

第05章 文件和文件夹管理

讲师：武永亮

课程目标

- 了解CentOS的文件目录结构
- 掌握CentOS的文件目录管理命令
- 掌握文件打包和压缩命令

课程内容



文件的类型

- 普通文件 (-)
- 目录 (d)
- 符号链接 (l)
- 字符设备文件 (c)
- 块设备文件 (b)
- 套接字 (s)
- 命名管道 (p)

普通文件

- 普通文件仅仅就是字节序列，Linux 并没有对其内容规定任何的结构。
- 普通文件可以是程序源代码（c、c++、python、perl等）、可执行文件（文件编辑器、数据库系统、出版工具、绘图工具等）、图片、声音、图像等。
- Linux 不会区别对待这些普通文件，只有处理这些文件的应用程序才会根据文件的内容赋予相应的含义。
- 在Linux环境下，只要是可执行的文件并具有可执行属性它就能执行，不管其文件名后缀是什么。但是对一些数据文件一般也遵循一些文件名后缀规则。

目录

- 目录文件是由一组目录项组成，目录项可以是对其他文件的指向也可以是其下的子目录指向。
- 一个文件的名称是存储在他的父目录中的，而并非同文件内容本身存储在一起。
- 硬连接文件实际上就是在某目录中创建目录项，从而使不止一个目录可以引用到同一个文件。这种链接关系由 `ln` 命令行来建立。
- 硬链接并不是一种特殊类型的文件，只是因为文件系统中允许不止一个目录项指向同一个文件。

用户主目录

- 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账号。如
✓ 用户osmond的主目录为：`/home/osmond`
- 在创建用户时，系统管理员会给每个用户建立一个主目录，通常在 `/home/` 目录下。
- 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。
- 每个用户名对应一个用户 ID 号（一个数字）；每个用户都被分配到一个指定的组（group）中。
- 默认情况下 RHEL/CentOS 在创建用户的同时会创建一个和用户同名的私有组。

符号链接

- 符号链接又称软链接，是指将一个文件指向另外一个文件的文件名。
- 这种符号链接的关系由 `ln -s` 命令行来建立。

硬链接和软链接的比较

- 硬链接

- ✓ 链接文件和被链接文件必须位于同一个文件系统内
- ✓ 不能建立指向目录的硬链接

- 软链接

- ✓ 链接文件和被链接文件可以位于不同文件系统
- ✓ 可以建立指向目录的软链接

设备文件

- 设备是指计算机中的外围硬件装置，即除了CPU和内存以外的所有设备。通常，设备中含有数据寄存器或数据缓存器、设备控制器，它们用于完成设备同CPU或内存的数据交换。
- 在Linux下，为了屏蔽用户对设备访问的复杂性，采用了设备文件，即可以通过象访问普通文件一样的方式来对设备进行访问读写。
- 设备文件用来访问硬件设备，包括硬盘、光驱、打印机等。每个硬件设备至少与一个设备文件相关联。
- 设备文件分为：字符设备（如：键盘）和块设备（如：磁盘）。

