

Grub 2

Материал из Викиучебника — открытых книг для открытого мира

GRUB (GRand Unified Bootloader) — загрузчик ОС, применяется для загрузки ядер Linux, OpenSolaris, BSD и Hurd (StartUp Manager (<http://web.archive.org/web/20070312180553/http://web.telia.com/~u88005282/sum/index.html>)).

GRUB 2 — это новая версия GRUB, сильно отличающаяся от предыдущей. Данное описание и инструкции касаются только таблицы разделов MBR и BIOS. Для GPT и UEFI используются немного другие команды.

Содержание

Определяем версию загрузчика

GRUB2 - первые шаги.

/etc/default/grub

Что делать, если файла /etc/default/grub нет

Изменение стандартного пункта загрузки

Пример

Изменение времени отображения меню

Изменение времени отображения в debian 8

"Скрытое" меню

Изменение стандартных параметров загрузки ядра

Пример

Остальные настройки файла /etc/default/grub

/etc/grub.d

Пример

Применение изменений

Нюанс с нумерацией дисков и разделов

Резервная копия

Оформление

Установка фонового изображения

Способ №2:

Способ для дистрибутива debian 8

Настройка цвета пунктов меню

Пример

Настройка цвета пунктов меню в debian 8

Изменение цвета меню навсегда

Восстановление GRUB2

Восстановление GRUB2 с LiveCD

Восстановление GRUB2 с LiveCD. способ 2 (без chroot)

Восстановление GRUB2 с LiveCD. способ 3 (совсем простой)

Копирование рабочего GRUB 2 в нужный раздел^[1]

Загрузка системы при частичной неработоспособности GRUB2

Совет

Загрузка Windows аналогичным образом

Advanced

Создание аварийно-спасательного GRUB2-liveUSB

Консоль grub2 - команды и хитрости

ls

cat

linux

initrd

load_env

chainloader

boot

root

save_env

set

search

lsfonts

help

halt

reboot

background_image

vbeinfo

configfile

terminal_output.console

terminal_output

play

tr

testspeed

time

sleep

Премудрости ;)

Загрузка другого grub2 из текущего

Создание дополнительных меню загрузки

Пример файла "custom.cfg"

MBR — бэкап, восстановление и загрузка через образ

Создание LiveCD с GRUB2

Пример файла "grub.cfg"

Создание LiveCD с GRUB2 на базе имеющегося дистрибутива

Добавление в меню пункта загрузки с CD/DVD & USB

Способ 1. Загрузка с CD & USB посредством Plop Boot Manager

Способ 2. Загрузка с CD посредством Smart Boot Manager

Деактивация recordfail grub2

Ручной сброс значения recordfail

См. также

Примечания

Ссылки

Определяем версию загрузчика

Для начала стоит определить версию загрузчика. Для этого можно использовать команды "grub-install --version" или "grub-terminfo --version".

GRUB2 - первые шаги.

В GRUB2 основным файлом конфигурации является не "/boot/grub/menu.lst", как это было в предыдущей версии, а "/boot/grub/grub.cfg". Однако редактировать его напрямую не нужно, ведь при каждом обновлении файл генерируется автоматически с помощью файла настроек /etc/default/grub и скриптов директории /etc/grub.d.

/etc/default/grub

Данный файл содержит в себе основные настройки для grub2. Через него, собственно, они и изменяются. Для наглядности ниже приводится примерное содержимое этого файла:

```
GRUB_DEFAULT=6
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT="2"
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null` || echo Debian
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# 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

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_LINUX_RECOVERY="true"
```

Мы видим, что файл представляет собой набор опций в человекопонятном формате ОПЦИЯ=ЗНАЧЕНИЕ. Мы не будем здесь рассматривать все представленные параметры, поскольку рядовому пользователю вполне достаточно понимания пары-тройки основных параметров. Наиболее часто встречающаяся потребность при настройке grub — изменение стандартного пункта загрузки и/или времени показа меню. Рассмотрим же, как это делается.

Что делать, если файла /etc/default/grub нет

Начинающие пользователи внимательно читают этот замечательный учебник и пытаются следовать описанным здесь сведениям. Однако тут же натываются на грабли: файла `/etc/default/grub` нет. 1) У вас версия Grub1 2) Проверить, есть ли файл `/etc/sysconfig/grub2`. Править его аналогичным образом. В этом разделе будет написано, почему этого файла нет и как настраивать загрузчик в этом случае.

Изменение стандартного пункта загрузки

По умолчанию стандартный пункт (выделенный при показе меню) — верхний в списке. После установки Ubuntu она окажется наверху, а Windows, например, будет последним пунктом. После обновления ядра первым пунктом списка становится загрузка с новым ядром. Это происходит потому, что именно первый пункт загрузки является стандартным по умолчанию. За это, собственно, отвечает параметр `"GRUB_DEFAULT"`. Значением его является номер пункта в меню загрузки, который должен быть выбран стандартным, причем нумерация начинается с нуля. Значение по умолчанию - 0, поэтому и выбирается первый пункт. Для того, чтобы выбрать другой пункт, нам нужно узнать, каким по счету он будет в списке. Тут есть два варианта: просмотреть содержимое `"/boot/grub/grub.cfg"` и сосчитать, какой по счету окажется нужная запись, или же перезагрузиться и более наглядно посмотреть то же самое в меню загрузки (перед этим побегайте стрелочками по меню, чтобы остановить таймер). Второй вариант отличается более наглядным представлением, что проще для неопытного пользователя. В обоих случаях не забывайте о порядке нумерации — 0, 1, 2, 3 и так далее, то есть пятому сверху пункту будет соответствовать значение 4, второму — 1, первому — 0. В вышеприведенном примере установлено значение 6, то есть стандартным задан седьмой пункт меню. Кроме задания конкретного пункта есть еще другие интересные варианты. Например, если в качестве значения указать `"saved"` (без кавычек) и добавить строку `GRUB_SAVEDEFAULT=true` - тогда при загрузке будет выбран тот пункт, который был загружен в прошлый раз. Или же можно указать точное название пункта. В данном случае оно должно быть именно таким, каким мы его видим в `"/boot/grub/grub.cfg"`. При этом значение должно указываться в кавычках. Данный способ удобен тем, что после обновления ядра не придется изменять настройки из-за сбившейся нумерации.

Пример

Если в `"grub.cfg"` пункт меню выглядит так:

```
menuentry "Ubuntu, Linux 2.6.32-020632rc6-generic" {  
    recordfail=1  
    ....  
}
```

то и значение параметра `"GRUB_DEFAULT"` должно указываться именно в виде

```
GRUB_DEFAULT="Ubuntu, Linux 2.6.32-020632rc6-generic"
```

Изменение времени отображения меню

По умолчанию меню загрузки отображается 10 секунд, после чего загружается стандартный пункт (если раньше не нажат `enter` и не выбран другой пункт, что останавливает таймер). С изменением задержки все совсем просто. За эту задержку отвечает параметр `"GRUB_TIMEOUT"` (не путайте с `"GRUB_HIDDEN_TIMEOUT"`!). Значение задается в секундах. Обратите внимание,

что цифра указывается в кавычках. В нашем примере это значение — 2, то есть меню отображается две секунды. Кроме того, есть одна хитрость. Если поставить значение "-1", то меню будет отображаться до тех пор, пока пользователь не выберет какой-либо пункт. То есть без всяких таймеров и утекающих секунд.

Изменение времени отображения в debian 8

Данный хак будет полезен и для владельцев систем построенных на debian. Дело в том что файл /etc/grub.d/00_header построен по непонятному принципу, а строчка в GRUB_TIMEOUT в /etc/default/grub не меняет время отображения. Для того чтобы это исправить надо открыть любым редактором файл /etc/grub.d/00_header и удалить данные строчки (делаем резервную копию на всякий случай):

```
if [ "$quick_boot" = 1 ]; then
    cat <<EOF
function recordfail {
    set recordfail=1
EOF
    FS="$(grub-probe --target=fs "${grubdir}")"
    case "$FS" in
        btrfs | cpiofs | newc | odc | romfs | squash4 | tarfs | zfs)
            cat <<EOF
            # GRUB lacks write support for $FS, so recordfail support is disabled.
EOF
        ;;
        *)
            cat <<EOF
            if [ -n "\${have_grubenv}" ]; then if [ -z "\${boot_once}" ]; then save_env recordfail; fi;
            fi
EOF
        esac
    cat <<EOF
}
EOF
fi
```

Эта функция запутанная и ничего не даёт. Далее ищем и удаляем полностью данный текст:

```
make_timeout ()
{
    cat << EOF
if [ "\${recordfail}" = 1 ] ; then
    set timeout=${GRUB_RECORDFAIL_TIMEOUT:-1}
else
EOF
    if [ "x${3}" != "x" ] ; then
        timeout="${2}"
        style="${3}"
    elif [ "x${1}" != "x" ] && \
        ([ "$quick_boot" = 1 ] || [ "x${1}" != "x0" ]) ; then
        # Handle the deprecated GRUB_HIDDEN_TIMEOUT scheme.
        timeout="${1}"
        if [ "x${2}" != "x0" ] ; then
            grub_warn "$(gettext "Setting GRUB_TIMEOUT to a non-zero value when
GRUB_HIDDEN_TIMEOUT is set is no longer supported.")"
        fi
        if [ "x${GRUB_HIDDEN_TIMEOUT_QUIET}" = "xtrue" ] ; then
            style="hidden"
            verbose=
        else
            style="countdown"
            verbose=" --verbose"
        fi
    else
        style="countdown"
        verbose=" --verbose"
    fi
    # No hidden timeout, so treat as GRUB_TIMEOUT_STYLE=menu
```

```

        timeout="${2}"
        style="menu"
    fi
    cat << EOF
if [ x${feature_timeout_style} = xy ] ; then
    set timeout_style=${style}
    set timeout=${timeout}
EOF
    if [ "x${style}" = "xmenu" ] ; then
        cat << EOF
        # Fallback normal timeout code in case the timeout_style feature is
        # unavailable.
        else
            set timeout=${timeout}
        EOF
    else
        cat << EOF
        # Fallback hidden-timeout code in case the timeout_style feature is
        # unavailable.
        elif sleep${verbose} --interruptible ${timeout} ; then
            set timeout=0
        EOF
    fi
    cat << EOF
fi
fi
EOF
}

```

Далее удаляем полностью из файла следующий текст:

```

if [ "x${GRUB_BUTTON_CMOS_ADDRESS}" != "x" ] ; then
    cat <<EOF
if cmostest $GRUB_BUTTON_CMOS_ADDRESS ; then
EOF
make_timeout "${GRUB_HIDDEN_TIMEOUT_BUTTON}" "${GRUB_TIMEOUT_BUTTON}"
"${GRUB_TIMEOUT_STYLE_BUTTON}"
echo else
make_timeout "${GRUB_HIDDEN_TIMEOUT}" "${GRUB_TIMEOUT}" "${GRUB_TIMEOUT_STYLE}"
echo fi
else
make_timeout "${GRUB_HIDDEN_TIMEOUT}" "${GRUB_TIMEOUT}" "${GRUB_TIMEOUT_STYLE}"
fi

```

Теперь я объясню почему данные строчки нам совсем не нужны. Для того чтобы установить значение timeout нам не нужны сотни строчек кода а достаточно всего одной. Таким образом мы избавились от лишнего мусора в файле. И теперь мы можем запустить `update-grub -o test-grub.cfg` для того чтобы удостовериться в том что ошибок не возникает (будет создан файл `test-grub.cfg` в текущей папке). Теперь для того чтобы установить желаемое значение timeout нам нужно открыть файл `/etc/grub.d/00_header`, и после строчки `set` -е добавить:

```

echo set timeout=15

```

Теперь для того чтобы изменить время отображения меню нам будет достаточно изменить данную строчку и выполнить `update-grub`. Таким же образом можно записывать любые переменные в наш готовый `grub.cfg` (после `echo` можно вводить что угодно, и это будет записано в наш готовый конфиг).

