

grub2基础教程-修订版

smallapple

目录

- [一、grub2新特性](#)
- [二、grub2安装与启动](#)
- [三、grub2配置文件详解](#)
- [四、脚本修改](#)
- [五、使用grub2常见错误及修复方法](#)

一、grub2新特性

1、grub2和grub有很多不同的地方：

- a) 图形接口
- b) 使用了模块机制，通过动态加载需要的模块来扩展功能
- c) 支持脚本语言，例如条件判断，循环、变量和函数
- d) 支持rescue模式，可用于系统无法引导的情况
- e) 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
- f) 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
- g) 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
- h) 支持多种文件系统格式
- i) 可访问已经安装的设备上的数据
- j) 支持自动解压

2、设备的命名

grub2同样以fd表示软盘，hd表示硬盘（包含IDE和SCSI硬盘）。设备是从0开始编号，分区则是从1开始，主分区从1-4，逻辑分区从5开始，而grub分区编号是从0开始的。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1个分区

(hd0,5)/boot/vmlinuz：表示BIOS中的第一个硬盘的第一个逻辑分区下的boot目录下的vmlinuz文件

二、grub2安装与启动

1、安装ubuntu9.10时，grub默认安装在第一硬盘（hd0）的mbr，其实就是把引导文件boot.img写入硬盘的mbr，当然，用户也可以选择不写入硬盘mbr而是写入linux分区的引导扇区。启动时根据mbr所提供信息找到启动分区后，加载分区内的grub核心文件core.img和配置文件grub.cfg，进入选择菜单画面，在菜单画面，按上下箭号选择需要系统菜单项，按Enter进入选项。

GNU GRUB version 1.97~beta1

```
Ubuntu, Linux 2.6.31-9-generic
Ubuntu, Linux 2.6.31-9-generic (recovery mode)
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Microsoft Windows XP Professional (on /dev/sdb1)
```

Use the ↑ and ↓ 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.

按e进入已选择菜单的编辑状态，在编辑状态下，由上下左右箭号来移动光标，enter键换行，ctrl+x以编辑的内容启动，ctrl+c进命令行状态，按Esc退出，回到菜单项。

在需要使用额外的内核参数启动时，比如要加上acpi=off时，把光标移动到linux这一行最后，在splash后加上一空格再输入acpi=off，然后按ctrl+x启动。

```
GNU GRUB version 1.97~beta1
```

```
insmod ext2
set root=(hd0,1)
search --no-floppy --fs-uuid --set f34fc83e-5bca-46ec-98f0-6b70dd264\
fa7
linux /vmlinuz-2.6.31-9-generic root=UUID=50acb630-1895-4016-bfe0-d1\
d1efe1fe57 ro quiet splash
initrd /initrd.img-2.6.31-9-generic
```

```
Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x to boot, Ctrl-c for a command-line
or ESC to return menu.
```

按c进入命令行状态，按Tab键可查看所有可用的命令。

```
GNU GRUB version 1.97~beta1
```

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists possible
device/file completions. ESC at any time exits. ]
```

```
sh:grub> _
```

在命令行状态，可以根据需要加载或移除相应模块，也可用来启动在菜单没有显现的系统。

比如，在第一硬盘的第一分区上装有windows xp系统，但在菜单上没显示出来，我们可以命令行状态下输入命令启动：

```
grub>set root=(hd0,1)
grub>chainloader +1
grub>boot
```

又比如启动第二硬盘第一逻辑分区上的ubuntu系统：

```
grub>set root=(hd1,5)
grub>linux /boot/vmlinuz-xxx-xxx root=/dev/sdb5
grub>initrd /boot/initrd.img-xxx-xxx
grub>boot
```

其中内核vmlinuz和initrd.img的版本号可用按Tab键自动查看。

2、下面介绍下在grub2中一些常用的命令

1)help

查看命令用法，如

help search：查看search命令详细用法

2)set

设置变量值，如

```
set default=0
```

```
set timeout=5
```

```
set root=(hd0,3)等等
```

需要调用变量AA的值时，使用\${AA}，如set root=(hd1,1)，则\${root}=(hd1,1)

3)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

4)timeout

定义在时间内用户没有按下键盘上的某个按键，自动引导default指定的操作系统。

5)root

指定用于启动系统的分区。

6)insmod和rmmod

加载或移除某模块，如

```
insmod jpeg
```

```
insmod png
```

```
insmod ntfs
```

```
rmmod png
```

等

7)drivemap

drivemap兼容grub的map，主要用于只能从第一硬盘(hd0)引导启动的系统如win2000 xp 2003，比如要添加第二硬盘第一分区上的xp系统：