Linux下设备的使用

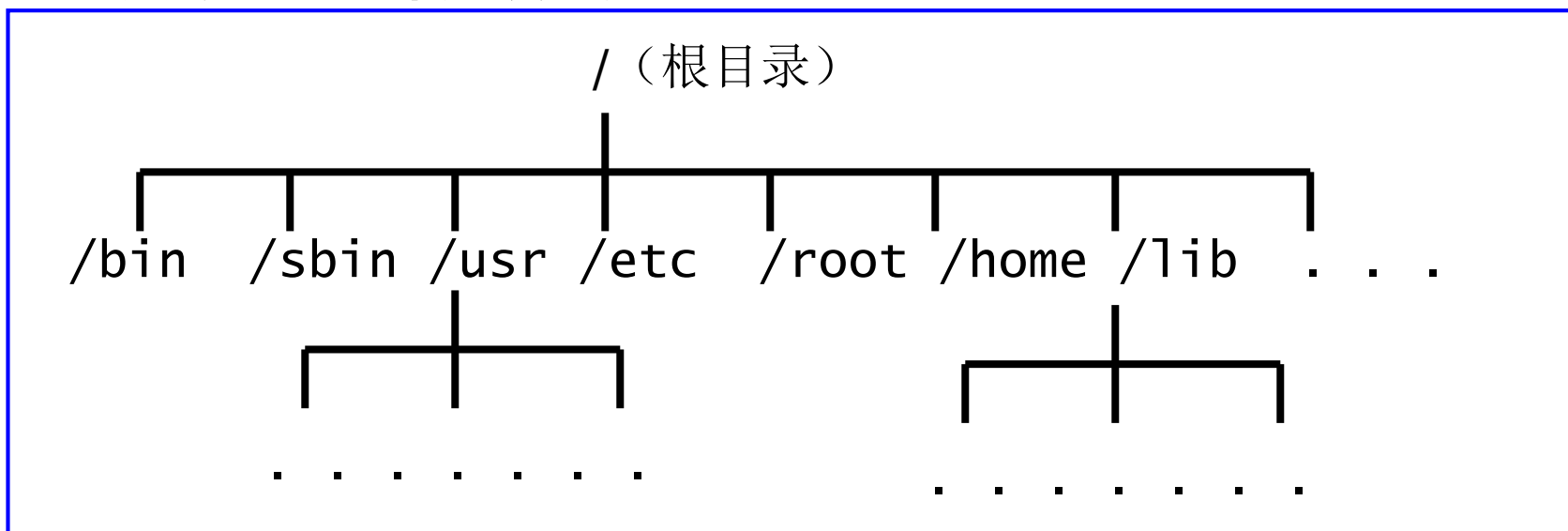
- 设备的使用方法
 - ✓ 用户可以用设备名来使用设备
 - ✓ 用户可以用访问文件的方法来使用设备
- 设备名以文件系统中的设备文件的形式存在
- 所有的设备文件存放在/dev目录下
- 几个特殊的设备
 - ✓ /dev/null - 空设备
 - ✓ /dev/zero - 零设备

套接字和命名管道

- 套接字和命名管道是 Linux 环境下实现进程间通信（IPC）的机制。
- 命名管道（FIFO）文件允许运行在同一台计算机上的两个进程之间进行通信。
- 套接字（socket）允许运行在不同计算机上的进程之间相互通信。
- 套接字和命名管道通常是在进程运行时创建或删除的，一般无需系统管理员干预。

Linux 的目录结构

- 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。
- Linux 的这种文件系统结构使得一个目录和它包含的文件/子目录之间形成一种层次关系。



/ 文件系统结构的起始位置，称为根

| -- bin 存放基本命令程序(任何用户都可以调用)

| -- boot 存放系统启动时所读取的文件，包括系统核心文件

| -- dev 存放设备文件接口，如打印机、硬盘等外围设备

| -- etc 存放与系统设置和管理相关的文件，如用户帐号、密码等

| | ...

| -- home 存放用户专属目录（用户主目录）

| -- lib 存放一些共享的函数库

| -- misc 一个空目录，供管理员存放公共文件

| -- proc 存放系统核心和执行程序之间的信息

| -- root 系统管理员（超级用户）专用目录

| -- tmp 与/bin类似，存放用于系统引导和管理命令，通常供root使用。

| -- usr 临时目录，供任何用户存放临时文件。

| | -- bin

| | -- sbin

| | -- man 此目录包含许多子目录，用来存放系统命令和程序等信息

| | ...

| -- var 存放经常变动的文件，如日志文件、临时文件、电子邮件等

...

课程内容



常用的目录操作命令

命令	功能
ls	显示文件和目录列表
cd	切换目录
pwd	显示当前工作目录
mkdir	创建目录
rmdir	删除空目录
tree	显示目录树

当前工作目录

- 用户目前所处的目录,用户登录后进入的目录通常是自己的主目录
- 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

路径 (path)

