

# Migrating between architectures (Русский)

**Состояние перевода:** На этой странице представлен перевод статьи [Migrating between architectures](#). Дата последней синхронизации: 27 декабря 2019. Вы можете [помочь](#) синхронизировать перевод, если в английской версии произошли [изменения](#) ([https://wiki.archlinux.org/index.php?title=Migrating\\_between\\_architectures&diff=0&oldid=585450](https://wiki.archlinux.org/index.php?title=Migrating_between_architectures&diff=0&oldid=585450)).

На этой странице описаны два возможных способа миграции установленных систем с архитектуры i686 (32 бита) на x86\_64 (64 бита). Методы избегают *полной* переустановки (т.е. очистки жёсткого диска). Один метод использует liveCD, другой изменяет систему изнутри.

**Примечание:** Технически этот процесс по-прежнему включает "переустановку", поскольку каждый пакет в системе должен быть заменён. Эти методы просто пытаются сохранить как можно больше из вашей существующей системы.

**Важно:** Если не указано явно, все эти методы **не проверены** и могут нанести непоправимый урон вашей системе. Продолжайте на свой риск.

## Contents

### Общая подготовка

- Проверка 64-битной архитектуры
- Дисковое пространство
- Источник питания
- Пакеты для отката изменений

### Метод 1: с использованием Arch LiveCD

### Метод 2: из работающей системы

- Подготовка пакетов
  - Кеширование старых пакетов
  - Установка busybox
  - Изменение архитектуры Pacman
  - Загрузка новых пакетов
- Установка пакетов
  - Установка ядра (64 бита)
  - Установка lib32-glibc

- [Перезагрузка](#)
- [Переключение в терминал](#)
- [Установка Расман](#)
- [Установка остальных пакетов](#)

#### Очистка

- [Флаги компиляции Makepkg](#)

#### Решение проблем

- [Busybox](#)
- [Lib32-glibc](#)
- [Миграция с систем 2017 года и старше](#)
- [KDE не запускается после миграции с 32-bit на 64-bit](#)
- [Проблемы Mutt при включённом кеше](#)

[Смотрите также](#)

## Общая подготовка

---

### Проверка 64-битной архитектуры

Для запуска 64-разрядного программного обеспечения, у вас должен быть 64-разрядный процессор. Большинство современных процессоров способны запускать 64-битное программное обеспечение. Вы можете проверить свой процессор с помощью следующей команды:

```
# grep --color -w lm /proc/cpuinfo
```

Для процессоров, поддерживающих x86\_64, это вернёт выделенный цветом флаг `lm` (“long mode”). Остерегайтесь того, что `lahf_lm` — это другой флаг, который не указывает на 64-битные возможности.

### Дисковое пространство

Вы должны быть готовы к увеличению размера директории `/var/cache/pacman/pkg` примерно в два раза. Предполагается, что в кеше находятся только установленные в данный момент пакеты, как будто “`pacman -Sc`” (clean) был недавно запущен. Увеличение дискового пространства связано с дублированием каждого пакета для версии i686 и x86\_64.

Если у вас недостаточно места, используйте [Parted \(Русский\)](#), чтобы изменить размер соответствующего раздела или примонтировать другой раздел в `/var/cache/pacman`.

Не удаляйте пакеты старой архитектуры из кеша, пока система не будет полностью работать в новой архитектуре. Слишком раннее удаление пакетов может оставить вас без возможности вернуть изменения.

## Источник питания

Миграция может занять значительное количество времени и было бы неудобно прерывать процесс. Вы должны рассчитывать как минимум минимум на час, в зависимости от количества и размера установленных пакетов и скорости интернет-соединения (хотя вы можете предварительно загрузить все пакеты перед началом критической части). Убедитесь, что вы подключены к стабильному источнику питания, предпочтительно с какой-либо отказоустойчивостью или резервным аккумулятором.

## Пакеты для отката изменений

Если процесс миграция прервётся на полпути, существуют пакеты, которые могут помочь разобраться в ситуации, но они должны быть установлены до переноса основных пакетов. Подробнее об их использовании рассказано в разделе [#Решение проблем](#).

