前言

长话短说吧，此翻译是经过一点一点地积累而成，因为有了参与人员的无私贡献与坚持，才能有今天的第一版公布。当然，第一版可能是不完善的，如果大家觉得此翻译文档有错误或需要修改之处，可以登陆到“有道云协作”里的协作群去作修改更正。地址 http://163.fm/ZQcbXLa 或通过查找群号 4599204 加入。如果改动足够的多，我会重新制作并重新发布。
至于为何要翻译此文档，一是因为本人觉得如此好的软件在经历了这么久的应用与发展后，却还不为国内技术人员所普遍认识与使用，以至于当本人去学习使用和了解 GRUB 时困难重重，于是就萌生出为何不整理好所需资料，编排成一本书，方便一次性地学习与了解 GRUB，而无需从众多的网上资料里苦苦寻觅，重复资料的阅读，因此萌生了去翻译官方文档的念头。二则因为 GRUB 的功能很强大，就本人而言，GRUB 已经能轻易地解决启动上的绝大部分问题。能够应付传统的 MBR 和最新的 EFI 启动，甚至支持 ISO 等镜像文件的虚拟加载实现 PE 或 linux 的镜像化启动，更加能够通过网络的 TFTP 启动从而实现网络方式的无盘启动。
所以说，GRUB 其实已经很广泛地应用在 IT 领域却少有人知。
因此，本人提议了一个 FREE 形式的翻译项目，让有兴趣的人都参与进来把官方文档翻译好，好让后来人方便使用与学习。最后得到以下会员的贡献，终于把此项目完成收尾，公布于众，让大家按照 FREE 理念从而自由使用。

以下为感谢人员名单：
01.Introduction to GRUB.txt
小马 120 (有道云协助会员)
SunAigo (有道云协助会员)

02.Naming convention.txt
Wraith001 (有道云协助会员)

03.Installation.txt
Wraith001 (有道云协助会员)
羚羊 006 (有道云协助会员)
小饼仙子 (ubuntu 论坛会员)

07.Booting GRUB from the network.txt
apaik (有道云协助会员)
officenotepad (有道云协助会员)

08.Using GRUB via a serial line.txt
小马 120 (有道云协助会员)

10.GRUB image files.txt
小马 120 (有道云协助会员)

11.Filesystem syntax and semantics.txt
小马 120 (有道云协助会员)
SunAigo (有道云协助会员)

13.GRUB environment variables.txt
SunAigo (有道云协助会员)

15.字符集-Charset.txt
SunAigo (有道云协助会员)

16.文件系统-Filesystems.txt
SunAigo (有道云协助会员)

欢迎点击这里的链接进入精彩的Linux公社网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

Linux公社是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：**Ubuntu 专题** **Fedora 专题** **Android 专题** **Oracle 专题** **Hadoop 专题** **RedHat 专题** **SUSE 专题** **红旗 Linux 专题** **CentOS 专题**

Linux 公社微信公众号：linuxidc_com

17.输出终端-Output terminal.txt
SunAigo (有道云协助会员)

18.输入终端-Input terminal.txt
SunAigo (有道云协助会员)

19.文本语言-Gettext.txt
SunAigo (有道云协助会员)

20.正则表达式-Regexp.txt
SunAigo (有道云协助会员)

21.其他-Other.txt
SunAigo (有道云协助会员)

22.身份验证与授权-Authentication and authorisation.txt
SunAigo (有道云协助会员)

23.平台限制-Platform limitations.txt
SunAigo (有道云协助会员)

24.概要-Outline.txt
SunAigo (有道云协助会员)

25.支持此引导的明细-Supported boot targets.txt
SunAigo (有道云协助会员)

最后需要特别再一次对会员 SunAigo 的感谢，以下为其公开资料：
姓名：SunAigo
性别：男
出生年月：94.8
学历：大学本科-计算机
电子邮箱：sunyangjian@hotmail.com
qq:872668076
因本次翻译与校验都是自由公开的方式去运作的，大家都是用彼此的业余时间去做，因此在如此大量的翻译与校验里，SunAigo 的贡献非常巨大，不单独自完成了大部分的翻译，甚至包办了校验工作，直至坚持到项目的完成。坦白说，没有他的帮助和贡献，此项目很难完成收尾，所以功不可没，特此感谢。
^_^而本人网名为"kin3z”，功劳不高，仅留此名。

01.介绍 GRUB-Installation to GRUB

****

1.1 Overview

1.1 概述

==========

Briefly, a "boot loader" is the first software program that runs when a computer starts. It is responsible for loading and transferring control to an operating system "kernel" software (such as Linux or GNU Mach).The kernel, in turn, initializes the rest of the operating system (e.g.a GNU system).

简单地说，一个"启动加载器"是电脑启动时所运行的第一个软件程序，它是负责加载读入与转移控制权给操作系统"内核"的软件(例如 Linux 或 GNU)。这个内核，依次转向，初始化这个操作系统的余下部分。

GNU GRUB is a very powerful boot loader, which can load a wide variety of free operating systems, as well as proprietary operating systems with chain-loading(1) (*note Overview-Footnote-1::). GRUB is designed to address the complexity of booting a personal computer; both the program and this manual are tightly bound to that computer platform,although porting to other platforms may be addressed in the future.

GNU GRUB 是非常强大的启动载入器，它能够加载品种繁多的免费(free)操作系统，同样地也能通过链式加载器而能够加载那些有版权保护的操作系统。GRUB 旨在解决引导个人电脑的复杂性；这个程序和本手册是紧密地绑定到计算机平台上的，虽然移植到另一些平台或许会在不久的将来有望解决。

One of the important features in GRUB is flexibility; GRUB understands filesystems and kernel executable formats, so you can load an arbitrary operating system the way you like, without recording the physical position of your kernel on the disk. Thus you can load the kernel just by specifying its file name and the drive and partition where the kernel resides.

GRUB 其中一个重大的特点是灵活性；GRUB 能够理解文件系统以及内核的可执行格式，因此你能够加载任何的操作系统而随你喜欢，甚至乎不用把内核保存到你的物理硬盘里。因为你只要通过指定内核的文件名与它所存放的驱动器和分区位置就能够去被加载。

When booting with GRUB, you can use either a command-line interface (*note Command-line interface::), or a menu interface (*note Menu interface::). Using the command-line interface, you type the drive specification and file name of the kernel manually. In the menu interface, you just select an OS using the arrow keys. The menu is based on a configuration file which you prepare beforehand (*note Configuration::). While in the menu, you can switch to the command-line mode, and vice-versa. You can even edit menu entries before using them.

当 GRUB 启动后，你能使用其中的命令行界面，或一个菜单界面。当使用命令行界面时，你需要手动去指定内核所在的分区名和文件名。在菜单接口，你能从箭头选择键直接选择要启动的系统。这个菜单是基于你之前准备的那个配置文件而成。就算在该菜单中，你也可以转换到命令行模式，或反之亦然。你甚至能够在使用该菜单选项之前去编辑它。

In the following chapters, you will learn how to specify a drive, a partition, and a file name (*note Naming convention::) to GRUB, how to install GRUB on your drive (*note Installation::), and how to boot your OSes (*note Booting::), step by step.

在接下来的章节，你将学习如何在 GRUB 里去指定一个驱动器，一个分区，和一个文件名，如何去安装 GRUB 到你的硬盘，和如何去引导你的系统，一步一步来。

1.2 History of GRUB

1.2 GRUB 的历史

==========

GRUB originated in 1995 when Erich Boleyn was trying to boot the GNU Hurd with the University of Utah's Mach 4 microkernel (now known as GNU Mach). Erich and Brian Ford designed the Multiboot

Specification (*note Multiboot Specification: (multiboot)Top.), because they were determined not to add to the large number of mutually-incompatible PC boot methods.
GRUB 起源于 1995 年当时 Erich Boleyn 想尝试去引导 Utah 大学研发的使用 Mach4 微内核(现在都被称为 GNU Mach)的一个 GNU Hurd。Erich 与 Brian Ford 一起设计出了多重引导规范，因为他们决定舍弃并解决掉那些大量的，彼此互不兼容的 PC 启动方式。

Erich then began modifying the FreeBSD boot loader so that it would understand Multiboot. He soon realized that it would be a lot easier to write his own boot loader from scratch than to keep working on the FreeBSD boot loader, and so GRUB was born.
Erich 之后开始修改 FreeBSD 引导加载器让其去兼容多重启动。他很快意识到他自己需要编写一个更容易更简单的引导加载程序，而不是继续坚持工作在 FreeBSD 的引导加载器上，因此 GRUB 诞生了。

Erich added many features to GRUB, but other priorities prevented him from keeping up with the demands of its quickly-expanding user base. In 1999, Gordon Matzigkeit and Yoshinori K. Okuji adopted GRUB as an official GNU package, and opened its development by making the latest sources available via anonymous CVS. *Note Obtaining and Building GRUB::, for more information.
Erich 给 GRUB 添加了许多特性，但其他优先考虑的事使得他无法满足迅速扩大的用户群需求。到了 1999 年，Gordon Matzigkeit 和 Yoshinori K.Okuji 采用 GRUB 作为一个官方的 GNU 软件包，并通过匿名的 CVS 去开放它的源代码从而让项目去发展更新。

Over the next few years, GRUB was extended to meet many needs, but it quickly became clear that its design was not keeping up with the extensions being made to it, and we reached the point where it was very difficult to make any further changes without breaking existing features. Around 2002, Yoshinori K. Okuji started work on PUPA (Preliminary Universal Programming Architecture for GNU GRUB), aiming to rewrite the core of GRUB to make it cleaner, safer, more robust, and more powerful. PUPA was eventually renamed to GRUB 2, and the original version of GRUB was renamed to GRUB Legacy. Small amounts of maintenance continued to be done on GRUB Legacy, but the last release (0.97) was made in 2005 and at the time of writing it seems unlikely that there will be another.
在接下来的几年里，GRUB 与它的扩展也满足了许多要求，但是很快 GRUB 便显露出它本身并不很好地兼容它的扩展的设计性问题，并且我们在这个问题点上很难在不破坏现有功能的基础上进行修补。大约在 2002 年，yoshinori K.Okuji 开始了 PUPA(初步通用编程体系架构 GNU GRUB)的工作,目的为重写出更简洁的、安全的、健壮的与强大的 GRUB 内核。PUPA 最后被命名为 GRUB2，并且把原来的 GRUB 版本改名为 GRUB Legacy.少量的 GRUB Legacy 版本还保持有维护，但最新版本也只有更新到 2005 年(0.97)而且在撰写本文为止拟乎不太可能再有更新。

By around 2007, GNU/Linux distributions started to use GRUB 2 to limited extents, and by the end of 2009 multiple major distributions were installing it by default.
大约到了 2007 年，GNU/Linux 派系开始在有限范围内使用 GRUB2，并且在 2009 年年底许多发行版都在默认的情况下安装了它。

1.3 Differences from previous versions
1.3 与以前的版本的区别

==========
GRUB 2 is a rewrite of GRUB (*note History::), although it shares many characteristics with the previous version, now known as GRUB Legacy. Users of GRUB Legacy may need some guidance to find their way around this new version.
GRUB2 是 GRUB 的重写,尽管它们有许多相同的特性，但过去的版本现在已明确地被称为 GRUB Legacy.使用 GRUB Legacy 的用户可能需要一些指导才能重新掌握新版本的使用。

* The configuration file has a new name ('grub.cfg' rather than 'menu.lst' or 'grub.conf'), new syntax (*note Configuration::) and many new commands (*note Commands::). Configuration cannot be copied over directly, although most GRUB Legacy users should not find the syntax too surprising.

* GRUB2 的配置文件有了新的名字(名为'grub.cfg'而不是'menu.lst'或'grub.conf'),新的语法和一些新的命令。因此新旧版本的配置文件是不能直接彼此复制使用的，尽管许多 GRUB Legacy 的用户可能会对新版本的这些语句并不陌生。

* 'grub.cfg' is typically automatically generated by 'grub-mkconfig' (*note Simple configuration::). This makes it easier to handle versioned kernel upgrades.
* 'grub.cfg'通常是由'grub-mkconfig'自动生成的.这让内核版本的升级更容易。

* Partition numbers in GRUB device names now start at 1, not 0 (*note Naming convention::).
* GRUB 设备的分区编号现在是 1 开始，而非 0.

* The configuration file is now written in something closer to a full scripting language: variables, conditionals, and loops are available.
* GRUB2 的配置文件内容更接近于一个完整的脚本语言：变量、条件与循环等都可以在这里使用。

* A small amount of persistent storage is available across reboots, using the 'save_env' and 'load_env' commands in GRUB and the 'grub-editenv' utility. This is not available in all configurations (*note Environment block::).
* 一些小批量的变量需要在重新启动后还能够保持有它所储存的数据，则需要使用到 GRUB2 的'save_env'与'load_env'命令或使用'grub-editenv'功能。这些不是在所有的配置中都能有效。

* GRUB 2 has more reliable ways to find its own files and those of target kernels on multiple-disk systems, and has commands (*note search::) to find devices using file system labels or Universally Unique Identifiers (UUIDs).
*GRUB2 有非常可靠的算法去从多个磁盘系统中找到它自己所需要的文件以及那些目标内核，并且还可以通过命令查找文件系统标签或者全局唯标识符(UUIDs)从而指定设备.

* GRUB 2 is available for several other types of system in addition to the PC BIOS systems supported by GRUB Legacy: PC EFI, PC coreboot, PowerPC, SPARC, and MIPS Lemote Yeeloong are all supported.
* GRUB2 是可用于 GRUB Legacy 所支持的 PC BIOS 系统外还支持一些其他类型的系统：PC EFI，PC coreboot，PowerPC，SPARC，和 MIPS lemote Yeeloong(龙芯)它们都被支持。

* Many more file systems are supported, including but not limited to ext4, HFS+, and NTFS.
* 支持更多的文件系统，包括 ext4、HFS+和 NTFS 但远远不只这些。

* GRUB 2 can read files directly from LVM and RAID devices.
* GRUB2 能够直接从 LVM 和 RAID 设备里读取文件。

* A graphical terminal and a graphical menu system are available.
* 提供了一个可用的图形终端和一个图形菜单系统。

* GRUB 2's interface can be translated, including menu entry names.
* GRUB 2 的界面能够被翻译，包括菜单项目名称。

* The image files (*note Images::) that make up GRUB have been reorganised; Stage 1, Stage 1.5, and Stage 2 are no more.
* GRUB 的镜像文件组成已经重组；Stage 1 , Stage 1.5 ,和 Stage 2 它们已经被去除。

* GRUB 2 puts many facilities in dynamically loaded modules, allowing the core image to be smaller, and allowing the core image to be built in more flexible ways.
* GRUB2 通过动态载入模块来扩展功能，所以核心镜像可以很小，因此能够构建出更具灵活性的内核镜像。

1.4 GRUB features
1.4 GRUB 的特点
==========
The primary requirement for GRUB is that it be compliant with the "Multiboot Specification", which is described in *note Multiboot Specification: (multiboot)Top.
建立 GRUB 的首要要求是符合"多重引导规范"，这个在多重引导规范章节部分有所描述。
The other goals, listed in approximate order of importance, are:
其他目标，大致以重要性来整理列出，为：

* Basic functions must be straightforward for end-users.
* 对于终端用户来说，基本的功能必须简单明了。

* Rich functionality to support kernel experts and designers.
* 向内核专家与设计者提供丰富的功能。

* Backward compatibility for booting FreeBSD, NetBSD, OpenBSD, and Linux. Proprietary kernels (such as DOS, Windows NT, and OS/2) are supported via a chain-loading function.
* 向后兼容启动 FreeBSD,NetBSD,OpenBSD,以及 Linux.通过链式加载器功能从而支持专有内核(比如 DOS,Windows NT,和 OS/2)。

Except for specific compatibility modes (chain-loading and the Linux "piggyback" format), all kernels will be started in much the same state as in the Multiboot Specification. Only kernels loaded at 1 megabyte or above are presently supported. Any attempt to load below that boundary will simply result in immediate failure and an error message reporting the problem.
除非使用特定的兼容性模式(链式-加载和 Linux "piggyback 背负式装运" 格式)，否则所有的内核都将按照多重引导规范而在相同的状态下去启动。目前只支持在磁盘头区域的 1 兆字节(1MB)及其以上的空间用来加载内核 。任何去尝试把内核加载到 1MB 以下范围的空间都将直接导致失败并且产生一个此问题的错误报告信息。

In addition to the requirements above, GRUB has the following features (note that the Multiboot Specification doesn't require all the features that GRUB supports):
除了以上的条件外，GRUB 还具有以下特性(注意多重引导规范不需要 GRUB 去支持其所有的功能)：

Recognize multiple executable formats
识别多种可执行文件格式:
Support many of the "a.out" variants plus "ELF". Symbol tables are also loaded.
支持加载多种"a.out"可执行文件以及其变体并包括"ELF"。符号表也一并被加载。

Support non-Multiboot kernels
支持非多重引导的内核:
Support many of the various free 32-bit kernels that lack Multiboot compliance (primarily FreeBSD, NetBSD, OpenBSD, and Linux). Chain-loading of other boot loaders is also supported.
支持许多各种各样的缺乏多重引导依赖性的自由的 32bit-free 内核(主要是 FreeBSD,NetBSD,OpenBSD,和 Linux).另外还支持其他引导载入程序的链式加载。

Load multiples modules
加载多个模块:
Fully support the Multiboot feature of loading multiple modules.
完全支持加载多个模块的多重引导特性

Load a configuration file
加载配置文件:
Support a human-readable text configuration file with preset boot commands. You can also load another configuration file dynamically and embed a preset configuration file in a GRUB image file. The

list of commands (*note Commands::) are a superset of those supported on the command-line. An example configuration file is provided in *note Configuration.

支持一个用预设引导命令生成的可读文本配置文件。你也可以动态地载入另一个配置文件或嵌入在GRUB 镜像里的一个预置的配置文件。这些命令(注：命令)是一个命令行所支持的一个扩展集。并且在"命令"章节里提供了一个配置文件的示例.

Provide a menu interface

提供一个菜单界面:

A menu interface listing preset boot commands, with a programmable timeout, is available. There is no fixed limit on the number of boot entries, and the current implementation has space for several hundred.

一个菜单界面列表已经预置好了引导命令，此界面可以通过可编程控制的超时时间去自动激活。它没有硬性地限制那些引导条目的数目，而且目前尝试过设置过数以百计个条目的例子。

Have a flexible command-line interface

具有一个灵活的命令行界面:

A fairly flexible command-line interface, accessible from the menu, is available to edit any preset commands, or write a new boot command set from scratch. If no configuration file is present,GRUB drops to the command-line.

一个相当灵活命令行界面，可选择进入的菜单，也可以很好地去编辑任何预置了命令的菜单项目，或从头开始编写一个新的引导命令并设置为暂用。如果没有配置文件存在，GRUB 将会跳转到命令行模式。

The list of commands (*note Commands::) are a subset of those supported for configuration files. Editing commands closely resembles the Bash command-line (*note Command-line interface::), with <TAB>-completion of commands, devices, partitions, and files in a directory depending on context.

这些列出的命令为配置文件所支持的一个子集。编辑命令的方式非常相似于 Bash 命令行方式，依赖于目录上下文的方式通过<TAB>键可以自动补全命令名，设备名，分区名，和文件名。

Support multiple filesystem types

支持多种文件系统类型:

Support multiple filesystem types transparently, plus a useful explicit blocklist notation. The currently supported filesystem types are "Amiga Fast FileSystem (AFFS)", "AtheOS fs", "BeFS", "BtrFS" (including raid0, raid1, raid10, gzip and lzo), "cpio" (little- and big-endian bin, odc and newc variants), "Linux ext2/ext3/ext4", "DOS FAT12/FAT16/FAT32", "exFAT", "HFS", "HFS+", "ISO9660" (including Joliet, Rock-ridge and multi-chunk files), "JFS", "Minix fs" (versions 1, 2 and 3), "nilfs2", "NTFS" (including compression), "ReiserFS", "ROMFS", "Amiga Smart FileSystem (SFS)", "Squash4", "tar", "UDF", "BSD UFS/UFS2", "XFS", and "ZFS" (including lzjb, gzip, zle, mirror, stripe, raidz1/2/3 and encryption in AES-CCM and AES-GCM). *Note Filesystem::, for more information.

明了简洁地支持多种文件系统模式，加上一个很好用的显式数据块列表编码。当前支持的文件系统类型"Amiga 阿米加快速文件系统(AFFS)","AtheOS fs","BeFS","BtrFS"(其中包括 raid0, raid1, raid10, gzip and lzo),"cpio"(大小端的字节存储次序，ODC 和新的版本),"Linux ext2/ext3/ext4","DOS FAT12/FAT16/FAT32","exFAT","HFS","HFS+","ISO9660"(包括 joliet 标准，Rock-ridge 和 multi-chunk 文件),"JFS","Minix fs"(版本 1,2 和 3),"nilfs2","NTFS"(包括压缩),"ReiserFS","阿米加智能文件系统(Amiga Smart FileSystem-SFS)","Squash4","tar","UDF","BSD UFS/UFS2","XFS",和"ZFS"(包括 lzjb,gzip,zle,mirror,stripe,raidz1/2/3 和加密的 AES-CCM 与 AES-GCM)。(注：文件系统-从中得到更多信息)

Support automatic decompression

支持自动解压缩:

Can decompress files which were compressed by 'gzip' or 'xz'(1) (*note Features-Footnote-1::). This function is both automatic and transparent to the user (i.e. all functions operate upon the uncompressed contents of the specified files). This greatly reduces a file size and loading time, a particularly great benefit for floppies.(2) (*note Features-Footnote-2::)

能从压缩为"gzip"或"xz"的文件中解压出文件。对于用户这个功能是自动的且显式的(也就是所有功能都像在未压缩内容的所指定文件上操作)。这大大地降低了文件大小和加载时间，对于磁盘来说这是极其的好。

It is conceivable that some kernel modules should be loaded in a compressed state, so a different module-loading command can be specified to avoid uncompressing the modules.

可以想像某些内核模块在加载时必须还处于压缩状态，因此一个不同的模块加载需要用命令去指定它从而避免去解压缩这个模块。

Access data on any installed device

在任何安装设备里访问数据：

Support reading data from any or all floppies or hard disk(s) recognized by the BIOS, independent of the setting of the root device.

支持被 BIOS 认出的所有任意的磁盘或软盘里读取数据，这不受限于 root 设备的设置。

Be independent of drive geometry translations

独立的驱动器位置排列转换：

Unlike many other boot loaders, GRUB makes the particular drive translation irrelevant. A drive installed and running with one translation may be converted to another translation without any adverse effects or changes in GRUB's configuration.

不同于很多其他的引导加载程序，GRUB 使得特定的驱动器之间的位置转换变得无关紧要。一个驱动器安装并以一种位置排列形式运行后可以被转换成另一种位置排列形式而不会对 GRUB 的配置产生不良的影响。

Detect all installed RAM

检测出所有安装了的 **RAM**：

GRUB can generally find all the installed RAM on a PC-compatible machine. It uses an advanced BIOS query technique for finding all memory regions. As described on the Multiboot Specification (*note Multiboot Specification: (multiboot)Top.), not all kernels make use of this information, but GRUB provides it for those who do.

GRUB 一般都能找出所有安装在 PC-兼容机机器里的 RAM。它使用一个高级的 BIOS 查询技术去找出所有的内存区域。这正如多重引导规范里所描述的，并不是所有内核都遵从多重引导规范里的内容，但 GRUB 提供对那些信息内容的支持从而实现支持那些内核。

Support Logical Block Address mode

支持逻辑块地址模式：

In traditional disk calls (called "CHS mode"), there is a geometry translation problem, that is, the BIOS cannot access over 1024 cylinders, so the accessible space is limited to at least 508 MB and to at most 8GB. GRUB can't universally solve this problem, as there is no standard interface used in all machines. However, several newer machines have the new interface, Logical Block Address ("LBA") mode. GRUB automatically detects if LBA mode is available and uses it if available. In LBA mode, GRUB can access the entire disk.

在传统的磁盘调用中(调用为"CHS 模式"),这里有一个物理几何寻址的转换问题，也就是说，BISO 不能访问超过 1024 个硬盘柱面(视乎硬盘逻辑参数中的磁头数)，所以可理解为可访问的空间最少为 508MB(16 个磁头数)与最大为 8GB(256 个磁头数)。GRUB 不能统一地处理这个问题，因为没有一个标准接口能够适用于所有的机器设备。不过，某些新的机器设备拥有了一些新的接口，例如逻辑单元地址("LBA")模式。如果 LBA 模式是可用的则 GRUB 会自动检测出并启用它。在 LBA 模式里，GRUB 可以访问整个硬盘。

Support network booting

支持网络引导：

GRUB is basically a disk-based boot loader but also has network support. You can load OS images from a network by using the "TFTP" protocol.

GRUB 主要是一个基于硬盘引导的加载程序但也可以支持网络。你可以通过使用"TFTP"协议来加载源自网络的 OS 镜像。

Support remote terminals

支持远程终端：

To support computers with no console, GRUB provides remote terminal support, so that you can control GRUB from a remote host. Only serial terminal support is implemented at the moment.

为了支持没有控制台(console)的电脑，GRUB 提供远程终端的支持，因此你能通过一台主机来远程控制 GRUB。目前只能实现的仅仅只有串口终端接口。

(1) Only CRC32 data integrity check is supported (xz default is CRC64 so one should use -check=crc32 option). LZMA BCJ filters are supported.

(1) 只支持 CRC32 的数据完整性检查(xz 默认为 CRC64 因此所有人都应该使用 -check=crc32 选项)。支持 LZMA BCJ 过滤器。

(2) There are a few pathological cases where loading a very badly organized ELF kernel might take longer, but in practice this never happen.

(2) 在一些少数的问题案例里载入一个组织得非常糟糕的 ELF 内核需要更长的时间，但实际上这个不会发生。

1.5 The role of a boot loader

1.5 引导载入器的角色

==========

The following is a quotation from Gordon Matzigkeit, a GRUB fanatic:

这里以下论述是引用自 Gordon Matzigkeit，一个 GRUB 狂热者：

Some people like to acknowledge both the operating system and kernel when they talk about their computers, so they might say they use "GNU/Linux" or "GNU/Hurd". Other people seem to think that the kernel is the most important part of the system, so they like to call their GNU operating systems "Linux systems."

有些人当谈及他们计算机里的操作系统与内核时偏向于说这两者是分开的，于是他们可能会说他们用的是"GNU/Linux"或"GNU/Hurd"。另一些人似乎认为内核是系统里最重要的组成部分，所以他们喜欢称他们的 GNU 操作系统为"Linux 系统"。

I, personally, believe that this is a grave injustice, because the_boot loader_ is the most important software of all. I used to refer to the above systems as either "LILO"(1) (*note Role of a boot loader-Footnote-1::) or "GRUB" systems.

我，就我而言，相信这是极为不公正的，因为引导加载器是所有软件中最重要的。我习惯把上述的系统称为"LILO"或"GRUB"系统。

Unfortunately, nobody ever understood what I was talking about; now I just use the word "GNU" as a pseudonym for GRUB.

不幸地，没有人明白我在说什么；现在我只是用"GNU"作为我在 GRUB 群里的一个别名。

So, if you ever hear people talking about their alleged "GNU"systems, remember that they are actually paying homage to the best boot loader around... GRUB!

因此，如果你曾经听到人们谈论他们的所谓"GNU"系统，记住他们实际上是向最好的引导加载器表示敬意...GRUB!

We, the GRUB maintainers, do not (usually) encourage Gordon's level of fanaticism, but it helps to remember that boot loaders deserve recognition. We hope that you enjoy using GNU GRUB as much as we did writing it.

到此，GRUB 的维护者们，我们(通常)不鼓励像 Gordon's 这样的狂热程度，但它有助于你记住引导加载器应该有的认可。我们希望你喜欢使用 GNU GRUB 并且跟我们一样熟练使用。

(1) The LInux LOader, a boot loader that everybody uses, but nobody likes.

(1) 这个 Linux 加载器，每个人都要使用的一个引导加载器，但没有人喜欢它。

02.命名规则-Installation

02.命名规则-Installation
****

2 Naming convention
2 命名规则
==========
The device syntax used in GRUB is a wee bit different from what you may have seen before in your operating system(s), and you need to know it so that you can specify a drive/partition.
在 GRUB 中设备的名称与你之前在操作系统中看到的有点不同，你需要了解它才能正确地指定一个驱动器或分区。
Look at the following examples and explanations:
看下面的例子和解释:
(fd0)
First of all, GRUB requires that the device name be enclosed with '(' and ')'. The 'fd' part means that it is a floppy disk. The number '0' is the drive number, which is counted from _zero_. This expression means that GRUB will use the whole floppy disk.
首先，GRUB 要求设备名称必须写在圆括号()中，当中的"fd"代表着软盘(floppy disk),而数字"0"为驱动器的编号，由 0 开始递增。这个表达式意味着 GRUB 会使用整个软盘。

(hd0,msdos2)
Here, 'hd' means it is a hard disk drive. The first integer '0' indicates the drive number, that is, the first hard disk, the string 'msdos' indicates the partition scheme, while the second integer, '2', indicates the partition number (or the PC slice number in the BSD terminology). The partition numbers are counted from _one_, not from zero (as was the case in previous versions of GRUB). This expression means the second partition of the first hard disk drive. In this case, GRUB uses one partition of the disk, instead of the whole disk.
在这里，"hd"意味着它是一块硬盘驱动器,第一个整数"0"表明了驱动器编号，也即是，它就是第一块硬盘，而字符串'msdos'指明了分区所使用的文件系统的格式类型，第二个整数"2"表示分区编号（或者是 BSD 中所描述的切片(slices)编号）。分区号从 1 开始，而不是 0（有别于 GRUB 以前的版本）,这个表达式表示第一块硬盘驱动器上的第二个分区，在这种情况下，GRUB 所使用的是磁盘里的一个分区，而非整个磁盘.

(hd0,msdos5)
This specifies the first "extended partition" of the first hard disk drive. Note that the partition numbers for extended partitions are counted from '5', regardless of the actual number of primary partitions on your hard disk.
这里表示第一块磁盘驱动器上的第一个"扩展分区"，注意的是扩展分区的分区号是从"5"开始，从而无需考虑你硬盘上的主分区到底有多少(因最多只能有 4 个主分区)。

(hd1,msdos1,bsd1)
This means the BSD 'a' partition on first PC slice number of the second hard disk.
这个意味着 BSD 系统上第"2"块硬盘的第"1"个 slice 切片号的"a"分区(BSD 使用 slice 管理磁盘，概念类似于主分区)。
Of course, to actually access the disks or partitions with GRUB, you need to use the device specification in a command, like 'set root=(fd0)' or 'parttool (hd0,msdos3) hidden-'. To help you find out which number specifies a partition you want, the GRUB command-line (*note Command-line interface::) options have argument completion. This means that, for example, you only need to type
当然，想通过 GRUB 真实地访问到磁盘或分区，你需要使用特定命令去指定各种规格的设备，如'set root=(fd0)'或'parttool(hd0,msdos3) hidden-'。为了帮助你找到想要访问的那个分区所相应的编号，GRUB 命令行选项有参数补全的功能。也就是说，假如，你只需要输入

欢迎点击这里的链接进入精彩的Linux公社网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

Linux公社是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：**Ubuntu 专题** **Fedora 专题** **Android 专题** **Oracle 专题** **Hadoop 专题** **RedHat 专题** **SUSE 专题** **红旗 Linux 专题** **CentOS 专题**

Linux 公社微信公众号：linuxidc_com

set root=(
followed by a <TAB>, and GRUB will display the list of drives,partitions, or file names. So it should be quite easy to determine the name of your target partition,even with minimal knowledge of the syntax.
接着按一下<tab>，GRUB 就会显示出驱动器、分区或文件名的列表。因此确定目标分区的名字应该是相当容易的事，即使你对语法了解甚少。

Note that GRUB does _not_ distinguish IDE from SCSI - it simply counts the drive numbers from zero, regardless of their type. Normally,any IDE drive number is less than any SCSI drive number, although that is not true if you change the boot sequence by swapping IDE and SCSI drives in your BIOS.
请注意 GRUB 是不会区分 IDE 驱动器和 SCSI 驱动器的，它只会直接从 0 开始计算驱动器号，而无视两者的类型。通常地，IDE 驱动器编号都会小于 SCSI 驱动器编号，除非你在 BIOS 中把 IDE 和 SCSI 驱动器的启动顺序对调交换。

Now the question is, how to specify a file? Again, consider an example:
接下来的问题是，如何指定一个文件？再来，考虑以下例子：

(hd0,msdos1)/vmlinuz
This specifies the file named 'vmlinuz', found on the first partition of the first hard disk drive. Note that the argument completion works with file names, too.
这里指定了一个名为"vmlinuz"的文件，它被创建在第一个硬盘的第一个分区里。请注意参数自动补全功能对于文件名也同样有效。

That was easy, admit it. Now read the next chapter, to find out how to actually install GRUB on your drive.
必须承认以上内容都很简单。现在来阅读下一章，去了解如何真正地安装 GRUB 到硬盘上。

03.安装-Installation
********** *****

3 Installation
3 安装
==========
In order to install GRUB as your boot loader, you need to first install the GRUB system and utilities under your UNIX-like operating system (*note Obtaining and Building GRUB::). You can do this either from the source tarball, or as a package for your OS.
为了把 GRUB 安装成为你的引导加载器，首先你需要在你的类 Unix 操作系统上安装好 GRUB 系统及其工具。你既可以通过编译原码来安装 GRUB，也可以通过针对于你的操作系统所构建的 GRUB 软件包来安装。

After you have done that, you need to install the boot loader on a drive (floppy or hard disk) by using the utility 'grub-install' (*note Invoking grub-install::) on a UNIX-like OS.
安装完了 GRUB 之后，你需要在你的类 Unix 操作系统上(以 root 的权限)使用"grub-install"工具来把一个启动加载器安装到你的驱动器上(软驱或硬盘)。

GRUB comes with boot images, which are normally put in the directory '/usr/lib/grub/<cpu>-<platform>' (for BIOS-based machines '/usr/lib/grub/i386-pc'). Hereafter, the directory where GRUB images are initially placed (normally '/usr/lib/grub/<cpu>-<platform>') will be called the "image directory", and the directory where the boot loader needs to find them (usually '/boot') will be called the "boot directory".
GRUB 自带了引导镜像，通常存放在"/usr/lib/grub/<cpu>-<platform>"(例如基于 BIOS 机器的镜像为"/usr/lib/grub/i386-pc")。因此，此目录就成为了保存 GRUB 镜像的初始位置(通常为"/usr/lib/grub/<cpu>-<platform>")并称它为"镜像目录"，并且此目也需要被引导加载器(通常位于"/boot")所找到因此也被称为"引导目录"。

3.1 Installing GRUB using grub-install
3.1 使用 grub-install 安装 GRUB
==========
For information on where GRUB should be installed on PC BIOS platforms, *note BIOS installation::.
若想知道关于 GRUB 安装在 PC BIOS 平台里的更多信息，请查阅"BIOS 安装"章节。

In order to install GRUB under a UNIX-like OS (such as GNU), invoke the program 'grub-install' (*note Invoking grub-install::) as the superuser ("root").
为了把 GRUB 安装到类 Unix 系统(如 GUN)，需要以超级用户权限 root 来运行 grub-install 命令。

The usage is basically very simple. You only need to specify one argument to the program, namely, where to install the boot loader. The argument has to be either a device file (like '/dev/hda'). For example, under Linux the following will install GRUB into the MBR of the first IDE disk:
用法很简单，你只需要给程序去指定一个参数，简单来说，就是要在哪里去安装这个引导加载器。这个参数必须是一个设备文件(类似"/dev/hda")，例如，在 Linux 下使用下面的命令将把 GRUB 安装到第一块 IDE 盘的 MBR 扇区里。
```
# grub-install /dev/hda
```
Likewise, under GNU/Hurd, this has the same effect:
同样地，在 GUN/Hurd 下，下面命令有同样的效果：
```
# grub-install /dev/hd0
```

But all the above examples assume that GRUB should put images under the '/boot' directory. If you

want GRUB to put images under a directory other than '/boot', you need to specify the option '--boot-directory'. The typical usage is that you create a GRUB boot floppy with a filesystem. Here is an example:
但是以上的例子都假设 GRUB 将镜像放在/boot 目录下。如果你想 GRUB 把镜像放在其它目录下，你需要指定"--boot-directory"选项。典型的用法是创建一个带有文件系统的 GRUB 启动盘。举个例子：

```
# mke2fs /dev/fd0
# mount -t ext2 /dev/fd0 /mnt
# mkdir /mnt/boot
# grub-install --boot-directory=/mnt/boot /dev/fd0
# umount /mnt
```

Some BIOSes have a bug of exposing the first partition of a USB drive as a floppy instead of exposing the USB drive as a hard disk (they call it "USB-FDD" boot). In such cases, you need to install like this:
某些 BIOS 存在一个把 USB 设备的第一个分区视为一个软区，而不是将一个 USB 设备作为一个硬盘的 bug(这被称为"USB-FDD"引导)。在这种情况下，你需要按照如下方式安装：

```
 # losetup /dev/loop0 /dev/sdb1
 # mount /dev/loop0 /mnt/usb
 # grub-install --boot-directory=/mnt/usb/bugbios --force --allow-floppy /dev/loop0
```

This install doesn't conflict with standard install as long as they are in separate directories.
只要它们存放于不同的目录，这种安装方法不会和标准的安装方法产生冲突。

Note that 'grub-install' is actually just a shell script and the real task is done by 'grub-mkimage' and 'grub-setup'. Therefore, you may run those commands directly to install GRUB, without using 'grub-install'. Don't do that, however, unless you are very familiar with the internals of GRUB. Installing a boot loader on a running OS may be extremely dangerous.
注意"grub-install"只是一个 shell 脚本，而真正起作用的是"grub-mkimage"和 "grub-setup"命令。因此，你也可以直接运行这两个命令安装 GRUB，而不必使用 "grub-install"；然而，如果你不是十分熟悉 GRUB 的内部原理，还是请不要使用这些命令。因为在一个正在运行的操作系统上安装引导加载器是一件极其危险的事情。