- 路径是指文件或目录在文件系统中所处的位置
- 绝对路径
 - ✓ 以斜线 (/) 开头
 - ✓ 描述到文件位置的完整说明
 - ✓ 一般用于查找系统文件
- 相对路径
 - ✓ 不以斜线 (/) 开头
 - ✓ 指定相对于你的当前工作目录而言的位置
 - ✓ 一般用于查找当前目录中的文件

ls命令

- 功能：显示文件或目录信息
- 格式：ls [选项] [目录或是文件]
- 说明：
 - ✓对于目录，该命令将列出其中的所有子目录与文件。
 - ✓对于文件，ls 将输出其文件名以及所要求的其他信息。
 - ✓默认情况下，输出条目按字母顺序排序。
 - ✓当未给出目录名或文件名时，就显示当前目录的信息。

ls命令选项

选项	说明
-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，通常称为“长格式”。
-d	输入参数是目录时，只显示该目录本身。
-A	显示除 “.” 和 “..” 外的所有文件。
-R	递归地列出所有子目录下的文件。
-h	以人类易读的单位显示文件大小。
-S	以文件大小排序输出。
-t	以时间排序输出。

ls命令举例

ls	列表显示当前目录下的文件和目录
ls -a	列表显示当前目录下的文件和目录（包括隐含文件和目录）
ls -l	以长格式列表显示结果
ls -R	递归地显示当前目录及其子目录下的文件和目录
ls -dl /usr/share/	仅显示/usr/share/目录本身，而非/usr/share/ 目录中的内容
ls -lt	按最后修改时间顺序，以长格式列出当前目录下的文件

mkdir和tree命令举例

● mkdir

- ✓ `mkdir /home/lrj/mybin` #创建一个空目录
- ✓ `mkdir -p mydoc/FAQ` #创建一个空目录树
- ✓ `mkdir -p /srv/www/{abc,bcd}/htdocs` #创建/srv/www/abc/htdocs和/srv/www/bcd/htdocs目录

● tree

- ✓ `tree /srv/www` # 显示/srv/www 目录树
- ✓ `tree -L 3 /srv` #显示 /srv/ 的 3级目录树

pwd和cd命令举例

- `$ pwd`
- `$ cd`
- `$ cd /some/dir/`
- `$ cd ~`
- `$ cd ..`
- `$ cd ../..`
- `$ cd -`

常用的文件操作命令

命令	功能
touch	生成一个空文件或更改文件的时间
cp	复制文件或目录
mv	移动文件或目录、文件或目录改名
rm	删除文件或目录
ln	建立链接文件
find	查找文件
file/stat	查看文件类型或文件属性信息

touch命令

- 功能：生成新的空文件或更改现有文件的时间
 - ✓ 若文件不存在，系统会建立一个文件
 - ✓ 默认情况下将文件的时间记录改为当前时间
- 格式：touch [参数] <文件> ...
- 参数
 - ✓ -a：只更改访问时间。
 - ✓ -m：只更改修改时间。
 - ✓ -t <STAMP>：使用[[CC]YY]MMDDhhmm[.ss]格式的时间而非当前时间。
 - ✓ -r <参考文件或目录>：使用指定文件的时间属性而非当前时间。

touch命令举例

- `$ touch newfile`
- `$ touch file`
- `$ touch -a file`
- `$ touch -m file`
- `$ touch -t 200701311200 file`
- `$ touch -r oldfile newfile`

cp命令

- 功能：复制文件或目录。
- 格式：cp [参数] <源> <目标>
- 说明
 - ✓若复制的目标文件已存在，则被覆盖。
 - ✓可以将多个源文件复制到目标目录中。
 - ✓可以将源目录复制为指定的目标目录（目标目录不存在）。
 - ✓可以将源目录复制到指定的目标目录中。

cp命令的常用选项

参数	说明
-a	等价于 <code>-dpR</code>
-R, -r	递归地复制目录及目录内的所有项目
-p	在复制文件过程中保留文件属性，包括属主、组、权限与时间戳
-d	当复制符号链接的源文件时，目标文件也将创建符号链接且指向源文件所链接的原始文件
-f	强制复制，不管目标是否存在
-i	交互式复制，覆盖文件前需要确认
-u	只有当源文件的状态改变时间（ <code>ctime</code> ）比目标文件更新时或目标尚不存在时才进行复制

cp命令举例

cp file1 file2	将当前目录下的file1复制为file2
cp some/dir/file1 someother/dir/	将某目录下的文件复制到另一个目录
cp /bin/cpio ~/bin	复制文件 /bin/cpio 到 ~/bin 目录
cp abc bcd mydoc	将两个指定的文件复制到mydoc目录下
cp some/dir/f* someother/dir/	将某目录下的以f开头的文件复制到另一个目录
cp /bin/?sh .	使用 “?”通配符复制多个文件到当前目录下
cp file{1,2,3} someother/dir/	将当前目录下的file1、file2、file3复制到另外一个目录
cp /etc/httpd/conf/httpd.conf{,.orig}	将/etc/httpd/conf/目录下的httpd.conf拷贝成httpd.conf.orig
cp -r some/dir/ someother/dir/	将某目录下的所有文件（包括子目录）复制到另一个目录

mv和rm命令举例

● mv

- ✓ mv file file.bak
- ✓ mv file1 mydata/
- ✓ mv -i file1 mydata/file2
- ✓ mv datadir1 /home/zhang/data