"Скрытое" меню

В случае, если на компьютере установлена только Ubuntu, меню загрузки по умолчанию не будет отображаться, а grub2 будет загружать вас напрямую в систему. Однако же, иногда может возникнуть необходимость загрузиться с другим ядром или же запустить проверку памяти. для этого предусмотрено "скрытое меню". За него отвечает параметр "GRUB_HIDDEN_TIMEOUT". В случае, когда установлены другие ОС, этот параметр закомментирован (# в начале строки). Если в меню загрузки присутствует Windows, этот параметр игнорируется, даже если он раскомментирован, т.е. использовать скрытое меню невозможно. В случае с единственной ОС он будет активен. значение его задает задержку в секундах. Grub2 приостановит загрузку на заданное количество секунд, давая пользователю возможность вызвать меню загрузки, нажав Escape. Если значение установлено в 0, то задержки не будет. Однако, пользователь все равно сможет вызвать отображение меню, удерживая при загрузке shift. Параметр "GRUB_HIDDEN_TIMEOUT_QUIET" отвечает за отображение таймера во время паузы. При значении "true" таймер показан не будет. При выборе "false" - будет отображаться.

Изменение стандартных параметров загрузки ядра

Иногда бывает необходимо загружать ядро системы с какими-либо особыми параметрами. Например, для корректной работы специфического оборудования. в этом случае весьма полезен будет параметр "GRUB_CMDLINE_LINUX_DEFAULT". Он отвечает за те параметры, с которыми запускаются linux-ядра при загрузке. Значение его по умолчанию - "quiet splash", что приводит к показу графической заставки при запуске системы, и отключению отображения какой-либо текстовой информации. Вы можете добавить необходимые вам параметры запуска ядра, приведя это значение к виду "quiet splash your_param1 your_param2", то есть дописав через пробел нужные параметры.

Пример

Вместо того, чтобы добавлять новые параметры, мы уберем стандартные. Отключим графическую заставку и "молчаливый режим". Заменим

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

на

```
GRUB_CMDLINE_LINUX_DEFAULT=""
```

Теперь все наши пункты загрузки linux-ядер будут преобразованы в verbose mode, то есть режим загрузки без графической заставки и с выводом на экран текстовой информации о загрузке компонентов системы, что позволяет следить за процессом загрузки и выявлять неполадки.

Обратите внимание, чтобы ядру не передавались лишние параметры через GRUB_CMDLINE_LINUX, если он не пустой. Все вышеописанное работает для

```
GRUB_CMDLINE_LINUX=""
```

Остальные настройки файла /etc/default/grub

```
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
```

- данная строка отвечает за названия дистрибутива, которое будет отображаться в меню GRUB2 (Ubuntu, Xubuntu, Debian, и т.д.)

```
GRUB_TERMINAL=console
```

- раскомментируйте данную строку, и это позволит отключить графический терминал (grub-rc only). Данный прием будет полезен тем, что отключение графического режима может существенно увеличить скорость работы меню (прокрутка, отзывчивость).

```
GRUB_DISABLE_LINUX_UUID=true
```

- раскомментируйте, если вы не хотите, чтобы GRUB передавал вашей системе параметр "root=UUID=xxx".

```
GRUB_GFXMODE=640x480
```

- этот пункт отвечает за разрешение графического меню GRUB2. Данный параметр предоставляет разрешение, которое поддерживается видеокартой пользователя (например: 640x480, 800x600, 1024x768, 1600x1200 - для соотношения сторон 4:3. И 640x400, 800x500, 1024x640, 1280x800, 1680x1050 - для широкоформатных мониторов с соотношением сторон 16:10). Настройки применимы только для меню загрузки. Так же у нас есть возможность увидеть все доступные разрешения для экрана из меню GRUB2. Для этого, находясь в меню, жмем "с", затем в строке grub> пишем vbeinfo.

Если строка GRUB_GFXMODE=640x480 закомментирована или выбранное разрешение недоступно, тогда GRUB2 использует настройки по умолчанию, определенные скриптом /etc/grub.d/00_header.

Несколько советов:

- Настройка одинакового разрешения в меню GRUB2 и вашей операционной системы, позволит немного уменьшить время загрузки.

- В качестве дополнительной опции пользователь может задать битность изображения, просто добавив необходимые данные в настройку разрешения. Например так: 1280x1024x24 или 640x480x32.

- Пользователь может добавить одновременно несколько разрешений. Как это работает? А вот так, если GRUB2 не может использовать первое заданное разрешение, тогда он будет пробовать следующее и т.д. Перечень разрешений необходимо разделять запятыми. Пример: 1280x1024x16,800x600x24,640x480.

- Если используется строка GRUB_GFXMODE= и при выполнении команды sudo update-grub вам выдается сообщение not found, просто попробуйте или добавить или изменить битность изображения.

```
GRUB_DISABLE_LINUX_RECOVERY=true
```

- для того, чтобы избежать появления в меню режима Recovery, добавьте (если отсутствует) или раскомментируйте данную строку. Если вам нужен режим Recovery только для конкретно взятого ядра - вам будет нужно создать необходимую запись в скрипт /etc/grub.d/40_custom.

```
GRUB_DISABLE_OS_PROBER="true"
```

- Включает/выключает проверку скриптом /etc/grub.d/30_os-prober наличия на других разделах операционных систем, включая такие как Windows, Linux, OSX и Hurd.

/etc/grub.d

Этот каталог содержит в себе скрипты, которые используются при создании "grub.cfg". При обновлении grub2 они находят все установленные на компьютере системы и ядра и формируют в "grub.cfg" меню загрузки, которое мы и видим. Два основных из них - "10_linux" и "30_os-prober" отвечают за поиск linux-ядер и остальных ОС на других разделах соответственно. Файл "40_custom" позволяет добавлять свои пункты загрузки. Это может быть полезно, если вы, например, хотите добавить какие-то особые варианты загрузки системы.

Примечание: файл "40_custom" должен заканчиваться пустой строкой, иначе последний пункт не будет отображаться в меню!

Пример

Добавляем verbose-mode (режим загрузки без графической заставки, с текстовым отображением процесса загрузки), позволяющий контролировать процесс загрузки системы. Для этого мы немного отредактируем обычный пункт загрузки. Допустим, он выглядит так (в "/boot/grub/grub.cfg"):

```
menuentry "Ubuntu, Linux 2.6.32-020632rc6-generic" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi
    set quiet=1
    insmod ext2
    set root=(hd0,5)
    search --no-floppy --fs-uuid --set 0e717c2a-24bd-4abe-acfe-ecf98fc814f8
    linux /boot/vmlinuz-2.6.32-020632rc6-generic root=UUID=0e717c2a-24bd-4abe-acfe-ecf98fc814f8 ro quiet splash
    initrd /boot/initrd.img-2.6.32-020632rc6-generic
}
```

Для того, чтобы сделать из этого verbose-mode, нам нужно убрать опции quiet и splash из строки "linux". Ну, и для понятности изменим название самого пункта. в итоге получаем:

```
echo "adding verbose mode" >&2
menuentry "Ubuntu, Linux 2.6.32.rc6 verbose-mode" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi
```

```

set quiet=1
insmod ext2
set root=(hd0,5)
search --no-floppy --fs-uuid --set 0e717c2a-24bd-4abe-acfe-ecf98fc814f8
linux /boot/vmlinuz-2.6.32-020632rc6-generic root=UUID=0e717c2a-24bd-4abe-acfe-ecf98fc814f8 ro
initrd /boot/initrd.img-2.6.32-020632rc6-generic
}

```

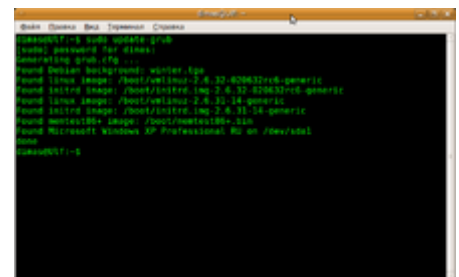
Все это и добавляем в "40_custom" в конец файла. Строка с echo не является обязательной, она лишь будет сигнализировать нам о том, что наш пункт найден и добавлен при обновлении grub2

Применение изменений

После того, как мы отредактировали и сохранили наши файлы, радоваться еще рано. Теперь нужно закончить дело, обновив наш "/boot/grub/grub.cfg". Это довольно просто - нужно всего лишь выполнить команду

```
sudo update-grub
```

Она выполнит скрипты в каталоге "/etc/grub.d" и применит заданные в "/etc/default/grub" параметры. После этого смело перезагружаемся и видим, что все работает как надо. Ну или не работает... Если все прошло успешно, то в выводе консоли вы увидите список найденных grub2 ОС и linux-ядер. Выглядит все это примерно так, как на скриншоте справа.



Тот же самый скрипт выполняется и при обновлении ядра.

Нюанс с нумерацией дисков и разделов

В grub2 имеется еще одно важное отличие от старого grub. Связано оно с нумерацией жестких дисков и их разделов. В grub нумерация физических дисков и нумерация разделов начинались одинаково - с нуля. Первый физический диск (sda в системе) обозначался "hd0", второй (sdb) - "hd1", и так далее. Это же осталось и в grub2. нумерация же разделов диска изменилась. Если в grub первый раздел первого диска (sda1) именовался "hd0,0", четвертый (sda4) - "hd0,3", то теперь, в grub2, цифра раздела соответствует цифре раздела в системе. То есть, sda1 теперь будет "hd0,1" (а не "hd0,0"), sdb4 - "hd1,4". Проще говоря, нумерация дисков идет с нуля, а нумерация разделов - с единицы!

