**一、实验介绍**

**1.1 实验内容**

1. Linux 的文件组织目录结构。
2. 相对路径和绝对路径。
3. 对文件的移动、复制、重命名、编辑等操作。

**1.2 实验知识点**

* 每个目录的大体内容
* 文件的属性
* touch,file,rm,mv 等基本命令

**二、Linux 目录结构**

在讲 Linux 目录结构之前，你首先要清楚一点，那就是 Linux 的目录与 Windows 的目录的区别，或许对于一般操作上的感受来说没有多大不同，但从它们的实现机制来说是完全不同的。

一种不同是体现在目录与存储介质（磁盘，内存，DVD 等）的关系上，以往的 **Windows 一直是以存储介质为主的，主要以盘符（C 盘，D 盘...）及分区来实现文件管理，然后之下才是目录，**目录就显得不是那么重要，除系统文件之外的用户文件放在任何地方任何目录也是没有多大关系。所以通常 Windows 在使用一段时间后，磁盘上面的文件目录会显得杂乱无章（少数善于整理的用户除外吧）。然而 UNIX/Linux 恰好相反，**UNIX 是以目录为主的，Linux 也继承了这一优良特性。 Linux 是以树形目录结构的形式来构建整个系统的，可以理解为树形目录是一个用户可操作系统的骨架。**虽然本质上无论是目录结构还是操作系统内核都是存储在磁盘上的，但**从逻辑上来说 Linux 的磁盘是“挂在”（挂载在）目录上的，每一个目录不仅能使用本地磁盘分区的文件系统，也可以使用网络上的文件系统。**举例来说，可以利用网络文件系统（Network File System，NFS）服务器载入某特定目录等。

1. FHS 标准

Linux 的目录结构说复杂很复杂，说简单也很简单。复杂在于，因为系统的正常运行是以目录结构为基础的，对于初学者来说里面大部分目录都不知道其作用，重要与否，特别对于那些曾经的重度 Windows 用户，他们会纠结很长时间，关于我安装的软件在哪里这类问题。说它简单是因为，其中大部分目录结构是规定好了的（FHS 标准），是死的，当你掌握后，你在里面的一切操作都会变得井然有序。

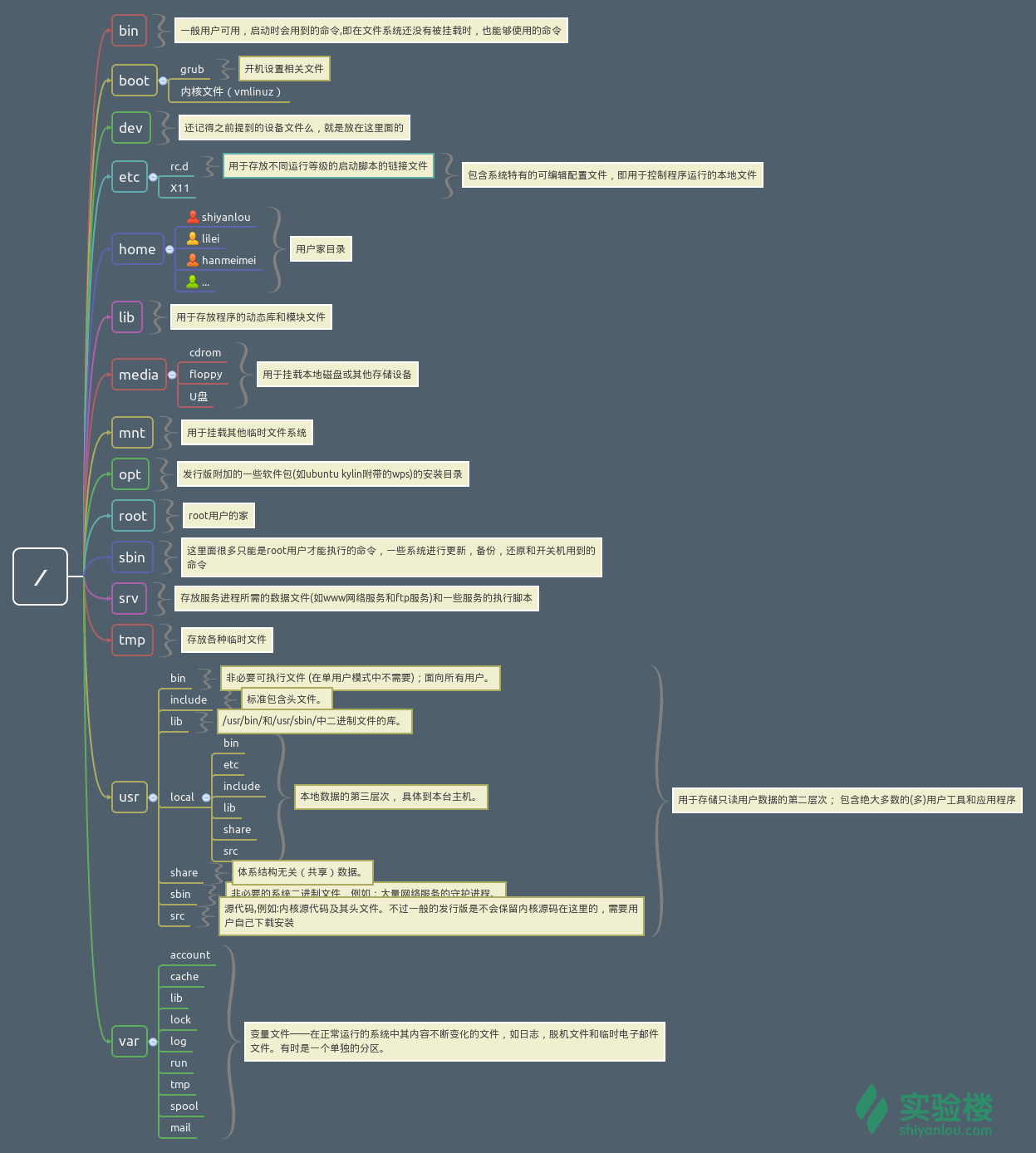
FHS（英文：Filesystem Hierarchy Standard 中文：文件系统层次结构标准），多数 Linux 版本采用这种文件组织形式，FHS 定义了系统中每个区域的用途、所需要的最小构成的文件和目录同时还给出了例外处理与矛盾处理。

