

# Система Построения Арки

Arch build system—это портоподобная система для сборки и упаковки программного обеспечения из исходного кода. В то время **как [расман](#)** является специализированным инструментом Arch для управления двоичными пакетами (включая пакеты, построенные с помощью ABS), ABS—это набор инструментов для компиляции исходного кода в устанавливаемые `.pkg.tar.xz` пакеты.

*Ports*—это система, используемая \*BSD для автоматизации процесса построения программного обеспечения из исходного кода. Система использует *порт* для загрузки, распаковки, исправления, компиляции и установки данного программного обеспечения. *Порт*—это всего лишь небольшой каталог на компьютере пользователя, названный в честь соответствующего устанавливаемого программного обеспечения, который содержит несколько файлов с инструкциями по сборке и установке программного обеспечения из исходного кода. Это делает установку программного обеспечения таким же простым, как ввод `make` текста или `make install clean` в каталоге порта.

ABS—это аналогичная концепция. Частью ABS является репозиторий SVN и эквивалентный ему репозиторий Git. Репозиторий содержит каталог, соответствующий каждому пакету, доступному в Arch Linux. Каталоги репозитория содержат файл PKGBUILD (а иногда и другие файлы), и не содержат ни исходного кода программного обеспечения, ни двоичного файла. При выпуске **[makepkg](#)** внутри каталога загружаются источники программного обеспечения, программное обеспечение компилируется, а затем упаковывается в каталог сборки. Затем вы можете использовать **[расман](#)** для установки пакета.

Соответствующие статьи

[Стандарты упаковки свода](#)  
[Хранилище Пользователей Arch](#)  
[Создание пакетов](#)  
[Компиляция ядра с ABS](#)  
[makepkg](#)  
[Официальные репозитории](#)  
[расман](#)  
[PKGBUILD](#)  
[Латать в АБС](#)

## Содержание

Обзор

[дерево репозитория](#)

[случая использования](#)

Использование

[извлечение источника PKGBUILD](#)

[извлечение источника PKGBUILD с помощью Git](#)

[извлечение источника PKGBUILD с помощью SVN](#)

[предварительные условия](#)

[нерекурсивная проверка](#)

[проверка пакета](#)

[извлечение более ранней версии пакета](#)

[пакет сборки](#)

Советы и рекомендации

[сохранение измененных пакетов](#)

[другие инструменты](#)

## Общие сведения

"АБС" может использоваться в качестве зонтичного термина, поскольку он включает в себя и опирается на несколько других компонентов; поэтому, хотя и не является технически точным, "АБС" может относиться к следующим инструментам в качестве полного набора инструментов:

### Дерево репозитория

Структура каталогов, содержащая файлы, необходимые для построения всех официальных пакетов, но не сами пакеты, ни исходные файлы программного обеспечения. Он доступен в [репозиториях svn \(https://www.archlinux.org/svn/\)](https://www.archlinux.org/svn/) и [\(https://projects.archlinux.org/svntogit/packages.git/\)](https://projects.archlinux.org/svntogit/packages.git/) git. Дополнительную информацию смотрите в разделе [# дерево репозитория](#).

### PKGBUILD

Сценарий **[Bash](#)**, содержащий URL исходного кода вместе с инструкциями по компиляции и упаковке.

### makepkg

оболочка командный инструмент, который читает PKGBUILDs, автоматически загружает и компилирует источники и создает в `.pkg.tar*` соответствии с PKGEXT массивом `makepkg.conf` В. Вы также можете использовать `makepkg` для создания собственных пользовательских пакетов из [AUR](#) или сторонних источников. Дополнительную информацию смотрите в разделе Создание пакетов.

### расман

расман полностью отделен, но обязательно вызывается либо `makepkg`, либо вручную, чтобы установить и удалить встроенные пакеты и для извлечения зависимостей.

## Аур

Пользовательский репозиторий Arch отделен от ABS, но Aur (неподдерживаемые) PKGBUILDs создаются с помощью `makepkg` для компиляции и упаковки программного обеспечения. В отличие от дерева ABS на вашем локальном компьютере, AUR существует как интерфейс веб-сайта. Он содержит много тысяч пользовательских PKGBUILDs для программного обеспечения, которое недоступно в качестве официального пакета Arch. Если вам нужно построить пакет вне официального дерева Arch, скорее всего, он находится в AUR.

**Предупреждение:** официальные PKGBUILDs предполагают, что пакеты **построены в чистом chroot**. Построение программного обеспечения на *грязной* системе сборки может завершиться неудачей или вызвать неожиданное поведение во время выполнения, потому что если система сборки обнаруживает зависимости динамически, результат зависит от того, какие пакеты доступны в системе сборки.

