

GRUB2 配置文件 "grub.cfg" 详解 (GRUB2 实战手册)

作者：[金步国](#)

版权声明

本文作者是一位开源理念的坚定支持者，所以本文虽然不是软件，但是遵照开源的精神发布。

- 无担保：本文作者不保证作品内容准确无误，亦不承担任何由于使用此文档所导致的损失。
- 自由使用：任何人都可以自由的[阅读/链接/打印](#)此文档，无需任何附加条件。
- 名誉权：任何人都可以自由的[转载/引用/再创作](#)此文档，但必须保留作者署名并注明出处。

其他作品

本文作者十分愿意与他人分享劳动成果，如果你对我的其他翻译作品或者技术文章有兴趣，可以在如下位置查看现有的作品集：

- [金步国作品集](http://www.jinbuguo.com/) [<http://www.jinbuguo.com/>]

联系方式

由于作者水平有限，因此不能保证作品内容准确无误。如果你发现了作品中的错误(哪怕是错别字也好)，请来信指出，任何提高作品质量的建议我都将虚心接纳。

- Email(QQ)：70171448 在 QQ 邮箱

招聘

公司：某保险公司信息技术部

职位：Linux SA, PostgreSQL DBA

坐标：上海/北京

要求：Gentoo 粉优先

有意者请直接加我 QQ70171448 了解详情(加我时请注明"应聘")

GRUB2 模块

对于 GRUB-2.0.2 版本来说，官方提供的模块一共有 200 多个，这些模块大致可以分为以下几类

(模块间的依赖关系位于"moddep.lst"文件中):

命令模块[command.lst]

提供了各种不同的功能，类似标准 Unix 命令，一共将近 100 个。例如：cat cpuid echo halt lspci chainloader initrd linux password ...

加密模块[crypto.lst]

提供了各种数据完整性校验与密码算法支持，一共 20 多个。例如：gcry_rijndael crc64 gcry_md5 ...

文件系统模块[fs.lst]

提供了访问各种文件系统的功能，一共 30 多个。例如：btrfs cpio exfat ext2 fat iso9660 ntfs tar xfs zfs ...

分区模块[partmap.lst]

提供了识别各种分区格式的功能，一共 10 多个。例如：part_bsd part_gpt part_msdos ...

分区工具[parttool.lst]

提供了操作各种分区格式的功能，目前只有 msdospart 这一个。

终端模块[terminal.lst]

提供了各种不同终端的支持，一共不到 10 个。例如：serial gfxterm vga_text at_keyboard ...

视频模块[video.lst]

提供了各种不同的视频模式支持，一共 6 个。例如：vga vbe efi_gop efi_uga ...

其他模块

所有未在上述分类文件中列出的模块都归为这一类，一共将近 100 个。值得关注的有以下几个：

- "all_video"可用于一次性加载当前所有可用的视频模块；
- "gfxmenu"可用于提供主题支持；
- "jpeg png tga"可用于提供特定格式的背景图片支持；
- "xzio gzio lzopio"可用于提供特定压缩格式支持(常配合"initrd"命令使用)；

GRUB2 救援模式

GRUB2 在 BIOS 平台上的常规启动步骤是这样的：BIOS --> boot.img[MBR] --> core.img[MBR gap/embedding area/BIOS Boot Partition] --> 设置"prefix root cmdpath"环境变量 --> 加载"normal.mod"模块[同时还包括它所依赖的 terminal crypto extcmd boot gettext 模块] --> 执行"normal \$prefix/grub.cfg"命令

GRUB2 在 UEFI 平台上的常规启动步骤是这样的：UEFI --> core.img[BOOTX64.EFI/BOOTX86.EFI] --> 设置"prefix root cmdpath"环境变量 --> 加

载"normal.mod"模块[同时还包括它所依赖的 terminal crypto extcmd boot gettext 模块] --> 执行"normal \$prefix/grub.cfg"命令

如果上述步骤全部成功，那么你将进入'普通模式'，一般是显示一个菜单(找到了'\$prefix/grub.cfg')，或者直接进入 GRUB SHELL(没找到'\$prefix/grub.cfg')。在普通模式中，命令模块[command.lst]与加密模块[crypto.lst]会被自动按需载入(无需使用"insmod"命令)，并且可使用完整的 GRUB 脚本功能。但是其他模块则可能需要明确使用"insmod"命令来载入。

如果在加载"normal.mod"模块这一步出现故障，那么你将进入 GRUB2 的'救援模式'，而不是常规的'普通模式'。在救援模式中，GRUB 只自动设置了"cmdpath prefix root"三个环境变量，并且只能使用"insmod ls set unset"四个命令。只有当额外的模块被加载之后，才可以使用一些其它的命令,变量,解析器,驱动程序。通常来说，进入救援模式可能意味着你的 GRUB2 没有正确安装。请认真阅读'grub-install --help'的输出选项，并使用正确的选项重新安装。更多细节请参考 GRUB2 手册中的"[GRUB only offers a rescue shell](#)"部分。

