

# Archiso (Русский)

**Состояние перевода:** На этой странице представлен перевод статьи [Archiso](#). Дата последней синхронизации: 27 февраля 2018. Вы можете [помочь](#) синхронизировать перевод, если в английской версии произошли [изменения](#).

## Ссылки по теме

- [Ремастеринг установочного ISO-образа](#)
- [PXE \(Русский\)](#)
- [Archboot \(Русский\)](#)

**Archiso** - набор bash скриптов, предназначенных для создания полностью функциональных Live-CD/DVD и Live-USB на базе Arch Linux. Это тот же инструмент, который используется для создания официальных образов, но он довольно гибкий инструмент, который может быть использован как для создания дисков восстановления или установочных, так и для специализированных live-CD/DVD/USB систем. Сердце Archiso - *mkarchiso*. Для получения подробного описания всех его опций достаточно вызвать его без параметров, так что здесь будет описано только создание live диска своими руками.

## Contents

[hide]

- 1Установка
- 2Настройка live носителя
  - 2.1Установка пакетов
    - 2.1.1Собственный локальный репозиторий
    - 2.1.2Предотвращение установки пакетов, принадлежащих основной группе
    - 2.1.3Установка пакетов из multilib
  - 2.2Добавление файлов в образ
  - 2.3Загрузчик
  - 2.4Вход в систему
  - 2.5Изменение автоматического входа в систему
- 3Сборка ISO
  - 3.1Пересборка ISO
- 4Использование ISO
- 5Советы и рекомендации
  - 5.1Установка без подключения к Интернету
    - 5.1.1Установка archiso в новый корень
    - 5.1.2Chroot и настройка базовой системы
      - 5.1.2.1Восстановление конфигурации журнала
      - 5.1.2.2Удаление особых правил udev
      - 5.1.2.3Отключение и удаление служб, созданных archiso
      - 5.1.2.4Удаление особых скриптов Live среды
      - 5.1.2.5Импорт ключей archlinux
      - 5.1.2.6Настройка системы
      - 5.1.2.7Включение графического входа (Опционально)
- 6Смотрите также
  - 6.1Документация и учебные пособия
  - 6.2Примеры шаблонов настройки

## Установка

### Примечание:

- Чтобы использовать Archiso, вы должны работать на платформе x86\_64.<sup>[1]</sup>
- Рекомендуется действовать с правами суперпользователя во всех последующих шагах. В противном случае возможны проблемы с ложными разрешениями позже.

Прежде чем начать, [установите](#) пакет [archiso](#) или [archiso-git](#)<sup>AUR</sup>.

Archiso поставляется с двумя "профилями": *releng* и *baseline*.

- Если вы хотите создать полностью индивидуальную версию Arch Linux, предварительно установленной со всеми вашими любимыми программами и конфигурациями, используйте профиль *releng*.
- Если вы просто хотите создать основу для live CD, без предварительно установленных пакетов и минимальной конфигурацией, используйте *baseline*.

Теперь скопируйте профиль на Ваш выбор, в каталог (`~/archlive` используется в примере ниже), в котором вы можете вносить корректировки. Выполните следующую команду, заменив `profile` либо на `releng`, либо на `baseline`.

```
# cp -r /usr/share/archiso/configs/profile/* archlive
```

- Если вы используете профиль `releng` для создания полностью индивидуального образа, вы можете переходить к [#Настройка Live носителя](#) [broken link: invalid section].
- Если вы используете профиль `baseline` для создания пустого образа, то вам не нужно его настраивать, поэтому переходите к [#Создание ISO](#) [broken link: invalid section].

## Настройка live носителя

В этом разделе подробно описывается настройка создаваемого вами образа, определение пакетов и конфигураций, которые вы хотите, чтобы ваш live образ содержал.

Внутри каталога `archlive`, созданного в [#Установка](#), имеется ряд файлов и каталогов; мы рассмотрим лишь несколько из них, в основном:

- `packages.*` - это где вы перечисляете построчно пакеты, которые вы хотите установить, и
- каталог `airootfs` - это каталог, действующий как наложение, и именно там вы делаете все настройки.

