

Linux

环境编程图文指南

(配视频教程)

林世霖 钟锦辉 李建辉 著
粤嵌教育教材研发中心 审校

- 全书400多幅图表展现逻辑脉络，拒绝枯燥纯文字代码，贴心为你
- 200多篇实战源代码，一键编译运行，并有丰富注释一看就懂
- 配套超过4G教学视频，一律采用UGEE素描画板板书，看着舒服



含DVD光盘1张



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Linux 环境编程图文指南

(配视频教程)

林世霖 钟锦辉 李建辉 著
粤嵌教育教材研发中心 审校

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书从零开始,循序渐进地攻破 Linux 环境编程所遇到的各级关卡,以图文并茂的形式帮助读者理解各个概念。本书内容翔实,囊括了 Linux 系统操作细节,Shell 脚本编程精要,各种编程环境所需要解决的技术难点,以及在 Linux 环境下的 C 语言编程技术、并发编程技术和音/视频编程等核心内容。全书用 400 余幅图表帮助读者理解复杂概念,因此读者不需要具备任何计算机编程经验,在本书的指导下就能进入编程的世界,并能在阅读和实践中享受编程的乐趣。同时,本书配套完整的视频教程,给读者以最直观、最容易吸收知识的方式,融会贯通书中所有的知识点。不仅如此,读者还能够得到作者及其团队的在线技术支援和答疑。

本书通俗易懂,适合从事 Linux/UNIX 编程开发、嵌入式开发、C 环境开发的读者,尤其适合计算机相关专业的高职院校的学生,以及希望转向 IT 类就业方向的在职人士。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Linux 环境编程图文指南 / 林世霖, 钟锦辉, 李建辉著. —北京: 电子工业出版社, 2016.4

配视频教程

ISBN 978-7-121-28075-7

I. ①L… II. ①林… ②钟… ③李… III. ①Linux 操作系统—教材 IV. ①TP316.89

中国版本图书馆 CIP 数据核字 (2016) 第 011968 号

策划编辑: 李树林 lisl@phei.com.cn

责任编辑: 底波

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 32.75 字数: 838 千字

版 次: 2016 年 4 月第 1 版

印 次: 2016 年 4 月第 1 次印刷

印 数: 3000 册 定价: 99.00 元 (含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

本书定位 Linux 环境编程入门与提高，全书配送近百个教学视频，400 余幅案例图表，200 多篇源代码，力争做到图文并茂。作为粤嵌教育的专业教员，我和我的同事们都深刻地认识到，很多编程初入行的朋友成长曲线平缓，不是因为概念和原理有多复杂，而是很多教程和图书没有将原理用容易理解的图画表现出来，所谓一图顶万言，讲的就是这个道理。基于这样的认识，粤嵌教育教材研发中心的同事们几乎对每一个概念都力争用图画的形式来表现，因此本书的出版和面世也迟缓很多，但我们认为这是值得的。

本书面向的读者人群，是所有希望从事 Linux/UNIX 编程开发、嵌入式开发、C 环境开发的朋友，尤其适合计算机相关专业的高职院校的毕业生，以及希望转向 IT 类就业方向的在职人士，阅读本书不需要掌握任何专门的计算机技术和编程经验，但是对计算机的运行原理需要有一定认知。当然，学习过任何一门编程语言将使读者在阅读和学习本书的内容时更能稳操胜券。

本书的作者和审校同事都是长期从事培训教育行业的一线培训工作者，购买本书的同时实际上也加入了由广州粤嵌教育主导的 IT 技术学习圈子，可以登录 www.geconline.cn 了解更多资讯。

本书共分 6 章，按照从易到难的路径顺序讲述。

第 1 章着重介绍整个 Linux 的编程环境，包括如何安装 Linux 系统，以及如何使用 Shell 来操作用户的系统，本章还详细介绍了 Linux 下编程的三大必备技能，Shell 脚本编程、Makefile 语法和 GNU 开源开发套件 autotools 的详细使用方法。

第 2 章深度剖析 C 语言，大量使用图文方式解释内存机制，从根本上解决初学者对内存认识不到位的问题，具体而真实地掌握内存是学好编程的一大秘诀。另外，本章还介绍了 Linux 下的 C 语言的一些扩展增强语法。

第 3 章讲解数据组织结构，并且联系 Linux 内核使用实况详细剖析了传统链表、内核链表、栈和队列、二叉搜索树以及内核红黑树等高级数据结构，全章图文并茂，一目了然，对于这些纯算法也能确保读者学习愉悦，不枯燥。