3.2 Making a GRUB bootable CD-ROM
3.2 制作一个 GRUB 启动光驱

==========

GRUB supports the "no emulation mode" in the El Torito specification(1) (*note Making a GRUB bootable CD-ROM-Footnote-1::). This means that you can use the whole CD-ROM from GRUB and you don't have to make a floppy or hard disk image file, which can cause compatibility problems.
GRUB 支持"非模拟模式"的 ElTorito 规范(1)。这意味着你可以使用完整的 CD-ROM 去启动 GRUB 而不必去制作一个软区或硬盘的镜像文件，从而避免导致兼容性问题。

For booting from a CD-ROM, GRUB uses a special image called 'cdboot.img', which is concatenated with 'core.img'. The 'core.img' used for this should be built with at least the 'iso9660' and 'biosdisk' modules. Your bootable CD-ROM will usually also need to include a configuration file 'grub.cfg' and some other GRUB modules.
要从光驱启动，GRUB 使用了一个叫做"cdboot.img"的特殊镜像，并且和"core.img"交互，用于交互的"core.img"镜像应至少构建有"iso9660"和 biosdisk"模块。你的可启动 CD-ROM 通常也需要包含一个"grub.cfg"配置文件及一些相关功能的 GRUB 模块。

To make a simple generic GRUB rescue CD, you can use the 'grub-mkrescue' program (*note Invoking grub-mkrescue::):
要制作一个简易通用的 GRUB 救援 CD，你可以使用"grub-mkrescue"命令：

```
   # grub-mkrescue -o grub.iso
```

You will often need to include other files in your image. To do this, first make a top directory for the bootable image, say, 'iso':
通常你需要在你的镜像里包含一些其它文件。为此，首先创建一个可启动镜像的顶层目录，命名为，"iso"：

```
# mkdir iso
```

Make a directory for GRUB:
创建一个 GRUB 的目录：

```
# mkdir -p iso/boot/grub
```

If desired, make the config file 'grub.cfg' under 'iso/boot/grub' (*note Configuration::), and copy any files and directories for the disc to the directory 'iso/'.
如有需要，可在"iso/boot/grub"的目录里创建一个配置文件"grub.cfg"，然后复制所有的文件和目录到用来制盘的"iso/"目录下。
Finally, make the image:
最后，制作成镜像：

```
# grub-mkrescue -o grub.iso iso
```

This produces a file named 'grub.iso', which then can be burned into a CD (or a DVD), or written to a USB mass storage device.
这将一个产生名为"grub.iso"的文件，它可以被刻录到一张 CD（或 DVD）中，也 可以被写入到一个 USB 大容量存储设备中。

The root device will be set up appropriately on entering your 'grub.cfg' configuration file, so you can refer to file names on the CD without needing to use an explicit device name. This makes it easier to produce rescue images that will work on both optical drives and USB mass storage devices.
而根设备的路径将被适当地设置到你的"grub.cfg"配置文件里，因此你可以在 CD 中引用文件名而不需要显式地使用设备名称。这将更加容易地制作出能同时工作在光驱和 USB 存储设备的救援镜像。
(1) El Torito is a specification for bootable CD using BIOS functions.
(1) EI Torito 是一个具有 BIOS 功能的可启动光驱的规范。

3.3 The map between BIOS drives and OS devices
3.3 BIOS 驱动器和操作系统设备之间的映射

==========

If the device map file exists, the GRUB utilities ('grub-probe', 'grub-setup', etc.) read it to map BIOS drives to OS devices. This file consists of lines like this:
如果存在设备的映射文件，那么 GRUB 公共程序（"grub-probe"、"grub-setup" 等等)会读取它并将 BIOS 驱动器映射至操作系统设备。该文件由像以下的文本行所组成：

```
(DEVICE) FILE
```

DEVICE is a drive specified in the GRUB syntax (*note Device syntax::), and FILE is an OS file, which is normally a device file.
DEVICE 是以 GRUB 语法去指定的一个驱动器，而 FILE 则为一个 OS 文件，它通常指定为一个设备文件。

Historically, the device map file was used because GRUB device names had to be used in the configuration file, and they were derived from BIOS drive numbers. The map between BIOS drives and OS devices cannot always be guessed correctly: for example, GRUB will get the order wrong if you exchange the boot sequence between IDE and SCSI in your BIOS.
以往，要使用设备映射文件是因为 GRUB 的配置文件中需要使用到设备名，并且它们都是由 BIOS 的驱动号而来。BIOS 驱动器和 OS 设备间的映射通常不能正确的被猜出：例如， 如果你在 BIOS 中改变 IDE 和 SCSI 的启动顺序，GRUB 就将会得到错误的次序。

Unfortunately, even OS device names are not always stable. Modern versions of the Linux kernel may

probe drives in a different order from boot to boot, and the prefix ('/dev/hd*' versus '/dev/sd*') may change depending on the driver subsystem in use. As a result, the device map file required frequent editing on some systems.
更不幸的是，甚至连 OS 设备名也不总是稳定的。不同版本的 Linux 内核在启动的时候可能会以不同的次序来加载驱动器，以及设备名所使用的路径前缀（如"/dev/hd"或"/dev/sd"）也依赖于所使用的设备子系统因而有可能发生改变。结果是，在一些系统上不得不需要频繁地去修改设备映射文件。

GRUB avoids this problem nowadays by using UUIDs or file system labels when generating 'grub.cfg', and we advise that you do the same for any custom menu entries you write. If the device map file does not exist, then the GRUB utilities will assume a temporary device map on the fly. This is often good enough, particularly in the common case of single-disk systems.
现今 GURB 为了避免这一难题于是通过使用 UUIDs 或文件系统卷标 Label 来生成"grub.cfg"，并且我们也建议你在自定义菜单选项的时候也这么做。如果设备映射文件不存在，那么 GRUB 公共程序会动态地假设一个临时的设备映射。这样通常已经足够应付了，尤其系统使用在单独硬盘的情况下。

However, the device map file is not entirely obsolete yet, and it is used for overriding when current environment is different from the one on boot. Most common case is if you use a partition or logical volume as a disk for virtual machine. You can put any comments in the file if needed, as the GRUB utilities assume that a line is just a comment if the first character is '#'.
然而，并不意味着设备映射文件该完全被淘汰，在某些情况下还是需要映射设备文件来实现引导。最常见的情况是你需要使用到虚拟机磁盘里的一个分区或一个逻辑卷。 如有必要的话你可以在文件中加入任何注释，因为 GRUB 会把一个以'#'开头的文本行当作成注释。

3.4 BIOS installation
3.4 安装到 BIOS
==========
MBR
MBR 方式
===
The partition table format traditionally used on PC BIOS platforms is called the Master Boot Record (MBR) format; this is the format that allows up to four primary partitions and additional logical partitions. With this partition table format, there are two ways to install GRUB: it can be embedded in the area between the MBR and the first partition (called by various names, such as the "boot track", "MBR gap", or "embedding area", and which is usually at least 31 KiB), or the core image can be installed in a file system and a list of the blocks that make it up can be stored in the first sector of that partition.
分区列表格式传统上是使用于 PC BIOS 平台并被称为主要引导记录(MBR)的格式；这样的格式最多只允许四个主分区及一个额外的逻辑分区。基于这样的分区列表格式，有两种安装 GRUB 的方法：他能够嵌入到 MBR 与第一个分区之间的区域(这有很多版本的称呼，例如叫"引导磁道(boot track)","MBR 间隔(MBR gap)",或"嵌入区域(embedding area)"，还有这区域通常至少有 31KiB)，或者把 GRUB 的内核镜像安装到文件系统的分区列表里从而实现让其储存到该分区的第一个扇区上。

Each of these has different problems. There is no way to reserve space in the embedding area with complete safety, and some proprietary software is known to use it to make it difficult for users to work around licensing restrictions; and systems are sometimes partitioned without leaving enough space before the first partition. On the other hand, installing to a filesystem means that GRUB is vulnerable to its blocks being moved around by filesystem features such as tail packing, or even by aggressive fsck implementations, so this approach is quite fragile; and this approach can only be used if the '/boot' filesystem is on the same disk that the BIOS boots from, so that GRUB does not have to rely on guessing BIOS drive numbers.
每一种都有不同的问题。无法保证嵌入到预留位置区域的做法是绝对安全的，因为已知的一些专有软件也在使用此方法从而造成这个区域保留版权信息的问题；况且某些系统在磁盘分出第一个分区之前并没有留下足够的空间。而另一方面，把 GRUB 安装到文件系统里意味着 GRUB 很脆弱因为储存它的数据块可能被文件系

统所具有的封装特性从而被移走，例如磁盘整理或一些数据块位置调整，甚至被具有侵略性的 fsck(file system check 文件系统检查工具)所破坏，导致 MBR 内的 GRUB 加载器无法找到 GRUB 的核心镜像，所以这个安装方式很不安全；而且最好在 BIOS 要引导的硬盘里分有"/boot"分区用来接收来自 MBR 的引导，从而让 GRUB 不需要依赖于猜测 BIOS 驱动器号。

The GRUB development team generally recommends embedding GRUB before the first partition, unless you have special requirements. You must ensure that the first partition starts at least 31 KiB (63 sectors) from the start of the disk; on modern disks, it is often a performance advantage to align partitions on larger boundaries anyway, so the first partition might start 1 MiB from the start of the disk.
GRUB 开发团队一般推荐把 GRUB 嵌入到第一个分区，除非你有特殊的要求。你必须确保第一分区的头部分最少有 31KiB(63 个扇区)的可用空间并且处于你要作为引导的硬盘之上，反正更大的对齐分区通常在性能上是有优势的，因此在磁盘分出第一个分区之前一般都会保留出 1MiB 的空间。

GPT
GPT
===
Some newer systems use the GUID Partition Table (GPT) format. This was specified as part of the Extensible Firmware Interface (EFI), but it can also be used on BIOS platforms if system software supports it; for example, GRUB and GNU/Linux can be used in this configuration. With this format, it is possible to reserve a whole partition for GRUB, called the BIOS Boot Partition. GRUB can then be embedded into that partition without the risk of being overwritten by other software and without being contained in a filesystem which might move its blocks around.
某些较新的系统会使用到 GUID 分区表(GPT)格式。这个格式是可扩展固件接口(EFI)所规范的一部分，但如果系统软件能够支持它那它也可以使用在 BIOS 平台上；例如，GRUB 和 GNU/Linux 都是可以使用这样的配置。使用这种格式，GRUB 是能够保存到一个专用的单独分区里，它被称为 BIOS 引导分区。从而让 GRUB 能够没有风险地嵌套到这个分区并且不被其他软件所覆盖或被覆盖，更不会被文件系统去移动它的块。

When creating a BIOS Boot Partition on a GPT system, you should make sure that it is at least 31 KiB in size. (GPT-formatted disks are not usually particularly small, so we recommend that you make it larger than the bare minimum, such as 1 MiB, to allow plenty of room for growth.) You must also make sure that it has the proper partition type. Using GNU Parted, you can set this using a command such as the following:
当在一个 GPT 系统里去创建 BIOS 引导分区时，你应该确保至少有 31KiBd 大小的保留空间。(GPT 格式的磁盘通常不会有这么小的分区，所以我们建议你让它大于最低限度，例如 1MiB，让其有足够的空间去增长。)你还必须确保它有正确的分区类型。在 GNU Parted 里，你可以使用例如以下命令来设置：

```
# parted /dev/DISK set PARTITION-NUMBER bios_grub on
```

If you are using gdisk, set the partition type to '0xEF02'. With partitioning programs that require setting the GUID directly, it should be '21686148-6449-6e6f-744e656564454649'.
如果你正在使用的是 Gdisk，需要把区模式设置分为'0xEF02'。并且按照分区程序的要求去设置它的 GUID，它应该为 '21686148-6449-6e6f-744e656564454649'.

Caution: Be very careful which partition you select! When GRUB finds a BIOS Boot Partition during installation, it will automatically overwrite part of it. Make sure that the partition does not contain any other data.
注意：你要非常小心地去选择哪个分区！当 GRUB 安装期间发现 BIOS 引导分区时，它会把安装内容自动覆盖到 BIOS 引导分区里。所以请确保这个分区里不包含任何其他的信息以免被覆盖。

04.启动系统-Booting

04.启动系统-Booting
********* **********

4 Booting
4 启动系统
==========
GRUB can load Multiboot-compliant kernels in a consistent way, but for some free operating systems you need to use some OS-specific magic.
GRUB 能以一致的方式加载兼容多重启动的内核，但对于一些特殊的 free 操作系统你需要使用一些 OS 所特定的技巧方法才能得以实现。

4.1 How to boot operating systems
4.1 如何启动操作系统
==========
GRUB has two distinct boot methods. One of the two is to load an operating system directly, and the other is to chain-load another boot loader which then wicll load an operating system actually. Generally speaking, the former is more desirable, because you don't need to install or maintain other boot loaders and GRUB is flexible enough to load an operating system from an arbitrary disk/partition. However, the latter is sometimes required, since GRUB doesn't support all the existing operating systems natively.
GRUB 有两种截然不同的引导方式。其中一种是直接地加载操作系统，而另一种实际上是以链式加载另一个引导加载器从而去加载一个操作系统。一般而言，前者是更可取的，因为你不需要去安装或维护其他的启动加载器并且 GRUB 本身能够灵活地去加载来自任意的磁盘或分区上的操作系统。然而，某些时候后者是必需的，因为 GRUB 本身并不能必然地支持现今的所有操作系统。

4.1.1 How to boot an OS directly with GRUB
4.1.1 GRUB 如何直接地引导到操作系统

Multiboot (*note Multiboot Specification: (multiboot)Top.) is the native format supported by GRUB. For the sake of convenience, there is also support for Linux, FreeBSD, NetBSD and OpenBSD. If you want to boot other operating systems, you will have to chain-load them (*note Chain-loading::).
多重引导对于 GRUB 来说是与生俱来就被支持的原生格式。为了方便理解，也可以说
Linux，FreeBSD，NetBSD 和 OpenBSD 等都支持它。但如果你需要启动其他类型的操作系统，你将需要使用链式加载的方式去启动它们。

FIXME: this section is incomplete.
注意这里：这部分并不完整。

1. Run the command 'boot' (*note boot::).
1.运行命令 "boot"来启动，请参阅"启动-boot"章节。

However, DOS and Windows have some deficiencies, so you might have to use more complicated instructions. *Note DOS/Windows::, for more information.
然而，DOS 和 Windows 有一些不足之处，所以你可能需要使用到更复杂的指令。请参阅"DOS/Windows"章节以获取更多相关信息。

4.1.2 Chain-loading an OS

4.1.2 链式-加载启动一个操作系统

Operating systems that do not support Multiboot and do not have specific support in GRUB (specific support is available for Linux, FreeBSD, NetBSD and OpenBSD) must be chain-loaded, which involves loading another boot loader and jumping to it in real mode.
操作系统完全不支持 GRUB 的多重引导与特性(并且需要特殊地支持 Linux，FreeBSD，NetBSD 和 OpenBSD 系统)的时候，就必须用到链式-加载的方式来加载，它涉及到加载另一个引导加载器并且跳转到实(址)模式来启动它。
(实模式-为汇编内容：此方式的 CPU 最大寻址为 1MB，逻辑段最大为 64KB，数据总线最大为 20 位，一般用 16 位的寄存器如：8086CPU)

The 'chainloader' command (*note chainloader::) is used to set this up. It is normally also necessary to load some GRUB modules and set the appropriate root device. Putting this together, we get something like this, for a Windows system on the first partition of the first hard disk:
命令"chainloader"(注：链式加载器)是用于启用设置的。通常它还需要加载一些 GRUB 的模块和设置好适当的引导设备。把它们放到一起，我们得到了(以下)这些，适合于 Windows 系统并且它在第一个硬盘的第一个分区：

```
menuentry "Windows" {
      insmod chain
      insmod ntfs
      set root=(hd0,1)
      chainloader +1
}
```

On systems with multiple hard disks, an additional workaround may be required. *Note DOS/Windows::.
在系统里有多个硬盘的时候，可能需要在相关的地方去配置更多额外的命令。更多内容请参阅"DOS/Windows"
Chain-loading is only supported on PC BIOS and EFI platforms.
链式-加载启动只是支持 PC BIOS 平台和 EFI 平台。

4.2 Loopback booting
4.2 Loopback booting(回路引导)

==========

GRUB is able to read from an image (be it one of CD or HDD) stored on any of its accessible storages (refer to *note loopback:: command). However the OS itself should be able to find its root. This usually involves running a userspace program running before the real root is discovered. This is achieved by GRUB loading a specially made small image and passing it as ramdisk to the kernel. This is achieved by commands 'kfreebsd_module', 'knetbsd_module_elf', 'kopenbsd_ramdisk', 'initrd' (*note initrd::), 'initrd16' (*note initrd::), 'multiboot_module', 'multiboot2_module' or 'xnu_ramdisk' depending on the loader. Note that for knetbsd the image must be put inside miniroot.kmod and the whole miniroot.kmod has to be loaded. In kopenbsd payload this is disabled by default. Aditionally behaviour of initial ramdisk depends on command line options. Several distributors provide the image for this purpose or it's integrated in their standard ramdisk and activated by special option. Consult your kernel and distribution manual for more details. Other loaders like appleloader, chainloader (BIOS, EFI, coreboot), freedos, ntldr and plan9 provide no possibility of loading initial ramdisk and as far as author is aware the payloads in question don't support either initial ramdisk or discovering loopback boot in other way and as such not bootable this way. Please consider alternative boot methods like copying all files from the image to actual partition. Consult your OS documentation for more details.
GRUB 能够读取一个储存在任何可访问储存设备中的镜像(无论是 CD 还是 HDD)。然而操作系统本身则应该能够找到它自身的根路径。这通常涉及到可能当中某些程序需要获取到根目录的位置。这是通过 GRUB 载入一个特制的小镜像并且载入内存虚拟盘作为内核。这些实现自命令"kfreebsd_module"，"knetbsd_module_elf"，"kopenbsd_ramdisk"，"initrd"，"initrd16"，"multiboot

_module"，"multiboot2_module"或"xnu_ramdisk"及其相关依赖的加载。在 kopenbsd 里的有效负载默认是关闭的，另外去初始内存虚拟盘的行为依赖于命令行选项。一些发行商提供了用于这一目的的单独镜像或者它已集成了标准内存虚拟盘并且通过特殊选项来激活，请查询你的内核和公布手册获得更多细节。另外一些加载程序可能是苹果加载器，链式加载器(BIOS,EFI,coreboot),freedos,ntldr 和 plan9 不可能提供加载初始化的内存虚拟盘并且作者也意识到有效负载的问题不能支持任何一个初始化内存虚拟盘或者通过一些找出地址引导的方法都是不能实现引导的。请参考备选的引导方法从镜像里复制所有的文件到实际的分区。请查询你的内核和公布手册获得更多细节。

4.3 Some caveats on OS-specific issues
4.3 一些 OS-特性的注意事项

Here, we describe some caveats on several operating systems.
在这里，我们以一些操作系统去描述那些警告。

4.3.1 GNU/Hurd
4.3.1 GNU/Hurd

Since GNU/Hurd is Multiboot-compliant, it is easy to boot it; there is nothing special about it. But do not forget that you have to specify a root partition to the kernel.
因为 GNU/Hurd 是遵从多重引导的，所以很容易启动它；这没什么特别的。但是别忘记了你必须正确地指定出一个内核所在的根分区。

1. Set GRUB's root device to the same drive as GNU/Hurd's. The command 'search --set=root --file /boot/gnumach.gz' or similar may help you (*note search::).
1.把 GRUB 的根设备设置为某个 GNU/Hurd 所在的驱动器。命令为"search --set=root --file /boot/gnumach.gz" 或其他类似的有效命令。此命令很有效，它会自动搜索 /boot/gnumach.gz 文件，并把存放此文件的分区设置为根设备。
注释：根分区和根设备不是一个含义，根分区是指文件系统根目录所在的分区。根设备是指 GRUB 加载 OS 内核时。一个名为 root 的选项值，它指向内核所在的那个分区及其所处的那个设备编号。
如：set root=(hd0,1)

2. Load the kernel and the modules, like this:
2. 加载所需的内核及其模块，如下：

```
grub> multiboot /boot/gnumach.gz root=device:hd0s1
grub> module  /hurd/ext2fs.static ext2fs --readonly \
  --multiboot-command-line='${kernel-command-line}' \
  --host-priv-port='${host-port}' \
  --device-master-port='${device-port}' \
  --exec-server-task='${exec-task}' -T typed '${root}' \
  '$(task-create)' '$(task-resume)'
grub> module /lib/ld.so.1 exec /hurd/exec '$(exec-task=task-create)'
```

3. Finally, run the command 'boot' (*note boot::).
3. 最后，运行启动命令 "boot"

4.3.2 GNU/Linux
4.3.2 GNU/Linux

It is relatively easy to boot GNU/Linux from GRUB, because it somewhat resembles to boot a Multiboot-compliant OS.

从 GRUB 去引导 GNU/Linux 也是相对容易的，因为它有点类似于去引导一个遵从多重引导规范的系统。

1. Set GRUB's root device to the same drive as GNU/Linux's. The command 'search --set=root --file /vmlinuz' or similar may help you (*note search::).
1. 设置 GRUB 的根设备属性为 GNU/Linux 所在的驱动器。命令为 "search --set=root --file /vmlinuz" 或其他类似的有效命令。

2. Load the kernel using the command 'linux' (*note linux::):
2. 通过命令"linux"去加载内核：
grub> linux /vmlinuz root=/dev/sda1

If you need to specify some kernel parameters, just append them to the command. For example, to set 'acpi' to 'off', do this:
如果你需要指定一些内核参数，只要将它们添加到命令后面即可。例如，要把"acpi"设置为"off"，可以这样做：
grub> linux /vmlinuz root=/dev/sda1 acpi=off
See the documentation in the Linux source tree for complete information on the available options.
至于想了解更完整的可用选项信息，请参考 Linux 源代码树中的帮助文档。

With 'linux' GRUB uses 32-bit protocol. Some BIOS services like APM or EDD aren't available with this protocol. In this case you need to use 'linux16'
在 GRUB 里的"linux"命令默认是使用 32-bit 协议的。这在某些 BIOS 服务如 APM 或 EDD 下是不可用的。在这种情况下，你需要使用"linux16"命令：
grub> linux16 /vmlinuz root=/dev/sda1 acpi=off

3. If you use an initrd, execute the command 'initrd' (*note initrd::) after 'linux':
3.如果你还需要去使用一个初始化内存磁盘(initrd)，则需要在执行了"linux"命令之后去执行"initrd"命令：
grub> initrd /initrd
If you used 'linux16' you need to use 'initrd16':
如果你使用的是"linux16"命令则必需使用"initrd16"命令：
grub> initrd16 /initrd

4. Finally, run the command 'boot' (*note boot::).
4. 最后，运行启动命令 "boot"。

Caution: If you use an initrd and specify the 'mem=' option to the kernel to let it use less than actual memory size, you will also have to specify the same memory size to GRUB. To let GRUB know the size, run the command 'uppermem' _before_ loading the kernel. *Note uppermem::, for more information.
警告：如果你使用了 initrd 指令并且想通过使用 mem=内核参数设置了小于实际内存大小的使用范围。你也必需在 GRUB 中设置相同的容量值。要让 GRUB 获取此数值，需要在加载启动 linux 内核之前去执行 uppermem 命令。

4.3.3 DOS/Windows
4.3.3 DOS/Windows

GRUB cannot boot DOS or Windows directly, so you must chain-load them (*note Chain-loading::). However, their boot loaders have some critical deficiencies, so it may not work to just chain-load them. To overcome the problems, GRUB provides you with two helper functions.
GRUB 不能直接启动 DOS 或 Windows，所以你必须使用链式-加载器去启动它们。然而，它们的引导载入程序存在着一些关键性的缺陷，所以单单使用链式-加载器去启动它们也可能会失败。为了克服这些问题，GRUB 为你提供了两个协助功能函数。

If you have installed DOS (or Windows) on a non-first hard disk, you have to use the disk swapping

technique, because that OS cannot boot from any disks but the first one. The workaround used in GRUB is the command 'drivemap' (*note drivemap::), like this:
如果你已经把 DOS(或 Windows)安装了但不是在第一个硬盘上，你必须使用硬盘交换技术，因为这些系统不能够启动于除了第一个硬盘以外的任何硬盘。GRUB 使用的解决方案是命令"drivemap"，像这样：

```
drivemap -s (hd0) (hd1)
```

This performs a "virtual" swap between your first and second hard drive.
这里"虚拟"地把你的第一个硬盘与第二个硬盘做了交换，也可以说虚拟交换了两块硬盘。
Caution: This is effective only if DOS (or Windows) uses BIOS to access the swapped disks. If that OS uses a special driver for the disks, this probably won't work.
注意：这只有效于操作系统通过 BIOS 去使用硬盘时有效。但如果那个操作系统使用了自带的驱动器号去访问硬盘，那硬盘虚拟交换将会对操作系统失效。

Another problem arises if you installed more than one set of DOS/Windows onto one disk, because they could be confused if there are more than one primary partitions for DOS/Windows. Certainly you should avoid doing this, but there is a solution if you do want to do so. Use the partition hiding/unhiding technique.
如果你安装更多的不止一套 DOS/Windows 到一个硬盘上也可能发生问题，因为如果有一个以上的 DOS/Windows 主分区可能会引起它们的混肴。当然你应该避免这样做，但这有一个解决方案，如果你想这样做。使用分区隐藏/反隐藏技术。

If GRUB "hides" a DOS (or Windows) partition (*note parttool::), DOS(or Windows) will ignore the partition. If GRUB "unhides" a DOS (or Windows) partition, DOS (or Windows) will detect the partition. Thus,if you have installed DOS (or Windows) on the first and the second partition of the first hard disk, and you want to boot the copy on the first partition, do the following:
如果 GRUB"隐藏(hides)"一个 DOS(或 Windows)分区，DOS(或 Windows)将忽略这个分区。如果 GRUB"反隐藏(unhides)"一个 DOS(或 Windows)分区，DOS(或 Windows)将会检测出这个分区。由此，如果已经把 DOS(或 Windows)分别安装在第一个硬盘的第一个分区与第二个分区里，而且你需要引导这个例子里的第一分区，可以执行以下操作：

```
parttool (hd0,1) hidden-
parttool (hd0,2) hidden+
set root=(hd0,1)
chainloader +1
parttool ${root} boot+
boot
```

05.编写你自己的配置文件-Writing your own configuration file

05.编写你自己的配置文件-Writing your own configuration file
****

5 Writing your own configuration file
5 编写你自己的配置文件
==========
GRUB is configured using 'grub.cfg', usually located under '/boot/grub'.This file is quite flexible, but most users will not need to write the whole thing by hand.
GRUB 所使用的配置文件名为"grub.cfg"，它通常放在"/boot/grub"里。这个配置文件非常灵活，绝大多数用户都不需要以手工的方式去完全编写出里面的所有内容。

5.1 Simple configuration handling
5.1 简单的配置操作
==========
The program 'grub-mkconfig' (*note Invoking grub-mkconfig) generates 'grub.cfg' files suitable for most cases. It is suitable for use when upgrading a distribution, and will discover available kernels and attempt to generate menu entries for them.
使用"grub-mkconfig"程序会生成出适用于大部分情况下"grub.cfg"文件。这个"grub-mkconfig"程序也适用于在升级时一起配用，程序将会去寻找出可用的内核并且尝试去生成出它的菜单条目。

'grub-mkconfig' does have some limitations. While adding extra custom menu entries to the end of the list can be done by editing '/etc/grub.d/40_custom' or creating '/boot/grub/custom.cfg', changing the order of menu entries or changing their titles may require making complex changes to shell scripts stored in '/etc/grub.d/'. This may be improved in the future. In the meantime, those who feel that it would be easier to write 'grub.cfg' directly are encouraged to do so (*note Booting::, and *note Shell-like scripting::), and to disable any system provided by their distribution to automatically run 'grub-mkconfig'.
"grub-mkconfig"程序确实存在一些局限性。当你希望通过菜单项列表的后面去添加额外的自定义菜单项目时你可以通过编辑"/etc/grub.d/40_custom"文件来实现或者创建出一个"/boot/grub/custom.cfg"配置文件，而要修改菜单项的顺序或者修改它们的标题时这可能需要对储存在"/etc/grub.d/"目录里的 shell 脚本文件进行更复杂的修改，或者通过修改那些 shell 脚本文件的前缀数字来实现，那些数字代表了排序顺序，数字越小则越往前排列。这些局限性可能在未来会有所改进。在此期间，那些认为修改"grub.cfg"配置文件会更容易地写好配置的人我们会直接地鼓励它这样做，并且建议他去禁用它们系统所提供的自动运行"grub-mkconfig"程序的任务进程。

The file '/etc/default/grub' controls the operation of 'grub-mkconfig'. It is sourced by a shell script, and so must be valid POSIX shell input; normally, it will just be a sequence of 'KEY=value' lines, but if the value contains spaces or other special characters then it must be quoted. For example:
这个"/etc/default/grub"文件是被"grub-mkconfig"程序控制操作的。它其实就是一个 shell 脚本，所以必须为有效的 POSIX shell(可移植性操作系统接口)输入；它实际就是由一列列的"KEY=value"所组成的序列，但如果它的值带有空格或其他特殊的字符的时候则必须用双引号来包裹，例如：
```
GRUB_TERMINAL_INPUT="console serial"
```

Valid keys in '/etc/default/grub' are as follows:
"/etc/default/grub"里有效的键值如下：

'GRUB_DEFAULT'
The default menu entry. This may be a number, in which case it identifies the Nth entry in the generated menu counted from zero, or the title of a menu entry, or the special string 'saved'. Using the title may be useful if you want to set a menu entry as the default even though there may be a variable number of

entries before it.
"GRUB_DEFAULT"
指定作为默认启动的菜单项。它的值可以是一个数字，这种情况下它能够识别生成菜单中的从 0 开始数的第 N 个条目。或是可以是一个菜单项的标题名，又或者是一个特殊符号串"saved"。使用菜单项的标题可能比较有好处，因为你想设置的那个默认的菜单项可能是在一个菜单条目数量与排序都在不断变化的配置菜单里。

For example, if you have:
例如，如果你有：

```
menuentry 'Example GNU/Linux distribution' --class gnu-linux {
...
}
```

then you can make this the default using:
然后你可以使用这个 default 键值：

```
GRUB_DEFAULT='Example GNU/Linux distribution'
```

If you set this to 'saved', then the default menu entry will be that saved by 'GRUB_SAVEDEFAULT', 'grub-set-default', or 'grub-reboot'. The default is '0'.
如果你设置为"saved"，那么手动所选的菜单项的排列位置数将被作为默认的菜单项排列位置数并且保存到"GRUB_SAVEDEFAULT"，"grub-set-default"，或"grub-reboot"里。而它的默认初始值为"0"。

'GRUB_SAVEDEFAULT'
If this option is set to 'true', then, when an entry is selected,save it as a new default entry for use by future runs of GRUB. This is only useful if 'GRUB_DEFAULT=saved'; it is a separate option because 'GRUB_DEFAULT=saved' is useful without this option, in conjunction with 'grub-set-default' or 'grub-reboot'. Unset by default. This option relies on the environment block, which may not be available in all situations (*note Environment block::).
"GRUB_SAVEDEFAULT"
如果将此选项设置为"true"，那么，当一个条目被选中时，会将其条目排列位数保存下来并且把此条目作为一个默认条目提供给 GRUB 来去运行。而且这个键值只唯一有效于"GRUB_DEFAULT=saved"的情况下；它只有单一选项"GRUB_DEFAULT=saved"，并且相互协调于"grub-set-default"或"grub-reboot"。默认情况下。这个选项依赖于环境模块，这可能不是在所有的情况下都为可用。

'GRUB_TIMEOUT'
Boot the default entry this many seconds after the menu is displayed, unless a key is pressed. The default is '5'. Set to '0' to boot immediately without displaying the menu, or to '-1' to wait indefinitely.
"GRUB_TIMEOUT"
其值为一个数，表示启动显示菜单后默认多少秒才进入默认条目，但当有一个键被按下除外。这里默认为"5"。设置为"0"则立即启动默认条目而不会显示菜单，或者设为"-1"则是无限期等待。

'GRUB_HIDDEN_TIMEOUT'
Wait this many seconds for a key to be pressed before displaying the menu. If no key is pressed during that time, display the menu for the number of seconds specified in GRUB_TIMEOUT before booting the default entry. We expect that most people who use GRUB_HIDDEN_TIMEOUT will want to have GRUB_TIMEOUT set to '0' so that the menu is not displayed at all unless a key is pressed. Unset by default.
"GRUB_HIDDEN_TIMEOUT"
在任意按键被按下之前等待多少秒才显示菜单。如果 GRUB_TIMEOUT 指定的秒数倒数完之后都没有按下任意按键，则直接加载 GRUB_DEFAULT 里的默认条目。我们希望大多数人用 GRUB_HIDDEN_TIMEOUT 来配合把 GRUB_TIMEOUT 属性设置为"0"的时候一起来使用，因为那样做会避免 GRUB_TIMEOUT 的等待操作时间为"0"而导致的就算按下了一个任意键的情况下菜单依然不会显示而是直接启动默认菜单项目的问题，而 GRUB_HIDDEN_TIMEOUT 能在其之前起到按下任意键之后立即显示出菜单的作用从而跳过 GRUB_TIMEOUT 的"0"秒倒数，而 GRUB_HIDDEN_TIMEOUT 的默认值为"空"。

'GRUB_HIDDEN_TIMEOUT_QUIET'
In conjunction with 'GRUB_HIDDEN_TIMEOUT', set this to 'true' to suppress the verbose countdown

while waiting for a key to be pressed before displaying the menu. Unset by default.
"GRUB_HIDDEN_TIMEOUT_QUIET"
它与"GRUB_HIDDEN_TIMEOUT"协同使用，设置它为"true"会以静默隐藏的模式去计时除非期间有一个键被按下转而去显示菜单。默认值为"空"。而设置为"afult"则为显示计时器。

'GRUB_DEFAULT_BUTTON'
"GRUB_DEFAULT_BUTTON"

'GRUB_TIMEOUT_BUTTON'
"GRUB_TIMEOUT_BUTTON"

'GRUB_HIDDEN_TIMEOUT_BUTTON'
"GRUB_HIDDEN_TIMEOUT_BUTTON"

'GRUB_BUTTON_CMOS_ADDRESS'
Variants of the corresponding variables without the '_BUTTON' suffix, used to support vendor-specific power buttons. *Note Vendor power-on keys.
"GRUB_BUTTON_CMOS_ADDRESS"
相应没有"_BUTTON"为后缀的变异变量，用于支持特定供应商的电源按钮。例如某些笔记本的多媒体特殊按键等。

'GRUB_DISTRIBUTOR'
Set by distributors of GRUB to their identifying name. This is used to generate more informative menu entry titles.
"GRUB_DISTRIBUTOR"
设置 GRUB 所显示的经销商名称。这是用于生成更多信息到菜单选项的标题里。

'GRUB_TERMINAL_INPUT'
Select the terminal input device. You may select multiple devices here, separated by spaces.
"GRUB_TERMINAL_INPUT"
选择终端输入设备。在这里你可以选择多个设备，用空格作分隔。
Valid terminal input names depend on the platform, but may include 'console' (PC BIOS and EFI consoles), 'serial' (serial terminal), 'ofconsole' (Open Firmware console), 'at_keyboard' (PC AT keyboard, mainly useful with Coreboot), or 'usb_keyboard' (USB keyboard using the HID Boot Protocol, for cases where the firmware does not handle this).
有效的终端输入名称取决于平台，它们可能包括"console"(PC BIOS 和 EFIconsoles 控制台)，"serial"(串口控制台)，"ofconsole"(开放固件控制台)，"at_keyboard"(PC 机键盘，主要有助于 Coreboot：段尾解析)，或"usb_keyboard"(使用 HID 启动协议的 USB 键盘，当固件不处理这个问题的情况下。)
注 Coreboot：
LinuxBIOS 計劃，一個 Free Software 的計劃，企圖取代掉現今多數電腦上的 BIOS 已經改名稱為 coreboot。實際上 coreboot 只是一個小型驅動硬體的韌體，只提供基本的硬體初始化，然後就把控制權交給其它眾多的載體之一。coreboot 並不是專為 Linux 而製作的，雖然它可以使用 Linux 核心來當作載體，像是你把 Linux 核心放到你的 flash ROM 晶片上面跟 coreboot 放在一起，或是間接透過 FILO 或 GRUB2 來啟動 Linux 核心，不過透過適當的載體也可以用來啟動其它作業系統像是 Plan 9、Windows、FreeBSD 等。(以上引自：http://www.oschina.net/p/coreboot/)
注 HID：
是 Human Interface Device 的缩写，由其名称可以了解 HID 设备是直接与人交互的设备，例如键盘、鼠标与游戏杆等。不过 HID 设备并不一定要有人机接口，只要符合 HID 类别规范的设备都是 HID 设备(引自：http://baike.baidu.com/link?url=e71NiaEFkjA4-azdAw3GKPqBKzZ12ifUhZpnULdMo7xLVStwjM49d1sSWvt9m4VqxL3onDzAzuBe-0WQo7vIla)
The default is to use the platform's native terminal input.
它默认是使用当前平台的本地终端输入。

'GRUB_TERMINAL_OUTPUT'

Select the terminal output device. You may select multiple devices here, separated by spaces.
"GRUB_TERMINAL_OUTPUT"
选择终端输出设备。这里你可以选择多个设备，用空格分开。
Valid terminal output names depend on the platform, but may include 'console' (PC BIOS and EFI consoles), 'serial' (serial terminal), 'gfxterm' (graphics-mode output), 'ofconsole' (Open Firmware console), or 'vga_text' (VGA text output, mainly useful with Coreboot).
有效的终端输出名称取决于平台，但可能包括"console"(PC BIOS 和 EFIconsoles 控制台)，"serial"(串口控制台)，"gfxterm"(图形-模式 输出)，"ofconsole"(开放固件控制台)，或"vga_text"(带 VGA 驱动的文本输出模式，主要有助于使用 Coreboot)。
The default is to use the platform's native terminal output.
它默认是使用当前平台的本地终端输出。

'GRUB_TERMINAL'
If this option is set, it overrides both 'GRUB_TERMINAL_INPUT' and 'GRUB_TERMINAL_OUTPUT' to the same value.
"GRUB_TERMINAL"
如果设置了这个选项，它会覆盖"GRUB_TERMINAL_INPUT"与"GRUB_TERMINAL_OUTPUT"这两个功能里的值，从而实现相同的输入输出。

'GRUB_SERIAL_COMMAND'
A command to configure the serial port when using the serial console. *Note serial::. Defaults to 'serial'.
"GRUB_SERIAL_COMMAND"
当前使用串行控制台的时候用来配置串行端口命令。它默认值为"serial"。

'GRUB_CMDLINE_LINUX'
Command-line arguments to add to menu entries for the Linux kernel.
"GRUB_CMDLINE_LINUX"
添加到菜单条目所启动的 Linux 内核的命令行参数。

'GRUB_CMDLINE_LINUX_DEFAULT'
Unless 'GRUB_DISABLE_RECOVERY' is set to 'true', two menu entries will be generated for each Linux kernel: one default entry and one entry for recovery mode. This option lists command-line arguments to add only to the default menu entry, after those listed in 'GRUB_CMDLINE_LINUX'.
"GRUB_CMDLINE_LINUX_DEFAULT"
除非"GRUB_DISABLE_RECOVERY"设置为"true"，否则每个基于 Linux 内核的菜单选项都将会生成出其各自的两个菜单选项版本：一个默认的正常模式选项和另一个恢复模式选项。至于那些内核的参数选项只会添加到默认的正常模式里，并且会添加在"GRUB_CMDLINE_LINUX"功能的参数列表后面。

'GRUB_CMDLINE_NETBSD'
'GRUB_CMDLINE_NETBSD_DEFAULT' As 'GRUB_CMDLINE_LINUX' and 'GRUB_CMDLINE_LINUX_DEFAULT', but for NetBSD.
"GRUB_CMDLINE_NETBSD"
"GRUB_CMDLINE_NETBSD_DEFAULT" 如同 "GRUB_CMDLINE_LINUX"以及"GRUB_CMDLINE_LINUX_DEFAULT"一样，但区分为它只用于 NetBSD。

'GRUB_CMDLINE_GNUMACH'
as 'GRUB_CMDLINE_LINUX', but for GNU Mach.
"GRUB_CMDLINE_GNUMACH"
和"GRUB_CMDLINE_LINUX"一样，但只用于 GNU Mach。

'GRUB_CMDLINE_XEN'
'GRUB_CMDLINE_XEN_DEFAULT'
The values of these options are appended to the values of 'GRUB_CMDLINE_LINUX' and 'GRUB_CMDLINE_LINUX_DEFAULT' for Linux and Xen menu entries.

