

Linux的文件系统

1.1. 万事万物皆文件

- 文件系统：
 - 操作系统如何管理文件，内部定义了一些规则或者定义
- 所以在Linux中所有的东西都是以文件的方式进行操作
- 在Linux中，文件的访问不和Window的一样。window依靠的是通过盘符进行访问

/bin:

bin是Binary的缩写，这个目录存放着最经常使用的命令。

/boot:

这里存放的是启动Linux时使用的一些核心文件，包括一些连接文件以及镜像文件。

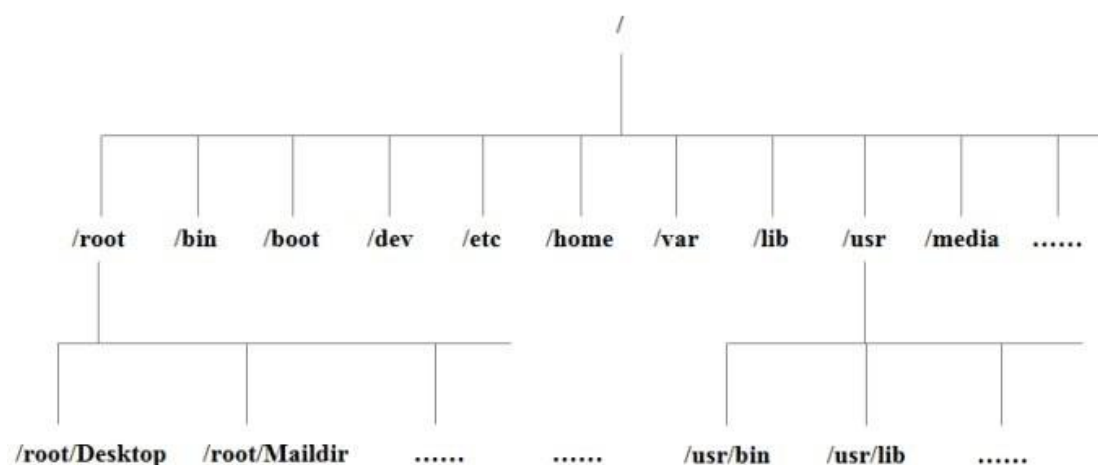
/dev :

dev是Device(设备)的缩写，该目录下存放的是Linux的外部设备，在Linux中访问设备的方式和访问文件的方式是相同的。

◆

- Linux维护着一个树状结构的文件模型
 - 只有一个根节点 ,他的名字叫做 /
 - 一个节点上可以有多个子节点
- 查找文件的方式
 - 相对路径
 - 以当前路径为基准点, 查找其他资源
 - vi ../etc/sysconfig/network
 - 绝对路径
 - 以根目录为基准点, 查找其他资源
 - vi /etc/sysconfig/network-scripts/ifcfg-ens33
 - 日常使用中, 只要找到路径即可, 但是如果是一些配置文件, 尽量写绝对路径
- 可以随意的挂载磁盘
 - ```
mount /dev/disk1 /usr/download
disk1 1T
mount /dev/disk2 /usr/upload
disk2 100T
mount /dev/disk3 /usr/upload/photo
disk3 1P
```