第 4 章讲解 Linux 文件 I/O 编程，详述标准 I/O 和系统 I/O，图解包括触摸屏在内的特殊设备文件的操作，读者在学习完本章之后对 Linux 的文件管理、目录操作会有本质上的提升。

第 5 章全面介绍 Linux 并发编程中的核心技术，包括多进程、多线程、IPC、同步互斥等，全章同样图文并茂，确保每一个知识点都能在图画中得到解答。

第 6 章是 Linux 应用编程的高级部分，在前面章节的基础上着重介绍了跟 Linux 音/视频相关的概念和使用，详细剖析 ALSA 机制、framebuffer、V4L2 机制、SDL 和 FFmpeg 库的使用等，让读者可以编程实现在 Linux 系统和嵌入式系统中实现图片显示、声音录制、音乐播放、视频播放等内容。

作 者

2016 年 2 月

目 录

第 1 章 Linux 编程环境	1	1.5.8 实用 make 选项集锦	85
1.1 基本工具	1	1.6 GNU-autotools	86
1.1.1 免费大餐: Ubuntu	1	1.6.1 autotools 简介	86
1.1.2 桌面系统: gnome	6	1.6.2 文件组织	87
1.1.3 网络配置: 纯手工打造	6	1.6.3 configure.ac 编写规则	88
1.1.4 软件集散地: APT	8	第 2 章 深度 Linux-C	92
1.1.5 无敌板斧: vi	10	2.1 基本要素	92
1.1.6 开发圣经: man	13	2.1.1 Linux 下 C 代码规范	93
1.1.7 配置共享目录	15	2.1.2 基本数据类型	97
1.2 Shell 命令	17	2.1.3 运算符	108
1.2.1 概念扫盲	17	2.1.4 控制流	116
1.2.2 命令详解	19	2.2 函数	124
1.2.3 上古神器	38	2.2.1 函数初体验	125
1.3 Shell 脚本编程	45	2.2.2 函数调用内幕	128
1.3.1 开场白	45	2.2.3 递归思维及其实现	130
1.3.2 脚本格式	45	2.2.4 变参函数	133
1.3.3 变量	46	2.2.5 回调函数	137
1.3.4 特殊符号们	48	2.2.6 内联函数	140
1.3.5 字符串处理	50	2.3 数组与指针	142
1.3.6 测试语句	51	2.3.1 数组初阶	142
1.3.7 脚本语法单元	52	2.3.2 内存地址	144
1.4 编译器: GCC	55	2.3.3 指针初阶	145
1.4.1 简述	55	2.3.4 复杂指针定义	147
1.4.2 编译过程简介	55	2.3.5 指针运算	151
1.4.3 实用的编译选项	58	2.3.6 数组与指针	152
1.5 解剖 Makefile	59	2.3.7 复杂数组剖析	155
1.5.1 工程管理器 make	59	2.3.8 const 指针	158
1.5.2 概览性示例	60	2.3.9 char 指针和 char 数组	160
1.5.3 书写格式	60	2.4 内存管理	162
1.5.4 变量详解	62	2.4.1 进程内存布局	162
1.5.5 各种规则	71	2.4.2 堆 (Heap)	164
1.5.6 条件判断	75	2.5 组合数据类型	167
1.5.7 函数	77		

2.5.1	结构体	167	4.4.2	TSLIB 库详解	333
2.5.2	共用体	171	4.4.3	划屏算法	338
2.5.3	枚举	172	第 5 章 Linux 进程线程		345
2.6	高级议题	173	5.1	Linux 进程入门	345
2.6.1	工程代码组织	173	5.1.1	进程概念	345
2.6.2	头文件	175	5.1.2	进程组织方式	346
2.6.3	宏 (macro)	176	5.2	进程的“生老病死”	348
2.6.4	条件编译	182	5.2.1	进程状态	348
2.6.5	复杂声明	184	5.2.2	相关重要 API	350
2.6.6	attribute 机制	185	5.3	进程的语言	358
第 3 章 Linux 的数据组织		188	5.3.1	管道	358
3.1	无所不在的链表	188	5.3.2	信号	363
3.1.1	开场白	188	5.3.3	system-V IPC 简介	380
3.1.2	单向链表	190	5.3.4	消息队列 (MSG)	381
3.1.3	单向循环链表	198	5.3.5	共享内存 (SHM)	387
3.1.4	双向循环链表	200	5.3.6	信号量 (SEM)	392
3.1.5	Linux 内核链表	210	5.4	Linux 线程入门	400
3.2	线性表变异体	227	5.4.1	线程基本概念	400
3.2.1	堆叠的盘子: 栈	227	5.4.2	线程 API 及特点	401
3.2.2	文明的社会: 队列	236	5.5	线程安全	410
3.3	小白慎入: 非线性结构	243	5.5.1	POSIX 信号量	410
3.3.1	基本概念	243	5.5.2	互斥锁与读写锁	415
3.3.2	玩转 BST	247	5.5.3	条件变量	418
3.3.3	各种的遍历算法	260	5.5.4	可重入函数	421
3.3.4	自平衡 AVL 树	263	5.6	线程池	422
3.3.5	自平衡 Linux 红黑树	273	5.6.1	实现原理	422
第 4 章 I/O 编程技术		289	5.6.2	接口设计	423
4.1	一切皆文件	289	5.6.3	实现源码	425
4.1.1	文件的概念	289	第 6 章 Linux 音频、视频编程		433
4.1.2	各类文件	290	6.1	基本背景	433
4.2	文件操作	290	6.2	Linux 音频	433
4.2.1	系统 I/O	291	6.2.1	音频概念	433
4.2.2	标准 I/O	306	6.2.2	标准音频接口 ALSA	436
4.2.3	文件属性	320	6.3	Linux 视频输出	450
4.3	目录检索	327	6.3.1	基本概念	450
4.3.1	基本概念	327	6.3.2	framebuffer	452
4.3.2	相关 API	328	6.3.3	在 LCD 上画图	462
4.4	触控屏应用接口	330	6.3.4	效果算法	469
4.4.1	输入子系统简介	330	6.4	Linux 视频输入	478

