Debian GNU/Linux 安装手册

April 23, 2025

Debian GNU/Linux 安装手册

版权 © 2004 –2023 Debian 安装程序团队

本手册属于自由软件，您可以在 GNU 通用公共授权（GPL）下重新发布和（或）修改它。请在附录 F 阅读该许可证的内容。

本手册的构建版本：20250418+rebuild 。

Contents

1 欢迎使用 Debian 1
1.1 什么是 Debian？ 1
1.2 什么是 GNU/Linux？ 1
1.3 什么是 Debian GNU/Linux? 2
1.4 什么是 Debian 安装程序？ 2
1.5 获得 Debian 3
1.6 获得本文档最新版本 3
1.7 文档的组织 3
1.8 关于版权与软件协议 3

2 系统需求 5
2.1 支持的硬件 5
2.1.1 支持的体系 5
2.1.2 zSeries 和 System z 机器类型 5
2.1.3 PAV 和 HyperPAV 5
2.1.4 多处理器 6
2.1.5 网络连接硬件 6
2.1.6 外围设备与其他硬件 6
2.2 安装介质 6
2.2.1 网络 6
2.2.2 硬盘 6
2.2.3 Un*x 或 GNU 系统 6
2.2.4 受支持的存储系统 6
2.3 内存与磁盘空间的需求 6

3 开始安装 Debian GNU/Linux 之前 7
3.1 安装概述 7
3.2 记得备份您所有的数据！ 8
3.3 一些有用的信息 8
3.3.1 文档 8
3.3.1.1 安装手册 8
3.3.1.2 硬件文档 8
3.3.1.3 S/390 Hardware References 8
3.3.2 提供硬件信息的资源 8
3.3.3 硬件兼容性 9
3.3.3.1 使用 Live-System 测试硬件兼容性 9
3.3.4 网络设置 10
3.4 满足最低的硬件要求 10
3.5 为多重启动系统事先分区 11
3.6 安装前的硬件和操作系统的相关设置 11
3.6.1 BIOS 设置 11
3.6.2 本地化 (Native) 和 LPAR 安装 11
3.6.3 作为一个 VM guest 安装 11
3.6.4 建立安装服务器 12

4 获取系统安装介质 13
4.1 官方的 Debian GNU/Linux 安装映像 13
4.2 从 Debian 镜像服务器下载文件 13
4.2.1 在哪里能找到安装文件 13
4.3 自动化安装 13
4.3.1 使用 Debian 安装程序进行自动安装 13
4.4 验证安装文件的完整性 14

5 引导安装系统 15
5.1 在 S/390 上引导安装程序 . . . 15
5.1.1 S/390 的限制 . . . 15
5.1.2 S/390 引导参数 . . . 15
5.2 引导参数 . . . 15
5.2.1 Debian 安装程序的参数 . . . 15
5.2.2 使用引导参数来回答问题 . . . 17
5.2.3 将参数传递给内核模块 . . . 17
5.2.4 内核模块黑名单 . . . 18
5.3 安装过程中的故障修复 . . . 18
5.3.1 引导配置 . . . 18
5.3.2 内核起始消息的注解 . . . 18
5.3.3 报告安装问题 . . . 18
5.3.4 提交安装报告 . . . 18

6 使用 Debian 安装程序 20
6.1 安装程序如何工作 . . . 20
6.2 组件介绍 . . . 20
6.3 使用单独的组件 . . . 21
6.3.1 启动 Debian 安装程序与配置硬件 . . . 22
6.3.1.1 检测可用内存/小内存模式 . . . 22
6.3.1.2 选择地区选项 . . . 22
6.3.1.3 选择键盘 . . . 23
6.3.1.4 寻找 Debian 安装程序 ISO 映像 . . . 23
6.3.1.5 配置网络 . . . 23
6.3.1.5.1 自动网络配置 . . . 23
6.3.1.5.2 手动网络配置 . . . 23
6.3.1.5.3 IPv4 和 IPv6 . . . 24
6.3.2 设置用户和密码 . . . 24
6.3.2.1 设置 root 密码 . . . 24
6.3.2.2 创建一个普通用户 . . . 24
6.3.3 配置时钟和时区 . . . 24
6.3.4 分区与选择挂载点 . . . 25
6.3.4.1 支持的分区选项 . . . 25
6.3.4.2 导引式分区 . . . 25
6.3.4.3 手动分区 . . . 27
6.3.4.4 配置多磁盘设备 (软 RAID) . . . 27
6.3.4.5 配置逻辑卷管理 (LVM) . . . 29
6.3.4.6 配置加密卷 . . . 30
6.3.5 安装基本系统 . . . 31
6.3.6 安装额外的软件 . . . 31
6.3.6.1 配置 apt . . . 32
6.3.6.1.1 从多个 DVD 映像来安装 . . . 32
6.3.6.1.2 使用网络镜像 . . . 32
6.3.6.1.3 选择网络镜像 . . . 33
6.3.6.2 选择和安装软件 . . . 33
6.3.7 创建启动系统 . . . 34
6.3.7.1 检测其他的操作系统 . . . 34
6.3.7.2 不使用 boot loader 继续进行 . . . 34
6.3.8 完成安装 . . . 34
6.3.8.1 重新启动系统 . . . 34
6.3.9 故障处理 . . . 34
6.3.9.1 保存安装记录 . . . 34
6.3.9.2 使用 Shell 查看记录 . . . 35
6.3.10 通过网络控制台（network-console）安装 . . . 35
6.4 个性化配置 . . . 36
6.4.1 安装替代的初始化系统 . . . 36

7 引导进入新的 Debian 系统 37
7.1 关键时刻 . . . 37
7.2 挂载加密的卷 . . . 37
7.2.1 故障处理 . . . 37
7.3 登录 . . . 38

8 下一步该干什么 39
8.1 关闭系统 . . . 39
8.2 让自己适应 Debian . . . 39
8.2.1 Debian 软件包系统 . . . 39
8.2.2 用于 Debian 的额外软件 . . . 39
8.2.3 应用程序版本管理 . . . 40
8.2.4 定时任务管理 . . . 40
8.3 更多信息 . . . 40
8.4 让您的系统使用 E-Mail . . . 40
8.4.1 默认的 E-Mail 配置 . . . 40
8.4.2 发送邮件到系统之外 . . . 41
8.4.3 配置 Exim4 MTA . . . 41
8.5 编译新内核 . . . 42
8.6 修复损坏的系统 . . . 42

A 安装指南 43
A.1 前言 . . . 43
A.2 安装套件的引导 . . . 43
A.2.1 光盘 . . . 43
A.2.2 从网络启动 . . . 43
A.2.3 从硬盘启动 . . . 43
A.3 安装 . . . 44
A.4 请给我们发一份安装报告 . . . 44
A.5 末了… . . . 44

B 使用预置自动进行安装 45
B.1 介绍 . . . 45
B.1.1 预置方法 . . . 45
B.1.2 限制 . . . 46
B.2 使用预置 . . . 46
B.2.1 加载预置文件 . . . 46
B.2.2 使用引导参数预置问题的答案 . . . 46
B.2.3 auto 模式 . . . 47
B.2.4 预置中有用的别名 . . . 48
B.2.5 引导提示预置的例子 . . . 49
B.2.6 使用 DHCP 服务器指定预置文件 . . . 49
B.3 创建预置文件 . . . 49
B.4 预置文件的内容 (trixie) . . . 50
B.4.1 本地化 . . . 50
B.4.2 语音合成 . . . 51
B.4.3 网络设置 . . . 51
B.4.4 网络控制台 . . . 52
B.4.5 镜像设置 . . . 53
B.4.6 帐号设置 . . . 53
B.4.7 时钟与时区设置 . . . 54
B.4.8 分区 . . . 54
B.4.8.1 分区示例 . . . 54
B.4.8.2 使用 RAID 分区 . . . 56
B.4.8.3 分区挂载控制 . . . 57
B.4.9 基本系统安装 . . . 57
B.4.10 设置 apt . . . 57
B.4.11 选择软件包 . . . 58
B.4.12 完成安装 . . . 59

B.4.13 预置其他的软件包 . . . 59
B.5 高级选项 . . . 60
B.5.1 安装过程中运行用户命令 . . . 60
B.5.2 使用预置修改默认值 . . . 60
B.5.3 预置文件链 . . . 60

C 为 Debian 准备分区 62
C.1 决定 Debian 分区以及大小 . . . 62
C.2 目录树 . . . 62
C.3 推荐的分区方案 . . . 63
C.4 Linux 里面的设备名称 . . . 63
C.5 Debian 分区程序 . . . 64

D 其他 65
D.1 Linux 里的设备 . . . 65
D.2 各软件集所需的磁盘空间 . . . 65
D.3 通过 Unix/Linux 系统来安装 Debian GNU/Linux . . . 66
D.3.1 准备开始 . . . 66
D.3.2 安装 debootstrap . . . 67
D.3.3 运行 debootstrap . . . 67
D.3.4 基本系统的配置 . . . 67
D.3.4.1 创建设备文件 . . . 68
D.3.4.2 分区的挂载 . . . 68
D.3.4.3 设置时区 . . . 69
D.3.4.4 网络的配置 . . . 69
D.3.4.5 配置 Apt . . . 70
D.3.4.6 本地化和键盘的配置 . . . 70
D.3.5 内核的安装 . . . 70
D.3.6 Boot Loader 的设置 . . . 71
D.3.7 远程访问：安装 SSH 并设置密码 . . . 71
D.3.8 最后一笔 . . . 71

E 文案 72
E.1 关于本文档 . . . 72
E.2 为本文档做出贡献 . . . 72
E.3 主要贡献 . . . 72
E.4 商标确认 . . . 73

F GNU 通用公共授权 74
F.1 导言 . . . 74
F.2 GNU 通用公共授权 . . . 75
F.3 您的新程序该如何采用这些条款 . . . 76

List of Tables

3 开始安装 Debian GNU/Linux 之前
3.1 有助于安装的硬件信息 9
3.2 推荐的最低系统配置 10

Abstract

本文档包含 Debian GNU/Linux 13 系统（代号“trixie”）在 S/390（“s390x”）硬件架构下的安装指引。它还包括关于如何使您的新 Debian 系统效用最大的信息，以及指向更多信息的链接。

如果您想参与翻译和校对，或对翻译项目有什么建议，请发邮件到 Debian 简体中文邮件列表：debian-chinese-gb@lists.debian.org。

在 s390x 上安装 Debian GNU/Linux 13

我们非常高兴地看到您决定试试 Debian，并确信您将发现 Debian 的 GNU/Linux 发行版是独一无二的。Debian GNU/Linux 将带给您来自世界各地的高质量的自由软件，并将它们整合成一个紧密的整体。您会发现整合后的结果一定会大于将各部分简单地相加。

我们也知道大多数用户都希望能够不用阅读本手册而直接安装 Debian，而 Debian 安装程序的设计意图也就是尽力实现这个想法。如果您现在的确没有时间来阅读整个安装手册，我们推荐您看看安装指南一文。该文包含了基本安装过程的简要介绍，以及当处理运行错误或需要较专业内容时查询本手册的链接。您可以在附录 A 阅读此文。

正如前面所提到的，我们还是希望您能抽出时间来阅读本手册的大部分内容。这通常都会使您的安装体验变得更详尽、更成功。

Chapter 1

欢迎使用 Debian

本章将简要介绍 Debian 计划和 Debian GNU/Linux。如果您已经对 Debian 计划的历史和 Debian GNU/Linux 发行版有一定的了解，请略过本章。

1.1 什么是 Debian？

Debian 是一个致力于自由软件开发并宣扬自由软件基金会之理念的自愿者组织。Debian 计划创建于 1993 年。当时，Ian Murdock 发出一份公开信，邀请软件开发者们参与构建一个基于较新的 Linux 内核的完整而紧密的软件发行版。经过多年的成长，那群由 自由软件基金会 资助并受 GNU 哲理影响的爱好者已经演变为一个拥有大约 1000 位 Debian 开发人员的组织。

Debian 开发人员所做的工作包括有：Web 和 FTP 站点管理、图形设计、软件许可协议的法律分析、编写文档，当然，还有维护软件包。

为了传递我们的理念并吸引那些与 Debian 有着相同信仰的开发人员，Debian 计划发表了众多文档，以阐明我们的价值观和成为 Debian 开发人员的意义：

- Debian 社群契约 是一份 Debian 对自由软件社群的承诺声明。任何同意遵守此社群契约的人都可以成为 维护人员。任何维护人员都能将新软件引入 Debian —条件是此软件必须满足我们对“自由”的标准要求，并且软件包必须符合我们的质量水准。
- Debian 自由软件准则 (DFSG) 是一份清晰简洁的有关 Debian 的自由软件标准的声明。在自由软件运动中，DFSG 是一份非常有影响力的文档，并且以其为基础衍生出了 开放源代码定义。
- Debian 政策文档是一份有关 Debian 计划质量标准的规格说明书。

Debian 开发人员还参与了许多其它计划。有些是专注于 Debian 的，还有些则是面向 Linux 社区。例如：

- Filesystem Hierarchy Standard (FHS) 则是朝 Linux 文件系统布局标准化方向的一种努力。FHS 将使软件开发者能集中精力设计程序，而不需要担心软件包在不同的 GNU/Linux 发行版的安装。
- Debian Jr. 是一个内部项目，目标是确保 Debian 能向年幼的用户提供一些有用的东西。

关于 Debian 的更多信息，请阅读 Debian FAQ。

1.2 什么是 GNU/Linux？

GNU/Linux 是一种计算机操作系统：一系列能让您与计算机进行交互操作并运行其它程序的程序。

操作系统由多种基础程序构成。它们使计算机可以与用户进行交流并接受指令，读取数据或将其写入硬盘、磁带或打印机，控制内存的使用，以及运行其它软件。操作系统最重要的组成部分是内核。在 GNU/Linux 系统中，Linux 就是内核组件。而该系统的其余部分主要是由 GNU 工程编写和提供的程序组成。因为单独的 Linux 内核并不能成为一个可以正常工作的操作系统，所以我们更倾向使用“GNU/Linux”一词来表达人们通常所说的“Linux”。

GNU/Linux 是以 Unix 操作系统为原型创造的。自从诞生之日起，它就被设计成一种多任务、多用户的系统。这些特点使 GNU/Linux 完全不同于其它著名的操作系统。事实上，GNU/Linux 比您所能想象到更加特别。与其它操作系统绝然相反的是，没人真正拥有 GNU/Linux，其大部分开发工作都是由无偿的志愿者完成的。

后来演变为 GNU/Linux 系统的开发工作开始于 1984 年。当时，自由软件基金会 开始研发被称为 GNU 的自由的类 Unix 操作系统。

GNU 工程 开发了大量用于 Unix™ 的自由软件工具和类 Unix 操作系统，例如 GNU/Linux。这些工具使用户能执行从日常俗事 (例如在系统中复制和删除文件) 到神秘操作 (例如书写和编译程序或对多种文档格式进行熟练的编辑工作) 的各种任务。

虽然有许多组织和个人都对 GNU/Linux 的发展作出了帮助，但是自由软件基金会依然是最大的单个贡献者。他不仅仅创造了绝大部分在 GNU/Linux 中使用的工具，还为 GNU/Linux 的存在提供了理论和社会基础。

Linux 内核 的首次面世是在 1991 年。当时，名为 Linus Torvalds 的芬兰计算机科学系学生在 Usenet 新闻组 `comp.os.minix` 上发布了一种 Minix 替代内核的早期版本。请看 Linux International 的 Linux 历史。

在一些子系统维护人员的帮助下，Linus Torvalds 一直协调着数百位开发人员的工作。Linux 内核有一个官方网站。有关 `linux-kernel` 邮件列表更多的信息可以在 linux-kernel mailing list FAQ 上找到。

GNU/Linux 用户在挑选软件方面有着极大的自由。例如，他们可以选择几十种不同的命令行外壳和多种图形桌面。这通常让其它操作系统的用户手足无措，他们甚至从来没想过命令行和桌面其实是可以更换的。

GNU/Linux 很少会崩溃、适合在同一时间运行多个程序，而且比大多数操作系统更为安全。有了这些优势，Linux 成为在服务器市场上增长最快的操作系统。近来，它还开始在家庭和商业用户中变得越来越流行。

1.3 什么是 Debian GNU/Linux?

将 Debian 哲学与方法论，GNU 工具集、Linux 内核，以及其他重要的自由软件结合在一起所构成的独特的软件发行版称为 Debian GNU/Linux。该发行版由大量的软件包组成。发行版中的每个软件包都包含了执行文件、脚本、文档和配置信息，并拥有一位维护者，其职责是保持软件包更新、跟踪 bug 报告，并与软件源作者保持联络。我们巨大的用户群，结合 bug 跟踪系统可以确保问题被及时发现和修复。

Debian 对细节的注重使我们可以生产出高质、稳定和灵活的发行版。安装程序可以方便地使之应用到各种场合，从精简的防火墙到桌面科学工作站，甚至高端网络服务器都可以轻松胜任。

Debian 在高级用户中非常流行的原因在于它具有优秀的技术，而且它对 Linux 的深入贡献满足了社区的需求与期望。Debian 为 Linux 引入的许多特性现在已经成为了非常通用的标准。

例如，Debian 是第一种使用包管理系统的 Linux 发行版，它让安装和删除软件变得非常容易。而且它还是第一个可以不用重新安装就能升级的 Linux 发行版。

Debian 一直充当着 Linux 开发的带头人。它的开发过程是开源开发模式如何工作的范例—即使是用于创建和维护一个完整操作系统这样复杂的任务。

Debian 与其他 Linux 发行版最大的不同之处在于包管理系统的特性。这些工具给予 Debian 系统管理员对安装到系统上的软件包的完全控制，包括安装单个软件包和自动升级整个操作系统。个别软件包也可以被保护不被升级。您甚至可以告诉包管理系统哪些软件是您自己编译的以及它们所需要的依赖关系。

为了提防“特洛伊木马”和其他恶意软件，更好地保护您的系统，Debian 服务器会校验 Debian 注册维护人员所上传的软件包。Debian 的打包人员也会特别注意以安全的方式配置软件包。当发布出去的软件包有安全问题时，通常会很快被修复。依靠 Debian 的简易更新选项，安全更新可以通过互联网自动下载和安装。

为您的 Debian GNU/Linux 系统寻求技术支持和联络 Debian 开发人员的主要的，也是最好的方法是通过 Debian 计划的邮件列表 (书写本文时已有 327 个邮件列表) 进行沟通。订阅这些列表的最简单方法是访问 Debian 邮件列表订阅页并填写相应表单。

1.4 什么是 Debian 安装程序？

Debian 安装程序，也被称为“d-i”，是安装 Debian 系统进行基本工作的软件系统。嵌入式设备、笔记本计算机、台式机和服务器这样的大范围硬件被支持，并且提供了用于很多目的的大量自由软件。

通过回答一组基本问题来进行安装。还具有专家模式，允许控制安装的每个方面，并提供高级特性来进行自动安装。安装的系统可以直接使用或进一步定制。可以从多种来源进行安装：USB、CD/DVD/Blu-Ray 或网络。安装程序支持超过 80 种语言的本地化安装。

安装程序来源于安装软盘项目，是由 Joey Hess 在 2000 年 首先提到 。从那时起，安装系统由志愿者不断开发，改进并添加了更多特性。

更多信息可以在 Debian Installer page 、Wiki 和 debian-boot mailing list 上找到。

1.5 获得 Debian

关于如何从因特网上下载 Debian GNU/Linux 或者购买官方 Debian 安装介质的信息，请参阅 distribution web page。list of Debian mirrors 包含完整的 Debian 官方镜像，您可以很容易找到最近的一个。

Debian 可以很容易地在安装之后升级。如果需要，安装步骤将帮助您建立系统，使您可以在安装之后立即进行升级。

1.6 获得本文档最新版本

本文档会不断地修订。请查阅 Debian 13 页面 以了解最近的关于 13 版 Debian GNU/Linux 系统的的信息。更新的安装手册也可以从 官方安装手册页得到。

1.7 文档的组织

本文意在成为 Debian 的新手的使用手册。尽管它对用户的专业知识水平已经做了尽量少的假设，但我们还是设想您对计算机硬件如何工作有一个基本的了解。

专业用户可以在本手册中找到一些有趣的参考信息，包括最小安装尺寸，Debian 安装程序支持的硬件细节，等等。我们鼓励专业用户有选择性地浏览本文档。

基本上，本手册是以单线方式组织，文章内容将沿着安装进程由始至终进行。以下是安装 Debian GNU/Linux 的步骤，以及每步相关的章节：

1. 确定您的硬件是否满足安装系统的要求，位于第 2 章。
2. 备份系统，在安装 Debian 之前进行必要的规划和硬件配置，位于第 3 章。如果需要多重引导系统，您可能需要为 Debian 创建分区空间。
3. 在第 4 章，您可以获得所选安装方式所需要的安装文件。
4. 下面的第 5 章讲述的是如何启动安装系统。本章还将讨论如何解决这步操作中会遇到的问题。
5. 执行真正的安装是在第 6 章。其中涉及选择语言、设置周边驱动模块、设置网络连接以从 Debian 服务器上获得后续安装文件（如果您不是采用一组 DVD 安装映像的方式来安装），给硬盘分区和安装基本系统。（关如何为 Debian 系统建立分区的背景知识位于附录 C。）
6. 从您新安装的基本系统启动，第 7 章。

一旦您安装完系统，可以阅读第 8 章。该章将解释在哪里寻找更多有关 Unix 和 Debian 的信息，以及如何替换您的内核。

最后，是关于本文档的信息以及如何贡献您的想法，附录 E。

1.8 关于版权与软件协议

我们相信您曾经阅读过许多商业软件的许可证—它们通常要求您只能在一台计算机上使用一份软件拷贝。本系统与它们完全不同。我们鼓励您将 Debian GNU/Linux 拷贝用到您的学校或商业机构的每台计算机上。将安装介质借给朋友，并帮助他们安装到他们的计算机上！您甚至可以制作上千份拷贝，然后出售它们—尽管有一些限制。您安装和使用系统的自由权利因为 Debian 是基于自由软件。

称软件为自由并不意味着该软件没有版权，也不是指包含该软件的安装映像必须免费分派。自由软件，某种程度上意味着您不须为分发和使用这些程序付费。自由软件还是指任何人可以扩展、改写、修改该软件，并且他们工作的成果也这样发布。

> **注意**
>
>
>
> 注意，Debian 计划因为考虑用户实用性，也有不符合我们自由规则的软件包存在。这些软件包不属于官方发行版，只存在于 `contrib` 或 `non-free` Debian 镜像区域或者第三方的 CD/DVD-ROM 中，请参阅 Debian FAQ 的“The Debian FTP archives”章节以了解档案布局和内容的更多信息。

系统中很多软件所使用的许可证都是 GNU General Public License (通用公共授权)，通常简称为“the GPL”。GPL 要求您将程序的源码随您发布的二进制拷贝一起提供；许可证的这项规定确保任何用户都可以修改软件。正是由于此规定，所有这些程序的源码[1]都可以在 Debian 系统中得到。

Debian 中的程序还有许多其他形式的版权声明和软件许可证。您可以在您的系统中找到每个软件包相关的版权和许可证，它们位于 `/usr/share/doc/软件包名/copyright` ，这些文件在您将软件包安装到系统后就存在了。

关于 Debian 如何判断软件是否足够自由到可以包括在主发行版中的信息，请参阅 Debian Free Software Guidelines。

在此，最重要的法律声明是此软件无担保。程序员创造此软件为的是使社群受益，而不能保证软件适用于任何特定用途。但是，因为软件是自由的，您可以修改此软件以使其能适用于您的需要—并享用其他人采用这种途径扩展后的修改成果。

[1]对于如何根据 Debian 源码包寻找、解包和创建二进制软件包，请查阅 Debian FAQ 的“Basics of the Debian Package Management System”章节。

Chapter 2

系统需求

本章包括运行 Debian 所需硬件的信息。您还能找到更多有关 GNU 和 Linux 所支持硬件的信息的链接。

2.1 支持的硬件

Debian 不会超出 Linux 内核与 GNU 工具集所支持的硬件范围之外。因此，任何被移植了 Linux 内核、libc、gcc 等，并拥有对应的 Debian 移植版的硬件体系或平台都可以运行 Debian。请参考移植页面 https://www.debian.org/ports/s390/ 以了解更多已被 Debian GNU/Linux 测试过的 S/390 体系。

本章仅包含一些通用的信息，以及在何处可以获得更多信息的指导，而不是试图列出支持 S/390 的所有不同硬件配置。

2.1.1 支持的体系

Debian GNU/Linux 13 supports 7 major architectures and several variations of each architecture known as “flavors”.

体系	Debian 命名	子体系	Flavor
AMD64 & Intel 64	amd64		
带 FPU 的 ARM	armhf	多平台	armmp
64 位 ARM	arm64		
MIPS（小尾端）	mips64el	MIPS Malta	5kc-malta
		Cavium Octeon	octeon
		Loongson 3	loongson-3
Power 系统	ppc64el	IBM POWER8 或更新的机器	
64bit RISC-V (little-endian)	riscv64		
64 位 IBM S/390	s390x	来自 VM-reader 和 DASD 的 IPL	generic

本文档主要讲述的是 S/390 体系下的安装。如果您在寻找其他 Debian 所支持的体系的信息，请访问 Debian-Ports 网页。

2.1.2 zSeries 和 System z 机器类型

z/Architecture 处理器至少需要 z196 型。关于 IBM Linux 最新的支持信息位于 Linux on System z® page on developerWorks 。

2.1.3 PAV 和 HyperPAV

PAV 和 HyperPAV 都透明支持，不需要通过多路 (multipathing) 获得这些存储系统的特性。要在 DASD 设备选取时配置。设备别名将无法格式化、分区和直接使用。

2.1.4 多处理器

多处理器支持—又称“symmetric multiprocessing”或 SMP —在本体系架构下可以使用。在一台计算机上安装多处理器原来只是高端的服务器才具有，但近年来随着“多核”处理器的出现已经应用到相对低端的桌面计算机和便携机上。这种处理器包含两个或多个称为“核心”的处理器单元集成在一个物理芯片上。

标准的 Debian 13 内核映像编译时已经加入 SMP 支持。它在非 SMP 系统上使用也没有问题。

2.1.5 网络连接硬件

几乎所有被 Linux 内核支持的网卡 (NIC) 都被安装系统支持；驱动程序通常会自动加载。

支持的网络设备有：

- Channel to Channel (CTC) 和 ESCON connection (实际或模拟)
- OSA-2 Token Ring/Ethernet 和 OSA-Express Fast Ethernet (non-QDIO)
- QDIO 模式的 OSA-Express，HiperSockets 和 Guest-LANs

2.1.6 外围设备与其他硬件

从 XPRAM 和磁带安装的软件包不受本系统支持。所有要安装的包必须位于 DASD 上或者通过 NFS、HTTP、FTP 从网络上取得。

2.2 安装介质

本节将帮助您决定使用哪种介质的版本来安装 Debian。有一整章专门用于说明介质：第 4 章，里面列出了每种类型的优缺点。当您阅读那一章的时候，可能需要回过头来参考本页。

2.2.1 网络

安装过程中可以使用网络来获取安装所需要的文件。是否使用网络取决于您选择的安装方法和安装过程中回答的特定问题。安装系统支持大多数类型的网络连接 (包括 PPPoE，但不含 ISDN 或 PPP)，使用 HTTP 或 FTP。安装完成之后，您还可以配置您的系统使用 ISDN 和 PPP。

2.2.2 硬盘

从硬盘启动安装系统是许多体系的另外一个选择。这将要求其他的操作系统从硬盘加载安装程序。该方式只推荐用在没有其他安装方式可以选的情况下。

2.2.3 Un*x 或 GNU 系统

如果您正在运行其它的类 Unix 系统，也可以用它来安装 Debian GNU/Linux 而无须本手册剩余部分讲述的 `debian-installer` 使用方法。这类安装方法可能对那些因为硬件不支持或机器停产的用户有用。如果您对此技术感兴趣，请直接跳到第 D.3 节。这种安装方法只推荐给高级用户在没有其他安装方法可选的情况下使用。

2.2.4 受支持的存储系统

Debian 安装程序包含的内核能运行在最大数量的系统上。

IDE 系统也被支持。

所有 Linux 内核支持的存储系统也都被启动系统所支持。这意味着 FBA 和 ECKD DASD 可以被旧的 Linux disk layout (ldl) 和新的 common S/390 disk layout (cdl) 支持。

2.3 内存与磁盘空间的需求

我们推荐至少有 512MB 的内存和 4GB 的硬盘空间来进行安装。如需了解详细的值，请参阅第 3.4 节。

Chapter 3

开始安装 Debian GNU/Linux 之前

本章涉及在启动安装程序安装 Debian 之前的准备工作。这包括备份您的数据、搜集您硬件的信息，以及获取其他一些必要的信息。

3.1 安装概述

先说一下有关重新安装的事情。使用 Debian 时，极少出现必须进行完全重新安装的情况，导致这种情况的可能大多是硬盘的物理故障。

许多常见的操作系统在遇到严重错误，或者要升级到新的版本时，必须进行重新安装。就算不需要完全重新安装，通常还得把您的各种程序在新的系统里也再重新安装一遍才行。

在 Debian GNU/Linux 里，如果操作系统遇到出错的情况，您大多时候都能把它修复而不用重新安装。升级不再需要进行大规模的安装操作，您可以就地升级它，而这些程序也总能和新版的操作系统和平共处。如果一个程序的新版本需要其他程序的更新版本来支持它，Debian 软件包管理系统会自动帮您确定所有必须的软件，并把它们安装上。关键是，Debian 为了避免重新安装而做了大量努力，所以您尽可以不用去烦心这些问题了。安装程序不是设计用来重装您的系统的。