```
menuentry "Windows XP" {
```

```
set root=(hd1,1)
```

```
drivemap -s (hd0) ${root}
```

```
chainloader +1
```

```
}
```

8)ls

列出当前的所有设备。如(hd0)，(hd0,1)，(hd0,5)，(hd1)，(hd1,1)，(hd1,2)等

```
ls -l
```

详细列出当前的所有设备。对于分区，会显示其label及uuid。

```
ls /
```

列出当前设为root的分区下的文件

```
ls (hd1,1)/
```

列出(hd1,1)分区根目录的文件

9)search

```
search -f /ntldr
```

列出根目录里包含ntldr文件的分区，返回为分区号

```
search -l LINUX
```

搜索label是LINUX的分区。

```
search --set -f /ntldr
```

搜索根目录包含ntldr文件的分区并设为root，注意如果多个分区含有ntldr文件，set失去作用。

```
10)loopback
```

loopback命令可用于建立回放设备，如

```
loopback lo0 (hd1,1)/abc.iso
```

可以使用lo0设备来访问abc.iso里的内容，比如说，可以从abc.iso里的软盘映像中启动

```
loopback lo0 (hd1,1)/aa.iso
```

```
linux (lo0)/memdisk
```

```
initrd (lo0)/abc.img
```

要删除某一回放设备，可以使用-d参数：

```
loopback -d lo0
```

```
11)pager
```

分页显示。

```
set pager=1
```

显示满一页时暂停，按space继续

```
set pager=0
```

取消分页显示

```
12)linux
```

用linux命令取代grub中的kernel命令

```
13)chainloader
```

调用另一个启动器，如

```
chainloader (hd0,1)+1
```

调用第一硬盘第一分区引导扇区内的启动器，可以是windows或linux的启动器

```
14) grub2挂载软盘镜像
```

```
menuentry "Boot from DOS IMG" {
```

```
    linux16 /memdisk
```

```
    initrd16 /win98.img
```

```
}
```

对非标准的1.4M和2.8M的其他镜像挂载方法，需要指定 CHS参数：

```
menuentry "Boot from IMG" {
```

```
    linux16 /memdisk c=* h=* s=* floppy
```

```
    initrd16 /xxx.img
```

```
}
```

三、grub2配置文件详解

grub2改用grub.cfg为配置文件，配置文件包含以下基本内容：（红色为解释部分）

```
### BEGIN /etc/grub.d/00_header ###
```

```
load_env
```

```
#加载变量，如果在grubenv保存变量，则启动时装载
```

```
set default="0"
```

```
#设置默认引导项，默认值为0
insmod ext2
#插入文件系统支持的模块，除了用作启动的分区外，其他分区格式可在menuentry底下再添加
set root=(hd0,8)
# 指定系统root分区，也就是 / 分区
search --no-floppy --fs-uuid --set 2d61e5f9-1d2a-4167-a6f1-b991ba00878b
#指定uuid=2d61e5f9-1d2a-4167-a6f1-b991ba00878b的分区为root分区，如果前面的分区号(hd0,8)的#uuid与这里的uuid一致，这两句作用一样，如果不一致，则指定uuid的起作用。
if loadfont /usr/share/grub/unicode.pf2 ; then
#设置终端字体，unicode.pf2支持中文字符显示
set gfxmode=640x480
#设置显示分辨率，默认为640x480，可用800x600，1024x768，建议跟你想设定的图片大小一致
insmod gfxterm
#插入终端模块gfxterm，支持中文字符显示和支持24位图像
insmod vbe
#插入vbe模块，GRUB2引入模块化机制，要使用它，需要在这里加入
if terminal_output gfxterm ; then true ; else
# For backward compatibility with versions of terminal.mod that don't
# understand terminal_output
terminal gfxterm
#设定grub2终端为gfxterm
fi
fi
if [ ${recordfail} = 1 ]; then
set timeout=-1 # 若有启动失败的记录，则菜单项不再倒计时
else
set timeout=10 #倒计时10秒后进按默认启动项启动
fi
#设定默认启动前等待时间，默认为10秒
### END /etc/grub.d/00_header ###

### BEGIN /etc/grub.d/05_debian_theme ###
set menu_color_normal=white/black
#设定菜单字体及背景颜色
set menu_color_highlight=black/blue
#设定选择项字体及背景颜色
#如果使用默认，背景将完全被蓝色挡住了，需要修改blue为black，背景图片才能显示
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/10_linux ###
#10_linux为系统自动添加的当前root分区linux引导项
#每个菜单项要包括menuentry双引号" "和大括号{ }才完整，否则不显示菜单
menuentry "Ubuntu, Linux 2.6.31-10-generic" {
```

```
set quiet=1
insmod ext2
set root=(hd0,8)
search --no-floppy --fs-uuid --set 2d61e5f9-1d2a-4167-a6f1-b991ba00878b
#这句与set root=(hd0,8)效果一样，可删除其一，二者不一致以这句为准
linux /boot/vmlinuz-2.6.31-10-generic
root=UUID=2d61e5f9-1d2a-4167-a6f1-b991ba00878b ro quiet splash
#不喜欢看到一长串的，roo=UUID=***可用root=/dev/sda8(/分区的分区号)代替
initrd /boot/initrd.img-2.6.31-10-generic
}
...
#### END /etc/grub.d/10_linux ####
```

#20_memtest86+为系统自动添加的内存测试菜单项

