

# GRUB 知识补充

## 目录

- [1 --- GRUB 基础](#)
- [2 --- 命名习惯](#)
- [3 --- 安装 GRUB](#)
- [4 --- GRUB 接口](#)
- [5 --- GRUB 常用命令](#)
- [6 --- 保护 GRUB 设置](#)
- [7 --- GRUB 和 stage 文件](#)
- [8 --- GRUB 常见错误信息](#)
- [9 --- 制作 GRUB 引导软盘](#)
- [10 --- 使用 GRUB 通过网络引导操作系统](#)
- [11 --- 把配置文件嵌入到 GRUB 中](#)
- [12 --- 总结](#)
- [13 --- 个人建议](#)

## 一、GRUB 基础

### 1.1 GRUB 简介

“boot loader”是计算机启动后第一个运行的软件。它负责加载一个操作系统的内核，并把控制权交给内核。内核则负责剩下的初始过程。GRUB 一个很重要的特色就是灵活性；GRUB 可以识别文件系统和二进制格式的内核，所以你可以加载任何操作系统，而不需要记录内核在磁盘上的物理位置。

### 1.2 GRUB 的现状

我不打算介绍 GRUB 的发展历史，有兴趣的朋友可以自己到官方站点或者执行 `info grub` 自行获取信息。我们着眼于 GRUB 当前的现状。目前 GRUB 分成 GRUB legacy 和 GRUB 2。版本号是 0.9x 以及之前的版本都称为 GRUB Legacy，从 1.x 开始的就称为 GRUB 2。目前 GRUB Legacy 已经停止开发了，只是出于一个 bug fix 的状态，不再增加新的功能了，所有的开发都转移到 GRUB 2 之上了。

### 1.3 GRUB Legacy 和 GRUB 2

GRUB 2 是源自于一个名为 PUPA 的项目，目的是建立下一代的 boot loader。GRUB 2 几乎是从零开始开始开发的，它和 GRUB Legacy 不同的地方有很多，例如支持 scripts、国际化语言、动态加载模块、带有 resume 模

式等等，这些都是 GRUB Legacy 所没有的。

下面是 GRUB2 新特性的介绍：

(原文请参考<http://www.gnu.org/software/grub/grub-2.en.html>.)

- 支持脚本语言，例如条件判断，循环、变量和函数
- 图形接口
- 在运行时动态加载需要的模块，而不是在 build 的时候指定加入那些模块（静态）
- 可以移植到不同的架构上
- 国际化语言。包括支持非 ASCII 的字符集和类似 gettext 的消息分类，字体，图形控制台等等。
- 实际内存管理。使得 GNU GRUB 更加具有可扩展性
- 针对文件系统、文件、设备、驱动、终端、命令、分区表、os loader 的模块化、层次化、基于对象的框架
- 跨平台的安装，允许从另外一个架构上安装 GRUB
- 支持 rescue 模式，可用于系统无法引导的情况。stage 1.5 被取消了
- 修复 GRUB Legacy 设计上的错误，同时又保持向后的兼容性。

## 1.4 Multiboot 规范

Multiboot 规范是在 boot loader 和 kernel 之间的一个协议，用于统一各种引导操作系统的方法。Linux 和 BSD 目前并不支持该规范。

## 1.5 可用资源

- GRUB 的 homepage (<http://www.gnu.org/software/grub/grub-legacy-faq.en.html>)
- info grub

## 1.6 旧版 GRUB (GRUB Legacy) 的功能特色

- 识别多种可执行格式，支持 “a.out”，ELF, symbol tables)
- 支持 non-Multiboot 内核。主要是 FreeBSD、NetBSD、OpenBSD、Linux。
- 支持加载多个模块
- 支持配置文件
- 提供菜单接口。菜单项的数量并没有限制，目前版本支持数百个菜单条目
- 有一个灵活的命令行接口。如果没有配置文件存在，GRUB 会自动进入命令模式
- 支持多种文件系统格式。主要有 BSD FFS”、“DOS FAT 16 、FAT32”，“Minix fs”、ext2/ext3、ReiserFS 、jfs、xfs、VSTa fs
- 支持自动解压

- 能够访问任何已经安装的设备上的数据
- **Be independent of drive geometry translations** （这个不知道怎么翻译好）

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.