FHS 定义了两层规范，**第一层是， / 下面的各个目录**应该要放什么文件数据，例如 /etc 应该放置设置文件，/bin 与 /sbin 则应该放置可执行文件等等。

**第二层则是针对 /usr 及 /var 这两个目录的子目录来定义**。例如 /var/log 放置系统日志文件，/usr/share放置共享数据等等。

[FHS\_2.3 标准文档](http://refspecs.linuxfoundation.org/FHS_2.3/fhs-2.3.pdf)

**如果觉得图片不清晰，建议另存为到本地放大查看：**



如果你觉得看这个不明白，那么可以试试最真实最直观的方式，执行如下命令：

$ tree /

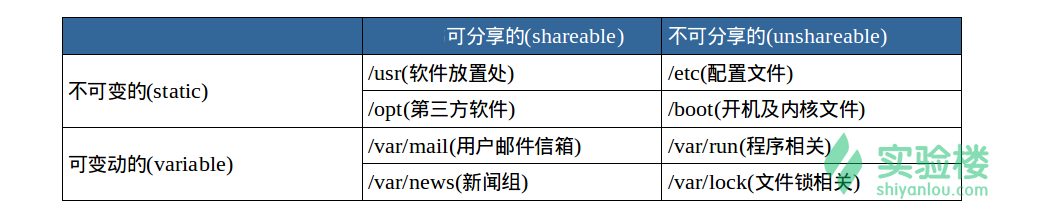
如果提示" command not found "，就先安装：

# 因为我们的环境的原因，每次新启动实验会清除系统恢复到初始状态，所以需要手动更新软件包索引，以便我们安装时能找到相应软件包的源。

sudo apt-get update

sudo apt-get install tree

关于上面提到的 FHS，这里还有个很重要的内容你一定要明白，FHS 是根据以往无数 Linux 用户和开发者的经验总结出来的，并且会维持更新，FHS 依据文件系统使用的频繁与否以及是否允许用户随意改动（注意，不是不能，学习过程中，不要怕这些），将目录定义为四种交互作用的形态，如下表所示：



2. 目录路径

#### 路径

有人可能不明白这路径是指什么，有什么用。顾名思义，路径就是你要去哪儿的路线嘛。如果你想进入某个具体的目录或者想获得某个目录的文件（目录本身也是文件）那就得用路径来找到了。

使用 cd 命令可以切换目录，在 Linux 里面使用 . 表示当前目录，.. 表示上一级目录（**注意，我们上一节介绍过的，以 . 开头的文件都是隐藏文件，所以这两个目录必然也是隐藏的，你可以使用 ls -a 命令查看隐藏文件**）, - 表示上一次所在目录，～ 通常表示当前用户的 home 目录。使用 pwd 命令可以获取当前所在路径（绝对路径）。

进入上一级目录：

$ cd ..