6.4.1	V4L2 简介	478	6.5.4	音频子系统	494
6.4.2	V4L2 视频采集流程	478	6.5.5	事件子系统	498
6.4.3	V4L2 核心命令字和结构体	481	6.5.6	处理 YUV 视频源	502
6.4.4	编码格式和媒体流	484	6.6	音/视频编解码库 FFmpeg	504
6.5	多媒体开发库 SDL	489	6.6.1	FFmpeg 简介	504
6.5.1	SDL 简介	489	6.6.2	核心结构体与常用 API	505
6.5.2	编译和移植	489	6.6.3	与 SDL 结合实现简单的播放器	511
6.5.3	视频子系统	490			

Linux 编程环境

1.1 基本工具

1.1.1 免费大餐：Ubuntu

工欲善其事，必先有把刀，我们首要的任务是搭建一个完整的 Linux 编程环境，我们选用 Ubuntu，建议选择最新的 LTS 版本，即长期支持版，在撰写本书时最新版的 LTS 是 Ubuntu-14.04。

第一步，下载一个虚拟平台 VMware workstation。其官网是：

<http://www.VMware.com>

第二步，安装 VMware，装完之后的样子如图 1-1 所示。

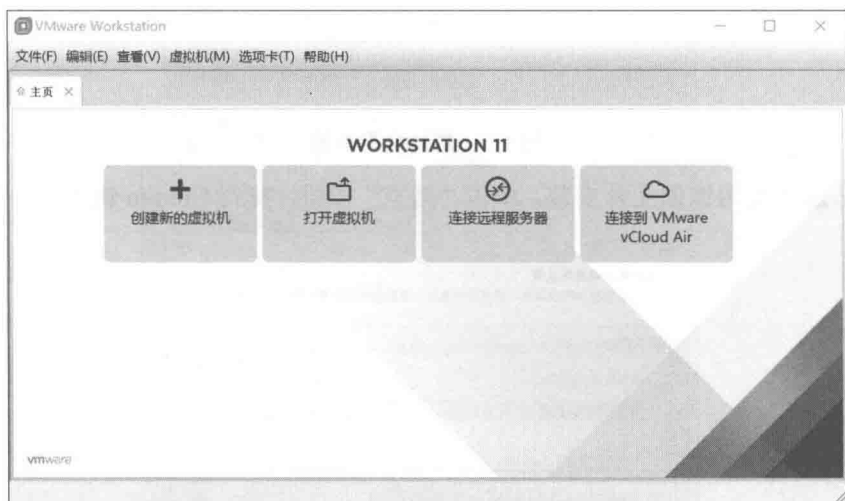


图 1-1 VMware workstation 11

第三步，下载当前最新的 LTS 版本的 Ubuntu-14.04，地址是：

<http://www.ubuntu.com/download/desktop>

第四步，下面开始做一台虚拟机，然后把 Ubuntu 镜像文件装进去。在 VMware 的欢迎页面中单击“创建新的虚拟机”按钮，创建一台虚拟机，如图 1-2 所示。