- 检测所有已经安装的 RAM
- 支持 LBA 模式。在传统的磁盘调用（称为 CHS 模式），存在一个限制：就是 BIOS 无法访问在 1024 柱面之后的内容，所以 BIOS 只能访问最小 508 MB，最大 8GB 的空间。（这也就是 /boot 分区必须在 1024 柱面之前的原因）。GRUB 并不能解决全部问题，因为并没有统一的解决方法，不过 GRUB 可以检测是否支持 LBA 模式，如果支持则使用 LBA 模式。
- 支持网络引导，可以使用 TFTP 协议通过网络加载内核
- 支持远程终端，目前只支持串口

## 二、命名习惯

首先 GRUB 要求设备名被括在一个 ( ) 中。fd 表示软盘，hd 表示硬盘（不区分 IDE 还是 SCSI）。其次设备是从 0 开始编号，分区也是如此，分区和设备之间用一个 '/' 分开。

下面给出几个例子：

(fd0)：表示整个软盘

(hd0,1)：表示 BIOS 中的第一个硬盘的第 2 个分区

(hd0,0)/boot/vmlinuz：表示 BIOS 中的第一个硬盘的第一个分区下的 boot/ 目录下的 vmlinuz 文件。

如果没有指定某个分区，则表示使用整个设备，否则只使用指定的分区。

## 三、安装 GRUB

### 3.1 下载 tarball

有两种方法，一种是 rpm；一种是 tarball。rpm 没有什么好说的，只是 rpm -Uvh 就可以了。不过一般不会启用网络引导功能，所以我们决定采用 tarball 的方式安装。首先到下面的地址下载 tarball：

**<ftp://alpha.gnu.org/gnu/grub>**

上面有 GRUB Legacy 和 GRUB 2，就选择目前 RHEL AS4 用的 0.95 版本做实验。

不过在安装之前先要用 `rpm -q grub` 检查一下 grub 是否安装了。如果是的话，用 `rpm -e grub` 删除，即

使你在安装时指定不安装 `boot loader`，安装程序还是会安装 `grub rpm` 包的。

### 3. 2 安装需求

INSTALL 文件中给出了 build 所需要的软件包条件：

- **GCC**：当前版本的 GCC 应该也可以用，不过还是建议用 2.95 或者以上版本的 GCC
- **GNU Make**：生成 Makefiles 所用
- **GNU binutils 2.9.1.0.23** 或者个更高版本
- 如果你想自己开发 GRUB，还需要 **Texinfo 4.0** 或者更高，同时还有 **GNU Autoconf 2.5x**，**GNU Automake 1.7** 或者更高颁布。

### 3. 3 参数介绍

下面我们介绍几个 configure 过程中可能用到的选项，要完整的选项列表，可以参看 INSTALL 文件和 netboot/README.netboot 文件。（注意选项名不含引号）

- **'--enable-preset-menu=FILE'**：默认情况下 grub.conf 和 stage2 文件是分离的，使用该选项，可以直接把配置文件嵌入到 stage2 中，这样启动时可以立即获得配置文件。对于一些嵌入式的系统来说比较有用，又或者在前面提到的没有显示器/键盘的主机有用，因为在切换显示输出到你的笔记本之前，你是看不到启动的画面的，所以执行切换显示操作也是在看不到输出的情况下进行的（切换显示之后就不会了）。通过这个选项你可以在 <FILE> 中指定启动时立即把显示输出切换到你的笔记本上了。
- **--enable-diskless**：如果使用了该选项，你会得到额外的两个文件，nbgrub 和 pxegrub。接下来你需要指定网卡的驱动程序，netboot/ 目录下有较常见的几种，可以看到几种网卡可以合用一个驱动。常见的有：
  - Intel Etherexpress Pro/100  
**--enable-eeepro100**
  - Novell NE2100 and NE1500  
**--enable-ne2100**
  - Realtek 8139  
SMC 1211  
D-Link DFE530TX+ and DFE538TX  
**--enable-rtl8139**
  - 3Com90x  
**--enable-3c90x**

如果你不想那么麻烦，则按照下面的过程进行：

- **./configure**

- **make**
- **make install**
- 安装结束。默认命令会放在 `/usr/local/bin` 和 `/usrlocal/sbin` 下。**Stage** 文件则放在 `/usr/local/share/grub/i386-pc/` 目录下。

