

Документация Calmira GNU/Linux

Автор: [Михаил Краснов](#)

В данном разделе приведена документация дистрибутива Calmira GNU/Linux (текущая версия создавалась для Calmira LX4 1.1 GNU/Linux). Отсюда вы узнаете о распаковке дистрибутива из squashfs снимка, копировании на предварительно заготовленный раздел, созданию LiveUSB образа и запаковку в образ виртуального жёсткого диска для Qemu/KVM (*qcow2*).

Дата написания: 27 июн. 2021

Дата принятия: 05 авг. 2021

Версия: 2021.1

Для версии Calmira: 2021.5 (LX4 1.1)

Если дате принятия больше года, значит, вероятно, есть более новая версия этого руководства. Посетите раздел документации, чтобы убедиться в этом.

Обратите внимание, что дистрибутив позиционируется для продвинутого пользователя, собственно, и это руководство тоже. Оно может быть непонятно начинающим пользователям, поэтому советуем чаще обращаться к поисковым системам [Google](https://www.google.com), [Yandex](https://www.yandex.ru) и другим для поиска дополнительной информации, которой нет в этом руководстве.



Copyright (C) 2021 [Михаил Linuxoid85 Краснов](#)

Предоставляется разрешение копировать, распространять и/или изменять этот документ на условиях Лицензии свободной документации GNU, версии 1.3 или любой более поздней версии, опубликованной Фондом свободного программного обеспечения; к Неизменяемым разделам относится "Манифест разработчиков Calmira", без Текстов для передней стороны обложки и без Текстов для задней стороны обложки. Копия лицензии помещена в разделе, озаглавленном "Лицензия свободной документации GNU".;

Содержание

- 1.0 [Принятые обозначения](#)
- 1.1 [О дистрибутиве](#)
- 1.2 [История развития Calmira GNU/Linux](#)
- 2.0 [Установка Calmira](#)
 - 2.1 [Загрузка дистрибутива из репозитория GitHub](#)
 - 2.2 [Распаковка дистрибутива, форматирование раздела и копирование на него системы](#)
 - 2.3 [Установка системы на ПК](#)
 - 2.4 [Установка системы в Qemu/KVM](#)
 - 2.5 [Первичная настройка дистрибутива после копирования](#)
 - 2.6 [Завершение установки. Выход из chroot и перезагрузка](#)
 - 2.7 [Вход в систему](#)
- 3.0 [Настройка дистрибутива](#)
 - 3.1 [Настройка окружения](#)
 - 3.2 [Настройка часов](#)
 - 3.3 [Настройка клавиатуры и шрифта в TTY](#)
 - 3.4 [Настройка inittab](#)
 - 3.5 [Установка дополнительных загрузочных скриптов](#)
- 4.0 [Управление пакетами](#)
 - 4.1 [Введение в порты](#)
 - 4.2 [Введение в srkg](#)

- 4.3 [Создание порта](#)
- 4.4 [Создание бинарного пакета для srkg](#)

5.0 [Администрирование системы](#)

- 5.1 [Права доступа](#)
- 5.2 [Управление процессами](#)

6.0 [Дополнительные материалы](#)

- 6.1 [Манифест разработчиков Calmira](#)
- 6.2 [GNU GPLv3](#)
- 6.3 [Лицензия свободной документации GNU](#)

1.0. Принятые обозначения

В данном руководстве используется несколько типографских обозначений.

Используется для отображения кода или команды, которую нужно ввести в терминал/редактор

Используется для отображения возможного вывода программы в терминал/файл или отображения содержимого предполагаемого/существующего файла

Используется для всевозможных пояснений

Используется для выделения важной информации

Используется для выделения каких-либо терминов, команд или имён файлов (только в абзацах, списках и таблицах).

Такой вариант так же приемлем.

Для этого используются теги `<tt> ... </tt>` и `<pre> ... </pre>`

▫ **Смотрите также:**

Используется для акцентирования внимания на каких-либо важных или дополнительных сведениях.

1.1. О Calmira GNU/Linux

Дистрибутив Calmira GNU/Linux задумывался как классический легковесный дистрибутив для гиков. Предоставляется минимальное окружение, предназначенное для дальнейшей настройки.

Calmira создавалась и сейчас разрабатывается одним человеком. Больше в команду никто не требуется, однако от помощи никто не откажется. На данный момент в процессе сборки новый релиз - Calmira LX4 1.1. В нём будет очень много изменений по сравнению с предыдущими релизами. Версию LX4 1.1 можно считать первой полностью самостоятельной версией дистрибутива, так как он почти не зависит от LX4 и других проектов, а разрабатывается самостоятельно. Утверждён новый цикл разработки, позволяющий сфокусироваться на исправлении ошибок релиза.