Один из пакетов — [busybox](https://www.archlinux.org/packages/?name=busybox) (<https://www.archlinux.org/packages/?name=busybox>), который можно использовать для отката изменений. Он использует статическую привязку и не зависит от каких-либо библиотек. Должна быть установлена 32-разрядная версия (i686).

Другой пакет — это [lib32-glibc](https://www.archlinux.org/packages/?name=lib32-glibc) (<https://www.archlinux.org/packages/?name=lib32-glibc>) из x86\_64-репозитория [multilib \(Русский\)](#), который может пригодиться только при миграции с 32 бит. Вы можете безопасно пропустить этот пакет, а для запуска 32-битных программ явно вызывать `/lib/ld-linux.so.2`.

## Метод 1: с использованием Arch LiveCD

1. [Скачайте \(https://www.archlinux.org/download/\)](https://www.archlinux.org/download/) и запишите свежий образ Arch Linux ISO.
2. Загрузите Arch LiveCD в режиме x86\_64.
3. Настройте доступ к сети.
4. Примонтируйте текущую установленную систему. Например: `mount /dev/sda1 /mnt`.
5. Отредактируйте репозитории LiveCD в `/etc/pacman.conf` для соответствия текущим репозиториям `/mnt/etc/pacman.conf`.
6. Обновите локальную базу данных pacman и очистите кэш при помощи следующих команд:

```
# pacman --root /mnt -Syy
# pacman --root /mnt -Scc
```

6. Сначала попробуйте переустановить только группу [base](https://www.archlinux.org/packages/?name=base) (<https://www.archlinux.org/packages/?name=base>), затем каждый пакет, при установке которого возникла ошибка. Найти эти пакеты можно при помощи команды `pacman --root /mnt -Qo <error file>`. Затем повторяйте установку группы [base](https://www.archlinux.org/packages/?name=base) (<https://www.archlinux.org/packages/?name=base>) до тех пор, пока установка не закончится без ошибок.

```
# pacman --root /mnt -S base
```

7. Получите список всех установленных пакетов с помощью следующей команды и переустановите их:

```
# pacman --root /mnt -Qnq | pacman --root /mnt -S -
```

8. Команду следует запускать дважды, потому что при первом запуске многие пакеты не смогут выполнить пост-установочные скрипты из-за неверной архитектуры утилит `sed`, `grep`, `perl` и других. Либо вы можете записать пакеты, установка которых завершилась ошибкой, а затем вернуться к ним после окончания миграции и переустановить.

Если же при установке закончилось свободное место, вы можете отфильтровать пакеты по алфавиту и обновлять их группами. Например, сначала `... | grep '^[a-k]' | ...`, затем `^l` и в конце `^[m-z]`. В таком случае, вам также придется запускать команду

`pacman --root /mnt -Scc` для освобождения места на диске после каждой установки. Или, так как весь пакет загружается в `ramfs LiveCD`, вы также можете смонтировать этот раздел или создать символическую ссылку на `/var/cache/pacman/pkg` (например, `ln -s /mnt/var/cache/pacman/pkg /var/cache/pacman/pkg`).

9. И наконец, выполните следующие команды:

```
# arch-chroot /mnt
# mkinitcpio -p linux
```

10. Также обратите внимание на загрузчик, который тоже может потребовать миграции. К примеру:

```
# grub-install --recheck /dev/sda
```

11. После перезагрузки в новую 64-битную систему, отредактируйте `/etc/makepkg.conf.pacnew` и переместите его в `/etc/makepkg.conf` для подготовки к миграции на новую архитектуру. Затем пересоберите сторонние пакеты, которые будут включать в себя пакеты из [AUR \(Русский\)](#).

Перед пересборкой может потребоваться удаление заброшенных пакетов. Для поиска 32-битных пакетов воспользуйтесь данной командой:

```
$ LC_ALL=C pacman -Qo `find /usr/bin -type f -exec bash -c 'file "{}" | grep 32-bit' \; | cut -d':' -f1` | cut -d' ' -f5 | sort | uniq | tee list
```

## Метод 2: из работающей системы

Убедитесь, что ваша система полностью обновлена и работоспособна.

```
# pacman -Syu
```

### Подготовка пакетов

#### Кеширование старых пакетов

**Примечание:** Если вы устанавливали пакеты без поддержки новой архитектуры из [AUR \(Русский\)](#) или сторонних репозиториев, pacman уведомит, если не сможет найти замену. Запишите их, чтобы переустановить пакеты после миграции и удалить при помощи команды `pacman -Rsn название_пакета`.