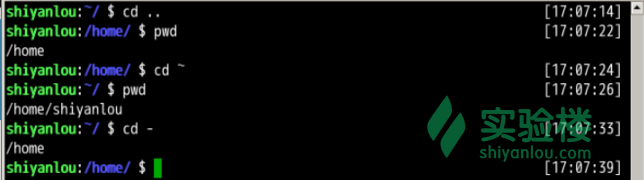
进入你的 home 目录：

$ cd ~

# 或者 cd /home/<你的用户名>

使用 pwd 获取当前路径：

$ pwd



#### 绝对路径

关于绝对路径，简单地说就是以根" / "目录为起点的完整路径，以你所要到的目录为终点，表现形式如： /usr/local/bin，表示根目录下的 usr 目录中的 local 目录中的 bin 目录。

#### 相对路径

相对路径，也就是相对于你当前的目录的路径，相对路径是以当前目录 . 为起点，以你所要到的目录为终点，表现形式如： usr/local/bin （这里假设你当前目录为根目录）。你可能注意到，我们表示相对路径实际并没有加上表示当前目录的那个 . ，而是直接以目录名开头，因为**这个 usr 目录为 / 目录下的子目录，是可以省略这个 . 的**（以后会讲到一个类似不能省略的情况）；如果是当前目录的上一级目录，则需要使用 ..，比如你当前目录为 home 目录，根目录就应该表示为 ../../ ，表示上一级目录（ home 目录）的上一级目录（ / 目录）。

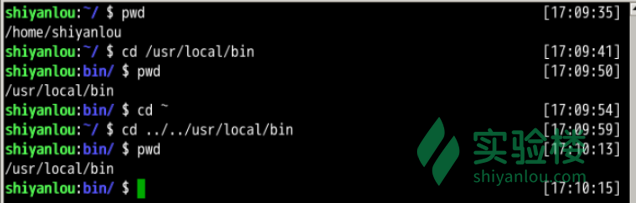
下面我们以你的 home目录为起点，分别以绝对路径和相对路径的方式进入 /usr/local/bin 目录：

# 绝对路径

$ cd /usr/local/bin

# 相对路径

$ cd ../../usr/local/bin



进入一个目录，可以使用绝对路径也可以使用相对路径，那我们应该在什么时候选择正确的方式进入某个目录呢。就是凭直觉嘛，你觉得怎样方便就使用哪一个，而不用特意只使用某一种。比如假设我当前在 /usr/local/bin 目录，我想进入上一级的 local 目录你说是使用 cd .. 方便还是 cd /usr/local 方便？而如果要进入的是 usr 目录，那么 cd /usr ，就比 cd ../.. 方便一点了。

**提示：在进行目录切换的过程中请多使用 Tab 键自动补全，可避免输入错误，连续按两次**Tab**可以显示全部候选结果。**

**三、Linux 文件的基本操作**

这一节我们主要讲解文件常用的基本操作，包括：新建、复制、删除、移动文件与文件重命名、查看文件、查看文件类型、以及编辑文件。

1. 新建

#### 新建空白文件

使用 touch 命令创建空白文件，关于 touch 命令，其主要作用是来更改已有文件的时间戳的（比如，最近访问时间，最近修改时间），但其在不加任何参数的情况下，只指定一个文件名，则可以创建一个指定文件名的空白文件（不会覆盖已有同名文件），当然你也可以同时指定该文件的时间戳，更多关于 touch 命令的用法，会在下一讲文件搜索中涉及。

