -r 对目录反向排序
- -t 以时间排序
- -s 在l文件名后输出该文件的大小。（大小排序，如何找到目录下最大的文件）
- -R 列出所有子目录下的文件。（递归）
- -l 一行只输出一个文件。

举例：

```

1 [whb@bite-alicloud test]$ ls
2 dir file.txt
3 [whb@bite-alicloud test]$ ls -a
4 . .. dir file.txt
5 [whb@bite-alicloud test]$ ls -a -l
6 total 12
7 drwxrwxr-x  3 whb whb 4096 Jan 11 15:01 .
8 drwx----- 22 whb whb 4096 Jan 11 14:21 ..
9 drwxrwxr-x  2 whb whb 4096 Jan 11 14:22 dir
10 -rw-rw-r--  1 whb whb   0 Jan 11 14:22 file.txt
11 [whb@bite-alicloud test]$ ls -a -l -F
12 total 12
13 drwxrwxr-x  3 whb whb 4096 Jan 11 15:01 ./
14 drwx----- 22 whb whb 4096 Jan 11 14:21 ../
15 drwxrwxr-x  2 whb whb 4096 Jan 11 14:22 dir/
16 -rw-rw-r--  1 whb whb   0 Jan 11 14:22 file.txt
17 [whb@bite-alicloud test]$ ls -alFn
18 total 12
19 drwxrwxr-x  3 1003 1003 4096 Jan 11 15:01 ./
20 drwx----- 22 1003 1003 4096 Jan 11 14:21 ../
21 drwxrwxr-x  2 1003 1003 4096 Jan 11 14:22 dir/
22 -rw-rw-r--  1 1003 1003   0 Jan 11 14:22 file.txt

```

pwd 命令

语法: `pwd`

功能：显示用户当前所在的目录

常用选项：

无

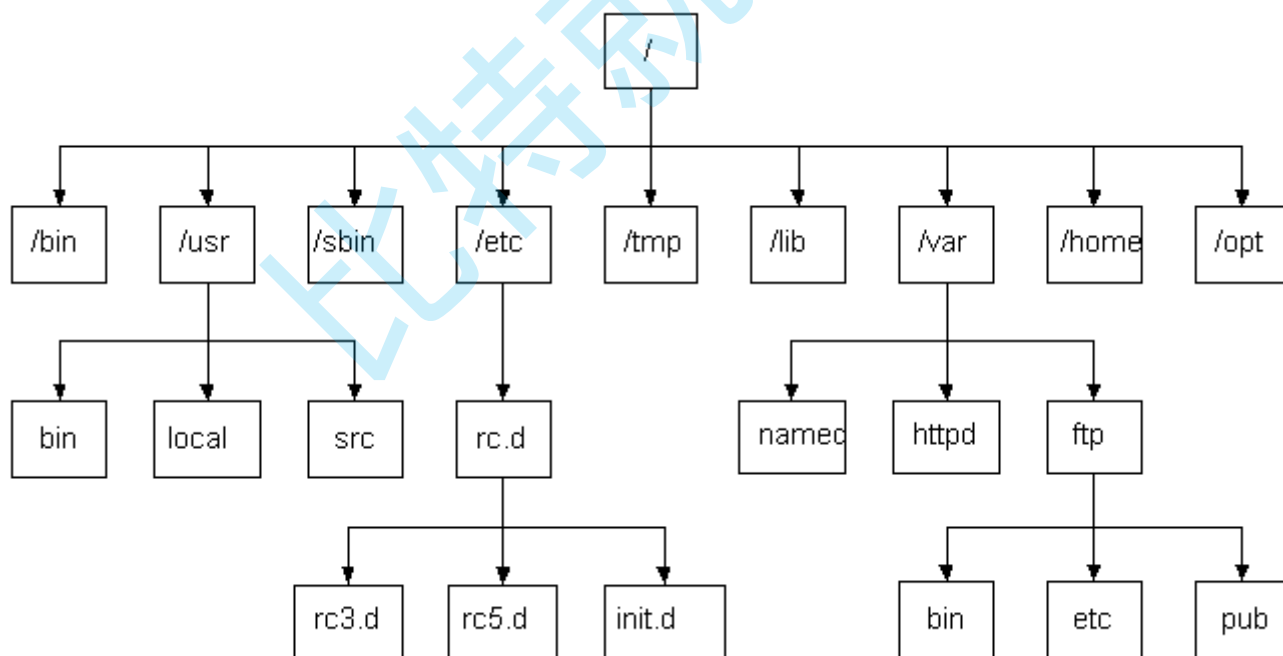
举例：


```
1 [whb@bite-alicloud test]$ pwd
2 /home/whb/test
```

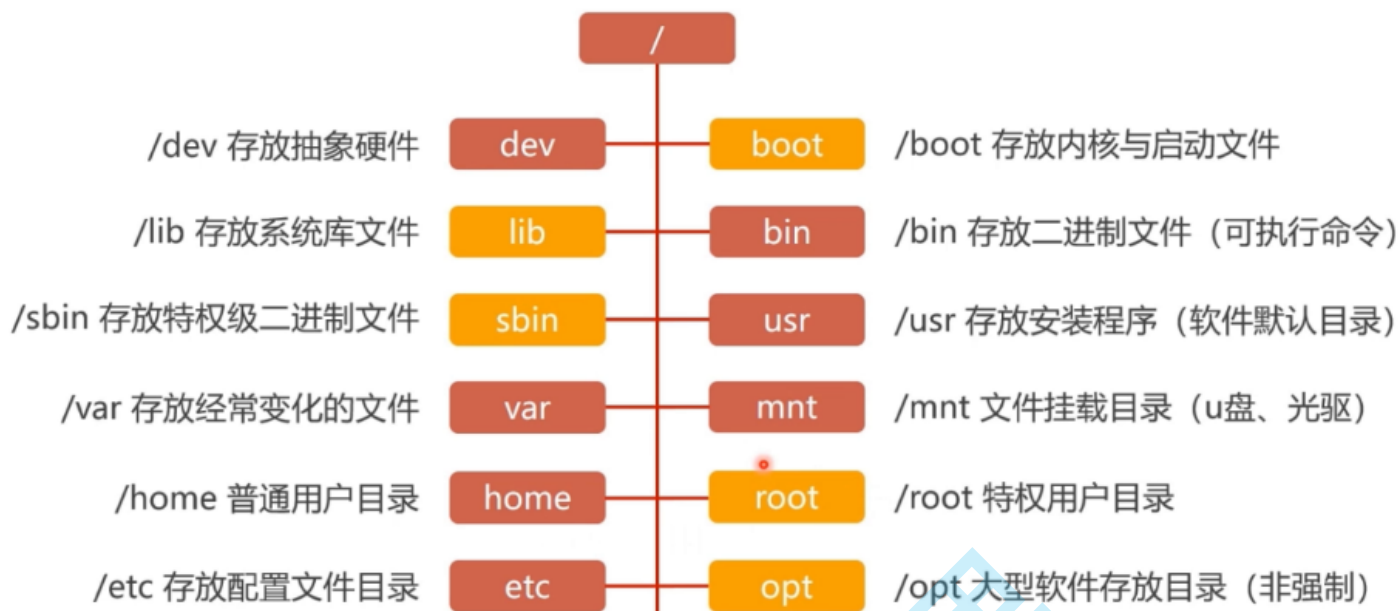
cd 指令

Linux理论知识：路径的认识

- Linux系统中，磁盘上的文件和目录被组成一棵目录树，每个节点都是目录或文件
- 其中普通文件一定是目录树的叶子节点
- 目录可能是叶子(空目录), 也可能是路上节点
- **理解路径存在的意义:** 树状组织方式，都是为了保证快速定位查找到指定的文件，而定位文件就需要具有唯一性的方案来进行定位文件。其中任何一个节点，都只有一个父节点，所以，从根目录开始，定位指定文件，路径具有唯一性
- 绝对路径：一般从/开始，不依赖其他目录的定位文件的方式
- 相对路径：相对于当前用户所处目录，定位文件的路径方式
- 绝对路径一般不会随着用户的路径变化而丧失唯一性，一般在特定服务的配置文件中经常被使用
- 相对路径因为它的便捷性，一般在命令行中使用较多