1.1.1. Причины появления дистрибутива

Calmira создавалась с несколькими целями. Во-первых, предоставление пользователю легковесной системы, которая смогла бы работать на оборудовании начала двухтысячных. Большинство "легковесных" дистрибутивов начинало отжирать столько ресурсов ПК, сколько не снилось даже самому жирному дистрибутиву. Разработчику Calmira же это не нравилось.

Во-вторых, предоставление классического "олдскульного" окружения, которое нравилось в своё время матёрым линуксоидам. В некоторых случаях это даже лучше современного. Многие к такому окружению привыкли, многие не хотят менять его на что-то новое. Хотя и сопротивление некоторым удачным новым технологиям является бессмысленной тратой времени, так что в дистрибутиве используется несколько весьма современных вещей.

В-третьих, недовольство современными дистрибутивами. Одни не уважают частную жизнь и свободу пользователя, другие очень медленные и нестабильные, у третьих ужасные системные компоненты и т.д. Поэтому неплохо было бы сделать дистрибутив "для себя", "для души", и поделиться наработками с сообществом, в котором не было тех недостатков. Так же дистрибутив должен быть надёжным. Конечно, на данный момент, Calmira не такая надёжная, как планировалось, но все баги и недоработки исправляются, дистрибутив активно развивается. О процессе разработки вы можете читать в главе "Дополнительные материалы".

В-четвёртых, в большинстве дистрибутивов Linux было установлено большое число абсолютно ненужного ПО, которое только занимало место, но не более. В новой системе Calmira этого же быть не должно. Она должна быть максимально лёгкой и без лишнего мусора, но, одновременно, и не в ущерб функциональности. Более того - в Calmira предоставляется только свободное/открытое ПО, что так же неплохо опытным пользователям.

Calmira была названа в честь одноимённой графической оболочки для Windows 3.11. Первоначально планировалось назвать дистрибутив "CalmiraOS", но через некоторое время было принято название "Calmira GNU/Linux".

Больше информации можете узнать в [истории](#) дистрибутива.

1.1.2. Позиционирование

Сейчас дистрибутив позиционируется для опытных пользователей или гиков, а так же владельцев слабого железа, так как Calmira очень слабо нагружает ПК. В дистрибутиве предоставлена большая свобода действий для пользователя. Большое число компонентов системы поддаётся настройке. Вы можете удалить один компонент системы, заменив его на другой.

1.1.3. План выхода релизов

Совсем недавно был сформирован план выхода релизов. Точных дат выхода нет, так как не всегда можно уложиться в поставленные сроки. Однако, тестовая версия дистрибутива выходит 5 числа определённого месяца, а стабильная - 15 числа (не обязательно того же месяца). Перед выходом тестовой версии на GitHub выкладывается временный инструментарий, с помощью которого собиралась система (если таковой инструментарий был создан, чаще всего используются давно созданные). 5-го числа выкладывается тестовая версия системы, а 15 (не обязательно того же месяца) - окончательный релиз. Всё это можно найти в разделе [Releases](#).

Кандидаты в релизы обычно не выкладываются на GitHub, а сохраняются в архиве у автора. Тестирование релиз-кандидатов закрытое. О внесении правок и изменений не известно никому. Однако, если есть необходимость публикации RC-версии, то эта версия будет опубликована.

1.1.4. Обновления пакетов

На данный момент, обновления пакетов предоставляются только для текущего релиза. Наладить подобный механизм в предыдущих версиях затруднительно.

[Назад](#)

Принятые обозначения

[Домой](#)

[Далее](#)

История развития Calmira
GNU/Linux

1.2. История развития Calmira GNU/Linux

Calmira GNU/Linux - молодой дистрибутив, разрабатываемый в одиночку бывшим владельцем нескольких небольших сообществ. Этот дистрибутив не такой удобный и функциональный, однако он развивается, хоть и медленно, а так же активно используется некоторыми людьми в своих целях.

1.2.1. Цели появления

Если дистрибутив создавался "просто так", без целей, то он не будет нужен никому.

Calmira создавалась с несколькими целями. Во-первых, предоставление пользователю легковесной системы, которая смогла бы работать на оборудовании начала двухтысячных. Большинство "легковесных" дистрибутивов начинало отжирать столько ресурсов ПК, сколько не снилось даже самому жирному дистрибутиву. Разработчику Calmira же это не нравилось.

Во-вторых, предоставление классического "олдскульного" окружения, которое нравилось в своё время матёрым линуксоидам. В некоторых случаях это даже лучше современного.