Резервная копия

Перед каким-либо редактированием обязательно сделайте бэкап файлов конфигурации. например, выполнив последовательно эти строки:

```

datev=$(date +%Y_%m_%d)
mkdir -p ~/.grub.bak/$datev
cd ~/.grub.bak/$datev
mkdir -p boot/grub etc/default
cp /boot/grub/grub.cfg boot/grub
cp -Rp /etc/grub.d etc
cp /etc/default/grub etc/default

```

(можно сохранить данный скрипт в /usr/bin, назвать, к примеру, "grub-backup" и дать права 755. Теперь для резервного копирования grub2 достаточно будет выполнить команду "grub-backup")

Немного улучшенная версия скрипта.

```
#!/bin/bash
datev=$(date +%Y_%m_%d)
mkdir -p ~/.grub.bak/$datev
cd ~/.grub.bak/$datev
mkdir -p boot/grub etc/default
cp /boot/grub/grub.cfg boot/grub
cp -Rp /etc/grub.d etc
cp /etc/default/grub etc/default
echo "Резервная копия создана. Папка: \"$HOME\"/.grub.bak/\"$datev"
```

Оформление

Установка фоновое изображения

Вы можете выбрать и установить полноцветное изображение в качестве фона в меню загрузки grub2. Размер изображения должен соответствовать разрешению grub2 (по умолчанию - 640x480, задается в "/etc/default/grub"), формат файла - png или tga. Вы можете установить готовый набор из нескольких подходящих для grub2 изображений, выполнив команду

```
sudo apt-get install grub2-splashimages
```

(не путайте пакет grub2-splashimages с пакетом grub-splashimages, который не совместим с grub2!). Эти изображения будут помещены в каталог "/usr/share/images/grub", который нужно будет добавить в файл конфигурации (см. ниже). Фоновое изображение задается в файле "/etc/grub.d/05_debian_theme". Найдите в нем строку такого вида:

```
for i in {/boot/grub,/usr/share/images/desktop-base}/moreblue-orbit-grub.{png,tga} ; do
```

Те, кто более-менее знаком с bash, узнают прием с перечислением в фигурных скобках. Для остальных: все не так сложно, как может показаться. В первых фигурных скобках (в нашем примере - "{/boot/grub,/usr/share/images/desktop-base}") перечислены каталоги, в которых grub2 будет искать подходящие для установки изображения. Далее (после слеша) указано имя файла изображения (без расширения!), которое мы хотим установить в качестве фона (здесь - "moreblue-orbit-grub"). Вторые фигурные скобки ("{png,tga}") определяют расширения файлов, которые grub2 будет пытаться использовать в качестве фона. Заметьте, что пункты в фигурных скобках перечисляются через запятую и не разделяются пробелами! По умолчанию изображения ищутся в каталогах "/boot/grub" и "/usr/share/images/desktop-base". Изображения из пакета grub2-splashimages у нас распаковались в "/usr/share/images/grub", поэтому мы должны добавить этот каталог в список мест, где grub2 будет искать изображения. Добавим еще один пункт в первые фигурные скобки. Сразу же определимся с выбором картинки. Я выбрал последнюю, переименовав файл для удобства в "winter.tga". Заменяем имя ("moreblue-orbit-grub") на имя выбранной картинки (в моем случае - "winter"). В итоге получим такую строку:

```
for i in {/boot/grub,/usr/share/images/desktop-base,/usr/share/images/grub}/winter.{png,tga}
; do
```

Заметьте, что между именем файла и вторыми фигурными скобками должна стоять точка! Как это работает: grub2 последовательно просматривает заданные каталоги и ищет в них файл с заданным именем и одним из перечисленных расширений. Цикл прекращается как только найдено первое подходящее изображение. Таким образом, файл с подходящим именем присутствует в нескольких каталогах, будет взят файл из того каталога, который указан раньше других в списке. Аналогично, если изображение представлено в разных форматах (например, "image.png" и "image.tga") - выбран будет файл с тем расширением, которое первым указано во вторых фигурных скобках. Интересности: вы можете добавить любой каталог с изображениями, владельцем которой вы являетесь. Это упростит добавление/изменение изображений. Однако, заметьте: при этом root должен иметь как минимум права r-x на этот каталог. Также не стоит добавлять каталоги с шифрованного home-раздела, т.к. на момент загрузки он еще не примонтирован. Лучше просто сделайте себя владельцем каталога "/usr/share/images/grub" и храните все картинки там. Помимо PNG и TGA поддерживается также формат JPEG. Но на текущий момент реализована только поддержка jpeg с 8-битным цветом. Поэтому используйте jpeg-файлы только если вы понимаете, что такое глубина цвета и как сохранить изображение в 8-битном цвете. Для того, чтобы добавить поддержку jpeg-файлов, приведите вторые фигурные скобки к такому виду: "{png,tga,jpg,jpeg}".

Обратите внимание, что в последних, на данный момент (23.10.2010), версиях GRUB, установка фонового изображения упростилась ещё больше. Теперь для установки фонового изображения достаточно в строке:

```
WALLPAPER="/usr/share/images/desktop-base/moreblue-orbit-grub.png"
```

указать в кавычках путь к файлу с одним из перечисленных выше расширений (.png,.tga,.jpg) и разрешением 640x480. После чего обновить конфигурацию:

```
sudo update-grub
```

Способ №2:

В файле /etc/default/grub определить константу GRUB_BACKGROUND и передать ей путь к файлу с картинкой.

Способ для дистрибутива debian 8

В debian 8 картинка устанавливается по другому. За поиск отвечают вот эти строчки в файле /etc/grub.d/05_debian_theme.

```
for background in *.jpg *.JPG *.jpeg *.JPEG *.png *.PNG *.tga *.TGA; do
    if set_background_image "${background}"; then
        exit 0
    fi
done
```

В этих строчках менять ничего не нужно. Сначала будет проверена константа GRUB_BACKGROUND в файле /etc/default/grub, если изображение будет найдено то оно устанавливается на фон grub. Если данная константа в /etc/default/grub не указана, то update-grub будет искать в каталоге /boot/grub файлы формата tga, jpg и png. Поиск выполняется по имени, например если в папке находятся файлы A.tga и B.tga, на фон будет выбрана картинка A.tga. О

картинке которая будет выбрана выведется информация при выполнении команды update-grub: Found background image: [путь]. Если в каталоге /boot/grub нет изображений, то поиск будет производиться оп переменной указанной в файле /usr/share/desktop-base/grub_background.sh, с именем WALLPAPER.

```
WALLPAPER=[путь до изображения]
```

Если по пути указанном в переменной WALLPAPER, файла /usr/share/desktop-base/grub_background.sh не будет найдено изображения, то изображение установится на /usr/share/images/desktop-base/desktop-grub.png. Для того чтобы вместо /usr/share/images/desktop-base/desktop-grub.png задать своё значение, надо найти вот такую строчку с текстом в файле /etc/grub.d/05_debian_theme.

```
if set_background_image "/usr/share/images/desktop-base/desktop-grub.png"; then
```

И изменить на путь до изображения. Порядок поиска:

1. GRUB_BACKGROUND в файле /etc/default/grub
2. Каталог /boot/grub
3. WALLPAPER в файле /usr/share/desktop-base/grub_background.sh
4. Строчка с текстом "/usr/share/images/desktop-base/desktop-grub.png" в файле /etc/grub.d/05_debian_theme

Настройка цвета пунктов меню

Цвета пунктов меню также задаются в файле "/etc/grub.d/05_debian_theme". В самом начале файла нужно всего-навсего изменить значения переменных:

```
COLOR_NORMAL=  
COLOR_HIGHLIGHT=
```

Здесь и настраиваются цвета для пунктов меню загрузки. Переменная "color_normal" отвечает за цвета обычного (невыделенного) пункта меню. "color_highlight" - за цвета выделенного пункта. Вот эти две переменные мы и будем изменять. Значение каждой из этих строк представляет собой два цвета, разделенные слешем. Первый цвет - это цвет текста. Второй цвет - цвет фона строки. Вот список доступных цветов:

black - черный; blue - синий; brown - коричневый; cyan - голубой; dark-gray - темно-серый; green - зеленый; light-cyan - светло-голубой; light-blue - светло-синий; light-green - ярко-зеленый; light-gray - серый; light-magenta - розовый; light-red - светло-красный; magenta - пурпурный; red - красный; white - белый; yellow - желтый.

Заметьте, что *black* (черный) при использовании в качестве второго параметра (т.е. цвета фона) в результате даст прозрачный фон, без какой-либо заливки, текст прямо поверх картинки. Использование *black* в качестве цвета текста дает черный текст.

Пример

```
COLOR_NORMAL=black/black  
COLOR_HIGHLIGHT=black/light-gray
```

В моем случае заданы следующие параметры: обычный пункт - черный текст, прозрачный фон; выделенный пункт - черный текст, серая подсветка строки.

Настройка цвета пунктов меню в debian 8

На debian 8 цвет меню устанавливается таким образом. Так как добавление переменных введённых выше в файл `/etc/grub.d/05_debian_theme` ничего не дают, то изменять надо в уже готовом файле. Выполняем в терминале:

```
update-grub -o grub.cfg
```

В текущем каталоге появится файл `grub.cfg`. Открываем его любым редактором и ищем строчку с:

```
if background_image [путь изображения]; then
```

Сразу под ним вставляем данные строчки:

```
set menu_color_normal=light-green/black  
set menu_color_highlight=light-red/black
```

Затем выполняем команду:

```
mv -v grub.cfg /boot/grub/
```

Перезагружаемся и наслаждаемся цветным меню.

Изменение цвета меню навсегда

Чтобы каждый раз не менять `grub.cfg` заходим в файл `/etc/grub.d/05_debian_theme` и ищем строчку с текстом:

```
echo "else"
```

И удаляем её. Затем ищем строчку с текстом:

```
# Set the traditional Debian blue theme.
```

И прямо под ней две строки отвечающие за цвет. Изменяем их и выполняем команду:

```
update-grub
```

Перезагружаемся и радуемся результату.

Восстановление GRUB2

После, например, установки Windows, вы можете столкнуться с тем, что при загрузке будет отображаться только меню загрузки Windows, а меню grub2 вы не увидите вовсе, что приведет к невозможности загрузить что-либо кроме Windows. Такое происходит из-за того, что Windows при установке затирает загрузочную область жесткого диска (так называемый MBR-раздел), удаляя оттуда код загрузчика grub2. Для восстановления grub2 вам понадобится liveCD Ubuntu 10.04 или любой другой современной системы, включающей grub2 в свой дистрибутив.

Восстановление GRUB2 с LiveCD

Загрузитесь с вашего LiveCD (предварительно выберите меню устройств загрузки при наличии такового или же установите cd-привод первым boot device в BIOS). Запустите консоль. Для начала нам нужно будет узнать, на каком диске и на каком разделе установлена Ubuntu. Если вы не помните этого - воспользуйтесь командой

```
sudo fdisk -l
```

Для работы нам понадобится /-раздел (root). Примонтируйте его командой

```
sudo mount /dev/sda5 /mnt
```

В моем случае корневой раздел системы - /dev/sda5, вы же используйте свое значение. Если у вас /boot сделан отдельным разделом, его также надо будет примонтировать. Я не использую отдельный boot-раздел, но предположим, что sda3 выделен под /boot. Монтируем его:

```
sudo mount /dev/sda3 /mnt/boot
```

Также монтируем каталог /dev нашей live-системы как /dev нашего root-раздела:

```
sudo mount --bind /dev /mnt/dev
```

и /proc live-системы как /proc root-раздела:

```
sudo mount --bind /proc /mnt/proc
```

теперь выполняем следующую команду:

```
sudo chroot /mnt
```

Если вы устанавливаете GRUB для Gentoo:

```
sudo mount sysfs /sys -t sysfs
```

Теперь вы - root-пользователь в системе, корневым разделом которой считается /mnt, то есть корневой раздел вашей настоящей системы. И вот мы готовы обновить MBR-раздел жесткого диска, переустановив grub2. Но для начала нам нужно определиться, с какого физического диска загружается наш компьютер. Если у вас один жесткий диск, разбитый на разделы - он будет именоваться sda. Если несколько - первый из них - sda, второй - sdb, и так далее. Определить загрузочный диск снова поможет команда

```
sudo fdisk -l
```

В полученной таблице есть столбец "загрузочный". В моем случае напротив раздела sda1 стоит звездочка, означающая, что у меня загрузочным является первый жесткий диск, то есть sda. Определившись с загрузочным диском, ставим на него grub2:

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

Если вы столкнетесь с какими-либо ошибками - попробуйте перезапустить команду с ключом --recheck:

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

Также в некоторых случаях может помочь вариант

```
grub-install --recheck --no-floppy /dev/sda
```

Обратите внимание: мы устанавливаем grub2 на физический диск (sda, sdb...) т.е. в mbr, а не на раздел (sda3, sdb1...)! Если все прошло успешно, выходим из chroot командой

```
exit
```

Отмонтируем наши диски и каталоги: /dev нашей live-системы:

```
sudo umount /mnt/dev
```

/proc:

```
sudo umount /mnt/proc
```

boot-раздел, если таковой монтировался отдельно:

```
sudo umount /mnt/boot
```

И собственно, корневой раздел:

```
sudo umount /mnt
```


Все! Перезагружаемся, восстанавливаем порядок загрузки в BIOS (ставим жесткий диск на первое место) и смотрим результат.

