Linux 文件、目录与操作

.....Information

.....Data

.....File

.....Directory

.....Device

- 文件类型
- 特殊文件
- 文件系统结构与挂载
- 文件与目录操作
- 磁盘使用情况

文件类型

- 普通文件： 保存数据，并位于某种类型的存储设备上
  - 文本文件：由可显示字符和新行字符构成
  - 二进制文件：包含非文本数据
- 目录(directory, folder)：组织、访问其它文件和目录
- 伪文件 (pseudo file)： 不存储数据，不占用空间
  - 特殊文件，或设备文件：物理设备的内部表示，并以之来进行访问
  - 命名管道：显式的管道文件
  - proc 文件：访问内核信息

特殊文件 (special file)

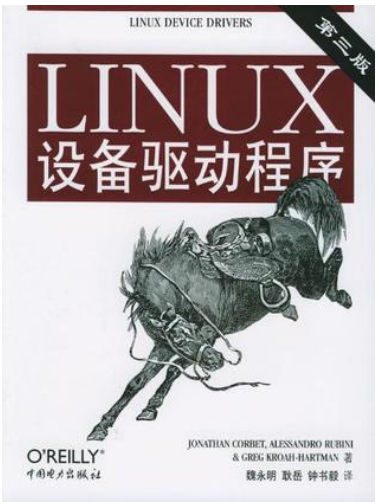
类型	文件名	含义
硬件	/dev/sda	SCSI 硬盘
	/dev/sda1	SCSI 硬盘第1分区
	/dev/usb/lp0	USB 打印机
终端	/dev/tty	当前终端
	/dev/tty1	虚拟控制台
	/dev/pts/0	伪终端
伪设备	/dev/null	放弃输出，输入不返回内容
	/dev/zero	放弃输出，输入返回null
	/dev/random	随机数生成器
	/dev/urandom	随机数生成器

LANANA Device List

Linux Assigned Names and Numbers Authority

硬件特殊文件

- 所有连接到计算机上的设备都通过特殊文件访问
- 设备名一般为 缩写 + 数字 的形式
- **IDE** 硬盘是 **/dev/hda**
  - 第二块是 **/dev/hdb**
  - 第一块的第一个分区则为**/dev/hda1**
- **SCSI,SATA,USB** 接口的都在**sda**



终端特殊文件

- 硬件终端（包括虚拟终端），均是 **/dev/ttyN**
- 使用GUI的终端仿真，或者是远程连接时，则创建所谓的伪终端，其命名方式为以下两种之一：
  - **/dev/ttypN**
  - **/dev/pts/N**
- **$ cp hello_file /dev/pts/1**

命名管道

- **$ mkfifo [-m mode] *pipe***
- 创建一个显式的管道文件
  - **$ mkfifo fifotest**
  - **$ grep bash /etc/passwd > fifotest**
  - **$ wc -l < fifotest**
- **mode 为 Unix中的文件模式(chmod)**

标准Linux 文件系统

FHS: Filesystem Hierarchy Standard

目录名	内容
/	根目录
/bin	基本程序
/boot	启动系统时需要的文件
/dev	设备文件
/etc	配置文件
/home	普通用户的home目录
/lib	基本共享库，内核模块

标准Linux 文件系统

目录名	内容
/lost+found	由fsck恢复的受损文件
/media	可移动介质的挂载点
/mnt	不能挂载在其他位置上的固定介质挂载点
/opt	第三方应用程序（可选软件）
/proc	proc文件 （进程文件）
/root	超级用户的home目录
/sbin	由超级用户运行的基本系统管理程序

标准Linux 文件系统

目录名	内容
/srv	本地系统所提供服务的数据
/tmp	临时文件
/usr	静态数据使用的辅助文件系统
/var	可变数据使用的辅助文件系统

标准Linux 文件系统

目录名	内容
/usr/bin	非基本程序（多数用户程序）
/usr/include	C程序头文件
/usr/lib	非基本共享库
/usr/local	本地安装程序
/usr/sbin	由超级用户运行的非基本系统管理程序
/usr/share	共享系统数据
/usr/src	源代码

标准Linux 文件系统

- `# ls -R / | wc -l`
  - 查询当前 os 下一共有多少个文件
- `$ ls /bin | wc -l`
- `$ ls /usr/local/bin/ | wc -l`