Как правило, любые административные задачи, которые вы обычно делаете после новой установки, могут быть выполнены в

скрипте `archlive/airootfs/root/customize_airootfs.sh`, за исключением установки пакетов. Скрипт должен быть написан с точки зрения новой среды, поэтому `/` в скрипте означает корень ISO-образа, который создается.

### Установка пакетов

[Отредактируйте](#) списки пакетов

в `packages.i686`, `packages.x86_64` или `packages.both` чтобы указать, какие пакеты должны быть установлены на live носителе. Суффикс здесь указывает, для какой архитектуры доступны пакеты.

**Примечание:** Если вы хотите использовать [оконный менеджер](#) в Live CD, то вы должны добавить необходимые и правильные [видео драйвера](#), или WM может зависнуть при загрузке.

### Собственный локальный репозиторий



This article or section is a candidate for merging with [Pacman/Tips and tricks](#)

[\(Русский\)#Собственный локальный репозиторий.](#)



**Notes:** Переместите общую информацию (например, дерево репо) в основную статью. (Discuss in [Talk:Archiso \(Русский\)#](#))

Для подготовки своих пакетов или пакетов из [AUR/ABS](#), вы можете [создать собственный локальный репозиторий](#). Если вам нужна поддержка нескольких архитектур, примените соответствующие меры для предотвращения ошибок в будущем. Каждая архитектура должна иметь собственное дерево каталогов:

Например:

```
$ tree ~/customrepo/ | sed "s/$(uname -m)/<arch>/g"

/home/archie/customrepo/
└─ <arch>
   ├── customrepo.db -> customrepo.db.tar.xz
   ├── customrepo.db.tar.xz
   ├── customrepo.files -> customrepo.files.tar.xz
   ├── customrepo.files.tar.xz
   └─ personal-website-git-b99cce0-1-<arch>.pkg.tar.xz

1 directory, 5 files
```

Затем вы можете добавить свой репозиторий, поместив следующее в `~/archlive/pacman.conf` выше других записей (для более высокого приоритета):

```
~/archlive/pacman.conf

...
# Свой (пользовательский) репозиторий
[customrepo]
SigLevel = Optional TrustAll
Server = file:///home/пользователь/customrepo/$arch
...
```

Если пакет подходит, `repo-add` выполняет проверку. Иначе, вы столкнетесь с сообщениями об ошибках, подобными этому:

```
==> ERROR: '/home/archie/customrepo/<arch>/foo-<arch>.pkg.tar.xz' does not
have a valid database archive extension.
```

### Предотвращение установки пакетов, принадлежащих основной группе

По умолчанию `/usr/bin/mkarchiso`, скрипт, который используется `~/archlive/build.sh`, вызывает один из скриптов [arch-install-scripts](#) с именем `pacstrap` без флага `-i`, что заставляет [Pacman](#) не ждать ввода пользователя во время процесса установки.

При внесении в черный список основных пакетов группы путем добавления их в строку `IgnorePkg` в файле `~/archlive/pacman.conf`, [Pacman](#) спрашивает, должны ли они все еще быть установлены, а это значит, что они будут при обходе ввода пользователя. Чтобы избавиться от этих пакетов существует несколько вариантов:

- **"Грязный" способ:** Добавьте флаг `-i` в каждую строку, вызывающую `pacstrap` в `/usr/bin/mkarchiso`.

- **"Чистый":** Создайте копию `/usr/bin/mkarchiso`, в которую вы добавите флаг и адаптируете `~/archlive/build.sh`, так чтобы он вызывал измененную версию скрипта `mkarchiso`.
- **Расширенный способ:** Создайте функцию для `~/archlive/build.sh`, которая явно удаляет пакеты после основной установки. Это принесет вам удовольствие от того, что вам не придется много вводить во время процесса установки.

