

# LINUX

## ЛЮДИ MONO

Нил Борнштейн и Эд Дамбилл о светлом будущем **с.38**



ЧИТАЮТ ВСЕ ПОЛЬЗОВАТЕЛИ LINUX

## FORMAT

# СНЯЛ ► ПОПРАВИЛ ► ВЫВЕЛ ► ОТПРАВИЛ!

Все инструменты, программы и хитрости для профессиональной фотосъемки – с Linux! **с.46**



## ВЫБИРАЙТЕ СВОБОДУ

Откройте для себя мощь Syllable! Нет, она действительно недурна... **с.54**

## ПОЧЕМУ WEB 2.0

НЕ ОЧЕРЕДНОЙ ЛИ ЭТО

# СЕТЕВОЙ МИФ

И КАК С НИМ ЖИТЬ? **с.42**

## ДЛЯ ДЕЛА И ДЛЯ ДУШИ

Модуль ядра – своими руками .....	<b>с.68</b>
make menconfig: готовые рецепты .....	<b>с.72</b>
Сортируем, фильтруем, отправляем .....	<b>с.76</b>
Игры света и тени – в Gimp .....	<b>с.80</b>
Динамический текст в Inkscape .....	<b>с.84</b>
Истинное лицо консольных утилит! .....	<b>с.88</b>
Запрягаем MySQL 5.0 в PHP .....	<b>с.92</b>
Знакомство с Qt: первые шаги .....	<b>с.94</b>

«При первой загрузке вам, возможно, придется подбирать с пола свою отвалившуюся челюсть – Syllable потрясающе быстра.»

Майк Сондерс беспристрастен

# КОНФИГУРАЦИЯ ЯДРА: ГОТОВЫЕ РЕЦЕПТЫ

Потерялись в `make menuconfig`? **Михаил Емельченков** научит, как оптимизировать ядро под ваш процессор, настроить USB и FireWire-устройства, Bluetooth, программный RAID...



Ядро — центральный компонент любой операционной системы. Ядро управляет памятью, процессами, внешними и внутренними устройствами, сетевыми протоколами и многим другим. Ядро Linux было создано Линусом Торвалдсом в 1991 году и с тех пор успешно развивается, дополняясь поддержкой нового оборудования, новых технологий и исправлениями ошибок.

На сегодняшний день Linux — монолитное ядро с поддержкой модулей. Драйверы устройств и расширения ядра могут подгружаться и выгружаться непосредственно во время работы системы без необходимости её перезагрузки.

От того, как настроено ядро, зависит функциональность системы. В этой статье мы рассмотрим, чего можно достичь с помощью некоторых опций ядра. Всё нижеказанное справедливо для последних версий ядер ветки 2.6.

## Способы настройки ядра

Для настройки ядра переключимся в режим суперпользователя и перейдём в директорию, содержащую его исходные тексты:

```
$ su -
# cd /usr/src/linux
```

Существует пять способов конфигурирования ядра:

### 1) # make config

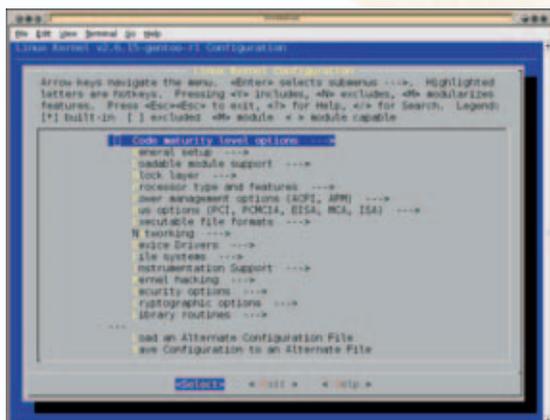
Самый примитивный и неудобный способ. Вам будут задаваться вопросы относительно настроек ядра строка за строкой, раздел за разделом. Отвечать нужно будет «Y», «N» или «M», то есть «Да», «нет» или «модуль».

