

Linux 简明教程

XDF 版

目的：让更多的人从零开始学会使用 *linux*，少走弯路

演示系统：*Ubuntu 10.04 desktop*

适用人群：未接触或刚刚接触 *linux* 的同志们

本教程使用 *Ubuntu 10.04 LTS - Lucid Lynx* 做演示，其他版本皆大同小异，有特殊的相异之处，我会特别指出，旨在为大家提供方便，勿做其他商业用途。

Ubuntu, Linux for human being !

献给全人类的 *Linux* !

发布日期：2010-9-13

Author:XDF

QQ: 1045163

Blog:www.xdf.me

About:www.xudafeng.com

E-mail:xdf@onsky.org

目 录

Linux 的简介-----

- 一、Linux 的起源-----
- 二、Linux 的优势-----

Linux 的安装-----

- 一、系统安装要求-----
- 二、安装前的知识介绍-----
 - 1. Linux 文件格式简介
 - 2. 推荐分区方法
- 三、安装方法的介绍-----
 - 1. Wubi 安装
 - 2. Windows+linux 双系统
 - 3. 只安装 Ubuntu
 - 4. 硬盘安装
- 四、硬件驱动的安装-----

Linux 的初体验-----

- 一、登录-----
 - 1. 本机登录
 - 2. 远程登录
 - 3. 登录失败
 - 4. 控制台的概念
 - 5. 更改密码
 - 6. 登出
 - 7. 注意事项
- 二、获取帮助-----
 - 1. help
 - 2. man
 - 3. info
 - 4. HOWTO
 - 5. 互联网大家庭
- 三、初识 shell-----

命令行工具-----

- 一、基本命令-----
 - 1. 显示文件名字
 - 2. 显示文本文件内容
 - 3. 删除文件
 - 4. 显示系统名字
 - 5. 关机或者重启-----
- 二、文件操作命令-----
 - 1. 复制文件
 - 2. 重新命名
 - 3. 测试文件内容
 - 4. 查找字符串
 - 5. 显示文件头部
 - 6. 显示文件尾部
 - 7. 忽略文件中的重复行

- 8. 比较两个文件
- 9. 按顺序显示文件内容

三、 进程间通信命令-----

四、 一些重要指令-----

- 1. 显示文本
- 2. 显示日期时间
- 3. 记录会话信息

五、 压缩和归档文件-----

- 1. 压缩文件
- 2. 解压缩文件
- 3. 其他压缩方法
- 4. 打包和解包文件

六、 定位命令-----

- 1. 定位工具
- 2. 搜索关键字
- 3. 搜索文件

七、 获取用户和系统信息-----

- 1. 列出系统上的用户
- 2. 另一种方法
- 3. 推荐使用的

八、 用户间通信-----

- 1. 发送消息
- 2. 接受消息和拒绝消息

九、 发一封邮件-----

文件系统-----

一、 目录操作-----

- 1. 创建目录
- 2. 工作目录
- 3. 更改工作目录
- 4. 删除目录
- 5. 移动目录

二、 访问权限-----

- 1. 显示访问权限
- 2. 改变访问权限
- 3. 目录访问权限
- 4. 链接的概念

Shell-----

一、 什么是 shell-----

二、 shell 命令行-----

- 1. 语法
- 2. 处理机制
- 3. 执行

三、 与 shell 相关的配置-----

- 1. shell 环境变量
- 2. 元字符详细操作表
- 3. bash 的相关配置

四、 标准输入输出-----

- 1. 标准输入输出的机制
- 2. 重定向(I/O 重定向)
 - 1. 重定向的基本概念
 - 2. I/O 重定向的预备知识
 - 3. 基本 I/O
 - 4. 进阶 I/O

5. 文件重写问题

3. 管道

- 3.1 管道的特点
- 3.2 管道的创建
- 23.3 两个方向输入

五、简单的 shell 编程-----

1. 运行 Shell 脚本

2. Shell 变量及其相关命令

- 2.1 读写 shell 变量
- 2.2 命令替换
- 2.3 导出环境变量
- 2.4 重置变量
- 2.5 创建只读用户自定义变量
- 2.6 读取标准输入

3. 脚本参数传递

4. 注释与程序头

5. 程序控制

- 5.1 if-then-elif-else-fi 语句
- 5.2 for 语句
- 5.3 while 语句
- 5.4 until 语句
- 5.5 break 和 continue 命令
- 5.6 case 语句

6. 常用命令集锦

六、shell 后台运行程序-----

编辑器的使用-----

一、vi 的基本概念-----

- 1. 命令行模式
- 2. 插入模式
- 3. 底行模式(最后一行模式)

二、vi 的基本操作-----

- 1. 进入 vi
- 2. 切换至插入模式
- 3. Insert 的切换
- 4. 退出 vi 及保存文件

三、命令行模式介绍-----

- 1. 插入模式
- 2. 从插入模式切换为命令行模式
- 3. 移动光标
- 4. 删除文字
- 5. 复制
- 6. 替换
- 7. 回复上一次操作
- 8. 更改
- 9. 跳至指定的行

四、最后一行模式命令简介-----

- 1. 列出行号
- 2. 跳到文件中的某一行
- 3. 查找字符
- 4. 保存文件
- 5. 离开 vi

深入高阶(编程技术)-----

一、输入源代码-----

二、编译，链接，运行-----

三、调试程序-----

服务器技术-----

一、引言-----

二、Tomcat 容器-----

 1.安装 j2sdk

 2.安装 tomcat

三、Apache Web 服务器-----

四、Tomcat 与 Apache 的整合-----

 1.配置前的初始化工作

 1.1 安装 j2sdk

 1.2 安装 Apache 服务器

 1.3 安装 Tomcat 服务器

 1.4 安装 mod_jk

 1.5 准备工作目录和测试文件

 2.配置 Apache 服务器

 3.配置 Tomcat 服务器

 3.1 建立 workers.properties 文件

 3.2 修改 server.xml 文件

 3.3 修改 jk2.properties 文件

 4.运行测试程序

start...

Linux 的简介

一.Linux 的起源

至少创始人大家要了解下，Linux 是由 Linus Torvalds（芬兰）开发的，1991 年 9 月发布了 0.01 版本，当然是起源于 UNIX。接下来介绍 Ubuntu 版本，Ubuntu 是 Ubuntu 将定期不断发布；每六个月就会有一个新版本。每个版本提供至少 18 个月的支持，LTS 版本支持长达 3 年。有意思的是，Ubuntu

的每个版本号都会以动物命名，例如（Lucid Lynx 既是 10.04 的版本号）。Ubuntu 默认的桌面环境是 GNOME（后面会讲到其他的桌面环境），而内核是基于 Debian（Linux 的一个优秀版本，其余的还有 Red Hat 等，操作差异不大）。别觉得我唠叨，尤其是关键的部分，要跟大家说的详细些，后面按需配有插图。

二.Linux 的优势

第一，应用标准友好，计算机业界曾经制定了可移植计算机环境操作系统接口（简称 POSIX），这个标准是基于 UNIX 的，并被大家广泛接受。

第二，软件源支持，只有自由软件远远不够的，大部分情况需要源代码，或者与编译好的二进制文件。

第三，对外为设备的广泛兼容，凡是能及时发布相关规范和驱动源码的厂商的硬件设备都会被 Linux 速度更新支持。

第四，平台范围广，Linux 可以运行在你所知道的绝大部分计算机上，Intel，Power PC，甚至可以在一部 htc 手机环境下模拟运行，更神奇的是早就传闻塞班的虚拟环境也能流畅运行 Linux 系统。

第五，兼容环境的友好，为了兼容其他操作系统的软件，Linux 提供了程序模拟器。例如，DOS，Macintosh，Linux 提供的 wine 程序可以方便用户在 Linux 环境下运行 Windows 程序。

另外，Linux 具有内核编程接口，支持多用户，多任务，而且支持分层文件系统，做编程开发也很有优势。其余的啥子高级特性大家谷歌吧，就不多废话了。

Linux 的安装

一、系统安装要求

几乎没有要求，只要是可以安装 windows 就能安装 linux

二、安装前的知识介绍

1.Linux 文件格式简介

Ext2：是 GNU/Linux 系统中标准的文件系统，其特点为存取文件的性能极好，对于中小型的文件更显示出优势，这主要得利于其簇快取层的优良设计。这是 Linux 中使用最多的一种文件系统，它是专门为 Linux 设计的，拥有极快的速度和极小的 CPU 占用率。Ext2 既可以用于标准的块设备（如硬盘），也被应用在软盘等移动存储设备上。

Ext3：文件系统，它属于一种日志文件系统，是对 ext2 系统的扩展。它兼容 ext2，并且从 ext2 转换成 ext3 并不复杂，就是保有 Ext2 的格式之下再加上日志功能。Ext3 是一种日志式文件系统（Journal File System），最大的特点是：它会将整个磁盘的写入动作完整的记录在磁盘的某个区域上，以便有需要时回溯追踪。当在某个过程中断时，系统可以根据这些记录直接回溯并重整被中断的部分，重整速度相当快。

Ext4：是一种针对 ext3 系统的扩展日志式文件系统，是专门为 Linux 开发的原始的扩展文件系统（ext 或 extfs）的第四版。Ext4：第 4 个扩展文件系统，是下一代的日志文件系统，它与上一代文件系统 ext3 是向后兼容的。ext4 在性能、伸缩性和可靠性方面进行了大量改进，支持 1 EB 的文件系统。引入到 2.6.19 内核中。Ubuntu9.04 开始支持 Ext4 文件系统。

Linux swap：它是 Linux 中一种专门用于交换分区的 swap 文件系统，相当与 windows 里面的虚拟内存。Linux 是使用这一整个分区作为交换空间。一般这个 swap 格式的交换分区是主内存的 2 倍。在内存不够时，Linux 会将部分数据写到交换分区上。

VFAT：VFAT 叫长文件名系统，这是一个与 Windows 系统兼容的 Linux 文件系统，支持长文件名，可以作为 Windows 与 Linux 交换文件的分区。