## 1.2. Linux二级文件目录



#### **/etc:**

这个目录用来存放所有的系统管理所需要的配置文件和子目录。

#### **/home:**

用户的主目录，在Linux中，每个用户都有一个自己的目录，一般该目录名是以用户的账号命名的。

#### **/lib:**

这个目录里存放着系统最基本的动态连接共享库，其作用类似于windows里的DLL文件。几乎所有的应用程序都需要用到这些共享库。

#### **/lost+found:**

这个目录一般情况下是空的，当系统非法关机后，这里就存放了一些文件。

#### **/media:**

linux 系统会自动识别一些设备，例如U盘、光驱等等，当识别后，linux会把识别的设备挂载到这个目录下。

#### **/mnt:**

系统提供该目录是为了让用户临时挂载别的文件系统的，我们可以将光驱挂载在/mnt/上，然后进入该目录就可以查看光驱里的内容了。

#### **/opt:**

这是给主机额外安装软件所摆放的目录。比如你安装一个ORACLE数据库则就可以放到这个目录下。默认是空的。

#### **/proc:**

这个目录是一个虚拟的目录，它是系统内存的映射，我们可以通过直接访问这个目录来获取系统信息。

这个目录的内容不在硬盘上而是在内存里，我们也可以直接修改里面的某些文件，

比如可以通过下面的命令来屏蔽主机的ping命令，使别人无法ping你的机器：

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

#### **/root:**

该目录为系统管理员，也称作超级权限者的用户主目录。

#### **/sbin:**

s就是Super User的意思，这里存放的是系统管理员使用的系统管理程序。

#### **/selinux:**

这个目录是Redhat/CentOS所特有的目录，selinux是一个安全机制，类似于windows的防火墙，但是这套机制比较复杂，这个目录就是存放selinux相关的文件的。

#### **/srv:**

该目录存放一些服务启动之后需要提取的数据。

#### **/sys:**

这是linux2.6内核的一个很大的变化。该目录下安装了2.6内核中新出现的一个文件系统 sysfs 。

sysfs文件系统集成了下面3种文件系统的信息：针对进程信息的proc文件系统、针对设备的devfs文件系统以及针对伪终端的devpts文件系统。该文件系统是内核设备树的一个直观反映。

当一个内核对象被创建的时候，对应的文件和目录也在内核对象子系统中被创建。

#### **/tmp:**

这个目录是用来存放一些临时文件的。

#### **/usr:**

这是一个非常重要的目录，用户的很多应用程序和文件都放在这个目录下，类似于windows下的program files目录。

#### **/usr/bin:**

系统用户使用的应用程序。

**/usr/sbin:**

超级用户使用的比较高级的管理程序和系统守护程序。

**/usr/src:**

内核源代码默认的放置目录。

**/var:**

这个目录中存放着在不断扩充着的东西，我们习惯将那些经常被修改的目录放在这个目录下。包括各种日志文件。

**/run:**

是一个临时文件系统，存储系统启动以来的信息。当系统重启时，这个目录下的文件应该被删掉或清除。如果你的系统上有 `/var/run` 目录，应该让它指向 `run`。

## 1.3. Linux的文件操作

- cd
  - 改变当前工作目录
- ls ll
  - 显示出指定目录下所有的文件
  - 文件的类型
    - -普通文  
件 ■ d文件  
夹
    - l软连接
  - rw-r--r--. 1 root root 3384 Nov 11 23:51 install.log.syslog
- mkdir
  - 创建文件目录
  - `mkdir -p a/b/c/d/e/f` 会自动创建文件父目录
  - `mkdir -p lucky/{1234}ls` 一次可以创建多个子目录
- rmdir
  - 删除空文件夹
    - `rmdir: failed to remove 'a1': Directory not empty`
    - `rmdir: failed to remove 'baidu': Not a directory`
  - 可以安全的删除文件目录
- cp
  - 拷贝文件或者文件目录
  - `cp 源文件 目标目录`
    - `cp abcd /opt`
    - `cp /opt/abcd ./`
  - 拷贝文件夹
    - `cp -r lucky /opt`
    - 拷贝文件夹下所有的内容
    - `cp: omitting directory '/root/a1'`
- mv