### 2) # make oldconfig

В новые релизы ядра часто включаются новые опции. `make oldconfig` похож на `make config`; он будет использовать ответы из прежнего файла конфигурации, но будет задавать вам вопросы только о вновь появившихся опциях.

### 3) # make menuconfig

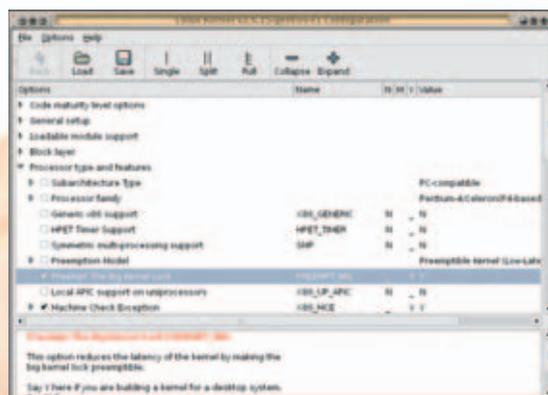
Конфигурация ядра с помощью псевдографической оболочки (интерфейс на базе `ncurses`). Наиболее широко используемый вариант при работе в консоли. Опции разделены иерархически на разделы и подразделы. Это самый универсальный способ настройки.



**Make menuconfig** предоставляет псевдографический интерфейс.

### 4) # make gconfig

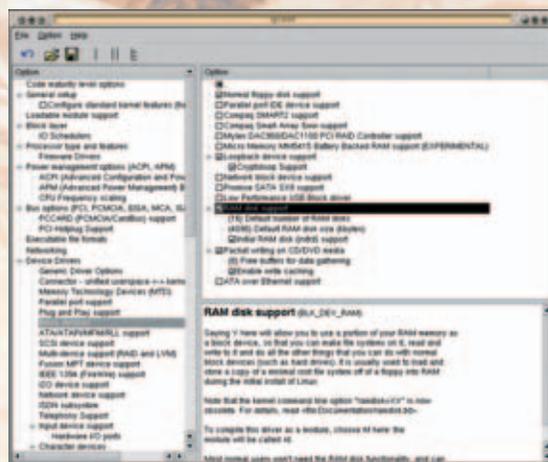
Графический интерфейс конфигурации ядра на базе библиотеки GTK+.



**Make gconfig** использует GTK+ и, наверное, не нравится Линусу Торвалдсу.

### 5) # make xconfig

Графический интерфейс конфигурации ядра на базе библиотеки Qt. Наиболее удобный из всех вышеперечисленных, т.к. отображает разделы и подразделы в виде дерева, а также показывает справку по каждому пункту.



**Make xconfig** собственной персоной. Как сказал Л. Торвалдс: «Просто используйте KDE».

## Оптимизация

Каждый пользователь мечтает о высокопроизводительной системе, но, к сожалению, не каждый знает, как достичь высокой производительности. Ответ на этот вопрос заключается не только в покупке самого современного аппаратного обеспечения, но и в оптимизации существующего ПО. Одним из способов такой оптимизации может служить настройка ядра Linux.

В разделе Processor type and features -> Processor family выберите тип вашего процессора. Каждый из типов включает в себя специфици-

ческие инструкции, применение которых, соответственно, ускоряет работу. Но помните, что если вы неверно укажете тип процессора, ваш компьютер будет работать медленнее, если вообще загрузится. В зависимости от того, является ли ваш компьютер сервером или рабочей станцией, нужно выбрать требуемую опцию в разделе **Processor type and features** -> **Preemption Model** (модель вытеснения задач). Опция **No Forced Preemption** будет соответствовать серверному варианту, а **Voluntary Kernel Preemption** и **Preemptible Kernel** – настольным, причём последний позволяет работать приложениям более «гладко» даже тогда, когда система находится под высокой загрузкой. Также для более «гладкой» работы можно включить опцию **Preempt The Big Kernel Lock**.

