

Система Сборки Ядра/Арки

< Ядро

Смотрите основные статьи В разделе ядра.

Система **Arch Build System** может быть использована для построения пользовательского ядра на основе официального <https://www.archlinux.org/packages/?name=linux> пакета linux. Этот метод компиляции позволяет автоматизировать весь процесс и основан на очень хорошо протестированном пакете. Вы можете изменить PKGBUILD, чтобы использовать пользовательскую конфигурацию ядра или добавить дополнительные исправления.

Содержание

[Получение ингредиентов](#)

[Изменение PKGBUILD](#)

[изменение подготовка\(\)](#)

[создание новых контрольных сумм](#)

[компилирование](#)

[Установка](#)

[загрузчик](#)

[обновление](#)

[очистка](#)

[см. также](#)

Получение ингредиентов

Поскольку вы будете использовать [makepkg](#), следуйте рекомендациям, изложенным там в первую очередь. Например, вы не можете запустить makepkg как root/sudo. Поэтому сначала создайте build каталог в домашнем компьютере пользователя.

```
$ компакт ~/
$ mkdir build
$ CD build/
```

[Установите](#) пакет [asp](https://www.archlinux.org/packages/?name=asp) (<https://www.archlinux.org/packages/?name=asp>) и [группу пакетов base-devel](#) (https://www.archlinux.org/groups/x86_64/base-devel/).

Вам нужно чистое ядро, чтобы начать настройку из него. [ABS#получить PKGBUILD источник с помощью Git](#) и несколько других файлов в каталог сборки, запустив:

```
$ asp обновление linux
$ asp экспорт linux
```

На этом этапе дерево каталогов выглядит следующим образом:

```
~ / build / linux/ -+
+--конфигурация
\_ _ PKGBUILD
```

Затем получите любой другой файл, который вам нужен (например, пользовательские файлы конфигурации, патчи и т. д.) из соответствующих источников.

Изменение PKGBUILD

Отредактируйте PKGBUILD и найдите этот pkgbase параметр. Измените это на ваше собственное имя пакета, например:

```
pkgbase=linux-custom
```

Изменение подготовки()

В prepare() функции можно [применить необходимые исправления ядра](#) или изменить конфигурацию сборки ядра.

Если вам нужно изменить несколько параметров конфигурации, вы можете отредактировать файл конфигурации в источнике.

Или вы можете использовать графический инструмент для настройки параметров. Прокомментируйте make olddefconfig в функции prepare() PKGBUILD и добавьте свой любимый инструмент:

PKGBUILD

```
...
msg2 " настройка конфигурации..."
КП../конфигурация .конфиг
#make olddefconfig

сделайте меню CLI nconfig # new для конфигурации
#make меню menuconfig # CLI для настройки
#make xconfig # конфигурация на основе X
#make oldconfig # используя старую конфигурацию из предыдущей версии ядра
# ... или вручную редактировать .конфиг
...
```

Предупреждение: systemd имеет ряд требований к конфигурации ядра для общего использования, для конкретных видов использования (например, UEFI) и для конкретной функциональности systemd (например, bootchart). Несоблюдение этих требований может привести к деградации или непригодности вашей системы. Список необходимых и рекомендуемых конфигураций ядра можно найти в разделе /usr/share/doc/systemd/README . Проверьте их перед компиляцией. Эти требования также меняются с течением времени. Поскольку Arch предполагает, что вы используете официальное ядро, об этих изменениях не будет объявлено. Прежде чем устанавливать новую версию systemd, проверьте примечания к выпуску версии, чтобы убедиться, что ваше текущее пользовательское ядро соответствует любым новым требованиям systemd.

Генерировать новые контрольные суммы

[#Changing prepare 0](#) предлагает возможные изменения `$_srcname/.config`. Поскольку этот путь не является местом, где закончилась загрузка файлов пакета, его контрольная сумма не была проверена `makepkg` (который фактически проверил `$_srcname/../../config`).

Если перед `config` запуском `makepkg` загруженный файл был заменен другим конфигурационным файлом, [установите](#) пакет [pacman-contrib](https://www.archlinux.org/packages/?name=pacman-contrib) (<https://www.archlinux.org/packages/?name=pacman-contrib>). Который будет генерировать новые контрольные суммы, запустив:

```
$ updpkgsums
```

Составление

Теперь вы можете приступить к компиляции ядра с помощью обычной команды `makepkg`

Если вы выбрали интерактивную программу для настройки параметров ядра (например, `menuconfig`), вы должны быть там во время компиляции.

```
$ makepkg -s
```

Этот `-s` параметр будет загружать любые дополнительные зависимости, используемые последними ядрами, такими как `xml` и `docs`.

Примечание:

- Источники ядра [подписаны PGP \(https://www.kernel.org/signature.html#kernel-org-web-of-trust\)](https://www.kernel.org/signature.html#kernel-org-web-of-trust), и `makepkg` попытается проверить их. [Дополнительную информацию смотрите в разделе Проверка подписи Makepkg#](#).
- Компиляция может занять до нескольких часов для завершения в зависимости от производительности оборудования. [Одновременное выполнение заданий компиляции](#) может значительно сократить время компиляции в многоядерных системах.