В-третьих, недовольство современными дистрибутивами. Одни не уважают частную жизнь и свободу пользователя, другие очень медленные и нестабильные, у третьих ужасные системные компоненты и т.д. Поэтому неплохо было бы сделать дистрибутив "для себя", "для души", и поделиться наработками с сообществом. Так же дистрибутив должен быть надёжным. Конечно, на данный момент, Calmira не такая надёжная, как планировалось, но все баги и недоработки исправляются, дистрибутив активно развивается.

В-четвёртых, в большинстве дистрибутивов Linux было установлено большое число абсолютно ненужного ПО, которое только занимало место, но не более. В новой системе Calmira этого же быть не должно. Она должна быть максимально лёгкой и без лишнего мусора, но, одновременно, и не в ущерб функциональности.

Calmira была названа в честь одноимённой графической оболочки для Windows 3.11. Первоначально планировалось назвать дистрибутив "CalmiraOS", но через некоторое время было принято название "Calmira GNU/Linux".

1.2.2. Первые версии

2021.1 - первый блин комом

Версия 2021.1 вышла 23 апреля 2021 года. Собиралась тогда по LFS 9.0. На данный момент образ с системой является утерянным. Система выдачи версий дистрибутиву была простой. До точки - год выпуска дистрибутива, после (мажорная версия) - номер версии. В некоторых релизах добавляется ещё одна точка и число после неё (минорная версия). Просто небольшое обновление с исправлением багов (например, 2021.2.1).

Первая версия (2021.1) содержала в себе очень много багов и недоработок, начиная от нерабочего `gmp`, заканчивая отсутствия какого-либо инструмента для работы с пакетами (кроме систем сборки, конечно). Потом вскрылись и другие проблемы: полуробочий `wget` и невозможность сборки `Xorg`. Чуть позже был написан `cpkgi-tools` - то, из чего впоследствии получился автоматизированная утилита для работы с пакетами `cpkg`. А `cpkgi-tools` (Calmira Package Installer Tools) в то время мог только устанавливать ПО, об удалении и прочих

функциях не было и речи. Хотя были какие-то концепты для добавления жизненно важных функций, но они не были реализованы окончательно ввиду большой разрозненности - разные программы для установки, удаления и просмотра информации о пакете, а так же абсолютно разные общие функции для этого.

Самым главным достоинством `crkg` было то, что пакет он ставил в отдельную директорию (`/usr/user`). Но были и минусы - сильное раздувание `PATH`, поэтому чуть позже от этого отказались. Были очень жёсткие ограничения на имя пакета (так, например, в нём не должно было содержаться ничего, кроме имени пакета и расширения `tar.xz`, которое в будущем сократилось до `txz`), а так же кол-во файлов. Были серьёзные проблемы с добавлением пакета в базу данных. Однако `crkgi-tools` задал формат пакетов для будущего `crkg`. Этот формат, кстати, используется до сих пор, только немного дополнялся.

Что и говорить - сначала у Calmira не было и собственного логотипа! Он появился почти перед выходом версии 2021.2 и являлся перерисованным изображением одноимённой оболочки для Win3.11. Логотип неофициальный, и до сих пор не считается логотипом дистрибутива, хотя некоторые его компоненты (например, просмотр информации о версии дистрибутива) используют его, а на официальном сайте выложен скриншот дистрибутива с выводом в терминал ASCII-версии логотипа.

Как говорится, первый блин комом. Но Calmira 2021.1 дала очень большой опыт в создании дистрибутивов (до этого автор "забавлялся" пересборкой Debian, AntiX и Mint), а так же после выхода этой версии появилось очень много планов, некоторые из которых не реализованы и сегодня.

Как назло, первая версия была утеряна при создании `sqsh`-снимка для отправки дистрибутива знакомым для их оценки. Поэтому этот релиз не был выложен на GitHub. Возможно, в скором времени будет воссоздана эта версия.

2021.2 - формирование новой редакции с сервером Xorg и учёт некоторых старых ошибок

Версия 2021.2 выгодно отличается от предыдущей. Из неё был убран `crkgi-tools` и добавлен новый `crkg`, исправлен баг с пакетом `gmp`, из-за небольшого упущения в сборке которого невозможно было скомпилировать 90% программ на Calmira. В минимальную поставку было добавлено несколько новых пакетов. Так же начались работы над созданием репозитория с бинарными пакетами для `crkg`.