以下是你在安装过程中的步骤路线图。

1. 把将要用来安装系统的硬盘上的所有数据或者文档都备份下来。
2. 在开始整个安装过程之前，收集您的计算机的硬件信息和可能会用到的文档。
3. 在硬盘上为 Debian 留出可以用来创建分区的空间。
4. 查找并（或）下载您机器所需要的安装程序软件、特殊的驱动程序或固件文件。
5. 制作 CD/DVD/U 盘引导介质或提供安装程序可以引导的网络引导环境。
6. 启动安装系统。
7. 配置一个网络接口。
8. 开启新系统的 ssh 连接。
9. 加上一个或多个直接存取设备 (DASD)。
10. 创建并挂载用来安装 Debian 的分区。
11. 等待自动下载、安装和设置基本系统。
12. 选择并安装 另外的软件。
13. 安装 boot loader，它负责启动 Debian GNU/Linux 和（或）已有的其他系统。
14. 第一次加载新安装的系统。

如果您在安装过程中遇到问题，下面的信息可以帮助您找到每个步骤里涉及到那些软件包。有关这场安装“戏剧”的“主角”的介绍：

安装软件，`debian-installer`，是本手册的关注焦点。它探测硬件并加载相应的驱动程序，使用 `dhcp-client` 建立网络连接，运行 `debootstrap` 安装基本系统的软件包，还运行 `tasksel` 让您安装

某些附加软件。在这个过程中，还有许多其他程序扮演着各自的角色，而 debian-installer 将伴随您到第一次启动新系统时才完成它的使命。

要调节系统满足需求，tasksel 允许您选择安装各种预先定义的软件包集合，例如 Web 服务器或一个桌面环境。

安装过程中一个重要的选项是要不要安装图形桌面环境，它由 X Window System 和一个图形桌面环境组成。如果没有选择“桌面环境”任务，就只有一个相对基本的、命令行驱动的系统。把安装桌面环境任务作为选项，是因为相对文本系统它需要大量的磁盘空间，而且由于许多 Debian GNU/Linux 系统是作为服务器，并不需要图形用户界面来完成它们的工作。

3.2 记得备份您所有的数据！

开始安装之前，请确定您已经把系统中所有数据都进行了备份。如果这是您首次安装非预装的操作系统，很可能需要对硬盘进行重新分区，来给 Debian 腾出安装的空间。不管使用什么程序，分区都会使硬盘上的所有数据全部丢失。在我们的安装 Debian GNU/Linux 的过程中使用的分区程序经过多年使用，被证明非常可靠，但它的功能也非常强大，您可能会为一次错误操作而付出代价。即使是已经备份过数据，也要谨慎使用，最好在每次操作之前先认真考虑一下。两分钟的思考可能会为您节省几个小时的不必要的恢复工作。

如果您是想把计算机做成多重引导的系统，请先把计算机里存在的这些操作系统的安装介质放在手边。即使一般用不上，偶尔也会需要您重新安装操作系统的引导程序 (boot loader) 使系统可以启动，或者更糟的情况下，得重新安装该操作系统并恢复之前备份的文件。

3.3 一些有用的信息

3.3.1 文档

3.3.1.1 安装手册

您正在阅读的文档，是 Debian trixie 发行版安装指南的正式版本; 还有各种文件格式和不同语种的翻译。

3.3.1.2 硬件文档

通常包含设置和使用您硬件的有用信息。

3.3.1.3 S/390 Hardware References

S/390-规格引导次序、命令和设备驱动程序文档 (如：DASD、XPRAM、Console、OSA、HiperSockets 和 z/VM interaction)

- 设备驱动程序、特性和命令 (Linux 内核 3.2)

IBM Redbook 讲述了如何将 Linux 与 zSeries 的 z/VM 和 S/390 硬件组合。

- 用于 S/390 的 Linux

IBM Redbook 介绍了可用于 mainframe 的 Linux 发行版。其中没有关于 Debian 的章节，但对于所有的 S/390 发行版，基本的安装概念是相同的。

- Linux for IBM eServer zSeries and S/390: Distributions

3.3.2 提供硬件信息的资源

许多情况下，安装程序能自动检测您的硬件。但作为预备，我们推荐您还是在安装之前熟悉一下您的硬件比较好。

获取硬件信息的途径有：

- 每个硬件附带的手册。
- 您计算机的 BIOS/UEFI 配置画面。在计算机启动时，可以通过按组合键查看这些画面。请确认您手册中的组合键。通常，它是 Delete 或 F2 键，但有些厂商会使用其他的组合键。一般在系统开机的时候会显示信息表明按哪些键来进入设置界面。

- 每个硬件的包装盒。
- 其他操作系统里面的系统命令或工具，包括文件管理器的显示。该资源对了解 RAM 和硬盘特别有用。
- 您的系统管理员或者互联网服务提供商。他们可以告诉您所需的网络和 e-mail 配置信息。

Table 3.1 有助于安装的硬件信息

硬件	您需要了解的信息
DASD	设备数。
	可用空间。
网络	适配器类型。
	设备数。
	OSA 卡的相对适配器号。

3.3.3 硬件兼容性

多数产品在 Linux 上运行不会有问题。而且，Linux 对硬件支持每天都在改善。然而，Linux 仍然不能像某些操作系统那样可以在各种不同的硬件上运行。

大多数情况下 Linux 里面的驱动程序并不是为某一厂商特定的“产品”或“品牌”开发的，而是基于特定的硬件/芯片。许多表面上看起来不同的产品/品牌实际上是基于相同的硬件设计；芯片厂商为芯片提供称为“公版”的产品，其他设备制造商采用这种设计并以不同的产品和品牌来销售是常见的做法。

这样做有利也有弊。优点在于只要不同厂商生产的产品基于相同的芯片，驱动程序就可以用在多种产品上。不足之处是确定某一产品/品牌到底使用哪种芯片并不总是那么容易。更不幸的是设备厂商经常在修改产品设计的同时并不更改产品名称甚至是产品编号，以至于在不同时段购买的同一品牌/产品，它们会采用不同的芯片，造成需要使用两种不同的驱动程序，或者其中一个没有任何驱动程序。

对于 USB 和 PCI/PCI-Express/ExpressCard 设备，发现所采用芯片好的方法是查看它们的设备 ID。所有的 USB/PCI/PCI-Express/ExpressCard 设备具有一个称为“vendor”和“product”的 ID，使用相同芯片的产品这两个组成的数值通常是一样的。

在 Linux 系统上，该 ID 可以使用 lsusb 命令读取 USB 设备上的值，lspci -nn 命令用于 PCI/PCI-Express/ExpressCard 设备。vendor 和 product ID 通常是十六进制数值形式，用冒号隔开，例如“1d6b:0001”。

lsusb 命令的输出的形式为：“Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub”，其中 1d6b 是 vendor ID 而 0002 是 product ID。

lspci -nn 输出的网卡设备形式为：“03:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/8168B PCI Express Gigabit Ethernet controller [10ec:8168] (rev 06)”。ID 位于最右边的方括号里面，这里 10ec 是 vendor ID 而 8168 是 product ID。

再举一个例子，显卡设备的输出形式为：“04:00.0 VGA compatible controller [0300]: Advanced Micro Devices [AMD] nee ATI RV710 [Radeon HD 4350] [1002:954f]”。

在 Windows 系统上，设备 ID 可以通过打开 Windows 设备管理器上的“details”页查看，其中 vendor ID 的前缀是 VEN_ 而 product ID 的前缀是 DEV_。在较新的 Windows 系统上，您需要再打开设备管理器页上的“Hardware IDs”属性才能看到 ID，默认是不显示的。

通过互联网搜索 vendor/product ID，使用“Linux”和“driver”作为搜索项得到的结果通常含有特定芯片对应驱动程序支持的情况。如果搜索 vendor/product ID 不能得到有用的结果，请换成芯片代码名称来搜索，该名称也是通过 lsusb 和 lspci(网卡例子中的“RTL8111”/“RTL8168B”和显卡例子中的“RV710”) 命令查出。

3.3.3.1 使用 Live-System 测试硬件兼容性

Debian GNU/Linux 在一些架构上可以使用称为“live 系统 (live system)”的形式。live 系统是一种经过压缩的预先配置好的系统，从只读的介质比如 CD 或 DVD 上直接引导就可以使用。它在默认的情况下并不对您的计算机进行永久性的改动。您可以在上面修改用户设置，安装额外的程序，但这些修改只存在于计算机的内存里面，也就是说，只要一关机，或者重启 live 系统，所有的东西都会恢复到默认状态。如果您想了解 Debian GNU/Linux 对硬件的支持情况，最简单的方法就是运行一个 Debian live 系统来试试。

使用 live 系统有一些限制。首先，您所有的改动都只存在于内存之中，这就要求系统有足够的内存，像安装一个大的软件包就有可能因内存无法满足而失败。另外一个限制与硬件兼容性测试相关，因为官方的 Debian GNU/Linux live 系统只使用自由的组件，造成非自由的固件无法包含在里面。这些非自由的

软件包当然可以通过手动方式安装，但这里没有 debian-installer 里可以自动检测所需固件的功能，因此需要的时候只能通过手动方式安装非自由的组件。

有关 Debian live 映像的各种信息可以参考 Debian Live Images website。

3.3.4 网络设置

如果您的计算机固定连接在其他人管理的网上（即，以太网或等价的连接—而不是拨号或 PPP 连接），您需要向网络系统管理员咨询这项信息：

- 您的主机名 (您也许可以自己决定)。
- 您的域名。
- 您的计算机 IP 地址。
- 您网络的网络掩码。
- 路由经过的默认网关的 IP 地址，如果您的网络有网关的话。
- 您的网络中作为 DNS(域名服务) 服务器的系统。

如果通过 DHCP (Dynamic Host Configuration Protocol) 设置连接网络，您不需要这些信息，因为 DHCP 服务器将在安装过程中直接将它们提供给您的计算机。

如果您通过 DSL 或有线猫 (即使用有线电视网络) 访问互联网，并由路由器 (一般由电信或有线电视商预先设置) 处理网络连接，通常默认都使用 DHCP。

3.4 满足最低的硬件要求

您一旦收集好计算机上硬件配置的相关信息，复查一下您的硬件，就可以让您如愿以偿，安装上系统。

Table 3.2 推荐的最低系统配置

安装类别	RAM (最低要求)	RAM (推荐配置)	硬盘
无桌面的系统	512MB	1GB	4GB

最低要求配置认为交换空间（swap）启用，且使用非 live 安装映像。“无桌面的系统”配置认为使用基于文本（非图形界面）的安装程序。

实际的内存最低要求会低于表中的数字很多。启动交换分区时，Debian 可以在 120MB 这样少的内存上安装。同样，对磁盘空间也是如此，取决于您选择安装的应用软件。有关所需磁盘空间的另外的信息，请参见第 D.2 节。

为了能在这类内存较小的系统之上运行，安装程序通常能够自动启用一些节省内存的优化手段，但是对于一些测试较不完善的架构，安装程序也可能不会这么做。不过，您也可以通过添加 lowmem=1 或更激进的 lowmem=2 引导参数手动启用这些优化（参见第 6.3.1.1 节以及第 5.2.1 节）。

警告

对于 s390x，这些低内存级别尚未被测试，所以自动检测可能已经失效。如果您的系统内存很小，您可能需要传递这些引导参数。

在较小的内存或磁盘空间上安装系统或许可行，但只针对有经验的用户。

很难说清楚服务器安装该需要多少内存和磁盘空间，这完全取决于服务器的用途。

要提醒您的是，上面所说的空间大小并没有把任何其它的资料数据包含在内。这通常包含用户的文件、信件或者数据。在考虑您自己的文件和数据空间时，越慷慨越好。

在推荐的系统需求中已经包含流畅运行 Debian GNU/Linux 所需的磁盘空间。尤其是 /var 分区里装有许多 Debian 的特定状态信息和类似日志文件这样的常规内容。dpkg 所用的文件 (含有所有已安装软件包的信息) 很容易就会消耗掉 40MB。另外，apt-get 会在安装前将下载的软件包放在这里。您应当最少分配 200MB 的空间给 /var，如果您安装了图形桌面系统，还要更多空间。

3.5 为多重启动系统事先分区

为您的硬盘分区仅仅指的是将您的硬盘空间切分成几块。分区之后，每一块都是独立于其余部分的单独空间。这和在一个大房子里砌几堵墙有几分相似，如果您在其中一间房间里安置家具，那么这不会对其它房间有任何影响。

本节中所有提到“硬盘”的地方，您应当把它理解为 S/390 世界中的 DASD 或者 VM minidisk。同时，在这里，一台机器意指的是 LPAR 或者客户虚拟机。

如果在您的机器中已经安装有操作系统 (VM、z/OS、OS/390，…) 占用了整个磁盘，同时您也希望把 Linux 装在同一块硬盘上，那么就必须重新对硬盘分区。Debian 需要它自己专用的硬盘分区。它不能被安装在 Windows 或者 MacOS 的分区上。它可以与其它 Unix 系统共享一些分区，但是我们在这里不会对此进行说明。最起码，您要为 Debian 的根目录准备一个专用的分区。

通过当前的操作系统中的分区工具，您可以获知现在的分区状况，如 VM diskmap。分区工具总会提供一种办法让您查看现有的分区情况，而不作任何改动。

通常情况下，改动一个已经建立文件系统的分区，会导致其中的数据信息遭到损毁。因而，您应当在重新分区之前总是先做一下备份。继续拿房子作比喻，在移动墙壁时，您最好在把挡路的家具都移开，否则就要冒家具被毁坏的危险。

很多新的操作系统都提供无损移动和调节已有分区的功能。使得在不丢失任何数据的情况下就能添加分区空间。即使大多时候都可以正常使用，由于修改分区本身就是危险操作，应该在进行完整备份后再进行。

3.6 安装前的硬件和操作系统的相关设置

在本节中，我们将讨论安装之前有关硬件设置的一些问题。如果这些问题的确存在的话，您就需要在安装 Debian 前先做一些准备工作了。一般来说，准备工作包括：检查或者可能修改系统中 BIOS/UEFI/系统固件 (system firmware) 的设定。所谓“BIOS/UEFI”或“系统固件”就是硬件运行所需的核心软件。它在系统引导过程 (即开机之后) 中起到了至关重要的作用。

3.6.1 BIOS 设置

为了安装 Debian GNU/Linux 到 S/390 或者 zSeries 机器上，您首先要将内核引导进系统。在此平台上的启动机制与其他类型有着本质上的不同，特别是类 PC 的系统，并且您还会注意到此平台上的另外一个显著差异：大多数 (如果不是全部) 的时间，您将借助于其他客户机上的 telnet 这样的软件或者一个浏览器，通过远程方式使用。这归因于该系统上的 3215/3270 控制台是线型，而不是字符型的。

Linux 在该平台上可以是本地化方式运行于裸机上，以所谓的 LPAR(Logical Partition) 方式或者由 VM 系统提供的虚拟机方式。您可以在所有的系统上使用启动磁带；您也可以使用其他的介质，但它们并不一定都可用。例如，您可以在虚拟机上使用虚拟读卡器，或者在 HMC (Hardware Management Console) 可用的情况下，从 LPAR 上的 HMC 启动。

在实际安装之前，您必须仔细检查一些规划和准备步骤。IBM 有关于整个过程的文档，如，怎样准备安装介质和如何从该介质启动。这里毋需复制那些信息。但是，我们会在此说明需要哪些 Debian 规格的数据，以及在哪里能找到它们。使用这些信息资源，在引导之前您必须准备好计算机和安装介质。当您在客户端会话中看到欢迎画面，再返回到本文档查看 Debian 规格的安装步骤。

3.6.2 本地化 (Native) 和 LPAR 安装

请参考 Linux for S/390 Redbook 第 5 章和 Linux for IBM eServer zSeries and S/390: Distributions Redbook 第 3.2 节有关如何为 Linux 建立 LPAR 的部分。

3.6.3 作为一个 VM guest 安装

请参考 Linux for S/390 Redbook 第 6 章和 Linux for IBM eServer zSeries and S/390: Distributions Redbook 第 3.1 节有关如何为运行 Linux 建立 VM guest 的部分。

您需要从 `generic` 子目录复制所有的文件到 CMS 磁盘。并确保 `kernel.debian` 和 `initrd.debian` 以 80 字符固定长度二进制模式传输 (FTP 客户端使用 `BINARY` 和 `LOCSITE FIX 80`)。`parmfile.debian` 可以是 ASCII 或 EBCDIC 格式。映像里面包含有一个 `debian.exec` 示例脚本，用于按正确次序存放文件。

3.6.4 建立安装服务器

如果您没有连接到互联网 (直接或者间接通过 web 代理)，您需要创建一个您的 S/390 能访问的本地安装服务器。该服务器包含所有您需要安装的软件包，并可以通过 NFS，HTTP 或 FTP 访问。

安装服务器需要从任意 Debian GNU/Linux 镜像复制精确的目录结构，当然只需 S/390 和一些独立于体系的文件。您也可以复制所有的安装映像到这样的目录树中。

Chapter 4

获取系统安装介质

4.1 官方的 Debian GNU/Linux 安装映像

By far the easiest way to install Debian GNU/Linux is from a set of official Debian installation images. You can buy a set of CDs/DVDs or an USB stick from a vendor (see the CD vendors page). You may also download the installation images from a Debian mirror and make your own set, if you have a fast network connection and a CD/DVD burner. Or the modern way: write such image to an USB stick (see the Debian CD/DVD page and Debian CD FAQ for detailed instructions). If you have such installation media, and they are bootable on your machine, you can skip right to 第 5 章. Much effort has been expended to ensure the most-used files are on the first DVD image, so that a basic desktop installation can be done with only the first DVD.

同样，要记住：如果使用的安装介质不包括需要的软件包的话，总是可以以后从运行的新 Debian 系统来安装这些软件包（在安装结束后）。如果需要知道在哪个安装映像找到特定的软件包，请访问https://cdimage-search.debian.org/。

如果您的机器不支持从光盘媒介引导，但您有一套 DVD，可以使用替代的方案，例如：VM 读取器、或者从光盘手工装载内核来初始化系统安装程序。光盘的里已经包含了用其他方法引导所需的文件；Debian 网络存档和文件夹组织方式相同。因此当引导需要特定的文件时，可以从安装媒介的相同目录和子目录中找到。

一旦安装程序被引导，它将能够获得光盘中其他所有必需的文件。

如果您没有安装介质，则需要下载安装程序系统文件并且把它们放到 VM minidisk，然后它们可以被用来引导安装程序。

4.2 从 Debian 镜像服务器下载文件

选择离您较近的 (因此很可能也是最快的) 镜像服务器，请参见 Debian 镜像列表。

4.2.1 在哪里能找到安装文件

各种安装文件可以在每个 Debian 镜像的 debian/dists/trixie/main/installer-s390x/current/images/ 中找到；MANIFEST 文件列出了每一个映像及其用途。

4.3 自动化安装

要同时安装多台计算机，可以使用完全自动方式安装。这方面的 Debian 软件包包括可以用来安装服务器的 `fai-quickstart`（请参阅 FAI home page 了解更多细节）和 Debian 安装器自身。

4.3.1 使用 Debian 安装程序进行自动安装

Debian 安装程序支持使用预先配置的文件进行自动安装。预置文件可以从网络或移动介质上加载，并自动回答安装过程中的问题。

在附录 B 是有关预置文件的完整文档，并有一些可以使用的例子供您修改。

4.4 验证安装文件的完整性

您可以相对 Debian 镜像上的 SHA256SUMS 或 SHA512SUMS 文件来验证下载的文件的完整性。你可以在相同位置发现它们作为安装映像本身。请访问后面的位置：

- CD 映像的校验和文件,
- DVD 映像的校验和文件,
- 其他安装文件的校验和文件.

为了计算下载的安装文件的校验和，请分别使用

```
sha256sum filename.iso
```

和

```
sha512sum filename.iso
```

，然后将显示的校验和与各自相应的 SHA256SUMS 和 SHA512SUMS 文件中的值进行比较。

在这个题目（如脚本check_debian_iso，将上述过程半自动化）上，Debian CD FAQ 具有 更多有用信息，以及指示，如何验证上述校验和文件自身的完整性。

Chapter 5

引导安装系统

5.1 在 S/390 上引导安装程序

5.1.1 S/390 的限制

为了在 S/390 上运行安装系统，需要工作网络设置和 ssh 会话。

引导过程开始时候提示要求一些网络参数来进行网络设置。如果设置是成功的，将会通过启动 ssh 会话来登录系统，这会启动标准的安装系统。

5.1.2 S/390 引导参数

在 S/390 上可以在 parm（参数）文件中添加引导参数。这个文件可以是 ASCII 或者 EBCDIC 格式的。要求它需要是每行 80 字符固定宽度的。parm（参数）文件的示例 `parmfile.debian` 提供有安装映像。如果参数太长无法与 80 个字符的限制匹配，那么可以简单地在下一行第一列继续。当传输到内核时所有的行都去掉空格连接在一起。

如果您在逻辑分区（LPAR）或能看到很多设备的虚拟机（VM）上引导安装程序，则可以指示内核将列表限制固定的一组设备上。如果大量磁盘可见，更可能是在 LPAR 模式中，建议在安装程序的引导过程中使用。“cio_ignore”选项支持黑名单（仅禁用少数设备）和白名单（仅允许特定设备）：

```
# blacklist: just ignore the two devices 300 and 301
cio_ignore=0.0.0300-0.0.0301
# whitelist: ignore everything but 1150, FD00, FD01 and FD02
cio_ignore=all,!0.0.1150,!0.0.fd00-0.0.fd02
```

请注意，所有设备编号的十六进制数字需要以小写形式指定。进一步，如果使用此引导参数，则需要列出所有设备：至少包括磁盘、网络设备和控制台。为了安装程序引导过程，上述选项需要添加到 `parmfile.debian` 中。

5.2 引导参数

引导参数是 Linux 内核参数，一般用于确保能够正确地处理外围设备。对于其中的大多数部分来说，内核能够自动检测外围设备的相关信息。但是在某些情况下，需要给内核一点小小的帮助。

如果这是第一次引导系统，尝试缺省的引导参数（即：不要尝试甚至任何参数），并且检查是否工作正确。这可能成功，如果不成功可以重新引导，并且查找向系统通知硬件的任何特定参数。

很多引导参数的信息可以查看 Linux BootPrompt HOWTO，其中包括一些老旧硬件的提示。这一节只包含一些最突出的基本参数。一些常见的问题描述见下面的第 5.3 节。

5.2.1 Debian 安装程序的参数

安装系统能识别一些会有用的额外的引导参数[1]。

许多参数都有“简写”，来帮助避免内核命令行选项的限制，并且更容易输入参数。如果参数有简写，那么会列在（正常的）长形式后面的括号里。本手册中的示例一般也使用简写。

[1]在当前的内核（2.6.9 或更新），您可以使用 32 个命令行参数和 32 个环境变量选项。如果超出内核将会崩溃。同样，整个内核命令行有 255 个字符的限制，超出这个限制的任何内容将被截掉而不会有提示。

debconf/priority (priority) 这个参数将显示的消息设置为最低的优先级。

默认安装使用 `priority=high`。这意味着显示高优先级和至关重要的消息，但是中和低优先级的消息将被跳过。如果其间遇到问题，那么安装程序将按照需要调整优先级。

如果添加 `priority=medium` 作为引导参数，那么将看到安装菜单，并在安装过程中得到更多的控制。当使用 `priority=low` 时，将显示所有的消息（这相当于采用 expert（专家）引导模式）。通过 `priority=critical`，安装系统只会显示至关重要的消息，并且镇静地尝试做正确的事。

DEBIAN_FRONTEND 这个引导参数控制安装程序使用的用户界面的类型。当前可能参数设置有：

- `DEBIAN_FRONTEND=noninteractive`
- `DEBIAN_FRONTEND=text`
- `DEBIAN_FRONTEND=newt`
- `DEBIAN_FRONTEND=gtk`

默认的前端是 `DEBIAN_FRONTEND=newt`。`DEBIAN_FRONTEND=text` 对于串口控制台安装会是首选的。一些特殊类型的安装介质可能只提供有限的前端选择，但 `newt` 和 `text` 在多数默认的安装介质上都提供。图形安装程序可以在所支持的架构上使用 `gtk` 前端。

BOOT_DEBUG 将这个引导参数设置为 2 会使安装程序的引导过程更冗长地记录到日志中。将其设置为 3 会在引导过程的战略要点上调出 debug shell。（退出 shell 会使引导过程继续。）

BOOT_DEBUG=0 此为默认。

BOOT_DEBUG=1 比一般情况更多的信息。

BOOT_DEBUG=2 大量的调试信息。

BOOT_DEBUG=3 在引导过程的各个点上运行 Shells，以进行细节上的调试。退出 Shell 会将引导过程继续。

log_host, log_port 使安装程序将日志消息发送到本地文件的同时，也发送给指定主机和端口上的远端 syslog 中。如果不指定，端口默认是标准 syslog 端口 514。

lowmem 可以根据可用的内存，将安装程序强制在高于安装程序默认设置的 lowmem 等级上。该值可设为 1 和 2。还请参见第 6.3.1.1 节。

noshell 放置安装程序在 tty2 和 tty3 上提供交互式 shells。这对于物理安全有限时的无人职守安装很有用。

debian-installer/framebuffer (fb) 一些架构使用内核帧缓存来以多种语言提供安装。如果帧缓存使您的系统出现问题，可以通过使用参数 `fb=false` 来关闭这个特性。这类问题的症状是关于 bterm 或者 bogl 的错误消息、空白的屏幕，或者在开始安装后几分钟内死机。

netcfg/disable_autoconfig `debian-installer` 默认自动检测 IPv6 的网络配置和 DHCP。如果检测成功的话，将没有机会回顾并且更改已经获得的设定。只能在自动检测失败的情况下手工配置网络。

如果本地网络上有 IPv6 路由器或 DHCP 服务器，但是因为例如它们给出错误的答案而不想使用它，那么可以使用参数 `netcfg/disable_autoconfig=true` 来防止网络进行任何自动配置，并手动输入信息。

hw-detect/start_pcmcia 如果遇到问题的话则设置为 `false`，来阻止启动 PCMCIA 服务。有些笔记本机总所周知会有这种故障现象。

preseed/url (url) 将 url 指定为要下载并使用云设置文件，来使安装自动化。请参见第 4.3 节。

preseed/file (file) 将路径指定到要加载的预设置文件，用于使安装自动化。请参见第 4.3 节。

preseed/interactive 设为 `true` 在即使已经预置时也显示问题。用于测试或调试预配置的文件。注意，这对于作为引导参数传递的那些参数无效，但对那些可以使用特殊语法。具体信息请参见第 B.5.2 节。

auto-install/enable (auto) 延迟那些通常在预置之前的提问，直到网络设置完毕。请参见第 B.2.3 节了解自动安装的详细信息。

finish-install/keep-consoles 从串口或者管理控制台安装的过程中，通常在 `/etc/inittab` 中禁止了常规的虚拟控制台（VT1 到 VT6）。将其设置 `true` 来将它们启用。

cdrom-detect/eject 在重新引导前，debian-installer 默认自动弹出安装时使用的光学介质。如果系统不是自动从这样的介质来引导，那么这是不必要的操作。在某些情况下甚至更为不便，例如，在光盘驱动器自己不能重新插入介质，而用户又不在旁边的情况。许多通过插口装入光盘的超薄型 caddy 风格的驱动器不能自动重新加载介质。

设为 false 可以禁止自动弹出，但要小心需要确保在初始安装后，系统不会从光盘驱动器自动引导。

base-installer/install-recommends (recommends) 通过将这个选项设置为 false，无论在安装过程中还是对于安装好的系统中，包管理系统都配置设为不自动安装这些“推荐的软件包（Recommends）”。还请参见第 6.3.5 节。

注意，虽然使用该选项可以得到一个瘦小的系统，但将会失去一些通常认为应有的特性。您会不得不手动安装一些推荐的软件包以获得想要的完整功能。因此，该选项应由非常有经验的用户来使用。

debian-installer/allow_unauthenticated 安装程序默认需要使用已知的 gpg 密钥来认证仓库。设为 true 可以禁止认证。警告：不安全，不推荐使用。

rescue/enable 设置为 true 来进入救援模式，而不是执行普通安装。请参见第 8.6 节。

5.2.2 使用引导参数来回答问题

除了一些例外，对于安装过程中询问的问题，值可以在引导提示符后设置，尽管这只在某些特定情况下真正有用。有关如何操作的一般指示可以在第 B.2.2 节中找到。下面是一些特定的例子。

debian-installer/language (language), debian-installer/country (country), debian-installer/locale (locale) 有两种方法为安装过程和安装好的系统来指定所使用的语言、国家和区域设置。

第一种且最简单的方法是只传递 locale（区域设置）参数。语言和国家将从该值中得到。例如，您可以使用 locale=de_CH.UTF-8 选择语言为德语，瑞士为国家（安装好的系统中默认将 de_CH.UTF-8 设置为默认的区域设置）。该方法的限制是，并非所有的语言、国家和区域设置的组合可以以这种方式实现。

第二种是更具灵活的选项，分开指定 language（语言）和 country（国家）。在这种情况下 locale（区域设置）可以可选地添加，为安装好的系统指定默认的区域设置。示例：language=en country=DE locale=en_GB.UTF-8。

anna/choose_modules (modules) 可用来自动加载那些默认不加载的安装程序组件。一些可能有用的可选组件的例子是 openssh-client-udeb（使您可以在安装过程中使用 scp）。

netcfg/disable_autoconfig 如果您想禁止 IPv6 自动配置和 DHCP，并强制使用静态网络设置来代替，那么将它设为 true。

mirror/protocol (protocol) 安装程序默认会使用 http 协议从 Debian 镜像下载文件，在普通优先级的安装过程中不能将协议修改为 ftp。通过将这个参数设置为 ftp，可以强制安装程序替换使用该协议。注意，不能从列表中选择 ftp 镜像，必须手动输入主机名。

tasksel:tasksel/first (tasks) 可以用来选择不在交互式任务列表中出现的任务，如 kde-desktop 任务。另外的信息请参见第 6.3.6.2 节。