## Дерево репозитория

Основные, дополнительные и тестовые **официальные репозитории** находятся в репозитории пакетов для **проверки**. Репозитории *community* и *multilib* находятся в репозитории *community*.

Каждый пакет имеет свой собственный подкаталог. Внутри него есть `repos` и `trunk` каталоги. `repos` далее разбивается по имени репозитория (например, `core`) и архитектуре. PKGBUILDs и файлы, найденные в `repos` используются в официальных сборках. Файлы, найденные в `trunk` используются разработчиками в процессе подготовки перед копированием `repos`.

Например, дерево для **acl** (<https://www.archlinux.org/packages/?name=acl>) выглядит следующим образом:

```

список контроля доступа
acl / repos
acl / repos / core-x86_64
acl / repos / core-x86_64 / PKGBUILD
acl / хобот
acl / trunk / PKGBUILD

```

Исходный код пакета отсутствует в справочнике ABS. Вместо PKGBUILD этого он содержит URL-адрес, который будет загружать исходный код при сборке пакета.

## Вариант использования

ABS автоматизирует некоторые задачи, связанные с компиляцией из исходного кода. Его случаи использования являются:

- Любой вариант использования, требующий компиляции или перекompиляции пакета.
- Создание и установка новых пакетов из источника программного обеспечения, для которого еще нет доступных пакетов (см. [раздел Создание пакетов](#)).
- Настройка существующих пакетов в соответствии с вашими потребностями (например, включение или отключение параметров, исправление).
- Перестройте всю свою систему, используя флаги компилятора, "à la FreeBSD" (например, с [pacman-src-git \(https://aur.archlinux.org/packages/pacman-src-git/\)](https://aur.archlinux.org/packages/pacman-src-git/)<sup>AUR</sup>).
- Аккуратно соберите и установите свое собственное собственное ядро (см. [раздел компиляция ядра](#)).
- Получите модули ядра, работающие с пользовательским ядром.
- Легко компилируйте и устанавливайте более новую, старую, бета-версию или версию для разработки пакета Arch, отредактировав номер версии в PKGBUILD.

## Использование

Чтобы получить **PKGBUILD**, необходимый для построения определенного пакета из исходного кода, можно использовать **SVN** или **git**-подход, используя пакет **asp** (<https://www.archlinux.org/packages/?name=asp>), который представляет собой тонкую оболочку вокруг репозитория `svntogit`. Ниже описывается метод на основе `svn`, а также метод [на основе git](#).

### Получить источник PKGBUILD

Существует два метода для извлечения PKGBUILD из исходного кода, первый заключается в использовании Git через `svntogit`, а второй — непосредственно использовать SVN.

#### Получение источника PKGBUILD с помощью Git

В качестве предварительного [условия установите](#) пакет **asp** (<https://www.archlinux.org/packages/?name=asp>). **Asp** (<https://github.com/falconindy/asp>) — это инструмент для извлечения исходных файлов сборки для пакетов Arch Linux с помощью интерфейса Git. Также смотрите раздел форум Arch Linux BBS [\[1\] \(https://bbs.archlinux.org/viewtopic.php?id=185075\)](https://bbs.archlinux.org/viewtopic.php?id=185075).

Чтобы клонировать `svntogit`-репозиторий для конкретного пакета, используйте:

```
$ asp checkout pkgname
```

Это приведет к клонированию репозитория `git` для данного пакета в каталог с именем, подобным пакету.

Чтобы обновить клонированный репозиторий git, запустите `asp update` его, а затем `git pull` внутри репозитория git.

Кроме того, вы можете использовать все другие команды git для проверки старой версии пакета или отслеживания пользовательских изменений. Дополнительную информацию об использовании git смотрите на странице git.

Если вы просто хотите скопировать снимок текущего [PKGBUILD](#) для определенного пакета, используйте:

```
$ asp экспорт pkgname
```

## Получение источника PKGBUILD с помощью SVN

### Предпосылки

Установите пакет [subversion](https://www.archlinux.org/packages/?name=subversion) (<https://www.archlinux.org/packages/?name=subversion>).

### Нерекурсивная проверка

**Предупреждение:** Не загружайте весь репозиторий целиком; только следуйте инструкциям ниже. Весь репозиторий SVN огромен. Мало того, что это займет неприличное количество дискового пространства, но это также будет облагать налогом archlinux.org сервер для вас, чтобы загрузить его. Если вы злоупотребляете этой услугой, ваш адрес может быть заблокирован. Никогда не используйте общедоступный SVN для каких-либо сценариев.