GRUB2 命名规则

设备与分区

GRUB2 对设备与分区的命名规则举例如下，看看就能明白。需要说明的是磁盘从"0"开始计数，分区从"1"开始计数。

(fd0)	第一软盘
(hd0)	第一硬盘[大多数 U 盘与 USB 接口的移动硬盘以及 SD 卡也都被当作硬盘看待]
(hd1,1)	第二硬盘的第一分区(通用于 MBR 与 GPT 分区)
(hd0,msdos2)	第一硬盘的第二 MBR 分区，也就是传统的 DOS 分区表
(hd1,msdos5)	第二硬盘的第五 MBR 分区，也就是第一个逻辑分区
(hd0,gpt1)	第一硬盘的第一 GPT 分区
(cd)	启动光盘[仅在从光盘启动 GRUB 时可用]
(cd0)	第一光盘

上面所举的例子仅是最常用的情形，更多高级的设备命名规则请参考 GRUB2 手册中的"[Naming convention](#)"与"[How to specify devices](#)"部分。此外，如果你想看看当前系统上有哪些设备可用，可以在 GRUB SHELL 中使用"ls"命令(可能需要先加载必要的驱动模块)。

文件

文件的命名方法有两种：(1)绝对路径表示法，(2)相对路径表示法。举例如下：

(fd0)/grldr	第一软盘根目录下的"grldr"文件[绝对路径]
(hd0,gpt1)/boot/vmlinuz	第一硬盘的第一 GPT 分区"boot"目录下的"vmlinuz"文件[绝对路径]
/boot/vmlinuz	根设备"boot"目录下的"vmlinuz"文件[相对路径]， 当"root"环境变量等于"(hd0,gpt1)"时，等价于"(hd0,gpt1)/boot/vmlinuz"

上面所举的例子仅是最常用的情形，更多高级的文件命名规则请参考 GRUB2 手册中的"[How to](#)

[specify files](#)"部分。

磁盘块

磁盘块的命名方法同样也有两种：(1)绝对路径表示法，(2)相对路径表示法。举例如下：

(hd1,1)0+1	在第二硬盘的第一分区上，从第"0"个磁盘块(首扇区)起，长度为"1"的连续块。[绝对路径]
(hd1,1)+1	含义与上一个相同，因为当从第"0"个磁盘块(首扇区)起时，"0"可以省略不写。[绝对路径]
+1	在根设备上，从第"0"个磁盘块(首扇区)起，长度为"1"的连续块。[相对路径]
	当"root"环境变量等于"(hd1,1)"时，等价于"(hd1,1)0+1"

磁盘块几乎只用于链式引导(chainloader)的场合。更多高级的磁盘块命名规则请参考 GRUB2 手册中的"[How to specify block lists](#)"部分。

GRUB2 环境变量

GRUB2 的环境变量大致可以分为两类，第一类是自动设置的变量，也就是这些变量的初始值由 GRUB2 自动设置，其值必定存在且不为空。第二类是手动设置的变量，它们没有初始值(或者初始值为空)，需要经过手动明确设置之后才能使用。

大多数有特定含义的环境变量都是附属于特定附加模块的，只有加载了这些模块之后，这些环境变量才变得有意义。所以从模块的角度看，GRUB2 的环境变量又可以分为三类：(1)核心变量，GRUB2 核心提供的变量，不依赖于任何可加载模块，这样的变量只有"cmdpath prefix root"三个，而且它们的初始值都由 GRUB2 自动设置。(2)模块变量，绝大多数有特定含义的环境变量都属此类。(3)脚本变量，这是为了方便编写 grub.cfg 脚本而设置的变量，没有特殊含义，也不依赖于特定模块，与一般的 bash 脚本中的变量类似。有关 GRUB2 脚本的完整说明可以参考 GRUB2 手册中的"[Writing full configuration files directly](#)"部分，基本上其语法与 bash 脚本完全一致，上手非常容易。

特殊变量

下面列出的变量都是有特定含义的变量，这里只列出常用的一些变量，完整的列表可以参考 GRUB2 手册中的"[Special environment variables](#)"部分。

?

上一条命令的返回值，零表示成功，非零表示失败[与 bash 一样]。由 GRUB2 自动设置。你只能使用此变量，而不能修改它。

check_signatures

是否在加载文件时强制验证签名，可以设为'yes'或'no'

chosen

当前被执行的菜单项名称(紧跟"menuentry"命令之后的字符串或者'--id'选项的参数)，例如'Windows 7'。由 GRUB2 自动设置。你只应该使用此变量，而不应该修改它。

cmdpath

当前被加载的"core.img"所在目录(绝对路径)。例如：UEFI 启动可能是'(hd0,gpt1)/EFI/UBUNTU'或'(cd0)/EFI/BOOT'，BIOS 启动可能是'(hd0)'。由 GRUB2 自动设置。你只应该使用此变量，而不应该修改它。

debug

设为'all'时表示开启调试输出[会显示大量信息,谨慎开启]

default

默认选中第几个菜单项(从'0'开始计数)

fallback

如果默认菜单项启动失败，那么就启动第几个菜单项(从'0'开始计数)

gfxmode

设置"gfxterm"模块所使用的视频模式，可以指定一组由逗号或分号分隔的模式以供逐一尝试：每个模式的格式必须是：'auto'(自动检测),'宽 x 高','宽 x 高 x 色深'之一，并且只能使用 VBE 标准指定的模式[640x480,800x600,1024x768,1280x1024]x[16,24,32]。可以在 GRUB SHELL 中使用"videoinfo"命令列出当前所有可用模式。默认值是'auto'。