Восстановление GRUB2 с LiveCD. способ 2 (без chroot)

Нам опять же понадобится LiveCD Ubuntu 9.10 или другой системы с поддержкой grub2. В отличие от предыдущего способа, мы не будем запускать установщик через chroot, что позволит сократить количество требуемых действий. Итак, загружаемся с выбранного LiveCD, запускаем консоль. Вместо использования chroot мы воспользуемся ключом "--root-directory". Но для начала убедимся, что данная опция поддерживается нашей live-системой. Если вы работаете с Ubuntu 9.10 Desktop, можно не волноваться. В ином случае выполните

```
man grub-install
```

Если в описании опций присутствует вышеупомянутый ключ "--root-directory" - все в порядке. Теперь нам нужно примонтировать корневой раздел системы. Если вы не помните, на каком разделе стоит система, вам поможет

```
sudo fdisk -l
```

или же можно глянуть

```
sudo blkid
```

Разница лишь в форме вывода информации. Определившись с корневым разделом, монтируем его. Пусть в нашем примере это будет sda5:

```
sudo mount /dev/sda5 /mnt
```

Если у вас выделен отдельный boot-раздел, примонтируйте и его. Допустим это sda2

```
sudo mount /dev/sda2 /mnt/boot
```

Теперь переходим собственно к установке grub2. Ключ "--root-directory" позволяет указать, что использовать в качестве корневой директории. Нам нужен корневой раздел нашей системы, который примонтирован в /mnt. Поэтому выполняем:

```
sudo grub-install --root-directory=/mnt /dev/sda
```

Как и в предыдущем способе, grub2 устанавливается на физический диск, а не на раздел! В качестве диска нужно указать тот диск, который установлен загрузочным в bios. Если все пройдет успешно, установщик выведет сообщение об успешном завершении и список обнаруженных разделов, которые были добавлены в "device.map". Если все это есть - отмонтируем диски и перезагружаемся. Если будет выведен неполный список разделов - отредактируйте файл "device.map" в корневом каталоге установленной системы (в описанном примере - "/mnt/boot/grub/device.map") поправляем его, добавив остальные диски и скорректировав нумерацию. Файл должен иметь вид

```
(hd0) /dev/sda
(hd1) /dev/sdb
```

При необходимости повторяем операции для всех жестких дисков. Сохраняем файл и повторно выполняем команду `grub-install`, как описано выше. Теперь должен отобразиться правильный список дисков. Отмонтируем диски и перезагружаемся.

Восстановление GRUB2 с LiveCD. способ 3 (совсем простой)

В случае, если нужно только восстановить MBR диска (например, после (пере)установки Windows), все можно сделать еще проще, вообще ничего не монтируя. Грузимся с LiveCD. Определяем нужный жесткий диск с помощью "`sudo fdisk -l`". Если он единственный - значит, `sda`. Вводим команду

```
sudo grub-install --root-directory=/mnt /dev/sda
```

изменив `/dev/sda` на нужный диск. Можно снова грузиться с этого диска.

Копирование рабочего GRUB 2 в нужный раздел^[1]

Порой бывает необходимость перенести корректно работающий GRUB 2 на нужный раздел (только основной раздел диска). Для этого можно загрузиться с LiveCD или использовать уже установленную систему (причём в том числе и для установки GRUB 2 на текущий раздел, с которого загружена работающая система). Для систем использующих ядро Linux достаточно выполнить в терминале:

```
grub-setup -d ../../boot/grub /dev/sdX
```

от имени root (суперпользователя) например:

```
sudo grub-setup -d ../../boot/grub /dev/sdX
```

Где:

- `/../../boot/grub` — полный путь до файлов GRUB 2, которые необходимо использовать,
- `/dev/sdX` — основной раздел диска в главную загрузочную запись (MBR) которого необходимо установить GRUB 2, а X — буква основного раздела диска, её нужно заменить на необходимую, уточнить которую можно например, выполнив в терминале:

```
sudo fdisk -l
```

После чего выполнить непосредственно инсталляцию GRUB 2 на выбранный ранее раздел командой терминала `update-grub`, например:

```
sudo update-grub
```

При этом программа инсталляции найдет и установит соответствующие пункты загрузки всех совместимых со спецификациями Multiboot операционных систем и, вероятно, загрузчиков других операционных систем (например Windows, OS/2 и т.п.).

Загрузка системы при частичной неработоспособности GRUB2

В некоторых случаях может оказаться так, что отдельные файлы grub2 будут повреждены (например, в результате сбоя при выполнении "update-grub"). Однако, иногда есть шанс загрузиться в свою систему и восстановить grub2 из нее без необходимости полной переустановки оного. Если при загрузке grub2 вы видите сообщения об ошибках - не спешите предаваться панике. Все может быть еще вполне поправимо. На данный момент наша главная цель - попасть в консоль grub2. Вы можете сразу увидеть ее на экране, или же будет предложено что-либо нажать или выбрать для запуска консоли. В любом случае, если вам удалось запустить консоль - радуйтесь, шансы на спасение увеличены! Консоль grub2 похожа на более привычную оболочку bash. Здесь также поддерживается автодополнение команд по нажатию tab. Если же нажать tab в пустой строке - мы увидим полный список доступных команд. Что ж, приступим к процедуре запуска системы. Для начала нам нужно задать root-раздел, то есть раздел, на который установлена наша система. В моем случае это "/dev/sda5" (в grub2 разделы задаются в виде "hd*,*"). Если вы мало знакомы с нумерацией разделов в grub2 - почитайте пункт "нюанс с нумерацией дисков и разделов" в I главе). root-раздел задается командой

```
set root=(hd0,5)
```

В моем случае это "hd0,5", он же "/dev/sda5", то есть пятый раздел первого диска. Полный список доступных разделов можно получить уже знакомой многим командой "ls". Она выдаст что-то вроде

```
(hd0,1) (hd0,2) (hd1,1) (hd2,1)
```

После того, как вы задали root-раздел, можно удостовериться в том, что все получилось, выполнив команду "root". Она выдаст сообщение следующего вида:

```
(hd0,5): filesystem is ext2
```

Некоторых, наверно, смутит упоминание ext2. Не волнуйтесь, так и должно быть. так же происходит каждый раз при нормальной загрузке системы. После того, как задан root-раздел, мы должны указать, какое ядро следует загрузить. Это задается командой "linux". Укажите то ядро, с которым вы обычно загружаетесь. В моем случае команда будет выглядеть так:

```
linux /boot/vmlinuz-2.6.32-020632rc6-generic root=/dev/sda5
```

Не пугайтесь, если вы не помните номер вашего ядра. Просто напишите "linux /boot/vmlinuz" и нажмите tab. На экран будет выведен список доступных ядер. Если же вам не важно какое ядро загружать, используйте просто vmlinuz - в последних версиях линукс это символическая ссылка на самое новое установленное ядро. Таким образом всё можно упростить до:

```
linux /boot/vmlinuz root=/dev/sda5
```

Обратите внимание на параметр "root=/dev/sda5" в конце строки! Здесь также нужно указать ваш root-раздел, но уже в более привычном формате. Без этого система может не загрузиться. Указав ядро и корневой раздел, мы готовы к загрузке системы. Для этого выполняем команду

```
boot
```

Вышеперечисленных действий должно хватить для загрузки системы. Если загрузка не удастся, попробуйте перед выполнением "boot" выполнить еще команду "initrd" (которая, вообще-то, выполняется при нормальной загрузке). Напишите "initrd /boot/initrd" и нажмите tab. Выберите вариант, соответствующий загружаемому ядру. В моем случае команда будет выглядеть так:

```
initrd /boot/initrd.img-2.6.32-020632rc6-generic
```

или просто:

```
initrd /boot/initrd
```

по аналогии с ядром создается символьная ссылка на последнюю версию initrd.

И уже после этого выполняйте "boot". В результате всех манипуляций мы должны успешно загрузиться в родную систему, а там уже можно выполнить "update-grub" и при необходимости восстановить нужные файлы из резервной копии (которую вы, конечно, делали).

Совет

Вы можете легко и быстро сделать себе инструкцию по аварийному запуску системы. Откройте ваш "/boot/grub/grub.cfg", найдите там пункт, который вы обычно загружаете, и скопируйте его оттуда. Допустим, он выглядит так:

```
menuentry "Ubuntu, Linux 2.6.32-020632rc6-generic" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi
    set quiet=1
    insmod ext2
    set root=(hd0,5)
    search --no-floppy --fs-uuid --set 0e717c2a-24bd-4abe-acfe-ecf98fc814f8
    linux /boot/vmlinuz-2.6.32-020632rc6-generic root=UUID=0e717c2a-24bd-4abe-acfe-
ecf98fc814f8 ro quiet splash
    initrd /boot/initrd.img-2.6.32-020632rc6-generic
}
```

Нам нужно оставить только три строки: "set root", "linux" и "initrd". Чтобы не забыть, допишем в конце команду "boot". В строке linux заменим "UUID=...." на ваш корневой раздел и уберем лишние параметры. В итоге получим такие строки:

```
set root=(hd0,5)
linux /boot/vmlinuz-2.6.32-020632rc6-generic root=/dev/sda5
initrd /boot/initrd.img-2.6.32-020632rc6-generic
boot
```

Осталось распечатать это на бумажке и сохранить оную в надежном месте. Теперь у вас всегда под рукой инструкция по загрузке на случай каких-либо неполадок!

Если же вышеприведенный метод не сработал (ядро не загрузилось или же вы вовсе не можете попасть в консоль grub2) и ничего не загружается - тут уж дело идет к переустановке grub2, что описана выше.

Загрузка Windows аналогичным образом

В случае неработоспособности Ubuntu вы можете загрузить другую установленную систему при наличии таковой. Для linux-систем схема будет той же: set root - linux - initrd - boot. Для Windows же все будет даже проще. Для начала так же укажем корневой раздел. На этот раз уже тот, на котором установлена Windows. В моем случае это "/dev/sda1", и команда будет иметь такой вид:

```
set root=(hd0,1)
```

После этого мы скомандуем grub2 передать управление загрузкой Windows-загрузчику NTLDR. Для этого просто выполним

```
chainloader +1
```

И наконец дадим команду на запуск:

```
boot
```

Если все прошло успешно - наслаждаемся загрузкой Windows. В случае, если на разных разделах установлено более одной версии Windows, нам нужно узнать, на каком из разделов находится NTLDR-загрузчик. Это лучше сделать заранее, посмотрев строку "set root" в пункте загрузки Windows файла "/boot/grub/grub.cfg"

Advanced

Даунгрейд до grub-legacy (возврат старой версии). В случае, если ваша дружба с grub2 так и не состоялась, вы можете заменить его на старый grub (ныне известный как grub-legacy), который на текущий момент более стабилен и привычен для многих. Для начала, конечно же, сделаем бэкап важных файлов. Создайте резервную копию файлов конфигурации как описано в конце I-ой главы. Также желательно забэкапить каталог "/boot/grub". Для этого выполните

```
sudo cp -R /boot/grub{, .bak}
```

Теперь мы можем приступить к удалению grub2. Для этого удаляем пакеты "grub-pc" и "grub2" (последний может быть не установлен):

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

Заметьте: удаление пакетов не приведет к полной очистке каталога "boot/grub". Теперь установим grub версии 0.97:

```
sudo apt-get install grub
```

Но после установки нам необходимо создать файл menu.lst и другие файлы загрузчика. Для этого нужно выполнить еще две команды:

```
sudo update-grub
```

Эта команда создаст файл меню загрузки. На заданные ей вопросы отвечайте "yes".