图 1-2 选择新建一台虚拟机

第五步，在弹出的对话框中选择推荐的“典型”选项，如图 1-3 所示。

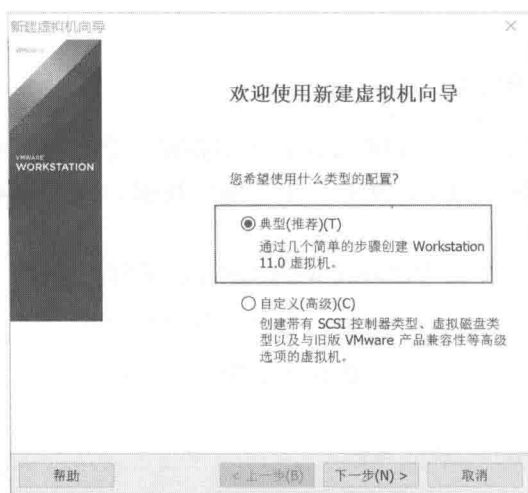


图 1-3 选择典型安装

第六步，选择采用镜像文件安装，单击“浏览”按钮，找到 Ubuntu 镜像，如图 1-4 所示。

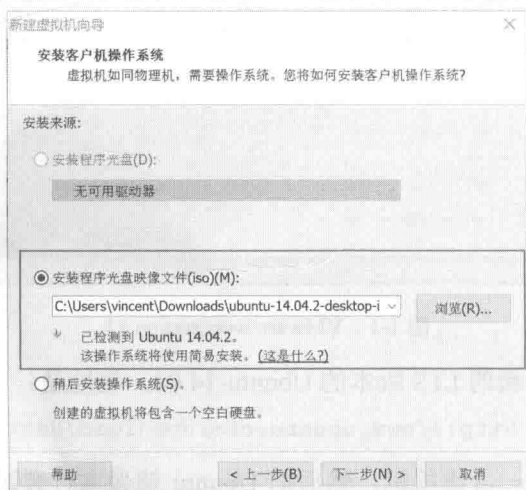


图 1-4 选择 Ubuntu 镜像文件（iso 文件）

第七步，写入用户名和密码（这是将来安装好之后登录 Ubuntu 系统用的），如图 1-5 所示。



图 1-5 填写用户名和密码

说明：这个用户将会在安装 Ubuntu 时作为一个普通用户（相对于超级用户）被自动建立，而且这个普通用户还在一个称为 `sudoer` 的文件中被设置了，换句话讲，这个普通用户可以执行 `sudo` 命令从而临时获得超级用户的权限。密码则是超级用户的密码。当普通用户执行 `sudo` 想要临时获得超级用户的权限时，也需要输入这个密码。

第八步：给新虚拟机起个名字，并且选择存放路径。

第九步：选择系统磁盘大小和分配方式，如图 1-6 所示。

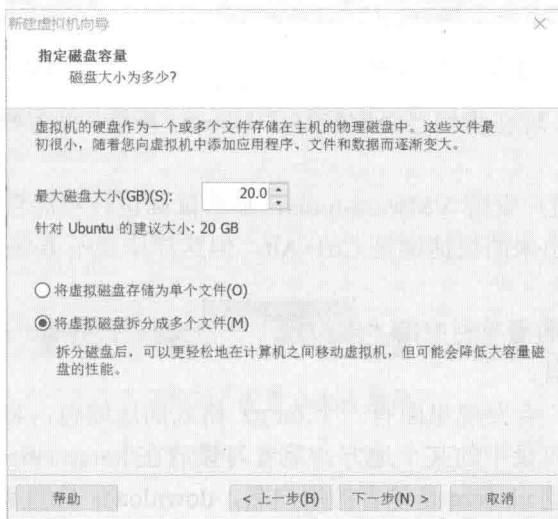


图 1-6 选择磁盘

说明：

- (1) 磁盘大小：完全根据用户的需求，但用户至少要分配 5GB 的磁盘给 Ubuntu 的根目录。
- (2) 分配方式：完全根据用户的需求，可以选择单一的文件当做虚拟的磁盘来使用 (Single File)，也可以选择把虚拟磁盘分开成多个文件存放 (Multiple Files)。

单一文件存放用户的虚拟磁盘，需要一次性占用用户所指定的物理磁盘的大小（如20GB空间），其好处是速度较快。多文件存放用户的虚拟磁盘则相反，无须一开始就占用指定的磁盘大小，而是随着使用的情况自动地为虚拟机增加空间，其代价是降低了 Ubuntu 系统读取跨虚拟磁盘文件的速度。

现在，我们已经配置好了一台虚拟计算机了，但是还没给这台“裸机”安装操作系统，接下来将用户的 Ubuntu 镜像装上去：把“创建后开启此虚拟机”复选框选上，单击“完成”按钮。此时用户可以稍稍休息，静候佳音。