5.2.3 将参数传递给内核模块

如果驱动程序编译进内核中，那么可以像内核文档中描述的那样将参数传递给内核。然而，如果驱动程序编译为模块，并且由于在安装过程中内核模块的加载与引导安装好的系统有些不同，像平时那样将参数传递给模块是不可能的。需要使用安装程序所识别的特定语法来替代，它们将确保参数能保存到合适的配置文件中，并当模块真正加载的时候去使用参数。这些参数还会自动传递到安装好的系统的配置中。

注意，需要将参数传递到模块的情况现在很少见。在多数情况下，内核将检测当前系统中出现的硬件，并很好地设置默认参数。然而，在某些情况下还会需要手动设置参数。

用于为模块设置参数的语法是：

```
module_name.parameter_name=value
```

如果需要将多个参数传递给相同或不同的模块，只需要重复上面内容即可。例如，要设置老式的 3Com 网卡来使用 BNC（同轴）端子和 IRQ 10，则需要传递：

```
3c509.xcvr=3 3c509.irq=10
```

5.2.4 内核模块黑名单

有时需要将一个模块加入黑名单以阻止它被内核和 udev 自动加载。一个原因会是特别的模块会与您的硬件产生问题。内核有时会为相同的硬件列出两个不同的驱动程序。如果驱动程序冲突或者首先加载了错误的驱动程序，那么会造成设备无法正常工作。

可以使用下面的语法将模块加入黑名单：`module_name.blacklist=yes`。这会让该模块在安装过程和安装好的系统中都加入 `/etc/modprobe.d/blacklist.local` 文件的黑名单中。

注意，安装系统自己仍然会加载模块。要阻止这种情况，可以以专家模式来运行安装程序，并且在硬件检测阶段中从显示的模块列表中取消该模块。

5.3 安装过程中的故障修复

5.3.1 引导配置

如果在引导过程中遇到了问题或者内核挂起，不识别实际拥有的外围设备，或者驱动不能被正确识别，那么首先要检查的是引导参数，如在第 5.2 节中所讨论的。

5.3.2 内核起始消息的注解

在引导顺序中，可以看到很多消息类似 `can't find something`（找不到什么东西），或者 `something not present`（什么东西不见了），`can't initialize something`（什么东西不能初始化），或者甚至 `this driver release depends on something`（这个驱动程序版本依赖于什么东西）。大多数这些消息都是无害的。之所以看到它们，是因为安装系统的内核被构建为运行在具有很多不同外围设备的计算机上。显然，没有一个计算机可能拥有所有的外围设备，因此操作系统可能在查找一些您没有的设备时发出一些报怨。有时可以看到系统暂停了一段时间。这是由于它在等待一台设备的响应，而这台设备并没有出现在您的系统上。如果您发现这个时间长得无法接受，那么以后可以创建一个定制的内核（请参见第 8.5 节）。

5.3.3 报告安装问题

如果您经过了初始化引导阶段，但还是不能完成安装，那么可以借助于 Save debug logs（存储调试日志）菜单选项。它让您将来自安装程序的系统错误日志和配置信息保存到存储介质上，或者用 web 浏览器下载。该信息或许提供了一些关于哪里出错和如何修复的线索。如果您正在提交一个 bug 报告，可以报告里附上该信息。

安装过程的其它有关安装消息可以在 `/var/log/` 内找到，而计算机引导进入新安装好的系统后消息放在 `/var/log/debian-installer/` 里。

5.3.4 提交安装报告

如果仍有问题，请提交安装报告（可能的话，请使用英文撰写）。即使已经安装成功，我们也鼓励发送安装报告，这让我们可以在最大的硬件配置数量上获得尽可能多的信息。

注意，您的安装包括将公布在 Debian Bug Tracking System（BTS，即 Debian Bug 跟踪系统）上，并转发到公共的邮件列表。请确定您使用的邮件地址可以被公开。

如果您有一个可以运行的 Debian 系统，发送安装报告最简单的方法是安装 `installation-report` 和 `reportbug` 软件包（aptitude install installation-report reportbug），配置 `reportbug`，如在第 8.4.2 节中所解释的，并且运行 reportbug installation-reports 命令。

另外，在填写安装报告时，您可以采用下列模板，并将其发送到 submit@bugs.debian.org，也就是通过向 `installation-reports` 伪软件包提交 bug 报告的方式提交安装报告。

```
Package: installation-reports

Boot method: <How did you boot the installer? CD/DVD? USB stick? Network?>
Image version: <Full URL to image you downloaded is best>
Date: <Date and time of the install>

Machine: <Description of machine (eg, IBM Thinkpad R32)>
Processor:
Memory:
Partitions: <df -Tl will do; the raw partition table is preferred>
```

```
Output of lspci -knn (or lspci -nn):

Base System Installation Checklist:
[O] = OK, [E] = Error (please elaborate below), [ ] = didn't try it

Initial boot:           [ ]
Detect network card:    [ ]
Configure network:      [ ]
Detect media:           [ ]
Load installer modules: [ ]
Detect hard drives:     [ ]
Partition hard drives:  [ ]
Install base system:    [ ]
Clock/timezone setup:   [ ]
User/password setup:    [ ]
Install tasks:          [ ]
Install boot loader:    [ ]
Overall install:        [ ]

Comments/Problems:

<Description of the install, in prose, and any thoughts, comments
      and ideas you had during the initial install.>

Please make sure that any installation logs that you think would
be useful are attached to this report. (You can find them in the installer
system in /var/log/ and later on the installed system under
/var/log/installer.) Please compress large files using gzip.
```

在 bug 报告中，请说明问题所在，包括内核挂起时最后见到的内核消息。描述进入问题状态所进行的步骤。

Chapter 6

使用 Debian 安装程序

6.1 安装程序如何工作

本架构的安装程序使用基于文本的用户界面。当前还不支持图形界面。

Debian 安装程序由一些特定功能的组件组成，它们完成各自的安装任务。每一组件执行任务时，会询问一些任务相关的问题。每个任务又有各自的优先级，在安装程序开始时可以设置优先级以决定哪些问题会被询问。

执行默认安装时，只会问到必要的 (高优先级) 问题。这使得安装过程高度自动化，几乎不需用户干预。组件自动按次序执行；至于哪些组件会执行，取决于使用的安装方法，以及您的硬件。对于没有问到的问题，安装程序将采用默认值。

假如出现问题，用户将看到错误信息画面，并且安装程序会显示菜单以进行其它替代操作。如果没有问题，用户将不会碰到菜单，只需依此回答每个组件的提问。严重错误警示的优先级是“critical”，因此用户一定能看到。

在 `debian-installer` 启动的时候，加上一些引导参数，它将会影响安装程序使用的默认值。例如，您希望指定静态网络连接配置 (默认会使用已有的 IPv6 自动配置和 DHCP)，可以加上引导参数 `netcfg/disable_autoconfig=true`。参阅第 5.2.1 节了解可用的选项。

高级用户可能更愿意采用菜单驱动的界面，安装中的每一步都可以被用户控制，而不是让安装程序自动执行。要使用菜单驱动的手动安装方式，需要添加引导参数 `priority=medium`。

假如您的硬件需要指定内核模块选项才能安装，就要使用“expert”模式启动安装程序。可以通过 expert 命令或者添加 `priority=low` 引导参数。专家 (export) 模式让您可以完全控制 `debian-installer`。

鼠标在基于文本的环境下无法使用。以下是对话框中您可以操作的键。在按钮和选项之间可以用 Tab 或 右箭头键“前进”，Shift-Tab 或 左箭头键“后退”。上和 下箭头键选择列表里面的选项，并可以滚动列表。另外，在长列表里面，您可以按下某个字母直接跳到对应字母开头的选项，按下 Pg-Up 和 Pg-Down 可以快速滚动列表。空格键可以选择选项，例如复选框。Enter 执行选择。

一些对话框可能会提供额外的帮助信息。如果有帮助信息，它们将在屏幕的最下面一行提示，可以通过 F1 键访问这些帮助信息。

S/390 不支持虚拟控制台。您可以打开第二或第三个 ssh 会话查看上述的 log。

错误信息和日志被重定向到第四个控制台。您可以通过按下左 Alt-F4 (按住左 Alt 键同时按下 F4 功能键) 访问该控制台；返回主安装进程使用左 Alt-F1。

这些信息还可以在 `/var/log/syslog` 里找到。完成安装之后，该记录会被复制到您新系统中的 `/var/log/debian-installer/syslog`。其它的安装信息安装时可以在 `/var/log/` 找到。在系统重新启动后记录的地方是 `/var/log/installer/`。

6.2 组件介绍

下面列出安装程序使用的组件，和各个组件目的的简要说明。如果您需要了解各组件详细的信息，见第 6.3 节。

main-menu 在安装过程中，显示组件列表，并执行选中的组件。主菜单的优先级为 medium，如果您的优先级设为 high 或 critical (默认为 high)，您不会看到菜单。另一方面，假如出现问题需要您干预，询问的优先级会临时调低，从而让您解决该问题，菜单则可能在这种情况下出现。

您可以通过不断地按 Go Back 按钮，从当前运行的组件回到主菜单。

localechooser 允许用户为安装程序和安装的系统选择地区选项: 语言、国家和位置。除非选中的语言没有被翻译，安装程序将以您选择的语言显示信息。对于没有翻译的语言，将显示英文信息。

console-setup 显示键盘（布局），用户在其中选择与自己的模式匹配的。

hw-detect 自动识别绝大多数系统硬件，包括网卡、硬盘和 PCMCIA。

cdrom-detect 寻找并挂载 Debian 安装介质。

netcfg 配置计算机网络连接，使它能连上互联网。

iso-scan 从硬盘上搜索 ISO 映象 (`.iso` 文件)。

choose-mirror 列出 Debian 软件包镜像。用户可以选择从哪里安装软件包。

cdrom-checker 检验安装介质的完整性。这种方法确保用户所用的安装映像没有损坏。

lowmem 它会检测小内存容量的系统，并采用一些技巧从内存中删除 `debian-installer` 不需要的部分 (代价是丧失一些特性)。

anna Anna’s Not Nearly APT。安装从选择的镜像或安装介质获取的软件包。

user-setup 建立 root 密码，然后添加一个非 root 用户。

clock-setup 更新系统时钟，并决定时钟是否使用 UTC。

tzsetup 选择时区，并基于前面选择的地点。

partman 让用户为系统中的磁盘分区，在分区上创建文件系统，并分配挂载点。它包括一些有趣的特性，如完全自动模式或 LVM 支持。Debian 里它是推荐使用的分区工具。

partman-lvm 协助用户配置 LVM (Logical Volume Manager)。

partman-md 让用户建立 Software RAID (Redundant Array of Inexpensive Disks)。Software RAID 是新主板提供的廉价 IDE (虚拟硬件) RAID 控制器的高级管理。

base-installer 安装一些基本的包，让计算机重启后能在 Debian GNU/Linux 下运行。

apt-setup 配置 apt，大部分是自动完成，基于安装程序所运行的介质。

pkgsel 用 `tasksel` 选择安装额外的软件。

os-prober 检测当前计算机上安装的操作系统，将信息传递给 bootloader-installer，使它能将检测到的操作系统添加到 bootloader 的启动菜单里面。方便用户选择启动哪个操作系统。

bootloader-installer bootloader 安装程序在硬盘上安装 boot loader 程序，这是必须的一步，使得计算机启动 Linux 不必依靠 U 盘和 CD-ROM。许多 boot loader 程序都允许用户在开机的时候选择启动哪个操作系统。

shell 让用户能从菜单或者第二控制台执行 shell。

save-logs 当用户遇到麻烦时，提供一种方式记录信息到 U 盘、网络、硬盘或其他介质，用于以后精确地给 Debian 开发人员报告安装程序的问题。

6.3 使用单独的组件

本节将会描述各个安装组件的细节。组件按用户可以辨别的阶段分组。它们在安装过程中依次出现。要知道，每次安装并不会使用所有的模块；至于哪些模块被用到，取决于安装方法以及使用的硬件。

6.3.1 启动 Debian 安装程序与配置硬件

假设 Debian 安装程序已经启动，然后您看到了初始画面。此时，debian-installer 的能力还很有限。它还不了解您的硬件，首选的语言，甚至不知道要执行什么任务。不必担心，因为 debian-installer 相当聪明，它能自动探测您的硬件，定位其余的组件，并更新自己以具备安装系统的能力。但您仍然要协助 debian-installer，提供给它不能自动决定的内容 (比如选择您首选的语言、键盘布局或网络镜像)。

您会留意到 debian-installer 在本阶段多次执行硬件侦测。第一次明确的目的是那些加载安装组件所需要的硬件 (比如，您的 CD-ROM 或者网卡)。在第一次运行的时候并不是所有的驱动程序都就绪，硬件侦测会在后续过程多次重复。

6.3.1.1 检测可用内存/小内存模式

debian-installer 首先要做的一件事是检测有效的内存。如果内存有限，该组件将修改后续的安装进程，使您可以将 Debian GNU/Linux 安装到您的系统上。

安装程序用于减少内存消耗的第一项措施是关闭翻译，也就是说，安装将只在英文下进行。当然，您在安装完成之后还是可以进行本地化设置。

如果还不行，安装程序只加载那些完成基本安装的模块，进一步减少内存消耗。这将减少安装系统的功能。您还有机会手动加载额外的组件，但要留意每个选择的组件都会使用更多的内存，有可能使安装失败。

如果安装程序运行在小内存模式下，推荐创建一个相对较大的交换分区 (1–2GB)。交换分区会被当作虚拟内存，使系统可用的内存总量增加。安装程序会在安装过程的尽早阶段激活交换分区。注意，过多地使用交换分区会降低系统性能，增加磁盘活动。

尽管如此，您的系统还可能冻结，因为系统内存溢出 (VT4 和 syslog 上产生“Out of memory”信息)，出现没有预期的错误，或者进程被内核杀死。

例如，有报告称，小内存模式下，当交换分区不够大，创建一个大的 ext3 文件系统会失败。如果大的交换分区也不能解决问题，请建立 ext2 文件系统 (它是安装程序的基本组件) 代替。安装之后是可以将 ext2 分区改成 ext3。

可以通过引导参数“lowmem”强制安装程序使用比实际内存更高的 lowmem 等级，说明见第 5.2.1 节。

6.3.1.2 选择地区选项

大多时候，您遇到的第一个问题是关于地区的选项，将用于安装过程和安装后的系统。地区选项由语言、位置和地区组成。

您选择的语言将用于紧接着的安装过程，对话框已经有各种语言翻译的版本。如果选择的语言没有对应翻译的版本，安装将默认使用英文。

在之后的安装过程中，所选择的地理位置 (多数情况下是国家) 还会被用于选择正确的时区和该国对应的 Debian 镜像。语言和国家合起来还有助于确定您系统默认的 locale，以及键盘布局。

您将被首先问到首选的语言。语言名称同时以英文 (左侧) 和该语言 (右侧) 列出；右侧名称是使用该语言来书写。列表以英文名称排列。在列表的顶端是一个额外的选项，允许您使用“C”locale 替代语言。选择“C”locale 使得安装过程使用英文；安装的系统不再有本地化支持，locales 软件包也不会安装。

接下来将要求您选择地理位置。如果您选择的语言被多个国家作为官方语言[1]，列表中只列出这些国家。如果想要选择列表之外的国家，选择 other(最后一项)。您将看到大洲列表；选择洲之后会看到该大洲上的国家。

如果语言只对应一个国家，国家列表只显示该国所属的大洲和地区，并且该国被选中作为默认。使用 Go Back 选项可以选择其他大洲的国家。

> **注意**
>
>
>
> 选择您居住的国家和地区非常重要，这将决定您安装好的系统所配置时区。

如果您组合的语言和国家没有对应的 locale，而该语言有多个对应的 locale，安装程序允许您从中选择一个默认的 locale 用于安装完成的系统。[2]。其余情况下，默认会选择基于该语言和国家的 locale。

[1] 科技术语：一种语言针对不同国家代码有多个不同 locale 存在。

[2] 在 medium 和 low 优先级下，如果该语言有多个对应的 locale，您总是可以选择所需要的那个 locale。

前面章节选择的任何默认 locale 都将使用 UTF-8 作为字符编码。

如果您使用 low 优先级安装，您可以选择额外的 locale，包括那些称为“传统”的 locale [3]，用于安装完成的系统；如果这样选择，您将被询问使用哪种 locale 作为安装完成系统中的默认 locale。

6.3.1.3 选择键盘

键盘经常按照不同语言使用的字符裁剪。为您使用的键盘选择一个合适布局，如果没有合适的布局，就选择一个近似的。一旦系统安装完毕，您可以从更大范围内选择键盘的布局 (在安装完之后以 root 身份执行 dpkg-reconfigure keyboard-configuration)。

将高亮显示部分移到您期望键盘选择，然后按下 Enter。用箭头键移到高亮显示—它们位于所有国家键盘布局相同的地方，因此独立于键盘配置。

6.3.1.4 寻找 Debian 安装程序 ISO 映像

当安装是使用 hd-media 方式，将会有一个动作，是寻找和挂载 Debian 安装程序 iso 映像，用于获取其它的安装文件。组件 iso-scan 正是用来完成此任务。

首先，iso-scan 自动挂载具有已知文件系统的块设备 (比如，分区和逻辑卷)，然后顺次搜索文件名结尾是 `.iso` (或 `.ISO` 之类)。注意，首次只扫描位于根目录的文件，和第一层子目录 (例如，它找到 `/whatever.iso`，`/data/whatever.iso`，而不是 `/data/tmp/whatever.iso`)。在 iso 映象文件找到后，iso-scan 检查它包含的内容，确定该映象文件是不是有效的 Debian iso 映象。如为前者，任务结束；若是后者，iso-scan 搜索其它的映象。

万一前面尝试寻找 iso 映像失败，iso-scan 会询问您是否愿意进行完整的搜索。这次不只查看最上级目录，而是真正地贯穿整个文件系统。

如果 iso-scan 找不到安装 iso 映像，请重新启动返回到您以前的操作系统，检查映像文件名是否正确 (以 `.iso` 结尾)，是否放在一个能被 `debian-installer` 识别的文件系统上，是否损坏 (用 checksum 校验)。有经验的 Unix 用户可以在第二个控制台上完成这些动作，而无需重新启动。

注意，作为 ISO 映像主机的分区（或磁盘）不能在安装过程中再次使用，因为它将被安装程序使用。为了绕过这个限制，假设有充足的系统内存，安装程序能够在挂载前将 ISO 映像复制到内存中。这由低优先性的 `iso-scan/copy_iso_to_ram` debconf 问题（只在满足内存要求的时候询问）来控制。

6.3.1.5 配置网络

进入到这一步，如果系统检测到有多个的网络设备，您将被要求选择哪个设备是主网络接口，比如，在安装中使用的那个。其他的网络接口这时不会配置。您可以在安装完毕之后去配置额外的接口；参考 interfaces(5) man 手册。

6.3.1.5.1 自动网络配置

默认情况下，`debian-installer` 会尽力自动配置您的计算机网络。如果自动配置失败，会有多种因素造成，从没有插好网线到错误的自动配置环境都有。要了解更多的出错原因，请检查第四控制台上的信息。不管是什么情况，都会要求您再试一次，或者让您选择手动设置。有时用于网络自动配置的网络服务器响应比较慢，假如您确信都配置正确，可以简单地再重新进行一次自动配置。要是自动配置还是失败，您可以换成手动网络设置方式。

6.3.1.5.2 手动网络配置

手动配置网络会依次询问一些网络问题，特别是 IP 地址 (IP address)、网络掩码 (Netmask)、网关 (Gateway)、域名服务器地址 (Name server addresses)，以及主机名 (Hostname)。另外，如果您有无线网卡，会被要求提供 Wireless ESSID(“无线网络名”)和 WEP 密钥 (WEP key) 或者 WPA/WPA2 密码 (WPA/WPA2 passphrase)。填写答案第 3.3 节。

[3] 传统 locale 是那些没有使用 UTF-8 的 locale，它们使用旧的字符编码标准，比如 ISO 8859-1 (西欧语言) 或 EUC-JP (日语)。

注意

您也许需要将一些技术细节放在手边：程序假定网络 IP 地址是您系统的 IP 地址和网络掩码按位与。默认的广播地址是通过您系统的 IP 地址和按位反网络掩码按位或计算得来。它也会猜测您的网关。如果您不能回答这些问题，让系统猜测—如果需要，在系统安装完毕之后，通过编辑 `/etc/network/interfaces` 文件进行修改。

6.3.1.5.3 IPv4 和 IPv6

从 Debian GNU/Linux 7.0 (“Wheezy”) 以后，`debian-installer` 将同时支持 IPv6 和“经典的”IPv4。IPv4 与 IPv6 的各种组合 (仅 IPv4、仅 IPv6 和双栈 dual-stack 配置) 都将被支持。

IPv4 通过 DHCP (Dynamic Host Configuration Protocol) 进行自动配置。自动配置 IPv6 使用 NDP (Neighbor Discovery Protocol，包括 recursive DNS server (RDNSS) 设置) 支持无状态自动配置，有状态自动配置通过 DHCPv6 和混合自动配置 (地址配置使用 NDP，额外参数使用 DHCPv6)。

6.3.2 设置用户和密码

在设置时钟之前，安装程序允许您创建“root”帐号和/或第一个用户帐号。其他的用户帐号可以在安装完成之后创建。

6.3.2.1 设置 root 密码

root 帐户也被称为超级用户；系统中的所有的安全防护措施对以该身份登录的都是无效的。root 帐户应该仅用来进行系统管理，而且使用时间应该尽可能短。

您所创建的任何密码都应该包含至少 6 个字符，同时包含大小写字母，并且最好带有标点符号等特殊字符。因为超级用户具有最高权限，因此在您设置 root 密码时尤其需要小心。请避免采用能够在字典中查到的单词或者很容易猜测的个人信息。

如果他人向您索取您的 root 密码，也需要特别谨慎。除非您所管理的系统有多位管理员，否则通常不应该将超级用户密码交给别人。

在这里没有为“root”用户指定密码的情况下，这个账户会被禁止，但 sudo 软件包将被安装，后面在新的系统上允许执行管理任务。默认系统上新建的第一个用户使用 sudo 命令变成根用户。

6.3.2.2 创建一个普通用户

系统会询问您现在是否希望创建一个普通帐户。您将使用该帐户进行日常登录操作。切记，平时不要使用 root 帐户登录或者将其作为个人帐号使用。

为什么呢？避免使用 root 特权帐户的一个原因是，它很容易对系统造成无法挽回的破坏。另一个原因是，您有可能被恶意诱使运行特洛伊木马程序—这是一种利用超级用户权限在您未知的情况下损害系统安全的程序。任何合格的 Unix 系统管理书籍中都会涉及到这一主题—如果您不是很了解这方面的内容，建议找一本进行学习。

您首先会被要求输入用户的全名。然后要求输入用户帐号名；通常您的名字或者类似的便满足要求，并为缺省值。最后，您将要求输入该帐号的密码。

如果在安装完毕后还希望建立其它新帐户，请使用 adduser 命令。

6.3.3 配置时钟和时区

在 S/390 平台，安装程序不会修改系统时钟。

基于安装过程前面所选择的地理位置，您将看到系统只列出与其地点相关的时区列表。如果该地只对应一个时区并且是默认安装，那么系统就不会询问而直接使用该时区。

expert 模式或者使用 medium 优先级安装，您可以通过额外的选项选择“Coordinated Universal Time”(UTC) 作为时区。

如果出于某些因素您不想设置时区为所选择的地点，这里有两个选项。

1. 最简单的方法，是在安装完成后重新启动，进入新系统，选择一个不同的时区。该命令为：

```
# dpkg-reconfigure tzdata
```

2. 另外一种方法是，时区可以在安装启动后的安装程序最初阶段通过传入参数 `time/zone=value` 进行设置。其中的 value 当然要求是一个有效的时区，例如 `Europe/London` 或 `UTC`。

自动安装可以通过预置设置任何所需的时区。

6.3.4 分区与选择挂载点

现在，经过最后的硬件检测，`debian-installer` 应该准备就绪，按照用户定制，进行实质性的工作。正如本节标题指明，接下来组件的主要任务是为您的硬盘分区、创建文件系统、安排挂载点，还可以选择为 RAID、LVM 或加密设备这组选项进行配置。

要是您对分区还有心存疑虑，或者只是想多了解一些细节，请参阅附录 C。

首先，您可以选择对整块硬盘或是其上的空闲空间进行自动分区。这也被称为“导引式 (guided)”分区。如果不想使用自动分区，也可以从菜单中选择 Manual 手动分区。

6.3.4.1 支持的分区选项

`debian-installer` 里面使用的 paritioner 近乎万能。它可以采用不同的分区表、文件系统和高级块设备创建多种分区方案。

具体可以使用哪些选项主要取决于体系架构，一些其他的因素也对此有影响。例如，有的系统上内存大小限制了一些选项的使用。默认值也会相应变化。比如大容量的硬盘比小容量的硬盘默认使用的分区表就不同。一些选项只能在使用 medium 或 low 优先级安装下改变；高优先级下会采用默认值。

安装程序支持多种高级分区和存储设备，很多情况下可以组合使用。

- 逻辑卷管理 (LVM)
- 加密
- Multipath (experimental)

  参考我们的 Wiki 了解更多信息。对于 multipath 只在安装程序启动时打开才支持。

支持下列的文件系统。

- ext2, ext3, ext4

  多数情况下默认的文件系统选择的是 ext4。
- jfs (并不是所有架构都支持)
- xfs (并不是所有架构都支持)
- reiserfs (选项；并不是所有架构都支持)

  默认情况下 Reiser 文件系统不再支持。安装程序运行在 medium 或 low 优先级下时，可以通过选择 `partman-reiserfs` 组件打开。只支持文件系统版本 3。
- FAT16, FAT32

分区工具支持基于 10 的幂的单位（1 MB = 1000 kB = 1000000 B），也支持基于 2 的幂的单位（1 MiB = 1024 kiB = 1048576 B）。单位前缀支持到 EB/EiB（exabyte/exbibyte）。

6.3.4.2 导引式分区

f 如果您选择导引式分区，可以有三种选择：直接在硬盘上创建分区（经典方式），或者使用逻辑卷管理 (LVM) 来创建，或者使用加密 LVM 来创建。[4]。

> **注意**
>
>
>
> (加密) LVM 的选项并不是在所有体系下都可用。

[4]安装程序将使用 256 位的 AES 密钥来加密 LVM 卷组，并使用内核的“dm-crypt”支持。

使用 LVM 或加密 LVM，安装程序将在一个大分区上创建很多分区；这种方式的优点在于该大分区内的分区以后改变尺寸相对比较容易。如果没有一个特殊的密码，加密 LVM 的大分区将无法读取，对您的 (私有) 数据提供了额外的安全性。

使用加密 LVM 时，安装程序将随机写入数据自动清除磁盘。这提升了安全性 (使得无法获知磁盘的哪一部分被使用，并使此前的安装轨迹被清除)，根据磁盘的大小会耗费一些时间。

> **注意**
>
>
>
> 假如您用导引式分区选择 LVM 或加密 LVM，在 LVM 建立的时候，分区表中的一些变更需要写入到所选的磁盘。这些变更会删除当前硬盘上的数据，以后将无法撤销。当然，安装程序会在写入磁盘之前让您确认。

假如用导引式分区 (经典方式或 (加密)LVM) 选择整个磁盘，您会首先被要求选取将要使用的磁盘。检查一下是否所有的磁盘都被列出，如果有多个磁盘，并确认所选的是真正要使用的那块。排列的次序可能与以前的不同，但磁盘的尺寸将会协助您辨别它们。

所选磁盘上的所有数据将会丢失，但总会在写入磁盘前提示您进行确认。如果选择的是经典分区方式，您可以在整个过程中撤销任何修改；但使用 (加密) LVM 方式则不能撤销。

Next, you will be able to choose from the schemes listed in the table below. All schemes have their pros and cons, some of which are discussed in 附录 C. If you are unsure, choose the first one. Bear in mind that guided partitioning needs a certain minimal amount of free space to operate with. If you don’t give it at least about 3GB of space (depends on chosen scheme), guided partitioning will fail. Also, on small disks you will probably not see all of the schemes mentioned below.