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

создаст остальные файлы загрузчика и запишет информацию в загрузочный раздел диска (MBR). Вместо "/dev/sda" в конце укажите тот жесткий диск, который используется в качестве загрузочного (установлен загрузочным в bios) Если установка прошла успешно - перезагружаемся и видим старый добрый grub с его menu.lst и без лишних хитростей

Из оригинального руководства не понятно, что делать с пакетом "grub-common". Данный пакет является зависимостью пакета "grub-rc", но вместе с последним не удаляется. Однако же, использование его вместе со старой версией grub - не лучшее решение. Наиболее безопасный вариант - после удаления "grub-rc" выполнить

```
sudo apt-get autoremove
```

Если будет предложено удалить "grub-common" как ненужный - значит, можно удалять.

Создание аварийно-спасательного GRUB2-liveUSB

Здесь мы рассмотрим, как создать загрузочную флэшку с установленным на нее grub2, через который можно будет загрузиться в случае, если grub2 на жестком диске испорчен. Делается это всего тремя командами (в примере /dev/sdz - флешка, в своей системе предварительно выясните точно, какой /dev/sd* соответствует флешке, набрав fdisk -l):

```
sudo mount /dev/sdz1 /mnt
```

```
sudo grub-install --root-directory=/mnt /dev/sdz
```

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

Либо

```
sudo update-grub -o /mnt/boot/grub/grub.cfg
```

На выбранном разделе (/dev/sdb1) будут созданы каталоги и скопированы файлы, необходимые для загрузчика, в MBR диска будет записан код, передающий управление на выбранный раздел. Загрузившись с флеш-диска, попадаем в меню grub2. Для его изменения достаточно вручную править файл конфигурации grub.cfg.

Консоль grub2 - команды и хитрости

Здесь мы рассмотрим некоторые команды консоли grub2. Все мы разбирать не будем, а изучим лишь самые необходимые и полезные. Напомню, чтобы попасть в консоль, нужно нажать "c" во время отображения меню загрузки. Итак, начнем!

ls

Довольно универсальная команда. при использовании в чистом виде выдает список жестких дисков и разделов. Также она может быть использована как одноименная команда в linux - для вывода содержимого каталога. Например,

```
ls /boot/grub
```

выведет содержимое каталога "/boot/grub" на разделе, который сейчас указан в качестве корневого (об этом чуть ниже). Еще одно полезное свойство команды "ls" - она позволяет получить информацию о любом разделе:

```
ls (hd0,5)
```

сообщит нам тип файловой системы на разделе, метку раздела (если таковая имеется), UUID и дату последнего изменения данных на разделе (в формате UTC).

cat

Данная команда аналогична своей тезке в linux - она выводит содержимое заданного файла. Используется в формате

```
cat /путь/имя_файл
```

linux

Загружает указанное linux-ядро. Аналог команды "kernel" в grub-legacy. Используется в формате

```
linux файл_ядра опция1=значение опция2 опция3
```

Например, так:

```
linux /boot/vmlinuz-2.6.32-020632-generic root=/dev/sda5 single
```

initrd

Загружает указанный initrd-образ. Используется так:

```
initrd /boot/initrd.img-2.6.32-020632-generic
```

Обратите внимание: версия initrd должна соответствовать версии загружаемого ядра!

load_env

Использование: `load_env [-f файл]`

Загружает переменные из файла блока среды в среду.

Опция `-f` необязательна, она переопределяет местоположение по умолчанию блока среды.

Что такое блок окружения GRUB? (GRUB environment block)

Часто полезно помнить небольшой объем информации от одной загрузки к другой. Например, вы можете захотеть установить запись меню по умолчанию на основе того, что было выбрано в последний раз. GRUB обычно не реализует поддержку записи файлов, чтобы свести к минимуму вероятность того, что загрузчик понесёт ответственность за повреждение файловой системы, поэтому конфигурационный файл GRUB не может просто создать файл обычным способом. Однако GRUB предоставляет «блок среды», который можно использовать для сохранения небольшого количества состояний.

Блок окружения представляет собой предварительно выделенный 1024-байтовый файл, который обычно находится в `/boot/grub/grubenv` (хотя вы и не должны это предполагать). Во время загрузки команда `load_env` загружает в нее переменные среды, а команда `save_env` сохраняет в ней переменные среды. Из работающей системы утилиту `grub-editenv` можно использовать для редактирования блока окружения.

`grub-mkconfig` использует эту возможность для реализации 'GRUB_SAVEDefault' (в `/etc/default/grub`).

chainloader

Передаёт управление загрузкой по цепочке другому загрузчику (загрузчик ищется на заданном в качестве `root` разделе). В общем случае требует указания файла для загрузки:

```
chainloader /путь/имя_файла
```

Однако, для NTLDR (загрузчика Windows) можно использовать

```
chainloader +1
```

boot

производит загрузку указанного ядра (заданного с "linux" и "initrd") или же другого загрузчика (заданного через "chainloader"). Используется без параметров

root

При использовании без параметров сообщает, какой раздел сейчас используется в качестве корневого (по умолчанию - корневой раздел системы, на которую установлен текущий grub2) и тип файловой системы на этом разделе (для ext3/4 тип будет указан как ext2, это нормально). Также команда может быть использована для задания другого root-раздела. Раздел задается в "grub device" формате, то есть "(hd*,*)". Например:

```
root (hd0,5)
```

После задания раздела команда сообщит новый root-раздел и тип файловой системы. Примечание: "root (hd*,*)" не всегда корректно срабатывает. Более предпочтительным вариантом является "set root" (см. ниже)

save_env

Использование: save_env [-f файл] значение ...

Сохранить значения из переменных среды в файл блока окружения.

Опция -f переопределяет местоположение по умолчанию блока среды.

set

Весьма универсальная команда для изменения различных параметров. Служит для задания значений переменных и используется в формате

```
set переменная=значение
```

Наиболее необходимое ее применение - для задания root-раздела. Для этого изменяется переменная "root", в качестве значения задается необходимый раздел, например:

```
set root=(hd0,5)
```

(именно такая технология используется в "grub.cfg") Также с ее помощью можно, например, "на лету" изменить цвет текста в меню и консоли, что позволяет опробовать цветовую схему перед установкой ее в качестве основной. Для этого изменяем переменные "color_normal" и "color_highlight" для обычного пункта (и текста в консоли) и выделенного пункта соответственно. Например, попробуем такой вариант:

```
set color_normal=magenta/green  
set color_highlight=light-blue/black
```

(Подробнее о цветах читайте в главе II, раздел "настройка цвета пунктов меню").

search

Пожалуй, одна из самых полезных команд. Потомок "find" и "uuid" из grub-legacy. Служит для поиска раздела по UUID, метке или заданному файлу. Имеет следующие ключи:

-u (или --fs-uuid) - поиск раздела по UUID;

-l (или --label) - поиск по метке раздела;

-f (или --file) - поиск по указанному файлу;

-n (или --no-floppy) - не проверять флоппи-дисковод (чтоб не трещал);

-s (или --set) - установить найденный раздел в качестве значения заданной переменной.

Зачем это нужно? Дело в том, что при подключении/отключении жестких дисков (и в некоторых других случаях) нумерацию дисков и разделов может "поехать", и тогда наша команда "set root=(hd0,5)" в "grub.cfg" будет вести или в никуда или на раздел не того диска. Поэтому в "grub.cfg" вы найдете строчку с "search". Для поиска раздела с известным нам uuid запустим "search" с ключом "-u":

```
search -u uuid_нужного_раздела
```

Команда сообщит нам, какой раздел соответствует заданному UUID. Однако, UUID длинный, и хотя его и можно узнать, вводить вручную такое количество символов проблематично, и существует большой шанс сделать ошибку. Вместо этого мы воспользуемся поиском по файлу, который точно имеется на нужном разделе. Допустим, мы ищем раздел boot-раздел нашей системы. Найдем его по файлу "grub.cfg":

```
search -f /grub/grub.cfg
```

Обратите внимание, что при использовании такого способа указанный файл должен иметься только на одном разделе! Получив искомое, можно вручную задать нужный раздел в качестве root-раздела, но гораздо проще воспользоваться ключом "-s". Если после него не указана переменная, будет использована "root", то есть найденный раздел как раз и будет задан в качестве корневого. Допустим, раздел с Windows имеет метку "C_drive". Установим его в качестве корневого для загрузки с него:

```
search -l C_drive -s
```

В качестве послесловия отмечу, что остается неясным, зачем разработчики используют в "grub.cfg" строчку с "set root", если вполне можно обойтись "search ...".

"search ..." может не хватить если хочется явно задать устройство для загрузки.

lsfonts

Эта команда отобразит список загруженных в настоящий момент шрифтов (полезно для тех, кто все-таки решил поразвлечься с настройкой шрифтов в grub2, конвертацией их в .pf-формат и прочим).

help

При использовании в чистом виде выведет список доступных команд. В формате

```
help r
```

выведет справку по всем командам, начинающимся на "r".

```
help search
```

отобразит справку по команде "search".

halt

Выключит компьютер (в теории) или введет его в ступор (у меня).

reboot

Перезагрузит компьютер

background_image

Позволяет "на лету" изменить фоновое изображение. Используется в формате

```
background_image /путь/имя_файла
```

Дает замечательную возможность посмотреть на выбранную картинку в действии, избегая лишних перезагрузок. В сочетании с заменой цветов через "set" позволит довольно быстро подобрать подходящий вариант оформления. При использовании без параметров сбросит текущее фоновое изображение. однако, заданные цвета останутся, так что если у вас черный цвет текста - на черном фоне вы его не увидите. Примечание: данная команда не заменит ваши настройки оформления, фон будет изменен лишь на текущий сеанс.

vbeinfo

Выведет список поддерживаемых видеорежимов. Этой командой полезно будет воспользоваться перед настройкой разрешения в "/etc/default/grub".

configfile

Потомок одноименной команды в grub-legacy. Позволяет загрузить другой файл конфигурации с другим меню вместо текущего "grub.cfg". Создав новый файл, вы можете сделать другое меню с другими настройками, оформлением и пунктами, не захламывая основное меню. Используется так:

```
configfile /путь/другой_файл_меню
```

Создание и использование новых меню позже будет рассмотрено отдельно.

terminal_output.console

Позволяет переключиться на обычную черно-белую цветовую гамму. Весьма полезно при работе с консолью в том случае, если у вас установлено фоновое изображение. Картинка на фоне - это, конечно, красиво, но на некоторых участках фона может быть не виден текст (если картинка не везде однородная). Выполнив эту команду, вы получите контрастную схему (белый текст на черном фоне) и хороший крупный шрифт. Если вы часто используете консоль grub2, имеет смысл добавить в 40_custom пункт с этой командой, например:

```
menuentry "black&white mode"{  
  terminal_output.console  
}
```

terminal_output

Использование: terminal_output [--append|--remove] [терминал1] [терминал2] ...