gfxpayload

设置 Linux 内核启动时的视频模式，可以指定一组由逗号或分号分隔的模式以供逐一尝试：每个模式的格式必须是：'text'(普通文本模式,不能用于 UEFI 平台),'keep'(继承"gfxmode"的值),'auto'(自动检测),'宽 x 高','宽 x 高 x 色深'之一，并且只能使用 VBE 标准指定的模式[640x480,800x600,1024x768,1280x1024]x[16,24,32]。在 BIOS 平台上的默认值是'text'，在 UEFI 平台上的默认值是'auto'。除非你想明确设置 Linux 控制台的分辨率(要求内核必须"CONFIG_FRAMEBUFFER_CONSOLE=y")，或者打算在 BIOS 平台上使用图形控制台(要求内核必须"CONFIG_FRAMEBUFFER_CONSOLE=y")，否则不要设置此变量。

gfxterm_font

设置"gfxterm"模块所使用的字体，默认使用所有可用字体

grub_cpu

此 GRUB 所适用的 CPU 类型。例如：'i386','x86_64'。由 GRUB2 自动设置。你只应该使用此变量，而不应该修改它。

grub_platform

此 GRUB 所适用的平台类型。例如：'pc','efi'。由 GRUB2 自动设置。你只应该使用此变量，而不应该修改它。

lang

设置 GRUB2 的界面语言，必须搭配"locale_dir"变量一起使用。简体中文应设为'zh_CN'。

locale_dir

设置翻译文件(*.mo)的目录，通常是'\$prefix/locale'，若未明确设置此目录，则禁止国际化。

pager

如果设为'1'，那么每一满屏后暂停输出，等待键盘输入。缺省是"，表示不暂停。

prefix

绝对路径形式的'/boot/grub'目录位置(也就是 GRUB2 的安装目录)，例如'(hd0,gpt1)/grub'或'(hd0,msdos2)/boot/grub'。初始值由 GRUB 在启动时根据"grub-install"在安装时提供的信息自动设置。你只应该使用此变量，而不应该修改它。

root

设置"根设备"。任何未指定设备名的文件都视为位于此设备。初始值由 GRUB 在启动时根据"prefix"变量的值自动设置。在大多数情况下，你都需要修改它。

superusers

设置一组"超级用户"(使用空格/逗号/分号进行分隔)，以开启安全认证的功能。

theme

设置菜单界面的主题风格文件的位置，例如："/boot/grub/themes/starfield/theme.txt"。关于如何定制界面风格(背景图片/字体/颜色/图标等)的细节，可以参考 GRUB2 手册中的"[Theme file format](#)"部分。

timeout

在启动默认菜单项前，等待键盘输入的秒数。默认值是'5'秒。'0'表示直接启动默认菜单项(不显示菜单)，'-1'表示永远等待。

GRUB2 命令

对于 GRUB-2.0.2 版本来说，所有可用的命令有大约 200 个之多，他们中的绝大多数由各种各样的模块提供。我们没有必要去了解所有这些 200 个命令，只需要了解一些常用的命令即可(实际上就连官方文档也没有给出全部的命令说明)。更多的命令说明可以参考 GRUB2 手册中的"[The list of available commands](#)"页面中列出的几个二级页面。

```
menuentry "title" [--class=class ...] [--users=users] [--unrestricted] [--hotkey=key] [--id=id] [arg ...]
{ command; ... }
```

定义一个名为"title"的菜单项。当此菜单项被选中时，GRUB 将会把环境变量"chosen"的值设为"id"(使用了[--id=id]选项)或"title"(未使用[--id=id]选项)，然后执行花括号中的命令列表，如果列表中最后一个命令执行成功，并且已经载入了一个内核，那么将执行"boot"命令。

可以使用 --class 选项指定菜单项所属的"样式类"。从而可以使用指定的主题样式显示菜单项。

可以使用 --users 选项指定只允许特定的用户访问此菜单项。如果没有使用此选项，则表示允许所有用户访问。

可以使用 --unrestricted 选项指明允许所有用户访问此菜单项。

可以使用 --hotkey 选项设置访问此菜单项的热键(快捷键)。“key”可以是一个单独的字母，或者'backspace','tab','delete'之一。

可以使用 `--id` 选项为此菜单项设置一个全局唯一的标识符。"id"必须由 ASCII 字母/数字/下划线组成，且不得以数字开头。

[arg ...]是可选的参数列表。你可以把它们理解为命令行参数。实际上"title"也是命令行参数，只不过这个参数是个必须参数而已。这些参数都可以在花括号内的命令列表中使用，"title"对应着"\$1"，其余的以此类推。