文件系统的挂载

- `$ df -h #` 磁盘设备被挂载到了不同的目录位置
- `$ mount [-t type] device dir`
- `$ umount`
- `/etc/fstab` ： 枚举了所有需要自动挂载的文件系统设备
  - `$ ls -l /dev/disk/by-uuid`
  - 可以查看不同磁盘设备的 `uuid`

文件系统的分类

- Windows: FAT32, NTFS , exFAT
- Unix: ext3, ext4
- Mac OSX: HFS+ , APFS
- CD: ISO 9660, UDF
- BSD, Solaris: UFS32
- Network: NFS, SMB
- proc: procfs
- pty: devpts

不同设备、位置、文件系统间在数据存储结构和读写方式上存在较大的差别，如何能够让它们无缝交流？

VFS

- **VFS：** 虚拟文件系统 **virtual file system**
- 它是一组**API**，充当各类文件系统间的中间人
- 任何**I/O** 操作都是向**VFS**发送请求，来定位合适的文件系统、设备驱动程序来执行
- 作为用户，无需对文件系统间的差异有任何了解
- 所有设备上的所有文件，都统一管理在**FHS**结构下
  - **Filesystem Hierarchy Standard** 文件系统层次化标准

目录操作

目录操作

- `$ /bin/ls`
- `$ cd /bin ; ./ls`
  - 此处的 `./` 并不是执行，而仅仅是定位当前目录下的文件

目录创建

- `$ mkdir [-p] directory...`
- 一次可以创建多个目录
- `-p`：一次创建多级目录

目录删除

- **$ rmdir [-p] *directory*...**
- 若目录内非空，默认不可删除
- **-p**：连续删除多级空目录
- 工作目录（**PWD**）和根目录之间的目录，不可删除