В семействе процессоров Intel Pentium Pro, Pentium II и выше используются специальные регистры – **MTTR** (**Memory Type Range Registers**). Эти регистры используются для доступа процессора к различным диапазонам памяти. Включение этой опции позволит существенно повысить производительность системы, особенно если вы используете видеокарту PCI или AGP.

Про следующий, не менее важный параметр, следует вспомнить, когда вы решите увеличить оперативную память. Если у вас менее 1 Гб памяти, выберите опцию **off** в разделе **Processor type and features** -> **High Memory Support**. Владельцам 1 – 4Гб RAM подойдет 4 Gb, а самым счастливым – «64 Gb».

В случае с многопроцессорной системой (а также с системой, процессор которой поддерживает Hyper Treading) следует включить опцию **Symmetric Multi-Processing Support**. После включения данной опции станет доступна опция **Maximum number of CPUs** (2 – 255). Укажите здесь реальное количество процессоров. На производительность она не влияет, но позволит сэкономить несколько килобайт на размере ядра. Также следует включить опцию **Enhanced Real Time Clock Support** раздела **Device Drivers** -> **Character devices**. Учтите, что в SMP-режиме не будет работать управление питанием «Advanced Power Management (APM)».

## Программный RAID

Чем программный RAID лучше аппаратного? Аппаратный RAID, как правило, строится на недешёвых контроллерах и SCSI-дисках, дешёвые контроллеры для дисков IDE (и SATA) часто оказываются псевдоаппаратными контроллерами, т.е. большинство их функций переключается на Windows-драйвер. Поэтому в случае невозможности использования аппаратного контроллера можно организовать программный RAID.

Для его поддержки необходимо включить опции ядра в разделе **Device Drivers** -> **Multi-device support (RAID and LVM)** -> **Multiple devices driver support (RAID and LVM)** -> **RAID support**. Здесь нужно выбрать опции в зависимости от потребностей (RAID-0, RAID-1, RAID-10 и т.д.). Одна из наиболее популярных утилит для работы с RAID называется **mdadm**. Более подробно о ней, а также вообще о программном RAID, можно узнать на <http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>.

Драйверы аппаратных RAID-контроллеров находятся в разделах **Device Drivers** -> **Block devices**, **Device Drivers** -> **ATA/ATAPI/MFM/RLL support**, **Device Drivers** -> **SCSI device support** -> **SCSI low-level drivers**, **Device Drivers** -> **SCSI device support** -> **SCSI low-level drivers** -> **Serial ATA (SATA) support**.

## USB

За поддержку USB отвечает раздел «**Device Drivers**» -> «**USB support**». В подразделе «**Support for Host-side USB**» производится основная конфигурация USB, обычно тут следует выбирать «**EHCI HCD (USB 2.0) support**» для поддержки USB 2.0 и «**UHCI HCD (most Intel and VIA) support**» для поддержки USB 1.1.

Поддержка USB-принтеров включается через «**USB Printer support**». После компилирования ядра с этой опцией и установки драйверов принтера (при необходимости) с принтером можно работать как обычно, например, с помощью CUPS.

Особый интерес представляют так называемые «**USB Storage**» – всевозможные накопители, подключаемые через USB. Одной из наиболее

полезных разновидностей таких накопителей являются флеш-карты, например, карты SD, используемые в фотоаппаратах, плеерах, принтерах и прочей микропроцессорной технике. Считывание и записывание данных с таких карт происходит с помощью адаптера карт памяти (кард-ридера).

Для включения поддержки «**USB Storage**» отметьте опцию **SCSI disk support** раздела **Device Drivers** -> **SCSI device support**, а также опцию **USB Mass Storage support** из раздела **Device Drivers** -> **USB support**. Если у вас какой-то специфичный кард-ридер, здесь можно найти набор подопций для конкретных моделей, но в большинстве случаев должно работать и так.