```
#### BEGIN /etc/grub.d/20_memtest86+ ####
menuentry "Memory test (memtest86+)" {
linux16 /boot/memtest86+.bin
}
menuentry "Memory test (memtest86+, serial console 115200)" {
linux16 /boot/memtest86+.bin console=ttyS0,115200n8
}
#### END /etc/grub.d/20_memtest86+ ####
```

#30_os-prober或30_others为系统自动查找并添加其他系统菜单项，按windows，linux，macos顺序

#查找并添加，支持windows 7识别

```
#### BEGIN /etc/grub.d/30_os-prober ####
#### END /etc/grub.d/30_os-prober ####
#### BEGIN /etc/grub.d/30_others ####
# This entry automatically added by the Debian installer for a non-linux OS
# on /dev/sda1
menuentry "Microsoft Windows XP Professional" {
set root=(hd0,1)
search --no-floppy --fs-uuid --set e852-230b
drivemap -s (hd0) $root
#对以ntldr引导的系统如win2000，xp，win2003，因其引导机制只能从第一硬盘启动，系统会自动添加#映射命令，对vista和win7就没有这句命令
chainloader +1
}
#### END /etc/grub.d/30_others ####
```

#40_custom为自定义的启动项，如启动cdlinux

```
#### BEGIN /etc/grub.d/40_custom ####
menuentry "CDLinux"{
set root=(hd0,8)
linux /CDlinux/bzImage root=/dev/ram0 vga=791 CDL_LANG=zh_CN.UTF-8
initrd /CDlinux/initrd
}
#### END /etc/grub.d/40_custom ####
```

定制个性化的配置文件，可以加入背景图片，使用中文字符，让启动画面独具特色，而不是单调的黑、白、蓝三色。下面是一张经美化启动画面：



四、脚本修改

该怎样定制个性化的配置文件呢？

系统安装完成后，用户就会发现/boot/grub/grub.cfg文件只有root权限可读，如果要直接修改grub.cfg文件，要先修改其权限。好不容易把grub.cfg修改好了，系统内核或grub升级时，会自动执行update-grub，grub.cfg文件就会被打回原形，如何保证修改后的配置文件能一直保留下来呢？

其实不用修改grub.cfg，只要把个性化配置写入/etc/default/目录下的grub和/etc/gurb.d目录下的脚本文件，以后不管升级内核或者是升级grub所执行的update-grub，都会按要求创建个性化的grub.cfg。

1、首先看下 /etc/default/grub，先从应用程序 - 附件里打开终端，输入

```
sudo gedit /etc/default/grub
```

用户密码



看看打开的文件可作什么修改：

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
```

```
GRUB_DEFAULT=0 ----->设置默认启动项，按menuentry顺序。比如要默认从第
四个菜单项启动，数字改为3，若改为 saved，则默认为上次启动项。
GRUB_HIDDEN_TIMEOUT=0 默认为0，单系统时启动菜单自动隐藏，要取消自
动隐藏菜单，改为大于0再sudo update-grub。
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT="3" ----->设置进入默认启动项的等候时间，默认值10
秒，按自己需要修改
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash" ----->添加内核启动参
数，这个为默认
GRUB_CMDLINE_LINUX="noresume" ----->手动添加内核启动参数，比
如 acpi=off noapic等可在这里添加，加 text 参数启动进入字符模式。
```

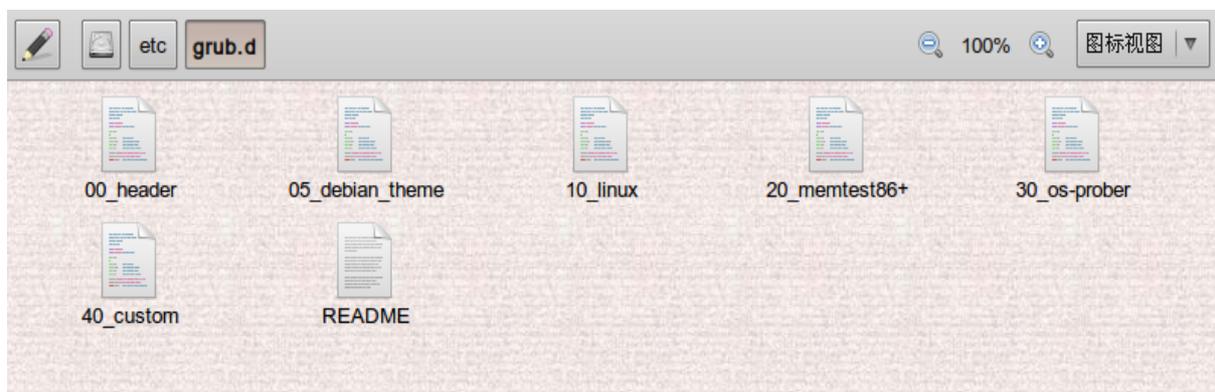
```
# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console ----->设置是否使用图形介面。去除前面#，仅使
用控制台终端，不使用图形介面
```

```
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480 设定图形介面分辨率，如不使用默认，把前面#去掉，
把分辨率改为800x600或1024x768
```

```
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true ----->设置grub命令是否使用UUID，去
掉#，使用root=/dev/sdax而不用root=UUID=xxx
```

```
# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_LINUX_RECOVERY="true" ----->设定是否创建修复模式
菜单项
```