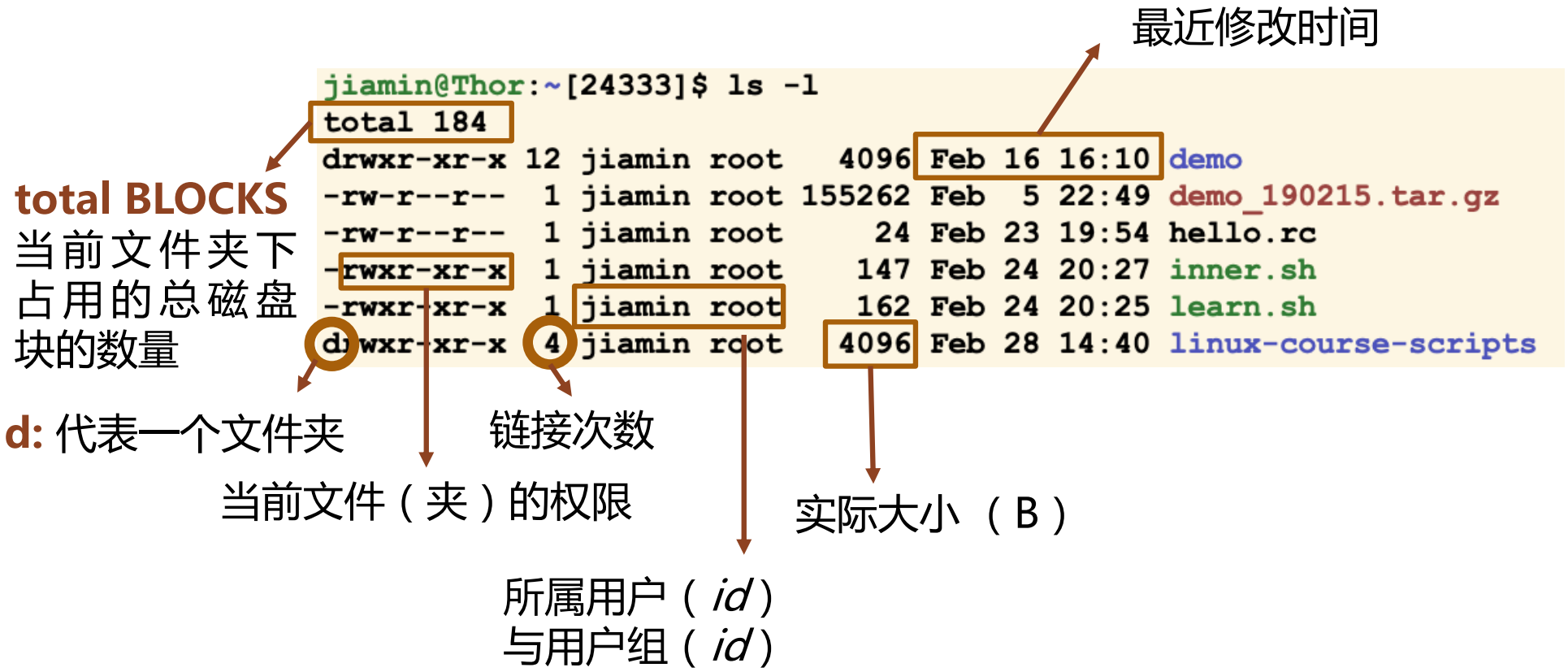
目录移动 / 重命名

- `$ mv directory target`
- 移动目录时，会同时移动目录中的所有文件和子目录

显示目录内容

- **$ ls [-aCdFglrRs1] [*name*...]**
- **-l : (long listing)** 显示目录内容详情
- **-R : (recursive)** 显示整个目录子树
- **-F : (flag)** 显示文件标识
  - * 可执行文件； / 目录；@ 符号链接 ； | 命名管道

显示目录内容


最近修改时间
jiamin@Thor:~[24333]$ ls -l
total 184
drwxr-xr-x 12 jiamin root 4096 Feb 16 16:10 demo
-rw-r--r-- 1 jiamin root 155262 Feb 5 22:49 demo_190215.tar.gz
-rw-r--r-- 1 jiamin root 24 Feb 23 19:54 hello.rc
-rwxr-xr-x 1 jiamin root 147 Feb 24 20:27 inner.sh
-rwxr-xr-x 1 jiamin root 162 Feb 24 20:25 learn.sh
drwxr-xr-x 4 jiamin root 4096 Feb 28 14:40 linux-course-scripts
total BLOCKS
当前文件夹下占用的总磁盘块的数量
d: 代表一个文件夹
链接次数
当前文件（夹）的权限
实际大小（B）
所属用户（id）
与用户组（id）

显示目录内容

- 默认情况下，`ls` 依据文件名排序
  - 排序序列取决于`$LC_COLLATE`设置
- `-t` ： 按修改时间从最新到最旧进行排序
- `-r` ： `(reverse)` 倒序

显示目录内容

- 通配符：不仅是 * 和 ？
- **$ cd demo/ch03/ls**
  - **[list] ： $ ls hello.[co]**
  - **[^list]： $ ls hello.[^mn]**
  - **[[:lower:]] : $ ls hello.[[:upper:]]**
  - 先由**shell**进行解释匹配，再执行命令

检查磁盘使用情况

- **$ ls -s (size)**：单位是**KB**，占用的磁盘块大小
- **$ ls -sh (human-readable)**：选择合适单位
- **$ du -sh**
- **$ df -h**

文件大小 vs. 文件的磁盘空间

- `$ cd ~/demo/ch03/size`
- `$ echo "hello" > normal`
- `$ ls -l normal` #文件有6B大小，但只有5个字符
- `$ cat -A normal` #查看不可见字符
- `$ od -c normal` #查看字符的ASCII码内容
- `$ ls -sh normal` #查看文件占用的存储空间大小
- `$ du -sh normal` #查看文件占用的存储空间大小
- `$ sudo dumpe2fs /dev/vda1 \`
  `| grep "Block size"` #查看系统设置的磁盘块大小

展示目录树

- `$ tree [-dfil] [-L level] [directory ...]`
- `-d` ： 只显示目录
- `-f` ： 显示完整的路径名
- `-i` ： 省略输出中的的缩进结构
- `-l` ： 显示符号链接的内容
- `-L` ： 限制目录树的查看深度 *`level`*

文件操作

创建新文件

- `$ vi file`
- `$ command > file`
- `$ cp source file`
- `$ touch file`

在 Linux 下没有专门的命令，来让你创建一个新的普通文件，这只是touch命令的副作用。

touch

- **`$ touch [-acm] [-t time] file...`**
- 用于同时更改文件的修改时间与访问时间
- **`-a: (access)`，** 只更改访问时间
- **`-m: (modify)`，** 只更改内容修改的时间
- **`-c: (no create)`**，只更改状态修改的时间
  - 默认若文件不存在，则创建

文件复制

- **`$ cp [-ip] file1 file2`**
- 复制单个文件
- 若**`file2`**存在，则它会被覆盖，**无法恢复！**
- **`-i : (interactive)`** 以交互形式来运行命令
  - 若文件存在，则会询会是否要覆盖它
- **`-p : (preserve)`** 使得目标文件**`file2`**的修改时间、访问时间，权限等与源文件 **`file1`**完全相同

