

Kernel (Русский)

Ссылки по теме

- [Модули ядра](#)
- [Compile kernel module](#)
- [Kernel Panics](#)
- [Linux-ck \(Русский\)](#)
- [sysctl](#)

Состояние перевода: На этой странице представлен перевод статьи [Kernels](#). Дата последней синхронизации: 29 марта 2016. Вы можете [помочь](#) синхронизировать перевод, если в английской версии произошли [изменения](#).

Из [Wikipedia](#):

Ядро — центральная часть операционной системы (ОС), обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как процессорное время, память, внешнее аппаратное обеспечение, внешнее устройство ввода и вывода информации. Также обычно ядро предоставляет сервисы файловой системы и сетевых протоколов.

Существуют различные альтернативные доступные ядра Arch Linux в дополнение к основным Linux ядрам. В этой статье перечислены некоторые из вариантов имеющихся в репозиториях, с кратким описанием каждого из них. Существует также описание патчей, которые могут быть применены к ядру. Статья заканчивается обзором пользовательской компиляции ядра со ссылками на различные методы.

Contents

[hide]

- [1Предварительно скомпилированные ядра](#)
 - [1.1Официальные пакеты](#)
 - [1.2Неофициальные пользовательские репозитории с готовыми, собранными ядрами Linux](#)
 - [1.3AUR пакеты](#)
- [2Патчи и наборы патчей \(патчсеты\)](#)
 - [2.1Как установить](#)
 - [2.2Основные патчи и патчсеты](#)
 - [2.2.1-ck](#)
 - [2.2.2-rt](#)
 - [2.2.3-bld](#)
 - [2.2.4-grsecurity](#)
 - [2.2.5Tiny-Патчи](#)
 - [2.2.6-pf](#)
 - [2.3Индивидуальные патчи](#)
 - [2.3.1Reiser4](#)
 - [2.3.2fbsplash](#)
- [3Компиляция ядра](#)
 - [3.1Используя Arch Build System \(ABS\)](#)
 - [3.2Традиционный метод](#)
 - [3.3Проприетарный NVIDIA драйвер](#)
- [4Смотрите также](#)
 - [4.1Тесты и обзоры](#)

Предварительно скомпилированные ядра

Официальные пакеты

[linux](#)

Linux ядро и модули из репозитория [core]. Ванильное ядро с [некоторыми патчами](#).

[linux-lts](#)

Версия ядра Linux и модулей с долгосрочной поддержкой (LTS - Long Term Support) из репозитория [core].

[linux-grsec](#)^{[[ссылка недействительна](#): replaced by [linux-hardened](#)]}

Linux ядро и модули с [Grsecurity Patchset](#) и PaX патчами для повышения безопасности.

[linux-zen](#)

[ZEN-ядро](#) результат совместной работы проекта Zen-kernel.

Для создания этого ядра берётся свежее стабильное официальное ядро Linux. И вносятся изменения проектом Zen-kernel (планировщик CPU BFS, BFQ-планировщик (I/O) ввода-вывода, Aufs, Unionfs, Reiser4, TuxOnIce, PNC и многие другие), которые улучшают отзывчивость и производительность системы.

Неофициальные пользовательские репозитории с готовыми, собранными ядрами Linux

Рекомендуется посмотреть [этот раздел](#)

AUR пакеты

[linux-aufs friendly](#)^{AUR[[ссылка недействительна](#): package not found]}

AUFS-совместимое ядро Linux и модули, полезно при использовании [Docker](#)

[linux-apparmor](#)^{AUR}

Linux ядро с включенными возможностями [AppArmor](#)

[linux-bfs](#)^{AUR[[ссылка недействительна](#): сохранено в [aur-mirror](#)]}

Ядро Linux и модули с [Brain Fuck Scheduler](#) (BFS) - созданное Коном Коливасом (Con Kolivas) для PC с меньшим, чем 4096 ядрами, и BFQ I/O планировщиком в качестве дополнительного

[linux-chromebook](#)^{AUR[[ссылка недействительна](#): сохранено в [aur-mirror](#)]}

Ядро Linux с добавлением аппаратной поддержки [chromebook](#)

[Linux-ck](#)^{AUR}

Ядро Linux, доступное в [AUR](#), которое позволяет пользователям запускать [ядро с набором патчей Кона Коливаса](#)^{[[устаревшая ссылка](#) 2015-08-29]}, включая "Brain Fuck Scheduler" (BFS)

Эти патчи предназначены для улучшения отклика системы с особым упором на PC, подходят для любого PC

[linux-eee-ck](#)^{AUR[[ссылка недействительна](#): сохранено в [aur-mirror](#)]}

Ядро Linux и модули для Asus Eee PC 701, собранные с ck1-патчами Кона Коливаса (Con Kolivas)

[linux-fbcondecor](#)^{AUR[[ссылка недействительна](#): package not found]}

Ядро Linux и модули с [fbcondecor поддержкой](#).

