

Beginners' guide (Русский)

From ArchWiki

Данная страница содержит подробное описание процесса установки Arch Linux с использованием Arch Install Scripts (<https://projects.archlinux.org/arch-install-scripts.git/>) (скриптов установки Arch). Первым делом рекомендуем вам ознакомиться с ЧаВо.

Поддерживаемая сообществом ArchWiki — основной источник информации, к которому следует обратиться в случае возникновения проблем. Канал IRC (<irc://irc.freenode.net/#archlinux>) и форумы (англоязычный (<https://bbs.archlinux.org/>) и русскоязычный (<http://archlinux.org.ru/forum>)) также могут оказаться полезными, если не удалось найти ответ на свой вопрос на вики. Согласно философии Arch, если вы не знакомы с какой-либо командой, рекомендуется выполнить *man команда* для прочтения соответствующей страницы справочного руководства.

Ссылки по теме

Category:Accessibili
(Русский)

Руководство по
установке

Diskless system

Установка по SSH

Основные
рекомендации

Устранение часто
встречающихся
неполадок

Contents

- 1 Системные требования
- 2 Подготовка установочного носителя
 - 2.1 USB и компакт-диски
 - 2.2 Установка по сети
 - 2.3 Установка из существующей системы Linux
 - 2.4 Установка на виртуальную машину
- 3 Загрузка установочного носителя
 - 3.1 Проверка того, произошла ли загрузка в режиме UEFI
 - 3.2 Устранение неполадок загрузки
- 4 Изменение языка
- 5 Настройка интернет-соединения
 - 5.1 Проводное
 - 5.2.1 Без использования wifi-menu
 - 5.2 Беспроводное
 - 5.3 Аналоговый модем, ISDN или PPPoE DSL
 - 5.4 Подключение через прокси
- 6 Подготовка дисковых устройств
 - 6.1 Определение устройств
 - 6.2 Типы таблиц разделов
 - 6.3 Инструменты разметки
 - 6.3.1 Использование parted в интерактивном режиме
 - 6.4 Создание новой таблицы разделов
 - 6.5 Схема разметки
 - 6.5.1 Примеры для UEFI/GPT

- 6.5.2 Примеры для BIOS/MBR
- 6.6 Создание файловых систем
- 6.7 Включение раздела подкачки
- 6.8 Монтирование разделов
- 7 Выбор зеркала
- 8 Установка базовой системы
- 9 Генерация fstab
- 10 chroot и настройка базовой системы
 - 10.1 Локаль
 - 10.2 Консольный шрифт и раскладка клавиатуры
 - 10.3 Временная зона
 - 10.4 Аппаратные часы
 - 10.5 Модули ядра
 - 10.6 Имя узла (hostname)
 - 10.7 Настройка сети
 - 10.7.1 Проводное
 - 10.7.1.1 Динамический IP
 - 10.7.1.2 Статический IP
 - 10.7.2 Беспроводное
 - 10.7.2.1 Добавление беспроводных сетей
 - 10.7.2.2 Автоматическое подключение к известным сетям
 - 10.7.3 Аналоговый модем, ISDN или PPPoE DSL
 - 10.8 Создание начального окружения виртуального диска
 - 10.9 Установка пароля суперпользователя
 - 10.10 Установка и настройка загрузчика
 - 10.10.1 Для материнских плат с BIOS
 - 10.10.2 Для материнских плат с UEFI
- 11 Размонтирование разделов и перезагрузка
- 12 После установки

Системные требования

Arch Linux должен работать на любой совместимой с архитектурой i686 машине, имеющей минимум 64 Мбайт ОЗУ. Базовая установка со всеми пакетами из группы base (https://www.archlinux.org/groups/x86_64/base/) должна занимать меньше 800 Мбайт дискового пространства. Если свободное место на вашем диске ограничено, размер системы можно уменьшить, но при этом вы должны четко понимать, что вы делаете.

Подготовка установочного носителя

Совет: В ISO-образы archboot (<https://downloads.archlinux.de/iso/archboot/latest>) включен скрипт установки, который помогает выполнить несколько шагов этого руководства автоматически в интерактивном режиме. Для получения дополнительной информации смотрите статью Archboot.

Загрузить самую свежую версию официального установочного образа можно со страницы загрузки (<https://archlinux.org/download/>). Обратите внимание, что он поддерживает как 32-битную, так и 64-битную архитектуру.

Настоятельно рекомендуется всегда использовать самую свежую версию ISO-образа.

Примечание:

Настоятельно рекомендуется проверять подпись установочного образа перед тем, как устанавливать с него систему. Загрузите подпись PGP (*PGP signature*, файл *.sig*) в каталог с ISO-образом и выполните:

```
$ gpg --verify archlinux-версия-dual.iso.sig
```

Если открытый ключ не найден, импортируйте его командой `gpg --recv-keys [1]` (<http://sparewotw.wordpress.com/2012/10/31/how-to-verify-signature-using-sig-file/>).

Если у вас уже есть работающая система Arch Linux, вы также можете использовать следующую команду для проверки подписи:

```
$ pacman-key -v archlinux-версия-dual.iso.sig
```

Для проверки контрольных сумм `md5` и `sha1` используйте, соответственно, команды `md5sum` и `sha256sum`.

USB и компакт-диски

На странице [Optical disc drive \(Русский\)#Запись](#) приведены инструкции по записи образа на диски CD/DVD; [USB flash installation media](#) поясняет, как записать загружаемый образ на сменный флеш-носитель USB.

Установка по сети

Смотрите [PXE](#).

Установка из существующей системы Linux

Смотрите [Install from existing Linux](#). Это особенно полезно при удаленной установке через VNC или SSH. Смотрите также [Install from SSH](#).

Установка на виртуальную машину

Установка Arch Linux на виртуальную машину — хороший способ ознакомиться с системой и процедурой установки без необходимости покидать вашу текущую операционную систему и пересоздавать разделы на жестком диске. Это также дает вам возможность оставить данное руководство открытым в браузере, пока вы производите установку. Некоторым пользователям, возможно, будет полезно иметь независимую систему Arch Linux в виртуальной машине для тестирования.

Примеры виртуальных машин смотрите на странице [Category:Hypervisors](#).

Конкретная последовательность действий по подготовке виртуальной машины зависит от используемого программного обеспечения, но в целом вам потребуется выполнить следующее:

1. Создать виртуальный диск, на который будет произведена установка.
2. Правильно настроить виртуальную машину.
3. Загрузиться с ISO-образа при помощи виртуального CD-привода.
4. Продолжить установку, перейдя к разделу [#Загрузка установочного носителя](#).

Следующие статьи могут оказаться полезными:

- [VirtualBox#Installation steps for Arch Linux guests](#)
- [VirtualBox#Install a native Arch Linux system from VirtualBox](#)^[broken link: invalid section]
- [VirtualBox#Run a native Arch Linux installation inside VirtualBox](#)^[broken link: invalid section]
- [Arch Linux as VMware guest](#)
- [Moving an existing install into \(or out of\) a virtual machine](#)

Загрузка установочного носителя

Большинство современных систем позволяют вам выбрать загрузочное устройство во время выполнения POST, обычно при нажатии на клавишу `F12` во время появления экрана загрузки BIOS. Выберите устройство, содержащее загрузочный образ Arch.