Установка

Рекомендуется [устанавливать](#) оба пакета вместе, поскольку они могут быть необходимы (например, DKMS.)

```
# pacman -U kernel-headers_package kernel_package
```

Загрузчик

Если вы изменили `pkgbase` для установки вашего нового ядра вместе с ядром по умолчанию, вам нужно будет обновить файл конфигурации загрузчика и добавить новые записи ('default' и 'fallback') для вашего пользовательского ядра и связанных с ним образов `initramfs`.

Обновление

Предполагая, что у одного есть источник ядра `arch`, который они хотят обновить, один метод для этого - <https://git.archlinux.org/linux.git>. в следующем примере предполагается, что верхний исходный каталог ядра находится в `~/build/linux/`.

В общем случае arch устанавливает источник ядра arch с двумя локальными репозиториями git. Один в archlinux-linux/ является локальным голым репозиторием git, указывающим на [git://git.archlinux.org/linux.git](https://git.archlinux.org/linux.git). другой находится в src/ archlinux-linux/, вытягивая из первого репозитория. Возможные локальные исправления и построение, как ожидается, в src / archlinux-linux/.

Для этого примера глава локально установленного голого источника репозитория git в archlinux-linux / первоначально указывал на 4010b622f1d2 Merge branch 'dax-fix-5.3-rc3' of [git://git.kernel.org/pub/scm/linux/kernel/git/nvdimm/nvdimm](https://git.kernel.org/pub/scm/linux/kernel/git/nvdimm/nvdimm) то, что находится где-то между v5.2.5-arch1 и v5.2.6-arch1.

```
$ cd ~ / build / linux / archlinux-linux/
$ git fetch --verbose
```

Можно увидеть, что он получил v5. 2. 7-arch1, который был самым новым тегом archlinux, потому что он печатает, какие новые теги были получены. Если новые теги не были получены, то более новый источник archlinux недоступен.

Теперь источник может быть обновлен, где будет происходить фактическая сборка.

```
$ cd ~ / build / linux / src / archlinux-linux/
$ git checkout master
$ git pull
$ git fetch --tags --verbose
$ git branch --verbose 5.2.7-arch1 v5. 2. 7-arch1
$ git checkout 5.2.7-arch1
```

Вы можете проверить, что вы находитесь на пути с чем-то вроде

```
$ git log --oneline 5.2.7-arch1 --max-count=7
13193bfc03d4 Arch ядро Linux v5. 2. 7-arch1
9475c6772d05 netfilter: nf_tabf676926c7f60les: исправлена загрузка модуля autoload для redir
498d650048f6 iwlmwifi: добавьте поддержку ограничения SAR South Korea
bb7293abdbbc7 iwlmwifi: mvm: отключить TX-AMSDU на старых сетевых картах
f676926c7f60 ZEN: добавление конфигурации для непривилегированного объекта users_clone
5e4e503f4f28 добавить sysctl запретить непривилегированный CLONE_NEWUSER по умолчанию
5697a9d3d55f Linux 5.2.7
```

Это показывает несколько конкретных патчей archlinux между Linux 5.2.7 и Arch Linux kernel v5.2.7-arch1. Важными линиями здесь являются Linux 5.2.7 и Arch Linux kernel v5.2.7-arch1. Очевидно, что в других версиях могут быть другие патчи, поэтому 7 for --max-count может потребовать корректировки. Аналогичным образом, идентификаторы фиксации, такие как f676926c7f60, а также версия ядра, будут отличаться для других версий.

Обновленный PKGBUILD, а также файл конфигурации ядра archlinux, может быть вызван asp командой:

```
$ cd ~ / build / linux/
$ asp обновление linux
$ asp-f экспорт linux
```

Примечание: иногда asp команда не обновляет файлы linux, даже если есть более новый исходный тег archlinux. Возможной причиной является то, что файлы archlinux linux отстают от источника archlinux linux.

Теперь вы должны [Vim # слияние файлов](#) `~/build/linux/linux/*`, расположенных в `~/build/linux/`. Слияние также может выполняться вручную или со [списком приложений#сравнение, diff, слияние](#). Просмотрите [#Changing prepare \(\)](#) и запустите вручную большинство, если не все, команды оболочки `PKGBUILD::prepare ()`.

В этот момент, `makepkg --verifysource` должно получиться. Во время [компиляции#](#), убедитесь, что вы также добавили `--noextract` опцию в `makepkg` команду, так как она должна быть в состоянии построить пакеты, как если бы исходный код был извлечен `makepkg --nobuild`. И вы снова возвращаетесь к [#установке](#).

Уборка

Один из них, вероятно, захочет удалить `~/build/linux/linux/` после слияния. Кроме того, `~/build/linux/src/archlinux` будут накапливаться ветви в виде `5.2.7-arch1`, если более последние обновления будут сделаны таким образом. Они могут быть удалены с помощью

```
$ cd ~ / build / linux / src / archlinux
$ git branch --delete --force --verbose 5.2.7-arch1
```

Смотреть также

- <https://kernel.org/doc/Documentation/kbuild/kconfig.txt> и родительский каталог

Извлечено из "https://wiki.archlinux.org/index.php?title=Kernel/Arch_Build_System&oldid=599883"

Последний раз эта страница редактировалась 2 март 2020, в 21:57.

Контент доступен в соответствии с [лицензией GNU Free Documentation License 1.3 или более поздней](#) версии, если не указано иное.