[linux-git](#)^{AUR}

Ядро Linux и модули собранное с [Linus Torvalds' Git репозитория](#).

[linux-ice](#)^{AUR[[ссылка недействительна](#): package not found]}

Ядро Linux и модули с gentoo-sources патчами и [TuxOnIce](#) поддержкой.

[linux-libre](#)^{AUR}, [linux-libre-lts](#)^{AUR[[ссылка недействительна](#): package not found]}, [linux-libre-grsec](#)^{AUR[[ссылка недействительна](#): package not found]}, [linux-libre-rt](#)^{AUR[[ссылка недействительна](#): package not found]}, [linux-libre-xen](#)^{AUR[[ссылка недействительна](#): package not found]}

Ядро Linux с "binary blobs".

[linux-lqx](#)^{AUR}

[Liquorix](#) - ядро, построенное с использованием Debian-конфигурации и ZEN-патчей. Предназначено для рабочего стола, мультимедийных, игровых и рабочих станций, часто используется в качестве замены ядра Debian Linux

Damentz, - сопровождающий набор патчей Liquorix, является также разработчиком для набора патчей ZEN

[linux-lts34](#)^{AUR[[ссылка недействительна](#): сохранено в [aur-mirror](#)]}

Ядро Linux 3.4 с долгосрочной поддержкой (LTS - Long Term Support).

[linux-lts310](#)^{AUR}[\[ссылка недействительна: package not found\]](#)

Ядро Linux 3.10 с долгосрочной поддержкой (LTS - Long Term Support).

[linux-lts312](#)^{AUR}[\[ссылка недействительна: package not found\]](#)

Ядро Linux 3.12 с долгосрочной поддержкой (LTS - Long Term Support).

[linux-mainline](#)^{AUR}

Ядро Linux-mainline.

[linux-mptcp](#)^{AUR}

Ядро Linux и модули с поддержкой [Multipath TCP](#)

[kernel-netbook](#)^{AUR}[\[ссылка недействительна: сохранено в aur-mirror\]](#)

Статичное ядро для нетбуков с Intel Atom N270/N280/N450/N550, таких как Eee PC, с добавлением внешней прошивки ([broadcom-wl](#)) и наборами патчей (BFS + TuxOnIce + BFQ опцией); только для Intel GPU

[linux-pax](#)^{AUR}[\[ссылка недействительна: package not found\]](#)

Ядро Linux и модули с [PaX](#) патчами для повышенной безопасности.

[linux-pf](#)^{AUR}

Ядро Linux и модули с pf-патчами ядра [-ck патчи (BFS included), TuxOnIce, BFQ] и aufs3.

Это ядро также доступно (уже собранное) [из неофициального пользовательского репозитория](#) [\[broken link: invalid section\]](#).

[linux-tresor](#)^{AUR}[\[ссылка недействительна: package not found\]](#)/[linux-lts-tresor](#)^{AUR}[\[ссылка недействительна: package not found\]](#)

Текущее/LTS Linux ядро и модули со встроенным [патчем TRESOR](#)

[linux-vfio](#)^{AUR}/[linux-vfio-lts](#)^{AUR}

Ядро Linux и несколько патчей написанных Алексом Уильямсон (Alex Williamson) (переопределяющих acs и i915) предоставляющих возможность сделать PCI Passthrough с KVM на некоторых машинах.

Используется для хранения ключей шифрования AES не в ОЗУ, а в регистрах CPU, [подробнее тут](#).

Патчи и наборы патчей (патчсеты)

Есть много причин, чтобы пропатчить ядро, основные из них для работы или для поддержки не-магистральных функций, таких как поддержка файловой системы Reiser4. Другие причины могут включать в себя забаву и посмотреть, как это делается и какие улучшения при этом появляются.