Автоматизированная утилита для работы с пакетами `crkg` тогда многого не умел. Работал он нестабильно, но был намного лучше предыдущих `crkgi-tools`. Добавили нормальную базу данных пакетов, установка и удаление пакета работали намного быстрее и надёжнее, было снято большое число ограничений. Добавили `pre-` и `postinstall` скрипты, а ближе к релизу Calmira 2021.3 `crkg` научится собирать ПО из исходного кода самостоятельно на основе предварительно подготовленных инструкций для сборки.

Да, пока ещё `crkg` - АУРП. Пакетным менеджером его в то время назвать было трудно - функционал был небольшим. Но скоро всё изменится в лучшую сторону. Релиз `crkg 1.0`pa4 поменяет скромное "Автоматизированная утилита для работы с пакетами" на гордое "пакетный менеджер". `pa4` означало "pre-alpha 4". У `crkg` была немного другая система наименования версий, но очень похожая на то, что будет через релиз в будущем.

Однако баги так же были (а как же без них?). Например, сломанный `make-ca` и невозможность скачивания файлов с помощью `wget` без ключа `--no-check-certificates`, нерабочий `git` и полурбочий `uptime` (возможно, сломанными были и другие утилиты).

Было обновлено большое число пакетов и конфигов, а так же дистрибутив начал потреблять меньше оперативной памяти.

Помимо этого, была подготовлена отдельная редакция с сервером Xorg, а так же бинарный пакет с оконным менеджером Fluxbox (который нужно было установить вручную посредством нового `crkg`).

2021.3 - небольшое обновление

В этом релизе было подготовлено несколько мелких, но довольно важных обновлений - новая предварительная версия `srkg`, некоторые изменения в поставке ПО из минимальной сборки, отказ от редакции с Xorg, так как в планах было создание бинарных пакетов с этим сервером.

Начиная с этой версии начинает меняться позиционирование дистрибутива. Теперь это не просто легковесный дистрибутив, а минималистичная гибкая и быстрая система для решения узкого круга задач. Был создан "Манифест разработчиков Calmira" (он же "Несколько принципов построения ОС Calmira GNU/Linux"), в котором были установлены жёсткие ограничения на потребление дистрибутивом ресурсов ПК и установленный в нём софт.

В этом манифесте было описано несколько принципов: делать систему максимально компактной и минималистичной, но не в ущерб функциональности; ориентировать систему не для новичка, а на продвинутого пользователя; обеспечивать максимальную свободу пользователю; не использовать наработки Calmira во вред; делиться наработками с сообществом, если автор наработок считает это нужным.

В `srkg` добавили массу новых функций, таких как скачивание пакета с репозитория, обновление списка пакетов, а так же создание локального репозитория. `srkg` научился собирать ПО из исходного кода. В этот релиз дистрибутива вошёл `srkg 1.0ra4` с массой новых изменений. На то время это был самый крупный релиз за всю историю `srkg`.

Но увы - разработка этой версии была окончена во время тестирования. Несмотря на то, что это был относительно неплохой релиз, изменений в нём было довольно мало, а предыдущая версия 2021.2 всех устраивала. Поэтому все изменения было принято отложить на будущее.

1.2.3. Последние версии

LX4 1.0 - смена структуры директорий на упрощённую, повышение стабильности дистрибутива и куча других изменений

Самое большое изменение - это отказ от LFS в сторону молодого Linux4Yourself, сокращённо называемым LX4Y, LX4U либо же совсем коротко - LX4 (Linux4Yourself, кстати, появился через некоторое время после старта разработки Calmira 2021.1). Вместе с этим переходом изменилась и структура каталогов системы. В LX4 (и будущих LFS 11, кстати, тоже) отсутствуют `/bin`, `/sbin`, `/usr/sbin` и `/lib` - они заменены ссылками на `/usr/bin` и `/usr/lib` соотв. Это менее безопасно, а так же делает систему не такой гибкой (предыдущие релизы могли вообще без `/usr` загружаться и нормально работать). Это противоречит философии Calmira. В релизе LX4 1.1 это будет исправлено.

Вы могли заметить, что и система наименования версий дистрибутива поменялась. Вместо запутанной "годовой" (так называет разработчик Calmira предыдущую систему наименования версий) используется более простая. Мажорная версия (число до точки) значит крупное изменение, новый крупный релиз; минорная версия (число после точки) значит небольшие изменения, новый промежуточный релиз с мелкими обновлениями и изменениями. Приставка LX4 означает, что используется *Linux4Yourself*.

Из поставки дистрибутива был удалён пакетный менеджер `srkg`, ввиду его нестабильности. Он часто приводил систему в полурбочее состояние, а с упрощённой структурой директорий может убить систему полностью. В релиз LX4 1.0 планировалось ввести систему портов (например, как в FreeBSD), но работа над ней не была закончена, поэтому введение этой системы было отложено на релиз LX4 1.1.