ReiserFS：也有日志功能，处理小档案时速度快。

XFS：稳定与高效，在大小档案混合的系统里有优越的表现

JFS：特色是最不占用 CPU 资源

2. 推荐分区方法：

如果是单独安装 linux 可以不这么分配，这是给爽系统准备的

/swap 通常是内存的两倍，本人内存 4G，分了 4G, 分大了也没什么用，按照你的内存大小酌情分配。

/这个是根分区, 分个 10G 就足够

/boot 分个 128M

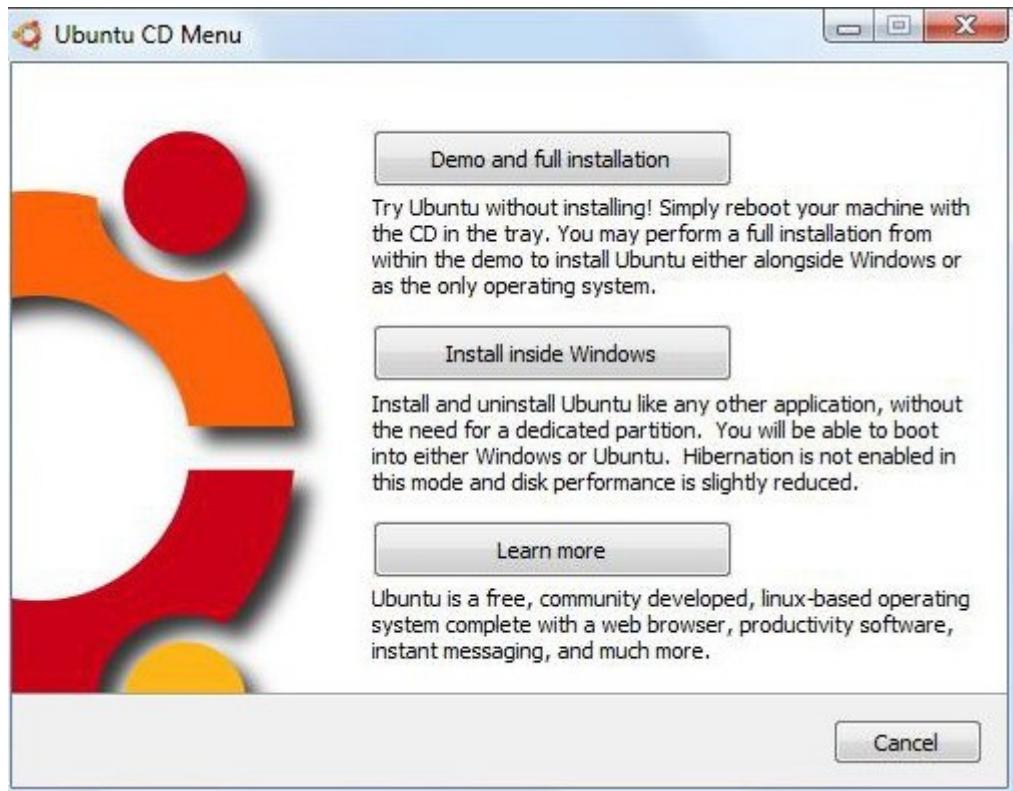
/home 是存放你的文件, 这个分最大剩余空间，确保/home 分区在最后面或最前面，至于文件格式, 选 Ext4。

注意：把/home 单独分出来, 要是重装系统的时候, 就可以把根分区 /, /boot 分区, /swap 都格掉或删除, 只要不动/home 分区, 你的个人资料就不会丢失。装系统的时候一定要选择到你原来的/home 分区并挂起, 这样系统才知道你还是要用原来的/home 分区的。关于重新挂载/home 分区: 在重装系统到硬盘分区步骤的时候, 保留/home 分区, 将 swap, /, /boot 分区删掉重新分割。找到原来的/home 分区, 在弹出的菜单“编辑分区”中“新建分区容量”保持不变, 在“用于”中选择到你原来的分区格式, 不勾选“格式化分区”, “挂载点”选择“/home”, 个人资料就不会丢失。

三、安装方法的介绍

1. Wubi 安装

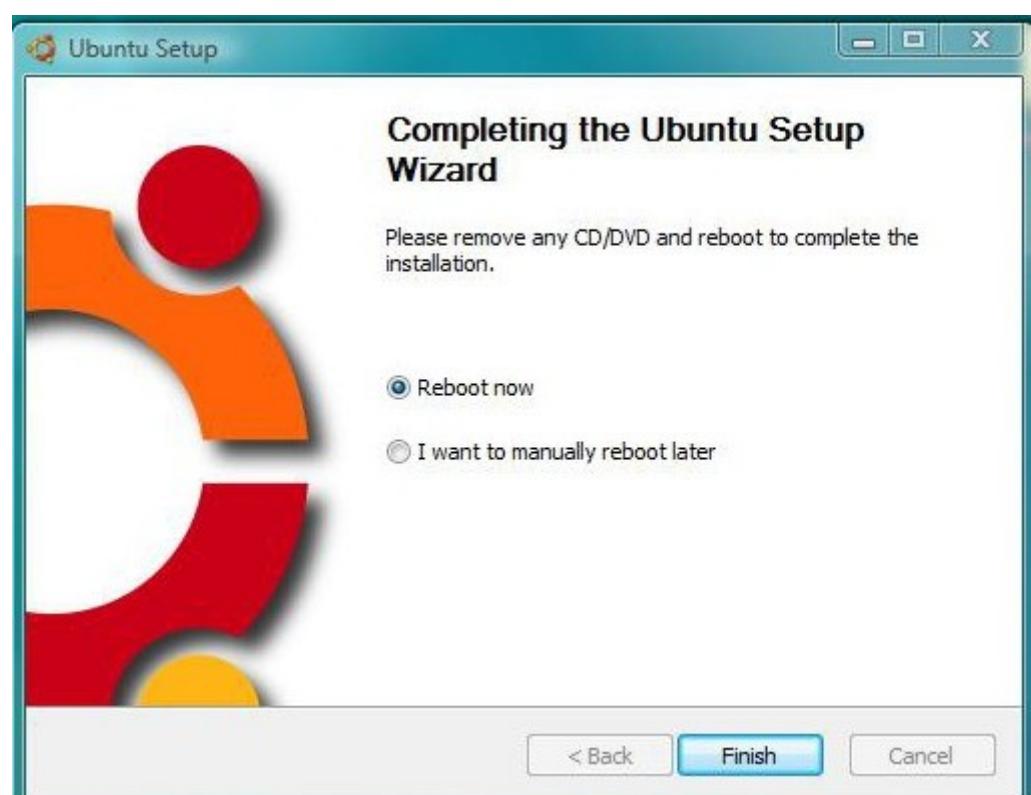
先保证你有至少剩余 15G 的硬盘空间。这种安装方法, 休眠功能不可用, 并且磁盘性能会略有降低。由于是采用 windows 的 mbr 引导, 所以不要升级跟 grub 有关的内核程序, 不然, 将无法进入 windows。



然后在弹出的对话框中选择好要安装的驱动器, 安装大小, 语言, 用户名, 和设置好密码:
点击安装, 系统开始安装:



系统会自动进行安装直到安装完成。安装完成后,系统要求重新启动:, 在系统刚刚开始启动的时候,会看到 ubuntu 的启动选项,选择“ubuntu”回车进入,系统进行初始化工作,大概需要 10 分钟左右就完全安装好了。



2.Windows+linux 双系统



在光驱放入 Ubuntu 的安装光盘,启动系统。系统由光盘引导后,你将见到一个语言选择画面:选择中文(简体)进入,就能见到中文的安装程序选择菜单了:



然后选择“安装 Ubuntu(I)”,进行安装:



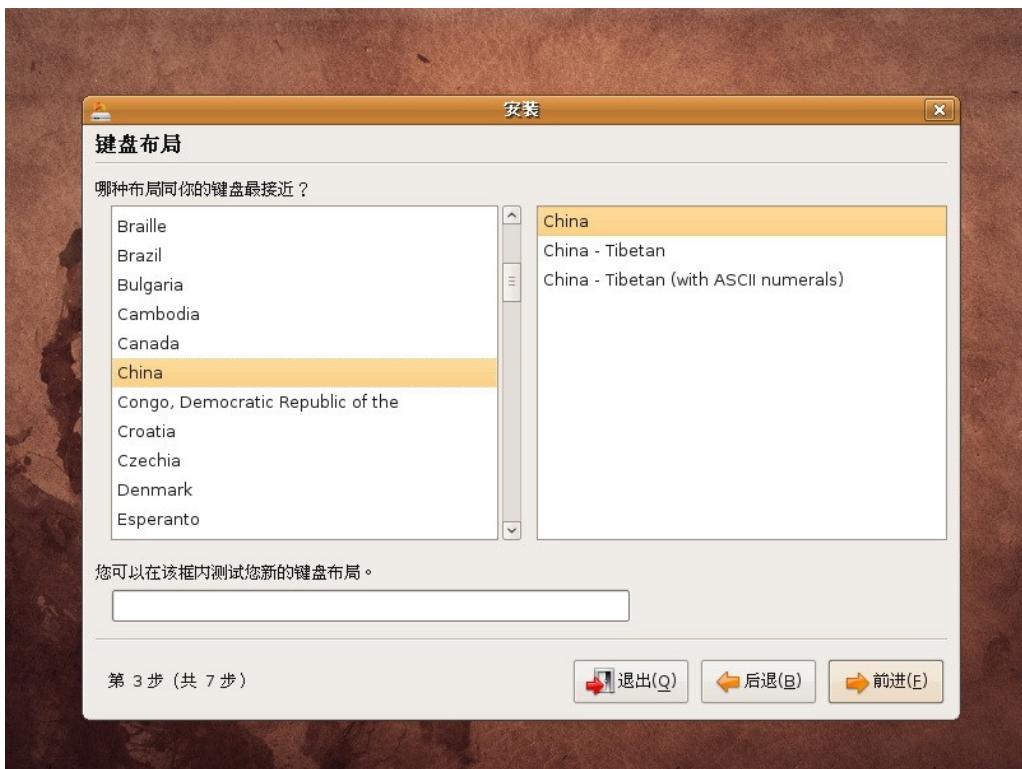
稍等一下，就会出现如下界面，由于我们前面已经选择了简体中文，所以在这里一进默认选好了中文（简体），前进，时区选择：



默认即可，前进



默认就好了,点前进,键盘布局:



通常情况下保持默认就可以, 前进,硬盘分区信息:



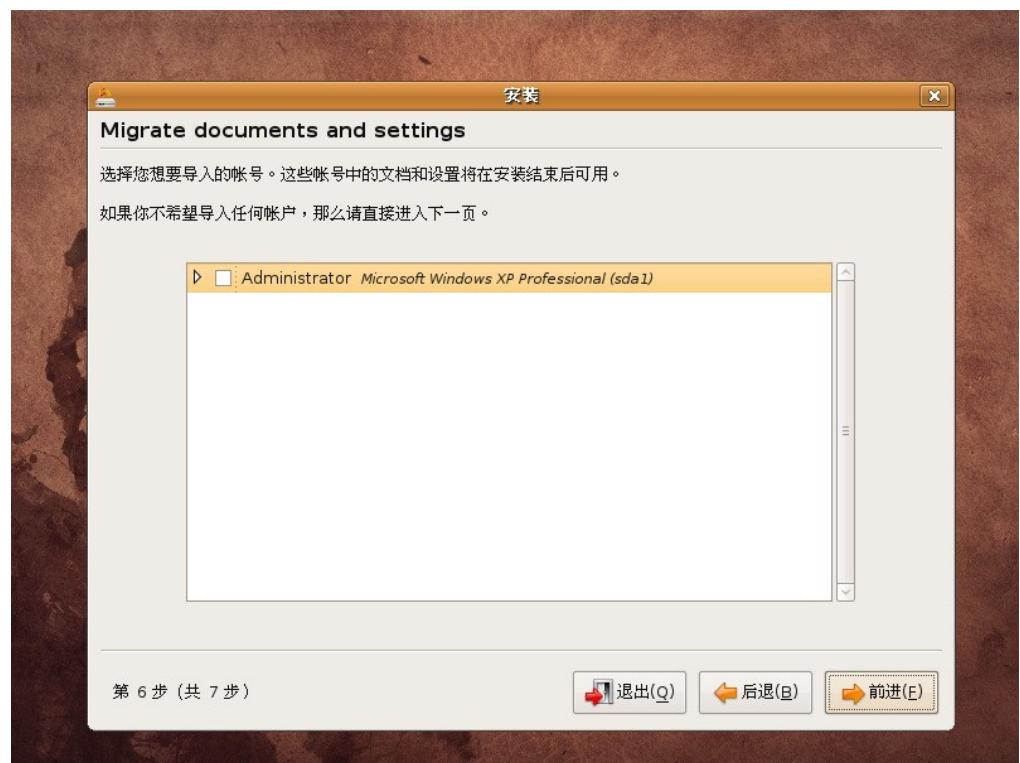
在这里选择手动分区,前进,根据自己的需要进行系统分区:



上一节已经将了如何合理的分区:



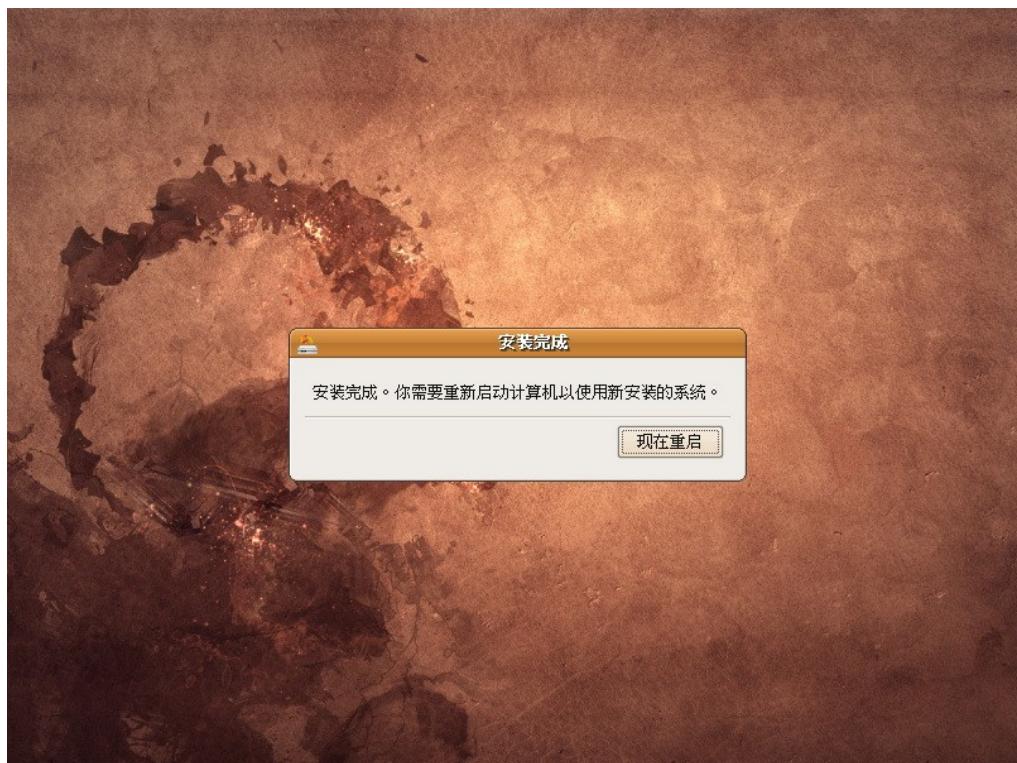
用户迁移向导,这一步将把机器里已有的 Windows 系统中的设置,比如收藏夹,个人文档等都转移到 Ubuntu 中:



确认分区信息:



点安装,正式开始安装:



系统重新启动后,就可以直接进入 linux。

3.只安装 Ubuntu

安装步骤方法如上中所述,只是在分区时可以选择“向导-使用整个磁盘”进行分区或手动分区就

可以了。这样安装,电脑中就只有一个 Ubuntu 系统系统了,而且使用了所有的硬盘空间。

4. 硬盘安装

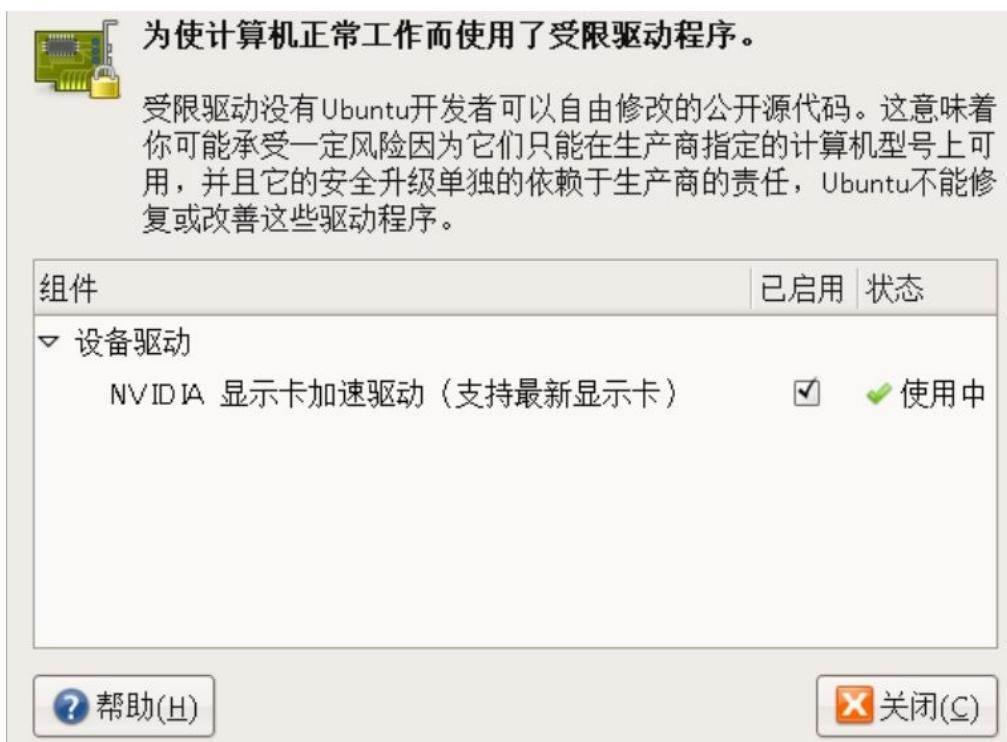
这个就有些麻烦了,需要下载 Grub4DOS, 所以不再赘述。如果一定要选择这种方式安装, 就只好到 <http://www.onsky.org>, 去看我原来发过的一段文章, 《linxu 安装手记》。

四、硬件驱动的安装

在 windows 中的习惯,安装好系统第一件事情就是安装硬件驱动了,Ubuntu 能自动识别到绝大多数硬件,并已配置好了相关驱动程序。查看是否有硬件没有被系统识别到,依次打开“系统-系统管理-硬件驱动”,系统不能识别的硬件会在这里列出了。最常出现的就是独立显卡不能识别了,这需要手动安装私有驱动。依次打开“系统-系统管理-硬件驱动”,这里会列出你的显卡:勾选“已启用”,系统自动从网络上下载相关私有驱动,安装完成后重新启动系统,驱动就安装好了。如果你还有别的硬件没有识别到,在“硬件驱动”里都会被相应的列出,同样的安装方法安装就好了。最好先设置好软件源, 163 或者 sohu 的镜像都很快, 由于使用教育网, 我就使用了 <http://ubuntu.uestc.edu.cn>。ubuntu 官方上海源, 提供 Kernel, Hiweed 文件存放在/etc/apt/sources.list。

代码:

```
deb http://mirror.rootguide.org/ubuntu/ lucid main universe restricted multiverse
deb-src http://mirror.rootguide.org/ubuntu/ lucid main universe restricted multiverse
deb http://mirror.rootguide.org/ubuntu/ lucid-security universe main multiverse restricted
deb-src http://mirror.rootguide.org/ubuntu/ lucid-security universe main multiverse restricted
deb http://mirror.rootguide.org/ubuntu/ lucid-updates universe main multiverse restricted
deb http://mirror.rootguide.org/ubuntu/ lucid-proposed universe main multiverse restricted
deb-src http://mirror.rootguide.org/ubuntu/ lucid-proposed universe main multiverse restricted
deb http://mirror.rootguide.org/ubuntu/ lucid-backports universe main multiverse restricted
deb-src http://mirror.rootguide.org/ubuntu/ lucid-backports universe main multiverse restricted
deb-src http://mirror.rootguide.org/ubuntu/ lucid-updates universe main multiverse restricted
```