2、再看下/etc/grub.d目录下的脚本文件：



00_header , 05_debian_theme , 10_linux , 20_memtest86+ , 30_os-prober , 40_custom这五个脚本对应grub.cfg上的各个部分，有的版本的grub可能把30_os-prober分出另一部分为 30_os-others，这并不影响。怎样修改这些脚本文件才能保证不会破坏update-grub的运行，又能让生成的grub.cfg合乎意愿呢？其实很简单，只要在脚本文件中找到

```
cat << EOF
*****
*****
*****
EOF
```

这类语句，EOF中间的文本会直接写入grub.cfg中相应位置，所以个性化的语句添加在这地方就可以了。

其他使用变量的语句不要随变做修改，除非你清楚修改后会有什么结果。

1)看看00_header，一样在终端中打开sudo gedit /etc/grub.d/00_header，找到这部分

```
cat << EOF
load_env ->加载由save_env保存在grubenv的变量
set default="${GRUB_DEFAULT}" ----->使用/etc/default/grub中的设定值
if [ \${prev_saved_entry} ]; then
  saved_entry=\${prev_saved_entry}
  save_env saved_entry
  prev_saved_entry= ----->变里设置的变量由下句保存到grubenv
  save_env prev_saved_entry
fi ----->这里回车添加新行后可插入需要添加的模块
insmod jpeg ----->添加背景图片格式支持，如jpeg，png，tga等。
insmod fat ----->添加文件系统格式支持如fat，ntfs，也可在menuentry下添加
EOF
```

再看看最后这一段

```
cat << EOF
if [ \${recordfail} = 1 ]; then
  set timeout=-1 ----->设定为若有启动失败记录，不倒计时，不会进入默认启动，
  需手动选择才能启动选择项，若要设置为即使曾启动错误也进行默认启动，可把时间
  设为
  合适的正值，如 set timeout=5
else
  set timeout=${GRUB_TIMEOUT} ----->使用/etc/default/grub中的设定值
fi ----->这行后添加行
background_image (hd0,7)/boot/images/002.jpg ----->添加背景图片的绝对路
径，可以按05_debian_theme修改图片路径。
EOF
```

2)05_debian_theme修改字体及背景颜色、添加背景图片
先打开文件 sudo gedit /etc/grub.d/05_debian_theme

看这里

```
cat << EOF
set menu_color_normal=white/black ----->设置菜单全局的字体和背景颜色
set menu_color_highlight=black/white ----->设置选择项的字体和背景颜色。
EOF
```

可供使用菜单颜色有：

black(当背景色时为透明), blue, green, cyan, red, magenta, brown, light-gray, dark-gray, light-blue, light-green, light-cyan, light-red, light-magenta, yellow, white

white	blue	magenta	cyan
light-gray	yellow	red	light-magenta
dark-gray	green	light-blue	light-red
black	light-cyan	light-green	brown

添加背景图片，看这一段

```
source /usr/lib/grub/grub-mkconfig_lib

# this allows desktop-base to override our settings
f=/usr/share/desktop-base/grub_background.sh
if test -e ${f}; then
  source ${f}
else
  WALLPAPER="/usr/share/images/desktop-base/moreblue-orbit-grub.png"
  COLOR_NORMAL="black/black"
  COLOR_HIGHLIGHT="magenta/black"
fi
```

这里会自动搜索背景文件，把合适的背景图片更名为 moreblue-orbit-grub.png(moreblue-orbit-grub.tga或moreblue-orbit-grub.jpeg)，再放在/usr/share/images/desktop-base/目录下，然后执行 sudo update-grub，会出现如下的

```
Generating grub.cfg ...
Found background image: moreblue-orbit-grub.png
.....
```

再重启，漂亮的背景图片就会出现。

3)10_linux这部分是自动搜索当前系统，建立当前系统的启动菜单，包括系统头，内核等信息，不要随便修改，要使用个性菜单名，比如中文菜单，可适当添加：先打开文件 sudo gedit /etc/grub.d/10_linux，找到

```
cat << EOF
menuentry "$1" { ----->双引号内可添加个性化字符，$1变量也可改为定值，如
```

```
改为menuentry "启动 Ubuntu Karmic" {
    recordfail=1
    save_env recordfail    ----->如不要保存启动失败记录的，这两句可删除
EOF
```

```
if [ "x$3" = "xquiet" ]; then
    cat << EOF
    set quiet=1
EOF ->启动画面不用splash，要用跳动字符的，这段可删除
```