Также вы просто можете изменить последовательность загрузки в BIOS вашего компьютера. Для этого нажмите соответствующую клавишу (обычно `Delete`, `F1`, `F2`, `F11` или `F12`) во время выполнения POST. После этого появится экран настройки BIOS, где можно указать последовательность, в которой система будет искать устройства для загрузки. Поставьте носитель, содержащий загрузочный образ Arch на первое место. Выберите пункт "Сохранить и выйти" (или эквивалент в вашем BIOS), после чего ваш компьютер должен перезагрузиться и обнаружить загрузочный образ.

Когда появится меню Arch, выберите "Boot Arch Linux" и нажмите `Enter` для входа в live-окружение, из которого вы будете осуществлять установку (в случае загрузки в режиме UEFI название может быть примерно следующим: "Arch Linux archiso x86_64 UEFI").

Проверка того, произошла ли загрузка в режиме UEFI

В случае, если у вас материнская плата UEFI с включенным режимом загрузки UEFI (и установлен более предпочтительным чем режим BIOS/Legacy), CD/USB-носитель автоматически запустит Arch Linux через `gummiboot` (Русский), в результате чего вы увидите следующее меню (с белыми буквами на черном фоне) с подсвеченным первым пунктом:

```
-----  
Arch Linux archiso x86_64 UEFI USB  
-----
```

```
UEFI Shell x86_64 v1
UEFI Shell x86_64 v2
UEFI Default Loader
```

Если вы не обратили внимание на то, каким было меню, или если вы хотите удостовериться, что произошла загрузка в режиме UEFI, выполните:

```
# efivar -l
```

Если *efivar* корректно выведет список переменных UEFI, значит вы загрузились в режиме UEFI. Если нет, проверьте, что удовлетворены все требования, описанные в разделе Unified Extensible Firmware Interface#Requirements for UEFI Variables support to work properly^[broken link: invalid section].

Устранение неполадок загрузки

- Если вы используете видеоадаптер Intel и экран гаснет в процессе загрузки, проблема, вероятнее всего, связана с настройками режима ядра. Смотрите Intel graphics (Русский)#Пустой экран во время загрузки системы на этапе "Loading modules" и Kernel mode setting#Disabling modesetting.
- Если экран *не* гаснет, но процесс загрузки просто зависает при попытке загрузки ядра, нажмите `Tab`, предварительно выбрав необходимый пункт меню, добавьте в конец строки опцию `acpi=off` и нажмите `Enter`.

Изменение языка

Теперь вы видите приглашение командной строки, автоматически войдя в live-систему от имени суперпользователя. Ваша текущая оболочка — `Zsh`, предоставляющая продвинутое автодополнение по нажатию `Tab`, а также другие возможности `grml` (<http://grml.org/zsh/>). Для редактирования текстовых файлов предлагается консольный редактор *nano*. Если вы не знакомы с ним, прочтите раздел *nano* (Русский)#Использование. Если вам необходима двойная загрузка с Windows, смотрите статью Двойная загрузка Windows и Arch.

Совет: Данные действия для большинства пользователей не являются обязательными. Они полезны только если вы планируете писать что-либо в конфигурационных файлах на другом языке, использовать диакритические знаки в пароле Wi-Fi, или получать системные сообщения (например, об ошибках) на вашем родном языке. Эти изменения затрагивают *только* процесс установки.

По умолчанию используется американская раскладка клавиатуры (`us`). Если вам необходима другая, используйте команду:

```
# loadkeys раскладка
```

где *раскладка* заменяется на *ru* (русская), *ua* (украинская), *fr*, *dvorak*, *be-latin1* и т.д. Для просмотра списка двухсимвольных кодов стран смотрите эту статью в Википедии. Чтобы отобразить список всех доступных раскладок, используйте команду `localectl list-keymaps`.

Если некоторые символы из алфавита вашего языка (например, не-латинские) отображаются как белые квадраты или другие символы, вы можете заменить консольный шрифт на любой из каталога `/usr/share/kbd/consolefonts/`. Например:

```
# setfont cyr-sun16
```

Можно использовать команду `showconsolefont`, чтобы отобразить все содержимое загруженного шрифта. Имейте в виду, что названия шрифтов чувствительны к регистру, поэтому пишите их *именно* так, как вы их видите. Для получения дополнительной информации смотрите раздел **Шрифты#Шрифт в консоли**.

По умолчанию в качестве системного языка устанавливается английский язык (US). Если вы хотите изменить язык, используемый в процессе установки (*в данном примере на русский*), в файле `/etc/locale.gen` удалите символ комментария `#` перед нужной вам локалью, оставив также английскую (US). Пожалуйста, выбирайте записи с кодировкой UTF-8:

```
# nano /etc/locale.gen
```

```
en_US.UTF-8 UTF-8
```

```
ru_RU.UTF-8 UTF-8
```

```
# locale-gen
```

```
# export LANG=ru_RU.UTF-8
```

Примечание: Переключение раскладки осуществляется строго указанным для каждого языка сочетанием клавиш, для русского — `LCtrl+LShift` (левый `Ctrl` + левый `Shift`).

Настройка интернет-соединения

Важно: С версии версии 197 (<http://cgit.freedesktop.org/systemd/systemd/tree/NEWS?id=dee4c244254bb49d1ffa8bd7171ae9cce596d2d0>) `udev` больше не присваивает сетевым интерфейсам имена вида `wlanX` и `ethX`. Если вы ранее пользовались другим дистрибутивом или переустанавливаете Arch и не знаете об этих изменениях, пожалуйста, имейте в виду, что вашему беспроводному интерфейсу больше не будет присвоено имя `wlan0`, а проводному — `eth0`. Чтобы узнать корректные имена ваших сетевых интерфейсов используйте команду `ip link`.

При старте системы сетевой демон `dhcpcd` автоматически загружается и пытается установить проводное соединение. Попробуйте отправить запросы

(ping) на какой-нибудь сервер, чтобы проверить, было ли установлено соединение. Вот пример для одного из серверов Яндекса:

```
# ping -c 3 ya.ru
PING ya.ru (93.158.134.203) 56(84) bytes of data:
64 bytes from www.yandex.ru (93.158.134.203): icmp_req=1 ttl=54 time=62.4 ms
64 bytes from www.yandex.ru (93.158.134.203): icmp_req=2 ttl=54 time=63.0 ms
64 bytes from www.yandex.ru (93.158.134.203): icmp_req=3 ttl=54 time=62.4 ms
--- ya.ru ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 62.423/62.623/63.009/0.273 ms
```

Если вы увидите сообщение об ошибке `ping: unknown host`, прежде всего проверьте, связана ли проблема с вашим кабелем или уровнем беспроводного сигнала. Если с ними все в порядке, придется настраивать сеть вручную, как описано ниже. Как только соединение будет установлено, переходите к разделу [#Подготовка дисковых устройств](#).

Проводное

Следуйте этой процедуре, если вам нужно настроить проводное соединение со статическим IP-адресом.