`terminal_input [--append|--remove] [terminal1] [terminal2] ...`

如果不带任何选项与参数，则表示列出当前激活的输入终端，以及所有其他可用的输入终端。

可以使用 `--append` 选项将指定的终端加入到激活的输入终端列表中，所有列表中的终端都可以用于向 GRUB 提供输入。

可以使用 `--remove` 选项将指定的终端从激活的输入终端列表中删除。

如果不使用任何选项，但是指定了一个或多个终端参数，则表示将当前激活的输入终端设置为参数指定的终端。

`terminal_output [--append|--remove] [terminal1] [terminal2] ...`

如果不带任何选项与参数，则表示列出当前激活的输出终端，以及所有其他可用的输出终端。

可以使用 `--append` 选项将指定的终端加入到激活的输出终端列表中，所有列表中的终端都将接受到 GRUB 的输出。

可以使用 `--remove` 选项将指定的终端从激活的输出终端列表中删除。

如果不使用任何选项，但是指定了一个或多个终端参数，则表示将当前激活的输出终端设置为参数指定的终端。

`authenticate [userlist]`

检查当前用户是否位于"userlist"或环境变量"superusers"中。[注意]如果环境变量"superusers"的值为空，此命令将返回'真'。

`background_color color`

设置当前激活的输出终端的背景颜色。"color"可以使用 HTML 风格的颜色表示法("#RRGGBB"或"#RGB")。

[注意]仅在使用'gfxterm'作为输出终端的时候，才能改变背景色。

`background_image [--mode 'stretch'|'normal'] file`

将当前激活的输出终端的背景图片设置为"file"文件。除非使用了"--mode 'normal'"选项，否则图片将被自动缩放以填满整个屏幕。

如果不带任何选项与参数，则表示删除背景图片。

[注意]仅在使用'gfxterm'作为输出终端的时候，才能改变背景图片。

`boot`

启动已经被载入的 OS 或链式加载器。仅在运行于交互式命令行的时候才是需要的。在一个

菜单项结束时是隐含的。

cat [--dos] file

显示文件"file"的内容。如果使用了"--dos"选项，那么"回车/换行符"将被显示为一个简单的换行符。否则，回车符将被显示为一个控制符(<d>)。

chainloader [--force] file

链式加载"file"文件。通常使用磁盘块表示法，例如用'+1'表示当前根分区的第一个扇区。

可以使用 --force 选项强制载入文件，而不管它是否有正确的签名。通常用于加载有缺陷的启动载入器(例如 SCO UnixWare 7.1)。

configfile file

将"file"作为配置文件加载。如果"file"中定义了菜单项，那么立即显示一个包含它们的菜单。

[注意]"file"文件对环境变量所做的任何变更都将在从此文件返回后失效。

cpuid [-l]

检查 CPU 特性。仅在 x86 系统上可用。

如果使用了 -l 选项，那么如果 CPU 是 64 位则返回真，否则返回假。

drivemap -l|-r[[-s] from_drive to_drive

如果不使用任何选项，表示将"from_drive"映射到"to_drive"。这主要用于链式加载 Windows 之类的操作系统，因为它们只能从第一个硬盘启动。出于方便的原因，分区后缀将被忽略，因此你可用安全地将"\${root}"作为磁盘使用。

可以使用 -s 选项，执行反向映射，也就是交换这两个磁盘。例如： drivemap -s (hd0) (hd1)

可以使用 -l 选项，列出当前已有的映射。

可以使用 -r 选项，把映射重置为默认值，也就是撤销所有当前已有的映射。

echo [-n] [-e] string ...

显示所要求的文本并换行(除非使用了 -n 选项)。如果有多个字符串，依次输出它们，并用空格分隔每一个。

和 bash 的习惯一样，可以在双引号内使用"\${var}"来引用变量的值，也可以使用 -e 选项激活对反斜杠转义符的解释(\\ \a \r \n \t ...)。

export envvar

导出环境变量"envvar"，以使其对于使用"configfile"命令载入的配置文件可见。

false

不做任何事，只返回一个失败的结果。主要用在 if/while 之类的控制构造中。

gettext string

把"string"翻译为环境变量"lang"指定的语言。MO 格式的翻译文件从环境变量"locale_dir"指

定的目录加载。

halt [--no-apm]

关闭计算机。如果指定了 --no-apm 选项，表示不执行 APM BIOS 调用。否则，计算机使用 APM 关闭。

help [pattern ...]

显示内建命令的帮助信息。如果没有指定"pattern"，那么将显示所有可用命令的简短描述。

如果指定了"pattern"，那么将只显示名字以这些"pattern"开头的命令的详细帮助信息。

initrd file

为以 32 位协议启动的 Linux 内核载入一个"initial ramdisk"，并在内存里的 Linux 设置区域设置合适的参数。

[注意]这个命令必须放在"linux"命令之后使用。

initrd16 file

为以 16 位协议启动的 Linux 内核载入一个"initial ramdisk"，并在内存里的 Linux 设置区域设置合适的参数。

[注意]这个命令必须放在"linux16"命令之后使用。

insmod module

载入名为"module"的 GRUB2 模块。

linux file ...

使用 32 位启动协议从"file"载入一个 Linux 内核映像，并将其余的字符作为内核的命令行参数逐字传入。

[注意]使用 32 位启动协议意味着'vga='启动选项将会失效。如果你希望明确设置一个特定的视频模式，那么应该使用"gfxpayload"环境变量。虽然 GRUB 可以自动地检测某些'vga='参数，并把它们翻译为合适的"gfxpayload"设置，但是并不建议这样做。

linux16 file ...

以传统的 16 位启动协议从"file"载入一个 Linux 内核映像，并将其余的字符作为内核的命令行参数逐字传入。这通常用于启动一些遵守 Linux 启动协议的特殊工具(例如 [MEMDISK](#))。

[注意]使用传统的 16 位启动协议意味着：(1)'vga='启动选项依然有效，(2)不能启动纯 64 位内核(也就是内核必须要'CONFIG_IA32_EMULATION=y'才行)。

loadfont file ...

从指定的"file"加载字体，除非使用了绝对路径，否则"file"将被视为"\$prefix/fonts/file.pf2"文件。

loopback [-d] device file

将"file"文件映射为"device"回环设备。例如：

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

可以使用 -d 选项，删除先前使用这个命令创建的设备。

ls [arg ...]