● rm

- ✓ rm ash
- ✓ rm .*
- ✓ rm -f file{ 1,3,5 }
- ✓ rm -r myusr/
- ✓ rm -rf myusr/



ln命令

- 功能：创建链接文件。
- 格式：ln [参数] <被链接的文件> <链接文件名>
- 参数：
 - ✓ -s：创建符号链接，而非硬链接。
 - ✓ -f：强行创建链接，不论其是否存在。
 - ✓ -i：覆盖原有文件之前先询问用户。
- 举例：
 - ✓ \$ ln somefile hardlinkfile
 - ✓ \$ ln -s somefile softlinkfile
 - ✓ \$ ln -s somedir softlinkfile

find命令

- 功能：在文件系统中查找匹配的文件,不带任何参数的 find 命令将在屏幕上递归显示当前目录下的文件列表
- 格式
 - ✓ find [<起始目录> ...] [<选项表达式>] [<条件匹配表达式>] [<动作表达式>]
- 说明
 - ✓ <起始目录>：对每个指定的 <起始目录> 递归搜索目录树
 - 省略<起始目录>时表示当前目录
 - ✓ <选项表达式>：控制 find 命令的行为
 - ✓ <条件匹配表达式>：根据匹配条件查找文件
 - ✓ <动作表达式>：指定对查找结果的操作，默认为显示在标准输出（-print）

find 命令——选项表达式

选项	说明
-L	如果遇到符号链接文件，就跟踪链接所指的文件
-regextype TYPE	指定 -regex 和 -iregex 使用的正则表达式类型，默认为 emacs
-depth/-d	查找目录自身之前先处理目录中的文件（即深度优先）
-mount/-xdev	查找文件时不跨越文件系统
-maxdepth LEVELS	设置最大的查找深度
--help	显示 find 命令帮助信息
--version	显示 find 的版本

find命令举例（1）

- 递归显示当前目录的文件列表
✓ `$ find`
- 递归显示 / 目录的文件列表
✓ `$ find /`
- 递归显示 / 目录的文件列表（仅限于3层目录）
✓ `$ find / -maxdepth 3`
- 递归显示 / 目录的文件列表（仅限于 / 文件系统）
✓ `$ find / -xdev`
- 递归显示 /home、/www、/srv 目录的文件列表
✓ `$ find /home /www /srv`

find 命令——条件表达式（1）文件名

条件	说明
-name PATTERN	匹配文件名
-iname PATTERN	匹配文件名（忽略大小写）
-lname PATTERN	匹配符号链接文件名
-ilname PATTERN	匹配符号链接文件名（忽略大小写）
-wholename PATTERN	匹配文件的完整路径（不把 ‘/’ 和 ‘.’ 作为特殊字符）
-iwholename PATTERN	匹配文件的完整路径（忽略大小写）

PATTERN 使用 Shell 的匹配模式

- 可以使用 Shell 的通配符（*、? []）
- 要用 “” 或 ‘’ 括起来

find命令举例 (2)

- `$ find -name myfile`
- `$ find -maxdepth 2 -name symfony`
- `$ find -name 'd*'`
- `$ find -name '???'`
- `$ find -name 'ch[0-2][0-9].txt*'`
- `$ find -iname '[a-z]*'`
- `$ find ~ -name "*.txt"`
- `$ find . -name "[A-Z]*"`
- `$ find /etc -name "host*"`

find 命令——条件表达式（2）文件名续

条件	说明
<code>-regex REPATTERN</code>	以正则表达式匹配文件名
<code>-iregex REPATTERN</code>	以正则表达式匹配文件名（忽略大小写）