个性化菜单项也可在这里修改，即上面的menuentry "\$1"不变，修改下面的数值：

```
linux_entry "${OS}, Linux ${version}" \    ----->可改为 linux_entry "启动 Ubuntu,
Linux ${version}" \
    "${GRUB_CMDLINE_LINUX} ${GRUB_CMDLINE_EXTRA}
${GRUB_CMDLINE_LINUX_DEFAULT}" \
    quiet
if [ "x${GRUB_DISABLE_LINUX_RECOVERY}" != "xtrue" ]; then
    linux_entry "${OS}, Linux ${version} (recovery mode)" \    ----->可改为
linux_entry "启动 Ubuntu, Linux ${version} (修复模式)" \
    "single ${GRUB_CMDLINE_LINUX}"
fi
```

注意只修改引号内的字符，前面的menuentry，linux_entry和后面的\不要做修改。

4)20_memtest86+自动添加内存测试启动项

打开文件 `sudo gedit /etc/grub.d/20_memtest86+`，找到

```
cat << EOF
menuentry "Memory test (memtest86+)" {    ----->双引号内可添加个性字符如
menuentry "启动 内存测试"
    linux16  $MEMTESTPATH
}
menuentry "Memory test (memtest86+, serial console 115200)" {
    linux16  $MEMTESTPATH console=ttyS0,115200n8
}
EOF
```

上面有两内存测试项，可删除其中一个。

5)30_os-prober查找其他分区中存在的系统并建立菜单项，依次为windows，linux，macos。

对于单系统，默认会隐藏菜单，可在这里去掉隐藏

先打开文件 `sudo gedit /etc/grub.d/30_os-prober`

单ubuntu系统会自动隐藏菜单，要取消隐藏菜单，可把这部份的三个timeout=0改为和 /etc/default/grub中的timeout一致，比如timeout=10。

```
cat <<EOF
if [ \${timeout} != -1 ]; then
    if keystatus; then
        if keystatus --shift; then
```

```

        set timeout=-1
    else
        set timeout=0
    fi
else
    if sleep$verbose --interruptible 3 ; then
        set timeout=0
    fi
fi
fi
EOF
    else
        cat << EOF
if [ \${timeout} != -1 ]; then
    if sleep$verbose --interruptible ${GRUB_HIDDEN_TIMEOUT} ; then
        set timeout=0
    fi
fi
EOF

```

下面这段自动查找并添加windows、linux和macos启动项，可在menuentry内修改个性字符

windows启动项

```
for OS in ${OSPROBED} ; do
```

```
.....
```

```
if [ -z "${LONGNAME}" ] ; then
```

```
    LONGNAME="${LABEL}"
```

```
fi
```

```
.....
```

```
    cat << EOF
```

```
menuentry "${LONGNAME} (on ${DEVICE})" { ----->比如改为menuentry "启动
```

```
Windows XP" {
```

```
EOF
```

linux启动项

```
linux)
```

```
    LINUXPROBED="linux-boot-prober ${DEVICE} 2> /dev/null | tr ' ' '^' | paste -s -d
, ""
```

```
.....
```

```
    if [ -z "${LLABEL}" ] ; then
```

```
        LLABEL="${LONGNAME}"
```

```
    fi
```

```
    cat << EOF
```

```
menuentry "${LLABEL} (on ${DEVICE})" { ----->比如改为menuentry "启动
```

```
Mandriva" {
```

```
EOF
```

macos启动项

```

macosx)
    OSXUUID="" grub-probe --target=fs_uuid --device ${DEVICE} 2> /dev/null`"
    cat << EOF
menuentry "${LONGNAME} (on ${DEVICE})" { ----->比如改为menuentry "启动
MacOS" {
EOF

```

6)40_custom 自定义启动项，按菜单标准格式在后面添加即可，如

```

#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.

menuentry "启动 Veket" {
    set root=(hd0,8)
    linux /veket/vmlinuz root=/dev/ram0 PMEDIA=hd
    initrd /veket/initrd.gz
}
menuentry "启动 CDLinux" {
    set root=(hd0,8)
    linux /CDlinux/bzImage root=/dev/ram0 vga=791 CDL_LANG=zh_CN.UTF-8
    initrd /CDlinux/initrd
}

```

3、把各项脚本修改保存后，在终端执行

```
sudo update-grub
```

再执行

```
sudo gedit /boot/grub/grub.cfg
```

看看配置文件是不是和你想要的一样了。

如果有不够满意的地方，重新修改下脚本文件，再执行sudo update-grub

4、ubuntu和windows双系统，怎样让系统默认为从windows启动

在 /etc/grub.d 目录中的脚本文件的文件名都是以数字开头，这确定了在执行update-grub时各文件内容被执行的顺序，我们只要把 30_os-prober 这个文件名的数字30改为05到10之间的数字即可（没多少可选06、07、08、09），比如改为08_os-prober，这样创建出来的grub.cfg内的菜单项，windows的排序就会自动在ubuntu之前。

五、使用grub2常见错误及修复方法

1、双系统，重装windows引起没有ubuntu启动项

使用安装版的windows重装windows时会改写mbr，造成grub丢失，可以用grub4dos引导进入ubuntu后修复grub或用livecd启动后修复grub。