- 移动文件或者文件夹
  - `mv a1 /opt`

- mv abc /opt
- 修改文件名称
  - mv a abcd
- ◆ rm
  - 删除文件
    - rm install.log
    - rm -f install.log
  - 删除文件夹
    - rm -r abcd
    - rm -rf abcd 谨慎使用，从删库到跑路
- ◆ touch
  - 如果没有就创建一个文件
  - 如果该文件已经存在，修改文件的三个时间，将三个时间改为当前时间
- ◆ stat
  - 查看文件的状态
  - Inode 当前文件在文件系统的唯一标识，类似于ID
  - 时间
    - access 访问时间
    - modify 修改文件内容时间
    - change 修改文件元数据信息时间
      - 文件大小，文件所有者，文件权限
      - 对于文件的描述信息
- ◆ ln
  - 创建文件的链接
  - 接 ○ 软(符号)链接
    - ln -s lucky01 sl
    - 软连接和原始文件不是同一个文件
      - lucky1 131085
      - sl 131074
    - rm -rf lucky1
  - 硬链接
    - ln lucky02 hl
    - 硬链接和原始文件使用文件系统中的同一个文件
    - 如果你害怕一个文件被别人误删，你可以使用硬链接保护这个文件
  - 软硬链接在链接文件的时候，推荐使用文件的绝对路径,否则有可能会出现问题
  -

```
[root@basenode ~]# stat file_sl
 File: 'file_sl'
 Size: 9 Blocks: 8 IO Block: 4096 regular file
Device: fd00h/64768d Inode: 134317919 Links: 1
Access: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Access: 2020-06-16 19:41:45.505579124 +0800
Modify: 2020-06-16 19:40:36.495575567 +0800
Change: 2020-06-16 19:40:36.496575567 +0800
 Birth: -
[root@basenode ~]# stat sl
 File: 'sl' -> 'file_sl'
 Size: 7 Blocks: 0 IO Block: 4096 symbolic link
Device: fd00h/64768d Inode: 134317902 Links: 1
Access: (0777/lrwxrwxrwx) Uid: (0/ root) Gid: (0/ root)
Access: 2020-06-16 19:41:12.731577435 +0800
Modify: 2020-06-16 19:41:11.649577379 +0800
Change: 2020-06-16 19:41:11.649577379 +0800
 Birth: -
```

- [root@basenode ~]# stat file\_hl
 

```
 File: 'file_hl'
 Size: 14 Blocks: 8 IO Block: 4096 regular file
Device: fd00h/64768d Inode: 134317917 Links: 2
Access: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Access: 2020-06-16 19:42:47.087582299 +0800
Modify: 2020-06-16 19:42:44.527582167 +0800
Change: 2020-06-16 19:42:44.528582167 +0800
 Birth: -
[root@basenode ~]# stat hl
 File: 'hl'
 Size: 14 Blocks: 8 IO Block: 4096 regular file
Device: fd00h/64768d Inode: 134317917 Links: 2
Access: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Access: 2020-06-16 19:42:47.087582299 +0800
Modify: 2020-06-16 19:42:44.527582167 +0800
Change: 2020-06-16 19:42:44.528582167 +0800
 Birth: -
```