"GRUB_CMDLINE_XEN"
"GRUB_CMDLINE_XEN_DEFAULT"
这些选项的值会被附加到"GRUB_CMDLINE_LINUX"与"GRUB_CMDLINE_LINUX_DEFAULT"里所对应的 Linux 菜单选项和 Xen 菜单选项里。也就是说类似于"GRUB_CMDLINE_LINUX"以及"GRUB_LINUX_DEFAULT"一样，但只作用于 linux 以及 Xen。
注：
Xen 是一个开放源代码虚拟机监视器，一个基于 X86 架构、发展最快、性能最稳定、占用资源最少的开源虚拟化技术。由剑桥大学开发。(引自：http://baike.baidu.com/link?url=koikxgrO03Mn12vYL4fC-auDr3pSjMFOKxih_tn82uQqcjeyuXZdhK5l_ISmVriQZO4Os9EvlAC0-qZ52dKaiK)

'GRUB_CMDLINE_LINUX_XEN_REPLACE'
'GRUB_CMDLINE_LINUX_XEN_REPLACE_DEFAULT'
The values of these options replace the values of 'GRUB_CMDLINE_LINUX' and 'GRUB_CMDLINE_LINUX_DEFAULT' for Linux and Xen menu entries.
"GRUB_CMDLINE_LINUX_XEN_REPLACE"
"GRUB_CMDLINE_LINUX_XEN_REPLACE_DEFAULT"
这些选项的值会替换到"GRUB_CMDLINE_LINUX"与"GRUB_CMDLINE_LINUX_DEFAULT"里对应的 Linux 和 Xen 菜单条目里的值。

'GRUB_DISABLE_LINUX_UUID'
Normally, 'grub-mkconfig' will generate menu entries that use universally-unique identifiers (UUIDs) to identify the root filesystem to the Linux kernel, using a 'root=UUID=...' kernel parameter. This is usually more reliable, but in some cases it may not be appropriate. To disable the use of UUIDs, set this option to 'true'.
"GRUB_DISABLE_LINUX_UUID"
一般情况下，"grub-mkconfig"将使用全局唯一标识符(UUIDs)来生成出的菜单条目来去识别 Linux 内核的根文件系统，使用一个"root=UUID=..."内核参数，"UUID"表示根文件系统。这通常更加可靠，但在某些情况下它可能不是适当的。若要禁止使用 UUIDs，则把这个选项设置为"true"。

'GRUB_DISABLE_RECOVERY'
If this option is set to 'true', disable the generation of recovery mode menu entries.
"GRUB_DISABLE_RECOVERY"
如果这个选项设置为"true"，则会关闭生成恢复模式的菜单条目。

'GRUB_VIDEO_BACKEND'
If graphical video support is required, either because the 'gfxterm' graphical terminal is in use or because 'GRUB_GFXPAYLOAD_LINUX' is set, then 'grub-mkconfig' will normally load all available GRUB video drivers and use the one most appropriate for your hardware. If you need to override this for some reason, then you can set this option.
"GRUB_VIDEO_BACKEND"
如果你需要得到图形视频的支持，要么可以通过使用"gfxterm"图形终端或要么可以通过设置"GRUB_GFXPAYLOAD_LINUX"参数选项来开启，然后使用"grub-mkconfig"命令在通常的情况下去加载所有有效的 GRUB 视频驱动程序并且自动使用一个最适合你硬件的视频驱动。如果你因某些原因而需要去重写它，那么你可以去设置这个键值的选项。
After 'grub-install' has been run, the available video drivers are listed in '/boot/grub/video.lst'.
在 "grub-install"程序被执行结束后，那些可用的视频驱动程序会在"/boot/grub/video.lst"文件里列出。

'GRUB_GFXMODE'
Set the resolution used on the 'gfxterm' graphical terminal. Note that you can only use modes which your graphics card supports via VESA BIOS Extensions (VBE), so for example native LCD panel resolutions may not be available. The default is 'auto', which tries to select a preferred resolution.
*Note gfxmode.
"GRUB_GFXMODE"
设置"gfxterm"图形终端的分辨率。注意你的显卡必需支持唯一可选的 VESA BIOS Extensions(VBE：视频电

子标准协会 BIOS 扩展)模式，因此可能不适用于某些本地液晶屏的分辨率。它的默认值为"auto"，此值会去试图选择一个最佳分辨率。也可理解为"auto"是最低的公用分辨率"640x480"。

'GRUB_BACKGROUND'
Set a background image for use with the 'gfxterm' graphical terminal. The value of this option must be a file readable by GRUB at boot time, and it must end with '.png', '.tga', '.jpg', or '.jpeg'. The image will be scaled if necessary to fit the screen.
"GRUB_BACKGROUND"
设置一个使用到"gfxterm"图形终端里的背景图片。这个选项的值必需是一个可读的文件并且在 GRUB 启动的过程中会被读到，而且此文件必需为以".png"，".tga"，".jpg"或"jpeg"为结尾的图片。这个图片为了适应屏幕分辨率的需要而会被自动缩放。

'GRUB_THEME'
Set a theme for use with the 'gfxterm' graphical terminal.
"GRUB_THEME"
设置一个使用到"gfxterm"图形终端的主题。

'GRUB_GFXPAYLOAD_LINUX'
Set to 'text' to force the Linux kernel to boot in normal text mode, 'keep' to preserve the graphics mode set using 'GRUB_GFXMODE', 'WIDTHxHEIGHT'['xDEPTH'] to set a particular graphics mode, or a sequence of these separated by commas or semicolons to try several modes in sequence. *Note gfxpayload.
"GRUB_GFXPAYLOAD_LINUX"
使用"text"参数会强制 Linux 内核启动到标准的文本模式，而使用"keep"参数则会保留起所使用的"GRUB_GFXMODE"图形模式里的设置，而通过使用"WIDTHxHEIGHT"["xDEPTH"]参数则能够去设置自定义的图形模式的"宽 x 高 x 刷新率"，或者可以用逗号或分号作分隔符来设置出多个配置从而实现多个尝试模式的序列。
Depending on your kernel, your distribution, your graphics card, and the phase of the moon, note that using this option may cause GNU/Linux to suffer from various display problems, particularly during the early part of the boot sequence. If you have problems, set this option to 'text' and GRUB will tell Linux to boot in normal text mode.
根据你所使用的内核，所使用的发布版本，所使用的显卡，和一些别的概率性问题，注意使用这个选项可能会导致 GNU/Linux 遭受很多各种各样的显示问题，特别是在启动的早期阶段。如果你在显示上出了问题，可以把此选项设置为"text"从而让 GRUB 告诉 Linux 启动到正常的文本模式里。

'GRUB_DISABLE_OS_PROBER'
Normally, 'grub-mkconfig' will try to use the external 'os-prober' program, if installed, to discover other operating systems installed on the same system and generate appropriate menu entries for them. Set this option to 'true' to disable this.
"GRUB_DISABLE_OS_PROBER"
通常地，"grub-mkconfig"将会尝试去使用一个外部的系统探测器"os-prober"程序，如果它已安装，则会在上述系统中去发现其他已安装了的操作系统并且生成出它们的相应菜单项。这个选项设为"true"则为关闭。

'GRUB_INIT_TUNE'
Play a tune on the speaker when GRUB starts. This is particularly useful for users unable to see the screen. The value of this option is passed directly to *note play.
"GRUB_INIT_TUNE"
当 GRUB 启动时从扬声器演奏一首曲子。这是对于无法看到屏幕上的用户尤其有用。至于这个选项的参数值请直接转到"play"章节查询。

'GRUB_BADRAM'
If this option is set, GRUB will issue a *note badram:: command to filter out specified regions of RAM.
"GRUB_BADRAM"
如果使用了此选项的设置，GRUB 则将会发出一个 badram 命令去过滤掉指定的 RAM(内存)区域。

'GRUB_PRELOAD_MODULES'
This option may be set to a list of GRUB module names separated by spaces. Each module will be loaded as early as possible, at the start of 'grub.cfg'.
"GRUB_PRELOAD_MODULES"
这个选项可以设置为一个 GRUB 模块名列表并且以空格键作为分隔。模块与模块之间将从前往后被合理地加载，全部加载完后才去执行"grub.cfg"。
For more detailed customisation of 'grub-mkconfig''s output, you may edit the scripts in '/etc/grub.d' directly. '/etc/grub.d/40_custom' is particularly useful for adding entire custom menu entries; simply type the menu entries you want to add at the end of that file, making sure to leave at least the first two lines intact.
如需定制化更详细的"grub-mkconfig"输出，你可以直接去编辑位于"/etc/grub.d"目录里的脚本文件。而里面的"etc/grub.d/40_custom"文件对于添加完整的自定义菜单项格外有用；你只需把想输入的菜单项目添加到文件的末尾，并且确保离开时前面的头两行原封不动。

5.2 Writing full configuration files directly
5.2 直接写出完整的配置文件
==========
'grub.cfg' is written in GRUB's built-in scripting language, which has a syntax quite similar to that of GNU Bash and other Bourne shell derivatives.
"grub.cfg"文件的内容是以 GRUB 的内置脚本语言来编写，它的语法类似于 GNU Bash 以及其他 Bourne shell 衍生产品。

Words
单词
=====
A "word" is a sequence of characters considered as a single unit by GRUB. Words are separated by "metacharacters", which are the following plus space, tab, and newline:
一个由"字母"所组成的字符序列被 GRUB 视为一个单元的单词。单词与单词之间以"元字符"作分隔，如下所述的哪些符号并且加上"空格符"，"制表符"与"新行符"：
{ } | & $; < >

Quoting may be used to include metacharacters in words; see below.
"引用"也许被来处理在单词中包含有"元字符"的情况下；请见下文。

Reserved words
保留单词(保留字)
==========
Reserved words have a special meaning to GRUB. The following words are recognised as reserved when unquoted and either the first word of a simple command or the third word of a 'for' command:
这些所保留的单词在 GRUB 里具有特殊意义。以下单词当未被括号所包含时是会被识别为保留字而不管它是否出现在由一个字母所组成的简单命令前还是由三个字母所组成的"for"命令里：
! [[ ]] { }
case do done elif else esac fi for function if in menuentry select then time until while

Not all of these reserved words have a useful purpose yet; some are reserved for future expansion.
这些保留单词并不是全部都是有作用的；有些只是为了张来的扩展而被预留。

Quoting
引用
=======

Quoting is used to remove the special meaning of certain characters or words. It can be used to treat

metacharacters as part of a word, to prevent reserved words from being recognised as such, and to prevent variable expansion.
引用的作用是用于移除元字符或单词的特殊含义。它能把元字符作为普通字符而成为单词的一部分，这有效防止了单词被误认为保留单词，并且防止意外地生成过多的变量而造成变量膨胀。

There are three quoting mechanisms: the escape character, single quotes, and double quotes.
这里有三种引用的作用机制：转义字符，单引号和双引号。

A non-quoted backslash (\) is the "escape character". It preserves the literal value of the next character that follows, with the exception of newline.
一个非引号内的反斜杠(\)是一个"转义字符(逃跑字符)"。它保留了紧跟其后的下一个字符的字面值也就是说把紧跟其后的任何字符(元字符)变成普通字符来输出，但除了换行符。

Enclosing characters in single quotes preserves the literal value of each character within the quotes. A single quote may not occur between single quotes, even when preceded by a backslash.
包含在单引号内的字符串将被保留了各自的文本值而成为普通字符。一个单引号内不能出现另一组单引号，即使在引号前加上反斜杠也不行。

Enclosing characters in double quotes preserves the literal value of all characters within the quotes, with the exception of '$' and '\'. The '$' character retains its special meaning within double quotes. The backslash retains its special meaning only when followed by one of the following characters: '$', '"', '\', or newline. A backslash-newline pair is treated as a line continuation (that is, it is removed from the input stream and effectively ignored(1) (*note Shell-like scripting-Footnote-1::)). A double quote may be quoted within double quotes by preceding it with a backslash.
包含在双引号内的字符串将被保留在引号内的所有字符串文本值，但"$" 和 "\" 除外(也就是说双引号内的"$"和"\"还能具有其自身的作用，而非以普通字符的方式被输出)。美元符"$" 在双引号内具有特别的意义。如需要以普通字符的形式输出需要利用反斜杠"\"并且在紧随其后加上下列字符之一："$" "," "\"或换行符。而当"反斜杠"后面紧跟一个"换行符"的时候会被当作是一个续行符。(也就是说，它是有效地忽视和删除了输入流)。而一个双引号可以通过反斜杠从而在其内引用另一个双引号。
例如：
```
line="OK\
_line2"
```
则会产生字符串"OK_line2"，续行符的作用只是让脚本的输出格式更美观。

Variable expansion
变量扩展
==========
The '$' character introduces variable expansion. The variable name to be expanded may be enclosed in braces, which are optional but serve to protect the variable to be expanded from characters immediately following it which could be interpreted as part of the name.
"$"字符用于变量扩展。被赋值的变量名字可以封闭在花括号内，然而这是可选的。这是为了保护变量的字符串赋值以免紧跟在其后的字符被误解为变量名字的一部分。
例如：
```
var = "_med_"
pre = "P"
suf = "S"
echo "$P${var}IL$suf"
```
输出
```
P_med_ILS
```
如果不使用花括号${var}IL，而是使用$varIL，那么系统会误会要搜索的扩展变量名为 varIL 而造成错误。

Normal variable names begin with an alphabetic character, followed by zero or more alphanumeric characters. These names refer to entries in the GRUB environment (*note Environment).
变量名字一般是以字母为开头，紧随为零个或多个字母的字符。这些名字会引用自 GRUB 环境里的记录。

Positional variable names consist of one or more digits. They represent parameters passed to function calls, with '$1' representing the first parameter, and so on.
位置的变量名字由一个或多个数字组成。它们代表为参数被传递到函数调用，用"$1"表示第一个参数，以此类推。

The special variable name '?' expands to the exit status of the most recently executed command. When positional variable names are active, other special variable names '@', '*' and '#' are defined and they expand to all positional parameters with necessary quoting, positional parameters without any quoting, and positional parameter count respectively.
特殊的变量名字"?"扩展到最近执行的命令的退出状态。当位置变量的名字为可用的时候，其他的特殊变量名字如"@"，"*" 和 "#" 它们分别扩展为："@"表示所有位置的变量参数并且带有双引号的引用符(例如会扩展成："$1" "$2" "$3" .....)，"*"表示所有位置的变量参数但不带双引号的引用符(例如会扩展成：$1 $2 $3 .....)，"#"表示所有有效的位置参数的个数。

Comments
注释
========
A word beginning with '#' causes that word and all remaining characters on that line to be ignored.
在一个单词的最前面加上"#"字符会导致这个词与这个词位于同一行上的后面的所有字符都会被忽视-注释掉。除了在字符串里或紧跟转义符后除外。

Simple commands
简单的命令
===============
A "simple command" is a sequence of words separated by spaces or tabs and terminated by a semicolon or a newline. The first word specifies the command to be executed. The remaining words are passed as arguments to the invoked command.
一个"简单命令"是由一组用空格或制表符作分隔并以与分号或换行符作为终止符的单词序列。第一个字词为指定要执行的命令。剩余的词句作为参数传递给被调用的命令。

The return value of a simple command is its exit status. If the reserved word '!' precedes the command, then the return value is instead the logical negation of the command's exit status.
简单命令的返回值为命令的退出状态。如果保留字"!"在命令之前，那么返回值是命令被逻辑反后的退出状态。

Compound commands
复合命令(复杂的命令)
=================
A "compound command" is one of the following:
for name in word ...; do list; done
The list of words following 'in' is expanded, generating a list of items. The variable NAME is set to each element of this list in turn, and LIST is executed each time. The return value is the exit status of the last command that executes. If the expansion of the items following 'in' results in an empty list, no commands are executed, and the return status is 0.
一个"复合命令"是下列之一(也就是指：选择语句，循环语句，流程控制语句等)：
for NAME in WORD ... ; do LIST ; done
跟在"in"后面的"WORD"列表会被扩展，生成一个项目列表。这个变量"NAME"将依次被设置到这个列表里的每个元素中，并且每次都执行一次"LIST"。而复合命令的返回值是最后一个命令执行结束后的这个最后命令的退出状态。如果跟在"in"后的这些项的扩展是一个空列表，表示没有任何命令被执行，那么返回状态为"0"。
也就是说：
NAME 是任意的变量名。WORD 是单词的列表。循环会在列表中迭代，每次循环 NAME 的值为列表中的某一元素。for 语句的退出状态为最后一个执行的命令的退出状态。如果 WORD 列表为空，那么循环一次也不

执行，退出状态为 0。

if LIST; then LIST; [elif LIST; then LIST;] ... [else LIST;] fi
The 'if' LIST is executed. If its exit status is zero, the 'then' LIST is executed. Otherwise, each 'elif' LIST is executed in turn, and if its exit status is zero, the corresponding 'then' LIST is executed and the command completes. Otherwise, the 'else' LIST is executed, if present. The exit status is the exit status of the last command executed, or zero if no condition tested true.
if LIST ; then LIST ; [elif LIST ; then LIST ;] ... [else LIST ;] fi
当那个"if" LIST 语句被执行。如果其退出状态为"0"("0"表示成功，非"0"表示不成功)，那么则会去执行"then"LIST 。否则，则会依次去执行属于其后的每一个"elif" LIST ，而当中如果"elif" LIST 的退出状态为"0"，那么则会接着执行其后的 "then" LIST 从而完整结束了这条执行语句。但如果退出结果不为"0"，则会直接跳去执行紧跟其后的那个"elif" LIST，直到最后还是返回非"0"则会直接执行到结尾的"else" LIST 并以其结果作为退出状态，当然前题是"else"是存在的。结束时为最后一个命令执行完后的退出状态或如果环境测试没有返回 true 则为"0"。

while COND; do LIST; done
until COND; do LIST; done
The 'while' command continuously executes the 'do' LIST as long as the last command in COND returns an exit status of zero. The 'until' command is identical to the 'while' command, except that the test is negated; the 'do' LIST is executed as long as the last command in COND returns a non-zero exit status. The exit status of the 'while' and 'until' commands is the exit status of the last 'do' LIST command executed, or zero if none was executed.
while COND; do LIST; done
until COND; do LIST; done
这个"while"命令将会不断重复地去执行那个"do" LIST，直到最后 COND 返回一个结束状态非"0"才会结束。"until"命令是等同于"while"命令，但它们唯一的区别在于循环条件的检测是相反的；"do"LIST 是不断别执行直到最后 COND 命令最后完成并返回一个"0"的退出状态。退出状态为"while"和"until"命令在最后一次执行完"do"LIST 命令时"do"LIST 所返回的状态，或者如果没有执行过任何部分则返回"0"。
也就是说：
while 命令里，当 COND 命令退出状态为 0 时，反复执行 LIST，直到 COND 退出状态不为 0。
until 命令里，当 COND 命令退出状态不为 0 时，反复执行 LIST，直到 COND 退出状态为 0。

function NAME { COMMAND; ... }
This defines a function named NAME. The "body" of the function is the list of commands within braces, each of which must be terminated with a semicolon or a newline. This list of commands will be executed whenever NAME is specified as the name of a simple command. Function definitions do not affect the exit status in '$?'. When executed, the exit status of a function is the exit status of the last command executed in the body.
function NAME { COMMAND; ... }
这里定义一个函数名字 NAME。函数的"主体(body)"为被包含在括号内的一系列命令，每个命令的结束都必须用分号或换行作为终止符。当 NAME 被定义好后可以作为一个简单命令名来被调用，调用后函数体内的命令列表将被执行一遍。自定义函数不影响"$?"里的退出状态。在执行的时候，一个函数的退出状态是主体里执行到最后一条命令的结束时的退出状态。
例如：
menuentry TITLE ['--class=class' ...] ['--users=users'] ['--unrestricted'] ['--hotkey=key'] { COMMAND; ... }
这条函数其实就是用于建立菜单选项的 grub.cfg 文本里的脚本代码。

Built-in Commands
内置命令
==========
Some built-in commands are also provided by GRUB script to help script writers perform actions that are otherwise not possible. For example, these include commands to jump out of a loop without fully completing it, etc.
GRUB 提供了一些内置的脚本命令去帮助脚本编写者执行其他方式不可能做到的动作。例如，这些包含在循

欢迎点击这里的链接进入精彩的Linux公社网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

Linux公社是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：**Ubuntu 专题** **Fedora 专题** **Android 专题** **Oracle 专题** **Hadoop 专题** **RedHat 专题** **SUSE 专题** **红旗 Linux 专题** **CentOS 专题**

Linux 公社微信公众号：linuxidc_com

环没有被完全执行完就让循环跳出的跳出命令，诸如此类等等。
可以说：内置命令是用来控制脚本流程的命令。

break ['n']
Exit from within a 'for', 'while', or 'until' loop. If 'n' is specified, break 'n' levels. 'n' must be greater than or equal to 1. If 'n' is greater than the number of enclosing loops, all enclosing loops are exited. The return value is 0 unless 'n' is not greater than or equal to 1.
break ['n']
从 break 位于其内的"for"，"while"，或"until"循环体中退出。如果参数"n"被指定值，则返回"n"级。"n"必须大于或等于 1。如果"n"大于其外的循环体数目，那代表所有的循环体都会被结束循环。最后它的返回值为 0 除非"n"为非大于或等于 1 的值。

continue ['n']
Resume the next iteration of the enclosing 'for', 'while' or 'until' loop. If 'n' is specified, resume at the 'n'th enclosing loop. 'n' must be greater than or equal to 1. If 'n' is greater than the number of enclosing loops, the last enclosing loop (the "top-level" loop) is resumed. The return value is 0 unless 'n' is not greater than or equal to 1.
continue ['n']
作用为从包含其内的"for"，"while"或"until"循环体中立刻开始下一次循环(也可以说：立刻开始循环体的下一轮迭代)。如果参数"n"被指定值，则从"n"层外的循环体去立刻开始下一次循环(迭代)。"n"必须大于或等于 1。如果"n"大于包含其循环体的嵌套层级数，则作用为对最外层的循环体去立刻开始下一次循环(为"顶级"循环)。它的返回值为"0"除非"n"的值非大于或等于 1。

return ['n']
Causes a function to exit with the return value specified by 'n'. If 'n' is omitted, the return status is that of the last command executed in the function body. If used outside a function the return status is false.
return ['n']
促使一个功能函数结束并且返回"n"所指定的值。如果忽略掉"n"，那返回状态为其函数体内的最后一条命令执行完成结束后的返回状态。假如使用在函数的外部则其返回状态为 false。
也就是说 return 是用在函数体内的，如果位于函数体的那对大括号外，则会返回 false。而正常情况下 return 会结束掉包含其内的功能函数体并把所指定的"n"值返回到上一级调用其的函数，如果忽略掉了"n"的参数，则会返回最后一条命令执行完成结束后的返回状态。

shift ['n']
The positional parameters from 'n'+1 ... are renamed to '$1'.... Parameters represented by the numbers '$#' down to '$#'-'n'+1 are unset. 'n' must be a non-negative number less than or equal to '$#'. If 'n' is 0, no parameters are changed. If 'n' is not given, it is assumed to be 1. If 'n' is greater than '$#', the positional parameters are not changed. The return status is greater than zero if 'n' is greater than '$#' or less than zero; otherwise 0.
shift ['n']
位置参数从所对应的"n"+1... 被重命名为"$1"... 直到参数所代表的数字 "$#" 往下直到 "$#" - "n" + 1 都会被设置。"n"必需为小于或等于"$#"的非负整数。如果"n"为 0，则没有参数被改变。如果"n"没给出，那它被假定为 1。如果"n"大于那个"$#"，那位置参数则不被改变。如果"n"是大于那个"$#"则其返回状态大于零或则小于零；否则为 0。
也就是说：
位置参数从"n"+1 被重命名为"$1"，当 n=0 时被赋为$1，当 n=1 时被赋为$2，也就是 a0=>b1，a1=>b2，a2=>b3，所有的数据都给更新到后一位，这种做法能够实现数据的从尾向前的排列方式去存放。
也就是说"shift"把所有的元素都向后移动"n"位。而"n"必需为大于或等于 0 并且小于或等于元素的最大个数"$#"。如果"n"为 0 则不作任何操作。如果"n"没有给出则被假定为 1。如果"n"大于"$#"那么位置参数则不会改变，并且会返回一个非 0 的状态返回值。如果正常执行结束则会返回状态返回值 0。

(1) Currently a backslash-newline pair within a variable name is not handled properly, so use this feature with some care.

(1) 目前在一对 反斜杠-换行 内定义变量名字是处理不当的，所以使用这个特征会有一些好处。

5.3 Multi-boot manual config
5.3 多重启动的手动配置

==========

Currently autogenerating config files for multi-boot environments depends on os-prober and has several shortcomings. While fixing it is scheduled for the next release, meanwhile you can make use of the power of GRUB syntax and do it yourself. A possible configuration is detailed here, feel free to adjust to your needs.
一般地多重-引导环境会依赖于 OS-探测器而自动生成配置文件虽然其有点缺点。而在下个发行版本的修复计划，你可以运用你的知识与力量去编写你自己的 GRUB 语法语句。一个详细的合理配置，感到自由地、免费的、解放...(free 思想之意)地适合着你的需要。

First create a separate GRUB partition, big enough to hold GRUB. Some of the following entries show how to load OS installer images from this same partition, for that you obviously need to make the partition large enough to hold those images as well. Mount this partition on/mnt/boot and disable GRUB in all OSes and manually install self-compiled latest GRUB with:
首先创建一个单独的 GRUB 分区，大得足以容纳 GRUB。以下的技术示范项目为从同一个硬盘里的同一个分区里去加载 OS 安装镜像，显然你的分区需要大得足够容纳这些镜像。把这个分区挂载到 /mnt/boot 接着在你的操作系统里删除掉就版本的 GRUB 并且手动去 自主-编译 安装最新版本的 GRUB：

```
'grub-install --boot-directory=/mnt/boot /dev/sda'
```

In all the OSes install GRUB tools but disable installing GRUB in bootsector, so you'll have menu.lst and grub.cfg available for use. Also disable os-prober use by setting:
在你的操作系统里使用 GRUB 工具去安装的时候禁止把 GRUB 安装到引导扇区，因为你必需得到有效的"menu.lst"或者"grub.cfg"文件以供使用。还要关闭使用"os-prober"程序通过以下设置：

```
'GRUB_DISABLE_OS_PROBER=true'
```

in /etc/default/grub Then write a grub.cfg (/mnt/boot/grub/grub.cfg):
接着在 /etc/default/grub 里的那个 grub.cfg 文件 里写入(/mnt/boot/grub/grub.cfg)：

```
menuentry "OS using grub2" {
  insmod xfs
  search --set=root --label OS1 --hint hd0,msdos8
  configfile /boot/grub/grub.cfg
}

menuentry "OS using grub2-legacy" {
  insmod ext2
  search --set=root --label OS2 --hint hd0,msdos6
  legacy_configfile /boot/grub/menu.lst
}

menuentry "Windows XP" {
  insmod ntfs
  search --set=root --label WINDOWS_XP --hint hd0,msdos1
  ntldr /ntldr
}

menuentry "Windows 7" {
  insmod ntfs
  search --set=root --label WINDOWS_7 --hint hd0,msdos2
  ntldr /bootmgr
```

```
}

menuentry "FreeBSD" {
   insmod zfs
   search --set=root --label freepool --hint hd0,msdos7
   kfreebsd /freebsd@/boot/kernel/kernel
   kfreebsd_module_elf /freebsd@/boot/kernel/opensolaris.ko
   kfreebsd_module_elf /freebsd@/boot/kernel/zfs.ko
   kfreebsd_module /freebsd@/boot/zfs/zpool.cache type=/boot/zfs/zpool.cache
   set kFreeBSD.vfs.root.mountfrom=zfs:freepool/freebsd
   set kFreeBSD.hw.psm.synaptics_support=1
}

menuentry "experimental GRUB" {
   search --set=root --label GRUB --hint hd0,msdos5
   multiboot /experimental/grub/i386-pc/core.img
}

menuentry "Fedora 16 installer" {
   search --set=root --label GRUB --hint hd0,msdos5
   linux /fedora/vmlinuz lang=en_US keymap=sg resolution=1280x800
   initrd /fedora/initrd.img
}

menuentry "Fedora rawhide installer" {
   search --set=root --label GRUB --hint hd0,msdos5
   linux /fedora/vmlinuz repo=ftp://mirror.switch.ch/mirror/fedora/linux/development/rawhide/x86_64
lang=en_US keymap=sg resolution=1280x800
   initrd /fedora/initrd.img
}

menuentry "Debian sid installer" {
   search --set=root --label GRUB --hint hd0,msdos5
   linux /debian/dists/sid/main/installer-amd64/current/images/hd-media/vmlinuz
   initrd /debian/dists/sid/main/installer-amd64/current/images/hd-media/initrd.gz
}
```

Notes:
* Argument to search after -label is FS LABEL. You can also use UUIDs with -fs-uuid UUID instead of -label LABEL. You could also use direct 'root=hd0,msdosX' but this is not recommened due to device name instability.
注释：
* -label 会去搜索位于 FS LABEL 后面的参数值。你也可以用 UUIDs 以 -fs-uuid UUID 的方式去代替掉 -label LABEL，你还可以直接地使用 "root=hd0,msdosX"指令去指定使用分区但这是不推介的因为设备名并不稳定。

5.4 Embedding a configuration file into GRUB

5.4 将配置文件嵌入到 GRUB

GRUB supports embedding a configuration file directly into the core image, so that it is loaded before entering normal mode. This is useful, for example, when it is not straightforward to find the real configuration file, or when you need to debug problems with loading that file. 'grub-install' uses this feature when it is not using BIOS disk functions or when installing to a different disk from the one containing '/boot/grub', in which case it needs to use the 'search' command (*note search) to find '/boot/grub'.
GRUB 支持把配置文件直接地嵌入到"core"核心镜像，以便它在进入正常模式之前被加载。这样做是有好处的，例如，它要去找到那个真正的配置文件已经并不简单，或当你需要去调试那个需要加载的配置文件上的问题的时候。当"grub-install"在不使用 BIOS 磁盘功能的时候或则当" /boot/grub"目录被安装到不同的磁盘里的时候"grub-install"程序会自动默认使用嵌入配置文件到内核的功能，在这种情况下需要需要使用到"search"命令去搜索出 " /boot/grub"目录去实现自动设置根目录到正确的磁盘。

To embed a configuration file, use the '-c' option to 'grub-mkimage'. The file is copied into the core image, so it may reside anywhere on the file system, and may be removed after running 'grub-mkimage'.
要去嵌入一个配置文件，需要在"grub-mkimage"程序后面使用到"-c"参数选项。它会把配置文件复制到核心镜像里，因此它能够驻留在文件系统里的任何位置，并且可能在"grub-mkimage"程序运行之后都不会被删除覆盖。

After the embedded configuration file (if any) is executed, GRUB will load the 'normal' module (*note normal::), which will then read the real configuration file from '$prefix/grub.cfg'. By this point, the 'root' variable will also have been set to the root device name. For example, 'prefix' might be set to '(hd0,1)/boot/grub', and 'root' might be set to 'hd0,1'. Thus, in most cases, the embedded configuration file only needs to set the 'prefix' and 'root' variables, and then drop through to GRUB's normal processing. A typical example of this might look like this:
在执行了嵌入配置文件之后(如果有)，GRUB 将加载"normal"模块，它将会读取来自"$prefix/grub.cfg"的真正的配置文件。但与此同时，这个"root"变量也必需被设置成根设备的名字。比如，"prefix"如果被设置为"(hd0,1)/boot/grub"，那么"root"可能会被设置为"hd0,1"。因此，在大部分情况下，设置嵌入配置文件只需要设置"prefix"变量和"root"变量，接着向下递交到 GRUB 的"normal"模块进程作处理就可以了。一个典型的设置例子看起来像这样：

```
search.fs_uuid 01234567-89ab-cdef-0123-456789abcdef root
set prefix=($root)/boot/grub
```

(The 'search_fs_uuid' module must be included in the core image for this example to work.)
("search_fs_uuid"模块必须包含到核心镜像并且在此例子中能够正常工作)

In more complex
cases, it may be useful to read other configuration files directly from the embedded configuration file. This allows such things as reading files not called 'grub.cfg', or reading files from a directory other than that where GRUB's loadable modules are installed. To do this, include the 'configfile' and 'normal' modules in the core image, and embed a configuration file that uses the 'configfile' command to load another file. The following example of this also requires the 'echo', 'search_label', and 'test' modules to be included in the core image:
在更复杂的情况下，它可能直接从嵌入的配置文件中去读取其他有用的配置文件。这样可以允许那些被读文件不以"grub.cfg"为名，或从 GRUB 的可加载模块目录以外的一个目录中读取文件。要做到这些，必需把"configfile"模块和"normal"模块包含到"core"核心镜像里，并且在被嵌入的那个配置文件里使用"configfile"命令去加载另一个所需的配置文件。下列例子还需要用到 "echo"，"search_label"，与 "test" 模块所以必需把它们一并包含到核心镜像里：

```
search.fs_label grub root
  if [ -e /boot/grub/example/test1.cfg ]; then
```

```
    set prefix=($root)/boot/grub
    configfile /boot/grub/example/test1.cfg
  else
    if [ -e /boot/grub/example/test2.cfg ]; then
      set prefix=($root)/boot/grub
      configfile /boot/grub/example/test2.cfg
    else
      echo "Could not find an example configuration file!"
    fi
  fi
```

The embedded configuration file may not contain menu entries directly, but may only read them from elsewhere using 'configfile'.
因此，被嵌入的配置文件里可能不会直接去包含所需的菜单项目，而是可能通过使用"configfile"命令来读取存放于其他地方的配置文件来实现菜单项目的显示。

06.主题文件格式-Theme file format

06.主题文件格式-Theme file format
********** **********

6 Theme file format
6 主题文件格式
==========

6.1 Introduction
6.1 简介

The GRUB graphical menu supports themes that can customize the layout and appearance of the GRUB boot menu. The theme is configured through a plain text file that specifies the layout of the various GUI components (including the boot menu, timeout progress bar, and text messages) as well as the appearance using colors, fonts, and images. Example is available in docs/example_theme.txt

GRUB 图像菜单支持主题因此可以自定义 GRUB 引导菜单的布局和外观。主题是通过一个纯文本文件指定配置各种 GUI 组件的布局(包括启动菜单，超时进度条，和文本信息)以及外观使用的颜色，字体，和图像，位于 docs/example_theme.txt 里有有效的文档示例。

6.2 Theme Elements
6.2 主题元素
==========

6.2.1 Colors
6.2.1 颜色

Colors can be specified in several ways:

颜色能以以下的几种方法来被定义：

* HTML-style "#RRGGBB" or "#RGB" format, where *R*, *G*, and *B* are hexadecimal digits (e.g., "#8899FF")

* 以 HTML 的形式来定义。"#RRGGBB" 或 "#RGB" 格式，其中 *R*，*G*，和 *B* 为十六进制数字(比如："#8899FF")

* as comma-separated decimal RGB values (e.g., "128, 128, 255")

* 以逗号作分隔的十进制 RGB 值。(比如："128,128,255")

* with "SVG 1.0 color names" (e.g., "cornflowerblue") which must be specified in lowercase.