大约几分钟后，我们会看到类似如图 1-7 所示的画面。

当不想耗费时间让 Ubuntu 从网络下载诸如语言包时，建议单击“skip”按钮，或者关闭计算机的网络。

耐心等待几分钟后，计算机会自动重启，重启后将出现如图 1-8 所示的欢迎画面，这就是我们的劳动成果！第一次启动可能会比较慢。

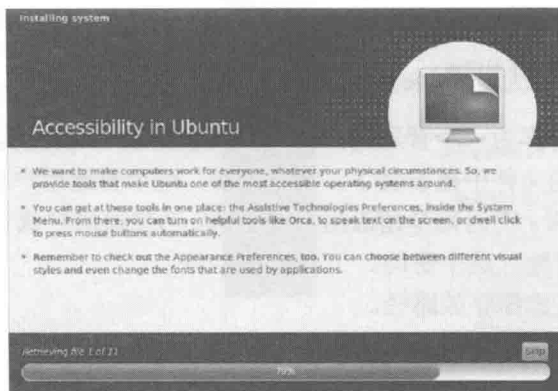


图 1-7 正在安装 Ubuntu

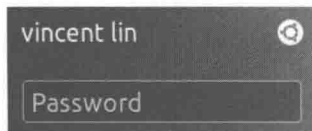


图 1-8 Ubuntu 登录界面

事情就这样完成了！现在我们已经安装好了 Ubuntu 系统，只需要输入在第七步设置的密码，即可使用它。

如果 Ubuntu 没帮用户安装 VMware-tools，那么鼠标也许不能自动在虚拟机和 Windows 之间自由切换，让鼠标出来的快捷键是 Ctrl+Alt，但这样毕竟不方便，安装 VMware-tools 的方法如下。

(1) 单击 VMware 的菜单栏中的“虚拟机”→“安装 VMware tools...”，这时会在虚拟机的桌面上出现一张光盘。

(2) 双击后打开它，会发现里面有一个.tar.gz 格式的压缩包，将它复制到用户平常下载文件的目录中（如家目录中的某个地方，笔者习惯放在/home/vincent/downloads 里面，请注意读者不要照抄，这里 vincent 是笔者的用户名，downloads 是笔者建的一个专门用来存放下载文件的地方，读者要根据自己的具体情况来操作），然后单击右键解压。

(3) 然后执行 VMware-install.pl 文件。因为安装程序需要管理员权限，因此我们需要在命令行模式下用 sudo 执行它，具体操作步骤如下。

① 按 Ctrl+Alt+t 组合键打开一个终端。

② 将工作目录转到 VMware-install.pl 所在的地方：如在计算机里就要敲入如下命令：`cd /home/vincent/downloads`（读者根据自己的情况改一下路径）。

③ 用管理员权限执行它：`sudo ./VMware-install.pl`（随后需要输入管理员密码，注意，在命令行中输入的密码是不回显的）。

④ 然后开始安装，安装过程中遇到任何提问直接按回车键即可。

提示

一般而言，在安装客户机 Ubuntu 时，VMware 会自动为用户安装 VMware-tools 增强包，但是有时候会安装失败，症状是：鼠标无法从虚拟机移动到 Windows 当中，或者共享文件夹无法使用等，这时需要重复以上步骤，重新安装 VMware-tools。

另外，出于安全考虑，POSIX 阵营的 OS（如 Linux）一般不推荐直接以 root 登录并操作，这会被视为一种不专业且粗鲁的行为。在默认状态下，Ubuntu 只能以普通用户登录，如果登录之后需要 root 权限，则使用 `sudo` 来临时获取，但不是每一个普通用户都可以使用 `sudo`，要使得一个普通用户能够通过 `sudo` 来获取管理员权限，则必须在配置文件 `/etc/sudoers` 中做相应的配置。

在安装完系统之后，系统中只有一个普通用户，就是上面我们输入的那个用户，这个用户是可以用 `sudo` 来临时获得管理员权限的，但是如果再新添加一个用户，如 `foo`，那么这个 `foo` 默认情况下是无法使用 `sudo` 来临时获得管理员权限的。接下来，通过配置 `/etc/sudoers` 来使 `foo` 可以使用 `sudo`，步骤如下。

首先，在命令行中输入 `sudo visudo`，得到如图 1-9 所示界面。

```

vincent@ubuntu: ~
GNU nano 2.2.4 File: /etc/sudoers.tmp Modified
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
# Host alias specification
# User alias specification
# Cmnd alias specification
#User privilege specification
root    ALL=(ALL) ALL
foo     ALL=(ALL) ALL
  