REPATTERN 使用正则表达式匹配模式

- 使用 `-regextype TYPE` 指定的正则表达式类型
- 默认为 `emacs` 类型

find命令举例 (3)

- `$ find ~ -regex '.*'`
- `$ find ~/book -regex '.*ch0.*'`
- `$ find ~/book -regex '.*ch[0-9]+'`
- `$ find ~/book -iregex '.*ch[0-9]+\\.txt'`

find 命令——条件表达式（3）时间属性

条件	说明
-amin N	查找 N 分钟以前被访问过的所有文件
-atime N	查找 N 天以前被访问过的所有文件
-cmin N	查找 N 分钟以前文件状态被修改过的所有文件（比如权限修改）
-ctime N	查找 N 天以前文件状态被修改过的所有文件（比如权限修改）
-mmin N	查找 N 分钟以前文件内容被修改过的所有文件
-mtime N	查找 N 天以前文件内容被修改过的所有文件
-used N	查找被改动过之后，在N天内被存取过的所有文件

注：以上选项中的 N 可以有三种输入方式：N, +N, -N

- 20 表示等于 20
- +20 表示大于 20（21, 22, 23 等）
- -20 表示小于 20（19, 18, 17 等）

find命令举例（4）

- 在/home下查找最近2天之内改动过的文件
✓ `$ find /home -mtime -2`
- 在/home下查找最近2天之内被访问过的文件
✓ `$ find /home -atime -2`
- 在/home下查找60分钟之前改动过的文件
✓ `$ find /home -mmin +60`
- 在/home下查找最近5分钟之内修改过的文件
✓ `$ find /home -mmin -5`
- 在/home下查找30分钟之前被访问过的文件
✓ `$ find /home -amin +30`
- 在/home下查找最近在2日内被改动之后被访问过的文件
✓ `$ find /home -used -2`

find 命令——条件表达式（4）时间属性续

条件	说明
-newer FILE	查找所有比 FILE 的内容修改时间新的文件
-cnewer FILE	查找所有比 FILE 的状态修改时间新的文件（比如权限修改）
-anewer FILE	查找所有比 FILE 的访问时间新的文件

FILE 为要比对的样板文件或目录

find命令举例（5）

- 在/home下查找更新时间比tmp.txt新的文件或目录
✓ \$ find /home -newer tmp.txt
- 在/home下查找存取时间比tmp.txt新的文件或目录
✓ \$ find /home -anewer tmp.txt

find 命令——条件表达式（5）用户和组

条件	说明
-uid N	用户ID为N的所有文件
-gid N	组ID为N的所有文件
-user USERNAME	用户名为 USERNAME 的所有文件
-group GROUPNAME	组名为 GROUPNAME 的所有文件
-nouser	文件属于不在 /etc/passwd 文件中的用户
-nogroup	文件属于不在 /etc/group 文件中的组

N 可以使用

N, +N, -N

find命令举例（6）

- 查找在系统中已作废用户（在/etc/passwd中无记录）的文件
✓ `$ find / -nouser`
- 查找在系统中已作废组（在/etc/group中无记录）的文件
✓ `$ find / -nogroup`
- 查找属于 osmond 用户的文件
✓ `$ find / -user osmond`
- 查找属于 students 组的文件
✓ `$ find / -group students`
- 查找UID为502的所有文件
✓ `$ find / -uid 502`
- 查找UID大于600的所有文件
✓ `$ find / -uid +600`
- 查找GID小于60的所有文件
✓ `$ find / -gid -60`

find 命令——条件表达式（6）文件权限

条件	说明
-perm MODE	精确匹配权限模式为 MODE 的文件
-perm -MODE	匹配权限模式至少为 MODE 的文件 （用户、组和其他人这三种权限 必须都匹配 ）
-perm /MODE 或 -perm +MODE	匹配权限模式至少为 MODE 的文件 （用户、组和其他人这三种权限中 有一种匹配即可 ）