### 3.4 安装之后的操作

- 把 `/usr/local/share/grub/i386-pc/` 目录下的所有文件统统拷贝到 `/boot/grub/` 目录下。
- 接下来是创建 `grub.conf`，并在 `/boot/grub/` 目录下建立一个 `menu.lst`，指向 `grub.conf`。
- 注意，这个 `symbol link` 在 Redhat 9 是必需的，否则启动时会直接进入命令模式，即使已经存在 `grub.conf`；但在 RHEL AS4 下却不是。
- 确认 `/usr/local/sbin` 和 `/usr/local/bin` 在 `PATH` 变量的值中。
- 执行 `grub-install /dev/hda1` 把 `stage1` 写入到 `/boot` 分区的引导扇区。  
也可以执行 `grub` 进入 `grub shell`，然后通过 `root` 和 `setup` 命令来安装 `stage1`。
- 重启机器，安装过程到此结束。

## 四、GRUB 接口

GRUB 除了菜单模式，还提供了一个命令模式和 `grub shell`。当系统启动并显示菜单时，按 `c` 键就可以进入命令模式。提示符是 `grub>`。

如果你已经成功引导操作系统并登录，可以在 `shell` 提示符下执行 `grub`，这会启动一个 `grub shell`。提示符也是 `grub>`。

GRUB 能够象 `bash` 一样使用上下键来访问 `command history`，`ctrl-u`、`ctrl-k`、`ctrl-l` 等快捷键也可以使用。想要编辑菜单，可以按下 `e` 键；想要新增一空白行，可以按下 `o` 键（当前行之后）或者 `O` 键（当前行之前）。想要删除当前行，按下 `d` 键；`ESC` 键放弃当次修改并返回上级菜单。

有些命令只能在命令模式下，有些命令只能用于配置文件（菜单模式），有些命令在两个模式下都可以使用。下面我们就介绍 GRUB 常用的命令。

## 五、GRUB 常用命令

### 5.1 只能用于配置文件的命令

下面这 5 个命令只能出现于配置文件中，分别是：

- **default**：定义默认引导的操作系统。0 表示第一个操作系统，1 表示第 2 个，依此类推
- **fallback**：如果 **default** 所指定的操作系统引导失败，则默认引导由 **fallback** 指定的操作系统

- **hiddenmenu** : 用于启动时隐藏菜单, 除非在 **timeout** 之前按下 **ESC** 才能看到菜单。
- **timeout** : 定义多少秒内如果用户没有按下键盘上的某个按键, 就自动引导 **default** 所指定的操作系统。
- **title** : 定义引导项目的名称。
- **password** : 用于定义进入 **GRUB** 命令模式的密码。你还可以为每个操作系统都定义一个密码, 方法是把 **password** 命令放在 **title** 行之后。而且每个操作系统的引导密码可以不同, 也可以和进入命令模式的那个密码不同, 最大程度的保证 **GRUB** 的安全。

## 5.2 只能用于命令模式或者 **grub shell** 的命令

- **device** : 用于定义 BIOS 中的磁盘设备和操作系统的磁盘设备之间的映射关系。  
例如 **device (hd0) /dev/hdb** 把 BIOS 中的第一个硬盘相当于 Linux 的 **/dev/hdb**
- **serial** : 用于初始化一个串口设备。  
例如 : **serial --unit=0 --speed=9600** 表示初始化 COM1 (**--unit=0**), 速率为 9600 Baud  
不过 **grub shell** 下执行该命令会显示 “无法识别的命令”
- **terminal** : 用于切换显示输出。有两这个值可选 : **console** 和 **serial** 。当你把笔记本接到一个没有显示器/键盘的主机上时, 执行 **terminal serial** 就可以在你的笔记本上操作这台主机的 **GRUB** 了。执行 **terminal console** 把显示输出还原到该主机。  
不过 **grub shell** 下执行该命令会显示 “无法识别的命令”
- **splashimage** : 用于定义启动时的背景图片。例如 **splashimage /grub/splash.xpm.gz**  
经过测试, 发现用 **tarball** 安装的不支持该命令, 这似乎是 **redhat grub rpm** 专有的。
- **blocklist** : 用于显示某个文件在文件系统中的位置, 以 **block list** 的形式显示。例如 **blocklist /vmlinuz**
- **cat** : 用于显示文件内容。由于 **GRUB** 可以直接访问多种文件系统, 所以可以用 **cat (hd0,1)/etc/passwd** 来在 **GRUB** 命令模式下查看 **/etc/passwd** 文件
- **root** : 用于指定含有 **stage** 文件的分区。如果有 **/boot** 分区, 则 **root device** 就是 **/boot** 分区, 否则就是 **/** 分区。
- **rootnoverify** : 和 **root** 类似, 但不会尝试挂载该分区。例如用于指定 **windows** 操作系统所在的分区。
- **chainloader** : 用于加载另外一个 **boot loader** , 通常是用于加载 **windows** 的 **boot loader** 。它的参数是一个 **block list** , 例如 **chainloader (hd0,0)0+1** 表示加载第一个磁盘的第一个分区的第 1 块, 0 是开始位置 (**block** 从 0 开始编号), +1 表示总共读取多少个 **block**。所以 **chainloader 2+0** 表示读取第 3 个 **block** (编号为 2)
- **find** : 可以用于在所有分区上搜索文件。注意, 是所有分区, 而不是某个分区。例如 **find /vmlinuz** 将会把所有分区上查找位于该分区的顶级目录下的 **vmlinuz** 文件, 如果找到就打印该分区名, 例如 **(hd0,1)**。
- **geometry** : 用于显示或者设置某个硬盘的 **CHS** 参数, 同时还显示分区信息。例如 **geometry (hd0)**
- **halt** : 关机