文件复制

- `$ cp [-ip] file... directory`
- 将多个文件复制到目标文件夹内
- 可以采用通配符的方式来选择源文件

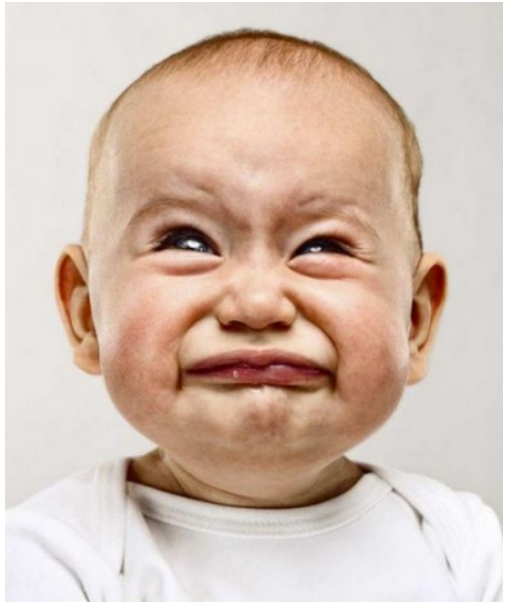
文件复制

- `$ cp -r [-ip]` ***`directory1... directory2`***
- 将多个文件夹复制到目标文件夹内

文件删除

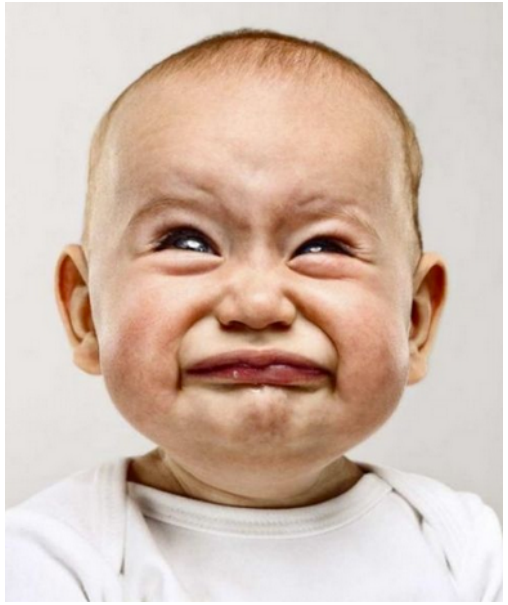
- `$ rm [-fir] file...`
- 删除文件**不可恢复**！
- `-i: (interactive)` 以交互方式运行
- `-f: (force)`
  - 默认情况下，如果没有写权限，用户则无法删除
  - 加上`force`选项后，**可以删除`root`的文件，没有权限的文件**
- `-r: (recursive)` 递归删除整个子目录树

已删除文件能否恢复？



没可能

已删除文件能否恢复？



你真没可能

已删除文件能否恢复？

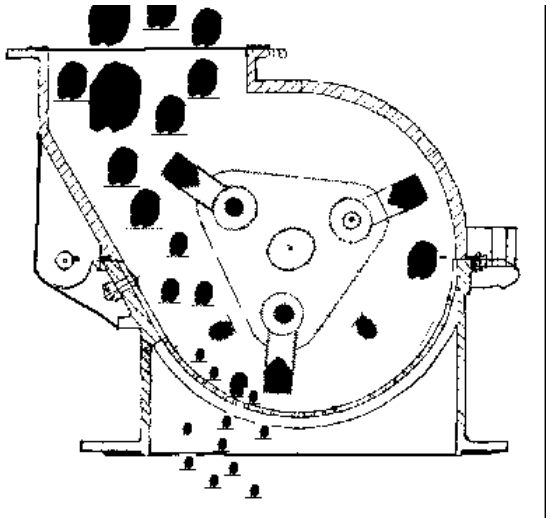
不相信？

请执行：

```
$ rm -rf /
```

如果想彻底清除文件内容？

- 那么你可以粉碎（**shred**）它
- **$ shred [-fvzu]** ***[file ...]***
- **-f:** 强制执行
- **-v:** 输出处理过程
- **-z:** 将文件全部填充为0
- **-u:** 清空文件内容后再将它删除



文件权限

- 普通文件
  - 读：读取文件
  - 写：写入文件
  - 执行：执行文件
- 目录
  - 读：读取目录
  - 写：创建、移动、复制或删除目录条目
  - 执行：搜索目录