* 支持 "SVG 1.0 颜色名称" (比如："cornflowerblue"-雏菊花蓝色) 并且必须都为小写。

6.2.2 Fonts
6.2.2 字体

The fonts GRUB uses "PFF2 font format" bitmap fonts. Fonts are specified with full font names. Currently there is no provision for a preference list of fonts, or deriving one font from another. Fonts are loaded with the "loadfont" command in GRUB. To see the list of loaded fonts, execute the "lsfonts" command. If there are too many fonts to fit on screen, do "set pager=1" before executing "lsfonts".

GRUB 的字体使用"PFF2 font format"的点阵图字体格式。字体通过指定完整的字体名称来使用。目前并不规定首选的字体列表，或者其他以外的字体。字体的加载需要用到 GRUB 里的"loadfont"命令。要查看已装载了的字体列表，可以执行"lsfonts"命令。如果列出太多字体以至于满出屏幕，则需要在执行"lsfonts"命名之前执行一次"set pager=1"命令。

例子：

grub-mkfont --verbose --range=0x0-0x7F --size=18 --output=DejaVuSans-18.pf2 $(locate DejaViewSans.ttf)

仿造上面的命令生成 DejaVuSansMono 字体，在终端下支持中文你还需要其他中文字体，比如 WenQuanYi。

6.2.3 Progress Bar
6.2.3 进度条

Figure 6.1
图 **6.1(**可能是例图，可惜并未提供。。。**)**

Figure 6.2
图 **6.2**
Progress bars are used to display the remaining time before GRUB boots the default menu entry. To create a progress bar that will display the remaining time before automatic boot, simply create a "progress_bar" component with the id "__timeout__". This indicates to GRUB that the progress bar should be updated as time passes, and it should be made invisible if the countdown to automatic boot is interrupted by the user.
进度条是用来显示在 GRUB 去启动默认菜单选项之前的剩余时间。创建一个进度条这将在自动启动之前显示出一个剩余时间，它将会简单地创建出一个带有 ID 标识为"_timeout_"属性的"progress_bar"组件。这表明 GRUB 的进度条应该是随着时间流逝而更新的，并且它应该能够自动启动并开始倒计时，除非在此过程之中被用户人为中断后自动关闭倒计时并且取消显示。

Progress bars may optionally have text displayed on them. This text is controlled by variable "text" which contains a printf template with the only argument %d is the number of seconds remaining. Additionally special values "@TIMEOUT_NOTIFICATION_SHORT@", "@TIMEOUT_NOTIFICATION_MIDDLE@", "@TIMEOUT_NOTIFICATION_LONG@" are replaced with standard and translated templates.
进度条可以选择为文本显示方式。这个文本是由控制变量"text"与包含一个 printf 函数给出唯一的参数%d 来表示剩余的秒数。此外特殊的值有 "@TIMEOUT_NOTIFICATION_SHORT@"，"@TIMEOUT_NOTIFICATION_MIDDLE@"，"@TIMEOUT_NOTIFICATION_LONG@"被替换为标准与翻译模版。
可以把进度条设置为透明，再通过"show text"属性来控制是否显示倒计时文本实现进度条以文本的方式来显示，而这个"show text"的属性可以被设置为"true"或者"false"。而"true"则为开启显示时间倒计时。

6.2.4 Circular Progress Indicator
6.2.4 圆形的进度指示器

The circular progress indicator functions similarly to the progress bar. When given an id of "__timeout__", GRUB updates the circular progress indicator's value to indicate the time remaining. For the circular progress indicator, there are two images used to render it: the *center* image, and the *tick* image. The center image is rendered in the center of the component, while the tick image is used to render each mark along the circumference of the indicator.
圆形的进度指示器功能类似于进度条。当给出一个"_timeout_"的 ID 标识时，GRUB 会去更新圆形进度指示器的值从而表示剩余时间。对于圆形进度指示器，需要用到两个图像去渲染它：一个"center"图像，和一个"tick"图像。那个 center 图像是呈现在中心位置的组件，然而那个 tick 图像是用于按照着那圆周的时间标号去呈现出每一个时间标号。
也就是说：
表盘进度条指示器类似于进度条，当设置了"timeout"的 ID 值时，GRUB 会以此值来作为倒计时时限。而对于表盘来说则需要渲染两个图像组件去实现：一个"center"图像与一个"tick"图像，那个"center"用来绘制表盘中心，而那个"tick"则用来绘制表周的刻度。

6.2.5 Labels
6.2.5 标签

Text labels can be placed on the boot screen. The font, color, and horizontal alignment can be specified for labels. If a label is given the id "__timeout__", then the "text" property for that label is also updated with a message informing the user of the number of seconds remaining until automatic boot. This is useful in case you want the text displayed somewhere else instead of directly on the progress bar.

标签的显示文本能被放置到引导界面的任何位置。在标签里可以设置文本所使用的字体，颜色，与水平对齐属性等等参数。如果标签的文本值被设置为_timeout_"的 ID 标识，那么这个标签的"text"属性也会更新于一个通知用户所剩秒数的文本消息，直到时间结束并自动启动为止。这有利于以防你在实际情况中想要在别的地方显示文本，而不是直接显示在进度条上。

6.2.6 Boot Menu
6.2.6 启动菜单

The boot menu where GRUB displays the menu entries from the "grub.cfg" file. It is a list of items, where each item has a title and an optional icon. The icon is selected based on the *classes* specified for the menu entry. If there is a PNG file named "myclass.png" in the "grub/themes/icons" directory, it will be displayed for items which have the class *myclass*. The boot menu can be customized in several ways, such as the font and color used for the menu entry title, and by specifying styled boxes for the menu itself and for the selected item highlight.
GRUB 在引导菜单中展示出的菜单项目是取自于"grub.cfg"文件里的菜单条目。他是一个项目形式的列表，每个项目都具有一个标题和一个可选的图标。那个图标的使用是需要在菜单项目里指定"classes"参数来实现的。例如这里有一个命名为"myclass.png"的 PNG 文件放于 "grub/themes/icons"目录里，那这个图标将被所有指定了"classes"参数的值为"myclass"的菜单条目所显示。实现启动菜单的自定义有很多种方式，例如可以指定菜单项目标题所使用的字体或字体颜色，以及可以通过指定样式框架来套用样式到菜单自身并且实现菜单项目被选择时所使用的高亮样式。

6.2.7 Styled Boxes
6.2.7 样式框架

One of the most important features for customizing the layout is the use of *styled boxes*. A styled box is composed of 9 rectangular (and potentially empty) regions, which are used to seamlessly draw the styled box on screen:
自定义布局最重要的特征之一是使用"样式框"。一个样式框是由 9 个矩形区域所组成(可能某些矩形区域为空)，用于在屏幕上无缝地绘画出样式框：

Northwest (nw)	**North (n)**	**Northeast (ne)**
西北	北	东北
West (w)	**Center (c)**	**East (e)**
西	中	东
Southwest (sw)	**South (s)**	**Southeast (se)**
西南	南	东南

To support any size of box on screen, the center slice and the slices for the top, bottom, and sides are all scaled to the correct size for the component on screen, using the following rules:
为了在屏幕里支持任何尺寸的框架。中间的切片部分会与切片的顶部，底部和两边的切片按照比例缩小到正确尺寸去成为组成屏幕的组件，下列为使用规则：

1. The edge slices (north, south, east, and west) are scaled in the direction of the edge they are adjacent to. For instance, the west slice is scaled vertically.
2. The corner slices (northwest, northeast, southeast, and southwest) are not scaled.
3. The center slice is scaled to fill the remaining space in the middle.

1.切片边缘(东，南，西，和北)是按比例拉伸缩放到附近的边缘。例如，西切片是按比例垂直拉伸缩放的，而南切片则按照比例水平拉伸缩放。
2.在拐角处的切片(西北，东北，东南和西南)并没有规模的缩放模式，因此它们不被缩放。
3.在中间处的切片按比例缩放并填充到中间的剩余空间里去。

As an example of how an image might be sliced up, consider the styled box used for a terminal view.
例如一个图像如何地切分，需要考虑到此切片样式框架能否适用到终端视图模式。

Figure 6.3
图 **6.3**

6.2.8 Creating Styled Box Images
6.2.8 创建样式框架的图像

The Inkscape_ scalable vector graphics editor is a very useful tool for creating styled box images. One process that works well for slicing a drawing into the necessary image slices is:
在 Inkscape_scalable(向量绘图_可扩展) 矢量图编辑器是一个非常有用的工具适合于创建样式框架所使用图像。实现切片一个绘图成为所需图像切片的一种流程：

1. Create or open the drawing you'd like use.
1. 创建或打开一个你想使用的绘图。
2. Create a new layer on the top of the layer stack. Make it visible. Select this layer as the current layer.
2. 创建一个新层于堆叠层的顶部。使其可见。选择这个层作为当前层。
3. Draw 9 rectangles on your drawing where you'd like the slices to be. Clear the fill option, and set the stroke to 1 pixel wide solid stroke. The corners of the slices must meet precisely; if it is off by a single pixel, it will probably be evident when the styled box is rendered in the GRUB menu. You should probably go to File | Document Properties | Grids and enable a grid or create a guide (click on one of the rulers next to the drawing and drag over the drawing; release the mouse button to place the guide) to help place the rectangles precisely.
3. 在你希望切片出现的绘图里绘制出 9 个矩形。清除矩形框里的填充选项，并设置笔画为 1 像素宽的实线。在位于四个角的切片大小必须恰到好处；假如仅仅偏差了一个像素，它可能都会在样式框架套用到 GRUB 菜单样式里的时候被很明显地呈现。所以你应该到<文件>里的<文档属性>选上<坐标>选项，接着从生成出的那个坐标系里指定坐标或创建一个坐标引导去帮助矩形能够精确地被放置(点击绘画视图中的标尺之后拖拽到图画上；到需要坐标引导的位置上释放掉鼠标按钮)。
4. Right click on the center slice rectangle and choose Object Properties. Change the "Id" to "slice_c" and click Set. Repeat this for the remaining 8 rectangles, giving them Id values of "slice_n", "slice_ne", "slice_e", and so on according to the location.
4. 右键点击中间切片的矩形并且选择<对象属性>。改变其"ID"为"slice_c"并点击<应用>。以这样的方式操作完剩下的 8 个矩形，给予它们的 ID 值为"slice_n"，"slice_ne"，"slice_e"，与相应的位置一一对应。
5. Save the drawing.
5. 保存绘图。
6. Select all the slice rectangles. With the slice layer selected,you can simply press Ctrl+A to select all rectangles. The status bar should indicate that 9 rectangles are selected.
6. 选上所有的矩形切片。切片的层被选上后，你能够简单地按 Ctrl+A 去选上所有矩形。状态栏里应该提示出有 9 个矩形被选上。
7. Click the layer hide icon for the slice layer in the layer palette.The rectangles will remain selected, even though they are hidden.
7. 在<调色板>里点击图层隐藏图标去隐藏切片所在的那个图层。那些矩形将仍然保持着被选中的状态，即使它们被隐藏了。
8. Choose File | Export Bitmap and check the *Batch export 9 selected objects* box. Make sure that *Hide all except selected* is unchecked. click *Export*. This will create PNG files in the same directory as the drawing, named after the slices. These can now be used for a styled box in a GRUB theme.
8. 选择<文件>里的<输出位图>接着勾上"Batch export 9 selected objects"功能块。确保那"Hide all except selected"处于"unchecked"关闭检查的状态。这将在绘图所在目录中创建出以切片名来命名的 PNG 格式文件，例如命名为 slices。至此这些图片可以直接使用到 GRUB 主题的样式框架里了。

6.3 Theme File Manual
6.3 主题文件手册
==========
The theme file is a plain text file. Lines that begin with "#" are ignored and considered comments. (Note: This may not be the case if the previous line ended where a value was expected.)
The theme file contains two types of statements:
主题文件是一个纯文本文件。以"#"作为开头的行将作为注释而被忽略掉。(注：这可能并非如此如果上一行结束自一个预期值)
主题文件包含两种类型的声明：

1. Global properties.

1. 全局属性。
2. Component construction.
2. 组件结构。

6.3.1 Global Properties
6.3.1 全局属性

6.3.2 Format http://note.youdao.com/signIn/index.html?callback=http%3A%2F%2Fnote.youdao.com%2Fgroup
6.3.2 格式

Global properties are specified with the simple format:
全局属性以简单格式来指定：

name1:value1
变量名 1:值 1
name2:"value which may contain spaces"
变量名 2:"字符串"
name3:#88F
变量名 3:参数值

In this example, name3 is assigned a color value.
在这个例子中，name3 被分配了一个颜色值。

6.3.3 Global Property List
6.3.3 全局属性列表
---

title-text	Specifies the text to display at the top center of the screen as a title.
title-text	把指定的文本作为标题去显示到屏幕顶部的居中位置。
title-font	Defines the font used for the title message at the top of the screen.
title-font	规定了在屏幕顶部的标题信息所用的字体。
title-color	Defines the color of the title message.
title-color	规定了标题信息的颜色。
message-font	Defines the font used for messages, such as when GRUB is unable to automatically boot an entry.
message-font	规定了提示信息所用的字体，例如当 GRUB 无法自动加载某个菜单项目时所显示出的提示信息。
message-color	Defines the color of the message text.
message-color	规定了提示信息的文本颜色。
message-bg-color	Defines the background color of the message text area.
message-bg-color	规定了提示信息的文本区域里的背景颜色。
desktop-image	Specifies the image to use as the background. It will be scaled to fit the screen size.
desktop-image	指定背景所使用的图片。它将被缩放以适合屏幕大小。
desktop-color	Specifies the color for the background if *desktop-image* is not specified.
desktop-color	指定了背景所使用的颜色，前提是"desktop-image"为未被设置的状态。
terminal-box	Specifies the file name pattern for the styled box slices used for the command line terminal window.
terminal-box	把指定文件名的图像切片作为样式框架使用到命令行终端窗口中的模式。

For example,"terminal-box: terminal_*.png" will use the images "terminal_c.png" as the center area, "terminal_n.png" as the north (top) edge, "terminal_nw.png" as the northwest (upper left) corner, and so on. If the image for any slice is not found, it will simply be left empty.
例如，"terminal-box:terminal_*.png"将会使用图像"terminal_c.png"到中间区域，"terminal_n.png"为北方(顶部)的边缘位置，"terminal_nw.png"为西北方(左上角)角落，以此类推。如果某一部分的图片没有找到，则它的位置将会留空。

6.3.4 Component Construction

6.3.4 组件化结构

Greater customizability comes is provided by components. A tree of components forms the user interface. *Containers* are components that can contain other components, and there is always a single root component which is an instance of a *canvas* container.
组件提供了更灵活的界面定制。一个树状的组件结构构成了用户接口。"Containers"组件是可以包含其他组件的组件，并且始终源自一个"root"根组件并作为"canvas"组件的一个实例。

Components are created in the theme file by prefixing the type of component with a '+' sign:
组件的创建通过在主题文件里把组件用一个前缀符号"+"放在一起的形式实现：

```
' + label { text="GRUB" font="aqui 11" color="#8FF" } '
```

properties of a component are specified as "name = value" (whitespace surrounding tokens is optional and is ignored) where *value* may be:
* a single word (e.g., "align = center", "color = #FF8080"),
* a quoted string (e.g., "text = "Hello, World!""), or
* a tuple (e.g., "preferred_size = (120, 80)").
一个组件的属性规定为"name = value"的形式组成(前后加空格符是可选的并且它会被忽视掉)，而那个"value"可以为：
*一个单词 (例如："align = center"，"color = #FF8080")
*一个双引号所引用的字符串 (例如："text="Hello,World!"")，或
*一个数据单元组 (例如："preferred_size = (120,80)")。

6.3.5 Component List
6.3.5 组件列表

The following is a list of the components and the properties they support.
以下列出组件与它们所支持的属性。

*** label**

A label displays a line of text.
一个"label"标签用以显示一个文本行。
Properties:
属性：

text	The text to display.
text	需要被显示的文本。
font	The font to use for text display.
font	显示文本所使用的字体。
color	The color of the text.
color	文本所使用的颜色。
align	The horizontal alignment of the text within the component. Options are "left", "center", and "right".
align	文本在组件内的靠齐方向。选项有"left"左，"center"中，和"right"右。

*** image**

A component that displays an image. The image is scaled to fit the component, although the preferred size defaults to the image's original size unless the "preferred_size" property is explicitly set.
一个显示出一个图像的组件。图像被缩放以适合组件，而已首选大小默认为图像的原始大小除非"preferred_size"属性被显式地设置。
Properties:
属性：

file	The full path to the image file to load.
file	载入图像文件的完整路径。

*** progress_bar**

Displays a horizontally oriented progress bar. It can be rendered using simple solid filled rectangles, or

using a pair of pixmap styled boxes.
显示出一个水平的进度条。它能够使用简单的实线作填充描绘出一个矩形，或者使用一对点阵图作样式框架。
Properties:
属性：
fg_color The foreground color for plain solid color rendering.
fg_color 前景颜色被简单地纯色渲染。
bg_color The background color for plain solid color rendering.
bg_color 背景颜色被简单地纯色渲染。
border_color The border color for plain solid color rendering.
border_color 边框颜色被简单地纯色渲染。
text_color The text color.
text_color 文本的颜色。
show_text Boolean value indicating whether or not text should be displayed on the progress bar. If set to *false*, then no text will be displayed on the bar. If set to any other value, text will be displayed on the bar.
show_text 其值为布尔类型表明是否把文本显示在进度条。如果设置为*false*，将没有文本被显示到进度条上。如果设置为任何的其他值，文本将被显示到进度条里。
bar_style The styled box specification for the frame of the progress bar. Example: "progress_frame_*.png" highlight_styleThe styled box specification for the highlighted region of the progress bar. This box will be used to paint just the highlighted region of the bar, and will be increased in size as the bar nears completion. Example: "progress_hl_*.png".
bar_style 样式框架被指定为进度条所使用的结构。例子："progress_frame_*.png"高亮_样式的样式框架被指定到进度条成为高亮区域。者框架将被使用到描绘刚才进度条的高亮区域，并且将被增加在接近完成的进度条大小。例子："progress_hl_*.png"。
text The text to display on the progress bar. If the progress bar's ID is set to "__timeout__", then GRUB will updated this property with an informative message as the timeout approaches.
text 文本显示到进度条里。如果进度条里的"_timeout_"ID 项被设置。那么 GRUB 将上载这个属性值到窗口信息并作倒计时处理。
value The progress bar current value. Normally not set manually.
value 进度条当前的值。通常不用手动设置。
start The progress bar start value. Normally not set manually.
start 进度条的开始值。通常不用手动设置。
end The progress bar end value. Normally not set manually.
end 进度条的结束值。通常不用手动设置。

*** circular_progress**

Displays a circular progress indicator. The appearance of this component is determined by two images: the *center* image and the *tick* image. The center image is generally larger and will be drawn in the center of the component. Around the circumference of a circle within the component, the tick image will be drawn a certain number of times, depending on the properties of the component.
显示一个圆形的进度指示器。它的外观组成由取决于两个图像：一个为*center*图像和另一个为*tick*的图像。这个居中(center)图像通常是比较大的并且将被绘画到居中位置成为一个组件。圆形的周长之内为一个组件，这个标记(tick)图像将被绘画成相应的时间号码，这取决于组件的属性。
Properties:
属性：
center_bitmap The file name of the image to draw in the center of the component.
center_bitmap 这文件名的图像被绘画在组件的居中位置。
tick_bitmap The file name of the image to draw for the tick marks.
tick_bitmap 这文件名的图像被绘画成刻度线。
num_ticks The number of ticks that make up a full circle.
num_ticks 布满整个圆周的刻度的数目。
ticks_disappear Boolean value indicating whether tick marks should progressively appear, or progressively disappear as *value* approaches *end*. Specify "true" or "false".
ticks_disappear 为布尔的值，表明刻度线是否应该逐步显现，或者随着*value*接近*end*逐步消失。值

为*true*或*false*。
value The progress indicator current value.Normally not set manually.
value 进度指示器的当前值。通常不用手工设置。
start The progress indicator start value.Normally not set manually.
start 进度指示器的开始值。通常不用手工设置。
end The progress indicator end value.Normally not set manually.
end 进度指示器的结束值。通常不用手工设置。

*** boot_menu**

Displays the GRUB boot menu. It allows selecting items and executing them.
显示 GRUB 的启动菜单。它允许选择项目并执行它。
Properties:
属性：
item_font The font to use for the menu item titles.
item_font 菜单项目标题所使用的字体。
selected_item_font The font to use for the selected menu item, or "inherit" (the default) to use "item_font" for the selected menu item as well.
selected_item_font 菜单项目被选上时所使用的字体，或者"inherit"(继承-这是默认值)去使用"item_font"里设置的字体为菜单项目被选中时的字体。
item_color The color to use for the menu item titles.
item_color 菜单项目标题所使用的颜色。
selected_item_color The color to use for the selected menu item, or "inherit" (the default) to use "item_color" for the selected menu item as well.
selected_item_color 菜单项目被选上时所使用的颜色，或者"inherit"(为默认)去使用"item_color"里设置的颜色为菜单项目被选中时的颜色。
icon_width The width of menu item icons.Icons are scaled to the specified size.
icon_width 菜单项目图标的宽度。图标被按比例缩放到指定大小。
icon_height The height of menu item icons.
icon_height 菜单项目图标的高度。
item_height The height of each menu item in pixels.
item_height 每个菜单项目的高度以像素为单位。
item_padding The amount of space in pixels to leave on each side of the menu item contents.
item_padding 菜单项目的内容距离两边的间距，以像素为单位。
item_icon_space The space between an item's icon and the title text, in pixels.
item_icon_space 项目图标和标题文本之间的间隔，按照像素为单位。
item_spacing The amount of space to leave between menu items, in pixels.
item_spacing 菜单项目之间的总间距量，以像素单位。
menu_pixmap_style The image file pattern for the menu frame styled box. Example: "menu_*.png" (this will use images such as "menu_c.png","menu_w.png", 'menu_nw.png", etc.)
menu_pixmap_style 菜单边框的样式框架的图像模式。例如："menu_*.png"(这将使用图像如"menu_c.png"，"menu_w.png"，"menu_nw.png"，等等。)
selected_item_pixmap_style The image file pattern for the selected item highlight styled box.
selected_item_pixmap_style 被选中菜单的高亮样式框架的图像模式。
scrollbar Boolean value indicating whether the scroll bar should be drawn if the frame and thumb styled boxes are configured.
scrollbar 值为布尔值，如果边框与翻页的样式框架被配置了，滚动条是否应该被画出。
scrollbar_frame The image file pattern for the entire scroll bar. Example: "scrollbar_*.png"
scrollbar_frame 用于全部的滚动条的图像文件的模式。例如："scrollbar_*.png"
scrollbar_thumb The image file pattern for the scroll bar thumb (the part of the scroll bar that moves as scrolling occurs). Example:"scrollbar_thumb_*.png"
scrollbar_thumb 用于滚动条翻阅的图像文件的模式。(当滚动条发生滚动时的那部分)。例如："scrollbar_thumb_*.png"
max_items_shown The maximum number of items to show on the menu. If there are more than *max_items_shown* items in the menu, the list will scroll to make all items accessible.

max_items_shown　　在菜单里显示项目的最大数目。如果有超过*max_items_shown*值的菜单条目，将滚动列表将使所有的项目可被访问。

*** canvas**
Canvas is a container that allows manual placement of components within it. It does not alter the positions of its child components. It assigns all child components their preferred sizes.
画布作为一个容器，允许手工放置那些组件于其之内。它不会改变它的子组件的位置。它分配所有子组件的首选大小。

*** hbox**
The *hbox* container lays out its children from left to right,giving each one its preferred width. The height of each child is set to the maximum of the preferred heights of all children.
这 *hbox* 容器将从左到右地去列出其子元素，并给予它们各自的首选宽度。而每个子元素的高度则设置为所有的子元素中的那个最大的首选高度。

*** vbox**
The *vbox* container lays out its children from top to bottom,giving each one its preferred height. The width of each child is set to the maximum of the preferred widths of all children.
这 *vbox* 容器将从上到下地去列出其子元素，并给予它们各自的首选高度。而每个子元素的宽度则设置为所有的子元素中的那个最大的首选宽度。

6.3.6 Common properties
6.3.6 公共属性

The following properties are supported by all components:
以下为支持所有组件的属性：

'left'
The distance from the left border of container to left border of the object in either of three formats:
从容器的左侧边框到对象的左边界之间的距离并以三种格式中的任何一种方式来表示：

x　　Value in pixels
x　　以像素为值
p%　　Percentage
p%　　百分比
p%+x　　mixture of both
P%+x　　两者混合使用

'top'
The distance from the left border of container to left border of the object in same format.
从容器的左侧边框到对象的左边界之间的距离并以相同的格式来表示：

'width'　　The width of object in same format.
"width"　　对象的宽度并以相同的格式。
'height'　　The height of object in same format.
"height"　　对象的高度并以相同的格式。
'id'　　The identifier for the component. This can be any arbitrary string. The ID can be used by scripts to refer to various components in the GUI component tree. Currently, there is one special ID value that GRUB recognizes: "_timeout_" Any component with this ID will have its *text*, *start*, *end*, *value*, and *visible* properties set by GRUB when it is counting down to an automatic boot of the default menu entry.
"id"　　为组件的标识符。这可以为任意的字符串。这个 ID 能够在 GUI 组件树中使用脚本去引用各种组件。目前，被 GRUB 所承认特殊的 ID 值只有一个："_timeout_"这个 ID 的任何组件的 *text*，*start*，*end*，*value*，和 *visible* 属性将被 GRUB 在倒计时结束并去自动引导一个默认的菜单条目的时候设置。

07.从网络去引导 GRUB - Booting GRUB from the network

****

7 Booting GRUB from the network

7 从网络引导 GRUB

==========

The following instructions only work on PC BIOS systems where the Preboot eXecution Environment (PXE) is available.
以下说明只针对预引导执行环境(PXE) 为可用的 PC BIOS 系统情况下。

To generate a PXE boot image, run:
Copy 'grub.pxe', '/boot/grub/*.mod', and '/boot/grub/*.lst' to the PXE (TFTP) server, ensuring that '*.mod' and '*.lst' are accessible via the '/boot/grub/' path from the TFTP server root. Set the DHCP server configuration to offer 'grub.pxe' as the boot file (the 'filename' option in ISC dhcpd).
生成一个 PXE 引导镜像，运行：
grub-mkimage --format=i386-pc-pxe --output=grub.pxe --prefix='(pxe)/boot/grub' pxe pxecmd
复制 "grub.pxe"，"/boot/grub/*.mod"，和 "/boot/grub/*.lst" 到 PXE(TFTP)服务器，确保可以在 TFTP 服务器的根目录通过路径 "/boot/grub" 可以访问到"*.mod"和"*.lst"文件。设置 DHCP 服务器的配置使 "grub.pxe" 作为引导文件 (这个 "filename" 选项在 ISC dhcpd 里。)。

You can also use the 'grub-mknetdir' utility to generate an image and a GRUB directory tree, rather than copying files around manually.
你也可以使用 "grub-mknetdir" 功能去生成一个镜像与一个 GRUB 的目录树，而不是手动地去复制文件。

After GRUB has started, files on the TFTP server will be accessible via the '(pxe)' device.
在 GRUB 启动之后，在 TFTP 服务器里的文件将通过 "(pxe)"驱动 从而被访问到。

The server and gateway IP address can be controlled by changing the '(pxe)' device name to '(pxe:SERVER-IP)' or '(pxe:SERVER-IP:GATEWAY-IP)'. Note that this should be changed both in the prefix and in any references to the device name in the configuration file.
服务器与网关 IP 地址的配置通过将"(pxe)" 设备名称改变成 "(pxe:SERVER-IP)" 或 "(pxe:SERVER-IP:GATEWAY-IP)"，可以用来配置服务器与网关的 IP 地址 。注意配置文件中的前缀和任何对此设备的引用应该同时改变。

GRUB provides several environment variables which may be used to inspect or change the behaviour of the PXE device:
GRUB 提供几个环境变量去用于检查或修改 PXE 设备的行为：

'net_pxe_ip'
The IP address of this machine. Read-only.
这台机器的 IP 地址。只读。

'net_pxe_mac'
The network interface's MAC address. Read-only.
网络接口的 MAC 地址。只读。

'net_pxe_hostname'
The client host name provided by DHCP. Read-only.
DHCP 服务器提供的客户端主机名称。只读。

'net_pxe_domain'
The client domain name provided by DHCP. Read-only.
DHCP 服务器提供的客户端域名。只读。

'net_pxe_rootpath'
The path to the client's root disk provided by DHCP. Read-only.
DHCP 服务器提供的通往客户端根磁盘的路径。只读。

'net_pxe_extensionspath'
The path to additional DHCP vendor extensions provided by DHCP.Read-only.
DHCP 服务器提供的通往附加的 DHCP 服务器供应商扩展的路径。只读。

'net_pxe_boot_file'
The boot file name provided by DHCP. Read-only.
DHCP 服务器提供的引导文件名称。只读。

'net_pxe_dhcp_server_name'
The name of the DHCP server responsible for these boot parameters.Read-only.
负责这些引导参数的 DHCP 服务器名称。只读。

'net_default_server'
The default server. Read-write, although setting this is only useful before opening a network device.
默认服务器。可读写，尽管这设置只能使用于在打开网络设备之前。

08.通过串口线使用 GRUB

08.通过串口线使用 GRUB-Using GRUB via a serial line

********** **********

8 Using GRUB via a serial line

8 通过串口线使用 GRUB

==========

This chapter describes how to use the serial terminal support in GRUB.
本章节描述如何使用 GRUB 支持的串口终端。

If you have many computers or computers with no display/keyboard, it could be very useful to control the computers through serial communications. To connect one computer with another via a serial line,you need to prepare a null-modem (cross) serial cable, and you may need to have multiport serial boards, if your computer doesn't have extra serial ports. In addition, a terminal emulator is also required, such as minicom. Refer to a manual of your operating system, for more information.
如果你有很多电脑或者很多没有显示器/键盘的电脑，那么通过串口通信去控制它们是非常有用的。为了通过串口线连接两台电脑，你需要准备一个零调制解调器电缆。如果你的电脑没有额外的串行端口，你需要准备一个多串口卡。另外，还需要一个终端仿真器，例如 minicom。详细信息请查看你的操作系统手册。

As for GRUB, the instruction to set up a serial terminal is quite simple. Here is an example:
至于 GRUB，设置串口终端的方法非常简单。例如下面这个例子：

```
grub> serial --unit=0 --speed=9600
grub> terminal_input serial; terminal_output serial
```

The command 'serial' initializes the serial unit 0 with the speed 9600bps. The serial unit 0 is usually called 'COM1', so, if you want to use COM2, you must specify '--unit=1' instead. This command accepts many other options, so please refer to *note serial::, for more details.
命令“serial”初始化 0 号串口单元并设置波特率为 9600bps。0 号串口单元通常被 称作“COM1”，因此，如果你想利用 COM2，需要用“--unit=2”指定。“serial” 命令有许多其他选项。详细信息查阅“serial”的手册页。

The commands 'terminal_input' (*note terminal_input::) and 'terminal_output' (*note terminal_output::) choose which type of terminal you want to use. In the case above, the terminal will be a serial terminal, but you can also pass 'console' to the command, as 'terminal_input serial console'. In this case, a terminal in which you press any key will be selected as a GRUB terminal. In the example above, note that you need to put both commands on the same command line, as you will lose the ability to type commands on the console after the first command.
命令“terminal_input”和“terminal_out”设置你使用的是哪种类型的终端。 在上面的情况中，选择的是一个串口终端。而且，通过命令“terminal_inout serial console”， 你可以选择一个“console”终端。如果这样的话，你可以像使用 GRUB 终端一样使用你的“console” 终端。在上面的例子中，注意应该将两个命令放在同一个命令行中，如果你先输入一条命令，那将无法输入另一条命令。

However, note that GRUB assumes that your terminal emulator is compatible with VT100 by default. This is true for most terminal emulators nowadays, but you should pass the option '--dumb' to the command if your terminal emulator is not VT100-compatible or implements few VT100 escape sequences. If you specify this option then GRUB provides you with an alternative menu interface, because the normal menu requires several fancy features of your terminal.
然而，GRUB 假定你的终端仿真器默认兼容 VT100 终端。今天的大多数终端都兼容 VT100 终端。 如果你的终端仿真器不兼容 VT100 终端，或者不能支持少量的 VT100 转义字符，那么你需要 给命令传递“--dump”选项。指定该选项后，GRUB 会提供给你一些额外的菜单接口，因为通常的 菜单需要一些你的终端独有的特性。

09 配合厂家定制的电源键使用 GRUB

09.配合厂家定制的电源键使用 GRUB - Using GRUB with vendor power-on keys

********** **********

09 Using GRUB with vendor power-on keys

09 配合厂家定制的电源键使用 GRUB

==========

Some laptop vendors provide an additional power-on button which boots another OS. GRUB supports such buttons with the 'GRUB_TIMEOUT_BUTTON','GRUB_DEFAULT_BUTTON','GRUB_HIDDEN_TIMEOUT_BUTTON' and 'GRUB_BUTTON_CMOS_ADDRESS' variables in default/grub (*note Simple configuration::). 'GRUB_TIMEOUT_BUTTON', 'GRUB_DEFAULT_BUTTON' and 'GRUB_HIDDEN_TIMEOUT_BUTTON' are used instead of the corresponding variables without the '_BUTTON' suffix when powered on using the special button. 'GRUB_BUTTON_CMOS_ADDRESS' is vendor-specific and partially model-specific. Values known to the GRUB team are:

一些便携式电脑供应商提供一个额外的电源键来启动另一个操作系统。 GRUB 通过 default/grub 中的“GRUB_TIMEOUT_BUTTON”、“GRUB_DEFAULT_BUTTON”、“GRUB_HIDDEN_TIMEOUT_BUTTON”和“GRUB_BUTTON_CMOS_ADDRESS” 变量来支持这种键。当使用特殊的键启动时“GRUB_TIMEOUT_BUTTON”、“GRUB_DEFAULT_BUTTON”和 “GRUB_HIDDEN_TIMEOUT_BUTTON”被用作取代 那些相对应的没有后缀“_BUTTON”的变量。“GRUB_BUTTON_CMOS_ADDRESS”变量对于不同的厂家是不同的，部分电脑对于不同的型号也是不同的。 下面列出 GRUB 团队所知道的部分值：

<Dell XPS M1530>

85:3

<Asus EeePC 1005PE>

84:1 (unconfirmed) (未经证实)

To take full advantage of this function, install GRUB into the MBR (*note Installing GRUB using grub-install::).

为了充分使用这个功能带来的好处，需要将 GRUB 安装入 MBR。 （使用 grub-install 安装 GRUB ）

If you have a laptop which has a similar feature and not in the above list could you figure your address and contribute? To discover the address do the following:

如果你有一台笔记本电脑有类似的特性并且不在支持的型号列表中， 你可以去计算出它的硬件码地址并且贡献出来使用？根据如下的步骤找出你的地址：

* boot normally 正常启动，执行下列命令

```
sudo modprobe nvram
sudo cat /dev/nvram | xxd > normal_button.txt
```

* boot using vendor button 使用厂家的键启动，执行下列命令

```
sudo modprobe nvram
sudo cat /dev/nvram | xxd > normal_vendor.txt
```

Then compare these text files and find where a bit was toggled. E.g. in case of Dell XPS it was:

然后比较这些文件找出哪个地方有一点小小的转变。例如：

```
byte 0x47: 20 --> 28
```

It's a bit number 3 as seen from following table:

从如下的表中可以看出发生变化在 3 号位地址。（字节 0x47 本来是 28 变为 28，根据下表对应于 3 ）

```
0       01
1       02
2       04
3       08
```

4 10
5 20
6 40
7 80

0x47 is decimal 71. Linux nvram implementation cuts first 14 bytes of CMOS. So the real byte address in CMOS is 71+14=85 So complete address is 85:3
0x47 的十进制表示为 71。 linux nvram 在执行时会剪去 CMOS 的前 14 个字节。所以在 CMOS 中真实的地址是 71+14=85。 所以完整的地址是 85：3

10.GRUB 镜像文件

10.GRUB 镜像文件-GRUB image files

****

10 GRUB image files

10 GRUB 镜像文件

==========

GRUB consists of several images: a variety of bootstrap images for starting GRUB in various ways, a kernel image, and a set of modules which are combined with the kernel image to form a core image. Here is a short overview of them.

GRUB 包含若干镜像文件：以各种方式来启动 GRUB 的引导镜像文件、一个内核镜像文件和 由一系列内核组件与内核镜像文件组成的核心镜像。 在这里对这些文件做个概况：

'boot.img'

On PC BIOS systems, this image is the first part of GRUB to start.It is written to a master boot record (MBR) or to the boot sector of a partition. Because a PC boot sector is 512 bytes, the size of this image is exactly 512 bytes.

The sole function of 'boot.img' is to read the first sector of the core image from a local disk and jump to it. Because of the size restriction, 'boot.img' cannot understand any file system structure, so 'grub-setup' hardcodes the location of the first sector of the core image into 'boot.img' when installing GRUB.

"boot.img"

在个人电脑 BIOS 系统中，这文件是 GRUB 开始启动执行的第一部分。 它一般被写入 MBR 中，或者被写入一个分区的启动扇区中。由于个人电脑的启动扇区只有 512 字节， 所以，这文件的大小也是 512 字节。"boot.img"的唯一功能是从一个本地磁盘中的核心镜像文件中读取第一个扇区并且把控制转移给它。 由于大小限制，"boot.img"不能理解任何文件系统结构，所以，在安装 GRUB 的时候，"grub-setup"把核心 镜像文件第一扇区的地址写入"boot.img"中。

'diskboot.img'

This image is used as the first sector of the core image when booting from a hard disk. It reads the rest of the core image into memory and starts the kernel.Since file system handling is not yet available, it encodes the location of the core image using a block list format.

"diskboot.img"

当从一个硬盘中启动时，这个镜像文件被用作核心镜像文件的第一个扇区。它读取剩余的核心镜像 文件并且启动内核。因为对文件系统的操作还不可用，它使用 block list format 解码核心镜像文件 的地址。

'cdboot.img'

This image is used as the first sector of the core image when booting from a CD-ROM drive. It performs a similar function to 'diskboot.img'.