MODE 与 chown 命令的书写方式一致，既可以使用字符模式也可以使用八进制的数值模式，**通常使用八进制的数值模式**

find命令举例（7）

- 在/home目录下查找权限为700的所有文件或目录
✓ \$ find /home -perm 700
- 在/home目录下查找权限至少为740的所有文件或目录
✓ \$ find /home -perm -740
- 在~/bin目录下查找权限至少为111的所有文件或目录（或者属主可执行，或者组可执行，或者其他人可执行）
✓ \$ find ~/bin -perm /111

find 命令——条件表达式（7）类型 and 大小

条件	说明
-type TYPE	查找类型为 TYPE 的文件，可用的类型： b -块设备文件; c -字符设备文件; d -目录文件; p -命名管道; f -普通文件; l -符号链接; s -socket 文件
-links N	查找有N个链接的文件
-inum N	查找文件inode为N的文件
-samefile NAME	查找与NAME文件具有相同inode的文件
-size N[bcwkMG]	查找文件大小为N的文件，单位可以为： b -512 字节的块（省略单位的默认值）; c -字节; w -双字节; k -Kilobytes; M -Megabytes; G -Gigabytes
-empty	查找空文件（大小为0）

N 可以使用： **N**, **+N**, **-N**

find命令举例（8）

- 在当前目录下查找目录
 - ✓ `$find . -type d`
- 在/home目录下查找硬连接数大于2的文件
 - ✓ `$find /home -links +2`
- 在当前目录下查找长度大于10kB的文件
 - ✓ `$find . -size +10k`
- 在/tmp目录下查找长度等于1GB的文件
 - ✓ `$find /tmp -size 1G`
- 在当前目录下查找长度小于10MB的文件
 - ✓ `$find . -size -10M`
- 在/var/log目录下查找所有的空文件或目录
 - ✓ `$ find /var/log -empty`

find 命令——组合条件表达式

- 可以使用逻辑运算符与、或、非组成的复合条件，并可以用()改变默认的操作符优先级。
- 若以空格作为各个表达式的间隔符，则各个表示式之间是与关系。

(EXPR)	改变操作符优先次序，一些 UNIX 版的 find 命令要使用 \ (EXPR \) 形式
! EXPR	表示对表达式取反
EXPR1 EXPR2	与逻辑，若 EXPR1 为假，将不再评估 EXPR2
EXPR1 -a EXPR2	与 EXPR1 EXPR2 功能一致
EXPR1 -o EXPR2	逻辑或，若 EXPR1 为真，将不再评估 EXPR2
EXPR1 , EXPR2	若 EXPR1 为假，继续评估 EXPR2

find命令举例（9）

- 查找 /tmp 目录下21天之前访问过的大于 10G 的文件
✓ `$ find /tmp -size +10M -a -atime +21`
- 查找 /home 目录下属主为 jjheng 或 osmond 的大于 10M 的文件
✓ `$ find /home \(-user jjheng -o -user osmond \) -size +10M`
- 查找 /tmp 目录下的属主不是 osmond 的文件
✓ `$ find /tmp ! -user osmond`
- 在 /mnt 下查找 *.txt 且文件系统类型不为 vfat 的文件
✓ `$ find /mnt -name '*.txt' ! -fstype vfat`
- 在 /tmp 下查找名为 l 开头且类型为符号链接的文件
✓ `$ find /tmp -name 'l*' -type l`
- 找出 /var/log 目录下所有的5天前修改过的.log 文件
✓ `$ find /var/log -name '*.log' -mtime +5`
- 查找所有比 FILE1 的内容修改时间新的且比 FILE2 旧的文件
✓ `$ find -newer FILE1 ! -newer FILE2`

find 命令——动作表达式

- -print
 - ✓在标准输出上列出查找结果（每行一个文件）
 - -ls
 - ✓使用 'ls -dils' 在标准输出上列出查找结果
 - -prune
 - ✓忽略对某个目录的查找
 - -exec COMMAND {} \;
 - -ok COMMAND {} \;
- {} 两个大括号之间不能有空格，表示查找到的对象
 - \; 表示命令结束，反斜杠与前面的大括号之间必须要留空格
- ✓对符合查找条件的文件执行 Linux 命令
 - ✓对符合查找条件的文件执行 Linux 命令；与 -exec 不同的是，它会询问用户是否需要执行

find命令举例（10）

- 查找并列出现当前目录下不安全的文件（世界可读写执行）
✓ `$ find . -perm -007 -ls`
- 查找 logs 目录下的所有的 .log 文件并查看它的详细信息
✓ `$ find logs -name "*.log" -type f -exec ls -l {} \;`
- 查找当天修改过的普通文件
✓ `$ find . -type f -mtime -1 -exec ls -l {} \;`
- 查找当前目录下的.php文件、并用grep过滤出包含include的行
✓ `$ find . -name "*.php" -exec grep "include" {} \; -print`