如果不使用参数，那么列出所有对 GRUB 已知的设备。

如果参数是包含在括号内的一个设备名，那么列出该设备根目录下的所有文件。

如果参数是以绝对路径给出的目录，那么列出这个目录的内容。

lsfonts

列出已经加载的所有字体

lsmod

列出已经加载的所有模块

normal [file]

进入普通模式，并显示 GRUB 菜单。[说明]只要当前没有处于救援模式，其实就已经是在普通模式中了，所以通常并不需要明确使用此命令。

在普通模式中，命令模块[command.lst]与加密模块[crypto.lst]会被自动按需载入(无需使用"insmod"命令)，并且可使用完整的 GRUB 脚本功能。但是其他模块则可能需要明确使用"insmod"命令来载入。

如果给出了"file"参数，那么将从这个文件中读入命令(也就是作为"grub.cfg"的替代)，否则将从"\$prefix/grub.cfg"中读入命令(如果存在的话)。你也可以理解为"file"的默认值是'\$prefix/grub.cfg'。

可以在普通模式中嵌套调用此命令，以构建一个嵌套的环境。不过一般不这么做，而是使用"configfile"命令来达到这目的。

normal_exit

退出当前的普通模式。如果这个普通模式实例不是嵌套在另一个普通模式里的话，就会返回到救援模式。

parttool partition commands

对分区表进行各种修改。目前只能作用于 MBR 分区表(DOS 分区表)，而不能用于 GPT 分区表。目前仅支持以下三种用法：

(1)设置或去掉分区的激活标记(仅对 Windows 系统有意义)。

例如："parttool (hd0,msdos2) +boot"表示为(hd0,msdos2)分区加上激活标记，而"parttool (hd0,msdos2) -boot"则表示去掉(hd0,msdos2)分区的激活标记。

(2)设置或去掉分区的隐藏标记(仅对 Windows 系统有意义)。

例如："parttool (hd0,msdos2) +hidden"表示为(hd0,msdos2)分区加上隐藏标记，而"parttool (hd0,msdos2) -hidden"则表示去掉(hd0,msdos2)分区的隐藏标记。

(3)更改分区的类型。其值必须是 0x00-0xFF 范围内的值。且应该使用'0xNN'格式的十六进制数。

例如: "parttool (hd0,msdos2) type=0x83"表示将(hd0,msdos2)分区类型修改为'0x83'(Linux 分区)。

password user clear-password

定义一个名为 user 的用户, 并使用明文口令'clear-password'。不建议使用此命令。

password_pbkdf2 user hashed-password

定义一个名为 user 的用户, 并使用哈希口令'hashed-password'(通过"grub-mkpasswd-pbkdf2"工具生成)。这是建议使用的命令, 因为它安全性更高。

probe [--set var] --driver|--partmap|--fs|--fs-uuid|--label device

提取"device"设备的特定信息。如果使用了 --set 选项, 则表示将提取的结果保存在"var"变量中, 否则将提取的结果直接显示出来。

read [var]

从用户读取一行输入。如果给定环境变量"var", 则把它设为所读取的行(不包括结尾的换行符)。

reboot

重新启动

rmmod module

卸载"module"模块

search [--file|--label|--fs-uuid] [--set [var]] [--no-floppy] name

通过文件[--file]、卷标[--label]、文件系统 UUID[--fs-uuid]来查找设备。

如果使用了 --set 选项, 那么会将第一个找到的设备设置为环境变量"var"的值。默认的"var"是'root'。

可以使用 --no-floppy 选项来禁止查找软盘设备, 因为这些设备非常慢。

set [envvar=value]

将环境变量"envvar"的值设为'value'。如果没有使用参数, 则打印出所有环境变量及其值。

source file

直接将"file"文件的内容插入到当前位置。与"configfile"不同, 此命令既不切换执行环境, 也不会显示一个新的菜单。

test expression

[expression]

计算"expression"的值, 并在结果为真时返回零值, 或者在结果为假时返回非零值, 主要用在 if/while 之类的控制构造中。

可用的"expression"模式如下(与 bash 类似):

```
string1 == string2  [string1 与 string2 完全相同]
string1 != string2  [string1 与 string2 不完全相同]
string1 < string2   [string1 在字母顺序上小于 string2]
string1 <= string2  [string1 在字母顺序上小于 string2 或与 string2 完全相同]
string1 > string2   [string1 在字母顺序上大于 string2]
string1 >= string2  [string1 在字母顺序上大于 string2 或与 string2 完全相同]
integer1 -eq integer2  [integer1 等于 integer2]
integer1 -ge integer2  [integer1 大于或等于 integer2]
integer1 -gt integer2  [integer1 大于 integer2]
integer1 -le integer2  [integer1 小于或等于 integer2]
integer1 -lt integer2  [integer1 小于 integer2]
integer1 -ne integer2  [integer1 不等于 integer2]
prefixinteger1 -pgt prefixinteger2  [剔除非数字字符首部之后, integer1 大于 integer2]
prefixinteger1 -plt prefixinteger2  [剔除非数字字符首部之后, integer1 小于 integer2]
file1 -nt file2  [file1 的修改时间比 file2 新]
file1 -ot file2  [file1 的修改时间比 file2 旧]
-d file  [file 存在并且是一个目录]
-e file  [file 存在]
-f file  [file 存在并且不是一个目录]
-s file  [file 存在并且文件尺寸大于零]
-n string  [string 的长度大于零]
string    [string 的长度大于零]
-z string  [string 的长度等于零]
( expression )  将 expression 视为一个整体(分组)
! expression    非(NOT)
expression1 -a expression2    与(AND)
expression1 -o expression2    或(OR)
```

true

不做任何事, 只返回一个成功的结果。主要用在 if/while 之类的控制构造中。

unset envvar

撤销环境变量"envvar"

videoinfo [[WxH]xD]