Linux 的初体验

一、登录

1.本机登录

安装完成之后终于可以使用期待已久的 Linux 了，这是面临的第一个问题就是系统的登录问题。相信大家已经都安装了 desktop 的，直接输入用户口令就可以登录了。如果没有安装 desktop 的话，那便需要从终端登录，只是缺少了图形界面而已，输入 login 用户 ID，和口令（口令是不可见的，* 之类也不会显示）即可。登录后应该会显示 shell 的一些提示信息。

2.远程登录

还有一种登录方式就是远程的登录，例如使用 crt 在 windows 下操作 Linux。这时我们就会面临选择链接方式的问题，ssh 或者 telnet。总所周知，telnet 是以明文传送密码，不安全度很高，推荐大家使用 ssh (port: 22) 连接远程终端。直接输入 telnet 或者 ssh 后面跟 ip 地址就 OK 了。

3.登录失败

密码输入错误会要求重新输入。显示 permission denied，而且要求重试。输入一定要注意大小写。

4.控制台的概念

控制台是为了方便大家操作的虚拟终端，按下 Ctrl + Alt + Fx 来切换控制台。

5.更改密码

当需要修改密码的时候，命令行上面输入 sudo passwd (由于 Ubuntu 的密码是随机的，所以要重新指定口令，而且要使用 sudo 语句 Unix 下直接输入 passwd，以后都以 Ubuntu 操作为准,Unix)

6.登出

或者叫注销，直接按下 Ctrl + D 或者输入命令 exit (与 windows 中 DOS 退出一样)。

7.注意事项

登录后默认是用户账户，而有些操作需要切换到超级账户 root 才能执行，如果大家对系统不熟，慎用超级账户。后面会讲进入 root 的方法。

二、获取帮助

1.*help*

一般是 (指令 + 空格 + --help)，是一个简单的帮助提示

2.*man*

(man + 空格 + 指令) 查看，man 类似于一个程序自带的 doc 文档

Linux 的 man 很强大，该手册分成很多 section，使用 man 时可以指定不同的 section 来浏览，各个 section 意义如下：

1 - commands-----	是普通的命令
2 - system calls-----	是系统调用
3 - library calls-----	是库函数
4 - special files-----	是特殊文件
5 - file formats and conventions-----	是指文件的格式
6 - games for linux-----	是关于游戏的
7 - macro packages and conventions-----	是附件还有一些变量
8 - system management commands-----	是系统管理用的命令
9 - 其他-----	

想要指定 section 就直接在 man 的后面加上数字,比如：

man 1 ls

man 3 printf

对于像 open,kill 这种既有命令,又有系统调用的来说,man open 则显示的是 open(1),也就是从最前面的 section 开始,如果想查看 open 系统调用的话,就得 man 2 open (来源百度)

3.info

info 是一种文档格式，也是阅读此格式文档的阅读器，我们常用它来查看 linux 命令的 info 文档。Menu 下方就是本主题内的命令，敲 M 键，然后输入命令的名称就可以查看该命令的 info 帮助文档了。几个常用快捷键：

? : 它就会显示 info 的常用快捷键
N: 显示下一节点的文档内容
P: 显示前一节点的文档内容
U: 进入当前命令所在的主题
M: 敲 M 键后输入命令的名称就可以查看该命令的帮助文档了
G: 敲 G 键后输入主题名称，进入该主题
L: 回到上一个访问的页面
SPACE: 向前滚动一页
BACKUP 或 DEL: 向后滚动一页
Q: 退出 info

4.HOWTO

Linux HOWTOs 是一份详细描述在某一方面规划或使用 Linux 的文件。例如，有一份 Installation HOWTO，说明如何安装 Linux，而 Mail HOWTO 说明了如何在 Linux 下建立及设定电子邮递系统。如何取得 Linux HOWTOs? HOWTOs 可经由下面的 FTP 站取得：<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO> 也可以在许多的镜像网站找到。关于 HOWTO 的翻译：HOWTO 的翻译也可以在 metalab.unc.edu 及映射站台找到，目前已有下面的语言版本：

- * 中文 Chinese (zh)
- * 克罗埃西亚语 Croatian (hr)
- * 法文 French (fr)
- * 德文 German (de)
- * 希腊语 Hellenic (el)
- * 义大利文 Italian (it)
- * 印尼语 Indonesian (id)
- * 日文 Japanese (ja)
- * 韩文 Korean (ko)
- * 波兰文 Polish (pl)
- * 斯拉维尼亚语 Slovenian (sl)
- * 西班牙文 Spanish (es)
- * 瑞典文 Swedish (sv)
- * 土耳其文 Turkish (tr)

5.互联网大家庭

学习最大的动力就是自主和兴趣，百度谷歌一定要有问题就去互联网上查，这样才能不断得快速进步推荐几个网站：

[中国最大的 Linux 技术社区](#)
[红旗 Linux](#)
[LinuxEden-Linux 伊甸园](#)
[Linux 频道—中国 IT 实验室](#)
[红联 Linux 门户 - 中国领先的 Linux 技术网站](#)
[Linux 宝库 - 全球最大中文开源门户](#)
[中国 Linux 论坛](#)
[Linux 中国](#)
[Ubuntu 中文论坛](#)
[天空工作室](#)

三、初识 shell

大家先记住，shell 简单来讲就是大家看见的 Linux 命令行，我可以通过它来操作 Linux，那么，后面章节会有更加详细的 shell 教程。

命令行工具

一、基本命令（经过上一节对 shell 的初步认识，这一节介绍一些基本的命令）

1. 显示文件名字

ls: 显示文件名字

用法: ls [选项]... [文件]...

列出 FILE 的信息(默认为当前目录)。

如果不指定 -cftuvSUX 或 --sort 选项，则根据字母大小排序。

ls 就是单词 list 的意思，可以列出当前工作目录下的文件

2. 显示文本文件内容

cat: 显示文本文件内容

用法: cat [选项]... [文件]...

将[文件]或标准输入组合输出到标准输出。

cat 就是单词 catenate 的意思，就是将文件内容连接到一起

3. 删除文件

rm: 删除文件

用法: rm [选项]... 文件...

删除 (unlink) 文件。

rm 就是单词 remove 的缩写，就是删除的意思，也可以使用 rm -i 命令做到询问后删除，输入字母 y 即为同意

4. 显示系统名字

直接输入命令 hostname 即可，我的显示 xdf-onsky，关于怎么改暂时没什么必要讲了

5. 关机或者重启

shutdown -h now: 指令关闭计算机

shutdown -r now: 指令重新启动计算机

reboot: 快速重启

halt: 直接关机

now 也可以换成相应的时间参数，换成 1 就代表一分钟后执行

二、文件操作命令

1. 复制文件

cp: 复制文件

用法: cp [选项]... [-T] 源文件 目标文件

或: cp [选项]... 源文件... 目录

或: cp [选项]... -t 目录 源文件...

将源文件复制至目标文件，或将多个源文件复制至目标目录。

cp 顾名思义就是 copy 啦，如果目标文件名字已经存在，那么就会重写目标文件，这是就被替换掉，而且系统没有提示，所以，同 rm 命令相似，cp -i 也可以安全的交互执行

2. 重新命名

mv: 更改文件名

用法: mv [选项]... [-T] 源文件 目标文件

或: mv [选项]... 源文件... 目录

或: mv [选项]... -t 目录 源文件...

将源文件重命名为目标文件，或将源文件移动至指定目录。

mv 就是 move 的意思啦，先接原文件名，后接更新后的文件名，同样的是，mv 也会覆盖，所以不多说了，使用 mv -i

3. 测试文件内容

file: 检测文件，相当于 windows 下面的文件属性

用法: file [选项]... 文件...

直接写成 file exfile 就可以了，可以看见文件的属性

4. 查找字符串

grep: 查找字符串

用法: grep [选项]... 模式 [文件]...

在每个文件中查找样式或标准输入。

PATTERN 默认的是一个基本的正则表达式(BRE)。

例: grep -i 'hello world' menu.h main.c

grep 有点长，global regular expression print 既是其缩写，大家试试吧，可以用到以前学的 cat 指令

5. 显示文件头部

head: 显示文件头部

用法: head [选项]... [文件]...

将每个指定文件的头 10 行显示到标准输出。

如果指定了多个文件，在每一段输出前会给出文件名作为文件头。

如果不指定文件，或者文件为"-”，则从标准输入读取数据。

head 不解释，后面可以加参数，比方说 xdf@xdf-onsky:/etc\$ head -1 passwd (passwd 是示例文件) 代表从文件中显示一行

6. 显示文件尾部

tail: 显示文件尾部

用法: tail [选项]... [文件]...

显示每个指定文件的最后 10 行到标准输出。

若指定了多个文件，程序会在每段输出的开始添加相应文件名作为头。

如果不指定文件或文件为"-”，则从标准输入读取数据。

tail 就是尾巴的意思了，用法同 head

7. 忽略文件中的重复行

uniq: 忽略文件中的重复行

用法: uniq [选项]... [文件]

从输入文件或者标准输入中筛选相邻的匹配行并写入到输出文件或标准输出。

uniq 就是 unique 的意思

8. 比较两个文件

diff: 比较两个文件

用法: diff [选项]... FILES

逐行比较文件。

diff 就是 different 的意思，diff -u exfile1 exfile2，里面的一 + 问题大家一眼就看明白了

9. 按顺序显示文件内容

sort: 按顺序显示文件内容

用法: sort [选项]... [文件]...