- **reboot** : 重启机器
- **kernel** : 指定操作系统内核文件, 还可以在内核文件名后加上参数。例如  
  
`kernel /vmlinuz ro root=LABEL=/  
  
表示以只读的方式挂载 / 分区, 且根分区设备是 label 为 / 的那个设备; 也可以直接给出设备名。`
- **initrd** : 用于指定 RAM Disk 文件, 例如 `initrd /initrd-2.xx.img` 。
- **boot** : 引导 kernel 所指定的内核, 对于 linux 来说, 必须先加载了 kernel 和 initrd 才能执行 boot 。
- **setup** : 用于安装 stage1 到磁盘的 MBR 或者某个分区的引导扇区。

## 六、保护 GRUB 设置

下面是一个例子

```
password --md5 <密码 1>  
  
timeout=30  
  
default=0  
  
title "redhat AS 4"  
  
password --md5 <密码 2>  
  
root (hd0,1)  
  
kernel /vmlinuz ro root=LABEL=/  
  
initrd /initrd-xxx.img  
  
title "windows"  
  
password --md5 <密码 3>  
  
rootnoverify (hd0,0)  
  
chainloader +1
```

当出现 GRUB 菜单时, 必须按下 **p** 并输入 **<密码 1>** 才能进入命令模式, 如果选择 **"redhat AS 4"** 并按下回车, 会出现提示信息, 要求输入密码, 这时应该输入 **<密码 2>**。同样只有输入 **<密码 3>** 才能进入 windows 系统。

**--md5** 告诉 GRUB 后面的内容是已经用 md5 算法加密过的了。所以 GRUB 会将你输入的口令用 MD5 加密后再与之比较。你可以命令模式下用 **md5crypt** 命令或者在 shell 下执行 **grub-md5-crypt** 来得出一个字符串用 MD5 加密后的结果, 再插入到配置文件中。

为每个引导项目设置 GRUB 密码有好处也有坏处 :

好处是 : 你可以禁止用户随便引导某个操作系统, 即使普通用户可以接触 GRUB 菜单, 没有密码也没有用;

坏处是 : 如果你一旦重启机器, 就必须手工输入口令, 否则会一直停在那里。如果你是在半夜远程重启某个主机的话.... Hoho

**建议** : 只设置 GRUB 密码, 也就是限制用户进入命令模式, 具体菜单项不设置密码。

## 七、GRUB 和 stage 文件

GRUB 含有几个 images 文件，两个基础（必需的）stages 文件（stage1 和 stage2），可选的 stage（也称为 stage1.5），和两个网络引导的 images 文件（nxgrub 和 pxegrub）。stage1 是用于引导 GRUB 的一个必须的映像文件。通常它是被嵌入到 MBR。或者一个分区的引导扇区之中。因为 PC 的引导扇区是 512 字节，所以 stage1 也是 512 字节。stage1 的作用就是从一个本地磁盘加载 stage 2 或者 stage 1.5。因为大小的限制，stage1 对 stage2 或者 stage 1.5 的位置进行编码，也就是说，stage1 是无法识别文件系统格式的。stage2 是 GRUB 的核心映像。它几乎处理所有的事情（除了加载它自己），通常可以把他放在文件系统上，但不是必须的。

一旦你执行了 grub-install 安装了 boot loader，stage1 文件就不是必须的，可以把它移动到任意位置，因为它已经被嵌入到 MBR 或者 PBR 了。

stage1 和 stage2 文件一般位于 /boot/grub/ 目录下，在这个目录下还有很多 stage 1.5 的文件，而且都是以文件系统格式命名的。它们的目的是在 stage1 和 stage2 之间搭建一个桥梁，也就是 stage 1 加载 stage 1.5, stage 1.5 加载 stage2。stage1 和 stage 1.5 的不同之处是 stage1 无法识别文件系统，stage 1.5 可以。因为 Stage2 太大了，无法被嵌入到某个固定的区域，而 Stage1.5 可以安装在 MBR 之后的位置。