```

图 1-9 配置 sudoer 清单

然后，在 `root ALL=(ALL) ALL` 下面手工添加一行：

```
foo ALL=(ALL) ALL
```

这一行的作用是将用户 `foo` 添加到 `sudoer` 中。第一个 `ALL` 代表所有的主机名，等号后面的 `(ALL)` 代表 `foo` 可以以任意的用户运行后面的命令，最后一个 `ALL` 指的是 `foo` 可以用 `sudo` 运行所有 Shell 命令。然后按提示保存（`Ctrl + O` 组合键）并退出（`Ctrl + X` 组合键）。另外，使 `foo` 可以使用 `sudo` 命令还有一个更加简单的方法：第一，使用 `su` 命令切换到管理员账户；第二，运行命令 `usermod foo -a -G sudo`；第三，运行 `su foo` 命令切换到 `foo` 账户，此时 `foo` 就可以使用 `sudo` 来获取管理员权限了。所有繁杂的事情完毕，登录后的 Ubuntu 如图 1-10 所示。



图 1-10 Ubuntu-14.04 初始界面

为了使用方便，我们需要再做如下几件事情。

第一，设置 Ubuntu 的桌面系统为经典 GNOME。

第二，配置 Ubuntu，使之可以上网。

第三，升级 vi 并稍做配置，使之更好用。

第四，安装 man 帮助文档，使用户在写程序或脚本时，随时获得权威帮助。

第五，设置宿主机（Windows）和客户机（Ubuntu）的共享目录，使用户的文件可以在两个 OS（操作系统）中自由穿行。下面逐一进行介绍。

1.1.2 桌面系统：gnome

自从 Ubuntu-11.04 开始，它使用了所谓 unity 的桌面系统，就是我们看到的图 1-10，系统的主菜单不见了，取而代之的是一条左边栏，网上诸多网友都在吐槽这个 unity 界面，因为实在太难用了！想恢复原先的经典 GNOME 界面只需要如下 3 步。

第一，下载安装 gnome 桌面环境，命令如下。

```
vincent@ubuntu:~$ sudo apt-get install gnome-session-fallback
```

第二，在登录界面单击桌面环境按钮，如图 1-11 所示。

第三，选择经典 GNOME 桌面，如图 1-12 所示。

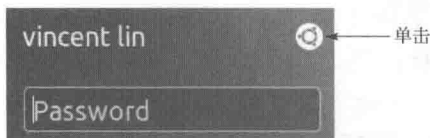


图 1-11 单击桌面环境选择按钮

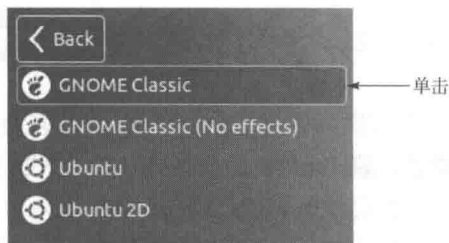


图 1-12 选择 GNOME 桌面系统

1.1.3 网络配置：纯手工打造

(1) 按 Ctrl+Alt+T 组合键或者双击桌面的“Terminal”调出伪终端，输入以下命令。

```
vincent@ubuntu:~$ sudo gedit /etc/network/interfaces
```

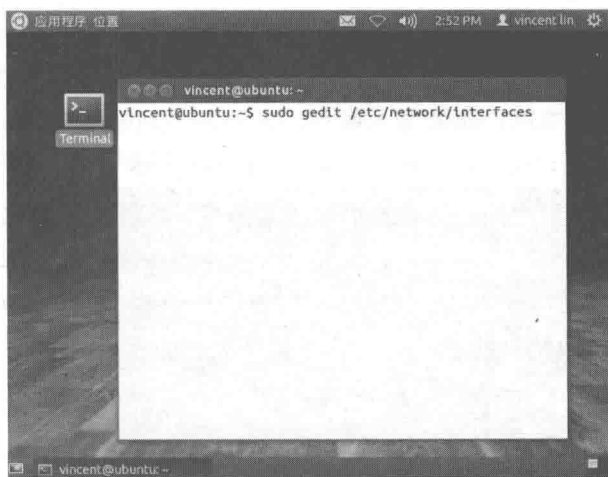


图 1-13 Linux 伪终端

照提示说明，输入用户的密码，当发现输入密码时终端“没反应”，请不要着急！看不见是因为密码是不回显的。没什么意外的话，`interfaces` 里面应该有以下两行信息：

```
auto lo
iface lo inet loopback
```