创建名为 test 的空白文件，因为在其它目录没有权限，所以需要先 cd ~ 切换回用户的 /home/shiyanlou 目录：

$ cd ~

$ touch test

#### 新建目录

使用 mkdir（make directories）命令可以创建一个空目录，也可同时指定创建目录的权限属性。

创建名为“ mydir ”的空目录：

$ mkdir mydir

使用 -p 参数，同时创建父目录（如果不存在该父目录），如下我们同时创建一个多级目录（这在安装软件、配置安装路径时非常有用）：

$ mkdir -p father/son/grandson



后面的目录路径，以绝对路径的方式表示也是可以的。

2. 复制

使用 cp（**copy**）命令复制一个文件到指定目录。

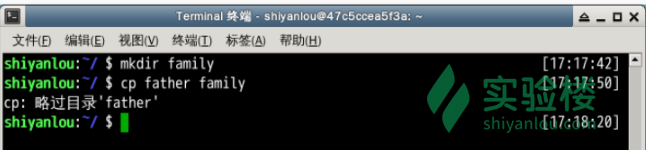
将之前创建的“ test ”文件复制到“ /home/shiyanlou/father/son/grandson ”目录中：

$ cp test father/son/grandson

是不是很方便啊，如果在图形界面则需要先在源目录复制文件，再进到目的目录粘贴文件，而命令行操作步骤就一步到位了嘛。

#### 复制目录

如果直接使用 cp 命令复制一个目录的话，会出现如下错误：



要成功复制**目录**需要加上 **-r 或者 -R 参数，表示递归复制**，就是说有点“株连九族”的意思：

$ cp -r father family

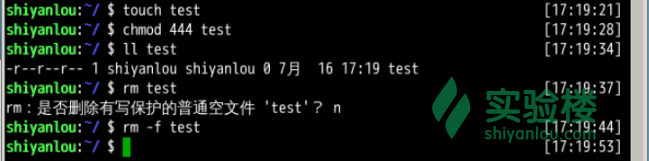
3. 删除

#### 删除文件

使用 rm（**remove** files or directories）命令删除一个文件：

$ rm test

有时候你会遇到想要删除一些为只读权限的文件，直接使用 rm 删除会显示一个提示，如下：



你如果想忽略这提示，直接删除文件，可以使用 **-f （force）参数强制删除**：

$ rm -f test

#### 删除目录

跟复制目录一样，要删除一个**目录**，也需要加上 **-r 或 -R**参数：

$ rm -r family

4. 移动文件与文件重命名

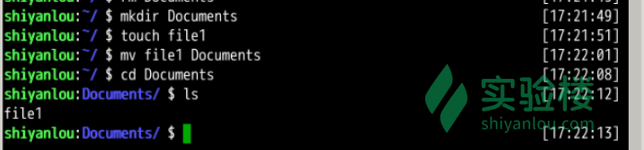
#### 移动文件

使用 mv（**move** or rename files）命令移动文件（剪切）。将文件“ file1 ”移动到 Documents 目录：

mv 源目录文件 目的目录：

$ mkdir Documents

$ mv file1 Documents



#### 重命名文件

将文件“ file1 ”重命名为“ myfile ”：

**mv 旧的文件名 新的文件名（两个都是文件类型）**：

$ mv file1 myfile

#### 批量重命名

要实现批量重命名，mv 命令就有点力不从心了，我们可以使用一个看起来更专业的命令**rename 来实现。不过它要用 perl 正则表达式来作为参数**，关于正则表达式我们要在后面才会介绍到，这里只做演示，你只要记得这个 rename 命令可以批量重命名就好了，以后再重新学习也不会有任何问题，毕竟你已经掌握了一个更常用的 mv 命令。

$ cd /home/shiyanlou/

# 使用通配符批量创建 5 个文件:

$ touch file{1..5}.txt

# 批量将这 5 个后缀为 .txt 的文本文件重命名为以 .c 为后缀的文件:

$ rename 's/\.txt/\.c/' \*.txt

#上面的例子中第一个参数中’s/A/B/’：将A替换为B，其中\.是转义符，就是表示.