find命令举例（10）续

- 查找并删除当前目录及其子目录下所有扩展名为.tmp 的文件
✓ `$ find . -name '*.tmp' -exec rm {} \;`
- 在logs目录中查找7天之内未修改过的文件并在删除前询问
✓ `$ find logs -type f -mtime +7 -exec -ok rm {} \;`
- 从当前目录下查找所有以.repo为后缀的文件，并为其改名（添加.bak后缀）
✓ `$ find . -name '*.repo' -type f -exec mv {} {}.bak \;`
- 查询并删除一周以来从未访问过的以 .o 结尾，或名为 a.out 且不存在于 vfat 文件系统中的所有文件
✓ `$ find / (-name a.out -o -name '*.o') -atime +7 ! -fstype vfat -exec rm {} \;`

find命令举例（10）续2

- 显示当前目录下除 book 目录之外的所有文件
✓ `$ find . -name book -prune -o -print`
- 查找当前目录下（除了 book 目录）之外的所有 .sh 文件
✓ `$ find . -name book -prune -o -name '*.sh' -print`
- 显示当前目录下（除了 book/server 目录）之外的所有文件
✓ `$ find . -path ./book/server -prune -o -print`
- 查找当前目录下（除了 book/server 目录）之外的所有 .sh 文件
✓ `$ find . -path ./book/server -prune -o -name '*.sh' -print`
- 显示当前目录下除 book/server 和 book/server-utf8 目录的所有文件
✓ `$ find . -path './book/server*' -prune -o -print`
- 查找当前目录下（除了 book/server 和 book/server-utf8 目录）的所有 .sh 文件
✓ `$ find . -path './book/server*' -prune -o -name '*.sh' -print`
- 查找当前目录下（除了 book/server 和 book/basic 目录）的所有 .sh 文件
✓ `$ find . \(-path ./book/server -o -path ./book/basic \) -prune -o -name '*.sh' -print`

find命令举例 (11)

- 下面 find 命令的书写形式均等价
- `$ find -name *.sh -exec cp {} /tmp \;`
- `$ find -name '*.sh' -exec cp {} /tmp ';`
- `$ find -name "*.sh" -exec cp {} /tmp ";`
- `$ find -name *.sh -exec cp {\} /tmp \;`
- `$ find -name '*.sh' -exec cp '{} ' /tmp ';`
- `$ find -name “*.sh” -exec cp “{ }” /tmp ";`

课程内容



常用的文件打包和压缩命令

命令	功能
xz	使用LZMA 算法的高性能压缩/解压工具
gzip	流行的 GNU gzip 数据压缩/解压程序
bzip2	免费的，无专利的高性能数据压缩工具
zip/unzip	与WinZIP兼容的压缩/解压工具
rar	与WinRAR兼容的压缩/解压工具
7za	使用LZMA 算法的高性能压缩/解压工具
tar	文件打包、归档工具

打包和压缩文件的文件后缀

文件后缀	说明
.bz2	用 bzip2 压缩的文件
.gz	用 gzip 压缩的文件
.xz	用 xz 压缩的文件
.tar	用 tar 打包的文件，也称 tar 文件
.tbz	tar 打包时用 bzip2 压缩的文件
.tgz	tar 打包时用 gzip 压缩的文件
.zip	用 zip/winzip 压缩的文件
.rar	用 rar 压缩的文件
.7z	用 7za 压缩的文件

gzip

- Linux下常用的压缩和解压缩命令。
- 由官方仓库的gzip软件包提供。
- 压缩后 gzip 会在每个文件的后面添加扩展名 .gz。
- 压缩后原文件会被自动删除。
- 在 windows 下可以用 winzip 或 winrar或7-zip 解压。