Определите название вашего проводного интерфейса:

```
# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0f0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether 00:11:25:31:69:20 brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DORMANT qlen 1000
    link/ether 01:02:03:04:05:06 brd ff:ff:ff:ff:ff:ff
```

В данном примере, нужным интерфейсом является `enp2s0f0`. Если вы не уверены, помните, что скорее всего название будет начинаться с буквы "e", и вряд ли с "lo" или "w".

Смотрите требуемые настройки на странице [Настройка сети#Статический IP-адрес](#). Настройте статический профиль для `dhcpcd` в `/etc/dhcpcd.conf` следующим образом:

```
interface enp2s0f0
static ip_address=192.168.0.10/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8
```

Перезапустите службу `dhcpcd.service`:

```
# systemctl restart dhcpcd.service
```

Теперь у вас должно появиться работающее сетевое соединение. Если это не так, смотрите страницу [Настройка сети](#).

Беспроводное

Важно: Прошивки для беспроводных контроллеров (для тех, которым они нужны) предустановлены в каталог `/usr/lib/firmware` в live-среде (на установочном образе), **но должны быть установлены вручную в вашей системе, чтобы адаптер беспроводной сети смог работать после загрузки в нее!** Установка пакетов описана далее в этом руководстве. Убедитесь, что установлена как нужная прошивка, так и модуль ядра, прежде чем покидать live-среду. Смотрите [Настройка беспроводной сети](#), если вы не уверены насчет необходимости установки прошивки на ваш контроллер.

Используйте *wifi-menu* из состава `netctl` для соединения с беспроводной сетью:

```
# wifi-menu
```

Если у вас только один адаптер Wi-Fi, вы должны сразу увидеть список доступных сетей (обычно это так для ноутбуков).

Если у вас несколько адаптеров, вам необходимо явно передать присвоенное ему имя в качестве аргумента *wifi-menu*. Для начала определите название вашего беспроводного интерфейса:

```
# iw dev
phy#0
    Interface wlp3s0
        ifindex 3
        wdev 0x1
        addr 00:11:22:33:44:55
        type managed
```

В данном примере доступен беспроводной интерфейс под названием `wlp3s0`. Если вы не уверены, помните, что скорее всего название будет начинаться с буквы "w", и вряд ли с "lo" или "e".

Теперь запустите *wifi-menu*, указав имя сетевого интерфейса:

```
# wifi-menu wlp3s0
```

Для сетей, которые требуют одновременно имя пользователя и пароль, смотрите пример настройки в [WPA2 Enterprise#netctl](#).

Теперь у вас должно появиться работающее беспроводное сетевое соединение. Если это не так или вы не смогли определить имя интерфейса, смотрите [#Без использования wifi-menu](#) далее либо более подробное руководство на странице [Настройка беспроводной сети](#).

Без использования `wifi-menu`

Поднимите интерфейс командой:

```
# ip link set wlp3s0 up
```

Удостоверьтесь, что интерфейс поднят, изучив вывод следующей команды:

```
# ip link show wlp3s0
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state DOWN mode DORMANT group
link/ether 00:11:22:33:44:55 brd ff:ff:ff:ff:ff:ff
```

Слово `UP` в списке состояний `<BROADCAST,MULTICAST,UP,LOWER_UP>` — это то, что нужно; последующее `state DOWN` относится уже к самому соединению.

Большинству беспроводных контроллеров в дополнение к драйверу требуется прошивка. Ядро пытается автоматически определить контроллер и загрузить прошивку. Если вы видите что-то вроде `SIOCIFFLAGS: No such file or directory`, это значит, что необходимо загрузить прошивку вручную. Если вы не уверены, вызовите `dmesg`, чтобы просмотреть лог ядра на наличие запроса прошивки от контроллера беспроводной сети. Например, если у вашей карты контроллер Intel, которому нужна прошивка, и он ее запросил из ядра при загрузке системы:

```
# dmesg | grep firmware
firmware: requesting iwlwifi-5000-1.ucode
```

Если подобного сообщения нет, можно сделать вывод, что беспроводному контроллеру системы прошивка не требуется.

Теперь просканируйте доступные сети при помощи команды `iw dev wlp3s0 scan | grep SSID`, после чего подключитесь к одной из них при помощи:

```
# wpa_supplicant -B -i wlp3s0 -c <(wpa_passphrase "ssid" "пароль")>
```

При этом необходимо заменить `ssid` на название вашей сети, а `пароль` — на пароль, **оставив кавычки вокруг них**.

Наконец, вам осталось присвоить IP-адрес вашему интерфейсу. Его можно задать вручную либо он может быть присвоен автоматически при помощи

dhcр:

```
# dhcрcd wlp3s0
```

Если это не сработает, используйте следующие команды:

```
# echo 'ctrl_interface=DIR=/run/wpa_supplicant' > /etc/wpa_supplicant.conf
# wpa_passphrase "ssid" "пароль" >> /etc/wpa_supplicant.conf
# ip link set интерфейс up
# wpa_supplicant -B -D nl80211,wext -c /etc/wpa_supplicant.conf -i интерфейс
# dhcрcd -A интерфейс
```

Возможно, не будет необходимости поднимать интерфейс третьей командой, но, в любом случае, никакого вреда это не принесет.

Аналоговый модем, ISDN или PPPoE DSL

Информацию о подключениях xDSL, dial-up и ISDN смотрите в статье Direct modem connection.

Подключение через прокси

Если вы используете прокси-сервер, вам необходимо экспортировать переменные окружения `http_proxy` и `ftp_proxy`. Для получения дополнительной информации смотрите статью Proxy settings.

Подготовка дисковых устройств

На этом шаге производится подготовка устройств хранения, которые будут использоваться новой системой. Общий обзор смотрите в статье Разметка диска.

Важно: Разметка диска уничтожит имеющиеся данные. Перед тем, как продолжить, мы **настоятельно** рекомендуем сделать резервную копию всех важных данных.

Совет:

- Если вы хотите создать какие-либо совмещенные блочные устройства для LVM, шифрования диска или RAID, сделайте это сейчас.
- Если вы производите установку на флеш-накопитель USB, смотрите статью Installing Arch Linux on a USB key.

Определение устройств

Первым делом необходимо определить имена устройств хранения, на которые будет установлена новая система. Следующая команда отобразит все доступные устройства:

```
# lsblk
```

Она выводит список всех дисковых устройств в вашей системе вместе с разделами на них, включая устройства и разделы, на которых находится загрузочный образ Arch. Поэтому не все показанные устройства подойдут для установки. Вы можете воспользоваться следующей командой, чтобы отобразить только подходящие для этой цели устройства:

```
# lsblk | grep -v "rom\|loop\|airoot"
```

Жесткие диски будут отображены как `sd x` , где x — строчная латинская буква: a для первого устройства (`sda`), b — для второго (`sdb`) и так далее. Существующие разделы на этих устройствах отображаются в виде `sd x y` , где y — номер раздела, начиная с 1 для первого раздела, 2 — для второго и так далее. В нашем примере ниже доступно единственное устройство `sda` с одним разделом `sda1`:

```
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   80G  0 disk
└─sda1                8:1    0   80G  0 part
```

Обозначения вида `sd x y` используются далее для указания таблиц разделов, самих разделов и файловых систем в примерах. Помните, что необходимо заменить `sd x y` на реальное имя устройства; не следует просто вслепую копировать указанные здесь команды.