Перечисляет или выбирает выходной терминал. Без аргументов перечисляет активные и доступные выходные терминалы.

Параметр --append добавляет терминалы в список активных выходных терминалов; Все они будут получать данные из GRUB.

Параметр --remove удаляет терминалы из активного списка.

Без параметров, кроме списка имен терминалов, активирует только перечисленные имена терминалов.

play

С помощью данной команды можно создавать любые шедевры. В качестве вывода звука будет использоваться пищалка на материнской плате. Рекомендуется подключать стороннюю пищалку для более качественного звука. Синтаксис команды с переводом:

```
play файл | темп [высота1 длина1] [высота2 длина2]
```

Чтобы проиграть файл, пишем play и указываем путь до файла формата MID. Для того чтобы создавать свои мелодии нужно задать темп, а также высоту и длительность каждого звука. Пример мелодии:

```
play 300 100 2 200 2 300 2 400 2 500 2 400 2 300 2 200 2 100 2 0 0 100 2 200 2 300 2 400 2  
500 2 400 2 300 2 200 2 100 2 0 0
```

Можно писать мелодии любого размера без ограничений.

tr

Заменяет буквы в тексте.

```
tr [опции] [что] [на что изменить] [текст]
```

--set=ЗНАЧЕНИЕ - устанавливает полученный результат в переменную ЗНАЧЕНИЕ. -U --upcase - переводит все символы в верхний регистр. -D --downcase - переводит все символы в нижний регистр.

testspeed

Даёт возможность протестировать скорость чтения файла с диска.

```
testspeed [-s РАЗМЕР] [путь до файла]
```

-s, --size=ЧИСЛО - объём данных каждой операции чтения.

time

Замеряет время выполнения любой команды.

```
time [команда] [аргументы]
```

sleep

Заморозка grub на указанное количество секунд. Аналогична команде sleep в linux, но без отмены командой Ctrl-C.

```
sleep [ключи] [СЕКУНДЫ]
```

-v, --verbose - показывает отсчёт в одной строке. -i, --interruptible - разрешает прерывание сна кнопкой ESC. Если не указать данный ключ, то пока не закончится отсчёт нельзя будет предпринимать какие-либо действия.

Премудрости ;)

Ну и для полного счастья ниже я приведу небольшой список премудростей, которые помогут вам облегчить работу в консоли: уже упомянутое автодополнение команд (как в bash) по нажатию tab. tab в пустой строке выведет полный список доступных команд. Это самое автодополнение имеет нетривиальный способ использования. Если вы не помните, какой раздел вам нужен, напишите

"root (hd0," (или "root (hd1." и далее по списку) и нажмите tab. Будет выведен список разделов заданного диска с указанием файловых систем, UUID и времени последнего изменения. Имя раздела вида "(hd*,*)" можно использовать в начале пути к файлу, например, вот так:

```
cat (hd0,6)/home/username/каталог/файл
```

Еще один полезный пример

```
ls (hd0,3)/
```

позволяет просмотреть содержимое заданного раздела без переключения root. И кстати в терминале linux тоже можно с успехом использовать данный хак.

Ещё одна фишка

```
ln -s /usr/sbin/update-grub /usr/sbin/grub-update
```

Позволяет облегчить себе жизнь и не путаться в названиях программы. Теперь обновлять grub.cfg можно командой grub-update. Если пишет ошибку то что файла /usr/sbin/update-grub не существует, то выполняем команду which update-grub> и заменяем /usr/sbin/ на путь до программы.

Загрузка другого grub2 из текущего

Как выяснилось в ходе экспериментов, все весьма просто. загружаемся в grub2, переходим в консоль нажатием "c". Загрузка другого grub2 происходит по методу "set root - chainloader - boot". Для начала определимся с диском, в MBR которого прописан grub2, который мы собираемся загрузить. Выполните команду

```
ls
```

чтобы увидеть список текущих дисков и разделов. Диск, с которого производилась загрузка будет именоваться (hd0). В нашем примере мы загрузим с флэшки основной grub2 на жестком диске. Если флэшка - (hd0), то жесткий диск будет (hd1). Выполняем

```
set root=(hd1)
```

или же

```
root (hd1)
```

Во втором случае видим сообщение, что файловая система неизвестна. Это нормально. Передаем управление по цепочке

```
chainloader +1
```

и даем команду на запуск

```
boot
```

и попадаем в искомый загрузчик.

Создание дополнительных меню загрузки

В этом разделе мы рассмотрим создание своих собственных загрузочных меню со своей конфигурацией. Суть в том, что вместо уже знакомого нам "grub.cfg" можно загрузить другой файл конфигурации. Это позволяет создать дополнительное меню, совершенно другое — с другими пунктами, настройками, оформлением. Те, кто имел дело с установкой ArchLinux, могли заметить несколько подменю в grub установочного диска. Примерно такой метод мы и будем использовать. Итак, создадим наш новый файл конфигурации. Писать с нуля мы его, конечно же, не будем, а сотворим на основе имеющегося "grub.cfg". Для чего скопируем последний, дав новое имя, например:

```
sudo cp /boot/grub/{grub,custom}.cfg
```

Поскольку файл скопируется с правами 444, выставим ему права на запись:

```
sudo chmod 744 /boot/grub/custom.cfg
```

Теперь открываем его в любимом текстовом редакторе и начинаем ковырять. Если вы в достаточной мере знаете структуру "grub.cfg" и понимаете, какие строки что делают, можно значительно сократить свой файл, вырезав лишнее, и привести его к более удобочитаемому виду. Однако же, можно и не трогать руками все подряд, а изменить лишь интересующие параметры, что мы и сделаем. Для начала найдем строку "set default=...." (у меня это 13-я строка). Здесь можно задать пункт меню по умолчанию, указав или его номер или точное название. Далее отыщем строку "set gfxmode...." (24-я в моем случае). Здесь задается разрешение экрана в привычном формате WxH. Можно также указать глубину цвета, например "1024x768x32". Можно указать несколько вариантов, при этом если первый не сможет быть установлен, будет загружен второй, если и он не сработает - третий и т.д. Варианты перечисляются через запятую и без пробелов:

```
set gfxmode=1024x768x32,1024x768x24,1024x768
```

Теперь зададим время отображения меню. Для этого ищем строку "set timeout", но именно ту, которая идет после "else" (36). Здесь и задается задержка в секундах. Установите -1 для отключения таймера. Настроим оформление нашего меню. Для задания фонового изображения найдем строку с "if background_image....." (45). (здесь подразумевается, что в вашем grub.cfg используется фоновое изображение. Если нет - смотрите пример файла в конце этого раздела). После команды background_image прописан путь к файлу картинки. Поменяйте его на нужный файл (обратите внимание, что разрешение картинки должно соответствовать заданному выше разрешению экрана! Но не обязательно, только чревато низким качеством фона). Строкой ниже мы можем задать цвета текста. Правятся они точно так же, как и в "/etc/grub.d/05_debian_theme" (подробнее о цветах читайте в главе II, раздел "настройка цвета пунктов меню"). Еще можно проделать один интересный трюк: уберите строки с "set color_highlight" и "else". Теперь настройка будет такой: color_normal задает цвет/фон текста сверху и снизу (версия grub и комментарии) и

текста в консоли `menu_color_normal` отвечает за цвет пунктов меню и фон прямоугольника, в котором они находятся `menu_color_highlight` задает цвет/фон выделенного пункта меню. Закончив с настройкой, перейдем собственно к пунктам. Ниже идут пункты из вашего "grub.cfg". Измените их, удалите ненужные и добавьте новые, пересортируйте на свой вкус. Ниже будет приведен пример с немного отредактированными тремя пунктами загрузки. Настроив свой файл, сохраните его. Теперь нам нужно добавить возможность загрузить его вместо "grub.cfg". Для этого в последний нужно будет добавить еще один пункт, что мы сделаем через "/etc/grub.d/40_custom". Для смены файла конфигурации в grub2 используется команда "configfile". Ей мы и воспользуемся, добавив в "40_custom" запись такого вида:

```
menuentry "my very custom menu"{
  configfile /boot/grub/custom.cfg
}
```

Если ваш путь к файлу отличается - исправьте его. Если вы используете отдельный boot-раздел, путь будет таким: "/grub/custom.cfg". Также учтите, что "40_custom" должен заканчиваться пустой строкой! Если вы хотите, чтобы ваше новое меню загружалось вместо стандартного "grub.cfg", установите новый пункт в качестве пункта по умолчанию. Также можете установить таймаут для "grub.cfg" в 0 (в "/etc/default/grub"), но перед этим рекомендую протестировать свежее испеченное меню на работоспособность. Обновите свой "grub.cfg" выполнив

```
sudo update-grub
```

Теперь пункт перехода в новое меню появится в обычном. для того чтобы выйти обратно в первое меню, достаточно нажать escape. Вот, в принципе, и вся премудрость. Таких файлов можете насоздавать себе хоть десять штук, тут уж зависит от потребностей.

Пример файла "custom.cfg"

Ниже я приведу в качестве образца то, что получилось у меня (файл сильно урезан до необходимого минимума):

```
have_grubenv=true
load_env

### выбранный пункт по умолчанию
set default=0
### таймаут меню
set timeout=2
### разрешение
set gfxmode=800x600x32,800x600x24,800x600

insmod ext2
search -u 0e717c2a-24bd-4abe-acfe-ecf98fc814f8 -s
insmod gfxterm
insmod vbe
if terminal_output gfxterm ; then true ; else terminal gfxterm; fi
insmod png

### фоновое изображение
background_image /usr/share/images/grub/yellow.png
### настройка цветов
set menu_color_normal=green/light-blue
set menu_color_highlight=red/blue

menuentry "9.10 @ 2.6.32" {
  search -u 0e717c2a-24bd-4abe-acfe-ecf98fc814f8 -s
  linux /boot/vmlinuz-2.6.32-020632-generic root=UUID=0e717c2a-24bd-4abe-acfe-ecf98fc814f8 ro
```



```
quiet splash
initrd /boot/initrd.img-2.6.32-020632-generic
}
menuentry "9.10 @ 2.6.32 (recovery)" {
search -u 0e717c2a-24bd-4abe-acfe-ecf98fc814f8 -s
linux /boot/vmlinuz-2.6.32-020632-generic root=UUID=0e717c2a-24bd-4abe-acfe-ecf98fc814f8 ro
single
initrd /boot/initrd.img-2.6.32-020632-generic
}
menuentry "windooz"{
search -u AC9C75E59C75AA8A -s
chainloader +1
}

menuentry "Назад"
{
configfile /boot/grub/grub.cfg
}
```