gzip命令

- 用法：gzip [选项] 文件列表
- 选项：
 - ✓ -d: 解开压缩文件。
 - ✓ -f: 强行压缩文件，不理睬文件名称或硬链接是否存在以及该文件是否为符号链接。
 - ✓ -l: 列出压缩文件的相关信息（压缩文件的大小；未压缩文件的大小；压缩比；未压缩文件的名字）。
 - ✓ -n: 压缩文件时，不保存原来的文件名称及时间戳（默认为保存，即-N）。
 - ✓ -r: 递归处理，将指定目录下的所有文件及子目录一同处理。
 - ✓ -t: 测试压缩文件是否正确无误。
 - ✓ -v: 显示指令执行过程。
 - ✓ -<压缩率>: 压缩率是一个介于1~9的数值，默认值为“6”，数值越大压缩率越高。
 - ✓ --best 参数等价于-9；--fast参数等价于-1。

gzip命令举例

- 压缩文件filename
 - ✓ `$ gzip filename`
- 压缩文件 file1和file2并显示执行过程
 - ✓ `$ gzip -v file1 file2`
- 递归地高度压缩mydir目录下的所有文件（ 逐个文件进行 ）
 - ✓ `$ gzip -9r mydir`
- 显示当前目录下所有压缩过的gz文件信息
 - ✓ `$ gzip -l *.gz`
- 解压filename.gz文件
 - ✓ `$ gzip -d filename.gz`
 - ✓ `$ gunzip filename.gz`

bzip2

- Linux下常用的压缩和解压缩命令。
- 由官方仓库的bzip2软件包提供。
- 比gzip的压缩比更高。
- 压缩后 bzip2 会在每个文件的后面添加扩展名 .bz2。
- 压缩后原文件会被自动删除。
- 在 windows 下可以用 winrar或7-zip 解压。
- bzip2命令的格式和参数与gzip类似。

bzip2命令举例

- 压缩文件filename
 - ✓ \$ bzip2 filename
- 高度压缩文件 file1和file2并显示执行过程
 - ✓ \$ bzip2 -9v file1 file2
- 解压filename.bz2文件
 - ✓ \$ bzip2 -d filename.bz2
 - ✓ \$ bunzip2 filename.bz2

zip/unzip

- 与windows下的 winzip兼容
- 由官方仓库的zip/unzip软件包提供
- 例如：
 - ✓ 压缩文件 file1为 file1.zip , 原文件保留
 - ✓ `$ zip file1.zip file1`
 - ✓ 将子目录 data1/ 下的所有文件压缩到文件 data1.zip
 - ✓ `$ zip -r data1.zip data1`
 - ✓ 解压释放压缩文件 data1.zip 中的所有文件
 - ✓ `$ unzip data1.zip`

rar和7za

- rar
 - ✓ 由RPMForge仓库的rar软件包提供
- 7za
 - ✓ <http://www.7-zip.org>
 - ✓ <http://p7zip.sourceforge.net/>
 - ✓ 由EPEL仓库的p7zip软件包提供

tar

- 基本功能：打包和解包
- 格式：tar [选项] 文件或者目录
- 常用选项
 - ✓ -c：创建新的打包文件。
 - ✓ -t：列出打包文件的内容，查看已经打包了哪些文件。
 - ✓ -x：从打包文件中释放文件。
 - ✓ -f：指定打包文件名。
 - ✓ -v：详细列出 tar 处理的文件信息。
 - ✓ -z：用 gzip 来压缩/解压缩打包文件。
 - ✓ -j：用 bzip2 来压缩/解压缩打包文件。
 - ✓ -J：用 xz 来压缩/解压缩打包文件。

tar命令举例

- `$ tar -cvf myball.tar somedirname`
- `$ tar -tf myball.tar`
- `$ tar -xvf myball.tar`
- `$ tar -zcvf myball.tar.gz somedirname`
- `$ tar -ztf myball.tar.gz`
- `$ tar -zxvf myball.tar.gz`
- `$ tar -jcvf myball.tar.bz2 somedirname`
- `$ tar -jtf myball.tar.bz2`
- `$ tar -jxvf myball.tar.bz2`

注意：“-f 文件名|设备名”是一个整体，所以 -cvf myball.tar 不能写成：-cfv myball.tar 或 -fcv myball.tar

课程总结



本章思考题

- 常用的文件和目录操作命令有哪些？各自的功能是什么？
- 常用的信息显示命令有哪些？各自的功能是什么？
- 打包和压缩有何不同？常用的打包和压缩命令有哪些？
- Vi的3种运行模式是什么？如何切换？

THANK YOU!