"cdboot.img"

当从 CD-ROM 驱动器启动时，这个镜像文件被当作核心镜像文件的第一个扇区。它和"diskboot.img"起 相似的作用。

'pxeboot.img'

This image is used as the start of the core image when booting from the network using PXE. *Note Network.

'pxeboot.img"

当通过预启动执行环境从网络启动时，这个镜像文件被当作核心镜像文件的开始。

'lnxboot.img'

This image may be placed at the start of the core image in order to make GRUB look enough like a Linux kernel that it can be booted by LILO using an 'image=' section.

“lnxboot.img"

如果这个镜像文件被安置在核心镜像文件的开始，它会使 GRUB 更像一个 linux 内核，并且通过使用一个

“image=”节，可以被 LILO 启动。

'kernel.img'
This image contains GRUB's basic run-time facilities: frameworks for device and file handling, environment variables, the rescue mode command-line parser, and so on. It is rarely used directly, but is built into all core images.
"kernel.img"
这个镜像文件包含 GRUB 的基础的运行设施：管理设备和文件的机制、环境变量、救援模式下的命令行 解析器等等。它很少直接被使用，但是所有的核心镜像都包含它。

'core.img'
This is the core image of GRUB. It is built dynamically from the kernel image and an arbitrary list of modules by the 'grub-mkimage' program. Usually, it contains enough modules to access '/boot/grub', and loads everything else (including menu handling, the ability to load target operating systems, and so on) from the file system at run-time. The modular design allows the core image to be kept small, since the areas of disk where it must be installed are often as small as 32KB.
"core.img"
这是 GRUB 的核心镜像文件。它能够被"grub-mkimage"程序由内核镜像文件和一系列任意的模块动态地 建立。 通常，它包含足够的模块使它能连接“/boot/grub”文件夹，并且能在运行时载入文件系统中的其他任何东西（包含菜单管理，加载目标操作系统等等）。模块化的设计允许内核镜像很小，由于在磁盘中它们必须安装在 32KB 那么小的区域里。
*Note BIOS installation: for details on where the core image can be installed on PC systems.
*注意*BIOS 安装：在一些复杂的情况下内核镜像可能被安装到 PC 的系统里。

'*.mod'
Everything else in GRUB resides in dynamically loadable modules. These are often loaded automatically, or built into the core image if they are essential, but may also be loaded manually using the 'insmod' command (*note insmod).
"*.mod"
GRUB 中的任何其他东西都存在于动态可加载的模块中。他们经常被动态地加载，或者当他们必不可少时会被加入核心镜像中，而且，也可以使用命令“ismod”手动地加载。

For GRUB Legacy users
对于 **GRUB Legacy** 用户
==========
GRUB 2 has a different design from GRUB Legacy, and so correspondences with the images it used cannot be exact. Nevertheless, GRUB Legacy users often ask questions in the terms they are familiar with, and so here is a brief guide to how GRUB 2's images relate to that.
GRUB2 的设计不同于 GRUB Legacy，所以它们使用的镜像文件不可能准确对应。然而，GRUB Legacy 用户经常询问在这个方面它们有什么相似。所以，在下面这个简短的手册中列出它们的镜像文件之间的联系。

'stage1'
Stage 1 from GRUB Legacy was very similar to 'boot.img' in GRUB 2, and they serve the same function.
"stage1"
GRUB Legacy 中的“Stage 1”和 GRUB 2 中的“boot.img”非常相似，并且他们提供相同的功能。

'*_stage1_5'
In GRUB Legacy, Stage 1.5's function was to include enough filesystem code to allow the much larger Stage 2 to be read from an ordinary filesystem. In this respect, its function was similar to 'core.img' in GRUB 2. However, 'core.img' is much more capable than Stage 1.5 was; since it offers a rescue shell, it is sometimes possible to recover manually in the event that it is unable to load any other modules, for example if partition numbers have changed. 'core.img' is built in a more flexible way, allowing GRUB 2 to support reading modules from advanced disk types such as LVM and RAID.
GRUB Legacy could run with only Stage 1 and Stage 2 in some limited configurations, while GRUB 2

欢迎点击这里的链接进入精彩的Linux公社网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

Linux公社是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：**Ubuntu 专题** **Fedora 专题** **Android 专题** **Oracle 专题** **Hadoop 专题** **RedHat 专题** **SUSE 专题** **红旗 Linux 专题** **CentOS 专题**

Linux 公社微信公众号：linuxidc_com

requires 'core.img' and cannot work without it.
"*_stage1_5"
在 GRUB Legacy 中，Stage 1.5 的功能是包含足够的文件系统代码让更大的 Stage 2 可以从一个一般的文件系统中被读取出来。在这个方面，它的功能类似于 GRUB2 中的“core.img”。然而，“core.img”比 Stage 1.5 更加强大。自从它提供一个拯救模式的 shell 之后，在你的分区编号被改变的情况下，你不能加载任何模块，这时，通过它也可能被手动地恢复。“core.img”以一种灵活的方式建立，允许 GRUB2 从例如 LVM 和 RAID 这种高级的磁盘类型中读取模块。
GRUB Legacy 只可以以一种限制的方式运行 Stage 1 和 Stage 2，然而，GRUB 2 必须要“core.img”，并且没有它不能运行。

'stage2'
GRUB 2 has no single Stage 2 image. Instead, it loads modules from '/boot/grub' at run-time.
"stage2"
GRUB2 没有单独的“Stage 2”镜像。取而代之，它运行时从“/boot/grub”中加载模块。

'stage2_eltorito'
In GRUB 2, images for booting from CD-ROM drives are now constructed using 'cdboot.img' and 'core.img', making sure that the core image contains the 'iso9660' module. It is usually best to use the 'grub-mkrescue' program for this.
"stage2_eltorito"
在 GRUB2 中，从 CD-ROM 驱动器启动的镜像文件由“cdboot.img”和“core.img”创建，确保内核镜像文件中包含 “iso9660”模块。通常最好是使用"grub-mkrescue"程序来实现这些。

'nbgrub'
There is as yet no equivalent for 'nbgrub' in GRUB 2; it was used by Etherboot and some other network boot loaders.
"nbgrub"
在 GRUB2 中没有与“nbgrub”对应的。“nbgrub”在利用 Etherboot 和其他一些网络启动器启动时会被用上。

'pxegrub'
In GRUB 2, images for PXE network booting are now constructed using 'pxeboot.img' and 'core.img', making sure that the core image contains the 'pxe' and 'pxecmd' modules. *Note Network::.
"pxegrub"
在 GRUB2 中，用于 PXE 网络启动的镜像文件由“pxeboot.img”和“core.img”创建，确保内核镜像文件中包含“pxe”和“pxecmd”模块。

11.文件系统的语法及其含义

11.文件系统的语法及其含义 - Filesystem syntax and semantics

********** **********

11 Filesystem syntax and semantics

11 文件系统的语法及其含义

==========

GRUB uses a special syntax for specifying disk drives which can be accessed by BIOS. Because of BIOS limitations, GRUB cannot distinguish between IDE, ESDI, SCSI, or others. You must know yourself which BIOS device is equivalent to which OS device. Normally, that will be clear if you see the files in a device or use the command 'search' (*note search::).

GRUB 使用一种特殊的语法来指定 BIOS 可以访问的磁盘驱动器。由于 BIOS 的局限性，GRUB 不能识别 IDE，ESDI，SCSI 等各种设备接口。你必须清楚 BIOS 设备和操作系统设备之间的对应关系。通常情况下，这会在设备文档中清楚的看到，或者你可以利用命令“search”查看。

11.1 How to specify devices

11.1 如何指定设备

==========

The device syntax is like this:

指定设备的语法如下：

(DEVICE[,PARTMAP-NAME1PART-NUM1[,PARTMAP-NAME2PART-NUM2[,...]]])

'[]' means the parameter is optional. DEVICE depends on the disk driver in use. BIOS and EFI disks use either 'fd' or 'hd' followed by a digit, like 'fd0', or 'cd'. AHCI, PATA (ata), crypto, USB use the name of driver followed by a number. Memdisk and host are limited to one disk and so it's refered just by driver name. RAID (md), ofdisk (ieee1275 and nand), LVM (lv), LDM and arcdisk (arc) use intrinsic name of disk prefixed by driver name. Additionally just "nand" refers to the disk aliased as "nand". Conflicts are solved by suffixing a number if necessarry. Commas need to be escaped. Loopback uses whatever name specified to 'loopback' command. Hostdisk uses names specified in device.map as long as it's of the form [fhc]d[0-9]* or hostdisk/<OS DEVICE>. For crypto and RAID (md) additionally you can use the syntax <driver name>uuid/<uuid>.

"[]"里的参数是可选的。DEVICE 取决于使用的磁盘驱动器。BIOS 和 EFI 的磁盘使用“fd”或者“hd”后面接着个数字，例如"fd0(软驱)"，"cd(光驱)"。而 AHCI，PATA(ata)，crypto，USB 需要在驱动器的名字后面接个数字。Memdisk(内存模拟盘)和 host(主机)被限制在一个磁盘并且它是仅仅通过驱动器名字来引用。RAID(md)，ofdisk(ieee1275 和 nand)，LVM(lv)，LDM 和 arcdisk(arc)它们使用使用一个由驱动器的前缀得到的内部名称。此外，“nand”用来引用那些把别名设为“nand”的设备。在必要的情况下在名称后加个数字后缀来解决可能的冲突。输出逗号时需要使用到转义符。回路可以指定任何的名称来给"loopback"命令使用。本地磁盘设备使用名称明确指定设备。这种名称与设备之间的映射关系遵循这种形式:

[fhc]d[0-9]*或者 hostdisk/<OSDEVICE>.此外，对于 crypto 和 RAID(md)，你可以使用这种形式：<driver name>uuid/<uuid>.

```
(fd0)
(hd0)
(cd)
(ahci0)
(ata0)
(crypto0)
(usb0)
(cryptouuid/123456789abcdef0123456789abcdef0)
(mduuid/123456789abcdef0123456789abcdef0)
(lv/system-root)
(md/myraid)
(md/0)
```

(ieee1275/disk2)
(ieee1275//pci@1f\,0/ide@d/disk@2)
(nand)
(memdisk)
(host)
(myloop)
(hostdisk//dev/sda)

PART-NUM represents the partition number of DEVICE, starting from one. PARTNAME is optional but is recommended since disk may have several top-level partmaps. Specifying third and later component you can access to subpartitions.
PART-NUM 代表 DEVICE 的分区编号，从 1 开始。PARTNAME 是一个可选的选项并且推荐使用，由于磁盘可能有几个顶层的部件映射。指定第三个和后面的部分，你可以访问到对应设备分区的子分区。

The syntax '(hd0)' represents using the entire disk (or the MBR when installing GRUB), while the syntax '(hd0,1)' represents using the first partition of the disk (or the boot sector of the partition when installing GRUB).
"(hd0)"代表使用整个磁盘（或者当安装 GRUB 时，它对应于磁盘的 MBR），“(hd0,1)”代表使用磁盘的第一个分区（或者当安装 GRUB 时，它对应分区的启动扇区）。
(hd0,msdos1)
(hd0,msdos1,msdos5)
(hd0,msdos1,bsd3)
(hd0,netbsd1)
(hd0,gpt1)
(hd0,1,3)

If you enabled the network support, the special drives '(tftp)','(http)' and so on ars also available. Before using the network drive,you must initialize the network. *Note Network::, for more information.
如果你启动了网络连接，特殊的驱动“(tftp)”和“(http)”等可以被使用。在使用网络驱动器之前必须启动网络。

If you boot GRUB from a CD-ROM, '(cd)' is available. *Note Making a GRUB bootable CD-ROM::, for details.
使用“(cd)”可以利用 GRUB 从 CD-ROM 中启动。

11.2 How to specify files
11.2 如何指定文件
==========
There are two ways to specify files, by "absolute file name" and by "block list".
在 GRUB 中有 2 中指定文件的方法，其一是通过“绝对文件路径”，另一种是通过块列表。

An absolute file name resembles a Unix absolute file name, using '/' for the directory separator (not '\' as in DOS). One example is '(hd0,1)/boot/grub/grub.cfg'. This means the file '/boot/grub/grub.cfg' in the first partition of the first hard disk. If you omit the device name in an absolute file name, GRUB uses GRUB's "root device" implicitly. So if you set the root device to, say, '(hd1,1)' by the command 'set root=(hd1,1)' (*note set::), then '/boot/kernel' is the same as '(hd1,1)/boot/kernel'.
一个文件绝对路径类似 Unix 的绝对文件路径，使用“/”作为目录分隔符（DOS 中使用“\”）。例如“(hd0,1)/boot/grub/grub.cfg”。这个例子表示文件“/boot/grub/grub.cfg”在第一块硬盘的第一块分区中。如果你在文件绝对路径中忽略设备名称，GRUB 隐含地使用 GRUB 中的“root device”。因此如果你使用命令“set root=(hd1,1)”设置了 root device，那么“/root/kernel”和“(hd1,1)/boot/kernel”具有相同的意思。

11.3 How to specify block lists
11.3 如何指定块列表
==========

A block list is used for specifying a file that doesn't appear in the filesystem, like a chainloader. The syntax is '[OFFSET]+LENGTH[,[OFFSET]+LENGTH]...'. Here is an example:
块列表用来指定没有出现在文件系统中的文件，如链式加载器。用法是：“[OFFSET]+LENGTH[,[OFFSET]+LENGTH]...”，下面是一个例子：

```
0+100,200+1,300+300
```

This represents that GRUB should read blocks 0 through 99, block 200,and blocks 300 through 599. If you omit an offset, then GRUB assumes the offset is zero.
这个例子表示 GRUB 会读取块 0 到块 99，块 200，和块 300 到块 599。如果忽略偏移量，GRUB 会假定偏移量是 0。

Like the file name syntax (*note File name syntax::), if a blocklist does not contain a device name, then GRUB uses GRUB's "root device". So '(hd0,2)+1' is the same as '+1' when the root device is '(hd0,2)'.
和文件路径的用法一样，如果块列表中没有包含设备名称，GRUB 会使用 GRUB 中的“root device”。所以，当 root device 是“(hd0,2)”，命令“(hd0,2)+1”和“+1”的作用是一样的。

12.GRUB 的用户界面

12.GRUB 的用户界面 - GRUB's user interface

********** **********

12 GRUB's user interface

12 GRUB 的用户界面

==========

GRUB has both a simple menu interface for choosing preset entries from a configuration file, and a highly flexible command-line for performing any desired combination of boot commands.
GRUB 既可以通过一个简单的菜单界面去选择从配置文件里预设好了的菜单条目，又可以通过一个高度灵活的命令行去执行于你想要的引导命令的组合。

GRUB looks for its configuration file as soon as it is loaded. If one is found, then the full menu interface is activated using whatever entries were found in the file. If you choose the "command-line" menu option, or if the configuration file was not found, then GRUB drops to the command-line interface.
GRUB 会查看它的配置文件并且一经找到则立即把它加载。如果找到一个，那么完整的菜单界面会被激活使用并从文件里查找出所有的条目。如果选择了"command-line"菜单选项，或找不到配置文件，那么 GRUB 会往下跳转到 command-line(命令行)界面。

12.1 The flexible command-line interface

12.1 灵活的命令行界面

==========

The command-line interface provides a prompt and after it an editable text area much like a command-line in Unix or DOS. Each command is immediately executed after it is entered(1) (*note Command-line interface-Footnote-1::). The commands (*note Command-line and menu entry commands::) are a subset of those available in the configuration file, used with exactly the same syntax.
命令行界面里提供了一个像 Unix 命令行或 DOS 命令行的提示符在可编辑文本区域里。每个命令都会立即执行于回车之后(注：命令行界面-注脚-1)。这些命令是一个有效的配置文件的子集，使用完全相同的语法(注：命令行与菜单条目命令)。

Cursor movement and editing of the text on the line can be done via a subset of the functions available in the Bash shell:
光标移动和在线编辑文本可以通过一个有效的功能子集实现于 Bash shell 之内：

<C-f>
<PC right key>
Move forward one character.
向前移动一个字符。

<C-b>
<PC left key>
Move back one character.
返回移动一个字符。

<C-a>
<HOME>
Move to the start of the line.
移动到行的开始位置。

<C-e>
<END>
Move the the end of the line.
移动到行的结束位置。

<C-d>
<DEL>
Delete the character underneath the cursor.
删除光标所指的字符。

<C-h>
<BS>
Delete the character to the left of the cursor.
删除光标左边的字符。

<C-k>
Kill the text from the current cursor position to the end of the line.
消除文本从当前的光标位置直到行的结束位置。

<C-u>
Kill backward from the cursor to the beginning of the line.
往前消除从光标位置直到行的起始位置。

<C-y>
Yank the killed text back into the buffer at the cursor.
接出光标处被删除的文本到缓冲区。

<C-p>
<PC up key>
Move up through the history list.
往上回滚(使用过的命令)历史列表。

<C-n>
<PC down key>
Move down through the history list.
往下回滚(使用过的命令)历史列表。

When typing commands interactively, if the cursor is within or before the first word in the command-line, pressing the <TAB> key (or <C-i>)will display a listing of the available commands, and if the cursor is after the first word, the '<TAB>' will provide a completion listing of disks, partitions, and file names depending on the context. Note that to obtain a list of drives, one must open a parenthesis, as 'root ('.
当键入交互式命令的时候，如果光标在命令行中的第一个字词之前或之内，按下<TAB>键(或<C-i>)将显示出一个可用命令的列表，并且如果这个光标是在第一个字词之后，那这个"<TAB>"将提供一个完整的磁盘，扇区和当前上下文相关文件名的清单。注意为了获取一个驱动器列表，你必须使用一个圆括号，就像"root ("。

Note that you cannot use the completion functionality in the TFTP filesystem. This is because TFTP doesn't support file name listing for the security.
注意你不能在 TFTP 文件系统中使用完整的功能。这是因为 TFTP 为了安全性而不支持文件名字列表显示功能。
(1) However, this behavior will be changed in the future version, in a user-invisible way.
(1) 然而，这些行为将在未来版本中有所改变，以一种用户不可见的方式实现。

12.2 The simple menu interface
12.2 一个简单的菜单界面
==========
The menu interface is quite easy to use. Its commands are both reasonably intuitive and described on screen.
一个菜单界面是相当容易地使用的。它是命令都是相当合理直观地描述于屏幕之上。

Basically, the menu interface provides a list of "boot entries" to the user to choose from. Use the arrow keys to select the entry of choice, then press <RET> to run it. An optional timeout is available to boot the default entry (the first one if not set), which is aborted by pressing any key.
基本上，一个菜单界面提供一个显示列表去把"引导条目"提供给用户去选择。使用箭头键去选择上条目选项，然后按<RET>(回车换行的缩写，一般可理解为回车)键去运行它。一个有效的倒计时选项去引导一个默认条目(如果没设置则为第一个条目)，随意按下一个键能让它(倒计时)失效。

Commands are available to enter a bare command-line by pressing <c>(which operates exactly like the non-config-file version of GRUB, but allows one to return to the menu if desired by pressing <ESC>) or to edit any of the "boot entries" by pressing <e>.
通过按下键<c>进入一个空命令行(这个操作完全相同于 non-config-file(没有配置文件)的 GRUB 版本，如果你想退出，按<ESC>可以返回到菜单。)或者通过按<e>去编辑任何“引导条目”。

If you protect the menu interface with a password (*note Security::),all you can do is choose an entry by pressing <RET>, or press <p> to enter the password.
如果你想通过设置一个 password(注：安全)保护你的菜单，你所能做的是选择上一个条目并通过按<RET>，或者按<p>去输入密码。

12.3 Editing a menu entry
12.3 编辑一个菜单条目
==========
The menu entry editor looks much like the main menu interface, but the lines in the menu are individual commands in the selected entry instead of entry names.
菜单条目的编辑看起来更像主菜单界面，但是菜单中的行里的个人命令的选择条目会代替为菜单界面的条目名称。

If an <ESC> is pressed in the editor, it aborts all the changes made to the configuration entry and returns to the main menu interface.
如果在编辑中按下<ESC>，它会取消配置条目里做出的所有修改并且返回到主菜单界面。

Each line in the menu entry can be edited freely, and you can add new lines by pressing <RET> at the end of a line. To boot the edited entry,press <Ctrl-x>.
每一行的菜单条目都可以随意编辑，并且你能够加入新行通过按<RET>键作用于此行的结束位置。想去启动一个编辑好的条目，可按<Ctrl-x>。

Although GRUB unfortunately does not support "undo", you can do almost the same thing by just returning to the main menu using <ESC>.
虽然 GRUB 很不幸地不能支持"undo"，但你可以实现几乎相同的事情只需通过使用<ESC>返回到主菜单。

13.GRUB 环境变量

13.GRUB 环境变量 - GRUB environment variables

********** **********

13 GRUB environment variables

13 GRUB 环境变量

==========

GRUB supports environment variables which are rather like those offered by all Unix-like systems. Environment variables have a name, which is unique and is usually a short identifier, and a value, which is an arbitrary string of characters. They may be set (*note set::), unset (*note unset::), or looked up (*note Shell-like scripting::) by name.

GRUB 支持环境变量就像那些所有 Unix 类系统所提供的一样。环境变量需要命名，要求具有唯一性的和简单易懂的标识符，并且赋予初始值，此值为一个任意的字符串。它们可以被设置，撤销，或者通过名称查找它。

A number of environment variables have special meanings to various parts of GRUB. Others may be used freely in GRUB configuration files.

一些环境变量在 GRUB 里的各个部分都有其特殊的意义。另一些也许可以随意地在 GRUB 的配置文件里使用。

13.1 Special environment variables

13.1 特殊的环境变量

==========

These variables have special meaning to GRUB.

在 GRUB 里具有特殊的意义的变量。

13.1.1 biosnum

13.1.1 biosnum

When chain-loading another boot loader (*note Chain-loading::), GRUB may need to know what BIOS drive number corresponds to the root device (*note root::) so that it can set up registers properly. If the BIOSNUM variable is set, it overrides GRUB's own means of guessing this.

当 链式-加载 另一个引导载入器时，GRUB 需要知道相应根设备的 BIOS 驱动器的编号(注：根)，以便它能够正确地设置寄存器的值。如果 BIOSNUM 的值被设置，它将覆盖 GRUB 对此的猜测值。

For an alternative approach which also changes BIOS drive mappings the chain-loaded system, *note drivemap::.

而另一种方法为从链式-加载系统里去改变 BIOS 驱动的映射，注：驱动器映射。

13.1.2 chosen

13.1.2 chosen

When executing a menu entry, GRUB sets the CHOSEN variable to the title of the entry being executed.

当执行一个登陆菜单时，GRUB 把 CHOSEN 变量设置为被执行登陆菜单的标题。

If the menu entry is in one or more submenus, then CHOSEN is set to the titles of each of the submenus starting from the top level followed by the title of the menu entry itself, separated by '>'.

如果登录菜单在一个或多个子菜单中时，“CHOSEN”被设置为从最高层次的菜单开始，后面依次连接着子菜单，最后接着登陆菜单，并且 “>”作为分隔符。

13.1.3 color_highlight

13.1.3 color_highlight

This variable contains the "highlight" foreground and background terminal colors, separated by a slash ('/'). Setting this variable changes those colors. For the available color names, *note color_normal.

此变量用来设置终端颜色中的“highlight”前景色和背景色，以“/”分隔。通过设置它可以改变“highlight”前景色和背景色。可用的颜色有，注:color_normal.

The default is 'black/white'.
此变量的默认值是“black/white”

13.1.4 color_normal
13.1.4 color_normal

This variable contains the "normal" foreground and background terminal colors, separated by a slash ('/'). Setting this variable changes those colors. Each color must be a name from the following list:
此变量用来设置终端颜色中的“normal”前景色和背景色，以“/”分隔。通过设置它可以改变“normal”前景色和背景色，以“/”分隔。可使用的颜色列表：
* black
* blue
* green
* cyan
* red
* magenta
* brown
* light-gray
* dark-gray
* light-blue
* light-green
* light-cyan
* light-red
* light-magenta
* yellow
* white

The default is 'white/black'.
此变量的默认值是“white/black”。

13.1.5 debug
13.1.5 debug

This variable may be set to enable debugging output from various components of GRUB. The value is a list of debug facility names separated by whitespace or ',', or 'all' to enable all available debugging output.
此变量被用来设置启动各种 GRUB 组件的调试输出。它的值是以空格或“，”分隔的调试设备名称的列表。

13.1.6 default
13.1.6 default

If this variable is set, it identifies a menu entry that should be selected by default, possibly after a timeout (*note timeout::). The entry may be identified by number or by title.
如果此变量被设置，在登录菜单选择超时后，系统默认启动变量所指定的菜单入口。通过编号或名称都可以识别入口。

If the entry is in a submenu, then it must be identified using the titles of each of the submenus starting from the top level followed by the number or title of the menu entry itself, separated by '>'. For example, take the following menu structure:
如果入口在一个子菜单中，通过从高到低分层次的菜单说明，并且用“>”分隔开菜单，就能识别指定入口。例如，假设有如下的菜单结构：

```
Submenu 1
    Menu Entry 1
    Menu Entry 2
Submenu 2
    Submenu 3
        Menu Entry 3
        Menu Entry 4
        Menu Entry 5
```

"Menu Entry 3" would then be identified as 'Submenu 2>Submenu 3>Menu Entry 3'.
“Submenu 2>Submenu 3>Menu Entry 3”被识别为入口“Menu Entry 3”。
This variable is often set by 'GRUB_DEFAULT' (*note Simple configuration::), 'grub-set-default', or 'grub-reboot'.
这变量通常用“GRUB_DEFAULT,grub-set-default”或“grub-reboot”来设置。

13.1.7 fallback
13.1.7 fallback

If this variable is set, it identifies a menu entry that should be selected if the default menu entry fails to boot. Entries are identified in the same way as for 'default' (*note default::).
在此变量被设置的情况下，如果默认的入口启动失败，它所指定的入口会被选择去启动。它的指定方法和“default”变量一样。

13.1.8 gfxmode
13.1.8 gfxmode

If this variable is set, it sets the resolution used on the 'gfxterm' graphical terminal. Note that you can only use modes which your graphics card supports via VESA BIOS Extensions (VBE), so for example native LCD panel resolutions may not be available. The default is 'auto', which selects a platform-specific default that should look reasonable.
如果这个变量设置好了，它会设置"gfxterm"图像终端中使用的分辨率。注意你只能使用你的图像卡通过 VESA BIOS 扩展（VBS）所支持的那些模式，所以，例如本地 LCD 屏的分辨率可能是不可用的。变量默认值为"auto"，它选择一个看起来合理的并且与平台-特性相关的值作为默认值。
The resolution may be specified as a sequence of one or more modes,separated by commas (',') or semicolons (';'); each will be tried in turn until one is found. Each mode should be either 'auto', 'WIDTHxHEIGHT', or 'WIDTHxHEIGHTxDEPTH'.
这个分辨率可能指定为一系列的一个或多个模式，以逗号分隔(",") 或 分号(";")；每个(模式)都将去依次尝试直到找到一个(能用起效的模式)。每个模式(配置参数)都应该是"auto"，"宽值 x 高值"，或"宽值 x 高值 x 色深"其中一个。

13.1.9 gfxpayload
13.1.9 gfxpayload

If this variable is set, it controls the video mode in which the Linux kernel starts up, replacing the 'vga=' boot option (*note linux). It may be set to 'text' to force the Linux kernel to boot in normal text mode, 'keep' to preserve the graphics mode set using 'gfxmode', or any of the permitted values for 'gfxmode' to set a particular graphics mode (*note gfxmode::).
如果这个变量设置好了，它所制定的视频模式会随着 Linux 内核启动时被启用，替换这个"vga="引导选项(注：linux)。它可能设置为"text"去迫使 Linux 内核去引导进入到正常的文本模式，设置为"keep"去保护使用“gfxmode”设置的图形模式，或设置为任何允许的值为使"gfxmode"设置一个特定的图形模式(注：gfxmode)。
Depending on your kernel, your distribution, your graphics card, and the phase of the moon, note that using this option may cause GNU/Linux to suffer from various display problems, particularly during the early part of the boot sequence. If you have problems, set this variable to 'text' and GRUB will tell Linux to boot in normal text mode.

具体取决于你的内核，你的发行版，你的图形卡，你的项目周期，注意因为使用此选项可能导致 GNU/Linux 遭受各种各样的显示问题，特别是早期的那部分启动顺序。如果你遇到问题，可以设置此值为"text"后 GRUB 将告诉 Linux 去引导进标准文本模式。
The default is platform-specific. On platforms with a native text mode (such as PC BIOS platforms), the default is 'text'. Otherwise the default may be 'auto' or a specific video mode.
它的默认值取决于 平台-特性。在一个平台的本地文本模式里(比如 PC 的 BIOS 平台)，它的默认值为"text"。另外一些的默认值可能为"auto"或者为一个特定的视频模式。
This variable is often set by 'GRUB_GFXPAYLOAD_LINUX' (*note Simple configuration).
这个变量通常是经由"GRUB_GFXPAYLOAD_LINUX"来设置 (注：简单配置-Simple configuration)。

13.1.10 gfxterm_font
13.1.10 gfxterm_font

If this variable is set, it names a font to use for text on the 'gfxterm' graphical terminal. Otherwise, 'gfxterm' may use any available font.
如果变量被设置，在“gfxterm”图形终端中的文本将会使用变量设置的字体，否则，终端将使用任何一种可用的字体。

13.1.11 icondir
13.1.11 icondir

If this variable is set, it names a directory in which the GRUB graphical menu should look for icons after looking in the theme's 'icons' directory. *Note Theme file format.
如果变量被设置，会被设置为一个文件夹，当 GRUB 图形菜单寻找图标时，会先在主题的图形文件夹中，之后会在此文件夹中寻找。

13.1.12 lang
13.1.12 lang

If this variable is set, it names the language code that the 'gettext' command (*note gettext::) uses to translate strings. For example,French would be named as 'fr',and Simplified Chinese as 'zh_CN'.
如果变量被设置，会被设置为某种语言的编码方式，命令“gettext”会根据编码方式翻译字符串。例如，French 会被设为“fr”，简体中文会被设为“zh_CN”。
'grub-mkconfig' (*note Simple configuration::) will try to set a reasonable default for this variable based on the system locale.
“grub-mkconfig”命令会根据本地系统为此变量设置一个合理的默认值。

13.1.13 locale_dir
13.1.13 locale_dir

If this variable is set, it names the directory where translation files may be found (*note gettext::), usually '/boot/grub/locale'. Otherwise,internationalization is disabled.
如果变量被设置，会被设为一个文件夹，在其中可以找到翻译文档，通常在“/boot/grub/locale”，否则，国际化的能力是不够的。
'grub-mkconfig' (*note Simple configuration::) will set a reasonable default for this variable if internationalization is needed and any translation files are available.
如果因国际化的需要，并且所有的翻译文档都具备，“grub-mkconfig”命令会为此变量生成一个合理的默认值。

13.1.14 menu_color_highlight
13.1.14 menu_color_highlight

This variable contains the foreground and background colors to be used for the highlighted menu entry, separated by a slash ('/'). Setting this variable changes those colors. For the available color names,*note

color_normal.
变量被用于去设置被高亮的入口菜单的前景色和背景色，值之间用“/”分隔。通过设置此变量去改变那些颜色，可用的颜色：参见 color_normal
The default is the value of 'color_highlight' (*note color_highlight).
变量的默认值是“color_highlight”的值。

13.1.15 menu_color_normal
13.1.15 menu_color_normal

This variable contains the foreground and background colors to be used for non-highlighted menu entries, separated by a slash ('/'). Setting this variable changes those colors. For the available color names,*note color_normal.
变量被用于去设置未被高亮的入口菜单的前景色和背景色，值之间用“/”分隔。通过设置此变量去改变那些颜色，可用的颜色：参见 color_normal。
The default is the value of 'color_normal' (*note color_normal).
变量的默认值是“color_normal”的值。

13.1.16 net_pxe_boot_file
13.1.16 net_pxe_boot_file
---
*Note Network.
*注 网络-Network。

13.1.17 net_pxe_dhcp_server_name
13.1.17 net_pxe_dhcp_server_name
---
*Note Network.
*注 网络-Network。

13.1.18 net_pxe_domain
13.1.18 net_pxe_domain
---
*Note Network.
*注 网络-Network。

13.1.19 net_pxe_extensionspath
13.1.19 net_pxe_extensionspath
---
*Note Network.
*注 网络-Network。

13.1.20 net_pxe_hostname
13.1.20 net_pxe_hostname
---
*Note Network.
*注 网络-Network。

13.1.21 net_pxe_ip
13.1.21 net_pxe_ip
---
*Note Network.
*注 网络-Network。

13.1.22 net_pxe_mac
13.1.22 net_pxe_mac
---
*Note Network.
*注 网络-Network。

13.1.23 net_pxe_rootpath
13.1.23 net_pxe_rootpath
---
*Note Network.
*注 网络-Network。

13.1.24 pager
13.1.24 pager

If set to '1', pause output after each screenful and wait for keyboard input. The default is not to pause output.
如果被设置为 1，会在每次满屏输出后暂停，等待键盘的输入，默认值为不暂停输出。

13.1.25 prefix
13.1.25 prefix

The location of the '/boot/grub' directory as an absolute file name (*note File name syntax::). This is normally set by GRUB at startup based on information provided by 'grub-install'. GRUB modules are dynamically loaded from this directory, so it must be set correctly in order for many parts of GRUB to work.
“/boot/grub”文件夹的位置被看作是一个绝对文件路径，正常情况下，它在启动时根据“grub-install”提供的信息被 GRUB 设置。GRUB 的模块是动态地从此文件夹中加载的，所以它必须被设置正确，以便 GRUB 的各部分的正常工作。

13.1.26 pxe_blksize
13.1.26 pxe_blksize
---
*Note Network.
*注 网络-Network。

13.1.27 pxe_default_gateway
13.1.27 pxe_default_gateway
---
*Note Network.
*注 网络-Network。

13.1.28 pxe_default_server
13.1.28 pxe_default_server
---
*Note Network.
*注 网络-Network。

13.1.29 root
13.1.29 root

The root device name (*note Device syntax::). Any file names that do not specify an explicit device name are read from this device. The default is normally set by GRUB at startup based on the value of 'prefix' (*note prefix).

根设备的名称。任何没有被明确指定设置名称的文件都会从这个设备中读取。默认值通常被 GRUB 在启动时根据“prefix”的值去设置，
For example, if GRUB was installed to the first partition of the first hard disk, then 'prefix' might be set to '(hd0,msdos1)/boot/grub' and 'root' to 'hd0,msdos1'.
例如，如果 GRUB 被安装在第一块硬盘的第一块分区，那么 "prefix” 会设置为 "(hd0,msdos1)/boot/grub”，“root” 被设置为 “hd0,msdos1” 。

13.1.30 superusers
13.1.30 superusers

This variable may be set to a list of superuser names to enable authentication support. *Note Security.
变量被设置为超级用户的列表，用来开启授权支持。

13.1.31 theme
13.1.31 theme

This variable may be set to a directory containing a GRUB graphical menu theme. *Note Theme file format.
变量被设置为包含 GRUB 图形终端菜单主题的文件夹。
This variable is often set by 'GRUB_THEME' (*note Simple configuration).
变量经常通过“GRUB_THEME”来设置。

13.1.32 timeout
13.1.32 timeout

If this variable is set, it specifies the time in seconds to wait for keyboard input before booting the default menu entry. A timeout of '0' means to boot the default entry immediately without displaying the menu;a timeout of '-1' (or unset) means to wait indefinitely.
如果变量被设置，它可以以秒的方式指定一个时间，在这段时间等待键盘输入，超时后会从默认的入口菜单启动。如果被设为 0，意味着不显示菜单，立即从默认入口启动。如果被设为-1，意味着无限期地等待键盘输入。
This variable is often set by 'GRUB_TIMEOUT' or 'GRUB_HIDDEN_TIMEOUT' (*note Simple configuration::).
变量经常通过“GRUB_TIMEOUT”或“GRUB_HIDDEN_TIMEOUT”来设置。

13.2 The GRUB environment block
13.2 GRUB 环境块
==========

It is often useful to be able to remember a small amount of information from one boot to the next. For example, you might want to set the default menu entry based on what was selected the last time. GRUB deliberately does not implement support for writing files in order to minimise the possibility of the boot loader being responsible for file system corruption, so a GRUB configuration file cannot just create a file in the ordinary way. However, GRUB provides an "environment block" which can be used to save a small amount of state.
能从此次启动中记住少量信息，为方便下次启动，这是非常有用的。例如，可能你想把默认入口菜单设置为上次所选择的菜单。为最小化启动加载器为文件系统损坏负责的可能性，GRUB 故意不支持对文件的写操作，所以，GRUB 配置文件不可以通过普通的方式创建文件。然而，GRUB 提供一种被用来保存少量状态的“环境块”。

The environment block is a preallocated 1024-byte file, which normally lives in '/boot/grub/grubenv' (although you should not assume this). At boot time, the 'load_env' command (*note load_env::) loads environment variables from it, and the 'save_env' (*note save_env) command saves environment variables to it. From a running system, the 'grub-editenv' utility can be used to edit the environment block.
环境块是一个被预分配为 1024 个字节的文件，通常位于“/boot/grub/grubenv”（但是不应该假定是这样）。启动时，“load_env”命令从其中加载环境变量，“save_env”命令将环境变量保存在其中。在一个运行着的操作系统中，工具“grub-editenv”可以用来编辑环境块。

For safety reasons, this storage is only available when installed on a plain disk (no LVM or RAID), using a non-checksumming filesystem (no ZFS), and using BIOS or EFI functions (no ATA, USB or IEEE1275).
为了安全的因素，当安装在普通硬盘中（非 LVM 和 RAID），不使用校验和文件系统（非 ZFS），使用 BOIS 或 EFI 技术（非 ATA，USB 和 IEEE1275）。

'grub-mkconfig' uses this facility to implement 'GRUB_SAVEDEFAULT' (*note Simple configuration).
“grub-mkconfig”命令使用上述功能来使“GRUB_SAVEDEFAULT”生效。

14.可用命令列表

14.可用命令列表-The list of available commands

********** **********

14 The list of available commands

14 有效命令的列表

==========

In this chapter, we list all commands that are available in GRUB.