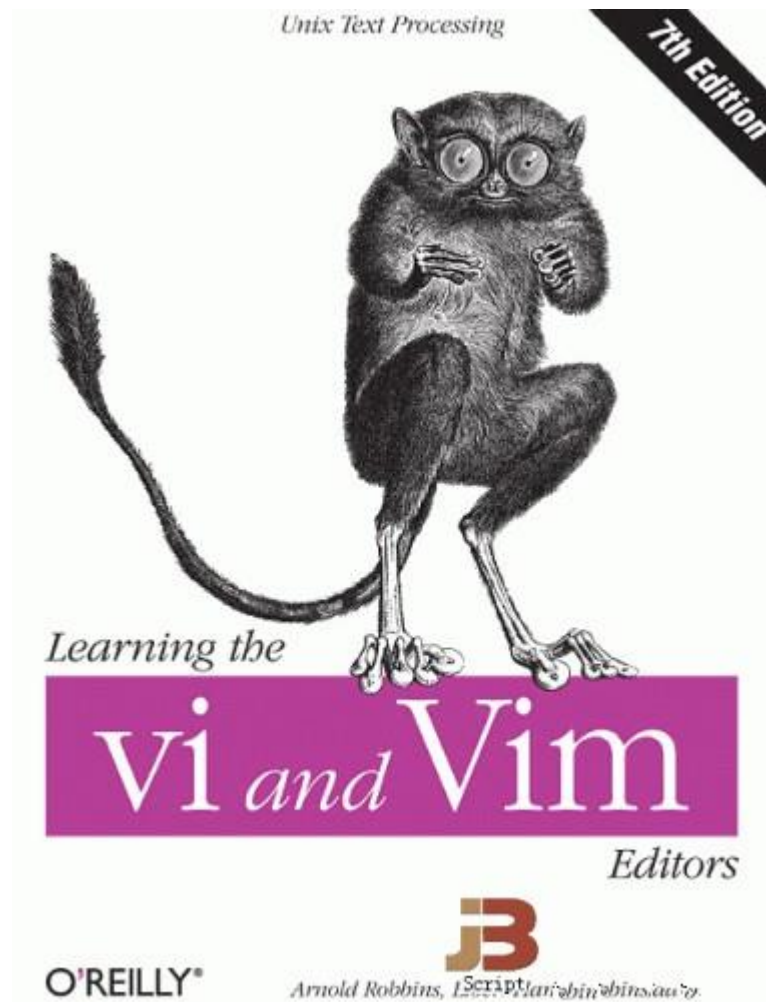
## 1.4. 读取文件信息

- cat
  - 将整个文档加载到内存中，并进行一次性显示除
  - 非后面使用管道，传递数据
- tac
  - 将整个文档加载到内存中，并进行一次性按行逆序显示
- more less
  - 分页查看文档内容
  - 快捷键
    - 回车 下一行
    - 空格 下一页
    - b 回退
    - q 退出
- head
  - 从文章开始读取N行
  - 默认如果超过10行读取10行,否则读取现在行数
  - head -5 profile
- tail
  - 从文章末尾读取N行
  - head -3 profile | tail -1
    - 利用管道只读取第N行