### Установка пакетов из multilib

Чтобы установить пакеты из репозитория [multilib](#), вам необходимо создать два файла конфигурации `pacman`: один для `x86_64` и другой для `i686`. Скопируйте файл `pacman.conf` в `pacmanx86_64.conf` и `pacmani686.conf`. Раскомментируйте следующие строки, чтобы включить *multilib* в `pacmanx86_64.conf`:

```
pacmanx86_64.conf

[multilib]
SigLevel = PackageRequired
Include = /etc/pacman.d/mirrorlist
```

Затем отредактируйте файл `build.sh` с помощью редактора. Замените следующие строки:

```
build.sh

run_once make_pacman_conf

# Do all stuff for each airootfs
for arch in i686 x86_64; do
    run_once make_basefs
    run_once make_packages
done

run_once make_packages_efi

for arch in i686 x86_64; do
    run_once make_setup_mkinitcpio
    run_once make_customize_airootfs
done
```

на:

```
build.sh

cp -v releng/pacmanx86_64.conf releng/pacman.conf
run_once make_pacman_conf

# Do all stuff for each airootfs
for arch in x86_64; do
    run_once make_basefs
```

```

run_once make_packages
run_once make_packages_efi
run_once make_setup_mkinitcpio
run_once make_customize_airootfs
done

echo make_pacman_conf i686
cp -v releng/pacmani686.conf releng/pacman.conf
cp -v releng/pacmani686.conf ${work_dir}/pacman.conf

for arch in i686; do
    run_once make_basefs
    run_once make_packages
    run_once make_packages_efi
    run_once make_setup_mkinitcpio
    run_once make_customize_airootfs
done

```

Таким образом, пакеты для x86\_64 и i686 будут установлены со своим собственным конфигурационным файлом pacman.

### Добавление файлов в образ

**Примечание:** Для этого вы должны быть суперпользователем, не изменяйте владельца для какого-либо скопированного вами файла, **все** в каталоге `airootfs` должно принадлежать суперпользователю. Правильные владельцы будут выяснены в ближайшее время.

Каталог `airootfs` действует как наложение, считайте его корневым каталогом '/' в вашей текущей системе, поэтому любые файлы, которые вы размещаете в этом каталоге, будут скопированы при загрузке.

Поэтому, если у вас есть набор скриптов `iptables` в вашей текущей системе, которые вы хотите использовать на вашем live образе, скопируйте их:

```
# cp -r /etc/iptables ~/archlive/airootfs/etc
```

Размещение файлов в домашнем каталоге пользователей немного отличается. Не помещайте их в `airootfs/home`, а вместо этого создайте каталог `skel` внутри `airootfs/` и разместите их там. Затем мы добавим соответствующие команды в `customize_airootfs.sh`, которые мы собираемся использовать для их копирования при загрузке и выяснения разрешений.

Сначала создайте каталог `skel`:

```
# mkdir ~/archlive/airootfs/etc/skel
```

Теперь скопируйте файлы 'home' в каталог `skel`, например для `.bashrc`:

```
# cp ~/.bashrc ~/archlive/airootfs/etc/skel/
```

Когда выполняется `~/archlive/airootfs/root/customize_airootfs.sh`, и создается новый пользователь, файлы из каталога `skel` будут автоматически скопированы в новую домашнюю папку, а разрешения будут установлены правильно.

Аналогичным образом, требуется обратить внимание на особые файлы конфигурации, которые находятся где-то внизу по иерархии. В качестве примера конфигурационный файл `/etc/X11/xinit/xinitrc` находится на пути, который может быть перезаписан путем установки пакета. Чтобы поместить файл конфигурации, следует поместить пользовательский `xinitrc` в `~/archlive/airootfs/etc/skel/`, а затем изменить `customize_airootfs.sh`, чтобы переместить его соответствующим образом.

## Загрузчик

Файл по умолчанию должен работать прекрасно, поэтому вам не нужно трогать его.

Из-за модульной природы `isolinux` вы можете использовать множество дополнений, так как все \*.c32 файлы копируются и доступны вам. Посмотрите [официальный сайт syslinux](#) и [репозиторий git archiso](#). Используя указанные аддоны, можно сделать визуально привлекательные и сложные меню. Для получения дополнительной информации смотрите [здесь](#).

## Вход в систему

Запуск X при загрузке выполняется путем включения службы `systemd` менеджера входа в систему. Если вы знаете, какой файл `.service` нуждается в программной ссылке: Отлично. Если нет, то вы можете легко узнать в случае, если вы используете одну и ту же программу в системе, в которой вы строите свой iso. Просто используйте:

```
$ ls -l /etc/systemd/system/display-manager.service
```

Теперь создайте ту же самую программную ссылку в `~/archlive/airootfs/etc/systemd/system`. Для LXDM:

```
# ln -s /usr/lib/systemd/system/lxdm.service  
~/archlive/airootfs/etc/systemd/system/display-manager.service
```

Это позволяет запустить LXDM при загрузке системы в вашей live системе.