MBR — бэкап, восстановление и загрузка через образ

Для тех, кто не знает, поясню: MBR — фрагмент первого сектора жесткого диска, в котором, в том числе, прописан код, запускающий загрузчик. При загрузке с диска оттуда считывается информация о том, как запустить загрузчик. grub2 прописывает свой код в MBR при установке (и это же мы делаем, выполняя «grub-install ... /dev/sda»). На случай повреждения MBR (причины бывают разными) и для возможности быстрого восстановления работоспособности grub2 полезно будет иметь резервную копию MBR жесткого диска. Чтобы создать ее, сначала нужно определиться, с какого из жестких дисков производится загрузка (можно посмотреть с помощью «sudo fdisk -l»). После этого скопируем первый сектор (первые 512 байт) диска в файл с помощью команды dd:

```
sudo dd if=/dev/sda of=mbr.bin bs=512 count=1
```

Если ваш загрузочный диск — не sda, поменяйте значение на свое (на самом деле, код загрузчика занимает первые 446 байт, а дальше записана таблица разделов, но для загрузки нашего образа он должен быть не меньше 512 байт). Наш образ готов! Данные сохранены в файл «mbr.bin» в вашей домашней директории. Используя полученный образ, можно загрузить основной grub2 из-под другого при поврежденном MBR диске. Если у вас есть другая установленная linux-система с grub2 на другом жестком диске или загрузочная флэшка с grub2 (создание таковой описано выше) — можете попробовать на практике нижеприведенный совет. Итак, в моем примере я загружаюсь со своей спасительной флэшки с grub2, на которую я предварительно закинул образ MBR своего жесткого диска. Загрузившись в grub2, выходим в консоль (нажав «с»). Указываем в качестве корневого раздела тот жесткий диск, на котором по сценарию попорчен MBR, и с которого мы хотим запустить grub2. Можно указать сам диск или любой из его разделов, например так:

```
set root=(hd1)
```

Проверим результат командой

```
root
```

Если все верно — загружаем наш образ с помощью chainloader. Здесь нужно указать полный путь, начиная с раздела, на котором лежит наш файл. Флэшка, с которой я загружен именуется (hd0), так что команда будет выглядеть так:

```
chainloader (hd0,1)/mbr.bin
```

Если все тихо — даем команду на запуск:

```
boot
```

и наслаждаемся родным grub2.

Примечание: в MBR (и, соответственно, в полученном образе) нет четкого указания жесткого диска, поэтому через наш образ загрузится grub2 с того раздела, который был указан в качестве корневого (root) в консоли grub2. Загрузив родную систему, мы можем восстановить из-под нее MBR диска. Но если лишнего grub2 у вас под рукой нет, действовать мы будем с LiveCD.

Грузимся с LiveCD любой linux-системы (в данном случае поддержка grub2 роли не играет). Для восстановления MBR нам нужно будет «раскатать» ранее созданный образ обратно в первый сектор диска. Но для начала проверим таблицу разделов, ведь вполне возможно, что grub2 не грузился и из-за нее. Для этого выполните

```
sudo fdisk -l
```

Если fdisk видит разделы - таблицу разделов трогать не будем, а восстановим только код загрузчика. В этом случае нам нужно перезаписать лишь первые 446 байт. Если же "убита" и таблица разделов - все 512. Определившись, что именно мы делаем, копируем нужное количество байт из образа в первый сектор диска. В примере ниже таблица разделов в порядке, восстанавливаем только grub2:

```
sudo dd if=/mnt/mbr.bin of=/dev/sda bs=446 count=1
```

Если нужно восстановить и таблицу разделов — укажите 512 вместо 446. В качестве значения if пропишите путь к файлу образа, в of — тот диск, загрузочный сектор которого мы восстанавливаем. Если все прошло успешно — перезагружаемся, грузимся с починенного диска и радуемся.

Еще одна полезная возможность — полученный образ можно использовать для загрузки grub2 через NTLDR (в случае, если grub2 не загружает нормально Windows, и вы хотите сделать NTLDR основным загрузчиком). Но об этом как-нибудь позже.

P.S. Через grub4dos данный трюк проверить не получится, возникает ошибка: Geom Error Missing Helper.

Создание LiveCD с GRUB2

В случае, если ваш компьютер не поддерживает загрузку с usb, можно создать cd-диск с grub2, через который можно будет загрузить свою систему при неисправности основного grub2. Для начала создадим в удобном месте каталог, в котором будем собирать наш диск, а в ней вложенные каталоги "/boot" и "/boot/grub":

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

Теперь сразу же определимся с оформлением. если вам нужно простое черно-белое загрузочное меню - следующее действие можно пропустить. если же вы хотите получить меню с настраиваемым разрешением, более гибкой настройкой цветов и даже по желанию фоновым изображением - нам нужно будет добавить на диск шрифт, который используется для отображения графического меню. Скопируем его:

```
cp /usr/share/grub/unicode.pf2 tmp/iso/boot/grub
```

Теперь нам нужно написать файл конфигурации "grub.cfg", содержащий загрузочное меню и настройки. Вы можете взять за основу ваш "/boot/grub/grub.cfg" и изменить его под текущие нужды, а можете написать его с нуля - оба варианта подходят если вы понимаете структуру "grub.cfg". В противном случае вы можете взять готовый шаблон, помещенный в конце этого раздела и отредактировать его. Для тех же, кто избрал "путь самурая" - подробнее о создании собственного "grub.cfg" вы можете почитать выше, в разделе "создание дополнительных меню загрузки". Здесь же опишу лишь некоторые специфичные для LiveCD вещи. Если вы берете за основу "/boot/grub/grub.cfg" - уберите из него строки, задающие ваш рут-раздел системы в качестве root ("search.... uuid_раздела...."). Корневым разделом должен быть cd-диск! Если вы выбрали вариант с графическим меню - нужно обязательно подгрузить скопированный ранее шрифт. Кроме того, следует, собственно, включить графический режим. Для этого добавьте такие строки:

```
loadfont /boot/grub/unicode.pf2
insmod gfxterm
insmod vbe
if terminal_output gfxterm ; then true ; else terminal gfxterm; fi
```

И еще один очень важный момент: в начале файла должна иметься такая строка: **insmod part_msdos** загрузка данного модуля **обязательна!** Без него полученный grub2 увидит только жесткий диск, но не увидит таблицу разделов на нем, и такой LiveCD будет бесполезен. Про настройки цветов/разрешения и всего прочего читайте в вышеупомянутом разделе. А теперь же перейдем к созданию самого меню. Вы можете скопировать нужные пункты из "/boot/grub/grub.cfg", но я рекомендую написать их с нуля, ибо для корректной работы тамошних пунктов потребуются загрузка дополнительных модулей. А написать их несложно, для Linux-системы пункт меню должен иметь такой вид:

```
menuentry "9.10 @ 2.6.32" {
  search -u uuid_раздела -s
  linux /boot/vmlinuz-xxxxxxx root=/dev/sdXY quiet splash single или_что_там_еще
  initrd /boot/initrd.img-xxxxxx
}
```

В качестве uuid нужно задать тот раздел, где находится ваш /boot (если у вас общий / - пишите его, если отдельный /boot-раздел - именно его и указываем, а не /. В случае с отдельным /boot пути будут выглядеть как "/vmlinuz...." и "/initrd....") UUID узнать проще всего командой blkid запущенной от рута. Для Windows будет что-то такое:

```
menuentry "winxp" {
  search -u uuid_раздела_c_win -s
  chainloader +1
}
```

Закончив с файлом, сохраняем его как "grub.cfg" в целевом каталоге (в моем случае - "~/tmp/iso/boot/grub/grub.cfg"). Теперь мы готовы к созданию iso-образа нашего LiveCD. В этом нам поможет утилита grub-mkrescue. команда будет выглядеть так:

```
grub-mkrescue --overlay=tmp/iso/ tmp/grub.iso
```

Вместо "tmp/iso" укажите свой каталог, а вместо "tmp/grub.iso" - путь, куда сохранить полученный iso-образ. Если все успешно - получаем на выходе очень небольшой iso-шник (~2.8M), прожигаем его на rw-шку (лучше программой cdrecord из пакета cdrtools) и пробуем грузиться!

Пример файла "grub.cfg"

Как и обещал, привожу пример файла "grub.cfg", корректно работающего на LiveCD с комментариями:

```
### модуль файловой системы iso9660 (cd-диск)
insmod iso9660
### модуль для распознавания таблицы разделов жесткого диска! must have!!
insmod part_msdos

### выбранный пункт по умолчанию
set default=0
### таймаут меню
set timeout=2
### разрешение
set gfxmode=800x600x32,800x600x24,800x600

### модуль файловой системы ext2/3/4
insmod ext2
### загружаем шрифт и включаем графическое меню
loadfont /boot/grub/unicode.pf2
insmod gfxterm
insmod vbe
if terminal_output gfxterm ; then true ; else terminal gfxterm; fi

### настройка цветов
set color_normal=white/black
set menu_color_normal=green/black
set menu_color_highlight=light-blue/black

#0
menuentry "9.10 @ 2.6.33rc1" {
search -u 0e717c2a-24bd-4abe-acfe-ecf98fc814f8 -s
linux /boot/vmlinuz-2.6.33-020633rc1-generic root=UUID=0e717c2a-24bd-4abe-acfe-ecf98fc814f8
ro quiet splash
initrd /boot/initrd.img-2.6.33-020633rc1-generic
}

#1
menuentry "9.10 @ 2.6.33rc1 (recovery)" {
search -u 0e717c2a-24bd-4abe-acfe-ecf98fc814f8 -s
linux /boot/vmlinuz-2.6.33-020633rc1-generic root=UUID=0e717c2a-24bd-4abe-acfe-ecf98fc814f8
ro single
initrd /boot/initrd.img-2.6.33-020633rc1-generic
}

#2
menuentry "windooz"{
search -u AC9C75E59C75AA8A -s
chainloader +1
}

menuentry "ubuntu-11.04.iso" {
loopback loop (hd0,2)/ubuntu-11.04.iso
linux (loop)/install/vmlinuz boot=install iso-scan/filename=/ubuntu-11.04.iso
initrd (loop)/install/initrd.gz
}
```

```

menuentry "linuxmint-12-kde-dvd-64bit.iso" {
search -l FAT -s
#set root=(hd0,msdos1)
loopback loop /linuxmint-12-kde-dvd-64bit.iso
linux (loop)/casper/vmlinuz file=(loop)/preseed/mint.seed boot=casper iso-
scan/filename=/linuxmint-12-kde-dvd-64bit.iso quiet splash --
initrd (loop)/casper/initrd.lz
}

menuentry "ubuntu-12.04-beta2-desktop-i386.iso" {
search -l FAT -s
#set root=(hd0,msdos1)
loopback loop /ubuntu-12.04-beta2-desktop-i386.iso
linux (loop)/casper/vmlinuz file=(loop)/preseed/ubuntu.seed boot=casper iso-
scan/filename=/ubuntu-12.04-beta2-desktop-i386.iso quiet splash --
initrd (loop)/casper/initrd.lz
}

menuentry "ubuntu live-dvd" {
loopback loop (hd0,2)/ubuntudvd.iso
linux (loop)/install/vmlinuz boot=install iso-scan/filename=/ubuntudvd.iso
initrd (loop)/install/initrd.gz
}

menuentry "Mint live-cd" {
loopback loop (hd0,2)/mint.iso
linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=/mint.iso
initrd (loop)/casper/initrd.gz
}

menuentry "pmagic-4.10.iso" {
loopback loop (hd0,2)/pmagic-4.10.iso
linux (loop)/pmagic/bzImage boot=casper iso-scan/filename=/pmagic-4.10.iso
initrd (loop)/pmagic/initramfs
}

menuentry "Plop.bin" {
root (hd0,2)
linux16 (hd0,2)/boot/grub/plpbt.bin
}

menuentry "Plop.com" {
root (hd0,2)
linux16 (hd0,2)/boot/grub/plpinstc.com
}

menuentry "ubuntu-11.10-desktop-i386.iso" {
loopback loop /boot/iso/ubuntu-11.10-desktop-i386.iso
linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=/boot/iso/ubuntu-11.10-desktop-
i386.iso noeject noprompt --
initrd (loop)/casper/initrd.lz
}

menuentry "linuxmint-12-gnome-dvd-32bit.iso" {
loopback loop /boot/iso/linuxmint-12-gnome-dvd-32bit.iso
linux (loop)/casper/vmlinuz file=(loop)/preseed/mint.seed boot=casper iso-
scan/filename=/boot/iso/linuxmint-12-gnome-dvd-32bit.iso quiet splash --
initrd (loop)/casper/initrd.lz
}

menuentry "Lucid ISO" {
loopback loop (hd0,1)/boot/iso/ubuntu-10.04-desktop-i386.iso
linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=/iso/ubuntu-10.04-desktop-i386.iso
noprompt noeject
initrd (loop)/casper/initrd.lz
}

menuentry "ubuntu-9.10-desktop-amd64.iso" {
loopback loop (hd0,1)/boot/iso/ubuntu-9.10-desktop-amd64.iso
linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=/iso/ubuntu-9.10-desktop-amd64.iso
noprompt quiet splash
initrd (loop)/casper/initrd.lz
}

menuentry "gparted-live-0.6.1-2.iso" {

