

## 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：

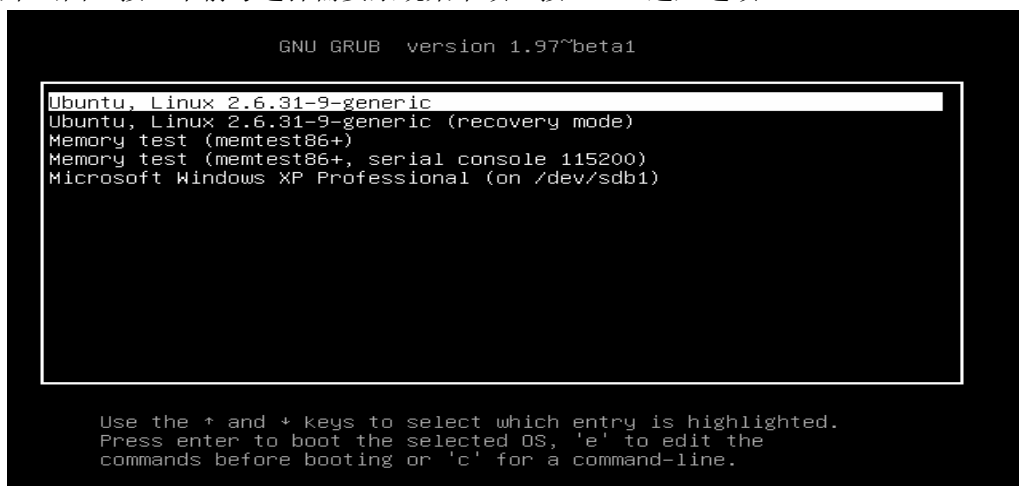
(fd0)：表示整个软盘

(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，按上下箭头选择需要系统菜单项，按 Enter 进入选项。



按 e 进入已选择菜单的编辑状态，在编辑状态下，由上下左右箭头来移动光标，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, 注意如果多外分区含有 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