至于 nxgrub 和 pxegrub 我们在第 10 节才介绍。

## 八、GRUB 常见错误信息

我们把出错信息分成 3 类，stage1 阶段的、stage 1.5 阶段的、stage2 阶段。后两个可以合在一起。

### 8.1 Stage1 阶段可能出现的错误：

GRUB 对于该阶段出现的错误一般是打印一个出错信息并挂起机器，除非按 CTRL-ALT-DEL 重启。

- Hard Disk Error：表示无法判读 stage 1.5 文件所在的磁盘的大小和 CHS 参数
- Floppy Error：表示无法 stage 1.5 文件所在的软盘的大小和 CHS 参数
- Read Error：表示无法读取 stage 1.5 文件或者 stage2 文件
- Geom Error：stage2 或者 stage 1.5 位于 BIOS 的 read 调用可以访问的范围之外。

### 8.2 Stage 1.5/2 阶段可能出现的错误：

一般会打印一个 “Error <N>” 的错误信息，并根据错误的严重程度看是否继续进行。

=====

1 : Filename must be either an absolute filename or blocklist

解释：1 号错误表示文件名格式错误。在 GRUB 中要么是以绝对路径给出文件

例子：

*grub> kernel vmlinuz root=label=*



*Error 1: Filename must be either an absolute pathname or blocklist  
grub>*

=====

## 2 : Bad file or directory type

**解释** : 2 号错误表示命令期望的是一个普通文件，但相应文件名的对象是一个符号链接、目录、FIFO

**例子** :

```
grub> kernel /testdir root=LABEL=/  
Error 2: Bad file or directory type  
grub>
```

=====

## 3 : Bad or corrupt data while decompressing file

**解释** : 3 号错误表示解压文件时发生错误。可能是因为这个文件被损坏了

=====

## 4 : Bad or incompatible header in compressed file

**解释** : 4 号错误表示压缩文件的头部格式不被兼容或者错误

=====

## 5 : Partition table invalid or corrupt

**解释** : 5 号错误表示分区表无效或者被破坏。这是一个不好的预兆

=====

## 6 : Mismatched or corrupt version of stage1/stage2

**解释** : 6 号错误表示 install 命令发现 stage1 和 stage2 的颁布号不被兼容

=====

## 7 : Loading below 1MB is not supported

**解释** : This error is returned if the lowest address in a kernel is below the 1MB boundary. The Linux zImage format is a special case and can be handled since it has a fixed loading address and maximum size

=====

## 8 : Kernel must be loaded before booting

**解释** : 8 号错误表示执行 boot 命令之前没有先执行 kernel 命令

=====

## 9 : Unknown boot failure

**解释** : 9 号错误表示未知的引导错误

=====

## 10 : Unsupported Multiboot features requested

**解释** : 10 号错误表示请求 Multiboot header 所要求功能不被 GRUB 所支持。

=====

## 11 : Unrecognized device string

**解释** : 11 号错误表示无法识别的设备字符串。

**例子** :

```
grub> root hd0
```

*Error 11: Unrecognized device string*  
*grub>*

=====

#### 12 : Invalid device requested

解释：12 号错误表示请求的设备无效

例子：

```
grub> root (hd2)  
Error 21: Selected disk does not exist  
grub> kernel /grub/grub.conf root=LABEL=/'  
Error 12: Invalid device requested  
grub>
```

=====

#### 13 : Invalid or unsupported executable format

解释：13 号错误表示无效或者无法识别的可执行格式

例子：

```
grub> kernel /grub/grub.conf root=LABEL=/'  
Error 13: Invalid or unsupported executable format  
grub>
```

=====

#### 14 : Filesystem compatibility error, cannot read whole file

解释：14 号错误表示文件系统兼容性错误，无法读取整个文件

=====

#### 15 : File not found

解释：请求的文件无法找到

例子：

```
grub> find /grub-noexist/grub.conf  
Error 15: File not found  
grub>
```

=====

#### 16 : Inconsistent filesystem structure

解释：16 号错误表示不一致的文件系统结构。可能是文件系统结构被破坏了。

=====

#### 17 : Cannot mount selected partition

解释：17 号错误表示无法挂载指定分区。例如 swap 分区

例子：

```
grub> root (hd0,2)    这是一个 swap 分区  
Filesystem type unknown, partition type 0x82  
grub> kernel /vmlinuz  
Error 17: Cannot mount selected partition  
grub>
```