# 批量将这 5 个文件，文件名和后缀改为大写:

$ rename 'y/a-z/A-Z/' \*.c

#上面的例子中第一个参数中’y/A/B/’：将A替换为B

简单解释一下上面的命令，rename 是先使用第二个参数的通配符匹配所有后缀为 .txt 的文件，然后使用第一个参数提供的正则表达式将匹配的这些文件的 .txt 后缀替换为 .c，这一点在我们后面学习了 sed 命令后，相信你会更好地理解。

5. 查看文件

#### 使用 cat，tac 和 nl 命令查看文件

前两个命令都是用来打印文件内容到标准输出（终端），其中 cat 为正序显示，**tac 为倒序显示**。

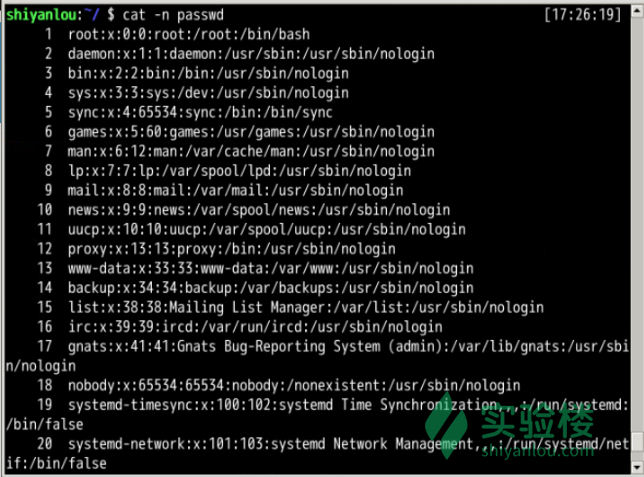
标准输入输出：当**我们执行一个 shell 命令行时通常会自动打开三个标准文件，即标准输入文件（stdin），默认对应终端的键盘、标准输出文件（stdout）和标准错误输出文件（stderr）**，后两个文件都对应被重定向到终端的屏幕，以便我们能直接看到输出内容。进程将从标准输入文件中得到输入数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。

比如我们要查看之前从 /etc 目录下拷贝来的 passwd文件：

$ cat passwd

可以加上**-n 参数显示行号**：

$ cat -n passwd



**nl 命令，添加行号并打印**，这是个比 cat -n 更专业的行号打印命令。

这里简单列举它的常用的几个参数：

-b : 指定添加行号的方式，主要有两种：

-b a:表示无论是否为空行，同样列出行号("cat -n"就是这种方式)

-b t:只列出非空行的编号并列出（默认为这种方式）

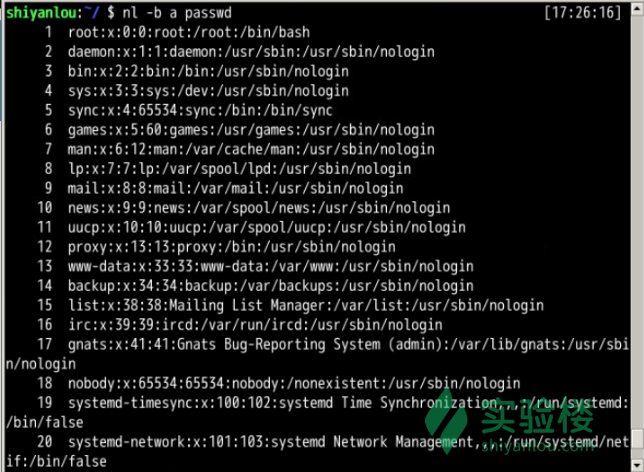
-n : 设置行号的样式，主要有三种：

-n ln:在行号字段最左端显示

-n rn:在行号字段最右边显示，且不加 0

-n rz:在行号字段最右边显示，且加 0

-w : 行号字段占用的位数(默认为 6 位)



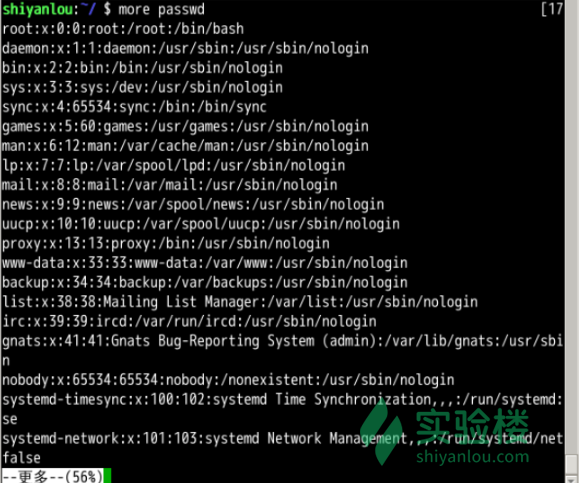
你会发现使用这几个命令，默认的终端窗口大小，一屏显示不完文本的内容，得用鼠标拖动滚动条或者滑动滚轮才能继续往下翻页，要是可以直接使用键盘操作翻页就好了，那么你就可以使用下面要介绍的命令。