### 三、grub2 配置文件详解

grub2 改用 grub.cfg 为配置文件，配置文件应包含以下内容：

```
### 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
```

```
if [ ${prev_saved_entry} ]; then
```

```
    saved_entry=${prev_saved_entry}
```

```
    save_env saved_entry
```

```
    prev_saved_entry=
```

```
    save_env prev_saved_entry
```

```
fi
```

```
# 可以用 save_env pre_saved_entry 把变量值存入 grubenv，也可用 save_env default 把 default 值存入 grubenv，供 load_env 调用。
```

```
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
```

```
set timeout=10
```

```
#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。展示一张启动画面：



#### 四、脚本修改

该怎样定制个性化的配置文件呢？

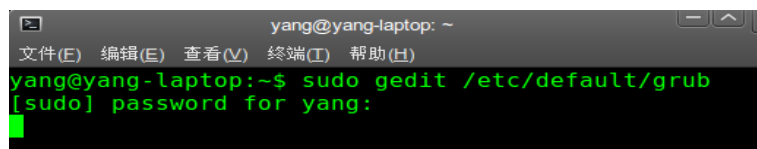
系统安装完成后，用户就会发现/boot/grub/grub.cfg 文件只有 root 权限可读，如果要直接修改 grub.cfg 文件，要先修改其权限。好不容易把 grub.cfg 修改好了，系统内核或 grub 升级时，会自动执行 update-grub，grub.cfg 文件就会被打回原形，如何保证修改后的配置文件能一直保留下来呢？

其实不用修改 grub.cfg，只要把个性化配置写入/etc/default/grub 和/etc/grub.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。

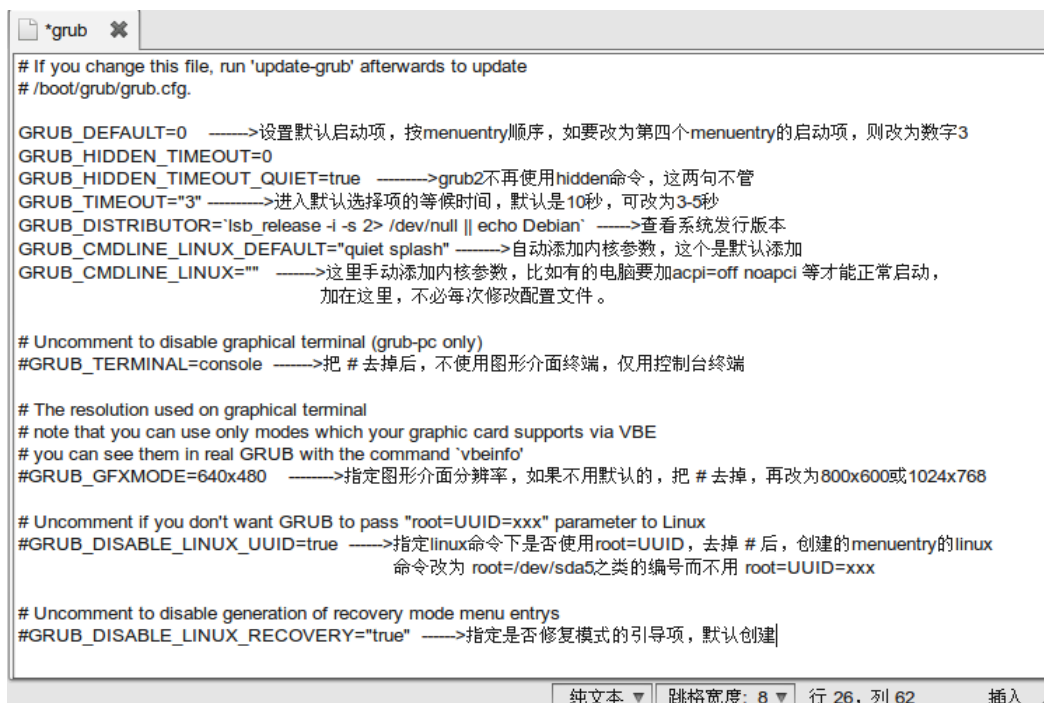
1、首先看下 /etc/default/grub，先从应用程序—附件里打开终端，输入

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

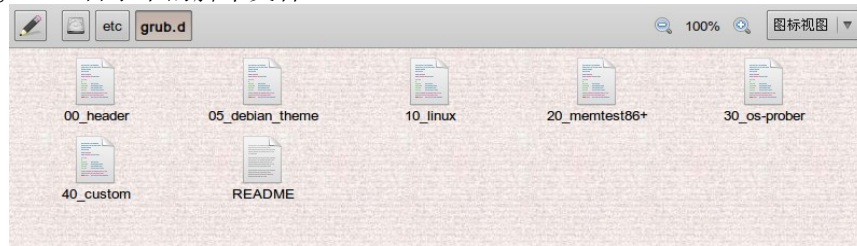
用户密码



看看打开的文件可作什么修改：



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便做修改，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39行—48行这块

```
39 cat << EOF
40 load_env
41 set default="${GRUB_DEFAULT}" ——>这时读取/etc/default/grub中的设定值，不要改|
42 if [ \${prev_saved_entry} ]; then
43   saved_entry=\${prev_saved_entry}
44   save_env saved_entry
45   prev_saved_entry=
46   save_env prev_saved_entry
47 fi ——>这里回车
48 insmod jpeg ——>在新一行下可插入需要的模块，如添加jpeg图片格式支持
49 insmod png ——>添加png格式图片支持
50 insmod ntfs ——>添加ntfs格式文件系统支持等等
51 EOF
```

再看看最后这一段

```
114 cat << EOF
115 set timeout=${GRUB_TIMEOUT} ——>这里读取/etc/default/grub的值，不用做修改
116 background_image (hd0,7)/boot/images/002.jpg ——>这里可加入要使用的背景图片的绝对路径|
117 EOF
```

2)05\_debian\_theme 修改字体及背景颜色

先打开文件 sudo gedit /etc/grub.d/05\_debian\_theme

看第7-10行

```
7 cat << EOF
8 set menu_color_normal=white/black ——>菜单字体和背景颜色
9 set menu_color_highlight=black/blue ——> 选择项字体和背景颜色，若用默认，背景会被蓝色挡住，把blue改为black，背景图片才能显示|
10 EOF
```

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

看58-60行

```
58 cat << EOF
59 menuentry "$1" { ——>在引号内"和$1之间可加入字符，$1变量也可改为定值如" 启动 $1"或"启动 Ubuntu Karmic"
60 EOF
```

或第111-116行