文件权限

- `$ ls -l /usr/bin/passwd`

```
-rwsr-xr-x 1 root root /usr/bin/passwd
```

- 特殊权限 **setUID**
- 任意用户都可以 以**passwd**的拥有者**root**来执行

显示用户所属组别

- `$ id`
  - 显示当前登录用户的情况
- `$ groups [`*`userid`*`...]`
  - 查询用户所属组的情况

改变文件权限

- `$ chmod mode file...`
- `400` 代表 ？
- `755` 代表 ？
- `555` 代表 ？

默认文件模式与掩码

- Unix根据文件类型为文件设定初始模式
  - 普通不可执行文件：666
  - 普通可执行文件：777
  - 目录：777
- 在此基础上，减去用户掩码(user mask)，设置该用户的默认模式
  - $ umask 022

默认文件模式与掩码

- `$ cd ~/demo/ch03/mask`
- `$ umask 022`
- `$ touch mfile`
- `$ mkdir mfolder`
- `$ ls -l`

检查文件类型

- `$ file [name...]`
- `$ cd ~/demo/ch04/compile`
- `$ file demo`
- `demo: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=1e3d7231c7b12f35c954defaf06acc5b17ccaa62, not stripped`

Unix 文件与链接

- **Unix** 下用 **i-node (index node)** 来存储文件的基本信息
- 所有的**i-node**存储在系统表中
- 在目录文件中，建立文件名和**i-node**的映射关系
- **$ ls -i #** 显示文件的**i-node**节点号
- **$ stat *filename* #** 显示文件的**i-node** 信息

Unix 文件与链接

- 硬链接：普通链接，直接映射文件名与**i-node**。
  - 只要文件还存有链接，就不会被删除
  - 不能创建目录的硬链接
  - 不能为不同文件系统中的文件创建链接
- 软链接：符号链接**(symbol link)**，为一个包含文件路径的普通文件。
  - 删除原文件，会导致链接的失效

Unix 文件与链接

- `$ cd ~/demo/ch03/link`
- `$ echo "hello" > file`
- `$ ln file hardlink`
- `$ ln -s file softlink`
- `$ stat file hardlink softlink`

Unix 文件与链接

- `$ rm file`
- `$ cat softlink`
- `$ echo "hello2" > file`
- `$ cat softlink`
- `$ cat hardlink`
- `$ stat hardlink`
- `$ stat file`

Unix 目录与链接

- `$ mkdir folder; ln -s folder  softfolder`
- `$ cd folder; pwd`
- `$ cd ../softfolder; pwd`
- cd 选项： (同样适用于pwd命令)
  - -L (logical)：将符号链接视为真实的目录，默认选项
  - -P (physical)：用真实目录替换符号链接
- `$ cd -P softfolder; pwd`

whereis ; locate; find

- **$ whereis [-bms] *commands...***
- **whereis** 用于查找与特定**Unix**命令相关的文件
  - 包括二进制（可执行）文件，源文件和文档文件
  - 不搜索整个文件系统，而只是查看此类文件可能存在的目录
- **$ whereis ls**
- **$ whereis man**

可以用**whereis** 来查询某命令的说明书页

whereis ; locate; find

- `$ whereis [-bms] commands...`
- `-b` ： 只查询二进制执行文件
- `-m` ： 只查询说明书页
- `-s` ： 只查询源代码

whereis ; locate; find

- **`$ locate [-bcirS] pattern...`**
  - 简单、易用，能查询所有可公共访问的路径名称
  - 默认情况查询 **`whole path`**，即文件的完整路径
  - `$ locate dict | grep words`
- 通过将所有可公共访问的文件路径加入到一个特殊数据库中来实现
  - 数据库采用定时更新机制，**存在时延**，无法找到新建文件

whereis ; locate; find

- **`$ locate [-bcirS] pattern...`**
  - **`-b`**: 只查询文件名
  - **`-r`** : 可以使用正则表达式
  - **`-c`** : 可以进行统计
  - **`-i`** : 忽略大小写
  - **`-S`**: 显示该数据库的统计信息

whereis ; locate; find

- **find** 是最古老，最复杂，最强大的文件搜索工具
- **$ find *path... test... action...***
  - **path** : 搜索path对应的(若干)目录，及其子目录
    - **GNU find: .** (当前目录)是path的默认值
  - **test** : 对遇到的每个文件进行测试的条件
  - **action** : 对满足测试条件的文件执行指定操作