=====

**18 : Selected cylinder exceeds maximum supported by BIOS**

**解释** : 18 号错误表示选择的柱面超过了 BIOS 支持的最大能力。这通常发生在不支持 LBA 模式的硬盘上。

=====

**19 : Linux kernel must be loaded before initrd**

**解释** : 19 号错误表示执行 `initrd` 命令前必须先执行 `kernel` 命令

=====

**20 : Multiboot kernel must be loaded before modules**

**解释** : 20 号错误表示执行 `module` 或者 `moduleunzip` 命令前必须先执行 `kernel` 命令

=====

**21 : Selected disk does not exist**

**解释** : 21 号错误表示选择的磁盘不存在

**例子** :

```
grub> root (hd2)
Error 21: Selected disk does not exist
grub>
```

=====

**22 : No such partition**

**解释** : 22 号错误表示分区不存在

**例子** :

```
grub> root (hd0,10)
Error 22: No such partition
grub>
```

=====

**23 : Error while parsing number**

**解释** : 23 号错误表示参数解释错误，希望是一个数值，但参数却是其他类型

**例子** :

```
grub> root (hda,0)
Error 23: Error while parsing number
grub>
```

=====

**24 : Attempt to access block outside partition**

**解释** : 24 号错误表示尝试访问的 `block` 超出了分区

=====

**25 : Disk read error**

**解释** : 25 号错误表示磁盘读错误

=====

26 : Too many symbolic links

解释 : 26 号错误表示太多的符号连接 (默认最多允许 5 个)

=====

27 : Unrecognized command

解释 : 无法识别的命令

=====

28 : Selected item cannot fit into memory

解释 : 选择的对象无法被加载到内存中。

例子 :

```
[root@monitor boot]# dd if=/dev/zero of=vmlinuz-2.4.20-31.9 bs=1024 count=1 seek=1
```

读入了 1+0 个块

输出了 1+0 个块

```
[root@monitor boot]# grub
```

```
grub> kernel /vmlinuz-2.4.20-31.9 root=label=/  
[Linux-bzImage, setup=0x1400, size=0xffff200]
```

*Error 28: Selected item cannot fit into memory*

```
grub>
```

=====

29 : Disk write error

解释 : 磁盘写错误

=====

30 : Invalid argument

解释 : 无效参数

例子 :

```
grub> serial --noarg=0
```

*Error 30: Invalid argument*

```
grub>
```

=====

31 : File is not sector aligned

解释 : This error may occur only when you access a ReiserFS partition by block-lists (e.g. the command `install'). In this case, you should mount the partition with the `-o notail' option.

=====

32 : Must be authenticated

解释：要求输入口令才能继续进行下面的操作。例如配置文件中有 password 或者 lock 命令

例子：

```
password root1234
title DOS
lock
rootnoverify (hd0,0)
chainloader +1
```

33 : Serial device not configured

解释：33 号错误表示串口还没有配置。这一般发生在你执行 terminal serial 的时候

34 : No spare sectors on the disk

解释：磁盘自由空间不足。可能发生在把 stage 1.5 嵌入到 MBR 之后的空间的时候。但这部分空间可能已经被分区表使用了

## 九、制作 GRUB 引导软盘

GRUB 不仅可以安装在硬盘上，也可以安装在软盘。有两种方式可用于创建 GRUB 引导软盘，区别在于后者带有文件系统，而前者只有 boot loader 本身。如果你误删除那些 stage 文件，后一种方式就大大有用了。

第一种：用 dd 命令创建 GRUB 软盘

- 按照 ./configure , make ,make install 的步骤安装 GRUB
- **dd if=/usr/local/share/grub/i386-pc/stage1 of=/dev/fd0**
- **dd if=/usr/local/share/grub/i386-pc/stage2 of=/dev/fd0 seek=1**
- 这样就创建好一张 GRUB 引导软盘了。

第二种：用 grub-install 创建带文件系统的 GRUB 引导软盘

- 同样是先安装 GRUB
- **mke2fs /dev/fd0**
- **mount /dev/fd0 /mnt/floppy**
- **grub-install --root-directory=/mnt/floppy '(fd0)'**
- 这样软盘上就会有一个 /boot/grub/ 目录，下面就是那些 stage 文件了。
- 上面的两种方法做出来的软盘都只能进入 GRUB 命令模式，你可能会觉得不爽，是否可以直接进入菜单呢？**答案是可以的，只需要多一个步骤：**
- 把 /boot/grub/ 目录下的 device.map ， grub.conf ， menu.lst 统统拷贝到 /mnt/floppy/boot/grub 目录下，这样就可以了。因为之前 GRUB 在软盘上找不到配置文件，