Linux目录结构



语法: `cd 目录名`

功能: 改变工作目录。将当前工作目录改变到指定的目录下

举例:

```
1 # 返回上级目录
2 [whb@bite-alicloud ~]$ pwd
3 /home/whb
4 [whb@bite-alicloud ~]$ cd ..
5 [whb@bite-alicloud home]$ pwd
6 /home
7
8 # 绝对路径定位目录或者文件
9 [whb@bite-alicloud home]$ pwd
10 /home
11 [whb@bite-alicloud home]$ cd /home/whb/test
12 [whb@bite-alicloud test]$ pwd
13 /home/whb/test
14
15 # 相对路径定位目录或者文件
16 [whb@bite-alicloud test]$ pwd
17 /home/whb/test
18 [whb@bite-alicloud test]$ cd ../temp/
19 [whb@bite-alicloud temp]$ pwd
20 /home/whb/temp
21
22 # 每一次新建一个用户都会会在/home目录下为新用户创建新的工作目录, 目录以新用户名称命名
23 # root用户在安装操作系统的时候, 就已经内置了工作目录: /root
24 [whb@bite-alicloud temp]$ ls /home
```

```
25 bss gaobo luojie whb wudu
26
27 # 快速进入自己的家目录
28 [whb@bite-alicloud test]$ whoami
29 whb
30 [whb@bite-alicloud test]$ pwd
31 /home/whb/test
32 [whb@bite-alicloud test]$ cd ~
33 [whb@bite-alicloud ~]$ pwd
34 /home/whb
35
36 # 返回到最近访问目录
37 [whb@bite-alicloud ~]$ pwd # 当前在/home/whb目录下
38 /home/whb
39 [whb@bite-alicloud ~]$ cd / # 进入/目录
40 [whb@bite-alicloud /]$ pwd
41 /
42 [whb@bite-alicloud /]$ cd - # 直接可以回退到最近一次所处的目录:/home/whb
43 /home/whb
44 [whb@bite-alicloud ~]$ pwd
45 /home/whb
46 [whb@bite-alicloud ~]$ cd - # 直接可以回退到最近一次所处的目录:/
47 /
48 [whb@bite-alicloud /]$ pwd
49 /
50 [whb@bite-alicloud /]$ cd - # 同上
51 /home/whb
52 [whb@bite-alicloud ~]$ pwd
53 /home/whb
```

touch 指令

Linux理论知识：文件类型的认识

语法: touch [选项]... 文件...

功能: touch命令参数可更改文档或目录的日期时间，包括存取时间和更改时间，或者新建一个不存在的文件。

常用选项:

- -a : change only the access time
- -c : change only the modification time

举例:

```
1 # 创建普通文件
2 [whb@bite-alicloud test]$ ls
```

```

3 dir file.txt
4 [whb@bite-alicloud test]$ touch newFile.txt
5 [whb@bite-alicloud test]$ ls
6 dir file.txt newFile.txt
7
8 # 查看文件时间
9 [whb@bite-alicloud test]$ stat newFile.txt
10 File: 'newFile.txt'
11 Size: 0                      Blocks: 0                      IO Block: 4096    regular empty
   file
12 Device: fd01h/64769d        Inode: 1844719          Links: 1
13 Access: (0664/-rw-rw-r--)  Uid: ( 1003/          whb)   Gid: ( 1003/          whb)
14 Access: 2023-01-11 15:06:45.832787572 +0800
15 Modify: 2023-01-11 15:06:45.832787572 +0800
16 Change: 2023-01-11 15:06:45.832787572 +0800
17 Birth: -
18
19 # 修改文件access时间
20 [whb@bite-alicloud test]$ touch -a newFile.txt
21 [whb@bite-alicloud test]$ stat newFile.txt
22 File: 'newFile.txt'
23 Size: 0                      Blocks: 0                      IO Block: 4096    regular empty
   file
24 Device: fd01h/64769d        Inode: 1844719          Links: 1
25 Access: (0664/-rw-rw-r--)  Uid: ( 1003/          whb)   Gid: ( 1003/          whb)
26 Access: 2023-01-11 15:07:00.062846804 +0800
27 Modify: 2023-01-11 15:06:45.832787572 +0800
28 Change: 2023-01-11 15:07:00.062846804 +0800 # 该时间为何会变化?
29 Birth: -
30
31 # 修改文件Modify时间
32 [whb@bite-alicloud test]$ touch -m newFile.txt
33 [whb@bite-alicloud test]$ stat newFile.txt
34 File: 'newFile.txt'
35 Size: 0                      Blocks: 0                      IO Block: 4096    regular empty
   file
36 Device: fd01h/64769d        Inode: 1844719          Links: 1
37 Access: (0664/-rw-rw-r--)  Uid: ( 1003/          whb)   Gid: ( 1003/          whb)
38 Access: 2023-01-11 15:07:00.062846804 +0800
39 Modify: 2023-01-11 15:09:19.428426903 +0800
40 Change: 2023-01-11 15:09:19.428426903 +0800
41 Birth: -
42
43 # 更多讲解，会在基础IO部分统一在说

```

mkdir 指令

语法: `mkdir [选项] dirname...`

功能: 在当前目录下创建一个名为“dirname”的目录

常用选项:

- `-p/--parents`: 可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录

举例:

```
1 # 创建普通空目录
2 [whb@bite-alicloud test]$ ls -l
3 total 4
4 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
5 -rw-rw-r-- 1 whb whb 0 Jan 11 14:22 file.txt
6 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
7 [whb@bite-alicloud test]$ mkdir mydir
8 [whb@bite-alicloud test]$ ls -l
9 total 8
10 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
11 -rw-rw-r-- 1 whb whb 0 Jan 11 14:22 file.txt
12 drwxrwxr-x 2 whb whb 4096 Jan 11 15:15 mydir
13 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
14
15 # 递归建立多个目录, 创建指定路径
16 [whb@bite-alicloud test]$ mkdir -p path1/path2/path3/path4
17 [whb@bite-alicloud test]$ ls -l
18 total 12
19 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
20 -rw-rw-r-- 1 whb whb 0 Jan 11 14:22 file.txt
21 drwxrwxr-x 2 whb whb 4096 Jan 11 15:15 mydir
22 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
23 drwxrwxr-x 3 whb whb 4096 Jan 11 15:16 path1
24 [whb@bite-alicloud test]$ tree path1
25 path1
26 └─ path2
27     └─ path3
28         └─ path4
29
30 # tree命令是一个递归以树状结构显示指定目录的命令, 如果没有安装
31 # 如果当前用户是root: yum install -y tree (因为现在开始用的都是root, 所以推荐这个用法)
32 # 如果当前用户是普通用户: sudo yum install -y tree (后面讲讲完vim, 大家才能将自己添加到系统信任列表中, 暂时普通用户无法使用sudo)
```

rmdir 指令 && rm 指令

rmdir是一个与mkdir相对应的命令, mkdir是建立目录, 而rmdir是删除命令

语法: `rmdir [-p] [dirName]`

适用对象: 具有当前目录操作权限的所有使用者

功能: 删除空目录

常用选项:

- `-p` 当子目录被删除后如果父目录也变成空目录的话,就连带父目录一起删除。

举例:

```
1 # 删除空目录
2 [whb@bite-alicloud test]$ ll
3 total 12
4 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
5 -rw-rw-r-- 1 whb whb 0 Jan 11 14:22 file.txt
6 drwxrwxr-x 2 whb whb 4096 Jan 11 15:15 mydir
7 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
8 drwxrwxr-x 3 whb whb 4096 Jan 11 15:16 path1
9 [whb@bite-alicloud test]$ tree mydir/
10 mydir/
11
12 0 directories, 0 files
13 [whb@bite-alicloud test]$ rmdir mydir
14 [whb@bite-alicloud test]$ ll
15 total 8
16 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
17 -rw-rw-r-- 1 whb whb 0 Jan 11 14:22 file.txt
18 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
19 drwxrwxr-x 3 whb whb 4096 Jan 11 15:16 path1
20
21 # 删除路径
22 [whb@bite-alicloud test]$ tree path1
23 path1
24 └── path2
25     └── path3
26         └── path4
27
28 3 directories, 0 files
29 [whb@bite-alicloud test]$ rmdir -p path1/path2/path3/path4
30 [whb@bite-alicloud test]$ ll
31 total 4
32 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
33 -rw-rw-r-- 1 whb whb 0 Jan 11 14:22 file.txt
34 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
35
36 # 指定路径中有不为空的路径, 便无法删除
37 [whb@bite-alicloud test]$ rmdir -p path1/path2/path3
```



```
38 rmdir: failed to remove 'path1/path2/path3': Directory not empty
39
40 [whb@bite-alicloud test]$ touch path1/myfile.txt
41 [whb@bite-alicloud test]$ touch path1/path2/myfile.txt
42 [whb@bite-alicloud test]$ tree path1/
43 path1/
44 |— myfile.txt
45 |— path2
46 |   |— myfile.txt
47 |   |— path3
48 |       |— path4
49
50 3 directories, 2 files
51 [whb@bite-alicloud test]$ rmdir -p path1/path2/path3/path4
52 rmdir: failed to remove directory 'path1/path2': Directory not empty
```

rm 命令可以同时删除文件或目录

语法: `rm [-f-i-r-v] [dirName/dir]`

适用对象: 所有使用者

功能: 删除文件或目录

常用选项:

- -f 即使文件属性为只读(即写保护), 亦直接删除
- -i 删除前逐一询问确认
- -r 删除目录及其下所有文件

举例:

```
1 # 删除普通文件
2 [whb@bite-alicloud test]$ ll
3 total 8
4 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
5 -rw-rw-r-- 1 whb whb 0 Jan 11 14:22 file.txt
6 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
7 drwxrwxr-x 3 whb whb 4096 Jan 11 15:26 path1
8 [whb@bite-alicloud test]$ rm file.txt
9 [whb@bite-alicloud test]$ ll
10 total 8
11 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
12 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
13 drwxrwxr-x 3 whb whb 4096 Jan 11 15:26 path1
14
15 # 删除目录文件
16 [whb@bite-alicloud test]$ ll
```

```
17 total 8
18 drwxrwxr-x 2 whb whb 4096 Jan 11 14:22 dir
19 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
20 drwxrwxr-x 3 whb whb 4096 Jan 11 15:26 path1
21 [whb@bite-alicloud test]$ rm dir
22 rm: cannot remove 'dir': Is a directory
23 [whb@bite-alicloud test]$ rm -r dir
24 [whb@bite-alicloud test]$ ll
25 total 4
26 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
27 drwxrwxr-x 3 whb whb 4096 Jan 11 15:26 path1
28
29 # 删除普通文件前询问
30 [whb@bite-alicloud test]$ ll
31 total 4
32 -rw-rw-r-- 1 whb whb 0 Jan 11 15:09 newFile.txt
33 drwxrwxr-x 3 whb whb 4096 Jan 11 15:26 path1
34 [whb@bite-alicloud test]$ rm -i newFile.txt
35 rm: remove regular empty file 'newFile.txt'? y
36 [whb@bite-alicloud test]$ ll
37 total 4
38 drwxrwxr-x 3 whb whb 4096 Jan 11 15:26 path1
39 [whb@bite-alicloud test]$ ll
40 total 4
41 drwxrwxr-x 3 whb whb 4096 Jan 11 15:26 path1
42
43 # 删除目录前询问
44 [whb@bite-alicloud test]$ mkdir -p d/d/d/d
45 [whb@bite-alicloud test]$ tree d
46 d
47 └─ d
48     └─ d
49         └─ d
50
51 3 directories, 0 files
52 [whb@bite-alicloud test]$ rm -ri d
53 rm: descend into directory 'd'? y
54 rm: descend into directory 'd/d'? y
55 rm: descend into directory 'd/d/d'? y
56 rm: remove directory 'd/d/d/d'? y
57 rm: remove directory 'd/d/d'? y
58 rm: remove directory 'd/d'? y
59 rm: remove directory 'd'? y
60 [whb@bite-alicloud test]$ ll
61 total 0
62
63 # 递归强制删除非空目录
```

```
64 [whb@bite-alicloud test]$ tree path1/
65 path1/
66 |— myfile.txt
67 |— path2
68 |   |— myfile.txt
69
70 1 directory, 2 files
71 [whb@bite-alicloud test]$ rm -f path1
72 rm: cannot remove 'path1': Is a directory
73 [whb@bite-alicloud test]$ rm -rf path1
74 [whb@bite-alicloud test]$ ll
75 total 0
```

man 指令

Linux的命令有很多参数，我们不可能全记住，可以通过查看联机手册获取帮助

语法: `man [选项] 命令`

常用选项

- -k 根据关键字搜索联机帮助
- num 只在第num章节查找
- -a 将所有章节的都显示出来，比如 `man printf` 它缺省从第一章开始搜索，知道就停止，用a选项，当按下q退出，他会继续往后面搜索，直到所有章节都搜索完毕

解释一下：man手册分为9章(不同系统可能会有差别)

- 1 是普通的命令
- 2 是系统调用,如open,write之类的(通过这个，至少可以很方便的查到调用这个函数，需要加什么头文件)
- 3 是库函数,如printf,fread4是特殊文件,也就是/dev下的各种设备文件
- 4 略
- 5 是指文件的格式,比如passwd, 就会说明这个文件中各个字段的含义
- 6 是给游戏留的,由各个游戏自己定义
- 7 是附件还有一些变量,比如像environ这种全局变量在这里就有说明
- 8 是系统管理用的命令,这些命令只能由root使用,如ifconfig
- 9 略

举例：

```
1 [whb@bite-alicloud test]$ man printf # 查看printf指令(没错，printf也是一个指令)
```

```
2 [whb@bite-alicloud test]$ man fork      # 查看系统调用，默认没有这个指令，man就自动去2号手册查找
3 [whb@bite-alicloud test]$ man 2 frok    # 查看系统调用
4 [whb@bite-alicloud test]$ man 3 printf  # 查看C库函数
5 [whb@bite-alicloud test]$ man 7 signal  # 查看变量
6 [whb@bite-alicloud test]$ man 7 environ
7 [whb@bite-alicloud test]$ man man      # 查看man手册自己
```

cp 指令

语法: `cp [选项] 源文件或目录 目标文件或目录`

功能: 复制文件或目录

说明:

- cp指令用于复制文件或目录
- 如同时指定两个以上的文件或目录，且最后的目的地是一个已经存在的目录，则它会把前面指定的所有文件或目录复制到此目录中

常用选项

- -f 或 --force 强行复制文件或目录，不论目的文件或目录是否已经存在
- -i 或 --interactive 覆盖文件之前先询问用户
- -r 递归处理，将指定目录下的文件与子目录一并处理。若源文件或目录的形态，不属于目录或符号链接，则一律视为普通文件处理

```
1 # cp普通文件
2 [whb@bite-alicloud test]$ echo "你好比特就业课">myfile.txt
3 [whb@bite-alicloud test]$ cat myfile.txt
4 你好比特就业课
5 [whb@bite-alicloud test]$ ll
6 total 4
7 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile.txt
8 [whb@bite-alicloud test]$ cp myfile.txt myfile-backup.txt
9 [whb@bite-alicloud test]$ ll
10 total 8
11 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile-backup.txt
12 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile.txt
13 [whb@bite-alicloud test]$ cat myfile-backup.txt
14 你好比特就业课
15
16 # 将多个文件拷贝到指定路径下
17 [whb@bite-alicloud test]$ ll
18 total 8
19 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile-backup.txt
```

```
20 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile.txt
21 [whb@bite-alicloud test]$ mkdir dir
22 [whb@bite-alicloud test]$ ll
23 total 12
24 drwxrwxr-x 2 whb whb 4096 Jan 11 15:48 dir
25 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile-backup.txt
26 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile.txt
27 [whb@bite-alicloud test]$ cp *.txt dir # *表示通配符, 匹配所有.txt结尾的文件名
28 [whb@bite-alicloud test]$ ll
29 total 12
30 drwxrwxr-x 2 whb whb 4096 Jan 11 15:48 dir
31 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile-backup.txt
32 -rw-rw-r-- 1 whb whb 22 Jan 11 15:47 myfile.txt
33 [whb@bite-alicloud test]$ tree dir
34 dir
35 |— myfile-backup.txt
36 |— myfile.txt
37
38 # cp如果目标文件存在, 就覆盖
39 [whb@bite-alicloud test]$ echo "hello bite tech" > myfile.txt
40 [whb@bite-alicloud test]$ cat myfile.txt
41 hello bite tech
42 [whb@bite-alicloud test]$ cp myfile.txt myfile-backup.txt
43 [whb@bite-alicloud test]$ cat myfile-backup.txt
44 hello bite tech
45
46 # 拷贝前询问
47 [whb@bite-alicloud test]$ cp -i myfile.txt myfile-backup.txt
48 cp: overwrite 'myfile-backup.txt'? y
49
50 # 递归强制拷贝整个目录
51 [whb@bite-alicloud ~]$ tree test
52 test
53 |— dir
54 |   |— myfile-backup.txt
55 |   |— myfile.txt
56 |— myfile-backup.txt
57 |— myfile.txt
58
59 1 directory, 4 files
60 [whb@bite-alicloud ~]$ cp -rf test test-backup
61 [whb@bite-alicloud ~]$ tree test-backup
62 test-backup
63 |— dir
64 |   |— myfile-backup.txt
65 |   |— myfile.txt
66 |— myfile-backup.txt
```

```
67 └─ myfile.txt
68
69 1 directory, 4 files
```

mv 指令

mv命令是move的缩写，可以用来移动文件或者将文件改名（move (rename) files,经常用来备份文件或者目录

语法: mv [选项] 源文件或目录 目标文件或目录

功能:

1. 视mv命令中第二个参数类型的不同（是目标文件还是目标目录），mv命令将文件重命名或将其移至一个新的目录中。
2. 当第二个参数类型是文件时，mv命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。
3. 当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，mv命令将各参数指定的源文件均移至目标目录中。

常用选项:

- -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 (destination) 已经存在时，就会询问是否覆盖！

举例:

```
1 # 更改名称
2 [whb@bite-alicloud test]$ touch myfile.txt
3 [whb@bite-alicloud test]$ ll
4 total 0
5 -rw-rw-r-- 1 whb whb 0 Jan 11 15:56 myfile.txt
6 [whb@bite-alicloud test]$ mv myfile.txt yourfile.txt
7 [whb@bite-alicloud test]$ ll
8 total 0
9 -rw-rw-r-- 1 whb whb 0 Jan 11 15:56 yourfile.txt
10
11 # 如果当前路径存在同名文件，改名即覆盖
12 [whb@bite-alicloud test]$ touch myfile.txt
13 [whb@bite-alicloud test]$ ll
14 total 0
15 -rw-rw-r-- 1 whb whb 0 Jan 11 15:58 myfile.txt
16 -rw-rw-r-- 1 whb whb 0 Jan 11 15:57 yourfile.txt
17 [whb@bite-alicloud test]$ mv yourfile.txt myfile.txt
18 [whb@bite-alicloud test]$ ll
19 total 0
```



```
20 -rw-rw-r-- 1 whb whb 0 Jan 11 15:57 myfile.txt
21
22 # 如果当前路径存在同名文件, 改名前可以添加i选项, 让系统提出警告供用户做选择
23 [whb@bite-alicloud test]$ touch yourfile.txt
24 [whb@bite-alicloud test]$ ll
25 total 0
26 -rw-rw-r-- 1 whb whb 0 Jan 11 15:57 myfile.txt
27 -rw-rw-r-- 1 whb whb 0 Jan 11 15:59 yourfile.txt
28 [whb@bite-alicloud test]$ mv -i yourfile.txt myfile.txt
29 mv: overwrite 'myfile.txt'? y
30 [whb@bite-alicloud test]$ ll
31 total 0
32 -rw-rw-r-- 1 whb whb 0 Jan 11 15:59 myfile.txt
33
34 # mv整个目录
35 [whb@bite-alicloud test]$ touch myfile.txt
36 [whb@bite-alicloud test]$ mkdir temp
37 [whb@bite-alicloud test]$ ll
38 total 4
39 -rw-rw-r-- 1 whb whb 0 Jan 11 17:56 myfile.txt
40 drwxrwxr-x 2 whb whb 4096 Jan 11 17:56 temp
41 [whb@bite-alicloud test]$ mv myfile.txt temp
42 [whb@bite-alicloud test]$ ls
43 temp
44 [whb@bite-alicloud test]$ mv temp ../
45 [whb@bite-alicloud test]$ ls
46 [whb@bite-alicloud test]$ ls -d ../temp
47 ../temp
```

cat 指令

语法: `cat [选项] [文件]`

功能: 查看目标文件的内容

常用选项:

- -b 对非空输出行编号
- -n 对输出的所有行编号
- -s 不输出多行空行

举例:

```
1 # 命令行构建多行文本
2 [whb@bite-alicloud test]$ cnt=0; while [ $cnt -le 20 ]; do echo "hello bite";
  let cnt++; done > temp.txt
3
```

```
4 # 测试cat基本命令
5 [whb@bite-alicloud test]$ cat temp.txt
6 hello bite
7 hello bite
8 hello bite
9 hello bite
10 hello bite
11 hello bite
12 hello bite
13 hello bite
14 hello bite
15 hello bite
16 hello bite
17 hello bite
18 hello bite
19 hello bite
20 hello bite
21 hello bite
22 hello bite
23 hello bite
24 hello bite
25 hello bite
26 hello bite
27
28 # cat 输出携带行号
29 [whb@bite-alicloud test]$ cat -b temp.txt
30      1      hello bite
31      2      hello bite
32      3      hello bite
33      4      hello bite
34      5      hello bite
35      6      hello bite
36      7      hello bite
37      ...
38
39 # 修改temp.txt, 使其携带多行空行
40 [whb@bite-alicloud test]$ vim temp.txt
41
42 # 测试 -b 对非空输出行编号
43 [whb@bite-alicloud test]$ cat -b temp.txt
44      1      hello bite
45      2      hello bite
46      3      hello bite
47
48
49
50      4      hello bite
```

```
51      5      hello bite
52
53      6      hello bite
54      7      hello bite
55      ...
56 # 测试-n 对输出的所有行编号
57 [whb@bite-alicloud test]$ cat -n temp.txt
58      1      hello bite
59      2      hello bite
60      3      hello bite
61      4
62      5
63      6
64      7      hello bite
65      8      hello bite
66      9
67     10      hello bite
68     11      hello bite
69     ...
70 # 测试 -s 不输出多行空行，多行空行压缩成为一行
71 [whb@bite-alicloud test]$ cat -s temp.txt
72 hello bite
73 hello bite
74 hello bite
75
76 hello bite
77 hello bite
78
79 hello bite
80 hello bite
81 hello bite
82 ...
```

more 指令

语法: `more [选项]`

功能: more命令，功能类似 cat

常用选项:

- -n 指定输出行数
- q 退出more

举例:

```
1 # 命令行输出多行文本
```

```
2 [whb@bite-alicloud test]$ cnt=0; while [ $cnt -le 2000 ]; do echo "hello
bite"; let cnt++; done > temp.txt
3
4 # -n 指定输出行数
5 [whb@bite-alicloud test]$ more -10 temp.txt
6 hello bite
7 hello bite
8 hello bite
9 hello bite
10 hello bite
11 hello bite
12 hello bite
13 hello bite
14 hello bite
15 hello bite
16 --More--(0%)
```

less 指令

- less 工具也是对文件或其它输出进行分页显示的工具，应该说是linux正统查看文件内容的工具，功能极其强大
- less 的用法比起 more 更加的有弹性，在 more 的时候，我们并没有办法向前面翻，只能往后面看
- 但若使用了 less 时，就可以使用 [pageup] [pagedown] 等按键的功能来往前往后翻看文件，更容易用来查看一个文件的内容
- 除此之外，在 less 里头可以拥有更多的搜索功能，不止可以向下搜，也可以向上搜。

语法： less [参数] 文件

功能： less与more类似，但使用less可以随意浏览文件，而more仅能向前移动，却不能向后移动，而且less在查看之前不会加载整个文件。

选项：

- -i 忽略搜索时的大小写
- -N 显示每行的行号
- /字符串：向下搜索“字符串”的功能
- ?字符串：向上搜索“字符串”的功能
- n：重复前一个搜索（与 / 或 ? 有关）
- N：反向重复前一个搜索（与 / 或 ? 有关）
- q:quit

举例：

```
1 # 命令行输出多行文本
2 [whb@bite-alicloud test]$ cnt=0; while [ $cnt -le 2000 ]; do echo "hello
  $cnt"; let cnt++; done > temp.txt
3
4 # 测试搜索和-N等功能
5 [whb@bite-alicloud test]$ less -N temp.txt
6      1 hello 0
7      2 hello 1
8      3 hello 2
9      4 hello 3
10     5 hello 4
11     6 hello 5
12     7 hello 6
13     8 hello 7
14     9 hello 8
15    10 hello 9
16    11 hello 10
17    12 hello 11
18    13 hello 12
19    14 hello 13
20     ...
21 # 其他选项也可以直接测试
```

head 指令

head 与 tail 就像它的名字一样的浅显易懂，它是用来显示开头或结尾某个数量的文字区块，head 用来显示档案的开头至标准输出中，而 tail 想当然尔就是看档案的结尾。

语法：

head [参数]... [文件]...

功能：

head 用来显示档案的开头至标准输出中，默认head命令打印其相应文件的开头10行。

选项：

- -n<行数> 显示的行数

举例：

```
1 [whb@bite-alicloud test]$ head temp.txt
2 hello 0
3 hello 1
4 hello 2
5 hello 3
6 hello 4
7 hello 5
8 hello 6
```

```
9 hello 7
10 hello 8
11 hello 9
12 [whb@bite-alicloud test]$ head -5 temp.txt
13 hello 0
14 hello 1
15 hello 2
16 hello 3
17 hello 4
```

tail 指令

tail 命令从指定点开始将文件写到标准输出.使用tail命令的-f选项可以方便的查阅正在改变的日志文件,tail -f filename会把filename里最尾部的内容显示在屏幕上,并且不断刷新,使你看到最新的文件内容.

语法: tail 必要参数 [文件]

功能: 用于显示指定文件末尾内容,不指定文件时,作为输入信息进行处理。常用查看日志文件。

选项:

- -f 循环读取
- -n<行数> 显示行数

举例: (简单讲解重定向和管道)

```
1 # 基本功能演示
2 [whb@bite-alicloud test]$ tail temp.txt
3 hello 1991
4 hello 1992
5 hello 1993
6 hello 1994
7 hello 1995
8 hello 1996
9 hello 1997
10 hello 1998
11 hello 1999
12 hello 2000
13 [whb@bite-alicloud test]$ tail -3 temp.txt
14 hello 1998
15 hello 1999
16 hello 2000
```

模拟动态日志查看功能


```
while ;; do echo "hello bite"; sleep 1; done >> temp.txt
```

```
[whb@bite-alicloud test]$ tail -f temp.txt  
hello 1998  
hello 1999  
hello 2000  
hello bite  
hello bite  
hello bite  
hello bite  
hello bite  
hello bite  
hello bite  
hello bite  
hello bite  
hello bite
```

```
1 # 综合问题  
2 # 如何显示文件的[180,200]行的内容  
3 [whb@bite-alicloud test]$ head -200 temp.txt | tail -20  
4 hello 180  
5 hello 181  
6 hello 182  
7 hello 183  
8 hello 184  
9 hello 185  
10 hello 186  
11 hello 187  
12 hello 188  
13 hello 189  
14 hello 190  
15 hello 191  
16 hello 192  
17 hello 193  
18 hello 194  
19 hello 195  
20 hello 196  
21 hello 197  
22 hello 198  
23 hello 199
```

date 指令

指定格式显示时间: `date +%Y:%m:%d`

用法: `date [OPTION]... [+FORMAT]`

1. 在显示方面, 使用者可以设定欲显示的格式, 格式设定为一个加号后接数个标记, 其中常用的标记列表如下 ☒

- %H: 小时(00..23)
- %M: 分钟(00..59)
- %S: 秒(00..61)

- %X: 相当于 %H:%M:%S
- %d: 日 (01..31)
- %m: 月份 (01..12)
- %Y: 完整年份 (0000..9999)
- %F: 相当于 %Y-%m-%d