在此章节中，我们会列举出在 GRUB 里所有的有效命令。

Commands belong to different groups. A few can only be used in the global section of the configuration file (or "menu"); most of them can be entered on the command-line and can be used either anywhere in the menu or specifically in the menu entries.

命令归属于各自不同的类别。只有小许能够仅仅被使用于配置文件的全局部分(或"菜单"部分)；它们中的大部分都能够输入到命令行 并且 能够使用到菜单或特定菜单条目里的任何位置。

In rescue mode, only the 'insmod' (*note insmod), 'ls' (*note ls), 'set' (*note set), and 'unset' (*note unset) commands are normally available. If you end up in rescue mode and do not know what to do, then *note GRUB only offers a rescue shell.

在 救援模式-rescue mode 里，只有"insmod"，"ls"，"set"，和"unset"命令能正常可用。如果你最终启动到救援模式里并且不知道该做什么，则 *注释：GRUB 仅仅提供一个救援的 shell。

14.1 The list of commands for the menu only

14.1 只应用于 menu-菜单 的命令列表

==========

The semantics used in parsing the configuration file are the following:

下列列出配置文件中所使用的语义的语法解析：

* The files must be in plain-text format.
* 文件必须是纯文本格式。
* '#' at the beginning of a line in a configuration file means it is only a comment.
* "#"在一个配置文件里的一行文本的开始位置意味着这一行文本仅仅是个注释。
* Options are separated by spaces.
* 选项间由空格作分隔。
* All numbers can be either decimal or hexadecimal. A hexadecimal number must be preceded by '0x', and is case-insensitive.
* 所有的数字可以为十进制或十六进制。一个十六进制数必须以"0x"开头，并且为大小写不敏感。

These commands can only be used in the menu:

这些命令只能用于菜单：

14.1.1 menuentry

14.1.1 menuentry

-- Command: menuentry TITLE ['--class=class' ...] ['--users=users'] ['--unrestricted'] ['--hotkey=key'] { COMMAND; ... }

This defines a GRUB menu entry named TITLE. When this entry is selected from the menu, GRUB will set the CHOSEN environment variable to TITLE, execute the list of commands given within braces, and if the last command in the list returned successfully and a kernel was loaded it will execute the 'boot' command.

这里定义一个 GRUB 菜单条目的名为 TITLE。当这个菜单的条目被选中时，GRUB 将设置 CHOSEN 项的环境变量为 TITLE 项的值，并执行括号内给出的命令列表，直到列表中的最后一个命令执行结束并返回成功后仅接着一个内核被成功加载之后，它将会自动去执行一个"boot"命令。

The '--class' option may be used any number of times to group menu entries into classes. Menu themes may display different classes using different styles.

这里的 "--class" 选项可以使用任意数量的菜单条目组合成一个类。菜单主题可以显示出不同类型的不同风格。

The '--users' option grants specific users access to specific menu entries. *Note Security.
这里的 "--users" 选项授予特定用户去访问特定的菜单项。*注：安全-Security。
The '--unrestricted' option grants all users access to specific menu entries. *Note Security.
这里的 "--unrestricted" 选项授予所有用户访问特定的菜单项。*注：安全-Security。
The '--hotkey' option associates a hotkey with a menu entry. KEY may be a single letter, or one of the aliases 'backspace', 'tab',or 'delete'.
这里的 "--hotkey" 选项为将一个热键和一个菜单联系起来。KEY 值可以为一个单一字母，或者为"backspace"键，"tab"键或"delete"键的别名。

14.1.2 submenu
14.1.2 submenu

-- Command: submenu TITLE ['--class=class' ...] ['--users=users'] ['--unrestricted'] ['--hotkey=key'] { MENU ENTRIES ... }
This defines a submenu. An entry called TITLE will be added to the menu; when that entry is selected, a new menu will be displayed showing all the entries within this submenu.
这里自定义一个子菜单。一个以 TITLE 的参数为命名的选项入口将被添加到主菜单上；当该条目被选上后，一个新的菜单将被显示出来并展示出子菜单的所有条目内容。
All options are the same as in the 'menuentry' command (*note menuentry).
所有选项都与"menuentry"命令的选项相同(*注：菜单条目-menuentry)。

14.2 The list of general commands
14.2 这里列表出常规命令
==========
Commands usable anywhere in the menu and in the command-line.
以下命令可以应用于任何的菜单与命令行里。
14.2.1 serial
14.2.1 serial

-- Command: serial ['--unit=unit'] ['--port=port'] ['--speed=speed'] ['--word=word'] ['--parity=parity'] ['--stop=stop']
Initialize a serial device. UNIT is a number in the range 0-3 specifying which serial port to use; default is 0, which corresponds to the port often called COM1. PORT is the I/O port where the UART is to be found; if specified it takes precedence over UNIT. SPEED is the transmission speed; default is 9600. WORD and STOP are the number of data bits and stop bits. Data bits must be in the range 5-8 and stop bits must be 1 or 2. Default is 8 data bits and one stop bit. PARITY is one of 'no', 'odd', 'even' and defaults to 'no'.
初始化一个串行设备。以一个范围从 0-3 的号码为单位去指定所使用的串口端口；默认值为 0，这个端口通常被称为 COM1。"port"为连接 UART(通用异步收发传输器)的 I/O 端口；如果它被定义了，它的优先级高于"unit"。"speed"为指定的传输速度；其默认值为 9600。"word"和"stop"表示数据位长度和停止位长度。数据位必须在范围 5-8 内并且停止位必须为 1 或者 2。默认值是 8 个数据位和 1 个停止位。"parity"(奇偶校验)只能为"no"，"odd"，"even"中的一个值并且默认为"no"。
The serial port is not used as a communication channel unless the 'terminal_input' or 'terminal_output' command is used (*note terminal_input, *note terminal_output).
串口端口是不会自动作为通讯信道的，除非你使用了"terminal_input"或"terminal_output"命令去启用它。(*注 terminal_input，*注 terminal_output)。
See also *note Serial terminal.
另请参阅 *注 串口终端-Serial terminal。

14.2.2 terminal_input
14.2.2 terminal_input

-- Command: terminal_input ['--append'|'--remove'] [terminal1] [terminal2] ...
List or select an input terminal.
用以列出或选择一个输入终端。
With no arguments, list the active and available input terminals.
当没有参数时，会列出所有有效的可用输入终端。
With '--append', add the named terminals to the list of active input terminals; any of these may be used to provide input to GRUB.
当带"--append"参数时，会把所指定位置的终端名加入到有效的输入终端列表里；在这里面所有列出的终端都为提供给 GRUB 的可用输入终端。
With '--remove', remove the named terminals from the active list.
当带"--remove"参数时，会从有效列表中删除指定的终端。
With no options but a list of terminal names, make only the listed terminal names active.
当没有选项但输入一列终端名称时，它仅仅会使列出列表里有效的终端名。

14.2.3 terminal_output
14.2.3 terminal_output

-- Command: terminal_output ['--append'|'--remove'] [terminal1] [terminal2] ...
List or select an output terminal.
用以列出或选择一个输出终端。
With no arguments, list the active and available output terminals.
当没有参数时，列出有效的和可用的输出终端。
With '--append', add the named terminals to the list of active output terminals; all of these will receive output from GRUB.
当带"--append"时，加入所指位置的终端到有效的输入终端列表里；所有列出的都将于 GRUB 里接受输出。
With '--remove', remove the named terminals from the active list.
当带"--remove"时，会从有效列表中删除指定位置的终端。
With no options but a list of terminal names, make only the listed terminal names active.
当没有选项但却输入一个终端名字时，它仅仅会使列出的终端有效。

14.2.4 terminfo
14.2.4 terminfo

-- Command: terminfo [-a|-u|-v] [term]
Define the capabilities of your terminal by giving the name of an entry in the terminfo database, which should correspond roughly to a 'TERM' environment variable in Unix.
通过给出一个 terminfo 数据库里的条目的名称，来定义终端里的功能，这个名称应该大致对应于一个在 Unix 里的"TERM"环境变量。
外文引用：
terminfo/termcap: 一個特殊的資料庫, 記載著各種不同的終端機所使用的控制碼. 它讓上層的程式庫 (例如 curses/ncurses 或是 slang) 可以視不同的環境 (終端機) 來送不同的控制字串, 而在不同的環境下製造出固定的特殊效果 (移動游標, 字串屬性 ...)
The currently available terminal types are 'vt100', 'vt100-color','ieee1275', and 'dumb'. If you need other terminal types, please contact us to discuss the best way to include support for these in GRUB.
当前可用的终端类型是"vt100"，"vt100-color"，"ieee1275"，和"dumb"。如果你需要其他的终端类型，请与我们联系并讨论如何最好地在 GRUB 里支持它。
The '-a' ('--ascii'), '-u' ('--utf8'), and '-v' ('--visual-utf8') options control how non-ASCII text is displayed. '-a' specifies an ASCII-only terminal; '-u' specifies logically-ordered UTF-8; and '-v' specifies "visually-ordered UTF-8" (in other words, arranged such that a terminal emulator without bidirectional text support will display right-to-left text in the proper order; this is not really proper UTF-8, but a workaround).
这里的"-a"("--ascii")，"-u"("--utf8")，和"-v"("--visual-utf8") 选项是控制如何去显示 非 ASCII 字符编码的文本。"-a"指定终端只使用 ASCII 字符编码；"-u"指定逻辑有序的 UTF-8；和"-v"指定"视觉有序的 UTF-8"(换言之，安排一个这样的没有双向文本支持的终端模拟器会以正确的次序显示从右到左书写格式的文本，虽然这不是真正彻底的支持 UTF-8，但不失为一个解决方案。)

If no option or terminal type is specified, the current terminal type is printed.
如果没有指定选项或终端类型，则会输出当前的终端类型。

14.3 The list of command-line and menu entry commands
14.3 列表出命令行和菜单条目的命令

These commands are usable in the command-line and in menu entries. If you forget a command, you can run the command 'help' (*note help).
这里的命令可用于命令行与菜单条目里。如果你忘记了一个命令，你可以运行命令"help"。

14.3.1 acpi
14.3.1 acpi

-- Command: acpi ['-1'|'-2']
['--exclude=table1,...'|'--load-only=table1,...']
['--oemid=id'] ['--oemtable=table'] ['--oemtablerev=rev']
['--oemtablecreator=creator'] ['--oemtablecreatorrev=rev']
['--no-ebda'] filename ...

Modern BIOS systems normally implement the Advanced Configuration and Power Interface (ACPI), and define various tables that describe the interface between an ACPI-compliant operating system and the firmware. In some cases, the tables provided by default only work well with certain operating systems, and it may be necessary to replace some of them.
现代的 BIOS 系统通常配有高级配置(Advanced Configuration)和电源接口(Power Interface-ACPI)的功能选项，并且定义了各种各样的表格去描述那些兼容 ACPI 的操作系统与固件之间的接口。在某些情况下，这些表格提供的默认值只能工作于某些特定的操作系统，而且它可能必需去更改其中的一些默认值。
Normally, this command will replace the Root System Description Pointer (RSDP) in the Extended BIOS Data Area to point to the new tables. If the '--no-ebda' option is used, the new tables will be known only to GRUB, but may be used by GRUB's EFI emulation.
通常地，这个命令将更改在 BIOS 扩展数据区域里的系统描述指针(RSDP)指向新表格。如果"--no-ebda"选项被使用上，这个新表格将被只作用于 GRUB，但它能被 GRUB 的 EFI 仿真使用。
外文引用：
根系统描述指针 - Root System Description Pointer (RSDP)：
兼容 ACPI 的系统必须在系统低地址空间（system's low address space）提供一个 RSDP。这种结构的唯一目的就是提供 RSDT 和 XSDT 的物理地址。

14.3.2 badram
14.3.2 badram

-- Command: badram addr,mask[,addr,mask...]

Filter out bad RAM.
过滤掉坏的 RAM(随机存取存储器)。
This command notifies the memory manager that specified regions of RAM ought to be filtered out (usually, because they're damaged). This remains in effect after a payload kernel has been loaded by GRUB, as long as the loaded kernel obtains its memory map from GRUB. Kernels that support this include Linux, GNU Mach, the kernel of FreeBSD and Multiboot kernels in general.
这个命令会通知内存管理器那个所指定区域的 RAM 应当被过滤掉(通常，因为它们损坏)。在有效内核被 GRUB 加载之后，余下的 RAM 将生效，只要被加载内核能从 GRUB 那里获取内存映射。通常情况下，支持这样内核包含 Linux，GNU Mach，基于 FreeBSD 的内核与基于 Multiboot 的内核。
Syntax is the same as provided by the Memtest86+ utility (http://www.memtest.org/): a list of address/mask pairs. Given a page-aligned address and a base address / mask pair, if all the bits of the page-aligned address that are enabled by the mask match with the base address, it means this page is to be filtered. This syntax makes it easy to represent patterns that are often result of memory damage,due

to physical distribution of memory cells.
语法与 Memtest86+工具所提供的相同(http://www.memtest.org/)：一个地址/掩码对的列表。给出一个页面-对齐的地址和一个基本的地址/掩码对，如果被掩码使能后的页面-对齐地址的所有的比特都能匹配上基本地址，意味这页面被过滤了。这种语法很容易就能表现出那些通常带有损坏的内存的模式，这归咎于物理分布结构的内存单元。

14.3.3 blocklist
14.3.3 blocklist

-- Command: blocklist file
Print a block list (*note Block list syntax) for FILE.
打印出 FILE 属性的块列表(注：块列表的语法-Block list syntax)。

14.3.4 boot
14.3.4 boot

-- Command: boot
Boot the OS or chain-loader which has been loaded. Only necessary if running the fully interactive command-line (it is implicit at the end of a menu entry).
引导已经被载入后的操作系统或链式-加载器。只有完全运行在交互式的命令行模式下它才是必要的(它是隐式地出现在菜单条目的末尾部分)。

14.3.5 cat
14.3.5 cat

-- Command: cat ['--dos'] file
Display the contents of the file FILE. This command may be useful to remind you of your OS's root partition:
显示 FILE 属性所指的文件里的内容。这个命令可能有效地提醒你你所处的系统根分区：

```
grub> cat /etc/fstab
```

If the '--dos' option is used, then carriage return / new line pairs will be displayed as a simple new line. Otherwise, the carriage return will be displayed as a control character ('<d>') to make it easier to see when boot problems are caused by a file formatted using DOS-style line endings.
如果"--dos"选项被用上，那么回车/换行符将会简单地视为换行符。否则，回车符将被显示为一个控制字符("<d>")这样就很容易发现当引导错误是因为一个文件格式使用了 DOS-风格的语句作结尾。

14.3.6 chainloader
14.3.6 chainloader

-- Command: chainloader ['--force'] file
Load FILE as a chain-loader. Like any other file loaded by the filesystem code, it can use the blocklist notation (*note Block list syntax) to grab the first sector of the current partition with '+1'. If you specify the option '--force', then load FILE forcibly, whether it has a correct signature or not. This is required when you want to load a defective boot loader, such as SCO UnixWare 7.1.
加载 FILE 所指的文件于链式-加载器。就像那些文件被文件系统代码加载一样，它能够使用块列表标记(注：块列表语法)去提取第一个分区的第一个扇区并且"+1"。如果你指定了选项"--force"，那么将强制地加载 FILE 所指的文件，无论其是否有正确的签名。这是必须的当你想要加载一个有缺陷的引导载入器的时候，例如 SCO UnixWare 7.1

14.3.7 cmp
14.3.7 cmp

-- Command: cmp file1 file2
Compare the file FILE1 with the file FILE2. If they differ in size, print the sizes like this:

以 FILE1 的文件内容去比较 FILE2 的文件内容。如果发现有不同的大小(判断为 ASCII 码比较，所以以整型数作大小对比)，则打印出它们大小的值：
Differ in size: 0x1234 [foo], 0x4321 [bar]
If the sizes are equal but the bytes at an offset differ, then print the bytes like this:
如果它们的大小相同，但是在某个偏移量处的字节不同，那么将打印出那些字节：
Differ at the offset 777: 0xbe [foo], 0xef [bar]
If they are completely identical, nothing will be printed.
如果它们完全一样，则没有什么会被打印出来。

14.3.8 configfile
14.3.8 configfile

-- Command: configfile file
Load FILE as a configuration file. If FILE defines any menu entries, then show a menu containing them immediately.
加载 FILE 所指定的配置文件。如果 FILE 定义为一个菜单项，则立即显示出一个新菜单去载入这个菜单项。

14.3.9 cpuid
14.3.9 cpuid

-- Command: cpuid [-l]
Check for CPU features. This command is only available on x86 systems.
检查 CPU 的特性。这个命令仅仅有效于 x86 的系统。
With the '-l' option, return true if the CPU supports long mode (64-bit).
如果加入"-l"选项，如果 CPU 支持长模式(64-bit)则会返回 true。
If invoked without options, this command currently behaves as if it had been invoked with '-l'. This may change in the future.
如果没有调用选项设置，此命令一般表现为设想它一直在调用着"-l"。这可能在将来有所改变。

14.3.10 crc
14.3.10 crc

-- Command: crc file
Display the CRC32 checksum of FILE.
显示 FILE 所指定的文件的 CRC32 校验和。

14.3.11 date
14.3.11 date

-- Command: date [[year-]month-day] [hour:minute[:second]]
With no arguments, print the current date and time.
并没有参数，打印出当前日期与时间。
Otherwise, take the current date and time, change any elements specified as arguments, and set the result as the new date and time. For example, 'date 01-01' will set the current month and day to January 1, but leave the year, hour, minute, and second unchanged.
另外，获得当前日期和时间，可以指定参数去改变所获取的任何一个元素的值，并且把这个结果设置为新的日期和时间。例如，"date 01-01"将月份和日期的结果设置为 1 月 1 日，但余下的年，小时，分钟，和秒则未被改变。

14.3.12 drivemap
14.3.12 drivemap

-- Command: drivemap '-l'|'-r'|['-s'] from_drive to_drive
Without options, map the drive FROM_DRIVE to the drive TO_DRIVE.This is necessary when you chain-

欢迎点击这里的链接进入精彩的Linux公社网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

Linux公社是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：**Ubuntu 专题** **Fedora 专题** **Android 专题** **Oracle 专题** **Hadoop 专题** **RedHat 专题** **SUSE 专题** **红旗 Linux 专题** **CentOS 专题**

Linux 公社微信公众号：linuxidc_com

load some operating systems, such as DOS, if such an OS resides at a non-first drive. For convenience, any partition suffix on the drive is ignored, so you can safely use ${root} as a drive specification.
没有选项，把 FROM_DRIVE 所指的驱动设备映射到 TO_DRIVE 所指的驱动位置。当你使用链式-加载器去加载那些操作系统的时候此命令是必须的，例如 DOS，又假如有这样一个 OS 它安装在非第一位置(non-first)的驱动器。为了方便起见，驱动器里所有的分区后缀都会被忽略，因此你才能安全地使用 ${root} 去定义一个驱动器。
With the '-s' option, perform the reverse mapping as well, swapping the two drives.
当使用"-s"选项，会执行反向映射，去交换这两个驱动器。
With the '-l' option, list the current mappings.
当使用"-l"选项，会列出当前的映射。
With the '-r' option, reset all mappings to the default values.
当使用"-r"选项，重置所有映射到默认的值。
For example:
例子：

```
drivemap -s (hd0) (hd1)
```

14.3.13 echo
14.3.13 echo

-- Command: echo ['-n'] ['-e'] string ...
Display the requested text and, unless the '-n' option is used, a trailing new line. If there is more than one string, they are separated by spaces in the output. As usual in GRUB commands,variables may be substituted using '${var}'.
显示出请求的文本，并且，除非"-n"选项被用上，才会有紧跟结尾的换行。如果有多个字符串，它们在输出时以空格符作分隔。正如普通的 GRUB 命令，变量可以使用"${var}"作替代。
The '-e' option enables interpretation of backslash escapes. The following sequences are recognised:
这个"-e"选项会开启反斜杠转义符的解释功能。下面将列出任何的：

'\\'
backslash
打印出单个反斜杠字符

'\a'
alert (BEL)
发出警告音

'\c'
suppress trailing new line
长按换行符

'\f'
form feed
执行换页动作

'\n'
new line
执行换行动作

'\r'
carriage return
执行回车符

'\t'
horizontal tab
执行水平制表符动作

'\v'
vertical tab
执行垂直制表符动作

When interpreting backslash escapes, backslash followed by any other character will print that character.
当对反斜杠转义符作解释时，紧跟反斜杠符后面的任何字符都将被以字符的形式打印出来。

14.3.14 export
14.3.14 export

-- Command: export envvar
Export the environment variable ENVVAR. Exported variables are visible to subsidiary configuration files loaded using 'configfile'.
导出环境变量 ENVVAR。导出的变量可见于使用了"configfile"命令加载的附属配置文件。

14.3.15 false
14.3.15 false

-- Command: false
Do nothing, unsuccessfully. This is mainly useful in control constructs such as 'if' and 'while' (*note Shell-like scripting).
不做任何事，返回非 0 的失败结果。这主要使用于控制结构语句上比如"if"和"while"(注：Shell-like scripting)。

14.3.16 gettext
14.3.16 gettext

-- Command: gettext string
Translate STRING into the current language.
转换 STRING 所指的字符串内容到当前语言。

The current language code is stored in the 'lang' variable in GRUB's environment (*note lang). Translation files in MO format are read from 'locale_dir' (*note locale_dir), usually '/boot/grub/locale'.
这个当前语言的代码是储存于 GRUB's 环境里的"lang"变量(注：lang)。于"locale_dir"所指定的位置中读取 MO 格式的转换翻译文件(注：locale_dir)，通常为"/boot/grub/locale"。

14.3.17 gptsync
14.3.17 gptsync

-- Command: gptsync device [partition[+/-[type]]] ...
Disks using the GUID Partition Table (GPT) also have a legacy Master Boot Record (MBR) partition table for compatibility with the BIOS and with older operating systems. The legacy MBR can only represent a limited subset of GPT partition entries.
磁盘使用 GUID 分区表(GPT)也可以支持传统的主引导记录(MBR)分区表因此可以兼容 BIOS 和一些比较老的操作系统。这传统的 MBR 能够单单表示为一个 GPT 分区条目的一个有限子集。
This command populates the legacy MBR with the specified PARTITION entries on DEVICE. Up to three partitions may be used.
这个命令以指定 DEVICE 驱动设备的 PARTITION 分区条目信息来填充成为传统的 MBR。最多可以填充三个分区。
TYPE is an MBR partition type code; prefix with '0x' if you want to enter this in hexadecimal. The separator between PARTITION and TYPE may be '+' to make the partition active, or '-' to make it inactive; only one partition may be active. If both the separator and type are omitted, then the partition will be

inactive.
TYPE 为一个 MBR 分区的类型代码；如果你需要输入十六进制数则要以"0x"为开头。在分隔 PARTITION 与 TYPE 之间的"+"号符表示分区为激活状态，或"-"号符表示为非激活状态；只能设置一个分区为激活状态。如果分隔符和 type 类型两者都被省略掉，那么这个分区将默认为非激活状态。

14.3.18 halt
14.3.18 halt

-- Command: halt '--no-apm'
The command halts the computer. If the '--no-apm' option is specified, no APM BIOS call is performed. Otherwise, the computer is shut down using APM.
这个命令为停机计算机。如果"--no-apm"选项被设置上，则没有 APM BIOS 的调用执行。另外，计算机是使用 APM 来实现关闭的。

14.3.19 help
14.3.19 help

-- Command: help [pattern ...]
Display helpful information about builtin commands. If you do not specify PATTERN, this command shows short descriptions of all available commands.
显示有关内置命令的帮助信息。如果你没有指定 PATTERN，这个命令显示所有可用命令的简短信息。
If you specify any PATTERNS, it displays longer information about each of the commands whose names begin with those PATTERNS.
如果你指定了 PATTERNS，它显示出带有 PATTERNS 所列出的名字为开头的所有命令。

14.3.20 initrd
14.3.20 initrd

-- Command: initrd file
Load an initial ramdisk for a Linux kernel image, and set the appropriate parameters in the Linux setup area in memory. This may only be used after the 'linux' command (*note linux::) has been run. See also *note GNU/Linux.
让一个 Linux 内核镜像去加载一个初始化内存虚拟盘，并且为处于内存区域里的 Linux 安装程序去设置适当的参数。这些必须只能在"linux"命令被使用运行之后才有效。

14.3.21 initrd16
14.3.21 initrd16

-- Command: initrd16 file
Load an initial ramdisk for a Linux kernel image to be booted in 16-bit mode, and set the appropriate parameters in the Linux setup area in memory. This may only be used after the 'linux16' command (*note linux16) has been run. See also *note GNU/Linux.
让一个 Linux 内核镜像使用 16bit 启动模式去加载一个初始化内存虚拟盘，并且为处于内存区域里的 Linux 安装程序去设置适当的参数。这些必须只能在"linux16"命令被使用运行之后才有效。(注：linux16 并且关注: GNU/Linux)
This command is only available on x86 systems.
这个命令只能有效于 x86 系统。

14.3.22 insmod
14.3.22 insmod

-- Command: insmod module
Insert the dynamic GRUB module called MODULE.
插入一个名称为“MODULE”的动态 GRUB 模块。

14.3.23 keystatus
14.3.23 keystatus

-- Command: keystatus ['--shift'] ['--ctrl'] ['--alt']
Return true if the Shift, Control, or Alt modifier keys are held down, as requested by options. This is useful in scripting, to allow some user control over behaviour without having to wait for a keypress.
如果 Shift、Control、或 Alt 修饰键被一直按下则会返回结果 true，通过命令的选项不同执行不同的请求。这命令可以完全适应于脚本，允许一些用户无须等待键盘按键来控制行为。
Checking key modifier status is only supported on some platforms. If invoked without any options, the 'keystatus' command returns true if and only if checking key modifier status is supported.
检查修饰键的状态功能仅仅在某些平台上支持。如果调用时没有加任何选项，当且仅当地使用检查修饰键状态的功能时，这个"keystatus"(键盘状态)命令会返回 true。

14.3.24 linux
14.3.24 linux

-- Command: linux file ...
Load a Linux kernel image from FILE. The rest of the line is passed verbatim as the "kernel command-line". Any initrd must be reloaded after using this command (*note initrd).
从 FILE 所指定的路径里去加载那个指定的 Linux 内核镜像。其余的文字信息一字不差地作为"内核命令行"的命令。使用了这个命令之后，任何的 initrd 都必须重新装载才能被使用。
On x86 systems, the kernel will be booted using the 32-bit boot protocol. Note that this means that the 'vga=' boot option will not work; if you want to set a special video mode, you will need to use GRUB commands such as 'set gfxpayload=1024x768' or 'set gfxpayload=keep' (to keep the same mode as used in GRUB) instead. GRUB can automatically detect some uses of 'vga=' and translate them to appropriate settings of 'gfxpayload'. The 'linux16' command (*note linux16) avoids this restriction.
在 x86 的系统，内核将使用 32bit 的引导协议去启动。注意这是意味着"vga="引导选项将不能工作；如果你希望去设置一个特定视频模式，你将需要去使用 GRUB 命令例如"set gfxpayload=1024x768"或"set gfxpayload=keep"(去保持一些模式能被应用到 GRUB 里)来取代。GRUB 能自动地检测出"vga="并把它们转换为适当的"gfxpayload"设置。这"linux16"避免了这个限制(注：linux16)。

14.3.25 linux16
14.3.25 linux16

-- Command: linux16 file ...
Load a Linux kernel image from FILE in 16-bit mode. The rest of the line is passed verbatim as the "kernel command-line". Any initrd must be reloaded after using this command (*note initrd16).
以 16bit 的模式从 FILE 所指定的位置去加载一个 Linux 内核镜像。剩余的文字信息一字不差地作为"内核命令行"的命令。当使用了 linux16 命令之后任何的 initrd 都必须被重新加载才能被使用(注：initrd16)。
The kernel will be booted using the traditional 16-bit boot protocol. As well as bypassing problems with 'vga=' described in *note linux, this permits booting some other programs that implement the Linux boot protocol for the sake of convenience.
内核将使用传统的 16bit 引导协议去启动。以及很好地绕过"vga="内核参数无效的问题(细节于注：linux)，为了方便起见而允许启动一些其他的程序去实现 Linux 的引导协议。
This command is only available on x86 systems.
这个命令只能有效于 x86 架构的系统。

14.3.26 list_env
14.3.26 list_env

-- Command: list_env ['-f' file]
List all variables in the environment block file. *Note Environment block.
列出所有的环境区块文件的变量名。注：环境区块。

The '-f' option overrides the default location of the environment block.
这"-f"的选项将会重写环境区块的默认位置。

14.3.27 load_env
14.3.27 load_env

-- Command: load_env ['-f' file]
Load all variables from the environment block file into the environment. *Note Environment block.
加载所有环境区块文件的变量到运行环境里。注：环境区块。
The '-f' option overrides the default location of the environment block.
这"-f"的选项将会重写环境区块的默认位置。

14.3.28 loopback
14.3.28 loopback

-- Command: loopback ['-d'] device file
Make the device named DEVICE correspond to the contents of the filesystem image in FILE. For example:
构建名为 DEVICE，并且内容完全相符于文件系统镜像 FILE 的设备。例如：

```
loopback loop0 /path/to/image
ls (loop0)/
```

With the '-d' option, delete a device previously created using this command.
使用"-d"选项，会先删除之前使用过此命令所创建出的设备。

14.3.29 ls
14.3.29 ls

-- Command: ls [arg ...]
List devices or files.
列出设备或者文件。
With no arguments, print all devices known to GRUB.
不带参数时，打印出所有 GRUB 已知的设备。
If the argument is a device name enclosed in parentheses (*note Device syntax), then list all files at the root directory of that device.
如果参数为一个带括号的设备名字(注：设备的语法)，那么则列示出该设备根目录下的所有文件。
If the argument is a directory given as an absolute file name (*note File name syntax), then list the contents of that directory.
如果参数为给出绝对路径的一个目录，那么则会列显出该目录里的内容。

14.3.30 normal
14.3.30 normal

-- Command: normal [file]
Enter normal mode and display the GRUB menu.
进入正常模式并且显示出 GRUB 菜单。
In normal mode, commands, filesystem modules, and cryptography modules are automatically loaded, and the full GRUB script parser is available. Other modules may be explicitly loaded using 'insmod' (*note insmod).
在正常模式里，所有的命令，所有的文件系统模块，和加密模块都会被自动加载，并且所有的 GRUB 脚本解析器都是可用的。其他模块可以显式地使用"insmod"命令去加载使用(注：insmod)。
If a FILE is given, then commands will be read from that file. Otherwise, they will be read from '$prefix/grub.cfg' if it exists.
如果 FILE 参数给赋值，那么命令将会去读取那个文件。否则，它们将会去读取"$prefix/grub.cfg"文件，如

果它是存在的。
'normal' may be called from within normal mode, creating a nested environment. It is more usual to use 'configfile' (*note configfile) for this.
"normal"命令可以在正常模式下被调用执行，创建一个嵌套的环境。在这种情况下通过此命令使用"configfile"更加平常。

14.3.31 normal_exit
14.3.31 normal_exit

-- Command: normal_exit
Exit normal mode (*note normal). If this instance of normal mode was not nested within another one, then return to rescue mode.
退出正常模式(注：normal)。如果这个正常模式的实例不是嵌套在另一个正常模式里，那么将返回到救援模式。

14.3.32 parttool
14.3.32 parttool

-- Command: parttool partition commands
Make various modifications to partition table entries.
对分区表项去进行各种修改。
Each COMMAND is either a boolean option, in which case it must be followed with '+' or '-' (with no intervening space) to enable or disable that option, or else it takes a value in the form 'COMMAND=VALUE'.
要么每个 COMMAND 都是一个布尔值的选项，在这种情况下它必须遵循"+"或者"-"去实现那个选项的有效或者无效(不能带有空格)，要么它会从"COMMAND=VALUE"里取出一个值。
Currently, 'parttool' is only useful on DOS partition tables (also known as Master Boot Record, or MBR). On these partition tables, the following commands are available:
目前，"parttool"只能唯一使用于 DOS 分区表(也被称为主引导记录，或 MBR)。在这些分区表里，以下的命令是可用的：
'boot' (boolean)
When enabled, this makes the selected partition be the active (bootable) partition on its disk, clearing the active flag on all other partitions. This command is limited to _primary_ partitions.
"boot"(boolean)
'boot' (boolean)
如果启用，这使得所选择的分区在其磁盘里为活动(可引导的)分区，并且清除掉其他所有分区里的活动标记。这个命令仅限于主分区上使用。
'type' (value)
Change the type of an existing partition. The value must be a number in the range 0-0xFF (prefix with '0x' to enter it in hexadecimal).
'type' (value)
修改现有的分区类型。其值必须在 0-0xFF 范围内的数字(特定的数字代表特定的类型)(使用十六进制需要加入"0x"前缀)。
'hidden' (boolean)
When enabled, this hides the selected partition by setting the "hidden" bit in its partition type code; when disabled, unhides the selected partition by clearing this bit. This is useful only when booting DOS or Windows and multiple primary FAT partitions exist in one disk. See also *note DOS/Windows.
"hidden"(boolean)
如果为有效的，这将会在被选择分区的分区类型码中设置“hidden”标志，以此来隐藏掉选择的分区。如果为无效的，通过清除这个 bit 值可以取消隐藏指定分区。这仅有效于当在一个硬盘上存在多个 FAT 文件系统主分区时实现去启动 DOS 或 Windows。另请参见，注：DOS/Windows。

14.3.33 password
14.3.33 password

-- Command: password user clear-password
Define a user named USER with password CLEAR-PASSWORD. *Note Security.
定义一个用户名 USER 并通过 CLEAR-PASSWORD 来设置密码(以明文的方式)。注：security。

14.3.34 password_pbkdf2
14.3.34 password_pbkdf2

-- Command: password_pbkdf2 user hashed-password
Define a user named USER with password hash HASHED-PASSWORD. Use 'grub-mkpasswd-pbkdf2' (*note Invoking grub-mkpasswd-pbkdf2) to generate password hashes. *Note Security.
定义一个用户名 USER 并通过 HASHED-PASSWORD 来设置哈希密码。使用"grub-mkpasswd-pbkdf2"去生成出哈希密码(注：调用 grub-mkpasswd-pbkdf2)。注：security。

14.3.35 play
14.3.35 play

-- Command: play file | tempo [pitch1 duration1] [pitch2 duration2] ...
Plays a tune
播放一个曲子
If the argument is a file name (*note File name syntax), play the tune recorded in it. The file format is first the tempo as an unsigned 32bit little-endian number, then pairs of unsigned 16bit little-endian numbers for pitch and duration pairs.
如果参数为一个文件名(注：File name syntax)，则会播放此文件里记录的曲子。这里的文件格式里第一个 tempo(拍子)参数的值是无符号 32bit 小端方式(从最小那端开始数)的数值，然后配对于一双无符号的 16bit 小端方式的数值去表示参数 pitch(音值)和 duration(音延)的值。
If the arguments are a series of numbers, play the inline tune.
如果参数是一系列的数字，则播放内置的曲子。
The tempo is the base for all note durations. 60 gives a 1-second base, 120 gives a half-second base, etc. Pitches are Hz. Set pitch to 0 to produce a rest.
tempo(拍子)是所有 duration(音延)的基准，60 给出以 1 秒为基准，120 给出以半秒为基准，以此类推。pitch 的值为 Hz(音频)。设置 pitch(音值)的值为 0 则会静音。

14.3.36 pxe_unload
14.3.36 pxe_unload

-- Command: pxe_unload
Unload the PXE environment (*note Network).
卸载 PXE 环境(注：Network)。
This command is only available on PC BIOS systems.
这个命令只能有效于 PC BIOS 系统。

14.3.37 read
14.3.37 read

-- Command: read [var]
Read a line of input from the user. If an environment variable VAR is given, set that environment variable to the line of input that was read, with no terminating newline.
读取一行来自用户的输入。如果一个环境变量 VAR 被赋值，则把那个环境变量的输入设置为所读入的行，不带结尾的换行符。

14.3.38 reboot
14.3.38 reboot

-- Command: reboot
Reboot the computer.
重新启动计算机。

14.3.39 save_env
14.3.39 save_env

-- Command: save_env ['-f' file] var ...
Save the named variables from the environment to the environment block file. *Note Environment block.
保存指定的环境变量到环境块文件中。注：环境块-Environment block。
The '-f' option overrides the default location of the environment block.
这"-f"选项会重写环境块的默认位置。

14.3.40 search
14.3.40 search

-- Command: search ['--file'|'--label'|'--fs-uuid'] ['--set' [var]] ['--no-floppy'] name
Search devices by file ('-f', '--file'), filesystem label ('-l','--label'), or filesystem UUID ('-u', '--fs-uuid').
通过搜索该设备里的文件名("-f"，"--file")，文件系统的标签("-l"，"--label")，或者文件系统的 UUID("-u"，"--fs-uuid")来搜索确认该设备。
If the '--set' option is used, the first device found is set as the value of environment variable VAR. The default variable is 'root'.
如果"--set"选项被使用上，那么第一个被找到的设备会作为目标设备名被设为 VAR 环境变量里的值。而它默认的值为"root"。
The '--no-floppy' option prevents searching floppy devices, which can be slow.
这个"--no-floppy"选项会阻止搜索软驱设备，因它可能会很慢。
The 'search.file', 'search.fs_label', and 'search.fs_uuid' commands are aliases for 'search --file', 'search --label', and 'search --fs-uuid' respectively.
这些"search.file"，"search.fs_label"，和"search.fs_uuid"命令各自分别是"search --file"，"search --label"，和"search --fs-uuid"的别名。

14.3.41 sendkey
14.3.41 sendkey

-- Command: sendkey ['--num'|'--caps'|'--scroll'|'--insert'|'--pause'|'--left-shift'|'--right-shift'|'--sysrq'|'--numkey'|'--capskey'|'--scrollkey'|'--insertkey'|'--left-alt'|'--right-alt'|'--left-ctrl'|'--right-ctrl' 'on'|'off']... ['no-led'] keystroke
Insert keystrokes into the keyboard buffer when booting. Sometimes an operating system or chainloaded boot loader requires particular keys to be pressed: for example, one might need to press a particular key to enter "safe mode", or when chainloading another boot loader one might send keystrokes to it to navigate its menu.
当启动的时候输入指定的击键值到键盘缓冲区里。有时候在去加载一个操作系统或去加载一个链式加载引导时需要按下一个特殊的键来实现：例如，一个可能需要去按下特殊键才会进入的"安全模式"，或者当要链式加载另一个引导加载器时它有可能需要发送一个击键来进入它的导航菜单。
You may provide up to 16 keystrokes (the length of the BIOS keyboard buffer). Keystroke names may be upper-case or lower-case letters, digits, or taken from the following table:
你最多可以提供 16 个按键(这取决于 BIOS 的键盘缓冲区大小)。击键名称可以是大写或小写的单个字母，数字，或者以下列表所列出的：

Name	Key
escape	Escape
exclam	!
at	@
numbersign	#
dollar	$
percent	%
caret	^
ampersand	&
asterisk	*
parenleft	(
parenright	)
minus	-
underscore	_
equal	=
plus	+
backspace	Backspace
tab	Tab
bracketleft	[
braceleft	{
bracketright	]
braceright	}
enter	Enter
control	press and release Control
semicolon	;
colon	:
quote	'
doublequote	"
backquote	`
tilde	~
shift	press and release left Shift
backslash	\
bar	|
comma	,
less	<
period	.
greater	>
slash	/
question	?
rshift	press and release right Shift
alt	press and release Alt
space	space bar
capslock	Caps Lock
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9
F10	F10

F11	F11
F12	F12
num1	1 (numeric keypad)
num2	2 (numeric keypad)
num3	3 (numeric keypad)
num4	4 (numeric keypad)
num5	5 (numeric keypad)
num6	6 (numeric keypad)
num7	7 (numeric keypad)
num8	8 (numeric keypad)
num9	9 (numeric keypad)
num0	0 (numeric keypad)
numperiod	. (numeric keypad)
numend	End (numeric keypad)
numdown	Down (numeric keypad)
numpgdown	Page Down (numeric keypad)
numleft	Left (numeric keypad)
numcenter	5 with Num Lock inactive (numeric keypad)
numright	Right (numeric keypad)
numhome	Home (numeric keypad)
numup	Up (numeric keypad)
numpgup	Page Up (numeric keypad)
numinsert	Insert (numeric keypad)
numdelete	Delete (numeric keypad)
numasterisk	* (numeric keypad)
numminus	- (numeric keypad)
numplus	+ (numeric keypad)
numslash	/ (numeric keypad)
numenter	Enter (numeric keypad)
delete	Delete
insert	Insert
home	Home
end	End
pgdown	Page Down
pgup	Page Up
down	Down
up	Up
left	Left
right	Right

As well as keystrokes, the 'sendkey' command takes various options that affect the BIOS keyboard status flags. These options take an 'on' or 'off' parameter, specifying that the corresponding status flag be set or unset; omitting the option for a given status flag will leave that flag at its initial state at boot. The '--num', '--caps', '--scroll', and '--insert' options emulate setting the corresponding mode, while the '--numkey', '--capskey', '--scrollkey', and '--insertkey' options emulate pressing and holding the corresponding key. The other status flag options are self-explanatory.