1) 用grub4dos修复ubuntu

先下载最新版的grub4dos，下载地址<http://nufans.net/grub4dos/>，如果是xp系统，把grub4dos压缩包内的grldr复制到C盘根目录下，修改boot.ini，在最后加上一行
[c:\grldr="grub4dos"](#)，

```
boot - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
[boot loader]
timeout=6
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional"
/noexecute=optin /fastdetect
C:\CMDCONS\BOOTSECT.DAT="Microsoft Windows XP Recovery Console" /cmdcons
c:\grldr="grub4dos"
```

对于vista/win7系统，把压缩包内的grldr.mbr和grldr复制到C盘根目录下，在C盘自己建立一个boot.ini文件，若有boot隐藏分区的，先给boot分区分配盘符，再把grldr，grldr.mbr和boot.ini放在boot分区下，boot.ini内容如下：

Win7使用软改激活的请不要使用此方法，否则会造成win7不能启动。

```
boot - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
[boot loader]
timeout=0
default=c:\grldr.mbr
[operating systems]
C:\grldr.mbr="Grub4Dos"
```

然后在根目录新建一个menu.lst，内容为

```
timeout 0
default 0

title grub2
find --set-root /boot/grub/core.img
kernel /boot/grub/core.img
boot
(有/boot分区的改为
find --set-root /grub/core.img
kernel /grub/core.img)
```

重启后选择Grub4Dos会自动转入grub2，从grub2菜单项选择ubuntu启动后，在终端执行

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

将grub装入第一硬盘的mbr，如要装入第二硬盘的mbr把sda改为sdb，第三硬盘为sdc，类推。

至于要把grub装到sda或sdb或sdc的mbr上，需根据自己情况确定，看看要设置哪一块硬盘为启动盘，就把grub装到那块硬盘的mbr上。

2)用livecd修复grub

用ubuntu9.10的livecd启动后，打开终端

假如你的ubuntu的 / 分区是sda7，又假如 /boot分区是 sda6，用livecd启动，在终端下输入

```
sudo -i
mount /dev/sda7 /mnt
mount /dev/sda6 /mnt/boot ( 如果没 /boot 单独分区这步跳过 )
grub-install --root-directory=/mnt /dev/sda
```

和前面一样，要装入第二硬盘的把sda改为sdb。

如果grub.cfg已丢失，或grub.cfg出现错误，需要重建的继续执行下面操作：

```
mount --bind /proc /mnt/proc
mount --bind /dev /mnt/dev
mount --bind /sys /mnt/sys
chroot /mnt update-grub
umount /mnt/sys
umount /mnt/dev
umount /mnt/proc
exit
```

3) 用带grub启动的光盘或U盘修复

如果手上有grub启动的工具盘，用工具盘启动，在grub菜单上按c进入命令行状态，在grub>提示符下输入

```
grub>find /boot/grub/core.img (有/boot分区的用find /grub/core.img)
(hdx,y) (显示查找到的分区号 )
grub>root (hdx,y)
grub>kernel /boot/grub/core.img (/boot分区的用 kernel /grub/core.img)
grub>boot
```

执行boot后能转入grub2菜单，启动ubuntu后，再在ubuntu终端下执行

```
sudo grub-install /dev/sda (或sdb，sdc等) 修复grub。
```

如果ubuntu的启动分区使用ext4格式，要有支持ext4格式的grub才能修复。

4) 没安装引导器的 grub4dos引导ubuntu的方法

按第一步的方法使用 boot.ini 建立grub4dos引导项，自己编写 menu.lst，内容如下

```
timeout 3
default 0
```

```
title Ubuntu 9.10
```

```
root (hdx,y) -----> (hd0x,y) 为 /boot分区的分区号
```

```
kernel /vmlinuz-xxx-generic root=/dev/sdxy ----->/dev/sdxy 为 / 分区的分区号
```

```
initrd /initrd.img-xxx-generic
```

```
没 /boot 分区的用
timeout 3
default 0
```

```
title Ubuntu 9.10
root (hdx,y)
kernel /boot/vmlinuz-xxx-generic root=/dev/sdxy
initrd /boot/initrd.img-xxx-generic
```

2、安装时grub没装在mbr上而选择装在linux分区的pbr上，怎么引导ubuntu

1) 按前面的方法一，以grub4dos引导转入grub2。

2) 把ubuntu分区/boot/grub目录下的boot.img提取出来，放到C盘根目录下，然后修改boot.ini，在最后添加一行 [C:\boot.img="grub2"](#)

对vista/7可在C盘新建一个boot.ini，内容为

```
[boot loader]
timeout=0
default=c:\boot.img
[operating systems]
c:\boot.img="grub2"
```