2. 在设定时间方面

- date -s //设置当前时间，只有root权限才能设置，其他只能查看。
- date -s 20080523 //设置成20080523，这样会把具体时间设置成空00:00:00
- date -s 01:01:01 //设置具体时间，不会对日期做更改
- date -s "01:01:01 2008-05-23" //这样可以设置全部时间
- date -s "01:01:01 20080523" //这样可以设置全部时间
- date -s "2008-05-23 01:01:01" //这样可以设置全部时间
- date -s "20080523 01:01:01" //这样可以设置全部时间

3. 时间戳

- 时间->时间戳: date +%s
- 时间戳->时间: date -d@1508749502
- Unix时间戳（英文为Unix epoch, Unix time, POSIX time 或 Unix timestamp）是从1970年1月1日（UTC/GMT的午夜）开始所经过的秒数，不考虑闰秒

4. 举例

```
1 # 显示常规时间
2 [whb@bite-alicloud test]$ date
3 Wed Jan 11 18:22:50 CST 2023
4 [whb@bite-alicloud test]$ date +%Y/%m/%d
5 2023/01/11
6 [whb@bite-alicloud test]$ date +%Y/%m/%d-%H:%M:%S
7 2023/01/11-18:23:20
8
9 # 显示时间戳
10 [whb@bite-alicloud test]$ date +%s
11 1673432636
12
13 # 时间戳转成可视时间
14 [whb@bite-alicloud test]$ date +%Y/%m/%d-%H:%M:%S -d @0
15 1970/01/01-08:00:00
16 [whb@bite-alicloud test]$ date +%Y/%m/%d-%H:%M:%S -d @100000
17 1970/01/02-11:46:40
```

```
18 [whb@bite-alicloud test]$ date +%Y/%m/%d-%H:%M:%S -d @10000000000
19 2001/09/09-09:46:40
```

cal 指令

cal命令可以用来显示公历（阳历）日历。公历是现在国际通用的历法，又称格列历，通称阳历。“阳历”又名“太阳历”，系以地球绕行太阳一周为一年，为西方各国所通用，故又名“西历”。

命令格式： cal 参数 [年份]

功能：用于查看日历等时间信息，如只有一个参数，则表示年份(1-9999)，如有两个参数，则表示月份和年份

常用选项：

- -3 显示系统前一个月，当前月，下一个月的月历
- -j 显示在当年中的第几天（一年日期按天算，从1月1号算起，默认显示当前月在一年中的天数）
- -y 显示当前年份的日历

举例：

```
1 # 常规样例
2 [whb@bite-alicloud test]$ cal
3     January 2023
4 Su Mo Tu We Th Fr Sa
5  1  2  3  4  5  6  7
6  8  9 10 11 12 13 14
7 15 16 17 18 19 20 21
8 22 23 24 25 26 27 28
9 29 30 31
10
11 [whb@bite-alicloud test]$ cal 1949
12                     1949
13
14             January             February             March
15 Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
16                      1          1  2  3  4  5          1  2  3  4  5
17  2  3  4  5  6  7  8    6  7  8  9 10 11 12    6  7  8  9 10 11 12
18  9 10 11 12 13 14 15    13 14 15 16 17 18 19    13 14 15 16 17 18 19
19 16 17 18 19 20 21 22    20 21 22 23 24 25 26    20 21 22 23 24 25 26
20 23 24 25 26 27 28 29    27 28                27 28 29 30 31
21 ...
22
23 [whb@bite-alicloud test]$ cal -3
24     December 2022          January 2023          February 2023
25 Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
```

| | | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 26 | | 1 | 2 | 3 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | 1 | 2 | 3 | 4 | | | | |
| 27 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 28 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 29 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 30 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | | 29 | 30 | 31 | | | | | 26 | 27 | 28 | | | | |
| 31 | | | | | | | | | | | | | | | | | | | | | | |
| 32 | | | | | | | | | | | | | | | | | | | | | | |

find 指令

- Linux下find命令在目录结构中搜索文件，并执行指定的操作。
- Linux下find命令提供了相当多的查找条件，功能很强大。由于find具有强大的功能，所以它的选项也很多，其中大部分选项都值得我们花时间来了解一下。
- 即使系统中含有网络文件系统(NFS)，find命令在该文件系统中同样有效，只你具有相应的权限。
- 在运行一个非常消耗资源的find命令时，很多人都倾向于把它放在后台执行，因为遍历一个大的文件系统可能会花费很长的时间(这里是指30G字节以上的文件系统)。

语法: `find pathname -options`

功能: 用于在文件树中查找文件，并作出相应的处理（可能访问磁盘）

常用选项

- -name 按照文件名查找文件
- 其他选项需要在查，这个命令其实比较复杂

```
1 # 在指定路径下搜索执行名称的文件
2 [whb@bite-alicloud test]$ find ~ -name test.c
3 /home/whb/test_code/old_code/test.c
4 /home/whb/test_code/test.c
5 ...
```

which 指令

功能: 搜索系统指定的命令

举例:

```
1 [whb@bite-alicloud test]$ which ls
2 alias ls='ls --color=auto'
3 /usr/bin/ls
4 [whb@bite-alicloud test]$ which pwd
```

```
5 /usr/bin/pwd
```

whereis 指令

功能：用于找到程序的源、二进制文件或手册

举例：

```
1 [whb@bite-alicloud test]$ whereis ls
2 ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz /usr/share/man/man1p/ls.1p.gz
3 [whb@bite-alicloud test]$ whereis libc.so
4 libc: /usr/lib64/libc.so /usr/lib64/libc.a /usr/share/man/man7/libc.7.gz
```

alias 指令

功能：设置命令的别名

举例：

```
1 [whb@bite-alicloud test]$ alias hello='ls -a -l -n'
2 [whb@bite-alicloud test]$ which hello
3 alias hello='ls -a -l -n'
4 /usr/bin/ls
5 [whb@bite-alicloud test]$ hello
6 total 36
7 drwxrwxr-x  2 1003 1003  4096 Jan 11 17:59 .
8 drwx----- 22 1003 1003  4096 Jan 11 17:57 ..
9 -rw-rw-r--  1 1003 1003 28667 Jan 11 18:29 temp.txt
```

grep 指令

语法： `grep [选项] 搜寻字符串 文件`

功能：在文件中搜索字符串，将找到的行打印出来

常用选项：

- -i：忽略大小写的不同，所以大小写视为相同
- -n：顺便输出行号
- -v：反向选择，亦即显示出没有 '搜寻字符串' 内容的那一行

举例：

```
1 # 文件内容
```

```
2 [whb@bite-alicloud test]$ cat temp.txt
3 abcd
4 ABCD
5 hello
6 bite
7 1234
8
9 #基本查找
10 [whb@bite-alicloud test]$ grep "abcd" temp.txt
11 abcd
12
13 # 忽略大小写的不同, 所以大小写视为相同
14 [whb@bite-alicloud test]$ grep -i "abcd" temp.txt
15 abcd
16 ABCD
17
18 # 顺便输出行号
19 [whb@bite-alicloud test]$ grep -n "abcd" temp.txt
20 1:abcd
21 [whb@bite-alicloud test]$ grep -ni "abcd" temp.txt
22 1:abcd
23 2:ABCD
24
25 # 反向选择, 亦即显示出没有 '搜寻字符串' 内容的那一行
26 [whb@bite-alicloud test]$ grep -v "abcd" temp.txt
27 ABCD
28 hello
29 bite
30 1234
31 [whb@bite-alicloud test]$ grep -vn "abcd" temp.txt
32 2:ABCD
33 3:hello
34 4:bite
35 5:1234
36 [whb@bite-alicloud test]$ grep -vni "abcd" temp.txt
37 3:hello
38 4:bite
39 5:1234
```

zip/unzip 指令

语法: zip 压缩文件.zip 目录或文件

功能: 将目录或文件压缩成zip格式

常用选项

- -r: 递归处理, 将指定目录下的所有文件和子目录一并处理

举例

将test2目录压缩: `zip test2.zip test2/*`

解压到tmp目录: `unzip test2.zip -d /tmp`

关于 rzsz

这个工具用于 windows 机器和远端的 Linux 机器通过 XShell 传输文件. 安装完毕之后可以通过拖拽的方式将文件上传过去.

```
1 sudo yum/apt install -y lrzsz
```