```

111 linux_entry "${OS}, Linux ${version}" \ ——> 引号内改为"启动 ${OS}, Linux ${version}"
112 "${GRUB_CMDLINE_LINUX} ${GRUB_CMDLINE_EXTRA} ${GRUB_CMDLINE_LINUX_DEFAULT}" \
113 quiet
114 if [ "x${GRUB_DISABLE_LINUX_RECOVERY}" != "xtrue" ]; then
115 linux_entry "${OS}, Linux ${version} (Recovery Mode)" \ ——>引号内可改为"启动 ${OS}, Linux ${version} (修复模式)"
116 "single ${GRUB_CMDLINE_LINUX}"

```

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+`

```

13 cat << EOF
14 menuentry "启动 Memory test (memtest86+)" { ——>引号内添加个性字符
15     linux16 $MEMTESTPATH
16 }
17 这里还有一个内存测试启动项，可删除其中一个。
18 EOF

```

#### 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`

找到 37-59 行

```

37 cat << EOF
38 if [ \${timeout} != -1 ]; then
39     if keystatus; then
40         if keystatus --shift; then
41             set timeout=-1
42         else
43             set timeout=0 ——>把0改为同/etc/default/grub中的数字，比如3
44         fi
45     else
46         if sleep$verbose --interruptible 3 ; then
47             set timeout=0 ——>把0改为同/etc/default/grub中的数字，比如3
48         fi
49     fi
50 fi
51 EOF
52 else
53     cat << EOF
54 if [ \${timeout} != -1 ]; then
55     if sleep$verbose --interruptible ${GRUB_HIDDEN_TIMEOUT} ; then
56         set timeout=0 ——>把0改为同/etc/default/grub中的数字，比如3
57     fi
58 fi
59 EOF

```

如果要添加个性字符，往下找到

```

cat << EOF
menuentry "${LONGNAME} (on ${DEVICE})" {
EOF

```

的地方，在引号中根据需要添加或修改，如改为 "启动 \${LONGNAME}"，"启动 Linux"，"启动 MacOS" 等等。

#### 6) 40\_custom 自定义启动项，按菜单标准格式在后面添加即可，如图



```
40_custom
1 #!/bin/sh
2 exec tail -n +3 $0
3 # This file provides an easy way to add custom menu entries.  Simply type the
4 # menu entries you want to add after this comment.  Be careful not to change
5 # the 'exec tail' line above.
6
7 menuentry "启动 PCLinuxOS 2009" {
8     set root=(hd0,3)
9     chainloader +1
10 }
11 menuentry "启动 Veket" {
12     set root=(hd0,8)
13     linux /veket/vmlinuz root=/dev/ram0 PMEDIA=hd
14     initrd /veket/initrd.gz
15 }
16 menuentry "启动 CDlinux" {
17     set root=(hd0,8)
18     linux /CDlinux/bzImage root=/dev/ram0 vga=791 CDL_LANG=zh_CN.UTF-8
19     initrd /CDlinux/initrd
20 }
21 menuentry "启动 Grub4Dos" {
22     set root=(hd0,8)
23     linux /grub.exe
24 }
25
```

3、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。

4、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

## 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：

```
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.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，其他硬盘类推。

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，需要重建的还要执行

```
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。

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。

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：

grub rescue>set	->1、查看grub当前的启动分区和路径
grub rescue>prefix=(hd0,8)/grub	->2、确定预启动路径
grub rescue>root=hd0,8	->3、确定启动分区
grub rescue>set root=hd0,8	->4、设定grub启动分区
grub rescue>set prefix=(hd0,8)/boot/grub	->5、设定grub启动路径
grub rescue>set	->6、查看grub设定内容，是否和实际的分区和路径一致
grub rescue>root=hd0,8	->7、确定启动分区
grub rescue>prefix=(hd0,8)/boot/grub	->8、确定启动路径
grub rescue>insmod /boot/grub/normal.mod	->9、加载正常模块
grub rescue>normal	->10、进入正常模式，出现菜单
normal grub>insmod /boot/grub/linux.mod	->11、在出现菜单后按c，在命令行下加载模块
normal grub>set root=hd0,8	->12、确定正常启动分区
normal grub>linux /boot/vmlinuz-*** root=/dev/sda8	->加载内核并确定 / 分区
normal grub>initrd /boot/initrd.img-****	->13、加载initrd.img
normal grub>boot	

1) 分区号 (hd0, 8) 在单独使用时可以不加小括弧，在使用路径是必须加小括弧，如第 2 和第 3 步。

2) 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。

3) 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。

4) 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。

5) 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

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