Однако, важно отметить, что лучшим способом увеличить скорость работы вашей системы будет сконфигурированное ядро именно для вашей системы, особенно под конкретную архитектуру и тип процессора. По этой причине использовать заранее упакованные версии пользовательских ядер с общими настройками архитектуры не рекомендуется, или не стоит. Еще одним преимуществом является то, что вы можете уменьшить размер вашего ядра (и, следовательно время сборки). Для этого не включая поддержку тех модулей, которыми вы не пользуетесь. Например вы можете убрать поддержку bluetooth, video4linux, 1000Mbit ethernet и т.д. Т.е. убрать тот ненужный для вас функционал и/или поддержку того оборудования которого у вас нет. Хотя эта статья не о конфигурации ядра, тем не менее рекомендуется в качестве первого шага, основы, чтобы понять какой набор патчей использовать.

Конфигурационные файлы для пакетов ядра Arch можно использовать в качестве отправной точки. Они находятся в Arch исходных файлов пакета, например [\[1\]](#) связано с [linux](#). Файл конфигурации вашего текущего ядра также всегда доступны в файловой системе в /proc/config.gz.

Как установить

Процесс установки пользовательских пакетов ядра опирается на систему сборки Arch (ABS). Если вы не собирали какие-либо пользовательские пакеты самостоятельно, вы

можете обратиться к следующим статьям: [Arch Build System \(Русский\)](#) и [Creating packages \(Русский\)](#).

Если вы не патчили или настраивали ядро ранее, то это не так сложно, и есть много PKGBUILD'ов на форуме для отдельных пакетов. Тем не менее, мы советуем вам начать с нуля, с некоторым исследованием преимуществ каждого набора патчей, а не просто произвольной выборки из всего множества. Таким образом, вы узнаете гораздо больше о том, что вы делаете, а не просто выберите ядро при запуске, а потом будите гадать, что же там изменилось.

Смотрите [#Compilation](#) [broken link: invalid section].

Примечание: Не забудьте изменить параметры загрузки в загрузчике [GRUB](#), чтобы использовать новое ядро.

Основные патчи и патчсеты

Прежде всего важно отметить, что патчи и патчсеты разработаны различными людьми. Некоторые из этих людей на самом деле участвуют в производстве ядра Linux. Но есть и любители, которые могут поменять функции ядра надежности и стабильности в лучшую/худшую сторону.

Стоит также отметить, что некоторые патчи строятся на старых версиях ядра и (или) патчах (которые могут или не могут быть отражены в названии патча). Патчсеты (и обновления ядра) часто и зачастую не идут в ногу со временем. Так что не стоит сходить с ума, если это не ваше хобби.

Вы можете воспользоваться поиском Google для поиска других видов патчей.

Примечание: Этот раздел **только для ознакомления** - никаких гарантий о надёжности или стабильности на этой странице.

-ck

[Linux-ck](#) содержит патчи, предназначенные для улучшения отклика системы с особым упором на рабочем столе, подходят для любой нагрузки. Патчи создаются и поддерживаются Коном Коливасом (Con Kolivas). Его сайт <http://users.on.net/~ckolivas/kernel/>. Кон поддерживает полный набор патчей, но также обеспечивает одиночные патчи, так что вы можете добавить только те, которые Вы предпочитаете.

-ck можно найти на <http://ck.kolivas.org/patches/4.0/>

-rt

Это набор патчей поддерживается небольшой группой разработчиков ядра, во главе с Инго Молнар (Ingo Molnar). Позволяет превратить обычный Linux в ОС реального времени. Главное применение такой системы – промышленные и встроенные системы, но на обычном компьютере она тоже может быть интересна. Например тем, кто часто занимается обработкой звука и видео или постоянно грузит систему какими-нибудь ресурсоемкими вычислениями. Также есть положительный эффект от применения этого ядра на highload-серверах.

Патч на <https://www.kernel.org/pub/linux/kernel/projects/rt/>

-bld

Важно: Этот патч находится в разработке.