列出所有当前可用的视频模式。如果指定了分辨率(或者还附加了色深), 那么仅显示与其匹配的模式。

GRUB2 安全

在默认情况下, GRUB 对于所有可以在物理上进入控制台的人都是可访问的。任何人都可以选择并编辑任意菜单项, 并且可以直接访问 GRUB SHELL。要启用认证支持, 必须将环境变量"superusers"设置为一组用户名(可用空格/逗号/分号作为分隔符), 这样, 将仅允许"superusers"中的用户使用 GRUB 命令行、编辑菜单项、以及执行任意菜单项。而其他

非"superusers"中的用户，只能执行那些没有设置 --users 选项的菜单，以及那些在 --users 选项中包含了该用户的菜单，但不能使用 GRUB 命令行、编辑菜单项。下面使用一个配置片段举例说明：

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

menuentry "所有人都可以执行此菜单" --unrestricted {
    ...
}

menuentry "仅允许超级用户执行此菜单" --users "" {
    ...
}

menuentry "允许 user1 和超级用户执行此菜单" --users user1 {
    ...
}
```

有关 GRUB2 安全的更多详情，请参考 GRUB2 手册中的["Security"](#)部分。

GRUB2 实用技巧

如何给 GRUB2 菜单加上背景图？

首先制作一张 PNG 格式的图片，分辨率最好是"1024x768"以保证较好的兼容性。然后将这张图片放到"\$prefix/themes/1024x768.png"("\$prefix"是 GRUB2 的安装目录)。然后在'grub.cfg'中加入如下内容：

```
set gfxmode=1024x768,auto
insmod gfxterm
insmod png
terminal_output gfxterm
background_image $prefix/themes/1024x768.png
```

如何让 GRUB2 显示中文界面(包括显示中文菜单项)？

由于 GRUB2 在内部使用 UTF-8 编码，并且所有文本文件(包括'grub.cfg')也都被假定为使用 UTF-8 编码，为了避免乱码，请务必以 UTF-8 编码保存'grub.cfg'文件。

```
set gfxterm_font=unicode
set lang=zh_CN
set locale_dir=$prefix/locale
insmod gfxterm
terminal_output gfxterm
loadfont unicode
```

如何更改 GRUB2 的字体？

如果你认为默认的 unicode 字体在 1024x768 或更高分辨率的屏幕上显得太小，或者你认为默认的字体不好看，想换换口味，那么如何自己动手制作一个 pf2 字体呢？那就要用到"grub-mkfont"工具。下面的示例展示了如何从一个 ttc 字体(文泉驿等宽微米黑)制作一个 24px 大小的 pf2 字体：

```
grub-mkfont -i1 -n WenQuanYiMicroHeiMono24px -o WenQuanYiMicroHeiMono24px.pf2 -s24 -v wqy-microhei.ttc
```

将制作好的字体文件(WenQuanYiMicroHeiMono24px.pf2)放到"\$prefix/fonts"目录中，修改'grub.cfg'文件中的两行：

```
set gfxterm_font=WenQuanYiMicroHeiMono24px
loadfont WenQuanYiMicroHeiMono24px
```

[注意]你最好使用等宽中文字体(推荐使用文泉驿等宽正黑或者等宽微米黑)，否则可能会让 GRUB2 的字体间距过大，十分难看。

如何使用 GRUB2 引导 WindowsPE 的 ISO 文件？

GRUB4DOS 有一个非常酷的'磁盘映射'功能，能够用于启动 WinPE 的 ISO 文件。其实，将 GRUB2 配合 [MEMDISK](#) 工具使用，同样可以引导各种镜像文件，包括 ISO 文件与软/硬盘镜像。

首先，你必须安装或者下载"[syslinux](#)"软件包，从中提取出"memdisk"文件(可能位于'/usr/share/syslinux/memdisk'或'bios/memdisk/memdisk')，然后将它复制到 GRUB2 的安装目录中，也就是位于"\$prefix/memdisk"。

然后，再将你想要引导的 WindowsPE 的 ISO 文件放到某个地方，这里假定你和"memdisk"放在一起，也就是位于"\$prefix/WinPE.ISO"。当然，为了节约磁盘空间，你也可以用 gzip 对 ISO 文件进行压缩，不过这个示例中没有这么做。

最后，在'grub.cfg'中加入如下菜单项(如果你对 ISO 进行了 gzip 压缩，那么还需要额外再加上"insmod gzio"命令)：