Если с опцией **USB Mass Storage support** всё понятно, то необходимость включения **SCSI disk support** требует некоторой аргументации. Так вот, это делается потому, что взаимодействие с USB-накопителем происходит посредством эмуляции SCSI, а соответствующие дисковые разделы имеют названия **/dev/sda1**, **/dev/sdb1** и т.д.

Работа с устройствами происходит обычным образом. Сначала создаётся точка монтирования, затем монтируется само устройство:

```
# mkdir /mnt/flash
# mount /dev/sda1 /mnt/flash
# ls /mnt/flash
# umount /mnt/flash
```

Также существует альтернативный драйвер для USB-накопителей. Называется он **Low Performance USB Block driver** и находится в разделе **Device Drivers** -> **Block devices**. При его использовании разделы именуются не **/dev/sda1**, **/dev/sdb1** и т. д., а **/dev/uba1**, **/dev/ubb1** и т. д. Это низкопроизводительный драйвер. Он медленнее «**USB Mass Storage support**», но более надёжный. Этот драйвер работает в обход эмуляции SCSI. Благодаря тому, что он реализует только самые необходимые для работы функции, с его помощью можно заставить работать более широкий спектр устройств. Поэтому, если ваше устройство не работает с «**USB Mass Storage support**», попробуйте этот драйвер.

## FireWire

FireWire (она же IEEE 1394, она же i.Link) – шина, по которой происходит обмен данными внешних устройств и компьютера со скоростью до 400 Мбит/с. Наиболее распространённое её применение – это передача видео с цифровой (DV) видеокамеры и организация сети Ethernet.

Поддержка этой шины включается в разделе **Device Drivers** -> **IEEE 1394 (FireWire) support** -> **IEEE 1394 (FireWire) support**. Большинство карт соответствует спецификации OHCI-1394, поэтому включите опцию «**OHCI-1394 support**».

Для работы с DV-камерами нужно также включить опции «**OHCI-1394 Video support**», «**OHCI-1394 I/O support**» и «**Raw IEEE1394 I/O support**». После этого можно будет воспользоваться такими программами для захвата и редактирования видео, как Kino (<http://www.kinodv.org>), CineLerra (<http://heroinewarrior.com/cinelerra.php3>), MainActor (<http://www.mainconcept.com/mainactor.shtml>) и другими. Не забудьте сделать деинтерлейсинг изображения, полученного с DV-камеры (что такое деинтерлейсинг, можно прочитать в энциклопедии – <http://en.wikipedia.org/wiki/Deinterlacing>).

С помощью FireWire можно также организовать сеть Ethernet. Для этого нужно включить опцию **Ethernet over 1394**. В таком случае можно будет соединить две Linux-станции посредством FireWire. Для работы с системами Windows XP и Mac OS X необходимо также включить опцию **Build in extra config rom entries for certain functionality** -> **IP-1394 Entry (NEW)**. По стандарту длина кабеля не должна превышать 4.5 м. С помощью удлинителей можно довести длину кабеля до 10 м, но это уже нестандартное решение, поэтому с ним возможны проблемы. Хотя FireWire и выигрывает в скорости у стандарта 100BaseT (100 Мбит/с, длина кабеля до 100 метров), он проигрывает в расстоянии, а с появлением Gigabit Ethernet соединение посредством FireWire проигрывает во всём. Одним из наиболее оправданных случаев применения Ethernet-сети на базе FireWire является соединение ноутбука с настольным компьютером, т.к. современные ноутбуки обычно комплектуются портом FireWire.

## Работа с сотовым телефоном через Bluetooth

Bluetooth – спецификация, которая позволяет использовать вместе такие устройства, как компьютер, КПК или телефон. Связь осуществляется на короткие расстояния (10 м, 100 м) по радиоканалу.