以上两行信息定义并开启了系统的所谓本地回环接口 `lo`，`lo` 是 `local loopback` 的缩写，是一个虚拟的网络设备，它用来接收和发送那些只在本机流转的数据包，而不流经真正的网络接口。例如，在同一台主机运行的网络服务器和客户端，虽然它们的数据会被打包成带 IP 地址和端口的网络封包，但系统发现它们其实是“一家子”，那么这个网络封包就不会经过网卡走出家门，而是仅仅从本地回环 `lo` 中兜了一圈。就像你刚学会写字的女儿给她妈妈写了封信，她爸爸一转手就交给了她妈，而不是跑去邮局。

接着，请在文件 `interfaces` 末尾添加以下信息：

```
auto eth0
iface eth0 inet static
address x.x.x.x (用户的真实环境下的 IP 地址)
gateway y.y.y.y (用户的真实环境下的路由器地址)
netmask z.z.z.z (用户的真实环境下的子网掩码)
```

其中：

`auto eth0` 代表系统自动识别且启动第 0 个以太网卡。

`static` 代表设置固定 IP（如果要让系统自动获取 IP 地址，请将 `static` 改成 `dhcp`，同时可以删除下面三行）。

`address` 是 Ubuntu 的 IP 地址，`gateway` 是路由器的 IP 地址。

`netmask` 是子网掩码。如果用户的网络是 C 类子网，那么子网掩码就是 `255.255.255.0`；如果用户的网络是 B 类子网，那么子网掩码就是 `255.255.0.0`；如果不知道用户的网络具体情况，可以在 Windows 的命令行中敲入 `ipconfig /all` 来查看。

(2) 配置 DNS 服务器。编辑 `/etc/resolv.conf`，在文件的最末一行添加以下信息：

```
nameserver 202.96.134.133
nameserver 202.96.128.143
```

注意，以上 DNS 的 IP 是广东珠三角地区的其中两个服务器地址，用户要根据自己所在地区填写离自己比较近的 DNS 地址，详情咨询百度。

(3) 为主机配置网关地址。在终端下输入：

```
vincent@ubuntu:~$ sudo route add default gw y.y.y.y
```

(4) 重新加载网络配置文件，并重新启动网络服务。在终端下输入以下两行命令：

```
vincent@ubuntu:~$ sudo /etc/init.d/networking force-reload
vincent@ubuntu:~$ sudo /etc/init.d/networking restart
```

(5) 如果有必要，重启系统：

```
vincent@ubuntu:~$ sudo shutdown -r now
```

1.1.4 软件集散地：APT

Ubuntu 继承了 Debian 系统一个非常优秀的特性：使用 APT 软件管理器来管理所有的软件。在此之前，Linux 系统的软件安装一直被人诟病——用户需要自己解决软件与库之间的依赖关系，通常这是一个冗长乏味的分析过程。APT 软件管理器旨在改变这一现状。

APT 软件管理原理图如图 1-14 所示。

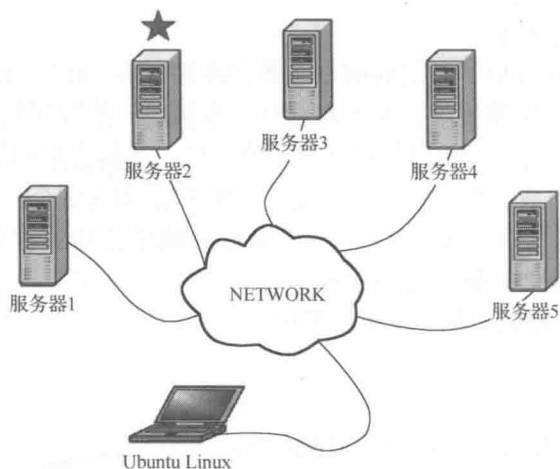


图 1-14 APT 软件管理拓扑结构

APT 将各个版本的 Ubuntu 所用到的软件，分门别类地放在世界各地的服务器中（图 1-14 所示的服务器 1、服务器 2……），我们下载安装软件，只需要联网指定一个服务器即可（如图 1-14 所示标注了星号的服务器 2），APT 将会帮助用户解决软件的互相依赖问题。

使用 APT，只需要以下两步。

第一步，指定一个服务器。原则很简单，指定一个用户觉得速度快的即可，如图 1-15 所示。

在“Edit”菜单找到“Software Sources”选项，调出软件源列表对话框，单击“Download from”下拉框，如图 1-16 所示。

找到自己所在的地区或国家，并选择距离最近的服务器站点，如图 1-17 所示。

第二步，需要将选定站点的软件列表信息，全部下载到本地（/var/lib/apt/lists），如软件名称、依赖关系、版本号等，这个过程称为 update，使用如下一条命令：

```
vincent@ubuntu:~$ sudo apt-get update
```