现场演示windows和Linux如何互传文件

tar 指令 (重要)

打包/解包, 不打开它, 直接看内容

语法: `tar [-cxtzjvf] 文件与目录`

参数: ☒

- -c : 建立一个压缩文件的参数指令(create 的意思);
- -x : 解开一个压缩文件的参数指令!
- -t : 查看 tarfile 里面的文件!
- -z : 是否同时具有 gzip 的属性? 亦即是否需要用 gzip 压缩?
- -j : 是否同时具有 bzip2 的属性? 亦即是否需要用 bzip2 压缩?
- -v : 压缩的过程中显示文件! 这个常用, 但不建议用在背景执行过程!
- -f : 使用档名, 请留意, 在 f 之后要立即接档名喔! 不要再加参数!
- -C : 解压到指定目录

案例:

范例一: 将整个 /etc 目录下的文件全部打包成为 `/tmp/etc.tar`

```
1 [root@linux ~]$ tar -cvf /tmp/etc.tar /etc<==仅打包, 不压缩!
2 [root@linux ~]$ tar -zcvf /tmp/etc.tar.gz /etc <==打包后, 以 gzip 压缩
3 [root@linux ~]$ tar -jcvf /tmp/etc.tar.bz2 /etc <==打包后, 以 bzip2 压缩
```

特别注意, 在参数 f 之后的文件档名是自己取的, 我们习惯上都用 .tar 来作为辨识。

如果加 z 参数，则以 .tar.gz 或 .tgz 来代表 gzip 压缩过的 tar file ～

如果加 j 参数，则以 .tar.bz2 来作为附档名啊～

上述指令在执行的时候，会显示一个警告讯息：

『`tar: Removing leading `/' from member names`』那是关于绝对路径的特殊设定。

范例二：查阅上述 /tmp/etc.tar.gz 文件内有哪些文件？

```
1 [root@linux ~]$ tar -ztvf /tmp/etc.tar.gz
```

由於我们使用 gzip 压缩，所以要查阅该 tar file 内的文件时，就得要加上 z 这个参数了！这很重要的！

范例三：将 /tmp/etc.tar.gz 文件解压缩在 /usr/local/src 底下

```
1 [root@linux ~]$ cd /usr/local/src
2 [root@linux src]$ tar -zxvf /tmp/etc.tar.gz
```

在预设的情况下，我们可以将压缩档在任何地方解开的！以这个范例来说，我先将工作目录变换到 /usr/local/src 底下，并且解开 /tmp/etc.tar.gz，则解开的目录会在 /usr/local/src/etc 呢！另外，如果您进入 /usr/local/src/etc

则会发现，该目录下的文件属性与 /etc/ 可能会有所不同喔！

范例四：在 /tmp 底下，我只想要将 /tmp/etc.tar.gz 内的 etc/passwd 解开而已

```
1 [root@linux ~]$ cd /tmp
2 [root@linux tmp]$ tar -zxvf /tmp/etc.tar.gz etc/passwd
```

我可以透过 tar -ztvf 来查阅 tarfile 内的文件名称，如果单只要一个文件，就可以透过这个方式来下达！注意到！ etc.tar.gz 内的根目录 / 是被拿掉了！

范例五：将 /etc/ 内的所有文件备份下来，并且保存其权限！

```
1 [root@linux ~]$ tar -zxvpf /tmp/etc.tar.gz /etc
2 # 这个 -p 的属性是很重要的，尤其是当您想要保留原本文件的属性时！
```

范例六：在 /home 当中，比 2005/06/01 新的文件才备份

```
1 [root@linux ~]$ tar -N "2005/06/01" -zcvf home.tar.gz /home
```

范例七：我要备份 /home,/etc ，但不要 /home/dmtsai

```
1 [root@linux ~]$ tar --exclude /home/dmtsai -zcvf myfile.tar.gz /home/* /etc
```

范例八：将 /etc/ 打包后直接解开在 /tmp 底下，而不产生文件！

```
1 [root@linux ~]$ cd /tmp
2 [root@linux tmp]$ tar -cvf - /etc | tar -xvf -
```

这个动作有点像是 `cp -r /etc /tmp` 啦～依旧是有其有用途的！

要注意的地方在於输出档变成 - 而输入档也变成 - ，又有一个 | 存在～

这分别代表 standard output, standard input 与管线命令啦！

现场演示如何在Linux和Linux之间互传压缩包

bc 指令

bc命令可以很方便的进行浮点运算

uname -r 指令

语法：uname [选项]

功能：uname用来获取电脑和操作系统的的相关信息。

补充说明：uname可显示linux主机所用的操作系统的版本、硬件的名称等基本信息。

常用选项：☒

- -a或-all 详细输出所有信息，依次为内核名称，主机名，内核版本号，内核版本，硬件名，处理器类型，硬件平台类型，操作系统名称

重要的几个热键 [Tab], [ctrl]-c, [ctrl]-d

- [Tab]按键---具有『命令补全』和『档案补齐』的功能
- [Ctrl]-c按键---让当前的程序『停掉』
- [Ctrl]-d按键---通常代表着：『键盘输入结束(End Of File, EOF 戒 End OfInput)』的意思；另外，他也可以用来取代exit

关机

语法：shutdown [选项]

常见选项：

- -h：将系统的服务停掉后，立即关机。
- -r：在将系统的服务停掉之后就重新启动
- -t sec：-t 后面加秒数，亦即『过几秒后关机』的意思

以下命令作为扩展

- **安装和登录命令**：login、shutdown、halt、reboot、install、mount、umount、chsh、exit、last；
- **文件处理命令**：file、mkdir、grep、dd、find、mv、ls、diff、cat、ln；
- **系统管理相关命令**：df、top、free、quota、at、lp、adduser、groupadd、kill、crontab；
- **网络操作命令**：ifconfig、ip、ping、netstat、telnet、ftp、route、rlogin、rcp、finger、mail、nslookup；
- **系统安全相关命令**：passwd、su、umask、chgrp、chmod、chown、chattr、sudo ps、who；
- **其它命令**：tar、unzip、gunzip、unarj、mtools、man、unendcode、uudecode。

shell命令以及运行原理

Linux严格意义上说的是一个操作系统，我们称之为“核心（kernel）”，但我们一般用户，不能直接使用kernel。而是通过kernel的“外壳”程序，也就是所谓的shell，来与kernel沟通。如何理解？为什么不能直接使用kernel？