重启后选择grub2就可进入grub2菜单。

此法兼容性较差，成功率低，仅供测试。

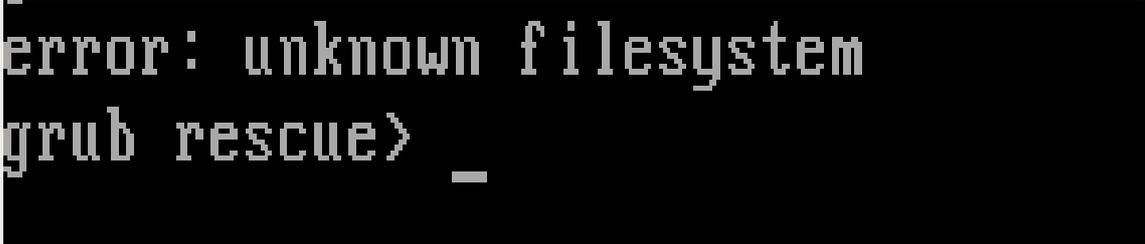
3、由于分区调整引起分区号或分区UUID改变造成的grub2不能正常启动

已安装好系统，对硬盘再次进行分区调整时可能会改变现有分区的分区号发生变化，或者某种原因改变启动分区的UUID，都会造成grub2不能正常启动，而启动进入修复模式 (grub rescue)，这时就要对grub进行修复。

1) 用ubuntu9.10的livecd光盘启动，进入试用桌面系统后，再修复grub，操作方法如第一种情况的第二种方法—

2) 在rescue模式下启动并修复

由于分区问题，启动时会自动进入rescue模式，只要grub核心文件还在分区内，就能由rescue模式转到normal模式，进而通过命令进入系统。下面这个就是rescue介面：



```
error: unknown filesystem
grub rescue> _
```

由于在rescue模式下，只有少量的基本命令可用，必须通过一定的操作才能加载正常模块，然后进入正常模式。

rescue模式下可使用的命令有：set，ls，insmod，root，prefix(设置启动路径)

先假设grub2的核心文件在(hd0,8)分区，再来看看怎样从rescue模式进入从(hd0,8)启动的正常模式(normal)。

在rescue模式下search命令不能用，对不清楚grub2文件处于哪个分区的，可以用ls命令查看，比如

ls (hd0,8)/ 查看(hd0,8)分区根目录，看看有没有boot文件夹

ls (hd0,8)/boot/ 查看(hd0,8)分区的/boot目录下文件

ls (hd0,8)/boot/grub/ 查看(hd0,8)分区/boot/grub目录下文件

通过文件查看，可以确定grub2核心文件处于哪个分区，接下来就可以进行从rescue到normal的转变动作：

先ls看看分区，根据分区列表，猜下 / 分区的编号再ls (hd0,x)/ 看分区目录下文件确定找到 / 分区，不对的话继续找。找到 / 分区的 (hd0,x) 继续

```
grub rescue>root=(hd0,x)
```

```
grub rescue>prefix=/boot/grub
```

```
grub rescue>set root=(hd0,x)
```

```
grub rescue>set prefix=(hd0,x)/boot/grub
```

```
grub rescue>insmod normal
```

```
rescue>normal ----->若出现启动菜单，按c进入命令行模式
```

```
rescue>linux /boot/vmlinuz-xxx-xxx root=/dev/sdax
```

```
rescue>initrd /boot/initrd.img-xxx-xxx
```

```
rescue>boot
```

内核版本号 -xxx-xxx可以按Tab键查看后再手动补全。

有 /boot分区的，要先找出 /boot 分区 (hd0,x)，再找出 / 分区的 (hd0,y)，同样用ls (hd0,x)/ 和ls (hd0,y)/ 的方式确定分区

```
grub rescue>root=(hd0,x)
```

```
grub rescue>prefix=/grub
```

```
grub rescue>set root=(hd0,x)
```

```
grub rescue>set prefix=(hd0,x)/grub
```

```
grub rescue>insmod normal
```

```
rescue>normal ----->若出现启动菜单，按c进入命令行模式
```

```
rescue>linux /vmlinuz-xxx-xxx root=/dev/sday
```

```
rescue>initrd /initrd.img-xxx-xxx
```

```
rescue>boot
```

说明：

1) 由于grub2版本的的不一致，有的可能在第9步insmod normal.mod加载正常模块后直接进入normal模式，即出现了normal grub>的提示符，这种情况就不能执行第10步，即可以跳过normal命令的输入。

2) 虽然输入normal命令会出现菜单，但由于缺少加载内核的Linux命令，直接从菜单不能进入系统，需要按c在命令行继续操作。

3) 使用/boot单独分区的，要正确修改路径，如

```
prefix=(hd0,8)/grub
```

```
insmod /grub/normal.mod
```

另外root=/dev/sda8也要修改根分区的分区号。

4) 按boot启动系统后，再在系统下打开终端，执行命令修复grub重建配置文件grub.cfg

```
sudo update-grub
重建grub到第一硬盘mbr
sudo grub-install /dev/sda
```

4、双硬盘双系统 Grub Loading时间过长的解决方案

grub2的boot.img设定root的uuid从第一分区开始搜索分区的/boot/grub下的模块并加载，如果linux分区处于第二硬盘甚至第三硬盘，会导致搜索时间过长而，出现菜单时间会长达10多秒。