分区方式	所需最小空间	所新建的分区
所有文件在同一分区	8GB	/, swap
建立 /home 分区	9GB	/, /home, swap
分别建立 /home、/var 和 /tmp 分区	12GB	/, /home, /var, /tmp, swap
Separate /srv and /var partitions; swap limited to 1GB (for server)	8GB	/, /srv, /var, swap
Scheme especially for small disks	3GB	/, swap

如果您选择使用 (加密) LVM 导引式分区，安装程序还会创建一个单独的 /boot 分区。其他的分区，除了交换分区，都会建在 LVM 分区之内。

在选定了分区方式后的下一个界面，您会看到关于自己分区的一张表格，上面标明了分区是否将被格式化，将以何种方式格式化，以及它们将被挂载到哪里的相关信息。

分区表格的格式如下：

```
SCSI1 (0,0,0) (sda) - 6.4 GB WDC AC36400L
      #1 primary   16.4 MB  B f ext2       /boot
      #2 primary  551.0 MB      swap       swap
      #3 primary    5.8 GB      ntfs
         pri/log    8.2 MB      FREE SPACE

SCSI2 (1,0,0) (sdb) - 80.0 GB ST380021A
      #1 primary   15.9 MB      ext3
      #2 primary  996.0 MB      fat16
      #3 primary    3.9 GB      xfs        /home
      #5 logical    6.0 GB    f ext4       /
      #6 logical    1.0 GB    f ext3       /var
      #7 logical  498.8 MB      ext3
```

本例中有两块硬盘，它们分别被分割成了几个分区。第一块硬盘上还有些空闲空间。表中每行列出了分区编号、类型、大小、选项标志、文件系统和挂载点 (如果有的话)。注意：导引式分区方式不能创建这些特殊的设置，但使用手动方式分区的结果会有所不同。

接下来，导引式分区就要完成了。如果您对上面生成的分区信息表感到满意，那么请在菜单中选择 Finish partitioning and write changes to disk 一项来创建新分区表 (本节末将会对此进行解释)。如果您

认为分区设置不如愿，那么可以选择 Undo changes to partitions，然后再次运行导引式分区，或者按照下面的介绍进行手动分区。

6.3.4.3 手动分区

如果您选择手动分区的话，它的界面和上面介绍的导引式分区大体上相同。不同之处在于，界面上显示的是您当前的分区列表，而且其中并没有挂载点的信息。关于如何手动设置您的分区表，以及新安装的 Debian 系统将如何使用这些分区，将会在本节的后面谈到。

倘若您选用的是一块全新的硬盘，它还没有被分过区，上面也没有空闲空间。那么系统会要求新建一个分区表 (只有这样，您才能创建新的分区)。分区表建好之后，在被选中的磁盘条目下出现一个新行，上面写着“FREE SPACE”。

如果您选中了某块空闲空间，那么就可以在上面建新分区了。接下来需要回答一系列简短的问题，它们会就分区大小、类型 (主分区还是逻辑分区)、以及分区的位置 (在空闲空间的开始部分还是在结束部分) 向您询问。回答完毕，您会看到一个新分区的小结。主要设置是 Use as:，它决定一个分区上是否会有文件系统，或被用作交换分区、软 RAID、LVM 和加密文件系统，或者根本就放着不用。其他的设置包括挂载点、挂载选项和启动标识；这些是否显示由分区的使用方式决定。如果您不喜欢预选的缺省设置的话，尽可按照自己的喜好更改。比如说，选中 Use as: 选项，然后您可以让这个分区改用其他的文件系统，把它用作交换分区、软 RAID、LVM，或者根本就放着不用。一旦您对新分区的设置感到满意了，就可以选择 Done setting up the partition，然后退回到 partman 的主界面。

若是您希望修改分区的设置，只要选中该分区，分区的配置菜单就会出现在您的面前。由于这个界面和新建分区时使用的界面是相同的，所以您可以像以前那样修改那些设置项。有一件事，可能第一眼看不大出来，就是您还可以通过选中显示分区大小的项目来调整分区的大小。已知适用的文件系统至少有 fat16、fat32、ext2、ext3 和 swap。在这个菜单中，您还可以删除分区。

请确保至少分出两个分区：其中一个作为 root 文件系统 (它必须挂载到 /)，另一个用于 swap。若是您忘记了挂载根 (root) 文件系统的话，partman 会拒绝让您继续下面的步骤，直到您纠正了这个错误。

由于 partman 的功能是通过安装本套件的各模块功能得以延伸和扩展的，但是具体又因您系统的体系架构而有所区别。因此，如果您发现安装的实际情况与我们所言不符，缺少了某些功能特性，那么请检查一下，确保已加载了所有必须的模块 (比如，`partman-ext3`、`partman-xfs` 或者 `partman-lvm`)。

在您对分区设置感到满意后，就可以选择分区菜单中的 Finish partitioning and write changes to disk 一项。在这之后，会出现一个清单，它列出了即将在硬盘上进行的所有操作。此时，安装程序会让您确认是否就照此分区。

6.3.4.4 配置多磁盘设备 (软 RAID)

如果您的计算机上有多个硬盘[5]，可以用 partman-md 配置硬盘以提升效能和/或更好的数据可靠性。这种结果称为多磁盘设备 (Multidisk Device)(或者更流行的 软 RAID)。

MD 本质上是一束位于部分磁盘上的分区，组合在一起形成一个逻辑设备。该设备可以像正常的分区一样使用 (比如可以用 partman 格式化，分配挂载点，等等)。

得到什么样的好处取决于您创建的 MD 类型。当前支持的类型有：

RAID0 它的主要目标是效能。RAID0 将进来的数据分割成 stripes 然后平均分配到组里面的每个硬盘上。这样可以提升读/写操作速度，但是一旦其中的一个硬盘损坏，您将丢失一切 (部分信息在好盘上，其他的曾经在坏盘上)。

典型的 RAID0 案例是应用在视频编辑的分区。

RAID1 适用于可靠性作为优先考虑的场合。它由多个 (通常两个) 相同尺寸的分区组成，每个分区容纳相同的数据。这意味着三件事。第一，如果其中的一个损坏，您仍然有数据镜像在其余的磁盘上。第二，您只能使用现有容量中的碎片 (更准确的说，它是 RAID 中尺寸最小的磁盘分区)。第三，文件读取在磁盘间负载平衡，这可以提升服务器的性能，如文件服务器，它倾向于读的负载大于写。

可选择的是，您可以用备用的磁盘放在组中，用于顶替事故中损坏的磁盘。

RAID5 这是一个速度、可靠性和数据冗余都不错的折衷方案。RAID5 将进来的数据分割成 strips (类似于 RAID0)，但只平均分配到一个磁盘上。与 RAID0 不同，RAID5 还会计算奇偶校验信息，这将会写入其他磁盘。奇偶检验磁盘不是静态的 (那被称为 RAID4)，会周期性的修改，所以奇偶校验信息平均分配到所有磁盘。当其中的一个磁盘损坏，丢失信息的部分可以从其他数据以及奇偶校验计算出来。RAID5 最少需要三个活动的分区。作为选项，您可以用备用的磁盘放在组中，用于顶替事故中损坏的磁盘。

[5]其实，您可以从单个物理硬盘上不同分区创建 MD 设备，但这样做不会给您带来任何好处。

如您所见，RAID5 近于 RAID1 的可靠性并具有较少冗余。另一方面，它只比 RAID0 在写入的时候慢，这是因为要计算校验信息。

RAID6 它与 RAID5 近似，但使用两个校验设备而不是一个。

RAID6 阵列可以容忍两个磁盘损坏。

RAID10 RAID10 组合了 striping (如 RAID0) 和 mirroring (如 RAID1)。它创建输入数据的 n 个拷贝，并将它们分配到不同的分区，使得相同数据的拷贝不会放在相同的的设备上。默认的 n 值为 2，但可以在专家模式 (export mode) 下进行设置。分区数至少是 n。RAID10 有不同的分配拷贝布局 (layout)。默认为 near copies。near copies 有所有磁盘上相同 offset 的所有拷贝。far copies 有磁盘上不同 offset 的拷贝。offset copies 复制 stripe，而不是整个拷贝。

RAID10 能可靠且冗余地存储，而没有计算校验的缺点。

总结：

类型	最少设备	备用设备	幸免于磁盘损坏？	可用空间
RAID0	2	否	否	容量为最小分区乘以 RAID 设备数
RAID1	2	可选	是	容量为 RAID 最小分区
RAID5	3	可选	是	容量为最小分区乘以 (RAID 设备数量减一)
RAID6	4	可选	是	容量为最小分区乘以 (RAID 设备数量减二)
RAID10	2	可选	是	所有的分区数除以 chunk 拷贝数 (默认为二)

如果您想真正全面了解软 RAID，看一下 Software RAID HOWTO。

为了创建 MD 设备，您需要将期望的分区标记为供 RAID 使用。(通过 Partition settings 菜单的 partman 完成，您应该选择 Use as: → physical volume for RAID。)

注意

请确保系统可以从您规划的分区引导。当使用 RAID 作为根 (/) 文件系统时，通常需要创建一个单独的文件系统给 /boot。多数的 boot loaders 支持 mirrored (而非 striped !) RAID1，所以例如使用 RAID5 给 / 同时 RAID1 给 /boot 是一种方式。

下一步，您应该从 partman 主菜单选择 Configure software RAID。(菜单只会在您用 physical volume for RAID 标记了至少一个分区时出现。) 在 partman-md 第一个画面选择 Create MD device。您将看到被支持的 MD 设备列表，您应从其中选择一项 (如 RAID1)。后续操作会根据您选择的 MD 类型而定。

- RAID0 是简单的—您会看到可用的 RAID 分区，然后您的任务仅是选择那些想要组成 MD 的分区。
- RAID1 需要一些技巧。首先，您将要求输入组成 MD 的活动设备和备用设备数量。其次，您需要从 RAID 可用分区列表中选择哪些是活动分区，哪些是备用的。选择的分区总数必须与之前提供的数目相同。别担心，如果您出错选择了不一样的分区数，debian-installer 将不会允许您继续下去，直到纠正错误。
- RAID5 的配置过程类似于 RAID1，只是您至少需要三个活动分区。
- RAID6 的配置过程类似于 RAID1，只是您至少需要四个活动分区。

- RAID10 除了专家 (export mode) 模式之外也和 RAID1 的设置过程相似。在专家模式下，debian-installer 将询问您布局 (layout)。布局由两部分组成。第一个部分是布局类型。可以是 n (对应 near copies)、f (对应 far copies) 或 o (对应 offset copies)。第二部分是组成数据的拷贝数。这些活动设备必须是让所有的拷贝可以分布到不同磁盘。

完美的解决方案或许是同时使用不同的 MD 类型。例如，您有三块 200 GB 的硬盘打算用于 MD，每块包含两个 100 GB 的分区，您可以将三块硬盘上的第一组分区组成 RAID0 (高速的 300 GB 视频编辑分区)，其余的三个分区 (2 个活动 1 个备用) 用于 RAID1 (相当可靠的 100 GB 分区用于 /home)。

按您需要配置 MD 设备之后，您可以 Finish partman-md 返回到 partman，去创建文件系统到您的新 MD 设备，并分配挂载点这样的属性。

6.3.4.5 配置逻辑卷管理 (LVM)

如果您做计算机系统管理员工作或者“高级”用户，一定遇到过磁盘分区 (经常是最重要的那个) 空间不足，同时其他的分区却不能平衡使用，然后不得不靠移到或符号链接等方法来折腾。

为了避免上面描述的情况，您可以采用逻辑卷管理 (LVM)。简而言之，使用 LVM 可以组合您的分区 (物理卷 (physical volumes)，LVM 术语) 形成一个虚拟盘 (称为卷组 (volume group))，它可以被分割成虚拟分区 (逻辑卷 (logical volumes))。逻辑卷 (当然底层是卷组) 的亮点在于它可以跨越多个物理磁盘。

当您认觉得需要更多的空间给已有的 160 GB 的 /home 分区时，只需加一个新的 300GB 磁盘到计算机，添加进已经存在的卷组，然后为 /home 分区的逻辑卷重新设置尺寸，瞧—您的用户在更新的 460GB 分区上又有了空间。这个例子当然过于简单。如果您还还没有读过，应该查阅 LVM HOWTO。

debian-installer 中的 LVM 设置很简单，并且在 partman 中被完全支持。首先，对于 LVM 必须将使用的分区标记为物理卷。这在 Partition settings 菜单里完成，在那里应该选择 Use as: → physical volume for LVM。

警告

注意：新的 LVM 设置将损坏由 LVM 类型编码标记的所有分区。所以，如果如果在您的一些盘上已经有 LVM 了，并想要另外安装 Debian 在那台机器上，那么旧的 (已经存在的) LVM 将被擦除！有些分区 (由于任何原因) 被错误地标记了 LVM 类型编码，但包含了一些不同的 (像加密的卷)，这对它们也很重要。在执行新的 LVM 设置前，需要从系统中删除这样的盘！

当您返回 partman 主界面，将看到一个新选项 Configure the Logical Volume Manager。选择它，首先会被要求确认修改分区表 (如果有的话)，然后 LVM 配置菜单才显示出来。菜单上部显示了 LVM 配置的小结。菜单项本身是与内容相关，只显示可用的操作。可能的操作有：

- Display configuration details: 显示 LVM 设备结构，名称和逻辑卷尺寸等
- Create volume group
- Create logical volume
- Delete volume group
- Delete logical volume
- Extend volume group
- Reduce volume group
- Finish: 返回 partman 主界面

使用该菜单内的选项首先创建卷组，然后在里面创建逻辑卷。

返回到 partman 的主界面之后，可以看到新建的逻辑卷与其他普通的分区一样 (您也应该这样对待它们)。

6.3.4.6 配置加密卷

debian-installer 允许您建立加密分区。写入该分区的每个文件都立即存为加密形式。只有在输入 passphrase 通过认证之后才能访问加密后的数据，口令是在加密分区创建时设置的。该特性用于保护您被盗的便携机或硬盘上的敏感数据。盗贼或许可以访问硬盘，但是没有正确的口令，硬盘上的数据看起来就像是随机的字符。

有两个需要加密的重要分区：home 分区，驻留您的私人数据，以及交换分区，敏感的数据在操作时会临时保存在这里。当然，这不影响您加密任何其他的分区。例如，数据库服务器、邮件服务器或打印服务器上用来保存数据的 /var。或者 /tmp，很多程序用来保存临时文件。有些人或许需要加密整个系统。通常这里唯一的例外是，/boot 分区必须保持未加密，这是因为以前没有办法从加密分区加载内核。(GRUB 现在可以这样做了，但 debian-installer 当前缺少对加密的 /boot 的内在支持。这样，设置在 separate document中覆盖。)

> **注意**
>
>
>
> 请注意，加密分区的性能将低于未加密的状态，这是由于数据在每次读写时都要进行加密和解密。性能将与您的 CPU 速度、选择的加密算法和密钥长度紧密相关。

为了使用加密，您必须在主分区菜单选择空闲的空间创建一个新分区。另一种是使用现有的分区 (例如，普通的分区、一个 LVM 逻辑卷或者 RAID 卷)。在 Partition setting 菜单，您需要选择 physical volume for encryption，其位于 Use as: 选项。菜单将包含分区的加密选项。

由 debian-installer 支持的加密方法是 dm-crypt (包括在较新的 Linux 内科中，能够作为 LVM 物理卷的主机)。

让我们来看看通过 Device-mapper (dm-crypt) 选择加密方法后的可用选项。一般规律：如果不能确定，就使用默认项，因为它们是经过审慎的安全考虑的选择。

Encryption: aes 该选项让您选择加密算法 (cipher)，它用来加密分区上的数据。debian-installer 当前支持以下 block ciphers：aes、blowfish、serpent 和 twofish。讨论这些算法的质量超出了本文的范畴，但是，您应该知道，在 2000 年，AES 被 American National Institute of Standards and Technology 选择作为标准的加密算法，用于保护 21 世纪的敏感信息。

Key size: 256 这里您可以指定密钥长度。使用较长的密钥，加密的能力自然得到提升。另一方面，增加长度自然会降低性能。cipher 决定可用的密钥长度。

IV algorithm: xts-plain64 Initialization Vector 或 IV 算法是确保加密过程中用同一个密钥对于同一 clear text 将产生惟一的 cipher text。用于防止攻击者根据加密数据中的重复模板推断信息。

与其替代者比较，默认的 xts-plain64 是当前已知的具有最少的攻击漏洞。您只有在需要保持与以前安装的系统兼容，而它们无法使用新的算法时，才使用其他的算法。

Encryption key: Passphrase 这里您可以为分区选择所使用的加密密钥类型。

Passphrase 加密密钥将基于 passphrase 计算[6]，使用时要求输入。

Random key 当您每次使用加密分区的时候，会从随机数据中生成一个新的加密密钥。换句话说：每次关机驻留在内存里面的密钥删除以后分区上的内容将丢失。(当然，您会猜想该密钥将面临暴力破解，但是除非加密算法中有未知的弱点，这辈子也不会被攻破。)

随机密钥用于交换分区，这样您就不必为记住 passphrase 或关机前清除交换分区上的敏感信息而费神。然而，这也意味着您将无法使用新 Linux 内核提供的“suspend-to-disk”功能，因为它将无法 (在下次启动时) 恢复写入到交换分区的数据。

Erase data: yes 决定在建立加密前是否使用随机数据覆盖分区内容。推荐这种方式，否则攻击者将会发现分区的哪些部分使用而哪些没有使用。另外，这将使安装前的内容很难恢复[7]。

当您选择好加密分区参数之后，返回分区主菜单。现在会有一个新的菜单项 Configure encrypted volumes。选择它，您会被要求确认删除所标记分区上的数据，以及写新分区表等动作。对于大的分区，这会花一些时间。

[6] 使用 passphrase 作为密钥意味着建立分区是使用 LUKS。

[7] 据信那些三个缩写字母部门的家伙可以恢复经过多次覆盖的磁记录设备上的数据。

接下来，您会被要求输入 passphrase 用于配置的分区。好的口令应该长于 8 个字符，并混合字母、数字和其他字符，而不应该是普通词典中的单词或者容易与您相关的信息 (比如生日、爱好、宠物的名字、以及家里其他成员的名字，等等)。

警告

在输入任何 passphrase 之前，您应该确定键盘配置正确可以输入所需要的字符。如果您不能肯定，可以切换到第二虚拟控制台，在提示符下敲一些文字。这能保证您以后不必惊慌失措，例如，想要使用 qwerty 键盘输入的 passphrase，而在安装时却使用的是 azerty 布局。造成的原因有多种。也许是您在安装过程中改变了键盘布局，或者是所选择的键盘布局未能正确建立，而且是在为根文件系统输入 passphrase 的时候。

如果您选择 passphrase 之外的方法创建加密密钥，它们现在就会生成。因为内核在安装的早期阶段还没有足够的熵，过程会长一些。您可以产生一些熵以加快速度：例如，随机的按下一些键，或者切换到第二虚拟控制台的 shell 产生一些网络或磁盘的动作 (下载一些文件，将一些大文件送到 `/dev/null` 等等)。这个过程在每个要加密的分区重复进行。

返回主分区菜单后，您将看到所有的加密卷作为额外的分区，它们可以像其他的普通分区一样配置。下面例子中显示了通过 dm-crypt 加密的卷。

```
Encrypted volume (sda2_crypt) - 115.1 GB Linux device-mapper
     #1 115.1 GB  F ext3
```

现在可以给卷设置挂载点，如果默认的文件系统类型对您不合适，可以进行修改。

要留意括号里的标识 (本例中 `sda2_crypt`) 和您为每个加密卷所设置的加载点。启动新系统的时候您会需要该信息。关于普通引导过程和使用加密后引导过程之间的差异，将在后面的第 7.2 节说明。

当您满意分区之后，可以继续安装过程。

6.3.5 安装基本系统

尽管这一阶段少有问题，但却需要大量时间用于整个基本系统的下载、校验和解包。如果您用较慢的计算机或网络，这要花费好一会儿时间。

在基本系统安装期间，软件包解包和配置信息会输出到 `tty4`。您可以通过按下左 Alt-F4 访问该终端；按下左 Alt-F1 返回主安装进程。

在安装程序通过串口控制台执行的情况下，基本系统安装时解包/配置信息保存在 `/var/log/syslog`。

作为安装的一部分，Linux 内核也要安装。在默认的优先级下，安装程序会选择一个与您硬件最匹配的内核。在较低的优先级下，您可以从列表中选择一个有效的内核。

在使用包管理系统安装软件包时，将默认安装那些软件包推荐的软件包。这些推荐的软件包并不影响要安装软件包的核心功能，但可以增强其功能。从软件包维护人员的角度看，正常情况下应该安装这些软件。

注意

由于技术的原因，安装基本系统时所安装的软件包并不会安装对应的“推荐软件包”。上面提到的原则只适用于之后的安装过程。

6.3.6 安装额外的软件

现在，您拥有了一个可用但有限的系统。大多数用户会安装额外的软件，并根据他们的需求进行调整，安装程序允许您这么做。如果您的计算机或网络比较慢，这一步或许比安装基本系统还要花时间。

6.3.6.1 配置 apt

在 Debian GNU/Linux 系统上，apt 是用来安装软件包的一个程序，它被包含在 apt 软件包中[8]。其他的包管理前端程序，像 aptitude 和 synaptic 也在使用。这些前端软件推荐给新用户使用，因为它们在良好的用户界面下集成了一些其他特性 (搜索包与状态检验)。

您必须设置 apt，使它明白该从哪里获取软件包。设置的结果写入到 `/etc/apt/sources.list` 文件里面，安装完成后您可以检查并编辑成自己喜欢的地方。

如果您在默认优先级下进行安装，安装程序会自动处理大部分配置，它是根据您使用的安装方法以及在安装开始时所做的选择。大多数情况下安装程序会自动加上安全镜像，而且如果您安装的是稳定版，还会加上“stable-updates”服务的镜像。

如果您采用较低的优先级安装 (比如专家模式)，就要自己做更多的决定。您可以选择是否使用安全更新、或者 stable-updates 更新服务，并可以选择添加软件仓库中“contrib”、“non-free”和“non-free-firmware”分区的软件包。

6.3.6.1.1 从多个 DVD 映像来安装

假如从 DVD 映像来安装，而它只属于所要安装程序的一部分，安装程序会询问您是否需要检测其它的安装介质。如果您手头有这样的介质，检测后安装程序就可以使用它们里面的软件包了。

假如您没有这些介质，也没关系：它们不是必须的。如果您也不使用网络镜像 (下节说明)，意味着下一步中选择的任务里面并非所有的软件包都会安装。

> **注意**
>
>
>
> DVD 映像中包含的软件包按照流行程度排列次序。也就是说，一组映像里面的第一个映像是最常用的软件，而最后一个映像包含的是很少有人用到的软件。
>
> 这也意味着购买或者下载刻录所有 DVD 是一种浪费，因为大部分光盘您根本不会用到。多数情况下，只需要使用前 1 到 2 张 DVD，然后从网络镜像安装其他要用的软件包。

如果您检测了多个安装介质，在需要的映像当前不在驱动器中时，安装程序会提示您换盘。注意，只有同属于一个套件的光盘需要检测。检测的次序并不是非常重要，但按照次序来可以降低出错的几率。

6.3.6.1.2 使用网络镜像

多数安装下会有一个问题被问到，是否需要使用网络镜像作为软件包的源。多数情况下默认答案会很好，但还是有一些例外。

如果没有使用 DVD 映像，您就应该使用网络镜像，否则只能得到一个有限的系统。然而，假如您有互联网连接限制，在安装过程的下一步最好不要选择 desktop 任务。

如果您从 DVD 安装，安装过程中需要的软件包都应该在第一张 DVD 映像里面。网络镜像的使用是一个选项。

添加网络镜像的优势在于有些内容已经更新，而 DVD 映像只是在版本发布时创建，所以可以安装已经更新的内容，确保了系统的安全性和稳定性。

总结：除非您不具备良好的互联网连接，选择网络镜像通常是一个好主意。如果安装介质上有软件包当前的版本，安装程序总会从这里安装。如果您选择了镜像，需要下载的数据量取决于

1. 安装下一步选择的任务，
2. 这些任务所需的软件包，
3. 已检测的安装介质上的软件包，并且
4. 安装介质上的软件包在网络镜像里面是否有更新版本 (普通软件包的网络镜像，或者安全或稳定更新的镜像)。

注意，最后一点意味着，即使您没有选择使用网络镜像，如果配置了安全检测服务，并且软件包有安全更新或稳定更新，仍然会从互联网下载更新后的软件包。

[8] 其实，真正用来安装软件包的程序是 dpkg。但这是一个比较底层的工具。apt 是一个高级工具，它可以在适当的时候调用 dpkg。它知道从您的安装介质、网络或其它地方获取软件包。它还可以在安装某个软件包时自动安装所需的其它软件包。

6.3.6.1.3 选择网络镜像

除非选择不使用网络镜像，根据前面的安装步骤中您选择的国家，会显示网络镜像的列表。通常选择默认的就行。

默认的是 deb.debian.org，这不是镜像自身，但将会重定向最新与最快的镜像。这些镜像支持 TLS (https 协议) 和 IPv6。这个服务由 Debian 系统管理 (DSA) 团队维护。

可以通过选择“enter information manually”来说动指定镜像。可以指定镜像主机名称和可选的端口号。这实际上必须基于 URL ，也就是当指定 IPv6 地址时，必须添加方括号将它包起来，例如“[2001:db8::1]”。

如果您的计算机处于仅有 IPv6 的网络 (大多数用户不会遇到这种情况)，使用所属国家默认的镜像可能无法正常工作。列出的网络镜像全部可以使用 IPv4 访问，但只有其中的某些可以通过 IPv6 访问。由于个别镜像连接的状况会不断更新，此信息可能在安装程序里面失效。如果还没有 IPv6 连接到所在国家的默认镜像，您可以试试其他的镜像，或者选择“手动输入信息”选项。您可以指定“ftp.ipv6.debian.org”作为镜像名，这是 IPv6 上的一个镜像别名，它有可能不是速度最快的哪个。

6.3.6.2 选择和安装软件

安装过程中，您有机会选择安装额外的软件包。不像从个软件包里面里面选择单独的软件包，这一阶段安装过程着眼于安装一组预先定义的软件包，来快速建立您的计算机应对各种任务。

任务松散地代表了您要在计算机上从事的一系列工作，比如“桌面环境”、“Web 服务器”或者“SSH 服务器”[9]。第 D.2 节列出了各种任务所需的空间。

注意，根据所安装计算机的特征，有些任务被预先选择。如果您不需要它们，可以取消这些任务。此时甚至可以不选中任何任务。

提示

在安装程序的标准用户界面下，您可以使用空格键切换任务的选择。

注意

“Desktop environment”任务将安装图形桌面环境。

默认 `debian-installer` 安装桌面环境。能够在安装过程中交互地选择不同的桌面环境。还能够安装多个桌面，但一些桌面的组合可能不能同时安装。

注意，这只在该桌面环境对应的软件包可以获得的情况下有效。假如您使用 DVD 映像或其他通过网络映像的安装方法，安装任何可用的桌面环境时这种方法应该有效工作。

各种服务器任务大致安装的软件如下。Web 服务器：`apache2`；SSH 服务器：`openssh`。

“Standard system utilities”任务会安装具有“standard”优先级的软件包。这包括很多在任何 Linux 和 Unix 系统上常用的工具。除非清楚自己的目的并需要一个最小化的系统，您应该让该任务选中。

选择语言的时候，如果默认的 locale 不是“C”，tasksel 会检查该 locale 是否有对应的本地化任务，然后自动安装相应的本地化软件包。这里面包含诸如对应语言的词汇列表或特殊字体的软件包。假如选择了桌面环境，(如果存在) 还会安装该桌面对应的本地化软件包。

[9]您应该知道，呈现这个列表，安装程序仅仅是执行 tasksel 程序。它可以在安装之后的任何时间安装 (或删除) 其他软件包，或者使用更精致的工具，例如 aptitude。如果您需要一个特定的软件包，在安装之后，只须执行 `aptitude install package`，其中 `package` 是您需要的软件包名。

注意

Starting with Debian Trixie, `debian-installer` allows to install Debian Pure Blends directly. There is an entry "Choose a Debian Blend for installation" at the bottom of the tasks list, which enables an additional choices dialog to appear, after tasksel has finished.

In that dialog (blendsel), you can choose which blend to install, similar to this tasks selection.

See the Debian website for more information about Debian Pure Blends.