[BLD](#) (Barbershop Load Distribution) лучше всего описывается как O(1) техника сортировки процессов. Это реализация альтернативного алгоритма планирования задач. BLD ограничивается решением задачи по корректному распределению нагрузки путем отслеживания не всех привязанных к CPU очередей, а только наиболее и наименее загруженных очередей выполнения (rq, runqueue). BLD не пытается балансировать нагрузку на систему в контексте отслеживания бездействующих idle-процессов, а акцентирует внимание на распределении всей нагрузки между имеющимися процессорами наиболее простым путём с минимальным числом усложнений. Главным достоинством BLD является сам подход, показывающий что достаточно простыми методами можно добиться равномерного распределения нагрузки, не заботясь особенно о

том сколько CPU используется в системе и соответственно без лавинообразного падения производительности на накладные расходы при увеличении числа CPU (BLD обеспечивает уровень производительности $O(i)$, где i - число CPU).

По материалам [этой новости](#)

-grsecurity

[Grsecurity](#) Из [2] Это проект для Linux, который включает в себя некоторые улучшения связанные с безопасностью, включая принудительный контроль доступа, рандомизацию ключевых локальных и сетевых информативных данных, ограничения /proc и chroot() jail, контроль сетевых сокетов, контроль возможностей, и добавочные функции аудита. Типичной областью применения являются web-сервера и системы, которые принимают удалённые соединения из сомнительных мест, такие как сервера, которые обеспечивают shell-доступ для пользователей. Патч grsecurity выпущен под GPL, является свободным ПО и включает в себя набор патчей PaX. Создатель и ведущий разработчик grsecurity — Brad Spengler aka. Spender.

Grsecurity патчи можно посмотреть на <https://grsecurity.net>

Tiny-Патчи

Цель [Linux Tiny](#) использование минимального дискового пространства и оперативной памяти, а также облегчение труда маломощных компьютеров. Целевые пользователи являются разработчиками встраиваемых систем и пользователями маломощных или устаревших машин, таких как 386/

Патч-релизы против господствующего Linux ядра были прекращены. Разработчики решили сосредоточиться на нескольких патчах и попытках поставить их в официальное ядро.

-pf

[linux-pf](#)^{AUR} Это набор патчей, поверх официального ядра, обеспечивающих повышенную отзывчивость системы, предоставляющих альтернативную подсистему гибернации (более быстрая, по сравнению с основной), а также уменьшают использование памяти с помощью техники объединения одинаковых страниц. Набор патчей включает в себя: планировщик процессов BFS от Кона Коливаса (Con Kolivas) с дополнительными исправлениями от Альфреда Чена (Alfred Chen), планировщик ввода-вывода BFQ Паоло Валенте (Paolo Valente), Арианны Аванзини (Arianna Avanzini) и Мауро Маринони (Mauro Marinoni), подсистема гибернации TuxOnIce от Найджела Каннингема (Nigel Cunningham), реализация техники слияния одинаковых страниц в памяти UKSM от Най Ся (Nai Xia), патч от Graysky, расширяющий список процессоров для оптимизации ядра компилятором.

Смотрите [linux-pf](#) для большей информации.

Индивидуальные патчи

Эти патчи могут быть просто включены в любую сборку ванильного ядра или включены (возможно, с какой-то крупной тонкой настройки) в другой набор патчей.

Reiser4

[Reiser4](#)

fb splash

[fb splash](#)

Компиляция ядра

Arch Linux предусматривает несколько методов компиляции ядра.

Используя Arch Build System (ABS)

Используя [ABS](#) воспользуетесь высоким качеством существующих [linux PKGBUILD](#) и преимущества [менеджера пакетов Pacman](#). PKGBUILD структурирован таким образом, что вы можете остановить сборку после получения исходного кода, и сконфигурировать ядро.

Смотрите [Kernels/Arch Build System](#).

Традиционный метод

Это загрузка архива с исходным кодом, распаковка, и компиляция. После компиляции доступны два способа установки: традиционный ручной метод, или [Makepkg](#) + [Pacman](#).

Преимуществом установки традиционным методом является то, что вы не привязаны к какому-либо дистрибутиву.

Смотрите [Kernels/Compilation/Traditional](#).

Проприетарный NVIDIA драйвер

Смотрите [NVIDIA#Alternate install: custom kernel](#) [broken link: invalid section] инструкцию по использованию проприетарного драйвера NVIDIA с патченным ядром.

Смотрите также

- [O'Reilly - Linux Kernel in a Nutshell](#) (свободная электронная книга)

Тесты и обзоры

- [Есть ли польза от кастомных ядер](#)
- [Тест планировщика ввода/вывода BFQ](#)
- [бенчмарки BFQ](#)

Categories:

- [Kernel \(Русский\)](#)
- [Lists of software \(Русский\)](#)
- [Русский](#)