从技术角度，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。

对比windows GUI，我们操作windows 不是直接操作windows内核，而是通过图形接口，点击，从而完成我们的操作（比如进入D盘的操作，我们通常是双击D盘盘符.或者运行起来一个应用程序）。

shell 对于Linux，有相同的作用，主要是对我们的指令进行解析，解析指令给Linux内核。反馈结果在通过内核运行出结果，通过shell解析给用户。

- 帮助理解：如果说你是一个闷骚且害羞的程序员，那shell就像媒婆，操作系统内核就是你们村头漂亮的且有让你心动的MM小花。你看上了小花，但是有不好意思直接表白，那就让你你家人找媒婆帮你提亲，所有的事情你都直接跟媒婆沟通，由媒婆转达你的意思给小花，而我们找到媒婆姓王，所以我们叫它王婆，它对应我们常使用的bash。

Linux权限的概念

Linux下有两种用户：超级用户（root）、普通用户。

- 超级用户：可以再linux系统下做任何事情，不受限制

- 普通用户：在linux下做有限的事情。
- 超级用户的命令提示符是“#”，普通用户的命令提示符是“\$”。

命令：su [用户名]

功能：切换用户。

例如，要从root用户切换到普通用户user，则使用 su user。

要从普通用户user切换到root用户则使用 su root（root可以省略），此时系统会提示输入root用户的口令。

Linux权限管理

1. 文件访问者的分类（人）

- 文件和文件目录的所有者：u---User（中国平民 法律问题）
- 文件和文件目录的所有者所在的组的用户：g---Group（不多说）
- 其它用户：o---Others（外国人）

2. 文件类型和访问权限（事物属性）

- 文件类型：
 - d：文件夹
 - -：普通文件
 - l：软链接（类似Windows的快捷方式）
 - b：块设备文件（例如硬盘、光驱等）
 - p：管道文件
 - c：字符设备文件（例如屏幕等串口设备）
 - s：套接口文件
- 基本权限：
 - 读（r/4）：Read对文件而言，具有读取文件内容的权限；对目录来说，具有浏览该目录信息的权限
 - 写（w/2）：Write对文件而言，具有修改文件内容的权限；对目录来说具有删除移动目录内文件的权限
 - 执行（x/1）：execute对文件而言，具有执行文件的权限；对目录来说，具有进入目录的权限
 - “—”表示不具有该项权限

文件权限值的表示方法☒

- 字符表示方法

- 8进制数值表示方法

文件访问权限的相关设置方法

1. chmod

功能：设置文件的访问权限

格式：chmod [参数] 权限 文件名

常用选项：

- R -> 递归修改目录文件的权限
- 说明：只有文件的拥有者和root才可以改变文件的权限

chmod命令权限值的格式：

- 用户表示符+/-=权限字符
 - +:向权限范围增加权限代号所表示的权限
 - -:向权限范围取消权限代号所表示的权限
 - =:向权限范围赋予权限代号所表示的权限
 - 用户符号：
 - u：拥有者
 - g：拥有者同组用
 - o：其它用户
 - a：所有用户
 - 实例：
 - `chmod u+w /home/abc.txt`
 - `chmod o-x /home/abc.txt`
 - `chmod a=x /home/abc.txt`
- 三位8进制数字
 - 实例：
 - `chmod 664 /home/abc.txt`
 - `chmod 640 /home/abc.txt`

2. chown

功能：修改文件的拥有者

格式：chown [参数] 用户名 文件名

实例：

- `chown user1 f1`
- `chown -R user1 filegroup1`

3. chgrp

功能：修改文件或目录的所属组

格式：`chgrp [参数] 用户组名 文件名`

常用选项：`-R` 递归修改文件或目录的所属组

实例： `chgrp users /abc/f2`

4. umask

功能：

- 查看或修改文件掩码
- 新建文件夹默认权限=0666
- 新建目录默认权限=0777
- 但实际上你所创建的文件和目录，看到的权限往往不是上面这个值。原因就是创建文件或目录的时候还要受到umask的影响。假设默认权限是mask，则实际创建的出来的文件权限是: `mask & ~umask`

格式：`umask 权限值`

说明：将现有的存取权限减去权限掩码后，即可产生建立文件时预设权限。超级用户默认掩码值为0022，普通用户默认为0002。

实例：

- `umask 755`
- `umask //查看`
- `umask 044//设置`

file 指令

功能说明：辨识文件类型。

语法： `file [选项] 文件或目录...`

常用选项：

- `-c`：详细显示指令执行过程，便于排错或分析程序执行的情形。
- `-z`：尝试去解读压缩文件的内容。

使用 sudo 分配权限

(1) 修改/etc/sudoers 文件分配文件

```
1 # chmod 740 /etc/sudoers
2 # vi /etc/sudoer
```

格式：接受权限的用户登陆的主机 = (执行命令的用户) 命令

(2) 使用 sudo 调用授权的命令

```
1 $ sudo -u 用户名 命令
```

实例：

```
1 $ sudo -u root /usr/sbin/useradd u2
```

目录的权限

- 可执行权限: 如果目录没有可执行权限, 则无法cd到目录中.
- 可读权限: 如果目录没有可读权限, 则无法用ls等命令查看目录中的文件内容.
- 可写权限: 如果目录没有可写权限, 则无法在目录中创建文件, 也无法在目录中删除文件.

于是, 问题来了~~ 换句话来讲, 就是只要用户具有目录的写权限, 用户就可以删除目录中的文件, 而不论这个用户是否有这个文件的写权限.

这好像不太科学啊, 我张三创建的一个文件, 凭什么被你李四可以删掉? 我们用下面的过程印证一下.

```
1 [root@localhost ~]$ chmod 0777 /home/
2 [root@localhost ~]$ ls /home/ -ld
3 drwxrwxrwx. 3 root root 4096 9月 19 15:58 /home/
4 [root@localhost ~]$ touch /home/root.c
5 [root@localhost ~]$ ls -l /home/
6 总用量 4
7 -rw-r--r--. 1 root root 0 9月 19 15:58 abc.c
8 drwxr-xr-x. 27 litao litao 4096 9月 19 15:53 litao
9 -rw-r--r--. 1 root root 0 9月 19 15:59 root.c
10
11 [root@localhost ~]$ su - litao
12 [litao@localhost ~]$ rm /home/root.c #litao可以删除root创建的文件
13 rm: 是否删除有写保护的普通空文件 "/home/root.c"? y
14
15 [litao@localhost ~]$ exit
16 logout
```


为了解决这个不科学的问题, Linux引入了粘滞位的概念.

关于权限的总结

- 目录的可执行权限是表示你可否在目录下执行命令。
- 如果目录没有 -x 权限, 则无法对目录执行任何命令, 甚至无法 cd 进入目, 即使目录仍然有 -r 读权限 (这个地方很容易犯错, 认为有读权限就可以进入目录读取目录下的文件)
- 而如果目录具有 -x 权限, 但没有 -r 权限, 则用户可以执行命令, 可以 cd 进入目录。但由于没有目录的读权限
- 所以在目录下, 即使可以执行 ls 命令, 但仍然没有权限读出目录下的文档。

粘滞位 [选学]

```
1 [root@localhost ~]$ chmod +t /home/ # 加上粘滞位
2 [root@localhost ~]$ ls -ld /home/
3 drwxrwxrwt. 3 root root 4096 9月 19 16:00 /home/
4 [root@localhost ~]$ su - litao
5 [litao@localhost ~]$ rm /home/abc.c #litao不能删除别人的文件
6 rm: 是否删除有写保护的普通空文件 "/home/abc.c"? y
7 rm: 无法删除"/home/abc.c": 不允许的操作
```

当一个目录被设置为"粘滞位"(用chmod +t),则该目录下的文件只能由

1. 超级管理员删除
2. 该目录的所有者删除
3. 该文件的所有者删除