一旦您确定了任务，请选择 Continue。此时，apt 将安装选中任务的软件包。假如某个程序需要用户输入信息，它将提示您指定。

您应当清楚安装桌面的任务非常巨大。特别是当使用网络安装 CD 映像，同时通过网络镜像安装网络安装 CD 上没有的软件包时，安装程序将通过网络下载大量的软件包。如果您的互联网接入较慢，这将花很长时间。一旦启动，安装软件包过程将无法取消。

如果镜像服务器上的软件包版本更新，即使旧版本在安装映像上存在，安装程序还是会从镜像上获取这些软件包。如果您安装的是稳定版，这会出现在次版本发布之后（主稳定版本的更新）；如果您安装的是测试版本，这会出现在使用旧映像安装时。

6.3.7 创建启动系统

6.3.7.1 检测其他的操作系统

在 boot loader 安装之前，安装程序会试着侦测已经安装到计算机上的其他操作系统。如果它找到支持的操作系统，您将在启动 boot loader 安装步骤里得到提示。与 Debian 一起，计算机也将配置为可以启动其他操作系统。

注意，从单机上启动多个操作系统仍然是黑色艺术。自动支持检测和设置 boot loader 启动其他操作系统，会依系统甚至是子系统而变化。如果它不能工作，您应该参考启动管理器的文档了解更多信息。

6.3.7.2 不使用 boot loader 继续进行

该选项用于即使没有安装 boot loader 时完成安装，这种情况要么是体系/子体系不支持，要么是因为不想要（比如，使用现有的 boot loader）。

如果您打算手动设置 bootloader，需要检测安装的内核，它位于 `/target/boot`。您还需要检测 initrd 存在的目录；如果有，也许需要指导 bootloader 使用它。其他需要的信息，包括您为 / 文件系统选择的磁盘和分区，并且，如果您打算安装 `/boot` 到一个独立的分区，还需要 `/boot` 文件系统。

6.3.8 完成安装

这是在您 Debian 安装过程的最后一步。在这最后的几分钟工作里，主要进行一些 `debian-installer` 之后的清理。

6.3.8.1 重新启动系统

最后一个提示信息后，由于重启不被 S/390 支持，系统将关机。您需要 DASD 上的 IPL Debian GNU/Linux，它在安装步骤第一步里被选做根文件系统。

6.3.9 故障处理

这里的组件通常不在安装进程之内，但会在后台待命，以帮助用户遇到麻烦时处理问题。

6.3.9.1 保存安装记录

如果安装成功，安装过程中创建的记录文件，会被自动保存在您的新 Debian 系统上的 `/var/log/installer/`。

从主菜单上选择 Save debug logs，允许您将记录文件保存到 U 盘上、网络、硬盘或其他介质。这可用于安装过程中遭遇严重错误的时，您想在其它系统上研究记录，或者用于报告的附件。

6.3.9.2 使用 Shell 查看记录

安装过程中您可以使用多种方法获得一个 shell。多数系统上，如果您不是通过串口控制台安装，最简单的方法是按下左 Alt-F2 切换到第二个虚拟控制台[10]（在 Mac 键盘上，是 Option-F2）。用左 Alt-F1 可以切换回安装程序。

如果您无法切换控制台，还可以在主菜单上执行 Execute a Shell 选项，它可以启动一个 shell。您可以使用一次或多次 Go Back 按钮从大多数对话框返回主菜单。键入 `exit` 关闭 shell 并回到安装程序。

因为这时您是从 RAM 盘启动，只有有限的 Unix 工具可以使用。您可以通过命令 ls /bin /sbin /usr/bin /usr/sbin 和键入 help 查看哪些程序可以使用。其 shell 称为 ash，是 Bourne shell 的克隆，有一些如自动完成与历史纪录这样的好特性。

要编辑和查看文件，请使用文本编辑器 nano。安装系统的日志可以在 `/var/log` 目录找到。

注意

虽然在 shell 里面您可以用存在的命令做任何事情，但 shell 真正的用途只是遇到问题时进行调试。

在 shell 里面手动操作可能会影响安装过程，造成错误或无法完成安装。特别是，您应该让安装程序激活交换分区，而不是在 shell 里面自己动手。

6.3.10 通过网络控制台（network-console）安装

在 S/390 上安装，建立网络之后这是默认的方法。

您会被问到新密码，以及确认密码用于连接到安装系统。这些就够了。现在您应该看到一个画面，提示使用 installer 用户名和刚才您提供的密码登录。另外一个屏幕上需留意的重要细节，是该系统的 fingerprint。您需要安全地传输该 fingerprint 给那些要继续远程安装的人。

当您决定继续本地安装时，不断地按 Enter，就可以返回主菜单，之后可以选择其他的组件。

现在转到网线的另一端。有一个先决条件，您需要配置您的终端使用 UTF-8 编码，因为安装系统使用这种编码。如果不这样做，远程安装也可以进行，但是您可能看到奇怪的显示，比如破碎的对话框边界，或者不可读的非 ASCII 字符。与安装系统建立连接只需键入:

```
$ ssh -l installer install_host
```

其中 `install_host` 是安装计算机的名字或 IP 地址。正式登录之前，会显示远程系统的 fingerprint，您需要确认是否正确。

注意

安装程序中的 ssh 服务器使用的是默认的配置，并不发送 keep-alive 包。原则上，已安装系统的连接应该无限期的保持打开。但是，某些情况下—由您的局域网设置决定—一段时间的不活动，连接将会断开。这种情况常发生在客户端和系统上安装有 Network Address Translation (NAT) 的地方。基于连接在安装过程中的哪一点断开，您在连接之后可能继续，也可能无法继续安装过程。

您可以通过在启动 ssh 连接时添加 `-o ServerAliveInterval=value` 选项来避免断开，或者把它加到 ssh 的配置文件里面。注意，某些情况下添加该选项反而会造成断开 (例如，如果 keep-alive 包在网络故障 (brief network outage) 期间送出，ssh 将会恢复)，因此它只在需要的时候采用。

[10] 就是：按下位于空格键左手边的 Alt 键，同时按下 F2 功能键。

注意

如果您依次安装多台计算机，他们碰巧使用相同的 IP 地址或主机名，那么 ssh 将拒绝连接。造成的原因是它具有不同的 fingerprint，这通常是欺骗攻击的标志。如果您确信不是那种情况，需要删除 `~/.ssh/known_hosts` 里面相关的行[a]，再重新来过。

[a] 下面的命令将删除一个主机的已有记录: ssh-keygen -R <hostname|IP address>.

登录之后您将看到一个初始画面，显示有两个可能，Start menu 和 Start shell。前一个将带您到安装主菜单，那么您可以像普通方式一样安装。后一个打开一个 shell，您可以检查修复远程的系统。在安装菜单只能打开一个 SSH 会话，但 shell 可以打开多个。

警告

使用 SSH 开始远程安装之后，您不该返回本地控制台运行的安装会话。这将破坏新系统配置的数据库。结果可能导致安装失败或者安装完成的系统出现问题。

6.4 个性化配置

在 shell 的帮助下（参见第 6.3.9.2 节），安装过程可以个性化进行配置以满足特别的使用需求：

6.4.1 安装替代的初始化系统

Debian 使用 systemd 作为默认的初始化系统。不过，也支持其他的初始化系统（例如 sysvinit 和 OpenRC），并且选择替代的初始化系统的最佳时机是在安装过程中。欲了解详细的操作步骤，请阅读 Debian 维基的初始化系统页面。

Chapter 7

引导进入新的 Debian 系统

7.1 关键时刻

当您的系统首次依靠自身引导，这在电子工程师圈子里面称为“冒烟测试”。

如果系统无法正确启动，不要惊慌。如果安装是成功的，很可能只是相对次要的问题阻止了系统引导 Debian。在多数情况下不需重新安装就能修复问题。修复引导问题的一种方法是使用安装程序内置的救援模式（请参见第 8.6 节）。

假如您刚接触 Debian 和 Linux，可能需要从更有经验的用户那里得到一些帮助。对于不常见的架构，比如 S/390，最适合的方式是在 debian-s390 mailing list（Debian 相应架构的邮件列表）上请教。也可以如第 5.3.4 节中所描述的去提交安装报告。请确保将问题描述清楚，并包括所显示的任何信息，从而有助于其他人诊断问题。

7.2 挂载加密的卷

如果在安装时创建了加密的卷，并为其指定了挂载点，那么在引导的时候会要求为每个这样的卷输入密码。

对于使用 dm-crypt 来加密的分区，引导的时候会显示下面的提示信息：

```
Starting early crypto disks... part_crypt(starting)
Enter LUKS passphrase:
```

在提示的第一行，part 是底下分区的名称，例如：sda2 或 md0。您现在也许想知道具体是 为哪一卷真正输入密码。它是与 /home 相关？还是与 /var 相关？当然，如果只有一个加密的卷，那么这很容易，只需在设置这个卷时输入使用的密码就可以了。如果在安装过程中设置了多个加密的卷，那么在上一步第 6.3.4.6 节中写下的笔记会有用。要是之前没有记下在 part_crypt 与挂载点之间的映射的笔记，那么仍然可以在新系统的 /etc/crypttab 和 /etc/fstab 里面找到它。

加密的根文件系统挂载的时候，提示信息或许会有些不同。这取决于使用了哪个 initamfs 生成程序，来生成用于引导系统的 initrd。下面的示例使用了 initramfs-tools 来生成 initrd：

```
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Enter LUKS passphrase:
```

输入密码时不会显示任何字符（甚至星号）。如果密码输错，那么还有两次机会来纠正。第三次输错后，引导过程将跳过该卷，并继续挂载下一个文件系统。更多信息请参见第 7.2.1 节。

在输入所有密码之后，引导过程应该像通常一样继续。

7.2.1 故障处理

如果由于输错密码而造成无法挂载加密的卷，那么不得不在引导后手动挂载。有下面几种情况。

- 第一种情况与根分区相关。当不能正确挂载时，引导过程将中止，而您不得不重新引导计算机并再次尝试。
- 最简单的情况是保存诸如 /home 或 /srv 数据的加密的卷。引导后可以简单地手动挂载它们。

  然而对于 dm-crypt，这需要一些技巧。首先，需要用 device mapper 来注册卷，这需要运行：

```
# /etc/init.d/cryptdisks start
```

这将扫描 /etc/crypttab 里面提到的所有卷，并在输入正确的密码之后，在 /dev 目录下面创建对应的设备。(已注册的卷会被跳过，因此可以放心地多次运行该命令。) 正确注册以后，就可以按一般的方式来挂载这些卷了：

```
# mount /mount_point
```

- 如果任何保存着非关键系统文件 (/usr 或 /var) 的卷不能挂载，那么系统仍然应该能够引导，并仍然可以像前一种情况那样手动挂载那些卷。但是需要将默认运行级别下的任何服务 (重新) 启动，因为它们很可能没有启动。最简单的做法只是重新引导计算机。

7.3 登录

一旦系统引导后，就会看到登录提示符。使用您在安装过程中选择的用户名和密码来登录。现在系统就能使用了。

如果您是新用户，开始使用时会想要浏览已经安装到系统上的文档。当前有多个文档系统，而将不同类型文档集成的工作仍在继续。这里是一些起点。

与已安装的程序相伴的文档可以在 /usr/share/doc/ 里找到，在该程序 (或者更准确地说，包含该程序的 Debian 软件包) 同名的子目录下。但范围更广的文档经常会另外打包成一个特殊的文档软件包，而文档软件包多数时候并默认不安装。例如，关于软件包管理工具 apt 的文档放在 apt-doc 或 apt-howto 软件包内。

另外，还有一些特殊目录位于 /usr/share/doc/ 层级中。Linux HOWTOs 以 .gz (压缩) 格式安装在 /usr/share/doc/HOWTO/en-txt/ 中。在安装 dhelp 之后，可以在 /usr/share/doc/HTML/index.html 中找到可浏览的文档索引。

使用基于文本的浏览器来查看这些文档的一种简单方法，是输入下面命令：

```
$ cd /usr/share/doc/
$ w3m .
```

在 w3m 命令后的点符号告知要显示当前目录的内容。

如果安装了图形桌面环境，那么也可以使用它的 web 浏览器。从应用程序菜单中启动 web 浏览器，并在地址栏内键入 /usr/share/doc/。

也可以键入 info command 或 man command，去查看命令提示符下可用的多数命令。键入 help 将显示 shell 命令的帮助。键入命令，后面再加上 --help，这通常会显示命令用法的简短摘要。如果命令的结果滚动超出屏幕顶端，那么命令后面加上 | more，则可以在滚动超出屏幕顶部之前将结果暂停。要查看以某字母开始的所有可用命令的列表，则键入该字母，然后按两次 tab 按键。

Chapter 8

下一步该干什么

8.1 关闭系统

关闭一个运行着的 Debian GNU/Linux 系统，不要使用计算机前面或后面的 reset 开关重启，或者直接关闭计算机。Debian GNU/Linux 应该使用可控的过程关闭，否则会造成文件丢失或/和磁盘损坏。假如您运行了桌面环境，通常会有一个“log out”选项位于应用程序菜单让您关闭 (或者重启) 系统。

或者，您也可以按组合键 Ctrl-Alt-Del 。如果组合键不起作用，最后的办法是以 root 用户身份登录并键入必要的命令。使用 reboot 重启系统。使用 halt 在不关闭电源的情况下停止系统 [1]。要关闭计算机电源，请使用 poweroff 或 shutdown -h now 。systemd init 系统提供了执行相同功能的其他命令，例如 systemctl reboot 或 systemctl poweroff。

8.2 让自己适应 Debian

Debian 和其它的发行版有些不同。即使您熟悉其它发行版的 Linux，您也应该了解 Debian 的一些东西，以便保持您的系统处在良好整洁的状态。本章包的材料可以帮助您适应；但这并不是一个如何使用 Debian 的入门指南，而仅仅是对整个系统的一个粗略的介绍。

8.2.1 Debian 软件包系统

要掌握的最重要概念是 Debian 的软件包系统。事实上，您系统的绝大部分都应该由软件包系统来控制。包括：

- `/usr` (除了 `/usr/local`)
- `/var` (您可以创建 `/var/local`，在那里面进行安全的操作)
- `/bin`
- `/sbin`
- `/lib`

例如，假设您替换了 `/usr/bin/perl`，它可以正常工作，但以后升级您的 `perl` 软件包，那里的文件将被替换。有经验的用户可以在 aptitude 里面将软件包设置为“hold”状态，以避免这种情况。

apt 是最好的安装方法之一。您可以使用命令行模式的 apt-get ，还有 aptitude 或 synaptic 那样的工具 (只是 apt 的图形化前端)。注意，apt 还可以让您合并 main、contrib、non-free 和 non-free-firmware，从而您可以同时具有限制的软件包 (严格地讲不属于 Debian) 以及来自 Debian GNU/Linux 的软件包。

8.2.2 用于 Debian 的额外软件

有一些官方或非官方的软件库，在默认的 Debian 安装下并未打开。它们之中含有一些重要的必备软件。关于这些软件库的信息放在 Debian Wiki 页面 The Software Available for Debian’s Stable Release。

[1] 在 SysV init 系统下，halt 与 poweroff 具有相同的效果，但在 systemd 作为 init 系统时 (自 jessie 以来的默认设置)，它们的效果有所不同。

8.2.3 应用程序版本管理

可选版本的应用程序可以通过 update-alternatives 来管理。如果您维护着应用程序的多个版本，请阅读 update-alternatives 的 man 手册。

8.2.4 定时任务管理

任何系统管理员权限内的任务都必须位于 /etc 内，这是因为它们都是配置文件。如果您有一些需要以管理员身份每天 (daily)、每周 (weekly) 或每月 (monthly) 运行的定时任务，请将它们放置在 /etc/cron.{daily,weekly,monthly} 中。它们将在 /etc/crontab 中被激活，然后按照文件名的字典顺序依次运行。

另一方面，如果您有某个定时任务 (a) 需要以某个特定的用户身份运行，或者 (b) 需要在某个特定时刻或以特定周期运行，您可以使用 /etc/crontab，或者更好的 /etc/cron.d/whatever。这些特殊的文件可以包含一个额外的域，允许您指定运行该定时任务的用户帐户。

在任何情况下，您只需编辑这些文件，cron 将会自动处理它们。没有必要再运行某个特殊的命令。欲了解更多的信息，请参阅 cron(8)、crontab(5) 和 /usr/share/doc/cron/README.Debian。

8.3 更多信息

在 Debian 网站 上有大量的 Debian 相关文档。尤其要看看 Debian GNU/Linux FAQ 和 Debian Reference。更多的 Debian 文档位于 Debian Documentation Project。Debian 社区是靠用户自己的力量支撑起来的；如果您想要订阅 Debian 的邮件列表，请访问 Mail List Subscription 页面。最后，相当重要的，Debian Mailing List Archives 包含了 Debian 丰富的信息。

如果您想查阅某个程序的更详细资料，应该先试试 man 程序名称，或者 info 程序名称。

在 /usr/share/doc 目录里有很多有用的文档，特别是 /usr/share/doc/HOWTO 和 /usr/share/doc/FAQ 里有很多值得一读的文章。如果您想提交错误报告，请先查阅这些文件：/usr/share/doc/debian/bug*。针对某个程序 Debian 规格的内容可以查阅文件：/usr/share/doc/(package name)/README.Debian。

在 GNU/Linux 世界里，最常用的文档资料来源是：Linux Documentation Project。在那里您可以找到有关于 GNU/Linux 系统各个部分的非常有价值的 HOWTO 文档以及其他指导。

Linux 是一种 Unix 的实施。Linux Documentation Project (LDP) 收集了几种关于 Linux 的 HOWTOs 和在线书籍。

如果不熟悉 Unix，您也许应该出去买些书来读。Unix 常见问题解答里面包含了大量的 UseNet 文档，它提供了很好的历史参考。

8.4 让您的系统使用 E-Mail

今天，email 是许多人生活中一个重要的组成部分。建立它有很多种选择，并且对于很多 Debian 工具如何设置正确非常重要，我们将在本节介绍一些基本的知识。

e-mail 系统共有三个主要的组成功能。首先是 Mail User Agent (MUA)，这是用户发送和读取邮件的程序。然后是 Mail Transfer Agent (MTA)，用来将邮件从一台计算机传递到另一台。最后是 Mail Delivery Agent (MDA)，用于将收到的邮件投递到用户的收件箱。

这三项功能可以由不同的程序执行，但也能合并到一个或两个程序里。还可以用不同的程序处理不同类型的邮件。

在 Linux 和 Unix 系统上 mutt 是历史悠久的常用 MUA。像其他传统的 Linux 程序一样，是基于纯文本的。它常与作为 MTA 的 exim 或 sendmail、作为 MDA 的 procmail 一起使用。

配合逐步流行的图形桌面系统，使用图形 e-mail 程序，像 GNOME 的 evolution、KDE 的 kmail 或 Mozilla 的 thunderbird 已经变得更流行。这些程序综合了 MUA、MTA 和 MDA 的功能，但也能—而且通常是—与传统的 Linux 工具一起使用。

8.4.1 默认的 E-Mail 配置

即使您打算使用图形界面的邮件程序，安装传统的 MTA/MDA 并正确架设在您的 Debian GNU/Linux 系统上会是很有用的。其原因是很多运行在系统中的工具[2] 会通过邮件发送重要的提示通知系统管理员潜在的问题或变化。

[2] 例如：cron、quota、logcheck、aide，…

对此，您可以通过使用 apt install exim4 mutt ，来安装 exim4 和 mutt 。exim4 组合了 MTA/MDA 功能并相对小巧和灵活。它默认配置为只处理系统本地的 e-mail，系统管理员 (root 帐号) 的 e-mail 地址会发送到安装时创建的普通用户帐号[3]。

当系统邮件传递后，它们会添加到一个 /var/mail/account_name 文件。这些邮件可以通过 mutt 读取。

8.4.2 发送邮件到系统之外

前面提到，安装的 Debian 系统只架设了处理本地系统的邮件，而不是用于发送或接收其他系统的邮件。

如果您希望使用 exim4 来处理外部邮件，请参考下一小节中可以使用的基本配置选项。确认邮件可以正确地发送和接收。

如果您打算使用图形界面的邮件程序，并使用您的 Internet Service Provider (ISP) 或公司提供的邮件服务器，就不需要配置 exim4 来处理外部 e-mail。只须配置您喜爱的图形界面邮件程序使用正确的服务器来发送和接收 e-mail (配置方法超出了本手册的范围)。

但是，这种情况下您可能还是要配置一些独立的工具来正确发送 e-mails。其中一个这样的工具是 reportbug，该程序的功能是提交 Debian 软件包对应的 bug 报告。它默认是使用 exim4 来提交 bug 报告。

要正确设置 reportbug 使用外部的邮件服务器，请执行 reportbug --configure 命令，并对是否 MTA 可用问题回答“no”。您接着将会被问到使用哪个 SMTP 服务器来提交 bug 报告。

8.4.3 配置 Exim4 MTA

假如您还打算让系统处理外部 e-mail，需要重新配置 exim4 软件包[4]：

```
# dpkg-reconfigure exim4-config
```

输入命令之后 (作为 root)，您会被问到是否需要将配置文件分成几个小文件。如果您拿不准，就选择默认选项。

接着您将看到几个常见的邮件方案，请选择一个最近似您需求的那个。

internet site 您的系统被连接到网络上，并且您通过 SMTP 直接收发邮件。在接下来的几页中，程序会询问您一些基本问题，如：您的机器的邮件名称、您接受或转发邮件的域等等。

mail sent by smarthost 本方案中您的送出邮件转发到另一台机器，称为“smarthost”，它来负责发送信息到最终目的地。smarthost 一般还用于保存您的计算机接收的邮件，所以您不需要长时间在线。这也意味着您需要使用类似 fetchmail 这样的程序从 smarthost 下载邮件。

大多时候 smarthost 是您 ISP 的邮件服务器，这对拨号用户非常适合。它也可以是公司的邮件服务器，或是您自己网络中的另外一台机器。

mail sent by smarthost; no local mail 该选项基本上与前一种情况相同，只有一点不同，本系统不再架设用于处理本地的 e-mail domain。在本系统上的邮件 (比如，给系统管理员的) 还是会被处理。

local delivery only 本选项是系统默认的配置。

no configuration at this time 除非您真的知道这是在干什么，否则请不要选择这一选项。这会留下一个未配置的邮件系统—在您再次配置它之前，您都无法收发任何邮件，并且可能会错过一些系统工具发来的重要信息。

如果没有合适的方案，或者需要更精确的设置，您需要在安装完成之后编辑 /etc/exim4 目录下的配置文件。有关 exim4 更多的信息可以在 /usr/share/doc/exim4 下找到；README.Debian.gz 里面有 exim4 配置方面的细节，并说明从哪里找到更多的文档。

注意，如果您没有正式的域名，直接发送邮件到互联网，因为接收服务器的反垃圾邮件策略会拒绝接收邮件。这时建议使用 ISP 的邮件服务器。假如您还想直接发送邮件，可能要用另一个邮件地址替换默认生成的那个。如果您使用的是 exim4 作为 MTA，可以添加一个条目到 /etc/email-addresses。

[3]转发 root 邮件到普通用户帐号是在 /etc/aliases 里配置。如果没有创建普通用户帐号，邮件自然会发送给 root 帐号。

[4]您也可以卸载 exim4 安装其他的 MTA/MDA。

8.5 编译新内核

为什么有人想要编译新的内核？非常可能是不必要的，因为 Debian 带的默认内核处理几乎所有配置。

如果您无论如何想要编译自己的内核，这当然是可能的，并且我们推荐使用“make deb-pkg”目标。更多信息请阅读 Debian Linux Kernel Handbook 。

8.6 修复损坏的系统

有时候，会遇到一些麻烦，您小心翼翼安装的系统无法启动。可能是由于调整启动加载程序时损坏了配置文件，或者新准备的内核无法启动，或者宇宙射线撞击了您的硬盘造成 `/sbin/init` 中的一位反转。无论什么原因，修复时您都需要一个可以工作的系统，rescue 模式正好用于此目的。

要进入救援模式，请在启动菜单中选择 `rescue`，或在 `boot:` 提示符后键入 `rescue`，或者启动时给给引导参数加上 `rescue/enable=true`。您会看到安装程序的前几个画面，并在显示器的角落里面提示处于救援模式，而不是完整安装。别担心，您的系统不会被覆盖！rescue 模式只是借助安装程序强大的硬件探测能力，确保修复系统的时候硬盘，网络设备等等可以使用。

取代分区工具的，是显示您系统上的分区列表，并请您选择。通常，您应该选择修复包含 root 文件系统的分区。您也可以选择 RAID 和 LVM 设备上的分区，与直接在磁盘上创建一样。

如可能，安装程序将在您选择的文件系统上给出一个 shell 提示符，通过它可以执行一些必要的修复操作。

假如安装程序在您选择的 root 文件系统上不能运行一个可用的 shell，这可能是因为文件系统已经损坏，那么将发出警告，并通过安装程序的环境提供一个 shell。您在这个环境下可能没有丰富的工具可用，但足以应付系统修复工作。您选择的 root 文件系统将挂载到 `/target` 目录。

无论哪种情况，退出 shell 之后，系统将重启。

最后，要注意修复一个损坏的系统可能非常困难，在这份手册里面不会列举所有的情况和修复方法。如果您遇到困难，请咨询专家。

Appendix A

安装指南

本文介绍了如何在 S/390 (“s390x”) 架构的计算机上，使用新版的 debian-installer 安装 Debian GNU/Linux trixie。这是一次对安装过程走马观花式的简介，它应该囊括了大多数情况下您所需的信息。当需要更多信息时，我们会链接到更详尽的本文其他部分的说明文档。

A.1 前言

如果您在安装时碰到程序出错，请参考第 5.3.4 节回报错误。倘若本文档不能够回答您所有的疑问，那么请带着这些问题去 debian-boot 的邮件列表 (debian-boot@lists.debian.org) 问问看，或者您也可以在 irc (OFTC 网络上的 #debian-boot 频道) 上直接提问。

A.2 安装套件的引导

Debian-cd 小组在 Debian CD/DVD 页上提供了使用 debian-installer 安装映像的已编译版本。如果您需要了解有关如何得到这些安装映像的更多信息，请看第 4.1 节。

有些安装方法需要其它种类的映像而非那些光介质。第 4.2.1 节会告诉您如何在 Debian 的映像站点找到那些映像文件。

下面小节会就每种可能的安装方法，具体讲授如何获得所需的映像文件。

A.2.1 光盘

netinst CD 映像是常用的映像可以用来通过 debian-installer 安装 trixie。使用该安装方法让计算机从映像启动，再由安装程序从网络上下载安装其它的软件包，因此称做“netinst”。该映像含有运行安装程序的软件模块，以及提供一个最小 trixie 系统的基本软件包。如果您乐意，可以下载一个 DVD 映像文件，用它的话就可以不借助网络安装了。只需要这套中的第一个映像就行了。

下载您打算使用的映像文件，把它刻成光盘。

A.2.2 从网络启动

要完全从网络上启动 (即 netboot) debian-installer 也是可行的。而具体采取哪种方法来从网络启动，这取决于您的硬件架构和网络启动的相关设置。在 netboot/ 目录里的文件是用来从网络启动 debian-installer 的。

A.2.3 从硬盘启动

除了从移动介质启动安装套件，其实从手中现有的安装了其它操作系统的硬盘启动也是可行的。把 hd-media/initrd.gz、hd-media/vmlinuz 和一个 Debian DVD 映像文件下载到硬盘最顶层的目录里。请确保这个映像文件的文件名是以 .iso 结尾的。接下来的问题就是用 initrd 启动 linux 了。

A.3 安装

一旦安装程序运行，会有一个初始画面迎接您。按 Enter 启动系统；或者先阅读一下其它的启动方式以及各种参数的说明 (参阅第 5.2 节)。

再过一会儿，您就会可以选择您所使用的语言了。用上下箭头键来选择一门语言，然后按 Enter 继续下面的步骤。接下来，需要选择您的国家，选项里包括了那些使用您所选语言的国家。如果在那个稍短的列表里找不到您要找的国家，还有一个含有世界上所有国家的列表以备选择。

您还可能被要求确认一下您的键盘布局。请选择缺省的那个，除非您清楚更合适的布局。

现在可以坐下来，松口气，休息一下了。这时 debian-installer 会侦测您的一部分硬件，并从安装映像中把它的其余部分加载进系统。

接下来，安装程序会想办法侦测您的网络相关的硬件，再用 DHCP 完成网络的设置。如果没有连接网络，或是没有用 DHCP，那么您也可以手动配置网络。

设置网络，然后创建用户帐户。默认情况下，您将会要求为“root”(管理员) 帐户提供密码，并创建一个常规用户帐户。如果您没有为“root”用户指定密码，则此帐户将被禁用，但稍后将安装 sudo 软件包，以便在新系统上执行管理任务。默认系统创建的第一个用户被允许使用 sudo 命令而称为 root。

下一步是建立您的时区和时间。安装程序将试着连接互联网上的时间服务器来校准时间。时区是基于前面所选的国家，假如一个国家有多个时区，安装程序会要您选择其中的一个。

现在该对硬盘分区了。首先，您可以选择对整个驱动器，或是对已有的空闲空间进行自动分区 (参阅第 6.3.4.2 节)。对于新手或者急于使用的朋友推荐使用这种方式。如果不想用自动分区，请在菜单中选择 Manual。

在接下来的屏幕上，您可以看到自己的分区表，各分区将以何种方式格式化，以及它们将来的挂载点在何处。选择一个分区进行修改或删除。如果您先前选择的是自动分区，那么现在只需从菜单选择 Finish partitioning and write changes to disk，使用它创建的即可。请不要忘记至少需要分配一个分区作为交换空间，以及把一个分区挂载到 / 上。有关使用分区工具详尽的资料，请参考第 6.3.4 节；附录附录 C 有更多关于分区方面的信息。

现在 debian-installer 会把您的分区格式化，然后安装基系统。这两步会花些时间。接着，它会安装内核。

先前安装的基本系统是一个可以工作，但却是非常小型的安装。要让系统具有更强的功能，下一步让您通过选择任务安装额外的软件包。软件包安装之前，需要先配置 apt，指定软件包从哪里获取。默认会选定安装“标准系统”任务。如果您打算安装完成后使用图形桌面，请选择“桌面环境”任务。参阅第 6.3.6.2 节了解本步骤的详细信息。

最后一步就是安装启动引导器了。如果安装程序发现您的计算机还安装有其它的操作系统，它会把它们加入启动菜单，并通知您。

debian-installer 现在会告诉您安装已经完成了。取出光盘或者其它启动介质，然后键入 Enter 就可以重新启动您的机器了。它将启动新安装的系统，接着让您登录。在第 7 章有对它的详尽说明。

如果您想要更多的安装过程相关信息，请看第 6 章。

A.4 请给我们发一份安装报告

如果您用 debian-installer 安装成功，请抽空给我们写一份报告。最简单的方法，是安装 reportbug 软件包 (apt install reportbug)，配置 reportbug 的说明见第 8.4.2 节，然后执行 reportbug installation-reports。

如果不能完成安装，那么您也许发现了一个 debian-installler 的错误。要改进安装程序，就很有必要让我们知道这些异状，因此，请抽空把这些错误报告给我们。您可以使用 installation report 来回报错误；如果安装彻底失败了，请参阅第 5.3.3 节。

A.5 末了…

我们祝愿您安装 Debian 顺利愉快，并希望您能发现 Debian 的用武之地。还有您也许可以读一下第 8 章。

Appendix B

使用预置自动进行安装

本附录说明如何预置 debian-installer 中提问的答案，让安装过程自动化。

本附录使用的配置片段还放在预置例子文件 https://www.debian.org/releases/trixie/example-preseed.txt 里面。

B.1 介绍

预置提供了一种在安装过程中回答问题，而无须手动键入答案的途径。这让完全自动进行大多数的安装成为可能，并提供了一些普通安装方式下不具备的特性。

预置文件并非必须。如果您使用空的预置文件，安装程序就会跟正常手动安装一样。每个问题的预置 (如果使用正确!) 都会以同样的方式修改对应的基准。

B.1.1 预置方法

一共有三种方法可用于预置: initrd、file 和 network。initrd 方法可以用于任何一种安装方式并支持较多的预置内容，但它需要做的准备工作也最多。file 和 network 方法分别用于不同的安装方式。