Если нет необходимости изменять текущую схему разметки, переходите сразу к разделу [#Создание файловых систем](#).

Типы таблиц разделов

Если вы устанавливаете Arch рядом с другой установленной системой (то есть, с двойной загрузкой), таблица разделов уже создана и используется этой системой. Если устройства не размечены, либо требуется изменить существующую таблицу разделов и/или разметку, первым делом следует определиться с выбором таблицы разделов (одна для каждого из устройств).

Важно: Если Arch и Windows загружаются с одного диска, Arch **должен** использовать тот же режим загрузки (BIOS/UEFI), что и Windows, в противном случае Windows не загрузится. Для получения дополнительной информации обратитесь к разделу [Двойная загрузка Windows и Arch#Важная информация](#).

Есть два типа таблиц разделов:

- MBR: для систем с BIOS (также известный как "msdos")
- GPT: для систем с UEFI

Для определения типа таблицы разделов на конкретном устройстве используйте следующую команду:

```
# parted /dev/sdx print
```

Инструменты разметки

Для каждого из устройств, которые необходимо разметить либо изменить существующую схему разметки необходимо выбрать подходящий инструмент в зависимости от типа используемой таблицы разделов. Установочный образ Arch предоставляет несколько таких инструментов:

- `parted`: MBR и GPT
- `fdisk`, **`cfdisk`**, **`sfdisk`**: MBR и GPT
- `gdisk`, **`cgdisk`**, **`sgdisk`**: GPT

Важно: Использование инструмента разметки, не совместимого с существующей таблицей разделов может ее испортить, что приведет к потере всех разделов и данных на устройстве.

Совет: Дисковые устройства можно разметить еще перед тем, как загружаться в установочный образ Arch, используя альтернативные live-образы с более удобными для вас инструментами разметки. Новичкам может быть проще использовать инструмент с графическим интерфейсом, например GParted, который также поставляется в виде live CD (<http://gparted.sourceforge.net/livecd.php>) и работает как с таблицами MBR, так и с GPT.

Использование `parted` в интерактивном режиме

Во всех примерах далее используется *parted*, так как он работает одновременно с BIOS/MBR и UEFI/GPT. Работа в интерактивном режиме упрощает процесс разметки и уменьшает вероятность совершения ошибок.

Чтобы начать работать с устройством, наберите:

```
# parted /dev/sdx
```

Приглашение командной строки поменяется с `#` на `(parted)`: это говорит о том, что `parted` готов принимать команды, то есть запущен в интерактивном режиме.

Список доступных команд можно отобразить командой `help`:

```
;(parted) help
```

После завершения работы наберите

```
;(parted) quit
```

Вы будете возвращены в командную оболочку (приглашение сменится на #), и теперь сможете выполнить разметку другого устройства, если нужно.

Создание новой таблицы разделов

Для устройства, которое еще никогда не размечалось нужно создать новую таблицу разделов. Также новую таблицу придется создать, если вы хотите использовать таблицу другого типа, нежели существующая или просто выполнить разметку "с нуля".

Важно:

- При двойной загрузке с уже установленной Windows на системе с UEFI/GPT не стирайте таблицу разделов. Это уничтожит все существующие данные, включая раздел UEFI с файлом *.efi*, который требуется для загрузки Windows.
- MBR разработан специально для использования с системами BIOS, а GPT — для UEFI. Не рекомендуем вам нарушать этот порядок, если вы не являетесь опытным пользователем. Если все же хочется, подробности и решения проблем смотрите на страницах <http://mjpg59.dreamwidth.org/8035.html> и <http://rodsbooks.com/gdisk/bios.html>.

Откройте устройство, для которого нужно (пере-)создать таблицу разделов:

```
# parted /dev/sdx
```

Чтобы создать новую таблицу разделов MBR для систем с BIOS, используйте команду:

```
;(parted) mklabel msdos
```

А для создания таблицы GPT для систем с UEFI, наберите:

```
;(parted) mklabel gpt
```

Схема разметки

В зависимости от ваших потребностей вы решаете, на какое количество разделов должен быть разбит диск и какие каталоги системы будут вынесены на отдельные разделы (известные как *точки монтирования*). Набор разделов и соответствующих им точек монтирования называется схемой разметки, которая должна следовать следующим требованиям:

- Должен быть создан по крайней мере один раздел, который будет использоваться для корневого / (*root*) каталога.
- В зависимости от интерфейса прошивки материнской платы, выбранного типа таблицы разделов, а в некоторых случаях и загрузчика, потребуется создать следующие дополнительные разделы:
 - **BIOS/MBR**: дополнительных разделов не требуется.
 - **BIOS/GPT**:
 - При использовании `syslinux`: дополнительных разделов не требуется.
 - При использовании `GRUB`: один раздел размером 1 Мбайт или 2 Мбайта загрузочный раздел BIOS типа `ef02`.
 - **UEFI/GPT**: один системный раздел EFI.
 - **UEFI/MBR**: один системный раздел EFI.

In the examples below it is assumed that a new and contiguous partitioning scheme is applied to a single device. Some optional partitions will also be created for the `/boot` and `/home` directories: see also Arch filesystem hierarchy for an explanation of the purpose of the various directories; if separate partitions for directories like `/boot` or `/home` are not created, these will simply be contained in the `/` partition. Also the creation of an optional partition for swap space will be illustrated.

Если вы еще не начали разметку диска в интерактивном режиме утилиты `parted`, запустите ее командой:

1. `parted /dev/sdx`

Для создания разделов используется следующая команда:

```
;(parted) mkpart part-type fs-type start end
```

- *part-type* — одно из значений: `primary`, `extended` или `logical`, для создания первичного, расширенного и логического раздела соответственно в таблице MBR.
- *fs-type* — тип файловой системы для раздела, одно из значений из списка [2] (<http://www.gnu.org/software/parted/manual/parted.html#mkpart>). Это всего лишь специальная метка, фактически сам раздел будет отформатирован на шаге `#Создание файловых систем`.
- *start* — позиция точки начала раздела относительно нулевого байта. Она состоит из числа, за которым следует единица измерения (<http://www.gnu.org/software/parted/manual/parted.html#unit>). Например, `1M` означает, что раздел начинается со смещения 1 Мбайт.
- *end* — позиция точки конца раздела относительно нулевого байта (а не значения *start*). Имеет тот же синтаксис, что и *start*, например `100%` означает, что раздел заканчивается в самом конце устройства.

Важно: Разделы не должны перекрываться друг другом. Если вы хотите использовать все возможное пространство, убедитесь, что каждый раздел начинается в том же месте, где заканчивается предыдущий.

Примечание: `parted` может выводить предупреждение:

```
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel?
```

В этом случае, прочтите [Partitioning#Partition alignment](#) и следуйте инструкциям на странице [GNU Parted#Alignment](#) для того, чтобы это исправить.

Следующая команда используется для того, чтобы пометить раздел с каталогом `/boot` как загрузаемый:

```
(parted) set partition boot on
```

- *partition* — номер раздела, который нужно пометить (смотрите вывод команды `print`).