```
menuentry "Windows PE" --unrestricted {
    linux16 $prefix/memdisk iso raw
    initrd16 $prefix/WinPE.ISO
}
```

[注意]与 GRUB4DOS 一样，由于 [MEMDISK](#) 对各种镜像文件的模拟是通过在实模式下拦截 BIOS 的 INT 13h 与 INT 15h 调用来实现的，所以有很大的局限性：

- 只能用于 BIOS 模式启动，不能用于 UEFI 模式启动
- 模拟出来的软盘/光盘/硬盘设备只能被基于实模式的操作系统所识别(DOS,FreeDOS)，不能被基于保护模式的操作系统所识别(Windows,Linux,BSD)
- 从实用的角度来说，只能用于引导 WinPE 的 ISO 以及基于 DOS/FreeDOS 的镜像，不能用

于引导各种 Linux 的 LiveCD ISO 以及微软原版的 Windows ISO 安装光盘。

[释疑]不要将 [MEMDISK](#) 与 GRUB2 的 "memdisk.mod" 模块混淆，他们两个毫不相干的东西。"memdisk.mod" 模块的作用是为 'core.img' 提供内存盘支持，其目的是为了 GRUB2 能够正确识别启动设备。如果把 'core.img' 比作 Linux 内核，那么 "memdisk.mod" 模块的作用就相当于为 'core.img' 这个 "内核" 提供了 'initramdisk'，用以加载磁盘与文件系统驱动，从而让 GRUB2 可以访问磁盘，进而加载其他的模块。但是实际上，在绝大多数情况下，更本不需要使用 "memdisk.mod"，因为在 'grub-install' 的时候，这些驱动已经被嵌入到 'core.img' 中了。如果你还是不明白 "memdisk.mod" 模块的作用的话，那就忘记它的存在吧，因为你更本不需要知道有这个东​​西，就像你不需要了解全部 GRUB2 模块一样。仅仅是因为这两个东西的名字相同，我才在这里多罗嗦了几句而已。

如何使用 GRUB2 引导(硬盘安装) Gentoo LiveCD 的 ISO 文件？

以 "install-amd64-minimal-*.iso" 为例。

第一步，将 ISO 中的 '/isolinux/{gentoo,gentoo.igz}'、'/image.squashfs' 三个文件放到 '(hd0,gpt3)/OS/gentoo/' 目录中；

第二步，将 ISO 中的 '/livecd' 放到相同分区 (hd0,gpt3) 的根目录下；

最后，在 'grub.cfg' 中加入如下菜单项：

```
menuentry "Gentoo Minimal Install LiveCD" --unrestricted {
    linux (hd0,gpt3)/OS/gentoo/gentoo cdroot looptype=squashfs
    loop=/OS/gentoo/image.squashfs
    initrd (hd0,gpt3)/OS/gentoo/gentoo.igz
}
```

[说明]'livecd' 是寻找 'image.squashfs' 所在磁盘分区的关键。

如何使用 GRUB2 引导(硬盘安装) Debian/Ubuntu/Fedora/RHEL/CentOS LiveCD 的 ISO 文件？

首先需要说明的是，这里给出的方法，只适用于提供了 "iso-scan/filename=" 或 "fromiso=" 之类参数的 LiveCD。不适用于未提供此类参数的 LiveCD (例如 Gentoo)。

下面以 'Debian 8.1 LiveCD' 为例说明。首先，假定你将 ISO 文件放在 '(hd0,gpt3)/ISO/Debian.iso'；然后，在 'grub.cfg' 中加入如下菜单项：

```
menuentry "Debian 8.1 LiveCD" --unrestricted {
    loopback loop0 (hd0,gpt3)/ISO/Debian.iso
    linux (loop0)/live/vmlinuz boot=live fromiso=/dev/sda3/ISO/Debian.iso
    initrd (loop0)/live/initrd.img
}
```

[说明]这里给出的方法，其实就是各种 "硬盘安装 XX Linux" 的翻版，只不过不再需要将 "vmlinuz" 与 "initrd" 从 ISO 中解压出来而已。

更多其它发行版的实例，请继续阅读下面的内容。

"grub.cfg"实例

下面是本文作者实际使用的一个"grub.cfg"文件，通用于 BIOS 与 UEFI 模式，放在这里当作一个实例，供读者参考：

```
#####
## (1)特殊变量 ##
#####
#默认启动第一个菜单项
set default=0
#如果第一个菜单项启动失败，转而启动第二个菜单项
set fallback=1
#优先使用最常规的 1024x768 分辨率，以保证在不同的屏幕上拥有一致的菜单效果，如果失败再自动匹配分辨率
set gfxmode=1024x768,auto
#使用自己制作的 24px 的大号字体以避免默认字体太小看不清
set gfxterm_font=WenQuanYiMicroHeiMono24px
#将 GRUB2 设置为简体中文界面
set lang=zh_CN
#指定翻译文件 (*.mo) 的目录，若未明确设置此目录，则无法显示中文界面。
set locale_dir=$prefix/locale
#每一满屏后暂停输出，以免信息太多一闪而过看不清
set pager=1
#开启密码验证功能，并设置一个名为 'root' 的超级用户
set superusers=root
#设置菜单的超时时间为 5 秒
set timeout=5

#####
## (2)公共模块 ##
#####
#两种最流行的磁盘分区格式
insmod part_gpt
insmod part_msdos
#常见文件系统驱动
insmod btrfs
insmod exfat
insmod ext2
insmod fat
insmod iso9660
insmod jfs
insmod ntfs
insmod reiserfs
insmod udf
insmod xfs
insmod zfs
```