Большинство Bluetooth-адаптеров подключаются по USB, поэтому удостоверьтесь, что вы собрали ядро с поддержкой USB.

Поддержка Bluetooth включается в разделе **Networking -> Bluetooth subsystem support**. В этом разделе следует включить опцию **L2CAP protocol support**, требуемую для работы большинства Bluetooth-приложений. Также потребуется **SCO links support**, если вы планируете использовать Bluetooth для передачи голоса, например, через гарнитуру. Опции **RFCOMM protocol support** и **RFCOMM TTY support** нужны для поддержки таких способов применения Bluetooth, как Dialup и OBEX (OBEX – Object EXchange – протокол для передачи объектов: контактов, файлов и т.д.). Включение **BNEP protocol support** понадобится для поддержки так называемых Bluetooth PAN (Personal Area Network) для объединения нескольких Bluetooth-адаптеров в сеть. Ну и ещё одна опция – **HIDP protocol support** – требуется для поддержки HID (Human Interface Devices), то есть устройств, которые предназначены для управления компьютером, таких как клавиатура, мышь и т.д.

Перейдем к подразделу **Bluetooth device drivers**. Наиболее распространёнными являются USB-адаптеры, поэтому выбираем опцию **HCI USB driver** и подопцию **SCO (voice) support**. Если же у вас адаптер с интерфейсом, отличным от USB, выбирайте одну из подходящих опций данного подраздела. Не забывайте, что для поддержки USB-устройств должны быть активированы соответствующие опции в разделе **Device Drivers -> USB support**.

Теперь ядро поддерживает Bluetooth. Но для работы приложений необходимо поставить официальный стек Bluetooth – BlueZ (<http://www.bluez.org>). Соответствующие пакеты в вашем дистрибутиве могут называться **bluez-libs**, **bluez-utils** или подобным образом.

Для работы с файлами на сотовом телефоне через Bluetooth в консоли можно воспользоваться программой ObexFTP (<http://trig.net/obex>). Для работы с сотовым телефоном в графической среде – GNOME Phone Manager (<http://usefulinc.com/software/phonemgr/>), KMobiletools (<http://kmobiletools.berlios.de>) и другими.

## NFS и SMB

Network File System (NFS) в мире Linux – почти то же самое, что папки общего доступа Windows (по крайней мере, по области применения). Главное отличие состоит в том, что соединяются не Windows-станции, а компьютеры Unix.

Для включения поддержки NFS в ядре отметьте опции **NFS file system support**, **Provide NFSv3 client support**, **NFS server support**, **Provide NFSv3 server support** в разделе **File systems -> Network File Systems**.

Для работы с NFS применяются стандартные утилиты. Обычно они содержатся в пакете **nfs-utils** или подобном. Для работы NFS также необходимо установить и запустить сервис **portmap**. Теперь можно монтировать директорию командой вида:

```
# mount x.x.x.x:/directory /mount_directory
```

Где «**x.x.x.x**» – IP-адрес или DNS-имя NFS-сервера, «**/directory**» – экспортированная директория, которую вы хотите подключить, «**/mount\_directory**» – локальная точка монтирования для экспортируемой директории.

При наличии в системе демона **nfsmount**, можно также монтировать сетевые директории при загрузке. Для этого следует добавить их в **/etc/fstab**:

```
x.x.x.x:/directory/mount_directory nfs rw 0 0
```

Для работы с NFS в качестве сервера, отредактируйте файл **/etc/exports**:

```
/directory x.x.x.x:/y.y.y.y(async,no_subtree_check,rw)
```

В данном случае директория будет доступна всей подсети «**x.x.x.x**» маски «**y.y.y.y**». Более подробно о способах задания экспортируемых директорий читайте в «**man exports**».

Для работы в качестве NFS-сервера не забудьте запустить демон **nfs**.