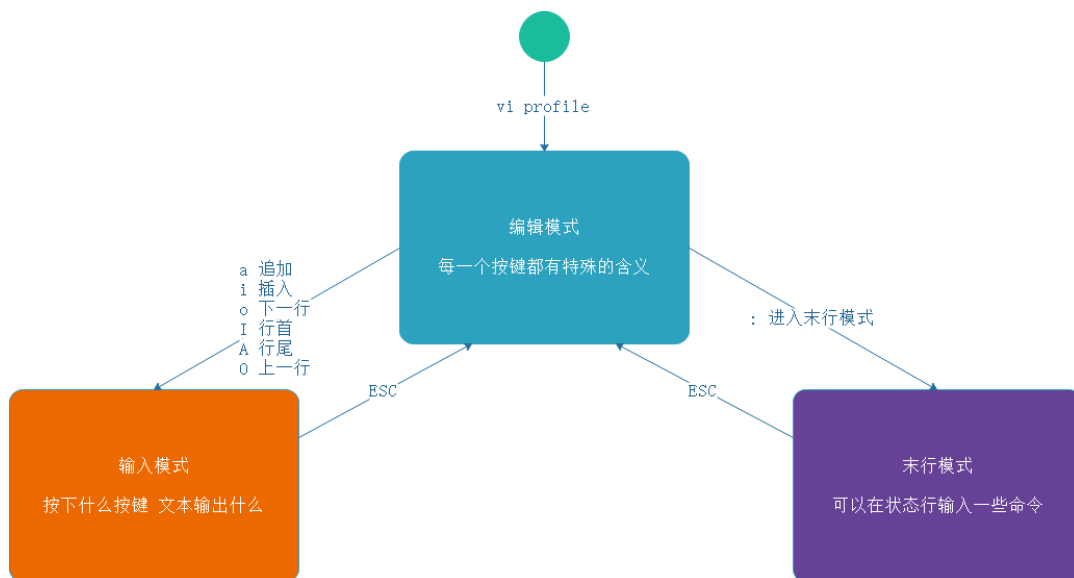


- 管道的作用就相当于把前面的结果以参数的方式传递给后面的命令
- 读取新增数据
  - ping [www.baidu.com](http://www.baidu.com)
  - >>baidu ■ tail -F baidu
  - 如果f:
    - 它会监听指定inode的文件数据变化，但是当文件被删除后
    - 即使创新创建，inode也会发生变化，于是监听失败
  - 如果F
    - 他会监听指定名字的文件,如果文件被删除后，重新创建
    - 他会重新监听新文件的数据变化，监听不受影响
- ♦ find
  - 查找指定的文件
  - find 要查找的范围 -name 名字
  - find /etc -name profile

## 1.5. VI和VIM编辑器







- 编辑模式-->输入模式
  - `i`在当前位置插入数据
  - `a`追加数据
  - `o`在当前行后面开启一个新的输入行
  - `I` 行首
  - `A` 行尾
  - `O` 上一行
- 输入模式-->编辑模式
  - 按下`ESC`
- 编辑模式-->末行模式
  - `:`
- 末行模式-->编辑模式
  - 按下`ESC`

#### 1.5.4. 编辑模式

- `G`最后一行
- `gg` 跳转到第一行
- 数字`gg` 跳转到第数字行
- `w` 下个单词
- 数字`w`
- `dw` 删除一个单词
- `3dw` 删除三个单词
- `dd` 删除一行
- `3dd` 删除三行
- `u`回退到前面的操作
- `.`回退`u`执行的操作
- `yw` 复制一个单词
- `3yw` 复制三个单词
- `yy` 复制一行
- `3yy`复制三行
- `p`粘贴
- `6p` 粘贴6次
- `x` 剪切
- `3x` 剪切三个字符

- r 替换，然后输入一个字符替换
- 3r 替换三个
- hjkl 方向键
- ZZ 保存并退出
- ctrl+s 锁屏 ctrl+q 解锁

### 1.5.5. 输入模式

- 没得讲。按啥是啥

### 1.5.6. 末行模式

- set nu 设置行号
- set nonu 取消行号
- w 保存
- q 退出
- wq 保存并退出
- q!强制退出，但是不保存
- 如果上次异常退出会保留同名隐藏文件，每次启动会给与提示
  - 如果确定当前文件没问题，请删除隐藏文件
- /pattern
  - 搜索指定的字符串
  - /usr n向下查找 N逆向查找
- s/p1/p2/g
  - 替换字符串
  - g 替换当前行所有 否则只替换当前行第一个
    - s/abc/lucky/g
  - 查找指定行
    - 3,8s/abc/lucky/g
  - 替换全文
    - g/abc/s//lucky/g

[illegible]

- 分区信息
  - `df -h`
- 指定文件目录大小
  - `du -h --max-depth=1 apache-tomcat-7.0.61`

- 一个特殊分区，以硬盘代替内存

- 当内存使用满的时候，可以将一部分数据写出到swap分区

## 1.8. 文件压缩

### 1.8.1. tar

- 主要针对的文件是 lucky.tar.gz
- 解压缩
  - tar -zx(解压)v(过程)f(文件) lucky.tar.gz
- ♦ 压缩
  - tar -zc(压缩)f(文件) tomcat.tar.gz(压缩后的名字) apache-tomcat-7.0.61(源文件)
  - tar -zxf tomcat.tar.gz -C /opt/
    - -C 指定解压缩的文件目录

### 1.8.2. zip和unzip

- 安装
  - yum install zip unzip -y
- ♦ 压缩
  - zip -r tomcat.zip apache-tomcat-7.0.61
- 解压缩
  - unzip tomcat.zip