所以自动进入命令模式，现在用它启动机器就可以直接进入菜单模式了。

## 十、使用 GRUB 通过网络引导操作系统

GRUB 不仅可以从本地硬盘引导操作系统，还可以通过网络引导操作系统内核。

假使你由于某种原因，无法启动操作系统。这时你需要在别的空闲的机器上先 build 一份 GRUB，并安装到一张软盘上。

不过由于要支持网络功能，所以 ./configure 时有所不同了：你必须指定启用网络功能，和网卡芯片的型号：（具体参考 netboot/README.netboot 文件）。由于测试机器的网卡芯片是 RTL-8139，所以加上 --enable-rtl8139，整个过程就是（在别的空闲的机器上进行的）：

```
# rpm -e grub （删除原来的 GRUB rpm，如果没有装就不用了）
# ./configure --enable-diskless --enable-rtl8139 （这是故障主机的网卡型号）
# make
# make install
# cp -fa /usr/local/share/grub/i386-pc/* /boot/grub/
# grub-install /dev/hda
# mke2fs /dev/fd0 （格式化软盘，建立 ext2 文件系统）
# mount /dev/fd0 /mnt/floppy
# grub-install --root-directory=/mnt/floppy '(fd0)'
# 检查 /mnt/floppy 下是否有 nbgrub 和 pxegrub 两个文件，这两个文件就是用于网路启动的。如果没有，就从 /boot/grub/ 目录拷贝到 /mnt/floppy/boot/grub/ 目录下。切记切记！！
```

这样你就制作好了一张带有 stage 文件，支持网络引导，且带有网卡驱动的 GRUB 引导盘了。

接下来就是 TFTP 服务器方面的工作了：

- > 1、启动 TFTP 服务器（编辑 /etc/xinetd.d/tftp 文件，把 disable 改为 no）。
- > 2、检查防火墙配置，确保故障主机可以顺利访问该 TFTP 服务器
- > 3、找出一台和故障机器具有相同内核的主机，把 /boot/vmlinuz-2.xx 和 /boot/initrd-2.xx.img 拷贝到 TFTP 服务器的 /tftpboot/boot 目录下
- > 4、把这张软盘插入故障机器的软驱并用它启动机器，就会自动进入 GRUB 命令模式。
- > 5、执行 ifconfig 命令将会显示下面的画面

```
grub> ifconfig
Found Realtek 8139 at 0xc400, ROM address 0x0000
Probing... [Realtek 8139] - ioaddr 0XC400, addr 00:E0:4C:E0:E4:85 100Mbps full-d
uplex
Not initialized yet.

grub>
```

-> 6、下面开始设置 ip 信息了：(ifconfig 命令可以分成多次执行)

```
grub> ifconfig --address=172.17.64.39 --mask=255.255.255.0
Address: 172.17.64.39
Netmask: 255.255.255.0
Server: 0.0.0.0
Gateway: 0.0.0.0

grub> ifconfig --server=172.17.64.34
Address: 172.17.64.39
Netmask: 255.255.255.0
Server: 172.17.64.34
Gateway: 0.0.0.0

grub> ifconfig
Address: 172.17.64.39
Netmask: 255.255.255.0
Server: 172.17.64.34
Gateway: 0.0.0.0
```

-> 7、现在网卡 ip 和 TFTP 服务器的地址都已经设置了。

-> 8、下面设置 root device、kernel、initrd。注意 root 括号中是 nd，切记切记!!!

当你执行 boot 时，内核将被解压并执行，一切就和往常一样。

```
grub> root (nd)
Filesystem type is tftp, using whole disk

grub> kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/
[Linux-bzImage, setup=0x1400, size=0x11098a]

grub> initrd /boot/initrd-2.4.20-8.img
[Linux-initrd @ 0xff9b000, 0x4432f bytes]

grub> boot
```

补充：如果你的网络有 dhcp 服务器，则可以直接执行 dhcp 命令来代替 ifconfig 命令获取 ip 信息

## 十一、把配置文件嵌入到 GRUB 中

这里介绍一种比较少见的情况。有时我们会遇到主机不带显示器/键盘，但由于某些原因需要重启主机并访问 GRUB，这时怎么办？你可能会说用串口连接。不错，通过串口的确可以控制 GRUB，但有一个问题，即使你把笔记本连接到该主机上，但启动时 GRUB 并不会自动使用笔记本的键盘，又如何通过笔记本控制 GRUB 呢？