В качестве альтернативы вы можете просто включить службу в `airootfs/root/customize_airootfs.sh` вместе с другими включенными там службами.

Если вы хотите, чтобы графическая среда фактически запускалась автоматически во время загрузки, обязательно отредактируйте `airootfs/root/customize_airootfs.sh` и замените

```
systemctl set-default multi-user.target
```

на

```
systemctl set-default graphical.target
```

## Изменение автоматического входа в систему

Файлы настроек автоматического входа для `getty` расположены в `airootfs/etc/systemd/system/getty@tty1.service.d/autologin.conf`.

Вы можете отредактировать этот файл для изменения автоматического входа пользователя в систему:

```
[Service]
ExecStart=
ExecStart=--/sbin/agetty --autologin isouser --noclear %I 38400 linux
```

Или удалите его вообще для отключения автоматического входа в систему.

## Сборка ISO

Теперь вы готовы превратить ваши файлы в .iso, которые вы можете записать на CD или USB:

Сначала создайте каталог `out/`,

```
# mkdir ~/archlive/out/
```

Затем внутри `~/archlive` выполните:

```
# ./build.sh -v
```

Теперь скрипт загрузит и установит пакеты, указанные в `work/*/airootfs`, создаст ядро и инициализирует образы, применит ваши настройки и, наконец, создаст iso в `out/`.

## Пересборка ISO

Пересборка iso после изменений официально не поддерживается. Однако это легко осуществить, применив два шага. Сначала вам нужно удалить файлы блокировки в рабочем каталоге:

```
# rm -v work/build.make_*
```

Кроме того, требуется отредактировать скрипт `airootfs/root/customize_airootfs.sh`, и добавить команду `id` в начале строки `useradd`, как показано здесь. В противном случае пересборка останавливается на этом моменте, потому что пользователь, который должен быть добавлен, уже существует [\[2\]](#).



**This article or section needs language, wiki syntax or style improvements.**



**Reason:** Опечатка? (Discuss in [Talk:Archiso \(Русский\)#](#))

```
! id arch && useradd -m -p "" -g users -G
"adm,audio,floppy,log,network,rfskill,scanner,storage,optical,power,wheel" -s
/usr/bin/zsh arch
```

Также удалите постоянные данные, например созданные пользователями или символические ссылки, такие как `/etc/sudoers`.



**This article or section needs expansion.**



**Reason:** Сообщите больше информации о том, что необходимо удалить или сбросить. (Discuss in [Talk:Archiso \(Русский\)#](#))

Сборку можно немного ускорить, отредактировав скрипт `pacstrap` (расположенный в `/bin/pacstrap`) и изменив следующее в строке 361:

До:

```
if ! pacman -r "$newroot" -Sy "${pacman_args[@]}"; then
```

После:

```
if ! pacman -r "$newroot" -Sy --needed "${pacman_args[@]}"; then
```

Это увеличивает скорость первоначальной загрузки, так как не нужно загружать и устанавливать какие-либо основные пакеты, которые уже установлены.

## Использование ISO

Для получения дополнительной информации смотрите раздел [Category:Getting and installing Arch \(Русский\)#Способы установки](#).

## Советы и рекомендации

### Установка без подключения к Интернету



This article or section is a candidate for moving to [Installation without Internet access](#).



**Notes:** наверно, это можно вынести в отдельную статью (Discuss in [Talk:Archiso \(Русский\)#](#))



**The factual accuracy of this article or section is disputed.**



**Reason:** Вместо копирования файлов из `archiso`, образ `"bootstrap"`, доступный на зеркалах Arch, может быть извлечен непосредственно на целевой диск. Вы получите минимальную систему без необходимости модификации `archiso`. (Discuss in [Talk:Archiso \(Русский\)#](#))

Если вы хотите установить `archiso` (например, [официальный ежемесячный выпуск](#)) без подключения к Интернету, или, если вы не хотите загружать пакеты снова:

Сначала следуйте инструкциям [руководства по установке](#), а потом пропустите разделы с [Соединение с Интернетом](#) до раздела [Установка основных пакетов](#).