或: sort [选项]... --files0-from=F

串联排序所有指定文件并将结果写到标准输出。

长选项必须使用的参数对于短选项时也是必需使用的。

三、进程间通信命令

此功能是通过管道实现的，管道的意思就是把一个进程的输出作为另一个进程的输入，两个命令之间用 | 隔开，比方说 cat exfile | head，有些不好理解，以后仔细体会。

四、一些重要指令

1. 显示文本

echo: 显示文本

echo 用过 php 的都知道什么意思了，就是个输出功能

2. 显示日期时间

date: 显示日期时间

用法: date [选项]... [+格式]

或: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

以给定的格式显示当前时间，或是设置系统日期。

指令直接使用即可

3. 记录会话信息

script: 记录会话信息

直接使用指令 script 记录开始，退出后输入 cat typescript 即可查看记录

五、压缩和归档文件

1. 压缩文件

bzip2: 压缩文件

bzip2 是个压缩工具，将文件压缩成为.bz2 格式，就像 Windows 下的 zip 道理一样了，下面介绍解压工具

2. 解压缩文件

bunzip2: 解压缩文件

bunzip2 就是 bzip 的还原工具了，用法直接在命令后面接文件名就可以。这里介绍一个查看方法，使用 bzcat 即可在不解压的情况下查看压缩文件的内容

3. 其他压缩方法

gunzip: 压缩文件，跟 bzip2 用法类似，只是版本较低

gunzip 用来压缩文件，跟 bzip2 用法类似，只是版本较低，生成后缀名为.gz 的文件，同样可以使用 zcat 查看。

4. 打包和解包文件

tar: 打包和解包文件

用法: tar [选项...] [FILE]...

GNU ‘tar’

将许多文件一起保存至一个单独的磁带或磁盘归档，并能从归档中单独还原所需文件。

tar 就是 tape archive 的简写，就是将很多文件归档为一个 tar 文件，tar 后面加选项，-c (creat 创建)，-v (verbose 包含详细信息)，和-f (file 从一个文件进行读写)。比方说，tar -cvf。Tar 包可以通过上述的 gunzip 来解压。

六、定位命令

1. 定位工具

which 和 whereis: 定位工具

which 主要是找到 path 下的文件目录，而 whereis 则是找到相关的文件

2. 搜索关键字

apropos: 搜索关键字

`apropos` 主要在 man 页面进搜索关键字，用来查找不知道的或者知道但是不太确定名字的命令

3. 搜索文件

`slocate`: 搜索文件

用于在本地系统搜索文件

七、获取用户和系统信息

1. 列出系统上的用户

`who`: 列出系统上的用户

用法: `who [选项]... [文件 | 参数 1 参数 2]`

显示当前已登录的用户信息。

有时候系统的一些用户是远程登录过来的，这是为了方便对用户的管理，可以使用 `who` 命令查看到底那些用户在使用系统

2. 另一种方法

`finger`: 同样可以列出系统上的用户

`finger` 工具显示的信息更加全面些，甚至告诉了你用户的个人信息和用户全名，有些安全隐患，一些不怀好意的人可以通过 `finger` 显示的信息对系统进行破坏。

3. 推荐使用的

`w`: 同样可以列出系统上的用户

具体的关于，`who,finger,w` 的比较，大家可以自己 google

八、用户间通信

1. 发送消息

`write`: 发送消息

`whier username [terminal]`

终端名字很重要哦，可以通过 `who,finger,w` 得到

2. 接受消息和拒绝消息

`mesg n`: 拒绝消息

`mesg y`: 接受消息

`yes` 还是 `no` 大家自己试试吧

九、发一封邮件

邮件大家都会发了吧，安装一个吧，我就用这个。

文件系统

一、目录操作

linux 下的目录文件管理方式跟 windows 下是一样的，其实只是表面上一样
弄清可见文件和不可见文件名，用“.”或者“..”开始的文件名都是不可见的

那么接下来介绍管理目录的命令行

1. 创建目录

`mkdir`: 创建目录

用法: `mkdir [选项]... 目录...`

若指定目录不存在则创建目录。

`makedir`就是创建目录的意思，相当与 windows 下的新建文件夹
`mkdir`创建了目录后，可是怎么分辨所看见的到底是目录还是文件呢？Ubuntu 为大家分颜色视图，非常方便，或者使用 `ls -F` 也可以，目录的后面会显示“/”，记住是大写的 F

2.工作目录

`pwd`: 显示工作目录
直接输入就可以了，默认的工作目录是`/home/username`

3.更改工作目录

`cd`: 更改工作目录
跟 windows 下的 dos 命令提示符是一样的

4.删除目录

`rmdir`: 删除目录
用法：`rmdir [选项]... 目录...`
删除指定的空目录
就像 `rm` 删除文件一样，只是现在删除 dir 目录，也可以使用 `rm -r filename` 来递归删除，这样方便一些

5.移动目录

`mv`: 移动目录
就是 move 的意思，之所以与重命名文件的指令相同，我还没搞懂

重要的标准目录和文件，给大家一一列出

-----Ubuntu 目录-----

目录说明

/	根目录：最主要的，位于结构树根的目录
bin/	基本命令和二进制文件：包含启动系统和用户运行所需的文件
boot/	启动引导系统加载的静态文件，所有与系统启动有关的文件都保存在这里
grub/	Grub 引导器相关的文件
dev/	设备文件：包含所有代表外围设备的文件
etc/	系统程序和大部分应用程序的全局配置文件
init.d/	SystemV 风格的启动脚本
rcX.d/	启动脚本的链接, 定义运行级别
network/	网络配置文件
X11/	图形界面配置文件
home/	home 主目录
user/	普通用户的\$HOME 目录
.../	
lib/	系统程序库文件，共享库和内核模块

media/	-----挂载媒体设备
mnt/	-----临时挂载，临时文件系统挂载点
opt/	-----附加软件包
proc/	内核与进程镜像，虚拟文件系统的内核和进程信息
root/	root 用户的\$HOME 目录
sbin/	管理员系统程序
tmp/	临时文件
usr/	
---bin/	-----应用程序
---sbin/	-----管理员应用程序
---lib/	-----应用程序库文件
---share/	-----应用程序资源文件
---src/	-----应用程序源代码
---local/	
---soft/	-----用户程序
---.../	-----通常使用单独文件夹
---X11R6/	-----图形界面系统
var/	-----动态数据
lost found/	-----磁盘修复文件

二、访问权限

1. 显示访问权限

ls -l: 显示访问权限

中间有空格，大家要注意了。显示顺序：文字类型（1个字符），访问权限（9个字符），链接数目，文件属主，用户组，大小，最后修改时间，文件名

2. 改变访问权限

chmod: 改变访问权限（这个暂时先这么些）

用法: chmod [选项]... 模式[,模式]... 文件...

或: chmod [选项]... 八进制模式 文件...

或: chmod [选项]... --reference=参考文件 文件...

将每个文件的模式更改为指定值。

直接在 chmod 后面加，这个命令很重要，优势编写的脚本文件是需要读取权限的，如果都没办法读取了，怎么执行呢？(setuid 和 setgid)

3. 目录访问权限

ls -ld: 显示当全目录的执行权限

可以在目录下使用 ls -l 来查看当前用户对当前目录的权限

4. 链接的概念

Shell

一、什么是 shell

之前已经有了初识 shell 的章节，那么现在要系统的介绍 shell，首先摒弃你的 MS-DOS 观点，shell

不是简单的命令行，而是全功能的编程环境，shell 是操作系统的一部分，用来与用户打交道，并且可以用来协调各个命令。在 Linux 中，标注的 shell 是'bash'，即 the GNU Bourne-Again Shell (bash 的英文意为“重击”——译注)。那么久介绍下 shell 的体现-终端 (Terminal)。

常见的 shell：

-shell	path	name -
-Bourne shell	/usr/bin/sh	sh
-C shell	/usr/bin/csh	csh
-Bourne Again shell	/bin/bash	bash
-Korn shell	/usr/bin/ksh	ksh
-TC shell	/usr/bin/tcsh	tcsh

二、shell 命令行

1. 语法

一般的命令都具有参数，比方说 \$cp file1 file2，也有一些没有参数的，比方说 who 指令。如果需要带参数的指令大家没有加参数，系统就会报错。需要加选项的命令给我们带来了很多方便，比方说 ls -l，按竖排显示目录，或者 ls -F 命令就是在 ls 后面加了 -F，即可加上文件类型的指示符号。更多的命令可以在后面使用—help 来查看，选项同样可以合并，不信你可以试一试，ls -F1。

2. 处理机制

接收命令行 (tty 设备驱动检查命令行交给 shell) — 分析命令行 (由 shell 处理解析命令)

3. 执行

进程：shell 将命令参数等传递给程序后便将启动一个进程，同时进入休眠状态，进程结束后，推出状态会传递给 shell，这是 shell 被激活。shell 会忽略参数，只是传递参数，交给程序执行。

三、与 shell 相关的配置

1. shell 环境变量

可通过 set 命令查看，给大家举个例子，列出一些常用的。

```
BASH_VERSION='3.2.48(1)-release'  
EUID=0  
HISTFILE=/root/.bash_history  
HOME=/root  
HOSTNAME=onsky  
HOSTTYPE=i486  
LOGNAME=root  
MACHTYPE=i486-pc-linux-gnu  
MAIL=/var/mail/root  
OSTYPE=linux-gnu  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games  
PWD=/var/www  
SHELL=/bin/bash  
SSH_CLIENT='121.49.103.16 1827 22'(之前提到过 ssh 需要 22 端口)  
SSH_CONNECTION='121.49.103.16 1827 10.20.208.18 22'  
SSH_TTY=/dev/pts/2  
UID=0  
USER=root
```

由于我是远程登录的终端，所以会显示 SSH 的客户端的 ip，也就是我的上网 ip 地址。

2. 元字符详细操作表

元字符	作用	举例
\$	结束一行；间接引用 shell 变量	uid:"\$UID"
&	让命令到后台执行	command &
-	引用多字符	'\$UID'
*	匹配 0 个或多个字符	ls /etc/p*sswd
[]	插入通配符	ls /etc/rc[0-6].d
\`	命令替换	NAME=`echo zgg`
	创建命令之间的管道	ls wc -l
;	顺序执行命令	date; sleep 1; date
命令的重定向输出	echo > /tmp/test.txt	
<	命令的重定向输入	wc -l < /tmp/test.txt
?	通配符为单个字符	ls /etc/pass??
{ }	在当前 shell 中执行命令列表	
^	开始一行以及否定符号	
()	执行子 shell 中的命令列表	
New Line	结束命令行	
Space	分隔命令行中的元素	
Tab	分隔命令行中的元素	
#	注释	
"	引用多字符，允许替换	
/	路径分隔符	
\	转义字符	
!	启动历史列表中指定的事件以及当前事件 !vi	
~	home 目录	~