Примеры для UEFI/GPT

Для загрузки с UEFI в любом случае потребуется создание специального загрузаемого Системного раздела EFI.

Важно: В случае использования UEFI/GPT при двойной загрузке с уже установленной системой Windows нельзя удалять уже созданный раздел UEFI. Если вы это сделаете, вы больше не сможете загрузаться в Windows.

Для создания нового системного раздела EFI используйте следующие команды (рекомендуется создавать раздел на 512 Мбайт):

```
(parted) mkpart ESP fat32 1M 513M
(parted) set 1 boot on
```

Остальную схему разметки вы устанавливаете самостоятельно. Например, для создания единственного раздела, который займет все оставшееся место, выполните:

```
(parted) mkpart primary ext3 513M 100%
```

Чтобы создать корневой раздел (`/`) на 20 Гбайт, а `/home` вынести в отдельный:

```
(parted) mkpart primary ext3 513M 20.5G
(parted) mkpart primary ext3 20.5G 100%
```

Или, кроме того, создать раздел подкачки на 4 Гбайта:

```
(parted) mkpart primary ext3 513M 20.5G
(parted) mkpart primary linux-swap 20.5G 24.5G
(parted) mkpart primary ext3 24.5G 100%
```

Примеры для BIOS/MBR

Для создания единственного первичного раздела, занимающего все доступное пространство, выполните:

```
(parted) mkpart primary ext3 1M 100%
(parted) set 1 boot on
```

Чтобы создать корневой первичный раздел (/) на 20 Гбайт, а /home вынести в отдельный:

```
(parted) mkpart primary ext3 1M 20G
(parted) set 1 boot on
(parted) mkpart primary ext3 20G 100%
```

Или, кроме того, вынести /boot в отдельный первичный раздел на 100 Мбайт и создать первичный раздел подкачки на 4 Гбайта:

```
(parted) mkpart primary ext3 1M 100M
(parted) set 1 boot on
(parted) mkpart primary ext3 100M 20G
(parted) mkpart primary linux-swaps 20G 24G
(parted) mkpart primary ext3 24G 100%
```

Создание файловых систем

После завершения разметки диска необходимо отформатировать каждый из созданных разделов (за исключением разделов подкачки), создав в них одну из файловых систем. Вы можете отобразить список всех доступных разделов на конкретном устройстве командой:

```
# lsblk /dev/sdx
```

За некоторыми исключениями, о которых написано чуть ниже, рекомендуется отформатировать все разделы в систему ext4 :

```
# mkfs.ext4 /dev/sdxY
```

Важно: В случае использования UEFI/GPT при двойной загрузке с уже установленной системой Windows, не форматируйте системный раздел UEFI. Если вы это сделаете, вы больше не сможете загружаться в Windows.

Примечание:

- Если вы создали новый системный раздел UEFI, обязательно отформатируйте его в систему fat32 или vfat32 :

```
# mkfs.vfat -F32 /dev/sdxY
```

Если этого не сделать, вы не сможете загрузиться в установленную систему.

- Если вы хотите использовать GRUB с конфигурацией BIOS/GPT, имейте в виду, что загрузочный раздел BIOS не имеет ничего общего с точкой монтирования `/boot`. Он будет использоваться GRUB напрямую. Не создавайте файловую систему на нем и не монтируйте его.

Включение раздела подкачки

Если был создан раздел подкачки, нужно создать файл подкачки и включить его командами:

```
# mkswap /dev/sdxY
# swapon /dev/sdxY
```

Монтирование разделов

Примечание: Не монтируйте разделы подкачки.

Корневой раздел (`/`) должен быть смонтирован первым, так как он содержит точки монтирования для остальных разделов (например, если содержимое `/boot` или `/home` вынесено на отдельные разделы). Каталог `/mnt` live-окружения используется для монтирования корневого раздела, а затем уже в него монтируются остальные. Чтобы смонтировать корневой раздел `sdxR`, наберите:

```
# mount /dev/sdxR /mnt
```

Как только корневой раздел будет смонтирован, оставшиеся разделы можно будет смонтировать в любом порядке. Для этого первым делом нужно создать каталоги, выступающие в роли точек монтирования, и затем уже монтировать в них сами разделы. Например, если `/boot` вынесен на отдельный раздел:

```
# mkdir -p /mnt/boot
# mount /dev/sdxB /mnt/boot
```

Примечание: Использование `/boot` также рекомендуется для монтирования системного раздела EFI в конфигурации UEFI/GPT. Подробнее смотрите статью EFISTUB и другие статьи по теме.

Для `/home` на отдельном разделе, аналогично:

```
# mkdir -p /mnt/home
# mount /dev/sdxH /mnt/home
```

После монтирования всех оставшихся разделов, вы будете готовы приступить к установке Arch Linux.

Выбор зеркала

Вы можете отредактировать файл `mirrorlist` и поместить в нем подходящее зеркало репозитория Arch на первое место. Копия этого файла будет установлена в вашу новую систему скриптом *pacstrap*, так что стоит уделить этому внимание на данном этапе.

```
# nano /etc/pacman.d/mirrorlist
###
### Arch Linux repository mirrorlist
### Sorted by mirror score from mirror status page
### Generated on YYYY-MM-DD
###
Server = http://mirror.example.xyz/archlinux/$repo/os/$arch
...
```

Если хотите, можете оставить *только одно* доступное зеркало, удалив все другие строки, однако лучше когда указано несколько зеркал на тот случай, если первое из них станет недоступно.

Совет:

- Используйте Mirrorlist Generator (<https://www.archlinux.org/mirrorlist/>), чтобы получить список подходящих зеркал в зависимости от вашего географического положения. Зеркала HTTP быстрее, чем зеркала FTP из-за использования постоянного HTTP-соединения: в случае с FTP каждый раз, когда *pacman* посылает новый запрос на загрузку следующего пакета, он должен вновь устанавливать соединение с сервером, что вызывает задержки. Чтобы узнать о других способах генерации списка зеркал, смотрите раздел Зеркала#Приоритет зеркал и статью Reflector (Русский).
- На странице Arch Linux MirrorStatus (<https://archlinux.org/mirrors/status/>) приведена информация о доступности официальных зеркал и текущему состоянию их обновления.

Примечание:

- При любом изменении списка зеркал всегда обновляйте все списки пакетов при помощи команды `pacman -Syu`. Для получения дополнительной информации смотрите статью Зеркала.
- Если вы используете устаревший установочный образ, ваш список зеркал также может оказаться устаревшим, что может привести к проблемам с обновлениями Arch Linux (смотрите отчет об ошибке FS#22510 (<https://bugs.archlinux.org/task/22510>)). В связи с этим рекомендуется первым делом обновить свой список зеркал, как описано выше.