GRUB 支持一个功能叫做 **preset menu**（预设菜单），工作原理：

- GRUB 检查是否启用了 preset menu 功能，如果是的话，加载 preset menu 文件（不是 grub.conf）
- GRUB 加载 grub.conf，这个过程是一定会执行的，不管有没有 preset menu 存在。
- 在 grub.conf 被加载后，不管 grub.conf 有没有引导项目，只要 preset menu 含有任何引导条目，都会被清掉。只有 grub.conf 文件不可用时，preset menu 的内容才会被保留。

看到这里，我想你应该明白 preset menu 怎么用了。我们可以定义一个文件 pre-menu.conf，内容是把显示

切换到 COM1 上，grub.conf 则是正常的内容。然后把笔记本接到故障主机的 COM1，启动超级终端。再启动故障主机，GRUB 会先把显示切换到 COM1，然后加载 grub.conf，后面的过程就和普通情况没有什么不同。

要使用 preset-menu，你必须在执行 configure 脚本时指定 --preset-menu，例如

```
#./configure --preset-menu=pre-menu.conf
```

```
# make
```

```
# make install
```

pre-menu.conf 文件的内容就是下面两句：

```
serial --unit=0 --speed=9600
```

```
terminal --timeout=0 serial
```

注意，当内核开始执行时，显示也就交回给原来的主机，而不再是你的笔记本了。要解决该问题，

可以在显示菜单是进入命令模式，在 kernel 行后面加上 console=ttyS0，这样 boot 时就可以在超级终端上看到启动的信息了。不过在 login 提示符出现时，显示输出又会回到原来的故障主机上了。这时你可以用 secureCRT 来登录了，不需要再用到超级终端了。

整体效果如下图：（通过串口看的）

```
GNU GRUB  version 0.95  (639K lower / 261056K upper memory)

+-----+
| "Red Hat 9" |
| "winxp"     |
+-----+

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

选择引导 Red Hat 9 项目：

```
Booting "Red Hat 9"
root (hd0,1)
Filesystem type is ext2fs, partition type 0x83
kernel /vmlinuz-2.4.20-8 ro root=LABEL=/
[Linux-bzImage, setup=0x1400, size=0x11098a]
initrd /initrd-2.4.20-8.img
[Linux-initrd @ 0xffbb000, 0x24692 bytes]
=====
```



一旦你不想使用 `preset-menu` 了，是否直接删除 `pre-menu.conf` 就可以呢？

不可以。因为 `pre-menu.conf` 已经被嵌入到 `grub` 中了，这时留在硬盘上的 `pre-menu.conf` 文件其实已经没有用了。要取消 `preset-menu` 功能，必须重新 `configure`，并去掉 `--enable-preset-menu` 选项，再重新安装 `GRUB`。

## 十二、 总结

写了这么多，现在总结一下大概的内容

- `GRUB` 是一种 `boot loader`，它分成两个阶段：`stage1` 和 `stage2`（中间可能有 `stage 1.5`）
- `stage1` 嵌入到 `MBR` 或者 `PBR` 中，`stage2` 留在文件系统中。
- `build GRUB` 三部曲：`configure`、`make`、`make install`、
- `grub-install` 和 `grub shell` 都可以用于安装 `stage1`，但 `grub-install` 功能更加强大
- `GRUB` 支持菜单模式、命令模式。命令模式下的快捷键：`c`、`o`、`O`、`e`、`d`、`ESC`
- 使用 `GRUB` 口令保护 `GRUB` 设置
- `GRUB` 支持从网络引导操作系统，不过在编译时事先指定 `--enable-diskless` 和网卡驱动。可惜的是，`GRUB` 支持的网卡驱动不多。
- `GRUB` 支持通过串口来访问 `GRUB` 的命令和菜单模式
- `GRUB` 可以把配置文件事先嵌入到 `GRUB`，但删除时也需要重装 `GRUB`。

## 十三、 个人建议

当安装好操作系统后，应该做好下面几项工作

- 安装光盘（建议刻成 `ISO` 文件便于保存）
- 制作一张 `bootdisk` 用于引导操作系统
- 制作一张带文件系统的 `GRUB` 引导盘用于引导 `GRUB`，也可以自带配置文件。
- 将 `MBR` 或者 `PBR` 备份成文件。