Если у вас нет в кеше всех установленных пакетов, загрузите их (для старой архитектуры) для возможности восстановления.

```
# pacman -Qqn | pacman -Sw -
```

или используйте пакет [fakepkg](https://aur.archlinux.org/packages/fakepkg/) (<https://aur.archlinux.org/packages/fakepkg/>)<sup>AUR</sup>, чтобы сгенерировать их.

#### Установка busybox

Если вы мигрируете с 32 бит на 64 бита, самое время установить 32-битный [busybox](http://www.archlinux.org/packages/?name=busybox) (<http://www.archlinux.org/packages/?name=busybox>):

```
# pacman -S busybox
```

#### Изменение архитектуры Pacman

Отредактируйте `/etc/pacman.conf` и измените *Architecture* с `auto` на `x86_64`.

Убедитесь, что в списках серверов в `/etc/pacman.conf` и `/etc/pacman.d/mirrorlist` используется `$arch` вместо принудительно заданных `i686` или `x86_64`. Затем принудительно синхронизируйте Pacman с репозиториями:

```
# pacman -Syu                                # принудительная синхронизация с новой архитектурой
```

## Загрузка новых пакетов

Скачайте для всех установленных пакетов версию для новой архитектуры:

```
# pacman -Sw $(pacman -Qqn|sed '/^lib32-/ d') # загрузка пакетов для новой архитектуры
```

**Важно:** Не устанавливайте сразу пакет *lib32-glibc*. После *ldconfig*, при установке *linux*, созданный образ будет включать библиотеки вида *librt.so* в */usr/lib32*, которые при загрузке не будут найдены и процесс загрузки прервётся.

## Установка пакетов

### Установка ядра (64 бита)

Обновить ядро до 64-битного (x86\_64) просто и безопасно: 32-битные и 64-битные приложения работают с 64-битным ядром одинаково хорошо.

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

```
# pacman -S linux
```

### Установка lib32-glibc

Установите [lib32-glibc](https://www.archlinux.org/packages/?name=lib32-glibc) (<https://www.archlinux.org/packages/?name=lib32-glibc>) на случай возникновения непредвиденных неполадок, а также 64-разрядную версию библиотеки [glibc](https://www.archlinux.org/packages/?name=glibc) (<https://www.archlinux.org/packages/?name=glibc>). Для этого может потребоваться включить репозиторий [multilib \(Русский\)](#) в */etc/pacman.conf*, если вы ещё не сделали этого ранее.

```
# pacman -S glibc lib32-glibc
```

**Важно:** Если вы не установите 64-разрядную версию *glibc* в то же время, что и *lib32-glibc*, то *lib32-glibc* не будет работать. Тем не менее, *pacman* не предупредит вас об этом, так как 32-разрядный пакет *glibc* также удовлетворяет зависимость.

## Перезагрузка

После перезагрузки убедитесь, что работаете в архитектуре x86\_64:

```
$ uname -m
```

```
x86_64
```

## Переключение в терминал

Для продолжения, переключитесь в текстовую виртуальную консоль (напр. при помощи Ctrl+Alt+F1), если это возможно. Если при использовании первой консоли возникают проблемы, используйте вместо неё вторую (Ctrl+Alt+F2). Псевдо-терминалы вроде SSH должны работать, но в целях предосторожности рекомендуется прямой доступ. В процессе миграции некоторые пакеты будут заменены, что может сказаться на X11 и загрузить систему будет невозможно.

## Установка Расман

**Важно:** После запуска обновления расман и его зависимостей нельзя прерывать процесс! Расман и все зависимости должны быть установлены одновременно и одной командой.

**Важно:** После выполнения этой команды только Busybox, Bash и Расман останутся исполняемыми до момента миграции остальных пакетов. Если вы используете sudo, вы должны получить привилегии root перед запуском следующей команды.

Воспользуйтесь pactree для установки Расман и всех его зависимостей:

```
# pactree -l расман | расман -S -
```

В процессе могут возникать ошибки, но они не страшны пока Расман продолжает работать.