Для проверки основных, дополнительных и тестирования [официальных репозиториях](#):

```
$ svn checkout --depth=нулевой svn:/ / svn.archlinux.org / packages
```

Для проверки репозитория *community* и *multilib*:

```
$ svn checkout --depth=нулевой svn: / / svn. archlinux. org / community
```

В обоих случаях он просто создает пустой каталог, но он знает, что это проверка svn.

### Оформить заказ на посылку

В каталоге, содержащем репозиторий svn, который вы проверили (т. е. *пакеты* или *сообщество*), выполните следующие действия:

```
$ svn update package-name
```

Это будет тянуть пакет, который вы просили в ваш заказ. Отныне, каждый раз, когда вы *svn обновление* на верхнем уровне, это будет обновляться, а также.

Если вы укажете пакет, который не существует, svn не предупредит вас. Он просто напечатает что-то вроде "на ревизии 115847", не создавая никаких файлов. Если это произойдет:

- проверьте правильность написания имени пакета
- убедитесь, что пакет не был перемещен в другой репозиторий (т. е. из сообщества в основной репозиторий)
- проверка <https://www.archlinux.org/packages> чтобы узнать, построен ли пакет из другого базового пакета (например, [python-tensorflow](https://www.archlinux.org/packages/?name=python-tensorflow) (<https://www.archlinux.org/packages/?name=python-tensorflow>) построен из [tensorflow](https://www.archlinux.org/package-s/?name=tensorflow) (<https://www.archlinux.org/package-s/?name=tensorflow>) PKGBUILD)

**Совет:** для получения более ранней версии пакета см. [раздел #Получение более ранней версии пакета](#) .

Вы должны периодически обновлять все ваши извлеченные пакеты, Если вы хотите выполнить перестроение на более поздних версиях репозитория. Чтобы сделать это, сделайте:

```
$ svn обновление
```

### Извлечение более ранней версии пакета

В репозитории svn, который вы проверили, как описано в [разделе #Non-recursive checkout](#) (т. е. "пакеты" или "сообщество"), сначала изучите журнал:

```
$ svn log package-имя пакета
```

Узнайте, какую ревизию вы хотите, изучив историю, а затем укажите ревизию, которую вы хотите извлечь. Например, чтобы проверить ревизию r1729 вы бы сделали:

```
$ svn update-r1729 имя пакета
```

Это позволит обновить существующую рабочую копию *имени пакета* до выбранной редакции.

Вы также можете указать дату. Если никакой ревизии в этот день не существует, svn захватит самый последний пакет до этого времени. В следующем примере проверяется редакция от 2009-03-03:

```
$ svn update -r {20090303} 'имя пакета'
```

Можно также извлечь пакеты в версиях, прежде чем они были перемещены в другой репозиторий; тщательно проверьте журналы на дату их перемещения или номер последней ревизии.

## Пакет сборки

Настройте *makepkg* для сборки пакетов из [pkgbuilds](#), которые вы проверили, как описано в [разделе makepkg#Configuration](#).

Затем скопируйте каталог, содержащий [PKGUILD](#), который вы хотите изменить, в новое расположение. Внесите необходимые изменения и используйте *makepkg* там, как описано в [разделе makepkg#Usage](#) для создания и установки нового пакета.

## Советы и рекомендации

### Сохранить измененные пакеты

Обновление системы с помощью *распап* приведет к замене модифицированного пакета из ABS на одноименный пакет из официальных репозиториях. Смотрите следующие инструкции, как этого избежать.

Вставьте массив группы в PKGBUILD и добавьте пакет в вызываемую группу `modified`.

```
PKGBUILD
-----
группы=('modified')
```

Добавьте эту группу в раздел `IgnoreGroup` in `/etc/pacman.conf`.

```
/ etc / pacman.conf
-----
IgnoreGroup = изменено
```

Если новые версии доступны в официальных репозиториях во время обновления системы, *распап* печатает примечание, что он пропускает это обновление, потому что оно находится в разделе `IgnoreGroup`. На этом этапе измененный пакет должен быть перестроен из ABS, чтобы избежать частичного обновления.

## Другие средства

- [pbget \(http://xyne.archlinux.ca/projects/pbget/\)](http://xyne.archlinux.ca/projects/pbget/)-получение PKGBUILDs для отдельных пакетов непосредственно из веб-интерфейса. Включает поддержку AUR.

Извлечено из "[https://wiki.archlinux.org/index.php?title=Arch\\_Build\\_System&oldid=597727](https://wiki.archlinux.org/index.php?title=Arch_Build_System&oldid=597727)"

Последний раз эта страница редактировалась 16 февраля 2020 года, в 18:06.

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