3.bash 的相关配置

Bash 相关配置文件：/etc/profile， 登录时的系统初始化文件

四、标准输入输出

1. 标准输入输出的机制

首先呢，shell 默认把键盘输入作为标准输入，而把屏幕输出作为标准输出。例如，cat 工具就是把键盘输入作为标准输入，而把屏幕输出作为标准输出。其实呢，每个计算机设备在系统中都有自己的编号，设备编号很重要，那么系统已经把键盘设备作为标准输入，而把显示屏幕作为标准输出。用 who 工具看看你的系统的情况吧，可以看见 ip 的。当前的设备查看可以使用 tty 命令。

2. 重定向(I/O 重定向)

1、重定向的基本概念

Linux 重定向用途很广，举个例子，上一节我们知道 shell 默认把键盘输入作为标准输入，而把屏幕输出作为标准输出。但如果我不想看见显示器的显示结果，而是想把结果输出到一个文件中保存起来，或者我想在后台操作中保存日志文件，想查看的时候查看，平时不显示怎么办呢？我们可以使用重定向来改变这一设置，使用 wc 指令要手动输入一篇文字来计算字符数的，用了重定向后可以直接把一个已经写好的文件用 '<' 指向这条命令，就可以直接统计这个文件的字符数了。输出也一样，你可以把屏幕输出重定向到一个文件里，再到文件里去看结果。

重定向操作符可以用来将命令输入和输出数据流从默认位置重定向到其他位置，其输入或输出数据流的位置称为句柄；常见的句柄有三种，更可以自行扩展。I/O 重定向通常与 FD (File Descriptor, linux 下，所有的操作都是对文件进行操作，而对文件的操作是利用文件描述符(file descriptor)来实现的。在源代码中，一般用 fd 作为文件描述符的标识) 有关，shell 的 FD 通常为 10 个，即 0 ~ 9；下面这三个：默认与 keyboard、monitor、monitor 有关

句柄	句柄代号	句柄描述
STDIN	0	键盘输入
STDOUT	1	输出信息到提示符窗口
STDERR	2	输出错误信息到提示符窗口

默认的'<'重定向输入操作符是 0，而默认的'>'重定向输出操作符是 1。键入'<'或'>'操作符之后，必须指定数据的读写位置，可以是文件名或其他现有的句柄。
要指定重定向到现有句柄，请使用与 & 字符，后面接要重定向的句柄号（即 &句柄号）。

举个例子：命令可以将句柄 2（即 STDERR）重定向到句柄 1（即 STDOUT）：2>&1

现在我们回过头来看看上面的那条语句 mysh > mylog.txt 2>&1 就可明白：

> mylog.txt 意思是将标准输出重定向到 mylog.txt，等价于 mysh 1> mylog.txt；

2 >& 1 意思是将错误输出重定向到句柄 1 标准输出；

综合起来就是 mysh 命令执行过程中产生的标准输出和错误输出都会被重定向到 mylog.txt 中；重定向的功能十分强大，甚至可以用 mysh >null 2>null 或者 mysh >/dev/null 2>/dev/null 来消除记录。

2.I/O 重定向的预备知识

在 IO 重定向中，STDOUT 与 STDERR 的管道会先准备好，才会从 STDIN 读进资料； bash (ksh) 执行命令的过程：分析命令—变量求值—命令替代（``和\$()）—重定向—通配符展开—确定路径—执行命令；() 将 command group 置于 sub-shell 去执行，也称 nested sub-shell，它有一点非常重要的特性是：继承父 shell 的 Standard input, output, and error plus any other open file descriptors。

3.基本 I/O

下表列出了可用于重定向输入和输出数据流的操作符：

操作符	功能描述
>	将命令输出写入文件或设备，而不是命令提示符或句柄
<	从文件而不是从键盘或句柄读入命令输入
>>	将命令输出添加到文件末尾而不删除文件中已有的信息
>&	将一个句柄的输出写入到另一个句柄的输入中
<&	从一个句柄读取输入并将其写入到另一个句柄输出中
	从一个命令中读取输出并将其写入另一个命令的输入中；也称为管道操作符
>	file 把 stdout 重定向到 file 文件中；
>>	file 把 stdout 重定向到 file 文件中(追加)；
1>	file 把 stdout 重定向到 file 文件中；
> file 2>&1	把 stdout 和 stderr 一起重定向到 file 文件中；
2> file	把 stderr 重定向到 file 文件中；
2>> file	把 stderr 重定向到 file 文件中(追加)；
>> file 2>&1	把 stderr 和 stderr 一起重定向到 file 文件中(追加)；
< file >file2	以 file 文件作为 stdin，以 file2 文件作为 stdout；
cat <>file	以读写的方式打开 file；
< file	以 file 文件作为 stdin；
<<	从 stdin 中读入，直至遇到 delimiter 分界符。

4.进阶 I/O

>&n 使用系统调用 dup (2) 复制文件描述符 n 并把结果用作标准输出；

<&n 标准输入复制自文件描述符 n；

<&- 关闭标准输入（键盘）；

>&- 关闭标准输出；

n<&- 表示将 n 号输入关闭；

n>&- 表示将 n 号输出关闭；

上述所有形式都可以前导一个数字，此时建立的文件描述符由这个数字指定而不是缺省的 0 或 1。如：

... 2>file 运行一个命令并把错误输出(文件描述符 2)定向到 file。

... 2>&1 运行一个命令并把它的标准输出和输出合并。(严格的是通过复制文件描述符 1 来建立文件描述符 2，但效果通常是合并了两个流。)

我们对 2>&1 详细说明一下：2>&1 也就是 FD2 = FD1，这里并不是说 FD2 的值 等于 FD1 的值，因为 > 是改变送出的数据信道，也就是说把 FD2 的“数据输出通道”改为 FD1 的“数据输出通道”。如果仅仅这样，这个改变好像没有什么作用，因为 FD2 的默认输出和 FD1 的默认输出本来都是 monitor，一样的！但是，当 FD1 是其他文件，甚至是其他 FD 时，这个就具有特殊的用途了。请大家务必理解这一点。exec 命令：常用来替代当前 shell 并重新启动一个 shell，换句话说，并没有启动子 shell。使用这一命令时任何现有环境都将会被清除。exec 在对文件描述符进行操作的时候，也只有在这时，exec 会覆盖你当前的 shell 环境

```
exec 0>exec 1>outfilename # 打开文件 outfilename 作为 stdout。  
exec 2>errfilename # 打开文件 errfilename 作为 stderr。  
exec 0<&- # 关闭 FD0。  
exec 1>&- # 关闭 FD1。  
exec 5>&- # 关闭 FD5。
```

5. 文件重写问题

```
set -o noclobber  
set +o noclobber
```

3. 管道

管道和有名管道是最早的进程间通信机制之一，管道可用于进程间的通信，有名管道克服了管道没有名字的限制，因此，除具有管道所具有的功能外，它还允许无亲缘关系进程间的通信。认清管道和有名管道的读写规则是在程序中应用它们的关键，在管道和有名管道的通信机制的基础上，用实例对其读写规则进行了程序验证。

1. 管道的特点：

管道是半双工的，数据只能向一个方向流动；需要双方通信时，需要建立起两个管道；只能用于父子进程或者兄弟进程之间（具有亲缘关系的进程）；单独构成一种独立的文件系统：管道对于管道两端的进程而言，就是一个文件，但它不是普通的文件，它不属于某种文件系统，而是自立门户，单独构成一种文件系统，并且只存在与内存中。简单来讲，就是将一个文件的输出重定向到一个文件，然后将该文件作为另一个命令的输入。

数据的读出和写入：一个进程向管道中写的内容被管道另一端的进程读出。写入的内容每次都添加在管道缓冲区的末尾，并且每次都是从缓冲区的头部读出数据。

2. 管道的创建：

```
#include int pipe(int fd[2])
```

该函数创建的管道的两端处于一个进程中间，在实际应用中没有太大意义，因此，一个进程在由 pipe() 创建管道后，一般再 fork 一个子进程，然后通过管道实现父子进程间的通信，因此也不难推出，只要两个进程中存在亲缘关系，这里的亲缘关系指的是具有共同的祖先，都可以采用管道方式来进行通信。

3. 管道的处理机制：

管道两端可分别用描述字 fd[0]以及 fd[1]来描述，需要注意的是，管道的两端是固定了任务的。即一端只能用于读，由描述字 fd[0]表示，称其为管道读端；另一端则只能用于写，由描述字 fd[1]来表示，称其为写端。如果试图从管道写端读取数据，或者向管道读端写入数据都将导致错误发生。一般文件的 I/O 函数都可以用于管道，如 close、read、write 等等。

从管道中读取数据：如果管道的写端不存在，则认为已经读到了数据的末尾，读函数返回的读出字节数为 0；当管道的写端存在时，如果请求的字节数目大于 PIPE_BUF，则返回管道中现有的数据字节数，如果 请求的字节数目不大于 PIPE_BUF，则返回管道中现有数据字节数，此时，管道中数据量小于请求的数据量，或者返回请求的字节数，此时，管道中数据量不小于请求的数据量。

3. 两个方向输出：

tee 工具可以实现将标准输入复制到文件和标准输出。

Tee 工具的用法。 \$ who | tee who.out | grep root

五、简单的 shell 编程

1. 运行 Shell 脚本

增加文件可执行权限

```
$ chmod +x script_file
```

直接通过 shell 命令执行

```
$ /bin/sh script_file
```

```
$ /bin/bash script_file
```

脚本文件开头指定 shell

```
#!/bin/bash
```

2. Shell 变量及其相关命令

可写环境变量

只读环境变量

环境变量	含义
\$0	程序名
\$1 - \$9	第 1-9 个命令行参数的值
\$*	所有命令行参数的值
\$@	所有命令行参数的值
\$#	命令行参数的总数
\$\$	当前的进程 ID(PID)
\$?	上一条命令的退出状态(0 - 成功, 其他输出 - 失败)
\$!	最近后台进程(在后台运行的最后一个进程)的进程 ID
相关命令	
set	
env(不如 set 命令完整)	

2.1 读写 shell 变量

```
variable1=value1 [variable2=value2 ... variableN=valueN]
```

注意:

等号(=)前后没有空格, 如果值中含有空格, 须使用引号将这个值括起。

多个赋值语句可写在一行, 但必须用空格或分号分隔

```
$ name=Guoqiang  
$ echo $name  
Guoqiang  
$ name='Zhou Guoqiang'  
$ echo $name  
Zhou Guoqiang
```

```
$ my_uid='$UID'  
$ echo $my_uid  
$UID  
$ my_uid="$UID"  
$ echo $my_uid  
1000
```

2.2 命令替换

```
`command`  
$ echo now: `date`  
now: Sun Mar 23 17:44:36 CST 2008  
  
$ now=date: `date`"  
$ echo $now  
date: Sun Mar 23 17:45:45 CST 2008  
  
$ now='date: `date`'  
$ echo $now  
date: `date`
```

2.3 导出环境变量

命令 export 可以向后继 shell 进程传递变量的值
export [name-list]

2.4 重置变量

unset [name-list]

2.5 创建只读用户自定义变量

```
readonly [name-list]  
类似 PHP 中的常量  
$ name=zgq gender=male  
$ readonly name gender  
$ echo $name $gender  
zgq male  
$ name=zhang3 gender=female  
bash: name: readonly variable
```

2.6 读取标准输入

read variable-list

3. 脚本参数传递

9 个参数(\$1 - \$9)有时不够用
可以使用 shift 命令将参数左移 N 个位置
shift [N]
set [options] [argument-list]
依次将各位置参数的值设为 argument-list 中指定的参数
如果不带任何参数, set 将显示所有 shell 变量以及他们当前的值

4. 注释与程序头

```
#!/bin/bash  
  
#Linux 简明教程-XDF 版  
#版本: 1.01 Alaph  
#作者: XDF  
#UESTC  
#2010 年 9 月 13 日
```

5. 程序控制

1. if-then-elif-else-fi 语句

if expression

```
then
    elif expression
    then
        ...
    else
        ...
fi
expression 可以使用 test 命令也可以使用括号
if test $# -eq 0
等价为:
if [ $# -eq 0 ]
(注意[和]前后的空格)
```

2. *for* 语句

```
for variable [in argument-list]
do
    command-list
done
```

如果没有提供 argument-list，参数由命令行提供

3. *while* 语句

```
while expression
do
    command-list
done
```

如果死循环，就<^C>吧

4. *until* 语句

```
until expression
do
    command-list
done
```

5. *break* 和 *continue* 命令

6. *case* 语句

```
case test-string in
    pattern1)
        command-list1;;
    pattern2)
        command-list2;;
    patternN)
        command-listN;;
    *)
        end-command-list;;
esac
```

6. 常用命令集锦

杀死 firefox 进程

```
$ ps aux | grep firefox | grep -v grep | cut -c 9-15 | xargs kill -9
```

将当前目录下 php 文件中的 foo 题换成 bar

```
$ sed *php -in-place -e's/foo/bar/g'
```

根据 ip 查看 bbs.insenz.com 的访问次数

```
$ cat bbs.insenz.com-access_log | awk '{print $1}' | sort | uniq -c | sort -nr
```

查看远程服务器的磁盘空间状况

```
$ ssh zhouguoqiang@10.0.1.33 'df -h'
```

```
查看 svn 副本中某两个目录下的代码总行数  
$ find src/ tests/ -type f | grep -v .svn | xargs wc -l | awk '{ sum += $1; } END { print sum;}'
```

六、shell 后台运行程序

当用 bash 启动一个程序,你可以用在程序后加&,实现程序在后台运行;

例如: emacs &但假如你忘了输入&, 但你又不想把程序停止后重新启动; 按 Ctrl + Z 暂停前台工作, 用 jobs 命令查看当前 bash 一共启动了多少程序, 可以看到你的 emacs 以及它的 jobID, 它的状态是 Stoped 的, 用命令 bg + jobID 这样你就可以达到和 emacs &一样的效果了。当你有些工作比如 updatedb, find 等需要一定时间, 这些工作可以考虑让它在后台运行, 而你可以用 bash 做其它工作。假如需要把任务回到前台, 只需要 fg + jobID。

编辑器的使用

一、vi 的基本概念

本文介绍了 vi (vim) 的基本使用方法, 但对于普通用户来说基本上够了! Vi 编辑器是所有 Unix 及 Linux 系统下标准的编辑器, 简单地介绍一下它的用法和一小部分指令。由于对 Linux 系统的任何版本, vi 编辑器是完全相同的, Vi 也是 Linux 中最基本的文本编辑器。基本上 vi 可以分为三种状态, 分别是命令模式 (command mode)、插入模式 (Insert mode) 和底行模式 (last line mode) 就叫他为最后一行模式, 各模式的区别 (这节的按键描述很多, 统一使用「」符号, 「q」代表按 q 键) :

1. 命令行模式

控制屏幕光标的移动, 字符、字或行的删除, 移动复制某区段及进入插入模式下, 或者到最后一行模式。

2. 插入模式

在插入模式下, 才可以做文字输入, 按「ESC」键可回到命令行模式。

3. 底行模式(最后一行模式)

将文件保存或退出 vi, 也可以设置编辑环境, 如寻找字符串、加入行号等一些功能。
不过一般我们在使用时把 vi 简化成两个模式, 就是将最后一行模式也算入命令行模式, 毕竟最后一行模式也是一些命令嘛。

二、vi 的基本操作

1. 进入 vi

在系统提示符号输入 vi 及文件名称后, 就进入 vi 全屏幕编辑画面:
\$ vi xdf, 随便创建一个文件作为练习, 不过有一点要特别注意, 就是您进入 vi 之后, 是处于命令行模式, 这时候还不能输入汉字, 所以要切换到插入模式才行。

2. 切换至插入模式

在命令行模式按一下字母「i」就可以进入插入模式, 这时候你就可以开始进行编辑了。

3. Insert 的切换

您目前处于插入模式, 您就只能一直输入文字, 如果您发现输错了字! 想用光标键往回移动, 将该字删除, 就要先按一下「ESC」键转到命令行模式再删除文字。

4. 退出 vi 及保存文件

在命令行模式下，按一下":冒号键进入最后一行模式

: w name (输入 w name 将文章以指定的文件名 name 保存)
: wq (输入「wq」，保存并退出 vi)
: q! (输入 q!，不保存退出 vi)
q 就是 quit 的意思

三、命令行模式介绍

1. 插入模式

按「i」切换进入插入模式，按“i”进入插入模式后是从光标当前位置开始输入文件，按「a」进入插入模式后，是从目前光标所在位置的下一个位置开始输入文字，按「o」进入插入模式后，是插入新的一行，从行首开始输入文字。

2. 从插入模式切换为命令行模式

按「ESC」键。

3. 移动光标

vi 可以直接用键盘上的光标来上下左右移动

按「h」、「j」、「k」、「l」，分别控制光标左、下、上、右移一格

按「ctrl+b」：屏幕往“后”移动一页

按「ctrl+f」：屏幕往“前”移动一页

按「ctrl+u」：屏幕往“后”移动半页

按「ctrl+d」：屏幕往“前”移动半页

按数字「0」：移到文章的开头

按「G」：移动到文章的最后

按「\$」：移动到光标所在行的“行尾”

按「^」：移动到光标所在行的“行首”

按「w」：光标跳到下个字的开头

按「e」：光标跳到下个字的字尾

按「b」：光标回到上个字的开头

按「#l」：光标移到该行的第#个位置

4. 删除文字

「x」：删除光标所在位置的后面一个字符

「#x」：「nx」表示删除光标所在位置的后面 n 个字符

「X」：大写的 X，删除光标所在位置的前面一个字符

「#X」：「nX」表示删除光标所在位置的前面 n 个字符

「dd」：删除光标所在行

「#dd」：从光标所在行开始删除#行

5. 复制

「yw」：将光标所在之处到字尾的字符复制到缓冲区中

「#yw」：复制#个字到缓冲区

「yy」：复制光标所在行到缓冲区

「#yy」：「nyy」表示拷贝从光标所在的该行下数 n 行文字

「p」：将缓冲区内的字符贴到光标所在位置。与“y”有关的复制命令都必须与“p”配合才能完成复制与粘贴功能

6. 替换

「r」：替换光标所在处的字符

「R」：替换光标所到之处的字符，直到按下「ESC」键为止

7. 回复上一次操作

「u」：如果您误执行一个命令，可以马上按下「u」，回到上一个操作，相当于撤销

8. 更改

「cw」：更改光标所在处的字到字尾处
「c#w」：「cnw」表示更改 n 个字

9. 跳至指定的行

「ctrl + g」列出光标所在行的行号
「#G」：「nG」，表示移动光标至文章的 n 行

四、最后一行模式命令简介

在使用最后一行模式之前，请记住先按「ESC」键确定您已经处于命令行模式下后，再按「:」冒号即可进入最后一行模式。

1. 列出行号

输入「set nu」后，会在文件中的每一行前面列出行号，输入「set nonu」后，会在取消在文件中的每一行前面列出行号。

2. 跳到文件中的某一行

「#」
：「#」号表示一个数字，在冒号后输入一个数字，再按回车键就会跳到该行了

3. 查找字符

「/关键字」：先按「/」键，再输入您想寻找的字符，如果第一次找的关键字不是您想要的，可以一直按「n」会往后寻找到您要的关键字为止。
「?关键字」：先按「?」键，再输入您想寻找的字符，如果第一次找的关键字不是您想要的，可以一直按「n」会往前寻找到您要的关键字为止。

4. 保存文件

「w」：在冒号输入字母「w」就可以将文件保存起来。

5. 离开 vi

「q」：按「q」就是退出，如果无法离开 vi，可以在「q」后跟一个「!」强制离开 vi。
「qw」：一般建议离开时，搭配「w」一起使用，这样在退出的时候还可以保存文件。

深入高阶（编程技术）

这一节在教程的测试版本里就不详细说了，会出现在正式发布版里，
我们说一下基本的 hello world 程序的流程：

一、输入源代码

```
$ emacs hello.c          进入 emacs 编辑器环境
#include <stdio.h>          包含 c 语言的头文件
int main()
{
    printf("xdf 's tutorial for Linux");    输出一行字
```

```
exit(0);
}
完成输入后，按住 CTRL 键，按下 x，再按下 c，最后松开 CTRL 程序保留并退出 emacs 环境。
```