下面的表格显示了各种安装方式可选择的预置方法。

安装方式	initrd	file	network
CD/DVD/USB	是	是	是[1]
netboot	是	否	是
硬盘	是	是	yes[1]
generic	是	否	是

这些预置方法的主要不同之处在于何时预置文件被加载和处理。对于 initrd 方法，它是在第一个问题询问之前，安装开始的时候。来自内核命令行的前置尽在其后发生。这样能够通过编辑内核命令行，来否决 initrd 中设置的配置 (或者在 bootloader 配置，或者对于允许的 bootloader 在引导时手动)。对于 file 方法，它是在加载安装映像之后。对 network 方法，它在网络配置完毕之后。

重要

显而易见，在加载预置文件之前需要的过程都不能被预置 (这里面包括只在 medium 或 low 优先级出现的问题，像首先运行的硬件检测)。避免询问这些问题的不那么方便的方法，是通过引导参数来预置它们，如在第 B.2.2 节中描述的。

为了容易地避免在预置之前询问，您可以让安装程序运行在“auto”模式。它延迟了预置之前出现询问问题 (例如，语言、国家和键盘选择)，直至连上网络，使它们可以进行预置。它还让安装运行在 critical 优先级下，避免不重要的问题。参阅第 B.2.3 节了解详细信息。

[1] 除非您有网络并正确设置 preseed/url

B.1.2 限制

虽然使用这种方法能预先设置 debian-installer 的绝大多数问答，但还是有明显的例外。您必须 (重新) 分区整个硬盘，或者使用磁盘上的剩余空间; 但无法使用已经存在的分区。

B.2 使用预置

您首先要创建一个预置文件，并将它置于希望使用的地方。创建预置文件在后面的附录中有说明。如果您打算使用网络预置或者从 U 盘读取，直接将它放到正确的地方就可以了。如果您希望将它放到安装 ISO 映像上面，那就需要重新制作一个映像文件。而如何将预置文件包含到 initrd 里面，超出了本文档的范围；请查询 debian-installer 开发者文档。

有一个预置文件的例子放在 https://www.debian.org/releases/trixie/example-preseed.txt，您可以拿它作为自己预置文件的基础。此文件基于本附录所列的配置文件片段。

B.2.1 加载预置文件

如果使用 initrd 预置，您只需确保一个名为 preseed.cfg 的文件包含在 initrd 的根目录里面。安装程序会自动检测并加载。

其他的预置方法，您需要在启动的时候告诉安装程序所使用的文件。这通过给内核传递引导参数完成，可以在启动时手动指定，或者编辑 bootloader 的配置文件 (如: syslinux.cfg)，并在内核的附加行后面添加参数。

如果在 bootloader 配置里指定预置文件，可以修改配置而不再需要键入回车来引导安装程序。对于 syslinux，这意味着在 syslinux.cfg 中将超时 timeout 设置为 1。

为了保证安装程序得到正确的预置文件，您可以选择为预置文件指定校验和。当前使用的是 md5sum，如果指定了，那么配置文件必须匹配，否则安装程序将拒绝使用。

```
b'' 要 b''b'' 指 b''b'' 定 b''b'' 的 b''b'' 引 b''b'' 导 b''b'' 参 b''b'' 数 b'':
- b'' 如 b''b'' 果 b''b'' 使 b''b'' 用 b''b'' 网 b''b'' 络 b''b'' 启 b''b'' 动 b'':
  preseed/url=http://host/path/to/preseed.cfg
  preseed/url/checksum=5da499872becccfeda2c4872f9171c3d
-b'' 或 b''b'' 者 b''
  preseed/url=tftp://host/path/to/preseed.cfg
  preseed/url/checksum=5da499872becccfeda2c4872f9171c3d

- b'' 如 b''b'' 果 b''b'' 从 b''b'' 定 b''b'' 制 b''b'' 的 b''b'' 安 b''b'' 装 b''b'' 映
    b''b'' 像 b''b'' 启 b''b'' 动 b'':
  preseed/file=/cdrom/preseed.cfg
  preseed/file/checksum=5da499872becccfeda2c4872f9171c3d

- b'' 如 b''b'' 果 b''b'' 用 b'' USB b'' 设 b''b'' 备 b''b'' 安 b''b'' 装 b''(b'' 将
    b''b'' 预 b''b'' 置 b''b'' 文 b''b'' 件 b''b'' 放 b''b'' 入 b'' U b'' 盘 b''b'' 的
    b''b'' 顶 b''b'' 极 b''b'' 目 b''b'' 录 b''):
  preseed/file=/hd-media/preseed.cfg
  preseed/file/checksum=5da499872becccfeda2c4872f9171c3d
```

注意，作为引导参数时 preseed/url 可以缩写为 url、preseed/file 缩写为 file、preseed/file/checksum 缩写为 preseed-md5。

B.2.2 使用引导参数预置问题的答案

如果预置文件无法设置一些步骤，安装也可以完全自动进行，这是因为您可以在引导安装程序时，在命令行上为内核传递预置的值。

引导参数可以在您不想使用预置的时候使用，但只提供一个特定问题的答案。在本手册中有一些这样的示例。

要设置 debian-installer 中使用的值，只须传递 path/to/variable=value，它们是本附录中所列的任何一个。如果该值是用于配置目标系统的软件包，您需要预先设置 owner:path/to/variable=value 中的 owner[2] 变量。如果您不指定 owner，该变量的值不会复制到目标系统上的 debconf 数据库，并在相关的软件包的配置中不使用。

[2] debconf 变量 (或模板) 的 owner 通常是包含对应 debconf 模板的软件包明。对于安装程序自己使用的变量，owner 是“d-i”。模板和变量可以有多个 owner，以助于删除软件包时判断它们是否要从 debconf 数据库中删除。

通常，采用这种方式预置是为了问题不再被询问。为了设置一个问题的默认答案，并仍然显示问题，请使用“?=”替换“=”作为操作符。参阅第 B.5.2 节。

注意，一些常用的引导参数变量有简写别名。如果有简写形式，它们将会被用在附录的例子里面替代全称。例如，preseed/url 变量简写为 url。另一个例子是 tasks，它转换成 tasksel:tasksel/first。

“---”在启动选项中有特殊含义。最后一个“---”后面出现的内核参数将被复制到安装的 bootloader 设置 (假如安装程序支持 bootloader)。安装程序可以自动发现它能识别的选项 (就像预置选项)。

注意

当前的 linux 内核 (2.6.9 和更新的)，可以指定最多 32 个命令行选项和 32 个环境变量选项，包括安装程序默认的选项。如果数目超出，内核将崩溃。(早期的内核，接受的数目更少。)

大多数安装过程，在引导配置文件中一些默认的选项，比如 vga=normal，可以放心地删除，这让您可以添加更多的预置选项。

注意

引导参数中即使使用引号，也可能无法夹杂空格。

B.2.3 auto 模式

有几个 Debian 安装程序的特性结合到一起，在引导提示时允许相当简单的命令行，导致了任何复杂的定制化自动安装。

这通过使用 Automated install 引导选择来启动，对一些架构或引导方法也被称为 auto。在这节中，这样 auto 不是参数，它意味着选择引导选项，并在引导提示上增补后面的引导参数。

为了表达清楚，这里有一些例子，可以用在引导提示符后面：

```
auto url=autoserver
```

这需要一个 DHCP 服务器能找到并且 DNS 可以解析 autoserver，或许要加上 DHCP 提供的本地域名。如果域名是 example.com，并有相应的 DHCP 设置，那么预置文件将从 http://autoserver.example.com/d-i/tr 获得。

url 的最后一部分 (d-i/trixie/./preseed.cfg) 来自 auto-install/defaultroot。默认包括 trixie 目录，让将来的版本在可控的方式下指定自己的代号。/./ 位用来定位相对于根目录的子目录 (用于 preseed/include 和 preseed/run)。使文件可以指定为完整路径、以 / 开始的位置或者相对于最后一个预置的文件。这样可以建立更具移植性的脚本，让各级脚本完整地移动到新的位置而不被破坏。比如，复制 web 服务器上的文件到 U 盘。本例中，如果预置文件的 preseed/run 为 /scripts/late_command.sh，那么文件将从 http://autoserver.example.com/d-i/trixie/./scripts 获得。

如果本地没有 DHCP 或 DNS，或者您不想使用 preseed.cfg 的默认路径，可以直接指定路径。并且，如果不使用 /./ 元素，它将定位到路径的起始点 (例如 URL 中的第三个 /)。下面的例子是局域网中最基本的要求：

```
auto url=http://192.168.1.2/path/to/mypreseed.file
```

它会以下列方式运作：

- 如果 URL 不含协议，就使用 http
- 如果主机名不含域名，将使用 DHCP 提供的，并且
- 如果主机名后面没有 /，那么将加上默认路径。

除了指定 url 之外，您还可以指定不直接影响 debian-installer 自己行为的设置，但是可以传递到预置文件加载的 preseed/run 中的脚本。当前，只有一个 auto-install/classes，它的别名是 classes。可以这样用：

```
auto url=example.com classes=class_A;class_B
```

classes 可以指定系统的类型，或者本地化信息。

只要您愿意，当然可以扩充这个概念，合理应用 auto-install 名字空间就是为了这个目的。这就可能会有诸如 `auto-install/style` 的内容加入到您的脚本。如果您觉得需要这样做，请发邮件到 debian-boot@lists.debian.org 邮件列表说明，我们可以避免名字空间的冲突，还可能会为您增加一个参数的别名。

`auto` 引导先择并未在所有架构上定义。在内核命令行上可以通过添加 `auto=true priority=critical` 两个参数达到同样的效果。`auto` 内核参数是 `auto-install/enable` 的别名，将其设为 `true` 可以延迟 locale 和键盘等问题的提出，使其可以提供预置，其中 `priority` 是 `debconf/priority` 的别名，设置它为 `critical` 可以禁止在较低优先级下的问题提出。

When an Automatic install is selected without also specifying a preseed configuration, the user will be prompted to provide a preseed location. Alternatively, if a preseed configuration has been specified, then that prompt is normally suppressed in order to enable a fully non-interactive install. However, if the intent is to modify defaults by using a `preseed.cfg` in an initrd, then the result should really behave as though no preseeding has yet been specified, even though it has. In that case, the debconf variable `auto-install/cloak_initrd_preseed` can be set to `true` (e.g. in the initrd's `preseed.cfg`). to get the required behaviour.

在使用 DHCP 自动安装时还有一个有趣的额外选项：`interface=auto netcfg/dhcp_timeout=60`，这让机器使用第一个可用的网卡并耐心地等待 DHCP 的查询。

提示

更多使用此框架的例子，包括示例脚本和类，可以在其开发者的网站找到。那里的例子还显示出预置可以达到更多的效果。

B.2.4 预置中有用的别名

下面这些别名在使用 (自动) 预置时很有用处。注意，它们是提问名称的简写别名，您同时需要指定一个对应的值：例如，`auto=true` 或 `interface=eth0`。

priority	debconf/priority
fb	debian-installer/framebuffer
auto	auto-install/enable
classes	auto-install/classes
file	preseed/file
url	preseed/url
theme	debian-installer/theme
language	debian-installer/language
country	debian-installer/country
locale	debian-installer/locale
keymap	keyboard-configuration/xkb-keymap
modules	anna/choose_modules
firmware	hw-detect/firmware-lookup
interface	netcfg/choose_interface
domain	netcfg/get_domain
hostname	netcfg/get_hostname
protocol	mirror/protocol
suite	mirror/suite
recommends	base-installer/install-recommends
tasks	tasksel:tasksel/first
desktop	tasksel:tasksel/desktop
preseed-md5	preseed/file/checksum

B.2.5 引导提示预置的例子

这里是一些引导提示看起来什么样子的例子（还需要将这个与您的需要相适应）。

```
# To set French as language and France as country:
/install.amd/vmlinuz vga=788 initrd=/install.amd/gtk/initrd.gz language=fr ↩
    country=FR --- quiet
# To set English as language and Germany as country, and use a German keyboard ↩
    layout:
/install.amd/vmlinuz vga=788 initrd=/install.amd/gtk/initrd.gz language=en ↩
    country=DE locale=en_US.UTF-8 keymap=de --- quiet
# To install the MATE desktop:
/install.amd/vmlinuz vga=788 initrd=/install.amd/gtk/initrd.gz desktop=mate- ↩
    desktop --- quiet
# To install the web-server task:
/install.amd/vmlinuz initrd=/install.amd/initrd.gz tasksel:tasksel/first=web- ↩
    server ---
```

B.2.6 使用 DHCP 服务器指定预置文件

还可以使用 DHCP 来指定从网络下载的预置文件。DHCP 允许指定文件名。通常是用于 netboot 的文件，但如果是一个 URL，那么支持网络预置的安装介质就会从该 URL 下载，并用作预置文件。这里有一个在 dhcpd.conf 中使用的例子，它用于 ISC DHCP 服务器第 3 版 (Debian 里面是 isc-dhcp-server 软件包)。

```
if substring (option vendor-class-identifier, 0, 3) = "d-i" {
    filename "http://host/preseed.cfg";
}
```

注意，上面的例子中使用的文件名限定 DHCP 客户端自己是“d-i”，因此仅适用于安装程序，而对普通的 DHCP 客户端无效。您可以将只用于一个特殊主机的文字放在一段，以避免预置网络中的其他安装。

使用 DHCP 预置的一个好方法是仅预置网络相关的值，比如所使用的 Debian 镜像。这种方式下在网络中安装将会自动选择一个合适的镜像，而其余的安装部分仍可以交互进行。使用 DHCP 预置进行完整的 Debian 自动安装应该特别小心。

B.3 创建预置文件

预置文件是 debconf-set-selections 命令使用的格式。通常预置文件中每一行的格式为：

```
<owner> <question name> <question type> <value>
```

文件应该以 #_preseed_V1 开始
编写预置文件的时候要留意几条规则。

- 在类型和值之间只有一个空格或者 tab：任何额外的空格将被解释为值的一部分。
- 一行可以通过行尾的反斜杠（“\”）连接符分成多行。分割一行比较妥当的地方是在问题后面；在类型和值之间分割则是糟糕做法。被分割的行连接时会把行首和行尾的空格缩减为一个空格，再组成单独一行。
- 只在安装程序自己使用的 debconf 变量 (模板)，owner 应该设置为“d-i”；要在安装后的系统里面预置变量，应该使用名称中包含对应 debconf 模板的软件包。只有那些 owner 为非“d-i”的变量将被保存到安装后系统的 debconf 数据库里面。
- 大多数问题需要使用有效的英文预置值，而不是翻译过的值。但有些问题 (例如 partman)，需要使用翻译过的值。
- 有些问题在安装过程中显示值的代码而不是英文文本。
- 以 #_preseed_V1 开始
- 注释行必须以井号（“#”）打头，一直延续到行尾。

建立预置文件最简单的方法是使用例子文件作为工作的基础，它位于第 B.4 节。

另一种方法是手动安装，重启后使用 debconf-utils 软件包里面的 debconf-get-selections，将 debconf 数据库和安装程序的 cdebconf 数据库导出到单一一个文件：

```
$ echo "#_preseed_V1" > file
$ debconf-get-selections --installer >> file
$ debconf-get-selections >> file
```

但用此方法产生的文件，有些项无法预置。对于大多数用户，示例文件是一个较佳的开始。

注意

这种方法在安装的最后，安装程序的 cdebconf 数据库被保存到新安装系统的 /var/log/installer/cdebconf 文件。由于数据库里面常还有敏感的信息，默认只为 root 可读。

如果您清除 installation-report 软件包，那么 /var/log/installer 目录及其中的所有文件将从您的系统中删除。

为了检测问题答案可用的值，您可以在安装进行中使用 nano 检查 /var/lib/cdebconf 下的文件。templates.dat 是原始的模板，questions.dat 是当前变量设置的值。

为了在安装之前测试您的预置文件是否有效，您可以使用 debconf-set-selections -c preseed.cfg 命令。

B.4 预置文件的内容 (trixie)

本附录使用的配置片段还放在预置例子文件 https://www.debian.org/releases/trixie/example-preseed.txt 里面。

请注意，这里的例子是基于 Intel x86 架构的安装过程。如果您安装到其他架构，其中的一些例子 (像键盘选择和 bootloader 安装) 可能不恰当，需要用适合您架构的 debconf 设置替换。

有关不同 Debian 安装程序模块的工作细节，请参阅第 6.3 节。

B.4.1 本地化

在一般的安装中，首先询问地区信息，所以这些值只通过 initrd 或内核引导参数方法来预置。自动模式 (第 B.2.3 节) 包括 auto-install/enable=true 的设置 (通常通过 auto 预置别名)。这延迟询问地区问题，使它们可以被任何方法预置。

locale 用于指定语言和国家或者任何 debian-installer 支持的语言与地区的组合。如果组合起来的不是有效的 locale，安装程序会自动选择一个指定语言可用的 locale。要在引导参数上指定本地，比如使用 locale=en_US.UTF-8。

虽然这个方法很容易使用，但它无法预置所有的语言、国家和地方组合[3]。因此另一种方法是单独预置各值。语言和国家可以在引导参数里面指定。

```
# Preseeding only locale sets language, country and locale.
d-i debian-installer/locale string en_US.UTF-8

# The values can also be preseeded individually for greater flexibility.
#d-i debian-installer/language string en
#d-i debian-installer/country string NL
#d-i debian-installer/locale string en_GB.UTF-8
# Optionally specify additional locales to be generated.
#d-i localechooser/supported-locales multiselect en_US.UTF-8, nl_NL.UTF-8
```

键盘配置包括选择的 keymap 和 (对于非拉丁 keymap) 切换键，该键用于非拉丁和 US keymap 来回切换。安装的时候只有几种基本的 keymap 可用。高级的要在安装好的系统下才能使用，使用 dpkg-reconfigure keyboard-configuration 配置。

[3]比如预置 locale 为 en_NL 在安装后的系统中默认的 locale 将是 en_US.UTF-8。如果想使用 en_GB.UTF-8，该值应该分别进行预置。

```
# Keyboard selection.
d-i keyboard-configuration/xkb-keymap select us
# d-i keyboard-configuration/toggle select No toggling
```

要跳过键盘配置，可通过 `skip-config` 来预置 keymap。这将导致内核 keymap 保持活动。

B.4.2 语音合成

When installing with software speech synthesis (i.e. `speakup.synth=soft` passed on the kernel command line), the sound board and the voice to be used can be preseeded.

Since speech synthesis is started very early, this can only be preseeded from initrd or the kernel command line.

Preseeding `espeakup/voice` also enables configuring the installed system with speech accessibility features. This takes effect even if software speech synthesis was not actually enabled inside the installer.

```
# Which ALSA card to be used for software speech.
# Can be a number from 0, or an ID as seen in
# /sys/class/sound/card*/id
#d-i espeakup/card string 0

# Which espeak-ng voice to be used
#d-i espeakup/voice string en
```

B.4.3 网络设置

显而易见，预置网络设置对于从网络加载预置文件无效。但对从光盘和 U 盘引导很有帮助。如果您计划从网络加载预置文件，应该使用传递网络设置给内核引导参数的方式。

如果您需要网络引导从网络加载预置文件前指定网卡，请用这种引导参数 `interface=eth1`。

虽然使用网络预置 (用“preseed/url”) 通常并不能配置网络，但是您可以使用下面的技巧实现。例如，为网卡设置静态地址。它使加载了预置文件以后网络预置再运行一次，这需要将下面的命令包含在“preseed/run”脚本里面：

```
kill-all-dhcp; netcfg
```

下面的 debconf 变量与网络配置相关。

```
# Disable network configuration entirely. This is useful for cdrom
# installations on non-networked devices where the network questions,
# warning and long timeouts are a nuisance.
#d-i netcfg/enable boolean false

# netcfg will choose an interface that has link if possible. This makes it
# skip displaying a list if there is more than one interface.
d-i netcfg/choose_interface select auto

# To pick a particular interface instead:
#d-i netcfg/choose_interface select eth1

# To set a different link detection timeout (default is 3 seconds).
# Values are interpreted as seconds.
#d-i netcfg/link_wait_timeout string 10

# If you have a slow dhcp server and the installer times out waiting for
# it, this might be useful.
#d-i netcfg/dhcp_timeout string 60
#d-i netcfg/dhcpv6_timeout string 60

# Automatic network configuration is the default.
# If you prefer to configure the network manually, uncomment this line and
# the static network configuration below.
#d-i netcfg/disable_autoconfig boolean true
```

```
# If you want the preconfiguration file to work on systems both with and
# without a dhcp server, uncomment these lines and the static network
# configuration below.
#d-i netcfg/dhcp_failed note
#d-i netcfg/dhcp_options select Configure network manually

# Static network configuration.
#
# IPv4 example
#d-i netcfg/get_ipaddress string 192.168.1.42
#d-i netcfg/get_netmask string 255.255.255.0
#d-i netcfg/get_gateway string 192.168.1.1
#d-i netcfg/get_nameservers string 192.168.1.1
#d-i netcfg/confirm_static boolean true
#
# IPv6 example
#d-i netcfg/get_ipaddress string fc00::2
#d-i netcfg/get_netmask string ffff:ffff:ffff:ffff::
#d-i netcfg/get_gateway string fc00::1
#d-i netcfg/get_nameservers string fc00::1
#d-i netcfg/confirm_static boolean true

# Any hostname and domain names assigned from dhcp take precedence over
# values set here. However, setting the values still prevents the questions
# from being shown, even if values come from dhcp.
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain

# If you want to force a hostname, regardless of what either the DHCP
# server returns or what the reverse DNS entry for the IP is, uncomment
# and adjust the following line.
#d-i netcfg/hostname string somehost

# Disable that annoying WEP key dialog.
d-i netcfg/wireless_wep string
# The wacky dhcp hostname that some ISPs use as a password of sorts.
#d-i netcfg/dhcp_hostname string radish

# If you want to completely disable firmware lookup (i.e. not use firmware
# files or packages that might be available on installation images):
#d-i hw-detect/firmware-lookup string never

# If non-free firmware is needed for the network or other hardware, you can
# configure the installer to always try to load it, without prompting. Or
# change to false to disable asking.
#d-i hw-detect/load_firmware boolean true
```

请注意，如果 netcfg/get_netmask 没有预置，那么 netcfg 将自动侦测掩码。在这种情况下，变量必须标记为 seen 来自动安装。类似地，如果没有设置 netcfg/get_gateway，那么 netcfg 将选择一个合适的地址。特殊情况下，可以将 netcfg/get_gateway 设置为“none”，来指定不使用网关。

B.4.4 网络控制台

```
# Use the following settings if you wish to make use of the network-console
# component for remote installation over SSH. This only makes sense if you
# intend to perform the remainder of the installation manually.
#d-i anna/choose_modules string network-console
#d-i network-console/authorized_keys_url string http://10.0.0.1/openssh-key
#d-i network-console/password password r00tme
#d-i network-console/password-again password r00tme
```

关于网络控制台的更多信息，请参阅第 6.3.10 节。

B.4.5 镜像设置

根据您使用的安装方式，镜像可用于下载安装程序的额外组件、安装基本系统以及为所安装的系统建立 `/etc/apt/sources.list`。

参数 `mirror/suite` 决定了安装好的系统使用的套件。

参数 `mirror/udeb/suite` 决定安装程序使用的额外组件的套件。它只在组件通过网络下载并与安装时使用的 initrd 建立套件相匹配时才有效。通常安装程序会自动安装并使用正确的值而毋需手动设置。

```
# Mirror protocol:
# If you select ftp, the mirror/country string does not need to be set.
# Default value for the mirror protocol: http.
#d-i mirror/protocol string ftp
d-i mirror/country string manual
d-i mirror/http/hostname string http.us.debian.org
d-i mirror/http/directory string /debian
d-i mirror/http/proxy string

# Suite to install.
#d-i mirror/suite string testing
# Suite to use for loading installer components (optional).
#d-i mirror/udeb/suite string testing
```

B.4.6 帐号设置

root 帐号的密码和第一个普通用户的名字和密码都可以预置。您可以给密码使用纯文本或 crypt(3) 哈希值。

警告

要知道预置密码并不安全，每个可以访问预置文件的用户都可以看到这些密码。存储哈希密码被认为是安全的，除非使用了可以被暴力破解的弱哈希算法，如 DES 或 MD5。推荐的密码哈希算法为 SHA-256 和 SHA512。

```
# Skip creation of a root account (normal user account will be able to
# use sudo).
#d-i passwd/root-login boolean false
# Alternatively, to skip creation of a normal user account.
#d-i passwd/make-user boolean false

# Root password, either in clear text
#d-i passwd/root-password password r00tme
#d-i passwd/root-password-again password r00tme
# or encrypted using a crypt(3)  hash.
#d-i passwd/root-password-crypted password [crypt(3) hash]

# To create a normal user account.
#d-i passwd/user-fullname string Debian User
#d-i passwd/username string debian
# Normal user's password, either in clear text
#d-i passwd/user-password password insecure
#d-i passwd/user-password-again password insecure
# or encrypted using a crypt(3) hash.
#d-i passwd/user-password-crypted password [crypt(3) hash]
# Create the first user with the specified UID instead of the default.
#d-i passwd/user-uid string 1010

# The user account will be added to some standard initial groups. To
# override that, use this.
#d-i passwd/user-default-groups string audio cdrom video
```

passwd/root-password-crypted 和 passwd/user-password-crypted 变量也可以使用“!”作为它们的预置值。在这种情况下，相应的帐号关闭。这对于 root 帐号会比较方便，提供了替代方法来设置允许管理活动或 root 登录（例如通过使用 SSH 密钥认证或 sudo）。

下面的命令（从 whois 软件包中获得）可用于生成密码的基于 crypt(3) 的 SHA-512 值：

```
mkpasswd -m sha-512
```

B.4.7 时钟与时区设置

```
# Controls whether or not the hardware clock is set to UTC.
d-i clock-setup/utc boolean true

# You may set this to any valid setting for $TZ; see the contents of
# /usr/share/zoneinfo/ for valid values.
d-i time/zone string US/Eastern

# Controls whether to use NTP to set the clock during the install
d-i clock-setup/ntp boolean true
# NTP server to use. The default is almost always fine here.
#d-i clock-setup/ntp-server string ntp.example.com
```

B.4.8 分区

使用预置进行硬盘分区受限于 partman-auto 提供的支持。您可以选择使用磁盘上已有的空闲分区或者整个磁盘。磁盘的布局将取决于所使用的预定义方案，用户自定义的方案文件或预置文件包含的方案。

已经支持包括 RAID、LVM 和加密高级分区设置的预置，但对于非预置安装的分区仍然弹性不足。

下面的例子仅提供了使用方案的最基本信息。详细的内容清参考文件 partman-auto-recipe.txt 和 partman-auto-raid-recipe.txt，它们含在 debian-installer 软件包里。这两个文件也可以从 debian-installer source repository获取。注意不同发布版支持的功能会有所改变。

警告

磁盘的标识基于对应驱动加载的次序。如果系统里面有多个磁盘，要确定预置使用了正确的那一个。

B.4.8.1 分区示例

```
# If the system has free space you can choose to only partition that space.
# This is only honoured if partman-auto/method (below) is not set.
#d-i partman-auto/init_automatically_partition select biggest_free

# Alternatively, you may specify a disk to partition. If the system has only
# one disk the installer will default to using that, but otherwise the device
# name must be given in traditional, non-devfs format (so e.g. /dev/sda
# and not e.g. /dev/discs/disc0/disc).
# For example, to use the first SCSI/SATA hard disk:
#d-i partman-auto/disk string /dev/sda
# In addition, you'll need to specify the method to use.
# The presently available methods are:
# - regular: use the usual partition types for your architecture
# - lvm:     use LVM to partition the disk
# - crypto:  use LVM within an encrypted partition
d-i partman-auto/method string lvm

# You can define the amount of space that will be used for the LVM volume
# group. It can either be a size with its unit (eg. 20 GB), a percentage of
```

```
# free space or the 'max' keyword.
d-i partman-auto-lvm/guided_size string max

# If one of the disks that are going to be automatically partitioned
# contains an old LVM configuration, the user will normally receive a
# warning. This can be preseeded away...
d-i partman-lvm/device_remove_lvm boolean true
# The same applies to pre-existing software RAID array:
d-i partman-md/device_remove_md boolean true
# And the same goes for the confirmation to write the lvm partitions.
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true

# You can choose one of the predefined partitioning recipes:
# - atomic: all files in one partition
# - home:   separate /home partition
# - multi:  separate /home, /var, and /tmp partitions
# - server: separate /var and /srv partitions; swap limitted to 1G
# - small_disk: scheme dedicated to small harddrives (under 10GB)
d-i partman-auto/choose_recipe select atomic

# Or provide a recipe of your own...
# If you have a way to get a recipe file into the d-i environment, you can
# just point at it.
#d-i partman-auto/expert_recipe_file string /hd-media/recipe

# If not, you can put an entire recipe into the preconfiguration file in one
# (logical) line. This example creates a small /boot partition, suitable
# swap, and uses the rest of the space for the root partition:
#d-i partman-auto/expert_recipe string                         \
#      boot-root ::                                            \
#              40 50 100 ext3                                  \
#                      $primary{ } $bootable{ }                \
#                      method{ format } format{ }              \
#                      use_filesystem{ } filesystem{ ext3 }    \
#                      mountpoint{ /boot }                     \
#              .                                               \
#              500 10000 1000000000 ext3                       \
#                      method{ format } format{ }              \
#                      use_filesystem{ } filesystem{ ext3 }    \
#                      mountpoint{ / }                         \
#              .                                               \
#              64 512 300% linux-swap                          \
#                      method{ swap } format{ }                \
#              .

# The full recipe format is documented in the file partman-auto-recipe.txt
# included in the 'debian-installer' package or available from D-I source
# repository. This also documents how to specify settings such as file
# system labels, volume group names and which physical devices to include
# in a volume group.

## Partitioning for EFI
# If your system needs an EFI partition you could add something like
# this to the recipe above, as the first element in the recipe:
#               538 538 1075 free                               \
#                      $iflabel{ gpt }                         \
#                      $reusemethod{ }                         \
#                      method{ efi }                           \
#                      format{ }                               \
#               .                                              \
#
# The fragment above is for the amd64 architecture; the details may be
# different on other architectures. The 'partman-auto' package in the
```

```
# D-I source repository may have an example you can follow.

# This makes partman automatically partition without confirmation, provided
# that you told it what to do using one of the methods above.
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# Force UEFI booting ('BIOS compatibility' will be lost). Default: false.
#d-i partman-efi/non_efi_system boolean true
# Ensure the partition table is GPT - this is required for EFI
#d-i partman-partitioning/choose_label select gpt
#d-i partman-partitioning/default_label string gpt

# When disk encryption is enabled, skip wiping the partitions beforehand.
#d-i partman-auto-crypto/erase_disks boolean false
```