这样，我们就可以使用 APT 提供的相关命令来下载安装软件了。例如一个称为 xxx 的软件，下载安装和卸载的命令分别是：

```
vincent@ubuntu:~$ sudo apt-get install xxx
vincent@ubuntu:~$ sudo apt-get remove xxx
```



图 1-15 打开 Ubuntu 软件中心



图 1-16 打开 Ubuntu 软件管理标签

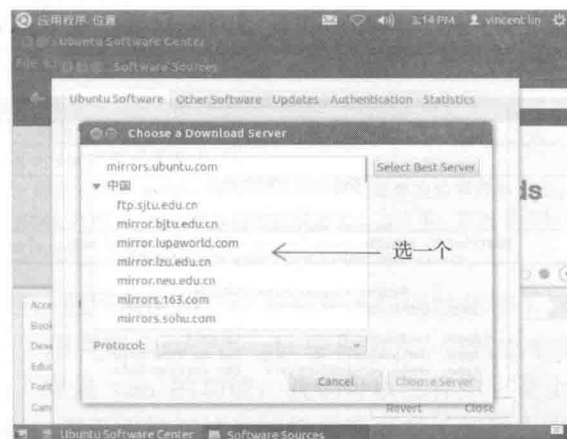


图 1-17 选择一个合适的软件源服务器

1.1.5 无敌板斧：vi

vi 是一款性感的字符界面编辑器，属于鼻祖级编辑神器。读者也许会认为，vi 太古老了，现代图形编辑器多如牛毛，各种“所见即所得”IDE 更是让编程如虎添翼，何必死磕这个老古董呢？答案是：想象某天，我们空降到一个南太平洋小岛的原始丛林中，手无寸铁，你是想要一台高级微波炉，还是更想要一把普通的斧头？微波炉显然是毫无用处的，因为你连插线板都找不到。IDE 的便捷性是需要代价的，这就是后台图形引擎的支持，假设我们的系统连图形库都没有，那是无法使用 IDE 的，而我们需要的仅仅是编辑，而不是花里胡哨的各种按钮和菜单，此时 vi 就是手头那把板斧，它不需要插电。

Ubuntu 为了减小安装文件的尺寸，默认安装了 vi 的原始版本，我们需要下载安装它的升级版本：强劲的 vim！利用刚刚配好的网络和软件源，输入以下命令：

```
vincent@ubuntu:~$ sudo apt-get install vim
```

字符界面的编辑器，跟我们熟悉的 Windows 的记事本有什么区别呢？读者会发现记事本是这样的，如图 1-18 所示。

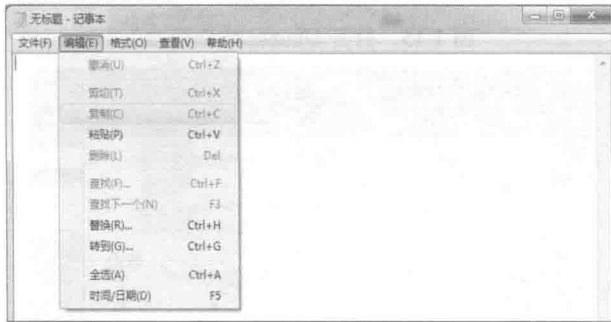


图 1-18 图形界面编辑器

在记事本中，我们要对文件执行的动作，如图 1-18 所示的撤销、剪切、复制、粘贴等，都可以通过单击菜单中相应的菜单项（或者对应的快捷键）来实现，不能指望在空白区输入“保存”两个字就保存了文件，也不能大吼一声“保存”来实现，相反，只需要单击菜单栏中的“保存”选项或者按下 Ctrl+S 组合键就可以了。

可惜，vim 就是一款没有菜单及其相应快捷键的编辑器，它只有键盘可用，在终端中输入 vim，启动 vim 程序，来一起看下它的真容，如图 1-19 所示。

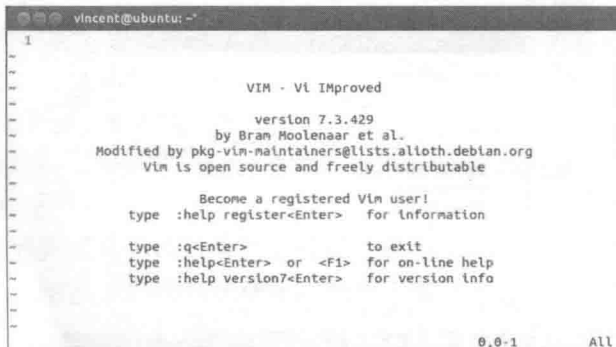


图 1-19 字符界面编辑器