SMB (Server Message Block) – протокол, предназначенный для обеспечения общего доступа к файлам и принтерам в локальных сетях. Для включения поддержки SMB в ядре отметьте опцию **SMB file system support (to mount Windows shares etc.)** раздела **File systems -> Network File Systems**. Также отметьте опцию **CIFS support (advanced network filesystem for Samba, Window and other CIFS compliant servers)** того же раздела, если вы планируете подключаться к серверам на базе Windows 2000, Windows 2003, Windows XP и некоторым другим.

Далее необходимо установить программы для работы с SMB. Обычно этот пакет называют коротко – **samba**.

В средах KDE, Gnome, XFCE пользоваться SMB можно с помощью штатных средств, но существует также возможность работы с SMB в консоли. Для монтирования общедоступной папки служит команда **smbmount**:

```
# smbmount //<windows machine name>/<shared folder> /<mountpoint> -o username=<user>,password=<pass>,uid=1000,umask=000
```

Здесь «**windows machine name**» – имя файл-сервера, «**shared folder**» – папка на сервере, «**mountpoint**» – точка монтирования, «**user**» и «**pass**» – имя пользователя и пароль соответственно.

Отмонтирование выполняется командой **smbumount**:

```
# smbumount <mountpoint>
```

Посмотреть список общедоступных папок можно с помощью команды

```
# smbclient -L <windows machine name> -U%
```

Также полезна утилита **smbtree**. Она выполняет функции «Сетевого окружения» Windows, то есть показывает дерево всех доменов, серверов в этих доменах и их сетевых ресурсов.

Конфигурация сервера выполняется с помощью файла **/etc/samba/smb.conf**. По умолчанию он пустой, но за основу можно взять файл **/etc/samba/smb.conf.example**. В этом файле даны примеры настройки общедоступных папок и принтеров.

Кроме NFS и SMB существует поддержка ещё нескольких сетевых файловых систем, таких, как NCP (для монтирования томов NetWare), Coda (сетевая файловая система, похожая на NFS, но обладающая более широкими возможностями), экспериментальная поддержка AFS и Plan 9 Resource Sharing.

## TUN/TAP

Драйвер TUN/TAP используется для создания туннельных интерфейсов. Иными словами, с помощью него и сопутствующих программ можно организовать простейшую виртуальную частную сеть, то есть VPN-соединение.

Включается драйвер опцией **Universal TUN/TAP device driver support** раздела **Device Drivers -> Network device support**.

VPN-соединение организуется с помощью таких программ, как VTun (<http://vtun.sourceforge.net/>), OpenVPN (<http://openvpn.net/>).

Драйвер TUN/TAP также используется для организации сети между виртуальной машиной и рабочей средой в таких эмуляторах ПК, как Bochs (<http://bochs.sourceforge.net/>) и QEMU (<http://fabrice.bellard.free.fr/qemu/>).

## Шифрование

В ядре Linux есть раздел **Cryptographic options**. Это реализация Cryptographic API – стандартного интерфейса для шифрования, хеширования и даже сжатия данных. Пользователю предоставляется на выбор множество алгоритмов: здесь есть и SHA256 digest algorithm, и AES cipher algorithms, и многие другие.

Cryptographic API используется модулями ядра, например, драйверами Wi-Fi карт, а также для поддержки шифрования с помощью **device-mapper**.

Для шифрования дисков с помощью **device-mapper** можно использовать программу **Dm-crypt** (<http://www.saout.de/misc/dm-crypt/>).

Cryptographic API также используется для шифрования сетевого трафика, например, в **IPSec**. IPSec-туннель можно реализовать с помощью **Openswan** (<http://www.openswan.org/>). **IPSec**

### Пишите нам

Ядро Linux неисчерпаемо, как и атом. Какие из интересных вам тем остались за рамками этой статьи? Пишите нам: [letters@linuxformat.ru](mailto:letters@linuxformat.ru) и, возможно, мы продолжим этот разговор на страницах журнала.