#### Установка `archiso` в новый корень

Вместо того, чтобы устанавливать пакеты с помощью `pacstrap` (которые будут пытаться загрузить из удаленных репозиториях), скопируйте все в live среду в новый корень:

```
# cp -ax / /mnt
```

**Примечание:** Опция `(-x)` исключает некоторые специальные каталоги, так как они не должны копироваться в новый корень.

Затем скопируйте образ ядра в новый корень, чтобы сохранить целостность новой системы:



```
# cp -vaT /run/archiso/bootmnt/arch/boot/$(uname -m)/vmlinuz  
/mnt/boot/vmlinuz-linux
```

После этого сгенерируйте fstab, как описано в [руководстве по установке#Fstab](#).

### Chroot и настройка базовой системы

Далее, сделать операцию chroot в вашей вновь установленной системы:

```
# arch-chroot /mnt /bin/bash
```

**Примечание:** Перед выполнением других шагов [руководства по установке#Настройка системы](#) (например, локаль, раскладка клавиатуры и т.д.) необходимо избавиться от следов Live среды (другими словами, настройка archiso, которая не соответствует Live среде).

### Восстановление конфигурации журнала

[Эта настройка archiso](#) приведет к сохранению системного журнала в ОЗУ, а это означает, что журнал после перезагрузки будет недоступен:

```
# sed -i 's/Storage=volatile/#Storage=auto/' /etc/systemd/journald.conf
```

### Удаление особых правил udev

[Это правило udev](#) автоматически запускает dhcpcd, если есть какие-либо проводные сетевые интерфейсы.

```
# rm /etc/udev/rules.d/81-dhcpcd.rules
```

### Отключение и удаление служб, созданных archiso

Некоторые служебные файлы создаются для Live среды, отключите службы и удалите файл, поскольку они не нужны для новой системы:

```
# systemctl disable pacman-init.service choose-mirror.service  
# rm -r /etc/systemd/system/{choose-mirror.service,pacman-init.service,etc-  
pacman.d-gnupg.mount,Getty@tty1.service.d}  
# rm /etc/systemd/scripts/choose-mirror  
# rm /etc/systemd/system/archiso-start.service  
# rm /etc/systemd/system/multi-user.target.wants/archiso-start.service
```

### Удаление особых скриптов Live среды

Есть некоторые скрипты, которые установлены в live системе скриптами archiso, которые не нужны для новой системы:

```
# rm /etc/systemd/system/Getty@tty1.service.d/autologin.conf  
# rm /root/{.automated_script.sh,.zlogin}  
# rm /etc/mkinitcpio-archiso.conf  
# rm -r /etc/initcpio
```

### Импорт ключей archlinux

Чтобы использовать официальные репозитории, нам нужно импортировать главные ключи archlinux ([pacman/Package signing \(Русский\)#Инициализация связки ключей](#)). Этот шаг обычно делается с помощью pacstrap, но может быть достигнут с помощью

```
# pacman-key --init
# pacman-key --populate archlinux
```

**Примечание:** Клавиатура или мышь должны быть подключенными для генерации энтропии и ускорения первого шага

### Настройка системы

Теперь вы можете выполнить пропущенные шаги раздела [руководства по установке#Настройка системы](#) (установка локали, часовой пояс, имя хоста и т.д.) и завершить установку, создав исходный ramdisk, как описано в [руководстве по установке#Initramfs](#).

### Включение графического входа (Опционально)

Если вы используете менеджер входа, такой как GDM, вы можете изменить параметр по умолчанию systemd multi-user.target на тот, который позволяет графический вход в систему.

```
# systemctl disable multi-user.target
# systemctl enable graphical.target
```

## Смотрите также

---

### Документация и учебные пособия

- [Страница проекта Archiso](#)
- [Official Официальная документация](#)

### Примеры шаблонов настройки

- [Live DJ дистрибутив, основанный на ArchLinux и созданный с помощью Archiso](#)