Установка базовой системы

Базовая система устанавливается при помощи скрипта *pacstrap*. Можно опустить флаг `-i`, если вы хотите установить все пакеты из группы `base` (https://www.archlinux.org/groups/x86_64/base/) без запросов подтверждения. Вы также можете установить пакеты из группы `base-devel` (https://www.archlinux.org/groups/x86_64/base-devel/), поскольку они вам понадобятся для сборки пакетов из AUR или при использовании ABS:

```
# pacstrap -i /mnt base base-devel
```

Примечание:

- Если *pacstrap* выдает сообщение об ошибке `error: failed retrieving file 'core.db' from mirror... : Connection time-out`, ХОТЯ ваши зеркала настроены правильно, попробуйте использовать другой DNS-сервер.
- Если в середине процесса установки базовых пакетов вы получаете запрос на импорт ключа PGP, для продолжения согласитесь скачать этот ключ. Чаще всего это происходит из-за использования устаревшего Arch ISO. Если добавить ключ PGP не удастся, попробуйте обновить пакет `archlinux-keyring` (<https://www.archlinux.org/packages/?name=archlinux-keyring>) при помощи команды `pacman -S archlinux-keyring`

При возникновении проблем смотрите также [Pacman \(Русский\)#Решение проблем и Pacman-key#Troubleshooting](#).

Генерация fstab

Сгенерируйте файл `fstab` приведенной ниже командой. Рекомендуется использовать `UUID` для указания конкретного раздела (смотрите раздел [fstab \(Русский\)#Определение файловой системы](#)). Если вы предпочитаете использовать метки, замените опцию `-U` на `-L`:

```
# genfstab -U -p /mnt >> /mnt/etc/fstab
```

```
# nano /mnt/etc/fstab
```

Важно: После каждой генерации файла `fstab` его следует открыть для проверки. Если вы столкнетесь с ошибками при запуске `genfstab` или позже в процессе установки, **не** запускайте `genfstab` вновь: просто отредактируйте файл `fstab` вручную.

Последнее поле (`pass`) определяет порядок проверки разделов во время загрузки системы: используйте значение `1` только для корневого раздела, поскольку он должен проверяться самым первым, `2` — для всех других разделов, которые вы хотите проверять, и `0` — для обозначения 'не проверять' (смотрите `fstab (Русский)#Формат строки`). Обратите однако внимание, что **все** разделы `Btrfs` должны иметь в этом поле значение `0` (даже корневой). Обычно для раздела *подкачки* также устанавливается `0`.

chroot и настройка базовой системы

Теперь войдите в новый сеанс командной оболочки, используя корневой каталог установленной системы при помощи `arch-chroot`:

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

На этом этапе установки вам предстоит настроить основные файлы конфигурации своей базовой системы Arch Linux. Эти файлы можно создать, если они не существуют или отредактировать, если вы хотите изменить настройки по умолчанию.

Внимательное следование приведенным ниже шагам и осознание своих действий имеют ключевое значение для правильной настройки системы.

Важно: Не все инструменты, которые вы можете использовать в *live*-окружении, устанавливаются в новую систему. Например, если вы использовали `wifi-menu` для соединения с беспроводной сетью во время установки, и хотите сделать также, вам будет нужно установить `dialog` (<https://www.archlinux.org/packages/?name=dialog>) перед тем, как загружаться в новую систему. В следующем разделе отмечены подобные моменты, поэтому читайте его внимательно.

Локаль

Локали определяют, какой язык использует система, а также устанавливают прочие предпочтения вроде форматов денег, времени и даты, наборов символов и других локальных особенностей. Список используемых в системе локалей устанавливается в файле `locale.gen`, а одна из них выбирается в качестве активной в `locale.conf`.

Изначально в файле `locale.gen` все строки закомментированы (начинаются с символа `#`). Включите локаль `en_US.UTF-8 UTF-8` и прочие нужные вам локали,

раскомментировав соответствующую строку (достаточно убрать #).
Рекомендуется для каждой локали выбрать вариант с кодировкой UTF-8 .

```
# nano /etc/locale.gen
.....
#en_SG ISO-8859-1
en_US.UTF-8 UTF-8
#en_US ISO-8859-1
.....
ru_RU.UTF-8 UTF-8
#ru_RU ISO-8859-1
.....
```

Чтобы теперь выбранные локали можно было использовать, их нужно *сгенерировать*:

```
# locale-gen
```

Создайте файл `/etc/locale.conf` , указав выбранную вами локаль. Обратите внимание, что переменная `LANG` должна содержать только первую часть одной из выбранных строк в `/etc/locale.gen` , до пробела:

```
# echo LANG=en_US.UTF-8 > /etc/locale.conf
```

Также экспортируйте переменную в текущий сеанс командной оболочки:

```
# export LANG=en_US.UTF-8
```

Совет:

- Установка `en_US.UTF-8` в качестве общесистемной локали позволит вести системные логи на английском языке, что упрощает решение возможных проблем. Отдельные пользователи смогут установить нужную им локаль, настроив свое окружение, как указано в Локаль#Установка локали.
- `LANG` предоставляет значение по умолчанию для всех `LC_*`-переменных, которые отвечают за установки локалей для отдельных параметров среды. Чтобы использовать другие локали для этих переменных, запустите `locale` для отображения доступных параметров и добавьте их в `locale.conf` . Не рекомендуется устанавливать переменную `LC_ALL` . Подробности смотрите на странице Locale (Русский).

Консольный шрифт и раскладка клавиатуры

Если вы меняли консольный шрифт и раскладку клавиатуры по умолчанию во время этапа #Изменение языка, необходимо *соответствующим образом* отредактировать файл `/etc/vconsole.conf` (создайте его, если он не существует),

чтобы сделать эти изменения постоянными для вновь установленной системы. Например:

```
# nano /etc/vconsole.conf
KEYMAP=ru
FONT=cyr-sun16
```

Важно: Если вы установите для `KEYMAP` значение, отличное от того, что вы устанавливали ранее при помощи `loadkeys` и выполните инструкции из шага **#Установка пароля суперпользователя**, после перезагрузки у вас могут возникнуть проблемы со входом в новую систему, поскольку некоторые клавиши могут иметь в этих раскладках разные коды.

Имейте в виду, что эти настройки имеют значение только для ваших виртуальных консолей и не затрагивают `Xorg`. Для получения дополнительной информации смотрите раздел **Шрифты#Шрифт в консоли**.

Временная зона

Доступные временные зоны и субзоны можно найти в каталогах `/usr/share/zoneinfo/Зона/Субзона`, и отобразить командой `ls`. Создайте символическую ссылку `/etc/localtime` на файл вашей субзоны `/usr/share/zoneinfo/Зона/Субзона`:

```
# ln -s /usr/share/zoneinfo/Зона/Субзона /etc/localtime
```

Совет: Используйте дополнение по `tab` (<http://tldp.org/LDP/abs/html/tabexpansion.html>) для того, чтобы отображать доступные варианты зон при вводе.

Пример:

```
# ln -s /usr/share/zoneinfo/Europe/Moscow /etc/localtime
```

Если появилась ошибка

```
ln: failed to create symbolic link '/etc/localtime': File exists, значит ссылка уже существует. Проверьте ее командой ls -l /etc/localtime. Если она неправильная, вызовите ln с опцией -f для того, чтобы перезаписать файл ссылки.
```

Аппаратные часы

Все операционные системы должны воспринимать часовой пояс аппаратных часов одинаковым образом, иначе время будет сбиваться при синхронизации. Команда `hwclock` генерирует файл `/etc/adjtime`, который содержит соответствующие настройки. Для UTC запустите:

```
# hwclock --systohc --utc
```

Примечание: Использование UTC для аппаратных часов не означает, что программное обеспечение будет отображать время в UTC.