```
#一次性加载所有可用的视频驱动
insmod all_video
#图形模式终端
insmod gfxterm
#背景图片支持
insmod png

#####
## (3)公共命令(必须放在模块和变量之后) ##
#####
#激活图形模式的输出终端，以允许使用中文和背景图
terminal_output gfxterm
#设置背景图片
background_image $prefix/themes/1024x768.png
#加载自己制作的 24px 的大号字体文件($prefix/fonts/WenQuanYiMicroHeiMono24px.ttf)
loadfont WenQuanYiMicroHeiMono24px
#设置'root'用户的哈希密码[通过"grub-mkpasswd-pbkdf2"工具生成]
password_pbkdf2 root
grub.pbkdf2.sha512.69.7DBCA469F80EA1C0A8A1E2FEB4F8463.B073C1C89EC1E85309C3D6A1BAFF
4356

#####
## (4)菜单项 ##
#####

menuentry '正常启动(Windows)' --unrestricted {
    if search --file --set --no-floppy /bootmgr ; then
        chainloader +1
    elif search --file --set --no-floppy /ntldr ; then
        chainloader +1
    else
        echo '没有找到Windows'
        sleep --verbose 5
    fi
}

if [ 'pc' == $grub_platform ] ; then
    menuentry '系统救援(WinPE)' --users=root {
        if search --file --set --no-floppy /OS/WinPE.iso ; then
            linux16 $prefix/memdisk iso raw
            initrd16 /OS/WinPE.iso
        fi
    }
fi

# 硬盘安装 Gentoo
menuentry 'Gentoo LiveCD' --users=root {
```

```

        if search --file --set --no-floppy /OS/gentoo/image.squashfs ; then
            if [ -f /livecd -a -f /OS/gentoo/gentoo -a -f /OS/gentoo/gentoo.igz ] ;
then
                linux /OS/gentoo/gentoo cdroot looptype=squashfs
loop=/OS/gentoo/image.squashfs rootwait doscsi nodmraid nokeymap docache dosshd
scandelay slowusb passwd=123
                initrd /OS/gentoo/gentoo.igz
            fi
        fi
    }

# 硬盘安装 Debian [假定'/OS/Debian.iso'所在分区的卷标是'GRUB2'] [这种方法也适用于 Kali
Linux]
# [加上"username=root"后, 桌面上的安装快捷方式会消失, 需要使用"system tools -> install
Debian sid"安装]
menuentry 'Debian LiveCD' --users=root {
    if search --file --set --no-floppy /OS/Debian.iso ; then
        loopback loop0 /OS/Debian.iso
        linux (loop0)/live/vmlinuz boot=live fromiso=/dev/disk/by-
label/GRUB2/OS/Debian.iso rootwait ro username=root
        initrd (loop0)/live/initrd.img
    fi
}

# 硬盘安装 Ubuntu [加上"username=root"后, 桌面上的安装快捷方式会消失, 需要使用"system tools
-> install Ubuntu"安装]
menuentry 'Ubuntu LiveCD amd64' --users=root {
    if search --file --set --no-floppy /OS/Ubuntu.iso ; then
        loopback loop0 /OS/Ubuntu.iso
        linux (loop0)/casper/vmlinuz.efi boot=casper iso-
scan/filename=/OS/Ubuntu.iso rootwait ro username=root
        initrd (loop0)/casper/initrd.lz
    fi
}

# 硬盘安装 Fedora 22 [ISO 卷标是"Fedora-Live-LXDE-x86_64-22-3"]
# 这里的方法也适用于 CentOS 7 LiveCD [其卷标是"CentOS-7-livecd-x86_64"]
menuentry 'Fedora-Live-LXDE-x86_64-22-3' --users=root {
    if search --file --set --no-floppy /OS/Fedora.iso ; then
        loopback loop0 /OS/Fedora.iso
        set root=(loop0)
        linux /isolinux/vmlinuz0 iso-scan/filename=/OS/Fedora.iso
root=live:LABEL=Fedora-Live-LXDE-x86_64-22-3 ro rd.live.image
        initrd /isolinux/initrd0.img
    fi
}

#https://raw.githubusercontent.com/rhinstaller/anaconda/master/docs/boot-
options.txt

```

```
menuentry '网络安装 CentOS 7 x86_64 [只能用于简单网络环境，不可用于WiFi环境]'
--unrestricted {
    if search --file --set --no-floppy /OS/CentOS-7-x86_64-NetInstall-1503.iso ;
then
    loopback loop0 /OS/CentOS-7-x86_64-NetInstall-1503.iso
    set root=(loop0)
    linux /images/pxeboot/vmlinuz ip=dhcp nameserver=223.6.6.6
inst.repo=http://mirrors.aliyun.com/centos/7/os/x86_64/ inst.lang=zh_CN.UTF-8
    initrd /images/pxeboot/initrd.img
    fi
}

#[假定'/OS/CentOS-7-x86_64-Minimal-1503-01.iso'所在分区的卷标是'GRUB2']
# 这里的方法也适用于 CentOS-7-x86_64-DVD-1503-01.iso 与 CentOS-7-x86_64-Everything-
1503-01.iso
menuentry '硬盘安装 CentOS 7 x86_64 [最小安装]' --unrestricted {
    if search --file --set --no-floppy /OS/CentOS-7-x86_64-Minimal-1503-01.iso ;
then
    loopback loop0 /OS/CentOS-7-x86_64-Minimal-1503-01.iso
    linux (loop0)/isolinux/vmlinuz inst.repo=hd:LABEL=GRUB2:/OS/CentOS-7-
x86_64-Minimal-1503-01.iso inst.lang=zh_CN.UTF-8
    initrd (loop0)/isolinux/initrd.img
    fi
}
```