B.4.8.2 使用 RAID 分区

您可以使用预置来建立软 RAID 阵列。支持 RAID 等级 0、1、5、6 和 10，建立降级阵列 (degraded arrays) 和指定额外设备。

警告

这种自动分区方式很容易出错。debian-installer 开发人员对它的功能只进行了相对较少的测试。让各种方案都正确 (有意义而不冲突) 的责任在于用户。如果您遇到问题，检查一下 /var/log/syslog。

```
# The method should be set to "raid".
#d-i partman-auto/method string raid
# Specify the disks to be partitioned. They will all get the same layout,
# so this will only work if the disks are the same size.
#d-i partman-auto/disk string /dev/sda /dev/sdb

# Next you need to specify the physical partitions that will be used.
#d-i partman-auto/expert_recipe string \
#      multiraid ::                                          \
#              1000 5000 4000 raid                           \
#                      $primary{ } method{ raid }            \
#              .                                             \
#              64 512 300% raid                              \
#                      method{ raid }                        \
#              .                                             \
#              500 10000 1000000000 raid                     \
#                      method{ raid }                        \
#              .

# Last you need to specify how the previously defined partitions will be
# used in the RAID setup. Remember to use the correct partition numbers
# for logical partitions. RAID levels 0, 1, 5, 6 and 10 are supported;
# devices are separated using "#".
# Parameters are:
# <raidtype> <devcount> <sparecount> <fstype> <mountpoint> \
#          <devices> <sparedevices>

#d-i partman-auto-raid/recipe string \
#    1 2 0 ext3 /                    \
#          /dev/sda1#/dev/sdb1       \
#    .                               \
```

```
#    1 2 0 swap -                       \
#          /dev/sda5#/dev/sdb5          \
#    .                                  \
#    0 2 0 ext3 /home                   \
#          /dev/sda6#/dev/sdb6          \
#    .

# For additional information see the file partman-auto-raid-recipe.txt
# included in the 'debian-installer' package or available from D-I source
# repository.

# This makes partman automatically partition without confirmation.
d-i partman-md/confirm boolean true
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
```

B.4.8.3 分区挂载控制

文件系统一般使用 UUID 作为关键字挂载; 这使得在设备名变更的情况下仍然可以正常挂载。UUID 本身很长不容易阅读，因此，安装程序也可以依照您的意愿使用传统设备名或者指定标签 (label) 挂载文件系统。假如安装程序使用标签方式，那些没有标签的文件系统仍然使用 UUID 方式挂载。

具有固定名称的设备，比如 LVM 逻辑卷，将继续使用它们自己的名字而不是 UUID 方式挂载。

> **警告**
>
> 传统设备名会根据内核在引导时发现的次序进行调整，这将导致挂载错误的文件系统。与此类似，假如您插入一个新磁盘或 USB 设备，标签也可能有冲突发生。这样系统启动后会出现一些随机的情况。

```
# The default is to mount by UUID, but you can also choose "traditional" to
# use traditional device names, or "label" to try filesystem labels before
# falling back to UUIDs.
#d-i partman/mount_style select uuid
```

B.4.9 基本系统安装

本阶段的安装并没有多少东西需要预置。仅有一个与内核安装相关的问题。

```
# Configure APT to not install recommended packages by default. Use of this
# option can result in an incomplete system and should only be used by very
# experienced users.
#d-i base-installer/install-recommends boolean false

# The kernel image (meta) package to be installed; "none" can be used if no
# kernel is to be installed.
#d-i base-installer/kernel/image string linux-image-686
```

B.4.10 设置 apt

设置 `/etc/apt/sources.list` 和其他的基本配置选项，将自动地基于您使用的安装方式以及前面问题的回答。您也可以选择性地安装其他 (或本地) 的仓库。

```
# Choose, if you want to scan additional installation media
# (default: false).
d-i apt-setup/cdrom/set-first boolean false
# You can choose to install non-free firmware.
```

```
#d-i apt-setup/non-free-firmware boolean true
# You can choose to install non-free and contrib software.
#d-i apt-setup/non-free boolean true
#d-i apt-setup/contrib boolean true
# Uncomment the following line, if you don't want to have the sources.list
# entry for a DVD/BD installation image active in the installed system
# (entries for netinst or CD images will be disabled anyway, regardless of
# this setting).
#d-i apt-setup/disable-cdrom-entries boolean true
# Uncomment this if you don't want to use a network mirror.
#d-i apt-setup/use_mirror boolean false
# Select which update services to use; define the mirrors to be used.
# Values shown below are the normal defaults.
#d-i apt-setup/services-select multiselect security, updates
#d-i apt-setup/security_host string security.debian.org

# Additional repositories, local[0-9] available
#d-i apt-setup/local0/repository string \
#       http://local.server/debian stable main
#d-i apt-setup/local0/comment string local server
# Enable deb-src lines
#d-i apt-setup/local0/source boolean true
# URL to the public key of the local repository; you must provide a key or
# apt will complain about the unauthenticated repository and so the
# sources.list line will be left commented out.
#d-i apt-setup/local0/key string http://local.server/key
# or one can provide it in-line by base64 encoding the contents of the
# key file (with 'base64 -w0') and specifying it thus:
#d-i apt-setup/local0/key string base64:// ↩
    LS0tLS1CRUdJTiBQR1AgUFVCTElDIEtFWSBCTE9DSy0tLS0tCi4uLgo=
# The content of the key file is checked to see if it appears to be ASCII- ↩
    armoured.
# If so it will be saved with an ".asc" extension, otherwise it gets a '.gpg'  ↩
    extension.
# "keybox database" format is currently not supported. (see generators/60local in ↩
     apt-setup's source)

# By default the installer requires that repositories be authenticated
# using a known gpg key. This setting can be used to disable that
# authentication. Warning: Insecure, not recommended.
#d-i debian-installer/allow_unauthenticated boolean true

# Uncomment this to add multiarch configuration for i386
#d-i apt-setup/multiarch string i386
```

B.4.11 选择软件包

您可以选择安装存在的任务的组合。本文编撰时已有的任务包括：

- standard（标准工具）
- desktop（图形化桌面）
- gnome-desktop （Gnome 桌面）
- xfce-desktop （XFCE 桌面）
- kde-desktop （KDE Plasma 桌面）
- cinnamon-desktop （Cinnamon 桌面）
- mate-desktop （MATE 桌面）desktop)

- lxde-desktop（LXDE 桌面）
- web-server（web 服务器）
- ssh-server（SSH 服务器）

您可以不选任务，并用其他方法安装一系列软件包。我们推荐选上 standard 任务。

或者，如果您根本不希望 tasksel 对话框出现，请预置 pkgsel/run_tasksel（这种情况将不会通过 tasksel 安装任何软件包）。

如果您打算安装一些安装任务之外的独立软件包，可以使用参数 pkgsel/include。该参数的值可以是用逗号或空格分开的软件包列表，便于在内核命令行上使用。

```
#tasksel tasksel/first multiselect standard, web-server, kde-desktop

# Or choose to not get the tasksel dialog displayed at all (and don't install
# any packages):
#d-i pkgsel/run_tasksel boolean false

# Individual additional packages to install
#d-i pkgsel/include string openssh-server build-essential
# Whether to upgrade packages after debootstrap.
# Allowed values: none, safe-upgrade, full-upgrade
#d-i pkgsel/upgrade select none

# You can choose, if your system will report back on what software you have
# installed, and what software you use. The default is not to report back,
# but sending reports helps the project determine what software is most
# popular and should be included on the first DVD.
#popularity-contest popularity-contest/participate boolean false
```

B.4.12 完成安装

```
# During installations from serial console, the regular virtual consoles
# (VT1-VT6) are normally disabled in /etc/inittab. Uncomment the next
# line to prevent this.
#d-i finish-install/keep-consoles boolean true

# Avoid that last message about the install being complete.
d-i finish-install/reboot_in_progress note

# This will prevent the installer from ejecting the disk during the reboot,
# which is useful in some situations.
#d-i cdrom-detect/eject boolean false

# This is how to make the installer shutdown when finished, but not
# reboot into the installed system.
#d-i debian-installer/exit/halt boolean true
# This will power off the machine instead of just halting it.
#d-i debian-installer/exit/poweroff boolean true
```

B.4.13 预置其他的软件包

```
# Depending on what software you choose to install, or if things go wrong
# during the installation process, it's possible that other questions may
# be asked. You can preseed those too, of course. To get a list of every
# possible question that could be asked during an install, do an
# installation, and then run these commands:
#   debconf-get-selections --installer > file
#   debconf-get-selections >> file
```

B.5 高级选项

B.5.1 安装过程中运行用户命令

预置工具提供了一个非常强大和灵活的选项，可以在安装过程的一些环节运行命令或脚本。

当目标系统的文件系统挂载后，它可以在 /target 中使用。如果使用安装 CD/DVD，则挂载后可以在 /cdrom 中使用。

```
# d-i preseeding is inherently not secure. Nothing in the installer checks
# for attempts at buffer overflows or other exploits of the values of a
# preconfiguration file like this one. Only use preconfiguration files from
# trusted locations! To drive that home, and because it's generally useful,
# here's a way to run any shell command you'd like inside the installer,
# automatically.

# This first command is run as early as possible, just after
# preseeding is read.
#d-i preseed/early_command string anna-install some-udeb
# This command is run immediately before the partitioner starts. It may be
# useful to apply dynamic partitioner preseeding that depends on the state
# of the disks (which may not be visible when preseed/early_command runs).
#d-i partman/early_command \
#       string debconf-set partman-auto/disk "$(list-devices disk | head -n1)"
# This command is run just before the install finishes, but when there is
# still a usable /target directory. You can chroot to /target and use it
# directly, or use the apt-install and in-target commands to easily install
# packages and run commands in the target system.
#d-i preseed/late_command string apt-install zsh; in-target chsh -s /bin/zsh
```

B.5.2 使用预置修改默认值

也可以使用预置修改问题的默认答案，但同时显示问题。这需要在问题设定值之后，必须将 seen 旗标重置为“false”。

```
d-i foo/bar string value
d-i foo/bar seen false
```

要对所有问题达到同样效果，可以通过在启动引导符后面设置 preseed/interactive=true。这对于测试和调试您的预置文件很有帮助。

注意，“d-i”owner 只应该用于安装程序里面的变量。对于那些属于目标系统软件包的变量，您应该使用软件包的名字替代。请参考第 B.2.2 节的脚注。

如果使用引导参数预置，您可以通过“?=”操作符让安装程序询问相关的问题，例如，foo/bar?=value (或者 owner:foo/bar?=value)。当然，这只会影响在安装时显示出来的问题对应的参数，而不是“内部的”参数。

获得更多的调试信息，需使用引导参数 DEBCONF_DEBUG=5。这将让 debconf 打印更多的细节，包括当前变量设置和每个软件包安装脚本的进度。

B.5.3 预置文件链

可以在一个预置文件里面包含另外的预置文件。这些文件里面的任何设置将覆盖前面加载的设置。有一种用法，例如，通用的网络设置放在一个文件，其他规格的设置放在另外的文件里面。

```
# More than one file can be listed, separated by spaces; all will be
# loaded. The included files can have preseed/include directives of their
# own as well. Note that if the filenames are relative, they are taken from
# the same directory as the preconfiguration file that includes them.
#d-i preseed/include string x.cfg

# The installer can optionally verify checksums of preconfiguration files
# before using them. Currently only md5sums are supported, list the md5sums
# in the same order as the list of files to include.
#d-i preseed/include/checksum string 5da499872becccfeda2c4872f9171c3d
```

```
# More flexibly, this runs a shell command and if it outputs the names of
# preconfiguration files, includes those files.
#d-i preseed/include_command \
#       string if [ "`hostname`" = bob ]; then echo bob.cfg; fi

# Most flexibly of all, this downloads a program and runs it. The program
# can use commands such as debconf-set to manipulate the debconf database.
# More than one script can be listed, separated by spaces.
# Note that if the filenames are relative, they are taken from the same
# directory as the preconfiguration file that runs them.
#d-i preseed/run string foo.sh
```

也可以通过在前面设置 preseed/url 从 initrd 或者文件预置链载进入网络预置。这将使网络预置在网络连通时执行。您应该小心使用，因为这是两种预置，意味着您有机会执行 preseed/early 命令，第二个可以在网络连通后使用。

Appendix C

为 Debian 准备分区

C.1 决定 Debian 分区以及大小

最小化情况下，GNU/Linux 需要一个分区。您可以有一个包含整个操作系统，应用程序以及您私人数据的分区。大多数人感到一个分离的交换分区也是很有必要的，尽管它并非严格要求。“交换分区”是一个操作系统用的空白空间，允许系统使用磁盘来提供“虚拟内存”。通过放置交换空间至一个分离的分区，Linux 可以非常有效地使用它。强迫 Linux 使用一个普通文件作为交换空间也是可行的，但是并不推荐这种做法。

但是大多数人选择给 GNU/Linux 更多的分区。有两个原因使您愿意把文件系统分置到一些较小的分区上。首先是安全性。如果出现问题破坏了文件系统，一般来说只有一个分区受到影响。因此，您只需要更换 (从您精心维护的备份中) 系统的一部分。在最小化情况下，您应该考虑创建所谓的“根分区”。它含有最基础的系统组件。如果其它分区出现了问题，您仍然可以引导至 GNU/Linux 来修复系统。这个可以让您不用从头重新安装系统。

第二个理由对商业设置来说更重要，但它实际上取决于机器的用途。比如，一个邮件服务器接受到很多垃圾邮件会很快填满一个分区，如果邮件服务器的 `/var/mail` 建立在一个单独的分区上面，您即使收到太多的垃圾邮件，大多数系统仍然会继续工作。

使用更多分区的惟一缺点很难知道您将来的需求。如果建立的一个分区容量太小，您要么重装系统，要么不停地移动数据来给分区分配空间。另一种情况，如果您建立的分区过大，将浪费一些原可以用在别的地方的空间。虽然今天的磁盘很便宜，但为什么要白白浪费金钱呢？

C.2 目录树

Debian GNU/Linux 依照 文件系统层次标准 (Filesystem Hierarchy Standard) 安排文件以及文件命名。该标准让用户和软件程序预知文件以及目录的位置。根目录只是简单表现为 `/`。在根层次，所有的 Debian 系统包括这些目录：

目录	内容
`bin`	基本命令执行文件
`boot`	boot loader 的静态链接文件
`dev`	设备文件
`etc`	主机特定的系统配置
`home`	用户目录
`lib`	基本共享库以及内核模块
`media`	用于移动介质的挂载点
`mnt`	用于临时挂载文件系统
`proc`	系统信息的虚拟目录
`root`	root 用户的目录
`run`	运行变量数据
`sbin`	基本系统命令执行文件
`sys`	系统信息的虚拟目录
`tmp`	临时文件
`usr`	第二级目录
`var`	不断变化的数据

目录	内容
srv	系统提供的用于 service 的数据
opt	附加的应用程序软件包

下面列出关于目录和分区的重要考虑。要注意磁盘的使用随系统配置和特定用途有很大变化。这里所推荐的是通用指引，提供如何分区的起点。

- 根分区 / 必须总是物理地包含 /etc、/bin、/sbin、/lib、/dev 和 /usr，否则您将不能启动系统。这意味着您需要为包含 /usr 分区的根分区提供 600–750MB 大小空间，或者为工作站或服务器的安装提供 5–6GB 大小空间。
- /var：所有的可变数据，如新闻组文章、电子邮件、网站、数据库、软件包系统的缓存等等，将被放入这个目录。这个目录的大小取决于您计算机的用途，但是对大多数人来说，将主要用于软件包系统的管理工具。如果您准备一次性地做一个全面的安装，包含 Debian 提供的所有程序，/var 下面 2 或 3GB 的空间应该够用。如果您准备逐个安装 (也就是说，安装服务程序和工具，然后是文本处理工具，然后是 X，...)，您准备 300–500 MB 就行。要是硬盘空间很紧张，并且不准备做主版本的系统升级，可以为其准备 30 或 40MB 大小。
- /tmp: 程序创建的临时数据大都存到这个目录。通常 40–100 MB 应该足够。一些应用程序—包括归档处理程序、CD/DVD 制作工具和多媒体软件—可能会使用 /tmp 临时保存映像文件。如果要使用这些程序，应该相应地调整 /tmp 目录的大小。
- /home：每个用户将放置他的私有数据到这个目录的子目录下。其大小取决于将有多少用户使用系统，以及有什么样文件放在他们的目录下。根据规划的用途，应该为每个用户准备 100MB 空间，不过应该按您的需求调整。假如您在 home 目录下计划保存大量的多媒体文件 (图片、MP3、电影)，该预备更多的空间。

C.3 推荐的分区方案

对于新用户，个人 Debian 系统、家庭系统以及其它单用户配置来说，一个单一 / 分区 (加上交换分区) 可能是最容易的方案。推荐的分区类型是 ext4。

对于多用户系统或者有大量磁盘空间的系统，最好将 /var、/tmp 和 /home 分配到他们各自的分区，与 / 分区分离开来。

如果您计划安装 Debian 发布版以外的程序，也许需要一个独立的 /usr/local 分区。如果您的机器是一个邮件服务器，也许需要为 /var/mail 建立一个独立的分区。如果您要创建一个有很多用户的服务器，一个独立的、大的 /home 分区比较好。一般来说，分区状况在每台计算机都不同，这取决于它们的用途。

对于非常复杂的系统来，您应该阅读 Multi Disk HOWTO。其中包含了更深入的内容，对于很多 ISP 和架设服务器的用户很有用。

在交换分区大小方面，这里有很多观点。其中一条经验是使用和内存相同大小的交换分区。在大多数情况下它也不应该小于 512MB。当然，这些规则有一些例外。

举一个例子，一个较老的家用机器上，可能有 512MB 内存和一个 /dev/sda 上的 20GB SATA 驱动器。/dev/sda1 上有一个 8GB 的分区用于其它操作系统，一个 512MB 交换分区位于 /dev/sda3，以及一个 11.4GB 空间 /dev/sda2 用于 Linux 分区。

对于不同任务需要多大的空间，安装系统完成后的情况有个概念，您或许会感兴趣知道，请查阅第 D.2 节。

C.4 Linux 里面的设备名称

Linux 的磁盘和分区命名可能与其它操作系统不同。您需要知道创建和挂接分区时 Linux 所使用的设备名称。这里是基本的命名方案：

- 第一个 DASD 设备被命名为 /dev/dasda。
- 第二个 DASD 设备被命名为 /dev/dasdb，等等。

每个磁盘的分区名的显示，是通过在磁盘名称后面附加一个十进制数字：dasda1 以及 dasda2 表示您系统上第一个 DASD 设备的第一个以及第二个分区。

C.5 Debian 分区程序

Debian 开发人员选用了几种在不同硬盘和不同体系结构上的分区程序。下面是一个适合您的体系的程序列表。

partman 推荐的 Debian 分区工具。这把瑞士军刀可以修改分区大小，创建文件系统和关联分区至挂接点。

fdasd S/390 fdisk 的版本; 请阅读 fdasd 手册页或者 Device Drivers and Installation Commands 的第 13 章获得详细信息。

当您选择了 Partition disks (或类似的)，默认会执行其中的一个程序。在 VT2 命令行上也可以使用其他的分区工具，但不推荐采用这种方式。

Appendix D

其他

D.1 Linux 里的设备

Linux 的许多特殊文件可以在 /dev 目录下找到。这些文件称为设备文件，其行为与普通文件不同。大多数设备文件的类型是块设备和字符设备。这些文件是访问硬件的驱动程序 (Linux 内核的一部分) 的接口。另外一些，不那么常见的类型，是命名管道 (pipe)。下表中列出了最重要的设备文件。

sda	第一块硬盘
sdb	第二块硬盘
sda1	第一块硬盘上的第一个分区
sdb7	第二块硬盘上的第七个分区

sr0	第一个 CD-ROM
sr1	第二个 CD-ROM

ttyS0	串口 0，也通常使用 COM1 指代
ttyS1	串口 1，也通常使用 COM2 指代
psaux	PS/2 鼠标设备
gpmdata	伪设备，中转从 GPM（鼠标）服务传来的数据

cdrom	指向光驱的符号链接
mouse	指向鼠标设备文件的符号链接

null	所有写入该设备的东西都会消失
zero	可以从该设备永无休止地读出零

D.2 各软件集所需的磁盘空间

对于 amd64 架构，基本安装默认使用内核，包含了所有的标准软件包，需要 1242MB 磁盘空间。最小化的基本安装，没有选择“Standard system utilities”任务，需要 1012MB。

重要

在各种情况下，这都是安装完成并且删除临时文件之后，实际使用的磁盘空间。这还不包括文件系统的额外消耗，比如 journal 文件。也就是说，在安装和使用过程中，都会需要更多的磁盘空间。

下面的表格一一列出了 tasksel 中每个软件集所需的磁盘空间大小。该报表是使用 aptitude 生成的。请注意，有些软件集之间会有相互重叠的部分，所以两个软件集一起安装后所占的空间，可能会小于这两个软件集分别安装所占的空间。

默认情况下，安装程序安装 GNOME 桌面环境，其他的桌面环境可以通过使用另外特殊的安装映象，或者在安装时指定所需的桌面环境 (参阅第 6.3.6.2 节)。

注意，您需要将表格中列出的尺寸加上基本安装尺寸以决定分区的大小。多数所列的尺寸，如“安装尺寸”用在 /usr 和 /lib 里面；列在“下载尺寸”的是 (临时地) 用在 /var 里面。

任务	安装大小 (MB)	下载尺寸 (MB)	用于安装所需空间 (MB)
桌面环境			
• GNOME (默认)	3216	859	4075
• KDE Plasma	4584	1316	5900
• Xfce	2509	683	3192
• LXDE	2539	693	3232
• MATE	2851	762	3613
• Cinnamon	4676	1324	6000
网页服务器	85	19	104
SSH 服务器	2	1	3

如果安装英语之外的其他语言，如果与您选择的语言相符，tasksel 可自动安装本地化任务。每种语言需要的空间不同；您应该留有 350MB 以上的空间用来下载和安装。

D.3 通过 Unix/Linux 系统来安装 Debian GNU/Linux

本节说明如何通过现有的 Unix 或 Linux 系统来安装 Debian GNU/Linux，而非使用本手册其余部分谈到的由菜单驱动的安装程序。这篇“交叉安装 (cross-install)”HOWTO 是应那些原来使用 Redhat、Mandriva 和 SUSE，现转到 Debian GNU/Linux 的用户撰写的。在本节中，我们假设您对于输入 *nix 命令以及在文件系统中穿梭移动并不生疏。同时，在本节中，$ 标志着这个命令是在用户当前使用的系统中输入的，而 # 则表示这个命令是在 Debian chroot 环境中输入的。

一旦您根据自己的要求配置好了新的 Debian 系统，就可以将您原来系统中的用户数据 (如果有的话) 悉数迁移到新系统里去，然后继续滚动。一言以蔽之，这是一次“不停机”的 Debian GNU/Linux 安装。同时，这也是处理那些因为硬件原因不能通过任何引导或者安装介质完成安装的权宜之计。

> **注意**
>
>
>
> 由于绝大多数操作都是手动进行，应该记住这将需要您自己对系统做大量地基本配置，并且要具有比常规安装更多的 Debian 和 Linux 知识。不要期望这种执行的结果与常规安装完全相同。您还应该明白这一过程只是建立一个系统的基本步骤。可能还需要额外的安装、或者配置步骤。

D.3.1 准备开始

如果需重新分区，用您当前的 *nix 的分区工具，至少要划分出一个文件系统，以及交换分区。如果只用控制台，至少要准备 1012MB 空间，倘若想装 X，就需要至少 2539MB 的空间了 (安装 GNOME 或 KDE 这样的桌面环境，还要更多)。

下一步，在分区上创建文件系统。例如，在 /dev/sda6 分区 (以后的例子中，将把它作为 root 分区) 建立一个 ext3 格式的文件系统：

```
# mke2fs -j /dev/sda6
```

建立 ext2 格式的文件系统，要去掉 -j。

初始化并激活交换分区 (请把分区号替换成您希望用作 Debian 交换分区的分区号)：

```
# mkswap /dev/sda5
# sync
# swapon /dev/sda5
```

把一个分区挂载到 /mnt/debinst(这是安装的位置，将来作为您新系统的根 (/) 文件系统)。挂载点的名称是任意的，后面的步骤将会用到。

```
# mkdir /mnt/debinst
# mount /dev/sda6 /mnt/debinst
```

> **注意**
>
>
>
> 如果您打算将部分文件系统 (例如，/usr) 放到独立的分区，在进入下一阶段之前，需要手动创建和加载这些目录。

D.3.2 安装 debootstrap

Debian 安装程序使用的工具，同时也作为安装 Debian 基本系统的正式方法，是 debootstrap。它调用了 wget 和 ar，而其他的仅依赖 /bin/sh 和基本的 Unix/Linux 工具[1]。如果您的系统里还没有 wget 和 ar，首先安装它们，然后下载并安装 debootstrap。

或者您也可以通过下面的步骤手动安装。先新建一个 work 目录，用于解压.deb 文件：

```
# mkdir work
# cd work
```

debootstrap 的二进制文件位于 Debian archive (请确保选择了合适的架构文件)。下载 debootstrap .deb，它位于 pool，把软件包拷贝到 work 目录，并解压该文件。要将这些文件安装到系统，您需要 root 权限。

```
# ar -x debootstrap_0.X.X_all.deb
# cd /
# zcat /full-path-to-work/work/data.tar.gz | tar xv
```

D.3.3 运行 debootstrap

运行 debootstrap 后，它会从 archive 直接下载所需的文件。在下面的命令示例中，您可以根据自己的网络情况，选择一个较近的 Debian archive 镜像，替换 http.us.debian.org/debian。镜像列表位于 http://www.debian.org/mirror/list。

如果您已经把一张 trixie 版本的 Debian GNU/Linux 安装映像挂载到 /cdrom，那么也可以把 http URL 替换成 file URL，即：file:/cdrom/debian/

用下面其中一个架构替换 debootstrap 命令中的 ARCH：amd64, arm64, armel, armhf, i386, mips64el, mipsel, ppc64el, riscv64, s390x。

```
# /usr/sbin/debootstrap --arch ARCH trixie \
    /mnt/debinst http://ftp.us.debian.org/debian
```

如果目的架构与主机不同，那么应该添加 --foreign 选项。

D.3.4 基本系统的配置

现在您的磁盘上已经有了一个功能虽弱但却是真正的 Debian 系统。键入 chroot 命令进入：

```
# LANG=C.UTF-8 chroot /mnt/debinst /bin/bash
```

如果目标架构与主机不同，那么需要首先将 qemu-user-static 复制到新的主机：

```
# cp /usr/bin/qemu-ARCH-static /mnt/debinst/usr/bin
# LANG=C.UTF-8 chroot /mnt/debinst qemu-ARCH-static /bin/bash
```

chrooting 之后，会需要设置终端定义与 Debian 基本系统兼容，例如：

[1] 它们包括 GNU core 工具和命令，像 sed、grep、tar 和 gzip。

```
# export TERM=xterm-color
```

依赖于 TERM 的值，会必须安装 ncurses-term 软件包，来从中得到支持。
如果目标架构与主机不同，则需要完成多阶段自举：

```
/debootstrap/debootstrap --second-stage
```

D.3.4.1 创建设备文件

此时，/dev/ 只含有非常基本的设备文件。安装的后续步骤可能还需要更多的设备文件。使用什么样的文件和方法，取决于安装使用的系统、是否采用内核模块，以及在新系统中采用动态 (比如，使用 udev) 或静态设备文件。
可以使用的选项有：

- 安装 makedev 软件包，并创建默认的静态设备文件，使用 (chroot 以后)

  ```
  # apt install makedev
  # mount none /proc -t proc
  # cd /dev
  # MAKEDEV generic
  ```

- 手动创建指定的设备文件，使用 MAKEDEV
- 将 host 系统上加载的 /dev 绑定到 target 系统的 /dev；注意，有些软件包的 postinst 脚本会创建设备文件，所以要小心使用改选项

D.3.4.2 分区的挂载

您需要新建 /etc/fstab。

```
# editor /etc/fstab
```

这里有个样板，您可以根据自己的情况对它进行修改：

```
# /etc/fstab: static file system information.
#
# file system    mount point   type    options                 dump pass
/dev/XXX         /             ext3    defaults                0    1
/dev/XXX         /boot         ext3    ro,nosuid,nodev         0    2

/dev/XXX         none          swap    sw                      0    0
proc             /proc         proc    defaults                0    0

/dev/cdrom       /media/cdrom  iso9660 noauto,ro,user,exec     0    0

/dev/XXX         /tmp          ext3    rw,nosuid,nodev         0    2
/dev/XXX         /var          ext3    rw,nosuid,nodev         0    2
/dev/XXX         /usr          ext3    rw,nodev                0    2
/dev/XXX         /home         ext3    rw,nosuid,nodev         0    2
```