Важно: Системы Windows используют *localtime* по умолчанию. Использование *localtime* в Arch приводит к появлению ряду известных, но неразрешимых проблем; однако, отказ от опции *localtime* не планируется. Чтобы ее использовать, запустите `hwclock --systohc --localtime`. Для Windows 7 и более современных систем Windows рекомендуется использовать UTC и отключить синхронизацию времени. Смотрите [Time#UTC in Windows](#).

Модули ядра

Все необходимые модули автоматически подгружает `udev`, поэтому необходимость загружать модули вручную возникает редко. Подробнее смотрите на странице Модули ядра.

Имя узла (`hostname`)

Установите имя узла на свое усмотрение:

```
# echo мое_имя_узла > /etc/hostname
```

Добавьте то же самое имя узла в файл `/etc/hosts`:

```
#<ip-address> <hostname.domain.org> <hostname>
127.0.0.1 localhost.localdomain localhost myhostname
::1 localhost.localdomain localhost myhostname
```

Настройка сети

Вам необходимо заново настроить сеть для установленной системы. Принцип настройки очень похож на тот, что описан выше, за исключением того, что мы собираемся сохранить настройки соединения для его автоматической установки после перезагрузки.

Первым делом определите имя сетевого интерфейса, для которого вы хотите настроить соединение при помощи `ip link`.

Примечание:

- Для получения более подробной информации о настройке сети посетите страницы [Настройка сети](#) и [Настройка беспроводной сети](#).
- Если вы желаете использовать старую схему именования интерфейсов (то есть, `ethX` и `wlanX`), вы можете создать пустой файл `/etc/udev/rules.d/80-net-setup-link.rules`, который будет перекрывать собой

файл с таким же именем в каталоге `/usr/lib/udev/rules.d`.

Проводное

Динамический IP

Используя `dhcpcd`

Если вы используете только одно фиксированное проводное подключение к сети, у вас нет необходимости использовать какой-либо сетевой менеджер, достаточно просто включить службу `dhcpcd` для конкретного интерфейса:

```
# systemctl enable dhcpcd@имя_интерфейса.service
```

Используя `netctl`

Скопируйте образец профиля из каталога `/etc/netctl/examples` в `/etc/netctl`:

```
# cd /etc/netctl
# cp examples/ethernet-dhcp my_network
```

Отредактируйте профиль, как вам нужно (измените строку `Interface`, заменив `eth0` на имя интерфейса системы):

```
# nano my_network
```

Включите профиль `my_network`:

```
# netctl enable my_network
```

Примечание: Вы увидите сообщение об ошибке "Running in chroot, ignoring request". Сейчас его можно проигнорировать.

Используя `netctl-ifplugd`

Важно: Вы не можете использовать этот способ одновременно с включением профилей в явном виде, например командой `netctl enable профиль`.

В качестве альтернативы вы можете использовать `netctl-ifplugd`, который автоматически устанавливает соединение при подключении сетевого кабеля.

Установите пакет `ifplugd` (<https://www.archlinux.org/packages/?name=ifplugd>), необходимый для работы `netctl-ifplugd`:

```
# pacman -S ifplugd
```

Затем включите службу для нужного сетевого интерфейса:

```
# systemctl enable netctl-ifplugd@интерфейс.service
```

Совет: netctl также предоставляет службу netctl-auto, которую можно использовать для автоматического соединения с беспроводными сетями.

Статический IP

Используя netctl

Скопируйте образец профиля из каталога /etc/netctl/examples в /etc/netctl:

```
# cd /etc/netctl
# cp examples/ethernet-static my_network
```

Отредактируйте профиль, как вам нужно (измените значения Interface, Address, Gateway и DNS):

```
# nano my_network
```

Будьте внимательны при указании маски подсети в строке Address (/24 в примере означает маску 255.255.255.0), иначе профиль не запустится. Смотрите также раздел адресация CIDR.

Включите созданный выше профиль в автозагрузку при каждом старте системы:

```
# netctl enable my_network
```

Используя systemd-networkd

Смотрите статью systemd-networkd (Русский).

Беспроводное

Примечание: Если вашему беспроводному адаптеру требуется прошивка (как было указано в разделе Настройка интернет-соединения, а также в Настройка беспроводной сети#Драйвер устройства), установите пакет, содержащий ее. Большинство прошивок содержится в пакете linux-firmware (<https://www.archlinux.org/packages/?name=linux-firmware>). Однако, для некоторых устройств может потребоваться собственный пакет. Например:

```
# pacman -S zd1211-firmware
```

Для получения дополнительной информации смотрите раздел Настройка беспроводной сети#Установка драйвера/прошивки.

Установите пакеты `iw` (<https://www.archlinux.org/packages/?name=iw>) и `wpa_supplicant` (https://www.archlinux.org/packages/?name=wpa_supplicant), которые потребуются для подключения к сети:

```
# pacman -S iw wpa_supplicant
```

Добавление беспроводных сетей

Используя `wifi-menu`

Установите пакет `dialog` (<https://www.archlinux.org/packages/?name=dialog>), необходимый для работы `wifi-menu`:

```
# pacman -S dialog
```

После завершения оставшейся части процесса установки и перезагрузки вы сможете подключиться к сети при помощи `wifi-menu имя_интерфейса` (где `имя_интерфейса` — интерфейс вашего беспроводного адаптера).

```
# wifi-menu имя_интерфейса
```

Важно: Это необходимо делать **после** перезагрузки, когда вы покинете окружение `chroot`. Процесс, порожденный этой командой, будет конфликтовать с тем, который вы запустили вне `chroot`. Вместо этого вы можете просто настроить профиль сети вручную как указано далее, тогда вам вообще не придется использовать `wifi-menu`.

Используя `netctl`

Скопируйте образец профиля из каталога `/etc/netctl/examples` в `/etc/netctl`:

```
# cd /etc/netctl
# cp examples/wireless-wpa my-network
```

Отредактируйте профиль, как вам нужно (измените значения `Interface`, `ESSID` и `Key`):

```
# nano my-network
```

Включите созданный выше профиль для автоматической установки соединения при старте системы:

```
# netctl enable my-network
```

Автоматическое подключение к известным сетям

Важно: Вы не можете использовать этот способ одновременно с включением профилей в явном виде, например командой `netctl enable профиль`.

Установите пакет `wpa_actiond` (https://www.archlinux.org/packages/?name=wpa_actiond), необходимый для работы `netctl-auto`:

```
# pacman -S wpa_actiond
```

Включите службу `netctl-auto`, которая будет автоматически подключаться к известным сетям в зоне покрытия:

```
# systemctl enable netctl-auto@имя_интерфейса.service
```

Совет: `netctl` также предоставляет службу `netctl-ifplugd`, которую можно использовать для автоматического соединения с проводными сетями.

Аналоговый модем, ISDN или PPPoE DSL

Информацию о подключениях xDSL, dial-up и ISDN смотрите в статье [Direct modem connection](#).