#### 使用 more 和 less 命令分页查看文件

如果说上面的 cat 是用来快速查看一个文件的内容的，那么这个 more 和 less 就是天生用来"阅读"一个文件的内容的，比如说 man 手册内部就是使用的 less 来显示内容。其中 more 命令比较简单，只能向一个方向滚动，而 less 为基于 more 和 vi （一个强大的编辑器，我们有单独的课程来让你学习）开发，功能更强大。less 的使用基本和 more 一致，具体使用请查看 man 手册，这里只介绍 more 命令的使用。

使用 more 命令打开 passwd 文件：

$ more passwd



打开后默认只显示一屏内容，终端底部显示当前阅读的进度。可以使**用 Enter 键向下滚动一行，使用 Space键向下滚动一屏，按下 h 显示帮助，q 退出**。

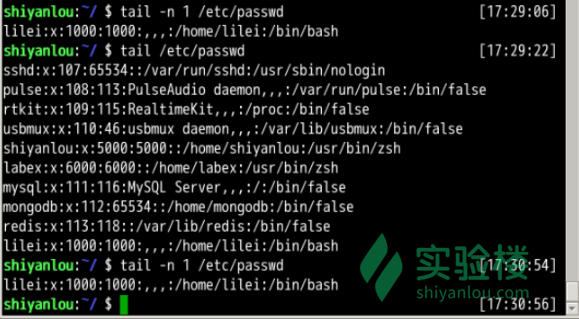
#### 使用 head 和 tail 命令查看文件

这两个命令，那些性子比较急的人应该会喜欢，因为它们一个是只查看文件的头几行（**默认为 10 行**，不足 10 行则显示全部）和尾几行。还是拿 passwd 文件举例，比如当我们想要查看最近新增加的用户，那么我们可以查看这个 /etc/passwd 文件，不过我们前面也看到了，这个文件里面一大堆乱糟糟的东西，看起来实在费神啊。因为系统新增加一个用户，会将用户的信息添加到 passwd 文件的最后，那么这时候我们就可以使用 tail命令了：

$ tail /etc/passwd

甚至**更直接的只看一行， 加上 -n 参数，后面紧跟行数**：

$ tail -n 1 /etc/passwd



关于 tail 命令，不得不提的还有它一个**很牛的参数 -f，这个参数可以实现不停地读取某个文件的内容并显示**。这可以让我们**动态查看**日志，达到实时监视的目的。不过我不会在这门基础课程中介绍它的更多细节，感兴趣的用户可以自己去了解。

6. 查看文件类型

前面我提到过，在 Linux 中文件的类型不是根据文件后缀来判断的，我们通常使用 file 命令查看文件的类型：

$ file /bin/ls



说明这是一个可执行文件，运行在 64 位平台，并使用了动态链接文件（共享库）。

7. 编辑文件

在 Linux 下面编辑文件通常我们会直接使用专门的命令行编辑器比如（emacs，vim，nano），由于涉及 Linux 上的编辑器的内容比较多，且非常重要，故我们有一门单独的基础课专门介绍这中一个编辑器 vim 。**强烈希望正在学习这门 Linux 基础课的你先在这里暂停一下，去学习**[**vim 编辑器**](http://www.shiyanlou.com/courses/2)**的使用（至少掌握基本的操作），然后再继续本课程后面的内容，因为后面的内容会假设你已经学会了 vim 编辑器的使用。**如果你想更加快速地入门，可以直接使用 Linux 内部的 vim 学习教程，输入如下命令即可开始：

$ vimtutor

**四、更多**

#### 轻松一下

你是不是觉得在我们的环境中学习轻松愉快毫无压力，所以偶尔偷偷懒也是没有问题的呢？要真是这样可不太好啊，要学会给自己一点压力，严格要求自己才行。你或许会想，要是有人能监督就好了，这样能学得更快。好吧，今天就教你怎么召唤一双眼睛出来监督你：

$ xeyes

你可以使用如下命令将它放到后台运行：

$ nohup xeyes &