whereis ; locate; find

- **`$ find /home/jiamin -name hello -print`**
  - **`path`** : 搜索 **`/home/jiamin`** 目录，及其子目录
  - **`test`** : 测试文件名是否 **`==`** **`hello`**
  - **`action`** : 对符合测试的文件，打印其路径

whereis ; locate; find

- **$ find *path... test... action...***

类型	选项	含义
文件名	-name *pattern*	包含pattern的文件名
	-iname *pattern*	包含pattern，不区分大小写
文件特征	-type *[df]*	类型 d=目录，f=普通文件...
	-perm *mode*	设置为mode权限的文件
	-user *userid*	属主为 userid
	-group *groupid*	组为 groupid
	-size *[-+]n[cbkMG]*	大小为 n[字符，块，K...]
	-empty	空文件

whereis ; locate; find

- **-name / -iname ：** 文件名称匹配指定的模式
  - 一般指文件名相同
  - 可以使用通配符来指定模式，但**需要将模式引用起来**
    - `$ find hello`
    - `$ find -name "hello*"`
      - 这样参数被传递给**find**命令时，不会先被**shell**解释

whereis ; locate; find

- `-type` ： 匹配文件类型
  - `f`: 普通文件
  - `d`: 目录
  - `b`: 块设备
  - `c`: 字符设备
  - `p`: 命名管道
  - `l`: 符号链接

whereis ; locate; find

- `-size` ： 文件大小
  - `+ / -` ： 大于 / 小于指定大小
  - `b/k/M/G` ： 查看的是文件使用的磁盘空间
    - `$ find ./ -size 1b`
  - `c` ： 字节单位，文件的实际大小

whereis ; locate; find

- 筛选掉部分结果，即对条件取反
  - `$ find -maxdepth 1 -size 1b ! -name ".*"`
  - `$ find -maxdepth 1 -size 1b '!' -name ".*"`
  - `$ find -maxdepth 1 -size 1b \! -name ".*"`

whereis ; locate; find

类型	选项	含义
访问时间 修改时间	-amin *[-+]n*	n分钟之前访问
	-anewer *file*	file文件之后访问
	-atime *[-+]n*	n天之前访问
	-cmin *[-+]n*	n分钟之前状态改变
	-cnewer *file*	file文件之后改变
	-ctime *[-+]n*	n天之前状态改变
	-mmin *[-+]n*	n分钟之前修改
	-newer *file*	file文件之后修改
	-mtime *[-+]n*	n天之前修改

whereis ; locate; find

- **a* : access,** 访问的文件
- **m* : modify,** 文件内容被修改
  - **vi, echo >**
- **c* : change,** 文件**本身**被修改
  - **chmod , chgrp,mv**

whereis ; locate; find

- 时间区间的表示
  - `....+ N -....`
  - **+N** 表示向左，从 前**N**天（分钟） 再往前推
  - **N** 表示时间点，前第**N**天（分钟）
  - **-N** 表示向右， 前**N**天（分钟）以内

whereis ; locate; find

- 时间区间的表示
  - `-mmin -30` # 30分钟内修改的文件
  - `-atime +180` # 180天前都未被访问过的文件
  - `-cmin -10` # 10分钟内被改过状的文件

whereis ; locate; find

- **$ find *path... test... action...***

动作	含义
-print	将路径名写入到**STDOUT** （默认）
-fprint *file*	将输出写入到文件中
-ls	显示长目录列表
-fls *file*	将列表写入到文件中
-delete	删除文件
-exec *command {}\;*	执行**command**，{ }指示匹配的文件名
-ok *command {}\;*	同**exec**，但在执行前进行确认

whereis ; locate; find

- **$ cd ~/demo/ch03/find**
- **$ find -type f -exec echo {} \;**
  - {} ： 用于指代每个被找到的内容，此处为目录
  - \; ： 用于表示命令末尾，

    `;’ 必须被转义/引用，以防被**shell**解释

whereis ; locate; find

- 对结果文件内容进行检索
- `$ cd ~/demo/ch03/find`
- `$ find -type f -exec grep -H hello {} \;`
- `$ grep hello `find -type f `
- `$ find ./ -type f  | xargs grep hello`

xargs 可以避免第二种方法中参数列表过长的麻烦

xargs

- `$ cd ~/demo/ch03/find`
- `$ find ./ -type f | grep hello`
- `$ find ./ -type f | xargs grep hello`

区别！！对文件内容执行 grep 操作，而非 find 输出的字符串

GEAR UP