对双（多）硬盘的情况建议把grub安装在ubuntu所在硬盘的mbr上，/boot分区或 / 分区 尽量靠前，并设该硬盘为启动盘，会大大缩短启动时间。

5、重复一下，单linux系统或硬盘安装时iso放在C盘，umount /isodevice引起的误认为单系统不能出现菜单项的几种处理方法。

1. 开机自检后时按几下shift键，可调出菜单项

2. sudo update-grub

重建grub.cfg，会发现新的系统而改写grub.cfg，一般能出现菜单项。

3.如第二种方法不能解决，直接修改grub.cfg

把在#### BEGIN /etc/grub.d/30_os-prober 中的这一段

```
if keystatus; then
```

```
if keystatus --shift; then
```

```
set timeout=-1
```

```
else
```

```
set timeout=0
```

```
fi
```

```
else
```

```
if sleep$verbose --interruptible 3 ; then
```

```
set timeout=0
```

```
fi
```

```
fi
```

整段删除或修改三处set timeout=<大于0>，再执行

```
sudo update-grub
```

6、安装 ubuntu 9.10 后出现 Error : No such device: xxx-xxx-xxx，不能启动的修复办法（未经实机测试）

由于grub2兼容性问题，少部分电脑安装完ubuntu9.10重启时会出现

```
Error : no such device:3c7c1d30-86c7-4ea3-ac16-30d6b0371b02
Failed to boot default entries.
```

Press any key to continue.

原因是电脑不支持 uuid的搜索，soier 的修复过程证实了这个问题，见

<http://forum.ubuntu.org.cn/viewtopic.php?f=139&t=238346>

原贴使用的是 live DVD进入修复系统模式，的shell 下change root 修复，livecd 没有修复系统模式，现的根据他的方法写个 livecd（能支持ext4读写其他版本的live

Linux一可以）下的修复办法，希望有这个问题的朋友测试下看行不行（我电脑没这个

问题，没办法测试)：

1)、livecd 启动进入试用系统

2)、挂载 / 分区，比如 / 分区为 /dev/sda7

```
sudo mount /dev/sda7 /mnt
```

如果有 /boot 单独分区，则挂载 /boot 分区

3)、修改 grub.cfg

```
sudo chmod +w /mnt/boot/grub/grub.cfg
```

```
sudo chmod +w /mnt/grub/grub.cfg (/boot 单独分区的)
```

```
sudo gedit /mnt/boot/grub/grub.cfg
```

```
sudo gedit /mnt/grub/grub.cfg (/boot 单独分区的)
```

找到 grub.cfg 可所有的这句

```
search --no-floppy --fs-uuid --set 3c7c1d30-86c7-4ea3-ac16-30d6b0371b02
```

的前面加上 # 号注释掉

```
# search --no-floppy --fs-uuid --set 3c7c1d30-86c7-4ea3-ac16-30d6b0371b02
```

这句的grub.cfg中有好几处，后面的uuid不尽相同，要全部找到并注释掉

4)、重启试试能否进入系统，可这进系统再进行下个步

5)、sudo gedit /usr/lib/grub/grub-mkconfig_lib

找到173-175行

```
if fs_uuid=""${grub_probe} --device ${device} --target=fs_uuid 2> /dev/null"; then
```

```
echo "search --no-floppy --fs-uuid --set ${fs_uuid}"
```

```
fi
```

前面全部加#注释掉，记住从 if 开始到 fi 结束，以防止语法错误

```
# if fs_uuid=""${grub_probe} --device ${device} --target=fs_uuid 2> /dev/null"; then
```

```
# echo "search --no-floppy --fs-uuid --set ${fs_uuid}"
```

```
# fi
```

然后

```
sudo update-grub
```

这样重建出来的 grub.cfg 就没有

```
search --no-floppy --fs-uuid --set 3c7c1d30-86c7-4ea3-ac16-30d6b0371b02
```

的语句

7、如何从grub2回到grub (wubi使用者勿试)

```
sudo apt-get purge grub-pc ----->清除grub2
```

```
sudo rm -fr /boot/grub/*
```

```
sudo apt-get install grub ----->安装 grub
```

```
sudo grub-install /dev/sda ----->安装grub到第一硬盘mbr，第二硬盘为/dev/sdb，
```

若安装到分区引导扇区则为分区号如 /dev/sda7等

```
sudo update-grub ----->重建menu.lst
```

8、重装windows后出现 error: no such device

由于重装windows时会改变windows引导分区的uuid，原来的配置文件grub.cfg中的 search --no-floppy --fs-uuid --set xxx-xxx 在搜索分区时会按原来的uuid查找分区，找不到相应的uuid，就出现 error: no such device，在ubuntu下执行

`sudo update-grub`

重建grub.cfg，就能解决。此方法也适用于用convert命令把fat改为ntfs格式时出现的error。

参考论坛资料: <http://bbs.znpc.net/forumdisplay.php?fid=57&page=1>