可以通过 mount -a 来挂载 /etc/fstab 中所标明的所有文件系统，您也可以逐个分别地挂载这些文件系统，像：

```
# mount /path   # e.g.: mount /usr
```

当前的 Debian 系统为可移动介质创建的挂载点位于 /media, 但为了保持兼容，仍有符号链接处于 / 目录下面。如果需要，可以像下面这样创建这些挂载点:

```
# cd /media
# mkdir cdrom0
# ln -s cdrom0 cdrom
# cd /
# ln -s media/cdrom
```

您可以多次挂载 proc 文件系统，通常是在 /proc 位置，但也可以选择任意的地方。如果没有使用 mount -a，请在进入下一步前先将 proc 挂载：

```
# mount -t proc proc /proc
```

ls /proc 命令现在应该列出非空目录。如果失败，您应该从 chroot: 以外加载 proc：

```
# mount -t proc proc /mnt/debinst/proc
```

D.3.4.3 设置时区

将 /etc/adjtime 的第三行设置为“UTC”或“LOCAL”，决定系统是否将认为硬件时钟被设置为 UTC 各自的本地时间。后面的命令允许设置它。

```
# editor /etc/adjtime
```

这是个例子：

```
0.0 0 0.0
0
UTC
```

后面的命令允许选择时区。

```
# dpkg-reconfigure tzdata
```

D.3.4.4 网络的配置

要配置网络，编辑 /etc/network/interfaces、/etc/resolv.conf、/etc/hostname 和 /etc/hosts。

```
# editor /etc/network/interfaces
```

以下的简单例子来自 /usr/share/doc/ifupdown/examples：

```
###############################################################################
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
# See the interfaces(5) manpage for information on what options are
# available.
###############################################################################

# The loopback interface isn't really required any longer, but can be used
# if needed.
#
# auto lo
# iface lo inet loopback

# To use dhcp:
#
# auto eth0
# iface eth0 inet dhcp

# An example static IP setup: (network, broadcast and gateway are optional)
#
# auto eth0
# iface eth0 inet static
#     address 192.168.0.42
#     network 192.168.0.0
#     netmask 255.255.255.0
#     broadcast 192.168.0.255
#     gateway 192.168.0.1
```

在 /etc/resolv.conf 输入您的名称服务器和搜索域：

```
# editor /etc/resolv.conf
```

来自 /etc/resolv.conf 的简单示例:

```
search example.com
nameserver 10.1.1.36
nameserver 192.168.9.100
```

输入系统的主机名 (2 到 63 个字符):

```
# echo DebianHostName > /etc/hostname
```

简单的带 IPv6 支持的 /etc/hosts:

```
127.0.0.1 localhost
127.0.1.1 DebianHostName

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

如果您有多张网卡，您应该在 /etc/modules 文件内以适当的顺序排列驱动程序模块名。然后在启动期间，每张卡将会获得您期望的接口名称 (eth0，eth1，等等)。

D.3.4.5 配置 Apt

debootstrap 会创建一个非常基本的 /etc/apt/sources.list 文件用于安装额外的软件包。但您还可以增加一些其他的源，比如源码包和安全更新:

```
deb-src http://ftp.us.debian.org/debian trixie main

deb http://security.debian.org/ trixie-security main
deb-src http://security.debian.org/ trixie-security main
```

修改源列表之后一定要执行 aptitude update。

D.3.4.6 本地化和键盘的配置

配置您的本地化设定，以便使用英语以外的语言，需要安装 locales 支持软件包，然后进行配置:

```
# aptitude install locales
# dpkg-reconfigure locales
```

配置您的键盘 (如果需要):

```
# apt install console-setup
# dpkg-reconfigure keyboard-configuration
```

注意，在 chroot 里面不能设置键盘，但重新启动以后可以。

D.3.5 内核的安装

要想启动这个系统，您很可能还需要一个 Linux 内核和一个启动加载程序。可以用下面指令找到那些预先打包的内核:

```
# apt search linux-image
```

然后根据所选取的软件包名称安装内核软件包。

```
# apt install linux-image-arch-etc
```

D.3.6 Boot Loader 的设置

要让您的 Debian GNU/Linux 系统能够启动，还要创建 boot loader，好让它能加载安装的内核并用上新的根分区。注意 debootstrap 不会安装 boot loader，因此您要在 Debian 的 chroot 中使用 aptitude 完成。

注意，这里假设已经创建了 /dev/sda 设备文件。另外一种方法是安装 grub2，但它不属于本附录讨论范围。

D.3.7 远程访问：安装 SSH 并设置密码

如果您可以通过控制台登录系统，可以跳过此节。如果以后可以通过网络访问系统，则需要安装 SSH 并设置访问权限。

```
# apt install ssh
```

默认禁用 root 登录密码，因此可以通过设置密码来设置访问权限，并用密码重新启用 root 登录：

```
# passwd
# editor /etc/ssh/sshd_config
```

这是启用的选项：

```
PermitRootLogin yes
```

也可通过在 root 帐户中添加 ssh 密钥来设置访问权限：

```
# mkdir /root/.ssh
# cat << EOF > /root/.ssh/authorized_keys
ssh-rsa ....
EOF
```

最后，还可以通过添加非 root 用户和设置密码来设置访问权限：

```
# adduser joe
# passwd joe
```

D.3.8 最后一笔

如前所述，安装的是很基本的系统。如果您想让它更成熟一些，有一个简单的方法，使用“standard”优先级来安装所有的软件包：

```
# tasksel install standard
```

当然，您也可以用 apt 来安装单独的软件包。

安装之后，/var/cache/apt/archives/ 里面会有大量下载的软件包。您可以运行下面命令释放空间：

```
# apt clean
```

Appendix E

文案

E.1 关于本文档

本手册是为 Sarge 版的 debian-installer 创建，基于 Woody 版的软盘安装手册，而那份手册又是基于更早版本的 Debian 安装手册，并且本手册也基于 2003 年 GPL 协议下的 Progeny 版本的手册。

本文档以 DocBook XML 写成。输出格式由很多程序使用来自 `docbook-xml` 和 `docbook-xsl` 软件包中的信息而生成。

为了提升本文档的可维护性，我们使用了一些 XML 的特性，比如 entities 和 profiling 属性。它们充当类似于编程语言中变量与条件的角色。本文档的 XML 源文件包含了每种不同的硬件架构信息—profiling 属性用于分开文档的某个特定架构的位。

Debian 安装手册简体中文版翻译项目由 Carlos Z.F. Liu 发起组织，参加翻译和校对的人员有 (按字母和拼音排列)：Carlos Liu、Huaxia Zhao、Kov Tchai、Ming Hua、szjungle、wide288、胡海星、吴鲁加、解彦博、肖盛文。在此表示感谢！另外，附录中的 GPL 简体中文版取自：http://gnu.freehostingguru.com/gpl/。

E.2 为本文档做出贡献

如果您有关于本文档的问题或者建议，恐怕应该将其提交作为软件包 `installation-guide` 的 bug 报告。请参见 `reportbug` 软件包或者阅读 Debian Bug Tracking System （Debian Bug 跟踪系统）的在线文档。良好的习惯是先，如果您能检查 installer-manual 处于 open 状态的 bug，看看您的问题是否已经被其他人报告，这将会很好。如果有，您可以将确认或者帮助信息提供给 XXXX@bugs.debian.org，其中 XXXX 是已报告的 bug 编号。

更好的是，取得一份本文档的 DocBook 源文件的复件，给它打一个补丁。DocBook 的源文件可以在 installation-guide project on salsa找到。如果您不熟悉 DocBook，不用担心：在 manuals （手册）目录下有一个简单的 cheatsheet 文件，可以帮您起步。它像 html 文件，但更侧重于文本的含义而不是显示。欢迎将补丁提交给 debian-boot 邮件列表（请参见下面）。关于如何从 git 检出（check out）源文件的指引，请参见源文件根目录中的 README。

请勿直接联系本文档的作者。还有一个关于 `debian-installer` 的邮件列表，包含本手册的讨论。这个邮件列表是 debian-boot@lists.debian.org。订阅此邮件列表的指引可以在 Debian Mailing List Subscription （Debian 邮件列表订阅）页面找到；或者您也可以在线浏览 Debian Mailing List Archives （Debian 邮件列表档案）。

E.3 主要贡献

本文档最初由 Bruce Perens、Sven Rudolph、Igor Grobman、James Treacy 和 Adam Di Carlo 撰写。Sebastian Ley 撰写了 Installation Howto（要安装该怎么做）。

Miroslav Kuře 为 Sarge 版 debian-installer 里面的很多新功能编写了文档。Frans Pop 曾经是 Etch、Lenny 和 Squeeze 发布版的主要编辑和版本经理。

有许许多多的 Debian 用户和开发人员为本文档做出了贡献。特别要感谢 Michael Schmitz（m68k 支持）、Frank Neumann（Amiga 安装手册的最初作者）、Arto Astala、Eric Delaunay/Ben Collins（SPARC 信息）、Tapio Lehtonen 和 Stéphane Bortzmeyer 做了大量的编辑和文字处理。我们还要感谢 Pascal Le Bail 提供了从 U 盘上引导的有用信息。

非常有帮助的文本和信息可以在 Jim Mintha’s 的网络启动 HOWTO（没有有效的 URL）、Debian FAQ、Linux/m68k FAQ、Linux for SPARC Processors FAQ、Linux/Alpha FAQ，以及其它中找到。必须要承认这些免费可用且资源丰富的信息的维护者们。

本手册中有关 chroot 安装的章节（第 D.3 节）部分来源于 Karsten M. Self 所有版权的文档。

E.4 商标确认

所有商标的所有权归各自的商标所有者。

Appendix F

GNU 通用公共授权

注意

This is an unofficial translation of the GNU General Public License into Chinese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL —only the original English text of the GNU GPL does that. However, we hope that this translation will help Chinese speakers to better understand the GNU GPL.

这是一份 GNU 通用公共授权非正式的中文翻译。它并非由自由软件基金会所发表，亦非使用 GNU 通用公共授权的软件的法定发布条款—只有 GNU 通用公共授权英文原文的版本始具有此等效力。然而，我们希望这份翻译能帮助中文的使用者更了解 GNU 通用公共授权。

第二版, 1991 年 6 月

```
b'' 著 b''b'' 作 b''b'' 权 b''b'' 所 b''b'' 有 b'' (C) 1989b'',
    b''1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.

b'' 允 b''b'' 许 b''b'' 每 b''b'' 个 b''b'' 人 b''b'' 复 b''b'' 制 b''b'' 和 b''b'' 发
    b''b'' 布 b''b'' 本 b''b'' 授 b''b'' 权 b''b'' 文 b''b'' 件 b''b'' 的 b''b'' 完
    b''b'' 整 b''b'' 副 b''b'' 本 b''b'', b''
b'' 但 b''b'' 不 b''b'' 允 b''b'' 许 b''b'' 对 b''b'' 它 b''b'' 进 b''b'' 行 b''b'' 任
    b''b'' 何 b''b'' 修 b''b'' 改 b''b''。b''
```

F.1 导言

大多数软件授权声明是设计用以剥夺您共享与修改软件的自由。相反地，GNU 通用公共授权力图保证您分享与修改自由软件的自由-确保软件对所有的使用者都是自由的。通用公共授权适用于大多数自由软件基金会的软件，以及任何作者指定使用本授权的其他软件。(有些自由软件基金会的软件，则适用 GNU 函式库通用公共授权规定。) 您也可以让您的软件适用本授权规定。

当我们在谈论自由软件时，我们所指的是自由，而不是价格。我们的通用公共授权是设计用以确保使您拥有发布自由软件备份的自由 (以及您可以决定此一服务是否收费)，确保您能收到源码或者在您需要时能得到它，确保您能变更软件或将它的一部分用于新的自由软件；并且确保您知道您可以做上述的这些事情。

为了保障您的权利，我们需要作出限制：禁止任何人否认您上述的权利，或者要求您放弃这些权利。如果您发布软件的副本，或者对之加以修改，这些限制就转化成为您的责任。

例如，假如您发布此类程序的副本，无论是免费或收取费用，您必须将您所享有的一切权利给予收受者。您也必须确保他们也能收到或得到原始程序码。而且您必须向他们展示这些条款的内容，使他们知到他们所享有的权利。

我们采取两项措施来保护您的权利：(1) 以著作权保护软件，以及 (2) 提供您本授权，赋与您复制、发布并且 / 或者修改软件的法律许可。

同时，为了保护作者与我们（按：指自由软件基金会），我们想要确定每个人都明白，自由软件是沒有担保责任的。如果软件被他人修改并加以传播，我们需要其收受者知道，他们所得到的并非原始版本，因此由他人所引出的任何问题对原作者的声誉将不会有任何的影响。

最后，所有自由软件不断地受到软件专利的威胁。我们希望能避免自由软件的再发布者以个人名义取得专利授权而使程序专有化的风险。为了防止上述的情事发生，我们在此明确声明：任何专利都必须为了每个人的自由使用而核准，否则就不应授与专利。

以下是有关复制、发布及修改的明确条款及条件。

F.2 GNU 通用公共授权

复制、发布与修改的条款与条件

0. 凡著作权人在其程序或其他著作中声明，该程序或著作会在通用公共授权条款下发布，本授权对其均有适用。以下所称的“程序”，是指任何一种适用通用公共授权的程序或著作；并且一个“基于本程序的著作”，则指本程序或任何基于著作权法所产生的衍生著作，换言之，是指包含本程序全部或一部的著作，不论是完整的或经过修改的程序，以及（或）翻译成其他语言的程序（以下“修改”一词包括但不限于翻译行为在內）。被授权人则称为“您”。

本授权不适用于复制、发布与修改以外的行为；这些行为不在本授权范围内。执行本程序的行为并不受限制，而本程序的输出只有在其內容构成基于本程序所生的著作（而非只是因为执行本程序所造成）时，始受本授权拘束。至于程序的输出內容是否构成本程序的衍生著作，则取决于本程序的具体用途。

1. 您可以对所收受的本程序源代码，无论以何种媒介，复制与发布其完整的复制物，然而您必须符合以下要件：以显著及适当的方式在每一份复制物上发布适当的著作权标示及无担保声明；维持所有有关本授权以及无担保声明的原貌；并将本授权的副本连同本程序一起交付予其他任何一位本程序的收受者。

您可以对让与复制物的实际行为收取一定的费用，您也可以自由决定是否提供担保以作为对价的交换。

2. 您可以修改本程序的一个或数个复制物或者本程序的任何部份，以此形成基于本程序所生的著作，并依前述第一条规定，复制与发布此一修改过的程序或著作，但您必须符合以下要件：

a. 您必须在所修改的挡案上附加显著的标示，阐明您修改过这些挡案，以及修改日期。

b. 您必须就您所发布或发行的著作，无论是包含本程序全部或一部的著作，或者是自本程序或其任何部份所衍生的著作，整体授权所有第三人依本授权规定使用，且不得因此项授权行为而收取任何费用。

c. 若经过修改的程序在执行时通常以交互方式读取命令时，您必须在最常被使用的方式下，于开始进入这种交互式使用时，列印或展示以下宣告：适当的著作权标示及无担保声明（或者声明您提供担保）、使用者可以依这些条件再发布此程序，以及告知使用者如何浏览本授权的副本。（例外：若本程序本身是以交互的方式执行，然而通常却不会列印该宣告时，则您基于本程序所生的著作便无需列印该宣告。）

这些要求对修改过的著作是整体适用的。倘著作中可识别的一部份并非衍生自本程序，并且可以合理地认为是一独立的、个别的著作，则当您将其作为个别著作加以发布时，本授权及其条款将不适用于该部分。然而当您将上述部分，作为基于本程序所生著作的一部而发布时，整个著作的发布必须符合本授权条款的规定，而本授权对于其他被授权人所为的许可及于著作整体。

因此，本条规定的意图不在于主张或剥夺您对于完全由您所完成著作的权利；应该說，本条规定意在行使对基于程序所生的之衍生著作或集合著作发布行为的控制权。

此外，非基于本程序所生的其他著作与本程序（或基于本程序所生的著作）在同一储存或发布的媒介上的单纯聚集行为，并不会使该著作因此受本授权条款约束。

3. 您可以依前述第一、二条规定，复制与发布本程序（或第二条所述基于本程序所产生的著作）的目的码或可执行形式，但您必须符合以下要件：

a. 附上完整、相对应的机器可判读源码，而这些源码必须依前述第一、二条规定在经常用以作为软件交换的媒介物上发布；或

b. 附上至少三年有效的书面报价文件，提供任何第三人在支付不超过实际发布源码所需成本的费用下，取得相同源码的完整机器可读复制物，并依前述第一、二条规定在经常用以作为软件交换的媒介物上发布该复制物；或

c. 附上您所收受有关发布相同源码的报价资讯。（本项选择仅在非赢利发布、且仅在您依前述 b 项方式自该书面报价文件收受程序目的码或可执行形式时，始有适用。）

著作的源码，是指对著作进行修改时适用的形式。对于一个可执行的著作而言，完整的源码是指著作中所包含所有模组的全部源码，加上相关介面的定义挡，还加上用以控制该著作编译与安装的描述。然而，特别的例外情况是，所发布的源码并不需包含任何通常会随著所执行作业系统的主要组成部分（编译器、核心等等）而发布的软件（无论以源码或二进位格式），除非该部分本身即附加在可执行程序中。

若可执行码或目的码的发布方式，是以指定的地点提供存取位置供人复制，则提供可自相同地点复制源码的使用机会，视同对于源码的发布，然而第三人并不因此而负有将目的码连同源码一起复制的义务。

4. 除本授权所明示的方式外，您不得对本程序加以复制、修改、再授权或发布。任何试图以其他方式进行复制、修改、再授权或者发布本程序的行为均为无效，并且将自动终止您基于本授权所得享有的权利。然而，依本授权规定自您手中收受复制物或权利之人，只要遵守本授权规定，他们所获得的授权并不会因此终止。

5. 因为您并未在本授权上签名，所以您无须接受本授权。然而，除此之外您别无其他修改或发布本程序或其衍生著作的授权许可。若您不接受本授权，则这些行为在法律上都是被禁止的。因此，藉由对本程序（或任何基于本程序所生的著作）的修改或发布行为，您表示了对于本授权的接受，以及接受所有关于复制、发布或修改本程序或基于本程序所生著作的条款与条件。

6. 每当您再发布本程序（或任何基于本程序所生的著作）时，收受者即自动获得原授权人所授予依本授权条款与条件复制、发布或修改本程序的权利。您不得就本授权所赋予收受者行使的权利附加任何进一步的限制。您对于第三人是否履行本授权一事，无须负责。

7. 若法院判决、专利侵权主张或者其他任何理由（不限于专利争议）的结果，使得加诸于您的条件（无论是由法院命令、协议书或其他方式造成）与本授权规定有所冲突，他们并不免除您对于本授权规定的遵守。若您无法同时符合依本授权所生义务及其他相关义务而进行发布，那么其结果便是您不得发布该程序。例如，若专利授权不允许其他人直接或间接取得复制物，通过您以免付权利金的方式再发布该程序，您唯一能同时满足该义务及本授权的方式就是徹底避免进行该程序的发布。

若本条任一部份在特殊情况下被认定无效或无法执行时，本条其余部分仍应适用，且本条全部于其他情况下仍应适用。

本条的目的并不在诱使您侵害专利或其他财产权的权利主张，或就此类主张的有效性加以争执；本条的唯一目的，是在保障藉由公共授权惯例所执行自由软件发布系统的完整性。许多人信赖该系统一贯使用的应用程序，而对经由此系统发布的大量软件有相当多的贡献；作者／贡献者有权决定他或她是否希望经由其他的系统发布软件，而被授权人则无该种选择权。

本条的用意在于将本授权其他不确定的部分徹底解释清楚。

8. 若因为专利或享有著作权保护的介面问题，而使得本程序的发布与／或使用局限于某些国家时，则将本程序置于本授权规范之下的原著作权人得增列明确的发布地区限制条款，将这些国家排除在外，而使发布的许可只限在未受排除的国家之內或之中。在该等情况下，该限制条款如同以书面方式订定于本授权內容中，而成为本授权的条款。

9. 自由软件基金会得随时发表通用公共授权的修正版与／或新版本。新版本在精神上将近似于目前的版本，然而在细节上或所不同以因应新的问题或状况。每一个版本都有个别的版本号码。若本程序指定有授权版本号码，表示其适用该版本或是“任何新版本”时，您可以选择遵循该版本或任何由自由软件基金会日后所发表新版本的条款与条件。若本程序并未指定授权版本号码时，您可以选择任一自由软件基金会所发表的版本。

10. 若您想将部分本程序纳入其他自由程序，而其发布的条件有所不同时，请写信取得作者的许可。若为自由软件基金会享有著作权的软件，请写信至自由软件基金会；我们有时会以例外方式予以处理。我们的决定取决于两项目标：确保我们自由软件的所有衍生著作均维持在自由的状态，并广泛地促进软件的分享与再利用。

无担保声明

11. 由于本程序是无偿授权，因此在法律许可范围內，本授权对本程序并不负担保责任。非经书面声明，著作权人与／或其他提供程序之人，无论明示或默许，均是依「现况」提供本程序而并无任何形式的担保责任，其包括但不限于，就适售性以及特定目的的适用性为默示性担保。有关本程序品质与效能的全部风险均由您承担。如本程序被证明有瑕疵，您应承担所有服务、修复或改正的费用。

12. 非经法律要求或书面同意，任何著作权人或任何可能依前述方式修改与／或发布本程序者，对于您因为使用或不能使用本程序所造成的一般性、特殊性、意外性或间接性损失，不负任何责任（包括但不限于，资料损失，资料执行不精确，或应由您或第三人承担的损失，或本程序无法与其他程序运作等），即便前述的著作权人或其他人已被告知该等损失的可能性时，也是一样。

条文结束

F.3 您的新程序该如何采用这些条款

如果您开发了一个新程序，并且希望能够让它尽可能地被大众使用，达成此目的的最好方式就是让它成为自由软件，任何人依这些条款规定都能就该软件再为发布及修改。

为了做到这一点，请将以下声明附加到程序上。最安全的作法，是将声明放在每份源码挡案的起始处，以有效传达无担保责任的讯息；且每份挡案至少应有“版权”行以及本份声明全文位置的提示。

```
<i>b'' 用 b''b'' 一 b''b'' 行 b''b'' 描 b''b'' 述 b''b'' 程 b''b'' 序 b''b'' 的 b''b''
    名 b''b'' 称 b''b'' 与 b''b'' 其 b''b'' 用 b''b'' 途 b''b'' 简 b''b'' 述 b''b''。
    b''</i>
b'' 著 b''b'' 作 b''b'' 权 b''b'' 所 b''b'' 有 b''(C) <i>b'' 年 b''b'' 份 b''  b'' 作
    b''b'' 者 b''b'' 姓 b''b'' 名 b''</i>

b'' 本 b''b'' 程 b''b'' 序 b''b'' 为 b''b'' 自 b''b'' 由 b''b'' 软 b''b'' 件 b''b'';
    b''b'' 您 b''b'' 可 b''b'' 依 b''b'' 据 b''b'' 自 b''b'' 由 b''b'' 软 b''b'' 件
    b''b'' 基 b''b'' 金 b''b'' 会 b''b'' 所 b''b'' 发 b''b'' 表 b''b'' 的 b'' GNU b'' 通
    b''b'' 用 b''b'' 公 b''b'' 共 b''b'' 授 b''b'' 权 b''b'' 条 b''b'' 款 b''b'' 规
    b''b'' 定 b''b'',b''b'' 就 b''b'' 本 b''b'' 程 b''b'' 序 b''b'' 再 b''b'' 为 b''b''
    发 b''b'' 布 b''b'' 与 b''b'' / b''b'' 或 b''b'' 修 b''b'' 改 b''b'';b''b'' 无
    b''b'' 论 b''b'' 您 b''b'' 依 b''b'' 据 b''b'' 的 b''b'' 是 b''b'' 本 b''b'' 授
    b''b'' 权 b''b'' 的 b''b'' 第 b''b'' 二 b''b'' 版 b''b'' 或 b''b''(b''b'' 您 b''b''
    自 b''b'' 行 b''b'' 选 b''b'' 择 b''b'' 的 b''b'')b''b'' 任 b''b'' 一 b''b'' 日
    b''b'' 后 b''b'' 发 b''b'' 行 b''b'' 的 b''b'' 版 b''b'' 本 b''b''。b''

b'' 本 b''b'' 程 b''b'' 序 b''b'' 是 b''b'' 基 b''b'' 于 b''b'' 使 b''b'' 用 b''b'' 目
    b''b'' 的 b''b'' 而 b''b'' 加 b''b'' 以 b''b'' 发 b''b'' 布 b''b'',b''b'' 然 b''b''
    而 b''b'' 不 b''b'' 负 b''b'' 任 b''b'' 何 b''b'' 担 b''b'' 保 b''b'' 责 b''b'' 任,
    b''b'';b''b'' 亦 b''b'' 无 b''b'' 对 b''b'' 适 b''b'' 售 b''b'' 性 b''b'' 或 b''b''
    特 b''b'' 定 b''b'' 目 b''b'' 的 b''b'' 适 b''b'' 用 b''b'' 性 b''b'' 所 b''b'' 为
    b''b'' 的 b''b'' 默 b''b'' 示 b''b'' 性 b''b'' 担 b''b'' 保 b''b''。b''b'' 详 b''b''
    情 b''b'' 请 b''b'' 参 b''b'' 照 b'' GNU b'' 通 b''b'' 用 b''b'' 公 b''b'' 共 b''b''
    授 b''b'' 权 b''b''。b''

b'' 您 b''b'' 应 b''b'' 已 b''b'' 收 b''b'' 到 b''b'' 附 b''b'' 随 b''b'' 于 b''b'' 本
    b''b'' 程 b''b'' 序 b''b'' 的 b'' GNU b'' 通 b''b'' 用 b''b'' 公 b''b'' 共 b''b'' 授
    b''b'' 权 b''b'' 的 b''b'' 副 b''b'' 本 b''b'';b''b'' 如 b''b'' 果 b''b'' 没 b''b''
    有 b''b'',b''b'' 请 b''b'' 写 b''b'' 信 b''b'' 至 b''b'' 自 b''b'' 由 b''b'' 软
    b''b'' 件 b''b'' 基 b''b'' 金 b''b'' 会 b''b'':
    b''51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USAb''。b''
```

同时附上如何以电子及书面信件与您联系的资料。

若程序是以交互方式运作时，请在交互式模式开始时，输出简短提示如下：

```
Gnomovision b'' 第 b''69b'' 版 b''b'',b''b'' 著 b''b'' 作 b''b'' 权 b''b'' 所 b''b''
    有 b'' (C) <i>b'' 年 b''b'' 份 b'' b'' 作 b''b'' 者 b''b'' 姓 b''b'' 名 b''</i>
Gnomovisionb'' 不 b''b'' 负 b''b'' 担 b''b'' 保 b''b'' 责 b''b'' 任 b''b'',b''b'' 欲
    b''b'' 知 b''b'' 详 b''b'' 情 b''b'' 请 b''b'' 键 b''b'' 入 b'' <b>show w</b>b''。
    b''b'' 这 b''b'' 是 b''b'' 一 b''b'' 个 b''b'' 自 b''b'' 由 b''b'' 软 b''b'' 件
    b''b'',b''b'' 欢 b''b'' 迎 b''b'' 您 b''b'' 在 b''b'' 特 b''b'' 定 b''b'' 条 b''b''
    件 b''b'' 下 b''b'' 再 b''b'' 发 b''b'' 布 b''b'' 本 b''b'' 程 b''b'' 序 b''b'';
    b''b'' 欲 b''b'' 知 b''b'' 详 b''b'' 情 b''b'' 请 b''b'' 键 b''b'' 入
    b'' <b>show c</b>b''。b''
```

所假设的指令“show w”与“show c”应显示通用公共授权的相对应条款。当然，您可以使用“show w”与“show c”以外的指令名称；甚至以鼠标键击或选菜单方式进行-只要是合于您程序需要的方式都可以。

如有需要，您亦应取得您的雇主（若您的工作为程序设计师）或学校就本程序所签署的“著作权放弃承諾书”。其范例如下，您只要修改姓名即可：

```
Yoyodyne b'' 公 b''b'' 司 b''b'',b''b'' 兹 b''b'' 此 b''b'' 放 b''b'' 弃
    b'' James Hacker b'' 所 b''b'' 写 b''b'' 之 b'' <b>Gnomovision</b>b'' 程 b''b'' 序
    b''b''(b''b'' 该 b''b'' 程 b''b'' 序 b''b'' 产 b''b'' 出 b''b'' 编 b''b'' 译 b''b''
    器 b''b'' 所 b''b'' 需 b''b'' 资 b''b'' 讯 b''b'')b''b'' 所 b''b'' 有 b''b'' 的
    b''b'' 著 b''b'' 作 b''b'' 权 b''b'' 利 b''b'' 益 b''b''。b''

<i>Ty Coon b'' 公 b''b'' 司 b''b'' 签 b''b'' 章 b''</i>, 1989 b'' 年 b''b'' 四 b''b''
    月 b''b'' 一 b''b'' 日 b''
Ty Coon b'' 公 b''b'' 司 b''b'',b''b'' 副 b''b'' 总 b''b'' 裁 b''
```

本通用公共授权并不允许您将本程序并入专有程序中。若您的程序是一子程序函数库时，您可能认为允许专有应用程序与该函式库相连结会更有帮助。若这是您所想做的，请使用 GNU 函式库通用公共授权代替本授权。