Создание начального окружения виртуального диска

Совет: Большинство пользователей могут пропустить данный шаг и использовать стандартные настройки в `mkinitcpio.conf`. Образ `initramfs` (из каталога `/boot`) уже был сгенерирован на основе данного файла, когда ранее при помощи `pacstrap` был установлен пакет `linux` (<https://www.archlinux.org/packages/?name=linux>) (ядро Linux).

Вам нужно правильно настроить хуки если корневой каталог находится на USB-накопителе, в RAID-массиве, LVM, разделах `Btrfs` на разных накопителях, или если `/usr` вынесен в отдельный раздел.

Отредактируйте файл `/etc/mkinitcpio.conf` так, как вам нужно, и регенерируйте образ `initramfs` командой:

```
# mkinitcpio -p linux
```

Установка пароля суперпользователя

Установите пароль суперпользователя командой:

```
# passwd
```

Установка и настройка загрузчика

Для материнских плат с BIOS

Для систем с BIOS доступно несколько загрузчиков, для просмотра полного списка смотрите статью [Boot loaders](#). Возможно, вам подойдут эти загрузчики:

- Syslinux (в настоящее время) имеет ограничение: он может загружать только файлы с того раздела, на котором установлен. Считается, что его конфигурационный файл легче понять. Пример конфигурации можно найти в разделе [Syslinux \(Русский\)#Примеры](#).
- GRUB имеет больше возможностей и более гибок в настройке. Его конфигурационные файлы больше похожи на 'sh'-скрипты, из-за чего начинающим пользователям может быть трудно с ними работать. Таким пользователям рекомендуется использовать инструменты для автоматического генерирования конфигурационных файлов.

Здесь мы покажем пример установки **GRUB** при использовании таблицы разделов **MBR**. Установите пакет `grub` (<https://www.archlinux.org/packages/?name=grub>) и запустите `grub-install` для установки загрузчика:

```
# pacman -S grub
# grub-install --target=i386-pc --recheck /dev/sdx
```

Примечание:

- Не забудьте заменить `/dev/sdx` на реальное имя устройства, на котором установлен Arch. Не указывайте номер раздела (то есть, не пишите `sdxY`).
- Файл `/boot/grub/grub.cfg` с примерами конфигурации также устанавливается с пакетом `grub` (<https://www.archlinux.org/packages/?name=grub>), и впоследствии команды `grub-*` могут его не перезаписать. Удостоверьтесь, что ваши изменения сохраняются в файле `grub.cfg`, а не в `grub.cfg.new` или каком-либо другом.

Автоматически сгенерируйте `grub.cfg` :

Совет: Для автоматического поиска на вашем компьютере других операционных систем перед запуском следующей команды установите пакет `os-prober` (<https://www.archlinux.org/packages/?name=os-prober>) (`pacman -S os-prober`).

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

Для получения дополнительной информации о настройке и использовании GRUB смотрите статью [GRUB \(Русский\)](#).

Для материнских плат с UEFI

Для систем с UEFI доступно несколько загрузчиков, полный список смотрите на странице [Boot loaders](#). Возможно, вам подойдут эти загрузчики:

- **gummiboot** (Русский) — минималистичный менеджер загрузки UEFI, предоставляющий меню для ядер EFISTUB и других приложений UEFI. Рекомендуется новичкам, особенно тем, кто желает иметь возможность загружаться и в другие установленные операционные системы, например Windows 8.
- GRUB — более полноценный загрузчик, полезный, если у вас возникают проблемы с Gummiboot.

Здесь мы покажем пример установки *gummiboot*. Сперва установите пакет *dosfstools* (<https://www.archlinux.org/packages/?name=dosfstools>), чтобы вы могли управлять системным разделом EFI после установки, а также *efibootmgr* (<https://www.archlinux.org/packages/?name=efibootmgr>) для создания загрузочной записи UEFI (которая используется установочными скриптами менеджера загрузки):

```
# pacman -S dosfstools efibootmgr
```

Примечание:

- Для загрузки в режиме UEFI жесткий диск должен быть размечен в GPT и иметь системный раздел EFI (512 Мбайт или больше, с типом раздела `ef00` и отформатированный в FAT32). В следующих примерах этот раздел монтируется в каталог `/boot`. Если вы следовали этому руководству с самого начала, у вас уже должно быть правильно все сделано.
- Очень желательно, чтобы системный раздел EFI был смонтирован в `/boot`, так как это требуется для автоматического обновления Gummiboot.

Установите пакет *gummiboot* (<https://www.archlinux.org/packages/?name=gummiboot>) и запустите автоматический установочный скрипт, заменив `$esp` на место расположения вашего системного раздела EFI (обычно это `/boot`):

```
# pacman -S gummiboot
# gummiboot --path=$esp install
```

По файлам `.efi` в `$esp/EFI/boot` прошивкой устанавливается наличие операционных систем. Для Gummiboot необходимо наличие загрузочного файла `bootx64.efi`. Также будет необходимо вручную создать файлы настроек для Gummiboot.

Сперва создайте файл `$esp/loader/entries/arch.conf` и добавьте в него следующее, заменив `/dev/sdax` на ваш **корневой раздел** (например, `/dev/sda1`):

```
# nano $esp/loader/entries/arch.conf
title          Arch Linux
linux          /vmlinuz-linux
```

```
initrd      /initramfs-linux.img
options     root=/dev/sdax rw
```

Затем создайте `$esp/loader/loader.conf` со следующим содержимым, заменив значение тайм-аута (опция `timeout`, указывается в секундах) на желаемое:

```
# nano $esp/loader/loader.conf
-----
default arch
timeout 5
```

Для получения дополнительной информации о настройке и использовании `gummiboot` смотрите статью `gummiboot` (Русский).

Размонтирование разделов и перезагрузка

Выйдите из окружения `chroot`:

```
# exit
```

Примечание: При выключении компьютера `systemd` автоматически размонтирует ваши разделы, но в целях безопасности вы можете сделать это вручную при помощи команды `umount -R /mnt`. Если раздел оказывается занят, вы можете использовать `fuser` для поиска причины.

Перезагрузите компьютер:

```
# reboot
```

Совет: Не забудьте вынуть установочный носитель, иначе вы снова попадете в live-окружение установочного образа. Вы можете войти в установленную систему в качестве суперпользователя (`root`), используя тот пароль, который вы указали при вызове `passwd`. Если пароль не был указан, то вход происходит без его запроса.

После установки

Теперь ваша новая базовая система Arch Linux является полностью функциональным окружением GNU/Linux, которое будет основой для создания системы, соответствующей вашим желаниям и потребностям. *Настоятельно* советуем вам прочитать статью `Основные рекомендации`, особенно два первых раздела. В остальных ее разделах представлены ссылки на руководства по послеустановочной настройке, например, настройке графического пользовательского интерфейса, звука и тачпада.

Для просмотра списка приложений, которые вам могут быть интересны, смотрите статью `Список приложений`.

Retrieved from "[https://wiki.archlinux.org/index.php?title=Beginners%27_guide_\(Русский\)&oldid=448498](https://wiki.archlinux.org/index.php?title=Beginners%27_guide_(Русский)&oldid=448498)"

Categories: [Getting and installing Arch \(Русский\)](#) | [Русский](#)

- This page was last modified on 28 August 2016, at 20:19.
- Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.