```

```

loopback loop (hd0,1)/boot/iso/gparted-live-0.6.1-2.iso
linux (loop)/live/vmlinuz boot=live union=aufs noswap noprompt ip=frommedia
findiso=/boot/iso/gparted-live-0.6.1-2.iso toram=filesystem.squashfs
initrd (loop)/live/initrd.img
}

menuentry "systemrescuecd-x86-1.5.8.iso" {
loopback loop (hd0,1)/boot/iso/systemrescuecd-x86-1.5.8.iso
linux (loop)/isolinux/rescue64 setkmap=us isoloop=/systemrescuecd-x86-1.5.8.iso
initrd (loop)/isolinux/initram.igz
}

menuentry "pmagic-5.2.iso" {
loopback loop (hd0,1)/boot/iso/pmagic-5.2.iso
linux (loop)/pmagic/bzImage iso_filename=/boot/iso/pmagic-5.2.iso boot=live load_ramdisk=1
prompt_ramdisk=0 noeject noprompt
initrd (loop)/pmagic/initramfs
}

menuentry "Boot IMG - Seagate Tools" {
linux16 /memdisk bigraw
initrd16 /SeaTools.img
}

# Live Gparted удалось загрузить, только распаковав диск и скопировав его на флешку
(директорию live)
menuentry "Gparted" {
linux /live/vmlinuz boot=live config noswap ip=frommedia nosplash --
initrd /live/initrd.img
}

#А если не LiveCD, а что-то полезное установить?
menuentry "debian-6.0.3-amd64-netinst.iso" {
loopback loop /boot/iso/debian-6.0.3-amd64-netinst.iso
linux (loop)/install.amd/vmlinuz vga=normal --
initrd (loop)/install.amd/initrd.gz
}

menuentry "debian installer amd64 netboot" {
linux /boot/debian/linux auto preseed/url=http://www.panticz.de/pxe/preseed/xen.seed
locale=en_US console-setup/layoutcode=de netcfg/choose_interface=eth0
debconf/priority=critical --
initrd /boot/debian/initrd.gz
}

# У "ванильной" убунты и alternate способы установки (и пути к ядрам) отличаются. Так что
отдельный пример для "alternate":
menuentry "Xubuntu 12.04 64 bit" {
loopback loop /xubuntu-12.04-alternate-amd64.iso
linux (loop)/install/vmlinuz iso-scan/filename=/xubuntu-12.04-alternate-amd64.iso
vga=normal noeject noprompt nosplash --
initrd (loop)/install/initrd.gz
}

#FreeDOS - ностальгирующая экзотика!
menuentry "FreeDOS 1.0" {
loopback loop /boot/iso/fdbasews.iso
linux16 (loop)/isolinux/data/memdisk
initrd16 (loop)/isolinux/data/fdboot.img
}

### 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
}

```

Создание LiveCD с GRUB2 на базе имеющегося дистрибутива

(за идею спасибо тов. komix)

Поскольку тратить целую болванку ради записи iso-шника с grub2 в 3 мегабайта было бы глупо, можно совместить приятное с полезным, а именно: LiveCD с grub2 и установочный диск какой-нибудь системы. Исходить мы будем из того, что имеется iso некоего LiveCD, к которому мы хотим прикрутить grub2. Порядок действий будет в целом аналогичен описанному в предыдущем пункте. Первым делом создадим каталог, в котором будем собирать наш новый диск. В нём же создадим каталог /boot/grub и каталог, в который будут помещены файлы нашей live-системы (я назову ее live):

```
mkdir -p tmp/iso/{boot/grub,live}
```

Теперь нам нужно примонтировать iso-образ нашей системы в свежесозданный каталог для нее:

```
sudo mount -o loop downloads/some-live-system.iso tmp/iso/live
```

Теперь создайте "grub.cfg", как это описано в предыдущем пункте (не забудьте скопировать файл шрифта, если хотите использовать графическое меню). Здесь все аналогично вышеописанному способу, но с той лишь разницей, что в меню мы добавим пункт для загрузки нашей live-системы. Как именно нужно запускать "живое ядро"? На этот вопрос нет универсального ответа. Узнать это можно, покопавшись в настройках прилагаемого загрузчика (обычно это isolinux или же старый grub - в обоих случаях понять логику несложно). Приведу конкретный пример для минидистрибутива Parted Magic:

```
menuentry "Parted Magic" {  
  linux /live/bzImage  
  initrd /live/initramfs  
}
```

В данном случае этого будет достаточно. Для других систем могут потребоваться дополнительные параметры загрузки ядра. Ядро же может лежать в "/live/boot" или в другом каталоге в зависимости от конкретного дистрибутива. Обратите внимание на то, что здесь не используются строки вида "set root..." или "search...". Если вы создали правильный "grub.cfg", то корневым разделом при загрузке устанавливается cdrom, поэтому нет нужды его задавать повторно. После того, как "grub.cfg" готов, можно приступить к созданию iso-образа:

```
sudo grub-mkrescue --overlay=tmp/iso tmp/my-live-system-grub2.iso
```

Продолжительность процесса зависит от размера оригинального дистрибутива. По окончании получаем свежее испеченный iso-образ с оригинальным дистрибутивом и бонусом в виде grub2. Отмонтируем оригинальный iso:

```
sudo umount tmp/iso/live
```

и прожигаем новый iso-шник на болванку. Грузимся и проверяем результат.

Добавление в меню пункта загрузки с CD/DVD & USB

Средствами исключительно GRUB загрузиться с CD-ROM/USB не получится, но "мир не без добрых людей". Существуют мультизагрузчики, не такие многофункциональные, как GRUB, но, зато, умеющие грузить систему с CD-ROM/USB, даже если BIOS такой возможности не предоставляет.

Способ 1. Загрузка с CD & USB посредством Plop Boot Manager

Сама загрузка будет осуществляться посредством Plop Boot Manager (<https://www.plop.at/en/bootmanager/download.html>). Для этого качаем архив Plop'a, откуда извлекаем файл plpbt.bin, и кидаем его в каталог /boot/ нашей системы. Далее открываем файл /etc/grub.d/40_custom:

```
nano /etc/grub.d/40_custom
```

И добавляем в него следующие строчки (мой системный раздел - (hd0,1)):

```
menuentry "Boot CD/DVD & USB -> Plop Boot Manager" {  
    set root=(hd0,1)  
    linux16 /boot/plpbt.bin  
}
```

Либо вот таким способом, чтобы не узнавать каждый раз номер раздела и диска.

```
menuentry "Boot CD/DVD & USB -> Plop Boot Manager" {  
    search -f /boot/plpbt.bin -s  
    linux16 /boot/plpbt.bin  
}
```

Затем обновляем grub.cfg:

```
sudo update-grub
```

Перезагружаемся, радуемся результату.

Способ 2. Загрузка с CD посредством Smart Boot Manager

В этом случае загрузка будет осуществляться посредством Smart Boot Manager (<http://btmgr.sourceforge.net>). Итак, качаем два файла: memdisk.bin - "крохотный" файл из пакета Питера Анвина syslinux (<http://syslinux.zytor.com>) и sbomgr.dsk - образ загружаемой дискеты из Smart Boot Manager. Кидаем оба файла в каталог /boot/ нашей системы. Далее открываем файл /etc/grub.d/40_custom:

```
nano /etc/grub.d/40_custom
```

И добавляем в него следующие строчки (мой системный раздел - (hd0,1)):

```
menuentry "Boot CD/DVD" {  
    set root=(hd0,1)  
    linux16 /boot/memdisk.bin  
}
```



```
initrd16 /boot/sbootmgr.dsk  
}
```

Затем обновляем grub.cfg:

```
sudo update-grub
```

Перезагружаемся, смотрим результат.

Деактивация recordfail grub2

GRUB2 поставляется с функцией, которая после неудачной попытки загрузки во время следующей загрузки автоматически останавливается в меню загрузки.

Это может быть нежелательным в определенных обстоятельствах. В зависимости от версии GRUB, изменения необходимо внести либо в файл /etc/grub.d/00_header, либо в переменную GRUB_RECORDFAIL_TIMEOUT в файле /etc/default/grub.

Сначала выполните поиск в файле /etc/grub.d/00_header.

```
if [ ${recordfail} = 1 ]; then  
    set timeout=-1  
else  
    set timeout=${GRUB_TIMEOUT}  
fi
```

Если это найдено (точное совпадение, или чуть-чуть изменённое), то просто добавьте комментарии:

```
#if [ \${recordfail} = 1 ]; then  
#    set timeout=-1  
#else  
#    set timeout=${GRUB_TIMEOUT}  
#fi
```

Также строчка может иметь такой вид:

```
if [ "\${recordfail}" = 1 ]; then  
    set timeout=${GRUB_RECORDFAIL_TIMEOUT:-1}  
else  
    set timeout=${2}  
fi
```

Затем отредактируйте файл /etc/default/grub и добавьте строку:

```
GRUB_RECORDFAIL_TIMEOUT=7
```

Для таймаута в 7 секунд. Потом выполните обновление конфигурации grub:

```
update-grub
```

Теперь GRUB2 будет всегда использовать установленное время ожидания.

Ручной сброс значения recordfail

Значение для grubenv можно сбросить вручную. Для этого нужно ввести в консоли grub следующее:

```
grub-editenv set recordfail=0
```

Или удалить

```
grub-editenv unset recordfail
```

См. также

- [x86 MultiBoot FAQ](#)

Примечания

1. <http://ubuntologia.ru/grub2-recovery#copy-files>

Ссылки

- [Официальный сайт \(http://www.gnu.org/software/grub/\)](http://www.gnu.org/software/grub/)

Источник — https://ru.wikibooks.org/w/index.php?title=Grub_2&oldid=223406

Эта страница в последний раз была отредактирована 21 ноября 2022 в 20:11.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike, в отдельных случаях могут действовать дополнительные условия. Подробнее см. Условия использования.