**Важно:** До окончания выполнения команды не перезагружайтесь. Если это всё-таки произошло, продолжите установку, используя [Chroot \(Русский\)](#) в другом Linux-окружении (например, с установочного образа)

## Установка остальных пакетов

Установите все предварительно скачанные пакеты для новой архитектуры. (Сварите кофе и сделайте пару бутербродов, это может занять некоторое время.)

```
# расман -Qqn | расман -S -
```

Если некоторые пакеты не были корректно установлены, теперь вы сможете их успешно переустановить. Если вам лень, то можно просто перезапустить предыдущую команду для переустановки всех пакетов.

После этого этапа миграция будет окончена, можно безопасно перезагрузить систему.

Однако, если были установлены пакеты из AUR, необходимо их переустановить отдельно. Список пакетов можно получить следующей командой:

```
$ pacman -Qqm
```

## Очистка

Теперь можно безопасно удалить **busybox** и **lib32-glibc**.

### Флаги компиляции Makepkg

При обновлении, новая версия `/etc/makepkg.conf` может быть сохранена как `/etc/makepkg.conf.pacnew`. Если это произошло, то необходимо заменить прежнюю версию на новую или отредактировать ее для сборки при помощи [makepkg \(Русский\)](#) в будущем.

```
# mv /etc/makepkg.conf /etc/makepkg.conf.backup && mv /etc/makepkg.conf.pacnew /etc/makepkg.conf
```

Также хорошей идеей может быть проверка всех "новых" файлов в `/etc`. Это можно сделать с помощью следующей команды:

```
# find /etc/ -type f -name \*.pac\*
```

## Решение проблем

Во время процесса обновления, когда происходит замена `glibc` на версию для новой архитектуры, приложения для старой архитектуры перестанут запускаться. Проблему можно решить при помощи [busybox](https://www.archlinux.org/packages/?name=busybox) (<https://www.archlinux.org/packages/?name=busybox>) и [lib32-glibc](https://www.archlinux.org/packages/?name=lib32-glibc) (<https://www.archlinux.org/packages/?name=lib32-glibc>).

### Busybox

В Arch Busybox статически связан. Он может быть запущен без библиотек и предоставить вам массу команд. Например, для извлечения i686-версии Pacman из пакета в кеше можно воспользоваться следующей командой:

```
# busybox tar xf /var/cache/pacman/pkg/pacman-3.3.2-1-i686.pkg.tar.gz -C <some folder>
```

### Lib32-glibc

Пример запуска 32-битного `/bin/ls`:

```
# /lib/ld-linux.so.2 /bin/ls
```



## Миграция с систем 2017 года и старше

Вы можете столкнуться с проблемами при обновлении старых 32-битных систем, так как в 2017 году Arch Linux прекратил поддержку архитектуры i686. Для безопасного обновления рекомендуется использовать [Arch Linux Archive](#). Например, сделайте снимок репозитория, обновите текущую систему и переключитесь на 64-битные версии пакетов **той же версии**.

Обратите внимание, что, возможно, вам придётся понизить уровень доверия в `pacman.conf` (например, `PackageTrustAll` для доверия ко всем пакетам), так как не все использованные для подписи пакетов GPG-ключи до сих пор действительны. Также рекомендуется удостовериться в использовании протокола HTTPS, чтобы как минимум быть уверенным в том, что пакеты загружаются с домена `archlinux.org`.

## KDE не запускается после миграции с 32-bit на 64-bit

KDE будет "падать" после перехода с 32-bit на 64-bit из-за некоторых 32-битных пакетов в `/var/tmp`. Для решения проблемы достаточно удалить все папки с кешем:

```
# rm -rf /var/tmp/kdecache-*
```

## Проблемы Mutt при включённом кеше

Если после завершения миграции [Mutt](#) зависает на открытии папок, попробуйте переименовать папку с кешем. Если это решит проблему, переименованную папку можно удалить — Mutt сам создаст новую.

## Смотрите также

- [Migrate installation to new hardware](#)

---

Retrieved from "[https://wiki.archlinux.org/index.php?title=Migrating\\_between\\_architectures\\_\(Русский\)&oldid=592779](https://wiki.archlinux.org/index.php?title=Migrating_between_architectures_(Русский)&oldid=592779)"

---

This page was last edited on 27 December 2019, at 20:23.

Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.