这个编辑器会在正式版里面讲，现在还是使用我们熟悉的 vi 吧。

```
$ vim hello.c          进入 vim 编辑器环境，进入 insert 模式
#include <stdio.h >      包含 c 语言的头文件
int main()
{
printf("xdf 's tutorial for Linux");    输出一行字
exit(0);
}
保存退出
```

二、编译，链接，运行

```
$ gcc -o hello hello.c
$ ./hello
xdf 's tutorial for Linux
$

使用 gcc 编译器进行编译，会看见本目录下还多了 a.out
./指明了所执行程序的路径是本目录下
好了，已经输出了我们要的结果
```

三、调试程序

如果要使用 gdb 调试程序，那么在上一步编译的时候，记得加上 -g 选项

```
$ gcc -g -o hello hello.c
$ gdb hello 进入 gdb 进行调试
```

服务器技术

一、引言

linux 最为广泛的用途莫过于做服务器，个人认为服务器 red hat 要更好些。在许多用户看来，一个网站 的成败主要在于它所提供的内容和功能，而支持这些内容和功能的 Web 服务器起着非常重要的作用。Linux 的效率是按年来计算的，所以可见 linux 做服务器的优势，确实要比 2003sever 稳定，接下来的很多也摘自互联网。如果配置出现问题可以联系我，一起解决问题。配置服务器问题很多的，大家可以到 www.onsky.org 去看，我分享可很多这方面的。

二、Tomcat 容器

Tomcat 是一个免费的开源的 Servlet 容器，主要用于 jsp 的开发环境，它是 Apache 基金会的 Jakarta 项目中的一个核心项目，由 Apache, Sun 和其它一些公司及个人共同开发而成。Tomcat 是稳固的独立的 Web 服务器与 Servlet Container，不过，其 Web 服务器的功能则不如许多更健全的 Web 服务器完整。

1. 安装 j2sdk

1.1 需要先安装 j2sdk，到 <http://www.java.sun.com> 下载 j2sdk 的二进制文件，如 j2sdk-1_4_2_04-linux-i586-rpm.bin

1.2 在终端中转到 j2sdk-1_4_2_04-linux-i586-rpm.bin 所在的目录，输入命令 chmod +x j2sdk-1_4_2_04-linux-i586-rpm.bin, 添加执行的权限

1.3 执行命令 ./j2sdk-1_4_2_04-linux-i586-rpm.bin，生成 j2sdk-1_4_2_04-linux-i586.rpm 的文件。

1.4 执行命令 chmod +x j2sdk-1_4_2_04-linux-i586.rpm，给 j2sdk-1_4_2_04-linux-i586.rpm 添加执行的权限

1.5. 执行命令 rpm -ivh j2sdk-1_4_2_04-linux-i586.rpm , 安装 j2sdk

1.6. 安装界面会出现授权协议，按 Enter 键接受，把 j2sd 安装在 /usr/java/j2sdk1.4.2_04

1.7. 设置环境变量，在 /etc/profile.d/ 目录下建立文件 java.sh，文件的内容如下：

```
#set java environment
export JAVA_HOME=/usr/java/j2sdk1.4.2_04
export CLASSPATH=.:${JAVA_HOME}/lib:${JAVA_HOME}/jre/lib:/usr/java/jdbc
export PATH=${JAVA_HOME}/bin:${JAVA_HOME}/jre/bin:$PATH
```

1.8. 执行命令 chmod 755 /etc/profile.d/java.sh，给 java.sh 分配权限

1.9. 在终端中分别执行命令 javac -help 和 java -version，如果看到有关的信息，则表示 j2sdk 已成功安装

2 接下来安装 tomcat，安装 tomcat 的步骤如下：

2.1 访问 <http://jakarta.apache.org/tomcat/index.html>，跟安装 sdk 一样，下载二进制版的 tomcat 如 jakarta-tomcat-5.5.10.tar.gz
解压到 /usr/local 目录： cd /usr/local
gzip -zxvf jakarta-tomcat-5.5.10.tar.gz

2.2 修改 \$tomcat/bin/startup.sh 和 shutdown.sh 文件

```
export JAVA_HOME=/usr/java/j2sdk1.4.2_04
export CATALINA_HOME=/usr/local/tomcat-5.5.10
export CLASSPATH=.:${JAVA_HOME}/lib:${JAVA_HOME}/jre/lib:/usr/java/jdbc
export PATH=${JAVA_HOME}/bin:${JAVA_HOME}/jre/bin:$PATH
```

2.3 执行 startup.sh

即可在 <http://localhost:8080/> 访问到 tomcat 的缺省页面

三、Apache Web 服务器

在发展初期，Apache 主要是一个基于 UNIX 系统的服务器，它的宗旨就是建成一个基于 UNIX 系统的、功能更强、效率更高并且速度更快的 WWW 服务器，这就决定了它是从其他的服务器演变而来的而且开放源代码。单独运行 Tomcat 的缺点：Tomcat 的支持软件比较少，Tomcat 内建 Web 服务器的支持软件比 Apache httpd Web 服务器少。Tomcat 的 Web 服务器功能少，相对于 Tomcat Web 服务器，Apache httpd 服务器有更完整的功能，没有 Apache httpd 快。Tomcat 服务器比 Apache httpd 慢，不过，它还在改进中，在提供静态页面内容上没有 Apache 快。

安装 Apache 服务器：

1. 下载软件包 apache_1.2.4.e.tar.gz，执行命令 tar xvzf apache_1.2.4.e.tar.gz

2. 如果使用的是 RedHat Linux 的话，也可以下载 apache_1.2.4.rpm 软件安装包，然后使用 rpm -ivh apache_1.2.4.rpm 命令安装

3. 如果想把 Apache 服务器充分利用起来的话，就一定要自己编译 Apache 定制其功能。下载包含 Apache 源代码的软件包 apache_1.2.4.tar.gz，大家自己试试吧，以后的正式版我就发出来。

四、Tomcat 与 Apache 的整合

1. 配置前的初始化工作

1.1 安装 j2sdk

运行 j2sdk 的自动安装包，安装 j2sdk1.4.2 到 C:\j2sdk\j2sdk1.4.2 目录下

1.2 安装 Apache 服务器

运行 Apache 的自动安装包，安装 Apache 到 D:\Apache\Apache2 目录下

1.3 安装 Tomcat 服务器

运行 Tomcat 的自动安装包，安装 Tomcat 到 D:\Apache\Tomcat5 目录下

1.4 安装 mod_jk

拷贝 mod_jk_2.0.47.dll 文件到 D:\Apache\Apache2\modules 文件夹下

1.5 准备工作目录和测试文件

在 D:盘新建目录 www，并创建一个基本的 index.jsp 文件，源码如下所示：

```
<%
out.println("<html>");
out.println("<body>");
out.println("Welcome");
out.println("</body>");
out.println("</html>");
%>
```

2. 配置 Apache 服务器

编辑 httpd.conf 文件，使用文本编辑器打开 httpd.conf 文件，在文件的最后添加如下内容：

```
# Using mod_jk2.dll to redirect dynamic calls to Tomcat
LoadModule jk_module modules/mod_jk_2.0.47.dll
JkWorkersFile "D:/Apache/Tomcat5/conf/workers.properties"
JkLogFile "D:/Apache/Tomcat5/logs/mod_jk2.log"
JkLogLevel info
JkMount /servlet/* ajp13
JkMount /*.jsp ajp13
```

```
Alias /web "D:/www/"
```

```
<Directory "D:/www">
Options Indexes MultiViews
AddOutputFilter Includes html
AllowOverride None
Order allow,deny
Allow from all
</Directory>
# 包含 Tomcat 中的 mod_jk.conf 文件
Include D:/Apache/Tomcat5/conf/auto/mod_jk.conf
```

3. 配置 Tomcat 服务器

3.1 建立 workers.properties 文件

在 D:/Apache/Tomcat5/conf/ 目录下新建文本文档并命名为 workers.properties，写入如下内容：

```
workers.tomcat_home=C:\Tomcat#让 mod_jk 模块知道 Tomcat
workers.java_home=C:\jdk#让 mod_jk 模块知道 j2sdk
ps=\
worker.list=ajp13#模块版本,现有 ajp14 了,不要修改
worker.ajp13.port=8009#工作端口,若没占用则不用修改
worker.ajp13.host=localhost#本机,若上面的 Apache 主机不为 localhost,作相应修改
worker.ajp13.type=ajp13#类型
worker.ajp13.lbfactor=1#代理数,不用修改
```

3.2 修改 server.xml 文件

找到如下的配置代码：

```
<!-- Tomcat Root Context -->
<!--
<Context path="" docBase="ROOT" debug="0">
-->
```

在其后面增加如下一行：

```
<Context path="/web" docBase="D:/www" debug="0" reloadable="true" ></Context>
```

3.3 修改 *jk2.properties* 文件

修改后内容如下：

```
## THIS FILE MAY BE OVERRIDEN AT RUNTIME. MAKE SURE TOMCAT IS STOPED
## WHEN YOU EDIT THE FILE.
```

```
## COMMENTS WILL BE _LOST_
## DOCUMENTATION OF THE FORMAT IN JkMain javadoc.
```

```
# Set the desired handler list
handler.list=apr,request,channelJni
#
# Override the default port for the socketChannel
# channelSocket.port=8019
# Default:
# channelUnix.file=${jkHome}/work/jk2.socket
# Just to check if the config is working
# shm.file=${jkHome}/work/jk2.shm
```

```
# In order to enable jni use any channelJni directive
# channelJni.disabled = 0
# And one of the following directives:
```

```
# apr.jniModeSo=/opt/apache2/modules/mod_jk2.so
```

```
# If set to inprocess the mod_jk2 will Register natives itself
# This will enable the starting of the Tomcat from mod_jk2
# apr.jniModeSo=inprocess
```

```
#request.tomcatAuthentication=true
request.tomcatAuthentication=false
```

4.运行测试程序

先启动 Tomcat 服务器，然后启动 Apache 服务器

打开 IE 浏览器，分别输入如下地址：

<http://127.0.0.1/web/index.jsp>

<http://127.0.0.1:8080/web/index.jsp>

如果出现的页面相同，均打印出

Welcome

说明配置成功

备注

能站在巨人的肩膀上，我觉得很光荣，本着学习的目的，不免有错误和疏漏，请大家批评指正。还有，正式版我会抓紧发布的 ^_^~~

结束