除了和击键有关，"sendkey"命令也通过获取各种选项从而影响 BIOS 的键盘状态标志。而这些选项只需一个 "on" 或者 "off" 参数，用以指定那个相应的状态标记为 被设置 或者 未设置；只有选项而省略掉选项里的状态标记参数则将保留选项在引导时的原始状态标记。这里的"--num"，"--caps"，"--scroll"，和"--insert"选项是以仿真的方式去实现相应的模式，而这里的"--numkey"，"--capskey"，"--scrollkey"，和"--insertkey"选项是仿真对那些相应的键的长按。那些其他的状态标记选项都是不言自明的。

If the '--no-led' option is given, the status flag options will have no effect on keyboard LEDs.

如果"—no-led"选项被给定，那么状态标记选项将不会影响到键盘的 LED 灯。

If the 'sendkey' command is given multiple times, then only the last invocation has any effect.
如果"sendkey"命令被重复调用使用，那么只有在最后调用的那次才被执行起效。
Since 'sendkey' manipulates the BIOS keyboard buffer, it may cause hangs, reboots, or other misbehaviour on some systems. If the operating system or boot loader that runs after GRUB uses its own keyboard driver rather than the BIOS keyboard functions, then 'sendkey' will have no effect.
由于"sendkey"操纵了 BIOS 的键盘缓冲区，它可能会导致某些系统的挂起(挡机、死机)，重新启动或一些意外行为。如果那些操作系统或引导加载器在那个 GRUB 运行之后去使用它们各自的键盘驱动而不是 BIOS 提供的键盘函数，那么"senkey"将会毫无影响。
This command is only available on PC BIOS systems.
这个命令只有效于 PC BIOS 系统。

14.3.42 set
14.3.42 set

-- Command: set [envvar=value]
Set the environment variable ENVVAR to VALUE. If invoked with no arguments, print all environment variables with their values.
把 VALUE 的值赋给 ENVVAR 环境变量。如果直接调用而不带任何参数，则会打印出所有环境变量以及它们的值。

14.3.43 true
14.3.43 true

-- Command: true
Do nothing, successfully. This is mainly useful in control constructs such as 'if' and 'while' (*note Shell-like scripting).
什么都不做，直接完成。这主要用于控制结构里比如"if"和"while"(注：Shell-like scripting)。

14.3.44 unset
14.3.44 unset

-- Command: unset envvar
Unset the environment variable ENVVAR.
释放 ENVVAR 环境变量的值。

14.3.45 uppermem
14.3.45 uppermem

This command is not yet implemented for GRUB 2, although it is planned.
这个命令在 GRUB2 里还没有实现，尽管它已在计划之内。
外文引用：
uppermem 能够把仿真镜像从内存的高位开始排列存放，从而不影响低位内存的使用。

15.字符集

15.字符集-Charset

********** **********

15 Charset

15 字符集

==========

GRUB uses UTF-8 internally other than in rendering where some GRUB-specific appropriate representation is used. All text files (including config) are assumed to be encoded in UTF-8.

GRUB 在内部使用 UTF-8 字符集，除此之外，在某些地方使用 GRUB 特定的、合适的表示方法。 所有的文本文件（包括配置文件）都采用 UTF-8 方式编码。

16.文件系统

16.文件系统-Filesystems

********** **********

16 Filesystems

16 文件系统

==========

NTFS, JFS, UDF, HFS+, exFAT, long filenames in FAT, Joliet part of ISO9660 are treated as UTF-16 as per specification.

NTFS,JFS,UDF,HFS+,exFAT,FAT(支持长文件名),ISO9660 的 Joliet 扩展都把 UTF-16 当作编码规范。

AFS and BFS are read as UTF-8, again according to specification. BtrFS, cpio, tar,squash4, minix, minix2, minix3, ROMFS, ReiserFS, XFS, ext2, ext3, ext4,FAT (short names), RockRidge part of ISO9660, nilfs2, UFS1, UFS2 and ZFS are assumed to be UTF-8.

根据规范，AFS 和 BFS 都支持 UTF-8 编码。

BtrFS,cpio,tar,squash4,minix,minix2,minix3,ROMFS,ReiserFS,XFS,ext2,ext3,ext4,FAT(支持短文件名),ISO9660 的 RockRidge 扩展,nilfs2,UFS2 和 ZFS 都采取 UTF-8 编码规范。

This might be false on systems configured with legacy charset but as long as the charset used is superset of ASCII you should be able to access ASCII-named files.

如果系统以传统字符集作为配置，以上的结论可能是错误的。但是，只要所使用的字符集是 ASCII 码的超集，那么你就能存取 ASCII 码命名的文件。

And it's recommended to configure your system to use UTF-8 to access the filesystem, convmv may help with migration. ISO9660 (plain) filenames are specified as being ASCII or being described with unspecified escape sequences. GRUB assumes that the ISO9660 names are UTF-8 (since any ASCII is valid UTF-8).

并且，系统建议以 UTF-8 编码方式来配置文件存取系统，命令 convmv 能帮助实现转换。ISO9660(plain)文件名用 ASCII 码表示或者用未指定转义序列描述。GRUB 认为 ISO9660 编码的名称属于 UTF-8 编码。（因为任何 ASCII 编码都是有效的 UTF-8 编码）。

There are some old CD-ROMs which use CP437 in non-compliant way. You're still able to access files with names containing only ASCII characters on such filesystems though. You're also able to access any file if the filesystem contains valid Joliet (UTF-16) or RockRidge (UTF-8). AFFS, SFS and HFS never use unicode and GRUB assumes them to be in Latin1, Latin1 and MacRoman respectively.

现在仍然有一些旧的 CD-ROMs 不服从规定而使用 CP437 方式编码。尽管如此，在这样的文件系统中，仍然可以存取那些文件名只包含 ASCII 码的文件。如果文件系统包含无效的 Joliet(UTF-16)或者 RockRidge(UTF-8)，就可以存取其中的任何文件。AFFS,SFS 和 HFS 不使用 unicode 字符集，并且 GRUB 认为它们分别使用 Latin1,Latin1 和 MacRoman 字符集。

GRUB handles filesystem case-insensitivity however no attempt is performed at case conversion of international characters so e.g. a file named lowercase greek alpha is treated as different from the one named as uppercase alpha. The filesystems in questions are NTFS (except POSIX namespace), HFS+ (configurable at mkfs time, default insensitive), SFS (configurable at mkfs time, default insensitive), JFS (configurable at mkfs time, default sensitive), HFS, AFFS, FAT, exFAT and ZFS (configurable on per-subvolume basis by property "casesensitivity", default sensitive). On ZFS subvolumes marked as case insensitive files containing lowercase
international characters are inaccessible. Also like all supported filesystems except HFS+ and ZFS (configurable on per-subvolume basis by property "normalization", default none) GRUB makes no attempt at check of canonical equivalence so a file name u-diaresis is treated as distinct from u+combining diaresis. This however means that in order to access file on HFS+ its name must be specified in normalisation form D. On normalized ZFS subvolumes filenames out of normalisation are inaccessible.

GRUB 处理文件系统不区分大小写，但是，国际字符的大小写变换不该作这样的假设。以小写希腊字母命名的文件不同于以大写希腊字母命名的文件。正在讨论和测试的文件系统有 NTFS（除了 POSIX 命名空间），HSF+（在文件系统建立时可配置，默认不区分大小写），SFS（在文件系统建立时可配置，默认不区分大小写），JFS（在文件系统建立时可配置，默认区分大小写），HFS，AFFS，FAT，exFAT 与 ZFS（由于大小写敏感的特性，可在子卷建立时配置，默认区分大小写）。标记为大小写不敏感的 ZFS 子卷中包含小写的国际字符的文件是不能存取的。除了 HFS+和 ZFS（由于标准化的特性，可在子卷建立时配置，默认无），其他所有被支持的文件系统，GRUB 不尝试检查规范等价性，所以，包含 u-diaresis 字符的文件名是区别于包含 u+combining diaresis 字符的。但是，这意味着，为了在 HFS+中存取文件，它的文件名必须符合规范格式 D。在标准化的 ZFS 子卷中未标准化的文件名是不能存取的。

17.输出终端

17.输出终端-Output terminal

********** **********

17 Output terminal

17 输出终端

==========

Firmware output console "console" on ARC and IEEE1275 are limited to ASCII. BIOS firmware console and VGA text are limited to ASCII and some pseudographics. None of above mentioned is appropriate for displaying international and any unsupported character is replaced with question mark except pseudographics which we attempt to approximate with ASCII. EFI console on the other hand nominally supports UTF-16 but actual language coverage depends on firmware and may be very limited. The encoding used on serial can be chosen with 'terminfo' as either ASCII, UTF-8 or "visual UTF-8". Last one is against the specification but results in correct rendering of right-to-left on some readers which don't have own bidi implementation. When using gfxterm or gfxmenu GRUB itself is responsible for rendering the text. In this case GRUB is limited by loaded fonts. If fonts contain all required characters then bidirectional text, cursive variants and combining marks other than enclosing, half (e.g. left half tilde or combining overline) and double ones. Ligatures aren't supported though. This should cover European, Middle Eastern (if you don't mind lack of lam-alif ligature in Arabic) and East Asian scripts. Notable unsupported scripts are Brahmic family and derived as well as Mongolian, Tifinagh, Korean Jamo (precomposed characters have no problem) and tonal writing (2e5-2e9). GRUB also ignores deprecated (as specified in Unicode) characters (e.g. tags).
GRUB also doesn't handle so called "annotation characters" If you can complete either of two lists or, better, propose a patch to improve rendering, please contact developper team.

ARC 和 IEEE1275 的固件输出控制台“console”被限制为只能使用 ASCII。BIOS 固件控制台和 VGA text 被限制为使用 ASCII 和一些伪图。以上提到的任何一个中，没有一个适合显示国际的字符和任何不被支持的字符，它们会被问号代替，除了那些用 ASCII 码近似的伪图。另一方面，EFI 控制台名义上支持 UTF-16，但是，事实上的字符编码方式的覆盖范围依赖于固件，并且是很有限的。当使用串口时，“terminfo”编码时，会选择 ASCII，UTF-8 或“visual UTF-8”三个中的其中一个。最后一个是违反规范的，然而在那些自己没有 bidi 实现方式的读取设备上导致正确的 right-to-left 渲染方式。当 GRUB 使用 gfxterm 方式或 gfxmenu 方式时，会负责渲染文本。这种情况下，GRUB 会被启动时所加载的字体限制。如果字体中包含所有需要的字符，那么 bidirectional text,cursive variants,combining marks other than enclosing,half(e.g.left half tilde or combining overline)and double ones.因此，连体字不被支持。这些规则覆盖了欧洲、中东（如果你不介意阿拉伯缺乏 lam-alif ligature）、东亚的书写方式。著名的不被支持的书写方式是 Brahmic 家族及其派生的 Mongolian,Tifinagh,Koream Jamo（也包含复合字符）和 tonal writing(2e5-2e9)。GRUB 忽略过时（在 Unicode 中特殊说明）的字符（例如标签）。GRUB 也不处理所谓的标记符，如果你可以解决 two list 中任何一个，或者更好，提出一个补丁来改善渲染效果，请联系开发人员。

18.输入终端

18.输入终端-Input terminal

********** **********

18 Input terminal

18 输入终端

==========

Firmware console on BIOS, IEEE1275 and ARC doesn't allow you to enter non-ASCII characters. EFI specification allows for such but author is unaware of any actual implementations. Serial input is currently limited for latin1 (unlikely to change). Own keyboard implementations (at_keyboard and usb_keyboard) supports any key but work on one-char-per-keystroke. So no dead keys or advanced input method. Also there is no keymap change hotkey. In practice it makes difficult to enter any text using non-Latin alphabet. Moreover all current input consumers are limited to ASCII.

BIOS，IEEE1275 和 ARC 上的固件终端不允许输入非 ASCII 字符。EFI 规范也有这样的允许，但是作者对于任何真正的实现一无所知。目前串行输入被限制为 latin1（不太可能改变）。特有的键盘的实现（AT 键盘和 USB 键盘）支持任何键，并且工作时一次击键一个字符。所以没有修饰键或高级输入方式。并且当前也没有改变快捷键的键盘映射。实际上，使用非 Latin 字母表输入任何文本是困难的。而且，目前所有的输入仅限于 ASCII 码。

19.文本语言

19.文本语言-Gettext

********** ********************************

19 Gettext

19 获取文字类型(文本语言)

==========

GRUB supports being translated. For this you need to have language *.mo files in $prefix/locale, load gettext module and set "lang" variable.

GRUB 支持其他语言。为了使用它，你需要在文件夹$perfix/locale 中有 language*.mo 文件，加载 gettext 模块和设置"lang"变量。

20.正则表达式

20.正则表达式-Regexp

********* **********

20 Regexp

20 正则表达式

==========

Regexps work on unicode characters, however no attempt at checking cannonical equivalence has been made. Moreover the classes like [:alpha:] match only ASCII subset.

正则表达式 unicode 字符环境下工作，然而，不会尝试对表达式做规范等价性的检查。此外，像[:alpha:]这样的等价类只会匹配 ASCII 码的子集。

21.其他

21.其他-Other

******** **********

21 Other

21 其他

==========

Currently GRUB always uses YEAR-MONTH-DAY HOUR:MINUTE:SECOND [WEEKDAY] 24-hour datetime format but weekdays are translated. GRUB always uses the decimal number format with [0-9] as digits and . as descimal separator and no group separator. IEEE1275 aliases are matched case-insensitively except non-ASCII which is matched as binary. Similar behaviour is for matching OSBundleRequired. Since IEEE1275 aliases and OSBundleRequired don't contain any non-ASCII it should never be a problem in practice. Case-sensitive identifiers are matched as raw strings, no canonical equivalence check is performed. Case-insenstive identifiers are matched as RAW but additionally [a-z] is equivalent to [A-Z]. GRUB-defined identifiers use only ASCII and so should user-defined ones. Identifiers containing non-ASCII may work but aren't supported. Only the ASCII space characters (space U+0020, tab U+000b, CR U+000d and LF U+000a) are recognised. Other unicode space characters aren't a valid field separator. 'test' tests <, >, <=, >=, -pgt and -plt compare the strings in the lexicographical order of unicode codepoints, replicating the behaviour of test from coreutils. environment variables and commands are listed in the same order.

现在，GRUB 使用 YEAR-MONTH-DAY HOUR：MINUTE：SECOND[WEEKDAY]、24 小时格式，但是，周工作日被转换了。GRUB 使用[0-9]10 个十进制数字，“.”作为小数的分隔符，并且没有分组符。IEEE1275 别名匹配时不区分大小写，除了那些用来匹配二进制的非 ASCII 字符。类似的行为在匹配属性 OSBundleRequired 时也存在。由于 IEEE1275 别名和属性 OSBundleRequired 不包含任何非 ASCII 字符，因此在实际中不会有问题。大小写敏感的标记符匹配原字符串，并且不执行规范等价性检查。大小写不敏感的标记符也匹配原字符串，并且认为[a-z]与[A-Z]是相同的。GRUB 定义的标记符只使用 ASCII 码，用户定义的标记符也是如此。包含非 ASCII 码的标记符也能使用，但是不建议使用。仅仅 ASCII 码的空白字符（space U+0020,tab U+000b,CR U+000d 和 LF U+000a）能被识别。其他 unicode 空白字符不是合法的区域分隔符。test 命令加上<,>,<=,>=,-pgt 和 -plt 选项 ，以 unicode 码点在字典中的顺序去比较字符串，它的行为和 GNU 下核心工具包中 test 命令的行为一样。环境变量和命令的列出同样以这种顺序列出。

22.身份验证与授权

22.身份验证与授权-Authentication and authorisation

********** **********

22 Authentication and authorisation

22 身份验证与授权

==========

By default, the boot loader interface is accessible to anyone with physical access to the console: anyone can select and edit any menu entry, and anyone can get direct access to a GRUB shell prompt. For most systems, this is reasonable since anyone with direct physical access has a variety of other ways to gain full access, and requiring authentication at the boot loader level would only serve to make it difficult to recover broken systems.

默认情况下，启动加载器接口对于那些任何物理上可以访问终端的人是可以使用的。那些人可以选择和编辑任何入口菜单，并且可以直接进入一个 GRUB shell。对于大多数系统，这样做的理由是，由于任何物理上可以访问系统的人有各种各样的其他方法进入系统，并且在启动加载器的水平上的请求身份验证仅仅会使得当要恢复被损坏的系统时更加困难。

However, in some environments, such as kiosks, it may be appropriate to lock down the boot loader to require authentication before performing certain operations.

然而，在一些环境中，例如 kiosks，在确定的操作之前，锁定启动加载器来要求身份验证是合适的。

The 'password' (*note password::) and 'password_pbkdf2' (*note password_pbkdf2::) commands can be used to define users, each of which has an associated password. 'password' sets the password in plain text,requiring 'grub.cfg' to be secure; 'password_pbkdf2' sets the password hashed using the Password-Based Key Derivation Function (RFC 2898),requiring the use of 'grub-mkpasswd-pbkdf2' (*note Invoking grub-mkpasswd-pbkdf2::) to generate password hashes.

“password”和“password_pbkdf2”命令被用来定义用户，每个都得提供一个相关联的密码。“password”命令以明文的方式设置密码，且需要“grub.cfg”来保证安全。“password_pbkdf2”命令设置一个用 Password-Based Key Derivation Function(RFC2898)生成的密码，且需要使用“grub-mkpasswd-pbkdf2”来生成密码散列值。

In order to enable authentication support, the 'superusers' environment variable must be set to a list of usernames, separated by any of spaces, commas, semicolons, pipes, or ampersands. Superusers are permitted to use the GRUB command line, edit menu entries, and execute any menu entry. If 'superusers' is set, then use of the command line is automatically restricted to superusers.

为了使用身份验证，环境变量“superusers”必须被设置为一个用户名的列表，用户名以空格、逗号、分号、管道线或者逻辑与符号。超级用户被允许使用 GRUB 命令行，编辑入口菜单，并且可以选择任何入口。如果环境变量“superusers”被设置了，命令行的使用会被自动限制只允许超级用户。

Other users may be given access to specific menu entries by giving a list of usernames (as above) using the '--users' option to the 'menuentry' command (*note menuentry::). If the '--unrestricted' option is used for a menu entry, then that entry is unrestricted. If the '--users' option is not used for a menu entry, then that only superusers are able to use it.

命令“menuentry”使用“--users”选项来提供用户名列表（如上）来给予用户对当前入口菜单的权限。如果为入口菜单使用“--unrestricted”选项为菜单设置为不受限制。如果入口菜单没有用选项“--users”设置，那么入口只有超级用户可以使用。

Putting this together, a typical 'grub.cfg' fragment might look like this:
综上所述，一个典型的"grub.cfg"片段看起来应该像这样：

```
set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.biglongstring
password user1 insecure

menuentry "May be run by any user" --unrestricted {
  set root=(hd0,1)
  linux /vmlinuz
}

menuentry "Superusers only" --users "" {
  set root=(hd0,1)
  linux /vmlinuz single
}

menuentry "May be run by user1 or a superuser" --users user1 {
  set root=(hd0,2)
  chainloader +1
}
```

The 'grub-mkconfig' program does not yet have built-in support for generating configuration files with authentication. You can use '/etc/grub.d/40_custom' to add simple superuser authentication, by adding 'set superusers=' and 'password' or 'password_pbkdf2' commands.
这个"grub-mkconfig"程序还没有内置地去生成出那个支持身份验证的配置文件。你可以使用"/etc/grub.d/40_custom"去简单地添加一个超级用户的身份验证，也可以通过 "set superusers=" 和 "password" 或 "password_pbkdf2" 命令来添加。

23.平台限制

23.平台限制-Platform limitations

********** **********

23 Platform limitations

23 平台限制

==========

GRUB2 is designed to be portable and is actually ported across platforms. We try to keep all platforms at the level. Unfortunately some platforms are better supported than others. This is detailed in current and 2 following sections.

GRUB2 被设计成可移植的，并且在事实上可以在平台之间移植。我们试图让所有的平台都在这个层次上。但是，不幸的是，一部分平台比另一部分平台提供更好的支持。本节将详细说明这些限制。

ARC platform is unable to change datetime (firmware doesn't seem to provide a function for it). EMU has similar limitation.

ARC 平台不能改变日期时间（固件似乎不提供这样一个函数）。EMU 有同样的限制。

ARC platform no serial port is available. EMU has similar limitation.

ARC 平台没有可用的串行口，EMU 有同样的限制。

Console charset refers only to firmware-assisted console. gfxterm is always Unicode (see Internationalisation section for its limitations). Serial is configurable to UTF-8 or ASCII (see Internationalisation). In case of qemu and coreboot ports the refered console is vga_text. Loongson always uses gfxterm.

对控制台字符集的引用仅仅指那些由固件辅助的控制台。gtxterm 总是使用 Unicode（查看国际化部分中对它的限制的描述）。串行输入可被配置成 UTF-8 编码或者 ASCII 编码（详细查看国际化部分）。假如模拟器和 coreboot 移植了被引用的控制台，则使用 vga_text。龙芯总是使用 gfxterm。

Most limited one is ASCII. CP437 provides additionally pseudographics. GRUB2 doesn't use any language characters from CP437 as often CP437 is replaced by national encoding compatible only in pseudographics. Unicode is the most versatile charset which supports many languages. However the actual console may be much more limited depending on firmware

最受限制的是 ASCII 码。CP437 额外提供伪图。GRUB2 不使用任何来自 CP437 的语言字符，CP437 经常被在伪图上兼容的国际字符集所替代。在支持多种语言方面，Unicode 是最通用的字符集。然而，事实上，控制台可能更多的受限于固件。

On BIOS network is supported only if the image is loaded through network. On sparc64 GRUB is unable to determine which server it was booted from.

仅仅当镜像文件是通过网络加载时，BIOS 的网络才能被使用。在 sparc64 上，GRUB 不能决定从哪个服务器启动加载。

On platforms not having direct serial support (as indicated in the line serial) you can still redirect firmware console to serial if it allows so.

在那些没有直接的串行端口支持(由串口线路表明)的平台上，如果被允许，也可以重定向固件控制台到串口。

Direct ATA/AHCI support allows to circumvent various firmware limitations but isn't needed for normal operation except on baremetal ports.

直接的 ATA/AHCI 支持去避免各种各样的固件限制，但是不支持一般的操作，除了在裸机的端口上。

AT keyboard support allows keyboard layout remapping and support for keys not available through firmware. It isn't needed for normal operation except baremetal ports.

AT 键盘支持键盘布局重映射，并且支持那些通过固件不能利用的键。对于一般操作，这不是必须的，除了在裸机的端口上。

USB support provides benefits similar to ATA (for USB disks) or AT (for USB keyboards). In addition it allows USBserial.
USB 提供类似 ATA（为支持 U 盘）或者 AT（为支持 USB 键盘）的支持。另外，也支持 USB 串口线。

Chainloading refers to the ability to load another bootloader through the same protocol
链式加载拥有通过同样协议加载另一个启动加载器的功能。

Hints allow faster disk discovery by already knowing in advance which is the disk in question. On some platforms hints are correct unless you move the disk between boots. On other platforms it's just an educated guess. Note that hint failure results in just reduced performance, not a failure
额外的线索使得能提前清楚哪个磁盘出现问题，所以能更快地发现磁盘。在一些平台上，线索一般是正确的，除非你在操作系统启动的时候移除磁盘。在另一些平台上，这仅仅是一个有根据的猜测。注意，线索的错误仅仅会造成性能的下降，而不是启动的失败。

BadRAM is the ability to mark some of the RAM as "bad". Note: due to protocol limitations mips-loongson (with Linux protocol) and mips-qemu_mips can use only memory up to first hole.
BadRAM 能把一些 RAM 作为“坏了的部分”标记出来。注意，由于协议限制，mips-loonson(使用 Linux 协议)和 mips-qemu_mips 仅仅只能连续读取到内存直至遇到第一个 hole 为止。

	BIOS	Coreboot	Multiboot	Qemu	a32-EFI	amd64-EFI	ia32-IEEE1275	Itanium	Loongson	sparc64	Powerpc	ARC
video	yes	yes	yes	yes	yes	yes	no	no	yes	no	yes	no
console charset	CP437	CP437	CP437	CP437	Unicode	Unicode	ASCII	Unicode	N/A	ASCII	ASCII	ASCII
network	yes (*)	no	no	no	yes	yes	yes	yes	no	yes (*)	yes	no
ATA/AHCI	yes	yes	yes	yes	yes	yes	yes	no	yes	no	no	no
AT keyboard	yes	yes	yes	yes	yes	yes	yes	no	yes	no	no	no
USB	yes	yes	yes	yes	yes	yes	yes	no	yes	no	no	no
chainloader	local	yes	yes	no	local	local	no	local	yes	no	no	no
cpuid	partial	partial	partial	partial	partial	partial	partial	no	no	no	no	no
hints	guess	guess	guess	guess	guess	guess	good	guess	good	good	good	no
PCI	yes	yes	yes	yes	yes	yes	yes	no	yes	no	no	no
badram	yes	yes	yes	yes	yes	yes	no	yes	yes (*)	no	no	no
compression	always	pointless	no	no	no	no	no	no	configurable	no	no	configurable
exit	yes	no	no	no	yes	yes	yes	yes	no	yes	yes	yes

	MIPS qemu	emu
video	no	yes
console charset	CP437	ASCII
network	no	yes
ATA/AHCI	yes	no
AT keyboard	yes	no
USB	N/A	yes
chainloader	yes	no
cpuid	no	no
hints	guess	no
PCI	no	no
badram	yes (*)	no
compression	configurable	no
exit	no	yes

24.概要

24.概要-Outline

********** **********

24 Outline

24 概要

==========

Some platforms have features which allows to implement some commands useless or not implementable on others.

部分平台允许执行一些没有太大价值的或者在别的平台上不可执行的命令。

Quick summary:

小结：

Information retrieval:

信息检索：

* mipsel-loongson: lsspd
* mips-arc: lsdev
* efi: lsefisystab, lssal, lsefimmap
* i386-pc: lsapm
* acpi-enabled (i386-pc, i386-coreboot, i386-multiboot, *-efi): lsacpi

Workarounds for platform-specific issues:

特定平台的应急方法：

* i386-efi/x86_64-efi: loadbios, fixvideo
* acpi-enabled (i386-pc, i386-coreboot, i386-multiboot, *-efi): acpi (override ACPI tables-会重写 ACPI 列表)
* i386-pc: drivemap
* i386-pc: sendkey

Advanced operations for power users:

高级用户的操作：

* x86: iorw (direct access to I/O ports-直接访问 I/O 端口)

Miscelaneous:

其他杂项：

* cmos (x86-*, ieee1275, mips-qemu_mips, mips-loongson): cmostest (used on some laptops to check for special power-on key-用于检查部分手提电脑上特殊的开机键)
* i386-pc: play

25.支持此引导的明细

25.支持此引导的明细-Supported boot targets

********** **********

25 Supported boot targets

25 支持此引导的明细

==========

X86 support is summarised in the following table. "Yes" means that the kernel works on the given platform, "crashes" means an early kernel crash which we hope will be fixed by concerned kernel developers. "no" means GRUB doesn't load the given kernel on a given platform. "headless" means that the kernel works but lacks console drivers (you can still use serial or network console). In case of "no" and "crashes" the reason is given in footnote.

下面的列表总结了关于 X86 的支持。“Yes”意味着内核能工作在指定的平台上，“crashes”意味着早期版本的内核崩溃有望将会被我们关切的内核开发者修补好。“no”意味着 GRUB 不能在指定的平台上启动指定的内核。“headless”意味着指定内核可以启动，但是缺少控制台驱动（不过你仍然可以使用串口控制台或者网络控制台）。如果是“no”和“crash”，它的原因在脚注中给出。

	BIOS	Coreboot	Multiboot	Qemu	ia32-EFI	amd64-EFI	ia32 IEEE1275
BIOS chainloading	yes	no (1)	no (1)	no (1)	no (1)	no (1)	no (1)
NTLDR	yes	no (1)	no (1)	no (1)	no (1)	no (1)	no (1)
Plan9	yes	no (1)	no (1)	no (1)	no (1)	no (1)	no (1)
Freedos	yes	no (1)	no (1)	no (1)	no (1)	no (1)	no (1)
FreeBSD bootloader	yes	crashes (1)	crashes (1)	crashes (1)	crashes (1)	crashes (1)	crashes (1)
32-bit kFreeBSD	yes	crashes (2,6)	crashes (6)	crashes (6)	headless	headless	crashes (6)
64-bit kFreeBSD	yes	crashes (2,6)	crashes (6)	crashes (6)	headless	headless	crashes (6)
32-bit kNetBSD	yes	crashes (1)	crashes (1)	crashes (1)	crashes (1)	crashes (1)	crashes (1)
64-bit kNetBSD	yes	crashes (2)	yes	yes	yes	yes	?
32-bit kOpenBSD	yes	yes	yes	yes	headless	headless	?
64-bit kOpenBSD	yes	yes	yes	yes	headless	headless	?
Multiboot	yes	yes	yes	yes	yes	yes	?
Multiboot2	yes	yes	yes	yes	yes	yes	?
32-bit Linux (legacy protocol)	yes	no (1)	no (1)	no (1)	no (1)	no (1)	no (1)
64-bit Linux (legacy protocol)	yes	no (1)	no (1)	no (1)	no (1)	no (1)	no (1)
32-bit Linux (modern protocol)	yes	yes	yes	yes	yes	yes	?
64-bit Linux (modern protocol)	yes	yes	yes	yes	yes	yes	?
32-bit XNU	yes	?	?	?	yes	yes	?
64-bit XNU	yes	?	?	?	yes (5)	yes	?
32-bit EFI chainloader	no (3)	no (3)	no (3)	no (3)	yes	no (4)	no (3)
64-bit EFI chainloader	no (3)	no (3)	no (3)	no (3)	no (4)	yes	no (3)
Appleloader	no (3)	no (3)	no (3)	no (3)	yes	yes	no (3)

1. Requires BIOS

1.需要 BIOS

2. Crashes because the memory at 0x0-0x1000 isn't available

2.发生崩溃是因为内存地址 0x0000-0x1000 之间是不可使用

3. EFI only

3.仅用于 EFI

4. 32-bit and 64-bit EFI have different structures and work in different CPU modes so it's not possible to chainload 32-bit bootloader on 64-bit platform and vice-versa

4.32-bit 和 64-bit 的 EFI 有着不同的结构并且工作在不同的 CPU 模式，因此不可能在 64 位平台上链式加载一个 32 位的引导加载程序，并且反之亦然。

5. Some modules may need to be disabled

5.一些模块可能需要被禁用

6. Requires ACPI

6.需要 ACPI

PowerPC, IA64 and Sparc64 ports support only Linux. MIPS port supports Linux and multiboot2.

PowerPC，IA64 和 SPARC64 的端口只支持 Linux。MIPS 的端口支持 Linux 与 multiboot2。

26.引导前的测试

26.引导前的测试-Boot tests

********** **********

26 Boot tests

26 引导前的测试

==========

As you have seen in previous chapter the support matrix is pretty big and some of the configurations are only rarely used. To ensure the quality bootchecks are available for all x86 targets except EFI chainloader, Appleloader and XNU. All x86 platforms have bootcheck facility except ieee1275. Multiboot, multiboot2, BIOS chainloader, ntldr and freebsd-bootloader boot targets are tested only with a fake kernel images. Only Linux is tested among the payloads using Linux protocols.

如你在之前的章节中所见到的，GRUB 支持的配置选项非常多，但是部分选项很少被用到。为了确保服务质量，启动检查会应用于所有的 x86 平台，但除了 EFI 的链式加载器、苹果加载器和 XNU。除了 ieee1275，所有的 x86 平台都具有引导检查的工具。Multiboot、multiboot2、BIOS 链式加载器、ntldr 和 freebsd-bootloader 在引导目标系统时只是测试目标是否是一个虚假的内核镜像。只有启动 LINUX 时才会在测试中使用 Linux 协议去测试内核镜像的有效加载。

Following variables must be defined:

下列变量必须被定义：

GRUB_PAYLOADS_DIR directory containing the required kernels-目录包含了必需的内核

GRUB_CBFSTOOL cbfstool from Coreboot package (for coreboot platform only)-Coreboot 里的 cbfstool 工具(只作用于 coreboot 平台)

GRUB_COREBOOT_ROM empty Coreboot ROM-空的 Coreboot ROM(只读储存器)

GRUB_QEMU_OPTS additional options to be supplied to QEMU-为支持 QEMU 提供额外的选项

Required files are:

必需的文件：

kfreebsd_env.i386 32-bit kFreeBSD device hints-32 位 kFreeBSD 的设备信息

kfreebsd.i386 32-bit FreeBSD kernel image-32 位 FreeBSD 的内核镜像

kfreebsd.x86_64, kfreebsd_env.x86_64 same from 64-bit kFreeBSD-64 位 kFreeBSD 的内核镜像

knetbsd.i386 32-bit NetBSD kernel image-32 位 NetBSD 的内核镜像

knetbsd.miniroot.i386 32-bit kNetBSD miniroot.kmod.-32 位 kNetBSD 的 miniroot 版本的内核模块

knetbsd.x86_64, knetbsd.miniroot.x86_64 same from 64-bit kNetBSD-64 位 kNetBSD 的内核镜像

kopenbsd.i386 32-bit OpenBSD kernel bsd.rd image-32 位 OpenBSD 的内核 bsd.rd 镜像

kopenbsd.x86_64 same from 64-bit kOpenBSD-64 位 kOpenBSD 的内核 bsd.rd 镜像

linux.i386 32-bit Linux-32 位的 Linux

linux.x86_64 64-bit Linux-64 位的 Linux

27.GRUB 产生的错误信息

27.GRUB 产生的错误信息-Error messages produced by GRUB

********** **********

27 Error messages produced by GRUB

27 GRUB 产生的错误信息

========== ==========

27.1 GRUB only offers a rescue shell

27.1 GRUB 只提供一个用作救援的 shell

GRUB's normal start-up procedure involves setting the 'prefix' environment variable to a value set in the core image by 'grub-install',
setting the 'root' variable to match, loading the 'normal' module from the prefix, and running the 'normal' command (*note normal). This command is responsible for reading '/boot/grub/grub.cfg', running the menu, and doing all the useful things GRUB is supposed to do.
GRUB 的正常启动过程包括设置"prefix"环境变量的值为通过"grub-install"设置在核心镜像里的值，设置"root"变量，为从“prefix”中匹配和加载“normal”模块，并且运行"normal"命令(注：normal)。这个命令是负责去读取"/boot/grub/grub.cfg"，运行目标菜单，并且实现 GRUB 该做的事情。

If, instead, you only get a rescue shell, this usually means that GRUB failed to load the 'normal' module for some reason. It may be possible to work around this temporarily: for instance, if the reason for the failure is that 'prefix' is wrong (perhaps it refers to the wrong device, or perhaps the path to '/boot/grub' was not correctly made relative to the device), then you can correct this and enter normal mode manually:
否则，反之，你只能得到一个救援 shell，这通常意味着 GRUB 因为某些原因造成加载"normal"模块失败。它也许可以用来暂时解决问题：例如，如果因为"prefix"配置错误的原因而出错(也许指向了错误的设备，或者可能"/boot/grub"路径错误造成不能指向相应的设备)，那么你能够正确地输入以下代码实现 normal 正常模式的手工启动：

```
# Inspect the current prefix (and other preset variables):
# 检查当前的 prefix(和其他预设的变量)：
set

# Find out which devices are available:
# 找出那些设备是可用的：
ls

# Set to the correct value, which might be something like this:
# 设置为正确的值，可能的输出如下：
set prefix=(hd0,1)/grub
set root=(hd0,1)
insmod normal
normal
```

However, any problem that leaves you in the rescue shell probably means that GRUB was not correctly installed. It may be more useful to try to reinstall it properly using 'grub-install DEVICE' (*note Invoking grub-install). When doing this, there are a few things to remember:
然而，因为任何的问题让你进入到救援 shell 很可能都是因为 GRUB 没有正确地安装好。尝试去重新安装和正确地使用"grub-install DEVICE"命令尝试重新安装它可能会更有用些(注：调用 grub-install)。当这样做时，有几件事要记住：

* Drive ordering in your operating system may not be the same as the boot drive ordering used by your firmware. Do not assume that your first hard drive (e.g. '/dev/sda') is the one that your firmware will boot from. 'device.map' (*note Device map) can be used to override this, but it is usually better to

use UUIDs or file system labels and avoid depending on drive ordering entirely.
* 操作系统中的驱动器次序可能不同于固件使用引导驱动器的次序。不要认为你的第一个硬盘(例如："/dev/sda")就是你的固件上首先引导的硬盘。"device.map"(注：驱动器映射)能够用来重写驱动器的次序，但它通常使用更好的 UUIDs 或文件系统标签从而避免完全依赖于驱动器次序。

* At least on BIOS systems, if you tell 'grub-install' to install GRUB to a partition but GRUB has already been installed in the master boot record, then the GRUB installation in the partition will be ignored.
* 至少在 BIOS 系统，如果你调佣"grub-install"把安装 GRUB 到一个分区里，但之前 GRUB 已经安装到过此驱动器的主引导记录里，那么此次安装将被忽略。

* If possible, it is generally best to avoid installing GRUB to a partition (unless it is a special partition for the use of GRUB alone, such as the BIOS Boot Partition used on GPT). Doing this means that GRUB may stop being able to read its core image due to a file system moving blocks around, such as while defragmenting,running checks, or even during normal operation. Installing to the whole disk device is normally more robust.
* 如果可能，一般最好不要安装 GRUB 到分区中(除非它是一个单独让 GRUB 使用的特殊分区，比如在 GPT 硬盘上的 BIOS 启动分区)。这样做意味着 GRUB 可能会因为一个文件系统块的左右移动从而造成不能够读取到相应的核心镜像，例如当磁盘整理碎片，运行磁盘检测的时候，或即使在正常操作的期间。将 GRUB 安装到整个磁盘驱动器通常来说更加健壮的。

* Check that GRUB actually knows how to read from the device and file system containing '/boot/grub'. It will not be able to read from encrypted devices, nor from file systems for which support has not yet been added to GRUB.
* 检查包含在"/boot/grub"里的设置事实上让你知道 GRUB 是如何读取设备和文件系统的。它无法读取加密的设备，也无法读取那些文件系统（如果对于此文件系统的支持未添加到 GRUB 中）。

28.调用 grub-install

28.调用 grub-install-Invoking grub-install

********** **********

28 Invoking grub-install

28 调用 grub-install

==========

The program 'grub-install' installs GRUB on your drive using 'grub-mkimage' and (on some platforms) 'grub-setup'. You must specify the device name on which you want to install GRUB, like this:

"grub-install"程序使用"grub-mkimage"和（在某些平台上）"grub-setup"来把 GRUB 安装到驱动器中。你必须指定一个设备名用来安装 GRUB，例如：

```
grub-install INSTALL_DEVICE
```

The device name INSTALL_DEVICE is an OS device name or a GRUB device name.'grub-install' accepts the following options:

设备名 INSTALL_DEVICE 可以是操作系统里的设备名称，也可以为 GRUB 中的设备名称。"grub-install"接受以下选项：

'—help'

Print a summary of the command-line options and exit.

输出一个命令行选项的综述后退出。

'--version'

Print the version number of GRUB and exit.

输出 GRUB 的版本号后退出。

'--boot-directory=DIR'

Install GRUB images under the directory 'DIR/grub/' This option is useful when you want to install GRUB into a separate partition or a removable disk. If this option is not specified then it defaults to '/boot', so

安装 GRUB 镜像到"DIR/grub"目录下。当你想把 GRUB 安装到一个独立分区或者一个可移动磁盘时，这个选项是非常有用的。如果没有指定此选项那么它的默认值是"/boot"，所以

```
grub-install /dev/sda
```

is equivalent to

等价于

```
grub-install --boot-directory=/boot/ /dev/sda
```

Here is an example in which you have a separate "boot" partition which is mounted on '/mnt/boot':

举个例子，如果你有一个单独的分区“/boot”，并且挂载在“/mnt/boot”下：

```
grub-install --boot-directory=/mnt/boot /dev/sdb
```

'--recheck'

Recheck the device map, even if '/boot/grub/device.map' already exists. You should use this option whenever you add/remove a disk into/from your computer.

添加这个选项后，会核对设备映射，无论"/boot/grub/device.map"是否已经存在。无论何时，当你添加驱动器到电脑，或是从电脑删除驱动器，都应该使用这个选项。

29.调用 grub-mkconfig

29.调用 grub-mkconfig - Invoking grub-mkconfig

********** **********

29 Invoking grub-mkconfig

29 调用 grub-mkconfig

==========

The program 'grub-mkconfig' generates a configuration file for GRUB(*note Simple configuration).

"grub-mkconfig"程序可以生成一个 GRUB 的配置文件(注：简单配置)。

```
grub-mkconfig -o /boot/grub/grub.cfg
```

'grub-mkconfig' accepts the following options:

"grub-mkconfig"接受以下的选项：

'--help'

Print a summary of the command-line options and exit.

打印出一个命令行选项的综述后退出。

'--version'

Print the version number of GRUB and exit.

打印出 GRUB 的版本号后退出。

'-o FILE'
'--output=FILE'

Send the generated configuration file to FILE. The default is to send it to standard output.

把生成的配置文件发送到 FILE 所指的位置里。默认情况下是发送文件到标准输出。

30.调用 grub-mkpasswd-pdkdf2

30.调用 grub-mkpasswd-pdkdf2 - Invoking grub-mkpasswd-pbkdf2

********** **********

30 Invoking grub-mkpasswd-pbkdf2

30 调用 grub-mkpasswd-pdkdf2

==========

The program 'grub-mkpasswd-pbkdf2' generates password hashes for GRUB (*note Security).

"grub-mkpasswd-pbkdf2"程序可以生成一个 GRUB 的哈希密码。(注：安全)

grub-mkpasswd-pbkdf2

'grub-mkpasswd-pbkdf2' accepts the following options:

"grub-mkpasswd-pbkdf2"接受以下的选项：

'-c NUMBER'

'--iteration-count=NUMBER'

Number of iterations of the underlying pseudo-random function.Defaults to 10000.

底层伪随机函数迭代的次数。默认值为 10000。

'-l NUMBER'

'--buflen=NUMBER'

Length of the generated hash. Defaults to 64.

所生成的哈希密码长度。默认值为 64。

'-s NUMBER'

'--salt=NUMBER'

Length of the salt. Defaults to 64.

用"盐"的方法把 hash 密码作加密，会产生另一个随即生成的密码。可以理解为密上密。默认生成长度为 64 位。

31.调用 grub-mkrescue

31.调用 grub-mkrescue - Invoking grub-mkrescue

********** **********

31 Invoking grub-mkrescue

31 调用 grub-mkrescue

==========

The program 'grub-mkrescue' generates a bootable GRUB rescue image (*note Making a GRUB bootable CD-ROM).

"grub-mkrescue"程序可以生成一个可引导的 GRUB 救援镜像(注：制作一个 GRUB 可引导 CD-ROM)。

```
grub-mkrescue -o grub.iso
```

All arguments not explicitly listed as 'grub-mkrescue' options are passed on directly to 'xorriso' in 'mkisofs' emulation mode. Options passed to 'xorriso' will normally be interpreted as 'mkisofs' options; if the option '--' is used, then anything after that will be interpreted as native 'xorriso' options.

所有那些没有明确作为“grub-mkrescue”命令选项的参数，都是在“mkisofs”仿真模式下直接传递给“xorriso”的。传递给“xorriso”的选项都会正常地被作为“mkisofs”选项解读。如果选项里使用了“--”，那么在这个后面的任何选项都将作为本地“xorriso”选项被解读。

Non-option arguments specify additional source directories. This is commonly used to add extra files to the image:

非选项参数指定额外的源目录。这通常用于添加额外的文件到镜像里：

```
mkdir -p disk/boot/grub
```

(add extra files to 'disk/boot/grub') (添加额外的文件到"disk/boot/grub")

```
grub-mkrescue -o grub.iso disk
```

'grub-mkrescue' accepts the following options:

"grub-mkrescue"接受以下选项：

'--help'

Print a summary of the command-line options and exit.

打印出一个命令行选项的综述后结束。

'--version'

Print the version number of GRUB and exit.

打印出 GRUB 的版本号后结束。

'-o FILE'

'--output=FILE'

Save output in FILE. This "option" is required.

把输出保存到 FILE 文件中。这个选项是必需的。

'--modules=MODULES'

Pre-load the named GRUB modules in the image. Multiple entries in MODULES should be separated by whitespace (so you will probably need to quote this for your shell).

预先加载指定的 GRUB 模块到镜像里。多个 MODULES 模块参数需要用空格符作分隔(所以你大概将要在你的 shell 里使用上它)。

'--rom-directory=DIR'

If generating images for the QEMU or Coreboot platforms, copy the resulting 'qemu.img' or 'coreboot.elf' files respectively to the DIR directory as well as including them in the image.

如生成的镜像是用于 QEMU 或 Coreboot 平台，需要依次复制所生成的"qemu.img"或"coreboot.elf"文件到 DIR 文件中，就像在镜像文件中包含它们那样。

'--xorriso=FILE'

Use FILE as the 'xorriso' program, rather than the built-in default.

使用 FILE 文件作为"xorriso"程序，而不是那个默认的内置程序。

'--grub-mkimage=FILE'

Use FILE as the 'grub-mkimage' program, rather than the built-in default.

使用 FILE 文件作为"grub-mkimage"程序，而不是那个默认的内置程序。

32.调用 grub-probe

32.调用 grub-probe - Invoking grub-probe

********** **********

32 Invoking grub-probe

32 调用 grub-probe

==========

The program 'grub-probe' probes device information for a given path or device.

"grub-probe"程序会探测一个给定了路径或设备的设备信息。

```
grub-probe --target=fs /boot/grub
grub-probe --target=drive --device /dev/sda1
```

'grub-probe' must be given a path or device as a non-option argument, and also accepts the following options:

"grub-probe"必需给定一个路径或设备作为一个无选项的参数，并且接受以下选项。

'--help'

Print a summary of the command-line options and exit.

打印出命令行选项的综述后结束。

'--version'

Print the version number of GRUB and exit.

打印出 GRUB 的版本号后结束。

'-d'
'--device'

If this option is given, then the non-option argument is a system device name (such as '/dev/sda1'), and 'grub-probe' will print information about that device. If it is not given, then the non-option argument is a filesystem path (such as '/boot/grub'),and 'grub-probe' will print information about the device containing that part of the filesystem.

如果这个选项被用上，那么无选项的参数会变成为一个系统里的设备名称(比如"/dev/sda1")，并且"grub-probe"将会打印出有关该设备的信息。如果这个选项没有被用上，那么这个无选项的参数将会是一个文件系统的路径(比如"/boot/grub")，并且"grub-probe"将会打印出包含此部分文件系统的设备的信息。

'-m FILE'
'--device-map=FILE'

Use FILE as the device map (*note Device map) rather than the default, usually '/boot/grub/device.map'.

使用 FILE 文件作为设备映射(注：设备映射)而不是使用那个默认的，默认的通常位于"/boot/grub/device.map"。

'-t TARGET'
'--target=TARGET'

Print information about the given path or device as defined by TARGET. The available targets and their meanings are:

打印出关于 TARGET 里所给定的路径或设备定义的信息。以下为可用的 targets 与它们的含义：

'fs'

GRUB filesystem module.

GRUB 文件系统模块。

'fs_uuid'

Filesystem Universally Unique Identifier (UUID).

文件系统的通用唯一标识码(UUID)。

'fs_label'
Filesystem label.
文件系统的标签。

'drive'
GRUB device name.
GRUB 的设备名称。

'device'
System device name.
系统的设备名称。

'partmap'
GRUB partition map module.
GRUB 的分区映射模块。

'abstraction'
GRUB abstraction module (e.g. 'lvm').
GRUB 的抽象层模块(例如"lvm")。

'cryptodisk_uuid'
Crypto device UUID.
加密设备的 UUID。

'msdos_parttype'
MBR partition type code (two hexadecimal digits).
MBR 的分区类型代码(两个十六进制的数字)。

'hints_string'
A string of platform search hints suitable for passing to the 'search' command (*note search).
用于平台搜索的提示字符串，用于传递给“search”命令。

'bios_hints'
Search hints for the PC BIOS platform.
用于 PC BIOS 平台的搜索提示信息。

'ieee1275_hints'
Search hints for the IEEE1275 platform.
用于 IEEE1275 平台的搜索提示信息。

'baremetal_hints'
Search hints for platforms where disks are addressed directly rather than via firmware.
对于那些磁盘是直接寻址，而不是通过固件的平台，用于这种平台的搜索提示信息。

'efi_hints'
Search hints for the EFI platform.
用于 EFI 平台的搜索提示信息。

'arc_hints'
Search hints for the ARC platform.
用于 ARC 平台的搜索提示信息。

'compatibility_hint'
A guess at a reasonable GRUB drive name for this device, which may be used as a fallback if the

'search' command fails.
在一个合理的 GRUB 设备名称下，对驱动器所作的推测，这可能在"search"命令使用失败之后用作后备。

'disk'
System device name for the whole disk.
整个硬盘里的系统驱动器名称。

'-v'
'--verbose'
Print verbose messages.
打印出冗余信息。

33.附录 A 如何获得并构建 GRUB - Appendix A How to obtain and build GRUB

********** **********

Appendix A How to obtain and build GRUB

附录 A 如何获得并构建 GRUB

==========

Caution: GRUB requires binutils-2.9.1.0.23 or later because the GNU assembler has been changed so that it can produce real 16bits machine code between 2.9.1 and 2.9.1.0.x. See <http://sources.redhat.com/binutils/>, to obtain information on how to get the latest version.

*警告：*GRUB 需要 2.9.1.0.23 或更新版本的工具集，因为 GNU 的汇编程序已经更改了以便在 2.91 和 2.9.1.0.x 版本之间能够生成出真正的 16bits 机器代码。查看<http://sources.redhat.com/binutils/>，获取如何取得最新版本的信息。

GRUB is available from the GNU alpha archive site <ftp://ftp.gnu.org/gnu/grub> or any of its mirrors. The file will be named grub-version.tar.gz. The current version is 2.00, so the file you should grab is:

可以通过 GNU alpha 档案网站<ftp://ftp.gnu.org/gnu/grub>或者它的其他镜像站点获取 GRUB。文件将被命名为 grub-version.tar.gz。而目前为止版本号为 2.00，所以你获取的文件为：

<ftp://ftp.gnu.org/gnu/grub/grub-2.00.tar.gz>

To unbundle GRUB use the instruction:

使用下面的命令解压 GRUB：

```
zcat grub-2.00.tar.gz | tar xvf -
```

which will create a directory called 'grub-2.00' with all the sources. You can look at the file 'INSTALL' for detailed instructions on how to build and install GRUB, but you should be able to just do:

这将会创建一个名为"grub-2.00"的目录并且存放了所有的源代码。你可以查看"INSTALL"文件去了解如何搭建和安装 GRUB 的详细使用说明，但你第一步要做的是以下工作：

```
cd grub-2.00
./configure
make install
```

Also, the latest version is available using Bazaar. See <http://www.gnu.org/software/grub/grub-download.en.html> for more information.

此外，使用 Bazaar 可以获取最新的版本。查看<http://www.gnu.org/software/grub/grub-download.en.html>获取更多信息。

外文引用：

GNU Bazaar (formerly Bazaar-NG, command line tool bzr) is a distributed revision control system sponsored by Canonical.

Bazaar can be used by a single developer working on multiple branches of local content, or by teams collaborating across a network.

Bazaar is written in the Python programming language, with packages for major GNU/Linux distributions, Mac OS X and Microsoft Windows. Bazaar is free software and part of the GNU Project.

34.附录 B 提交漏洞

34.附录 B 提交漏洞-Appendix B Reporting bugs

********** **********

Appendix B Reporting bugs

附录 B 提交漏洞

==========

These are the guideline for how to report bugs. Take a look at this list below before you submit bugs:
以下为如何提交漏洞的指南。在你提交漏洞之前请先查阅以下内容：

1. Before getting unsettled, read this manual through and through. Also, see the GNU GRUB FAQ(http://www.gnu.org/software/grub/grub-faq.html).
1. 在问题变得不可解决之前之前，请你先完全阅读完本手册。并且，请到 GNU GRUB 的常见问题区 FAQ 查找(http://www.gnu.org/software/grub/grub-faq.html)

2. Always mention the information on your GRUB. The version number and the configuration are quite important. If you build it yourself, write the options specified to the configure script and your operating system, including the versions of gcc and binutils.
2.总是关注 GRUB 给出的信息。版本和配置是十分重要的。如果你自己构建 GRUB，记录针对配置脚本和操作系统的选项，包括 gcc 和各种工具集。

3. If you have trouble with the installation, inform us of how you installed GRUB. Don't omit error messages, if any. Just 'GRUB hangs up when it boots' is not enough.
The information on your hardware is also essential. These are especially important: the geometries and the partition tables of your hard disk drives and your BIOS.
3.如果你在安装过程中遇到麻烦，告知我们你是如何安装 GRUB。无论如何，不要忽略错误信息。仅仅说“启动时，GRUB 挂起了”是不够的。电脑的硬件信息是必不可少的。以下信息是特别重要的：硬盘的几何信息（包含柱面，磁道，扇区，扇区字节数和容量）和分区表，BIOS 信息。

4. If GRUB cannot boot your operating system, write down _everything_ you see on the screen. Don't paraphrase them, like 'The foo OS crashes with GRUB, even though it can boot with the bar boot loader just fine'. Mention the commands you executed, the messages printed by them, and information on your operating system including the version number.
4.如果 GRUB 不可以引导你的操作系统，记录下屏幕上的所有信息。不要意译它们，如“GRUB 引导 foo OS 时崩溃，虽然在 bar 引导加载器下可用”。注意，所执行命令的输出信息，操作系统的信息（包含版本号）。

5. Explain what you wanted to do. It is very useful to know your purpose and your wish, and how GRUB didn't satisfy you.
5.解释你想做的。为了了解你的目的和愿望，了解 GRUB 如何无法满足你的需求，这样做是非常有用的。

6. If you can investigate the problem yourself, please do. That will give you and us much more information on the problem. Attaching a patch is even better.
When you attach a patch, make the patch in unified diff format, and write ChangeLog entries. But, even when you make a patch, don't forget to explain the problem, so that we can understand what your patch is for.
6.如果你想自己研究遇到的问题，那就去做吧。这会给你和我们带来更多关于此问题的信息。如果能提交一个补丁当然更好。当你提交补丁时，请使用统一的 diff 格式，并且记录 ChangLog 条目。而且，当你制作补丁时，别忘记解释所遇到的问题，这样我们就能够知道你的补丁的作用是什么。

7. Write down anything that you think might be related. Please understand that we often need to reproduce the same problem you encountered in our environment. So your information should be sufficient for us to do the same thing--Don't forget that we cannot see your computer directly. If you are not sure whether to state a fact or leave it out, state it! Reporting too many things is much better

than omitting something important.
If you follow the guideline above, submit a report to the Bug Tracking System (http://savannah.gnu.org/bugs/?group=grub). Alternatively, you can submit a report via electronic mail to <bug-grub@gnu.org>, but we strongly recommend that you use the Bug Tracking System, because e-mail can be passed over easily.
7.写下任何你觉得可能与此问题相关的东西。你要理解我们经常需要重现你遭遇到的问题。你给出的信息应该足够充分，以便我们能重现那个问题。不要忘记我们不能直接接触到你的电脑。如果你不确定是否应该告知我们某个情况，那最好还是告知我们吧。告知我们过多的东西好过忽略某些重要的东西。如果你完成了上述步骤，之后请提交报告到漏洞跟踪系统（http://savannah.gnu.org/bugs/?grub@gnu.org)。此外，也可以通过发电子邮件到<bug-grub@gnu.org>，提交你的报告。但是，我们强烈建议你使用漏洞跟踪系统，因为电子邮件容易被忽略。

Once we get your report, we will try to fix the bugs.
一旦你提交了报告，我们会尽快修复漏洞。

35.附录 C GRUB 的未来

35.附录 C GRUB 的未来 - Appendix C Where GRUB will go

********** **********

Appendix C Where GRUB will go

附录 C GRUB 的未来

==========

GRUB 2 is now quite stable and used in many production systems. We are currently working towards a 2.0 release.

If you are interested in the development of GRUB 2, take a look at the homepage (http://www.gnu.org/software/grub/grub.html).

GRUB 2 是十分稳定的，并且在许多作为产品的系统中使用。我们现在正致力于发行版 2.0。

如果你对 GRUB 2 的开发工作感兴趣，请访问我们的主页（http://www.gnu.org/sofeware/grub/grub.html）。

36.附录 D 手册的复制

36.附录 D 手册的复制-Appendix D Copying This Manual

********** **********

Appendix D Copying This Manual

附录 D 这份手册的复制传播

==========

D.1 GNU Free Documentation License

D.1 GNU 自由文档许可证

Version 1.2, November 2002

版本 1.2，2002 年 10 月

0. PREAMBLE

0.前言

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it,with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

本协议着意于保证手册、教程或其他功能、用途的文档的自由：以确保任何人不管在商业领域还是非商业领域都可以复制和（修改或者没有修改）重新发布这些文档的自由，其次，本协议保护文档作者和发布者由他们的工作获得的信誉不会因他人对其文档的修改而受损。

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense.It complements the GNU General Public License, which is a copyleft license designed for free software.

本协议是一种“copyleft”，其含义为：文档的派生产物必须与原文档同样是自由的。本文档是 GNU 通用公共协议（为自由软件设计的一种 copyleft 协议）的补充。

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

我们设计本协议供自由软件的手册使用，因为自由软件需要自由文档：一个自由软件附带的手册当然要享有与该软件同等的自由。不过本协议并不仅限于对程序的手册使用，所有文档作品都可以使用本协议，不管它是关于什么主题的或者是否它会作为出版物出版。我们建议本协议主要地面向那些指导性质或参考性质的文档作品。

1.APPLICABILITY AND DEFINITIONS

1.适用的范围和约定

This License applies to any manual or other work, in any medium,that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration,to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a

way requiring permission under copyright law.
本协议适用于任何媒体上的任何手册或其他文档，文档版权所有者只需在文档中声明使用本协议的条款发布。该声明允许在后文中的规定下，在世界范围内、免版税、无时限地使用该作品。以下的“文档”都是指此类手册或作品。公众的任何一个人都是本协议的许可对象，这里用“你”来称呼。如果你复制、修改或发布了这些（在某种意义上也可以说是在版权法保护下的）文档，就表明你已经接受了本协议。
A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.
文档的“Modified Version”（修正版）是指任何包含全部或部分文档的作品，不论是照搬照抄或是翻译成其他语言。
A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.
“Secondary Section(附属章节)”是指定的附录或文档的前序部分，专门描述有关文档主题的文档出版者或作者的相关信息 (或对一些相关情况进行说明)并且包含文档主题不会直接提及的内容。 (如果文档是数学教材的一部分，附属章节或许连一点数学都不会提到。) 这些相关信息可以是与主题有关的历史关联、相关事物或者相关的法律、商业、哲理、道德或政治立场。
The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.
“Invariant Sections(固定章节)”是某些指定了标题的附属章节，固定章节做为文档的一部分也象声明提到的那样，在本协议的保护下发布。如果某章节不符合上面 关于附属章节的定义，那它就不能被称为固定章节。文档可以没有固定章节。如果看不出文档中有什么固定章节，那就是没有了。
The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.
“Cover Texts(封皮文字)”是那些简短的小段文字列，就象封面或封底的文字。封皮文字做为文档的一部分也象声明提到的那样，在本协议的保护下发布。封面文字一般最多 5 个词语，封底文字一般最多 25 个词语。
A "Transparent" copy of the Document means a machine-readable copy,represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".
文档的“Transparent(透明，此处理解为 兼容)”副本是指可以用计算机处理的副本，表现为其格式符合一般通用规范，这样就适于直接用通用文本编辑器或通用绘画程序(对由像素构成的图片)或那些被广泛应用的图像编辑器(用来绘图)修改文档，并且其格式适于输入文本格式器或可转换为适于输入文本格式器的兼容格式。如果某副本(不论有没有标记)为反对或防止读者的后续修正而采用其它格式，那么就是不兼容的。如果图片格式的使用有任何实质性专利条文的限制，那么该图片格式就是不兼容的。文档副本的不“Transparent”我们称为“Opaque(不透明，此处理解为 不兼容)”。
Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format,SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification.Examples of transparent image formats include PNG, XCF and JPG.Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.
文档副本适用的兼容格式包括没有标记的纯 ASCII 码、Texinfo 格式、LaTeX 格式、 SGML 格式或 XML 格式

这些公认有效的文件(格式)类型定义，还有与标准兼容的精简 HTML、PostScript 或 PDF 这些便于编辑的格式。图片适 用的兼容格式包括 PNG、XCF 和 JPG。不兼容格式包括用专有文档处理器才能阅读编辑的专有格式、使用非公认的文件格式定义和/或使用非公认的处理工具 的 SGML 格式或 XML 格式，还有那些机器生成的 HTML 及某些文档处理器产生的 PostScript 或 PDF 等只用于输出的格式。
The "Title Page" means, for a printed book, the title page itself,plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.
“Title Page(扉页)”是指刊印成册的书的扉页本身及附加下列有必要明显地保留的页面：本协议要求在扉页出现的材料。像那些版式中没有扉页的作品，“Title Page”是指在作品正文之前最显著的标题附近的文字。
A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.
章节“Entitled XYZ(题为 XYZ)”意为文档某指定片断标题恰为或包含 XYZ，后面括号中用不同的术语解释 XYZ。(这里的 XYZ 代指下面提及的特殊章节的名称，例 如：鸣谢、贡献、注记、历史。) 当你修改文档时，对此类章节要“Preserve the Title(维持标题)”，就是说，要依据这里的定义保留该章节题为 XYZ。
The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.
文档可能会在声明本协议适用于该文档的通告后包含免责声明。这个免责声明被考虑包含到本协议的参考中，但仅仅关于免责：该免责声明的任何其它可能的含义是无效的，并对本协议的含义不造成影响。

2. VERBATIM COPYING

2. 原样复制

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However,you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.
在提供本协议和版权声明，同时在许可声明中指明文档及其所有副本均在本协议保护下，并且不加入任何其它条件到文档的协议的前提下，你可以在任何媒体复制和 发布该文档，不论是用于商业或非商业。你不可以对你 制作或发布 的副本进行技术处理以 妨碍或控制 阅读或再复制。不过，你可以因交换副本而接受他人给你的一些补偿。如果你发布的副本数量大到一定程度，你还必须看看下面的第 3 节。
You may also lend copies, under the same conditions stated above,and you may publicly display copies.
在与上述相同的情形下，你还可以出借文档副本和公开展示文档副本。

3. COPYING IN QUANTITY

3. 大量发布

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.
如果你发布了付印的文档副本(或使用其它一般有印有封皮的媒质的副本)数量超过 100 份，你必须把副本用清晰明了地印有文档的许可声明所要求的封皮文字的封皮封装好，这些封皮文字包括封面文字和封底文字。封面和封底还都必须写明你是该副本的发布者。封面必须印上完整的书名，书名的每个字都必须同样地显著。你可以在封面添一些其它素材做为补充。在满足了遵循对封皮改动的限制、维持文档标题等这些条件下，副

本在某方面上可以被视同原样复制的副本。
If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.
如果封面或封底必要的文字太多而导致看起来不清晰明了，那么你可以把优先的条目(尽量适合地)列在封面或封底，剩下的那些接在邻近的页面里。
If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.
如果你出版或分发不兼容的文档副本数量超过 100 份，那么，你要么随同每份不兼容副本带一份 可机读 的兼容副本，要么在每份不兼容副本里或随同该副本指明一个计算机网络的地址，让大部分使用网络的群众可以用通用网络通信协议到该处下载完全兼容的没有附加 材料的文档副本。如果你采用后一种办法，你必须适当采取一些慎重的措施：当你开始大量发行不兼容副本时，要保证这些兼容副本在你(直接或通过你的代理商或 零售商)发布最后一份对应版本的不兼容副本后至少一年在指定的地址可以访问到。
It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies,to give them a chance to provide you with an updated version of the Document.
在你对文档副本进行大量再发布前取得与文档作者良好的联系，这虽然不是必须的，但是是有必要的，这样你就有机会得到他们提供的文档的最新版本。

4. MODIFICATIONS

4. 修正

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:
你可以在上文 2、3 两节的条件下复制和发布文档的修正版，只要你严格地在本协议下发布 承担了原来文档的任务的 修正版，这样就允许任何持有该修正版一份副本的人可对修正版进行发行和修改。而且，你必须对修正版做如下一些事情：

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

A. 在扉页(如果有封皮，则封皮和扉页都要)使用的书名要与原来文档的书名明显地区别开来，以及那些先前版本(哪些是需要区别的呢？如果原来文档有历史小节，那么历史小节中列出的都是)。如果最初的发行人允许的话，你也可以使用相同的书名。

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

B. 扉页上必须列出作者、对修正版文档的修改负责的个人或团体或机构、文档的至少五个主要作者(如果主要作者少于五个，就全部写上)，除非他们允许你不这么做。

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

C. 在扉页放上修正版文档的发布者的名字，如同(前面文档副本的)发布者那样。

D. Preserve all the copyright notices of the Document.

D. 保留原有文档的所有版权声明不变。

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

E. 在其它版权声明附近为你所做的修正添加适当的版权声明。

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

F. 在版权声明后面立即接上允许公众在本协议下使用修正版文档的许可声明，其形式如下面附录所示。

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

G. 在文档的许可声明里保留所有固定章节和封皮文字所要求的许可声明。

H. Include an unaltered copy of this License.

H. 放上这些许可证的一份原件。

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

I. 保留题为“历史”的章节，不要改动其标题，并且给它添加一条包括修改版的书名、日期、作者、发布者的条目，就像扉页里给出的那样。如果原文档没有题为“历 史”的章节，你就建立该章节，创建一条包括原文档的书名、日期、作者、发布者的条目，如原文档的扉页里给出的那样，然后再象上面规定那样，添一条描述修改 版的条目。

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

J. 如果文档中给出了访问兼容文档的链接，你应该保护网址的有效，并且该网址还应该提供该文档的前承版本。这些链结可以在“历史”章节给出。如果前承文档发布已超过四年或者原作者在文档中提到不必指明前承版本，那么你可以在作品中略去其链结。

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

K. 原文档中的任何题为“鸣谢”或“贡献”的章节都必须维持其标题和章节中对每位贡献者的谢词的原旨和感情。

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

L. 维持原文档的固定章节的标题和内容不变。当然，章节标号不属于章节标题的一部分(可以改动)。

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

M. 删除原文档中任何题为“注记”的章节。这些章节可能是之针对原文档的。

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

N. 不要把任何已有章节标题改为“注记”，或改为与固定章节有冲突的标题。

O. Preserve any Warranty Disclaimers.

O. 保留原文所有免责声明。

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.
如果文档修正版包含了新的没有从原文档摘取素材的有附属章节资格的前序章节或附录，你可以在自己的选择下指定这些章节的全部或一些为固定章节。你需要为修正版的许可声明中的固定章节序列添加标题。这些标题必须与其它章节的标题明显地区别开来。
You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.
你可以添加题为“注记”的章节，只要包含各方面对你的修正版的评议，例如：综合评论或被某组织认可的某标准的权威定义的原文。
You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.
你可以在修正版封皮文字序列末尾添加 5 个词左右的书名和 25 个词左右的封底文字。一个实体(进行出版的个人或组织)可能只会添加(或通过整理形成)一段封 面文字和封底文字。如果文档已经包含同样用于封皮的封皮文字(先前由你添加或由你代理的实体整理)，你就不要再另外添加了；不过在添加旧封皮的原发布者的 明确许可下你可以把旧的替换掉。
The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.
本协议不允许文档作者和发布者用他们的名字公开声称或暗示对任何修正版的认可。

5. COMBINING DOCUMENTS
5. 合并文档
You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.
你可以在本协议下依据第 4 节里对修正版定义的条款把文档与其它文档合并起来发布，只要你在合并文档里原封不动地包含所有原始文档的所有固定章节，并在你的合并文档的许可声明中将其全部列为固定章节，而且保留其所有免责声明。
The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.
合并后文档只需包含一份本协议，重复的多处固定章节应该用单独的章节替代。如果重复的固定章节只是标题相同，但内容是不同的，就在每一小段标题末端的括号 里添加已知的原作者或原发布者(的名字)，或给每小段一个唯一的标号。在合并文档的许可声明里的固定章节中的章节标题也要做相应的调整。
In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".
在合并文档里，你必须把不同原文档的“历史”章节合并成一个“历史”章节；同样需要合并的章节有“鸣谢”、“贡献”。你还必需把所有“注记”章节删除。

6. COLLECTIONS OF DOCUMENTS
6. 文档合集
You may make a collection consisting of the Document and other documents released under this

License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.
如果你对每份文档不论在哪方面都遵循了本协议的原样复制条款，那么你可以在本协议下发布你以该文档和其他文档制作的合集，并以一个单独的副本来替代合集中各文档里分别使用的本协议副本。
You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.
你可以从该合集中提取一个单独的文档，并在本协议下单独发布，只要你在该提取出的文档中加入一份本协议的副本，并在文档的其他所有方面都遵循本协议的原样复制的条款。

7. AGGREGATION WITH INDEPENDENT WORKS
7. 独立作品汇集
A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.
文档和其它独立的不受本协议约束的派生文档或作品编纂在一个大文库中或大媒体上，如果汇编作品的版权对该汇编作品用户的权利限制没有超出那些独立作品的许 可范围，该汇编作品称为“aggregate(集合体，这里译为：汇集或汇编)”。当文档被包含在汇集中的时候，本协议不对汇集中该文档的非直承(不是以 本协议发布的)作品起作用。
If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.
如果第 3 节中对封皮文字的要求适用于文档的该副本，那么如果文档在汇集中所占的比例小于全文的一半，文档的封皮文字可以被置于汇集内该文档的内封页上，或是电子文档中的等效部分(如果文档以电子版发布)。否则，必须将其现于封装整个汇集的印好的封皮上。

8. TRANSLATION
8. 翻译
Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.
我们认为翻译是一种修改，所以你可以按照第 4 节的条款发布文档的翻译版本。如果要将文档的固定章节用译文取代，需要得到版权持有者的许可，但你可以将部分 或全部固定章节的译文附加在原始版本的后面。你可以包含一份本协议、所有许可声明及免责声明的译文，只要你同时包含其原始英文版本即可。当译文和原文有歧 义的时候，以原文为准。
If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.
文档中的鸣谢、贡献、历史等要求维持其标题的章节这里必须将其标题替换成当前译文的标题。

9. TERMINATION
9. 协议终止
You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

除非你明确遵循本协议，否则你不可以对文档进行复制、修改、分授许可及发布。对文档的任何其它(违反本协议的)意图的复制、修改、分授许可及发布是无效的，并且这将自动终止你在本协议下享有的权利。
however,partied who have received copies,or rights,from you under this license will not have their licenses terminated as long as such partied remain in full compliance.
不过，从你这里获得文档副本或权利的群体的协议不会被终止，只要他们完全遵循本协议。

10. FUTURE REVISIONS OF THIS LICENSE
10. 本协议的未来版本
The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.
自由软件基金会有时会发布新的 GNU 自由文档协议 修订版。新版本将会和当前版本体现类似的精神，但在处理某些新的问题和利害关系的细节上会有所不同。详细查看 http://www.gnu.org/copyleft/.
Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.
本协议的每个版本都有一个唯一的版本号。如果文档指定遵循一个特定的本协议版本"或任何后续版本"，你可以选择遵循指定版本或自由软件基金会发布的任何后 续(非草案)版本的条款和条件。如果文档没有指定本许可证的版本，那么你可以选择遵循任何自由软件基金会已发布的(非草案)版本。

D.1.1 ADDENDUM: How to use this License for your documents
D.1.1 附录：如何让自己的文档使用这样一份证书

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:
为了在自己的文档中使用这份证书，需要在文档中包含证书的副本，并且把下面的版权证书公告放置在封面后。
Copyright (C) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation;with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ``GNU Free Documentation License''.
If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:
如果你有固定章节、封皮文字和封底文字，请将“with...Texts.“替换为：

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.
如果你有固定章节而没有封皮文字或上述三项的其它组合，将以上两个方法结合使用以满足实际情况。
If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.
如果你的文档包含有意义的程序代码范例，我们建议你将这些范例相应地使用你选择的自由软件协议发布，比如 GNU 通用公共协议，以授权它们作为自由软件使用。

<本文的译文来自百度百科词条“GNU 自由文档许可证”>

欢迎点击这里的链接进入精彩的Linux公社网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

Linux公社是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：**Ubuntu 专题** **Fedora 专题** **Android 专题** **Oracle 专题** **Hadoop 专题** **RedHat 专题** **SUSE 专题** **红旗 Linux 专题** **CentOS 专题**

Linux 公社微信公众号：linuxidc_com