Так же некоторое ПО было заменено на более совершенные форки или аналоги (например, `zlib` на `zlib-ng`).

Окончательно утвердился метод сжатия пакетов и образов системы. Теперь используется `xz` как основной ввиду своей эффективности сжатия, хотя

сжимает чуть медленнее всех остальных.

На этапе формирования новый цикл разработки дистрибутива. Но об этом позже.

Что потом?

Сейчас LX4 1.0 - последняя версия дистрибутива. В неё вошло много исправлений и улучшений. Система работает быстрее и надёжнее, а так же скоро появится много возможностей. Через пару месяцев ожидаем окончательную версию Calmira LX4 1.1 с огромным числом изменений. За ходом разработки вы можете посмотреть у нас на GitHub. А пока подумайте - что будет в новой версии дистрибутива?

[Назад](#)

О Calmira GNU/Linux

[Домой](#)

[Далее](#)

Установка дистрибутива

2.0. Установка дистрибутива

В данном разделе пойдёт речь об установке системы. Установка Calmira не похожа на установку других дистрибутивов. Вам нужно самому распаковать образ, скопировать данные и настроить конфиги, а некоторые написать самостоятельно (в т.ч. конфиг загрузчика GRUB).

Скоро будет готов установочный iso/img образ системы, и процесс установки будет более предсказуемым и понятным - сейчас же для установки Calmira нужен другой дистрибутив GNU/Linux, с помощью которого будет распакован снимок с системой, подготовлены необходимые разделы и скопированы на них данные из снимка. С помощью установочного iso/img не нужно будет загружаться в уже установленный GNU/Linux для подготовки ПК к установке Calmira.

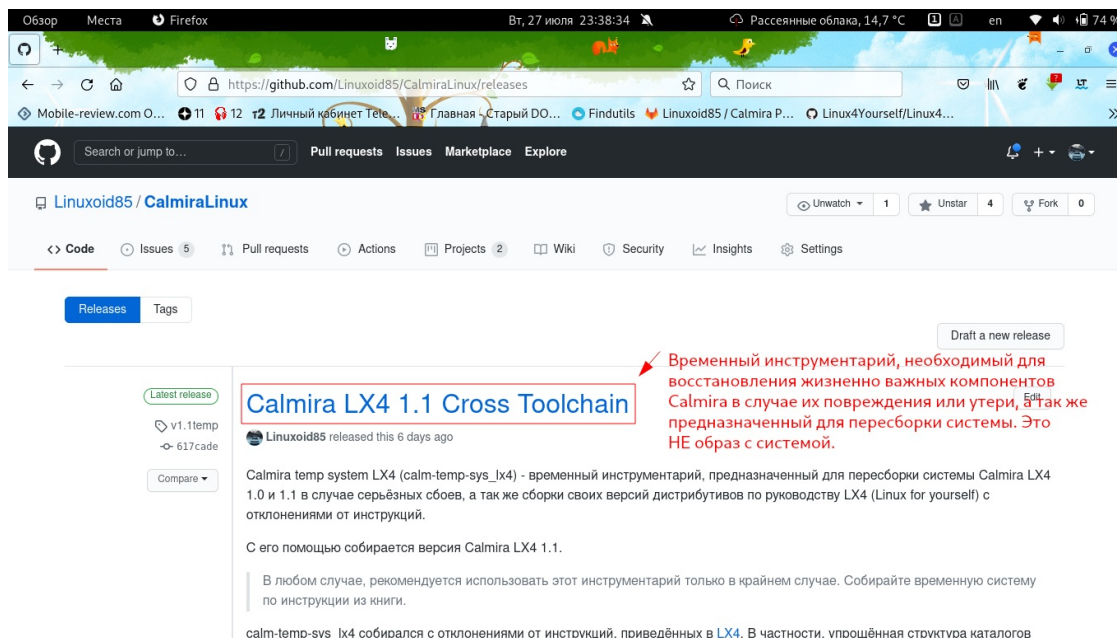
Содержание

- 2.1 [Загрузка дистрибутива из репозитория GitHub](#)
- 2.2 [Распаковка дистрибутива, форматирование раздела и копирование на него системы](#)
- 2.3 [Установка системы на ПК](#)
- 2.4 [Установка системы в Qemu/KVM](#)
- 2.5 [Первичная настройка дистрибутива после копирования](#)
- 2.6 [Завершение установки. Выход из chroot и перезагрузка](#)
- 2.7 [Вход в систему](#)

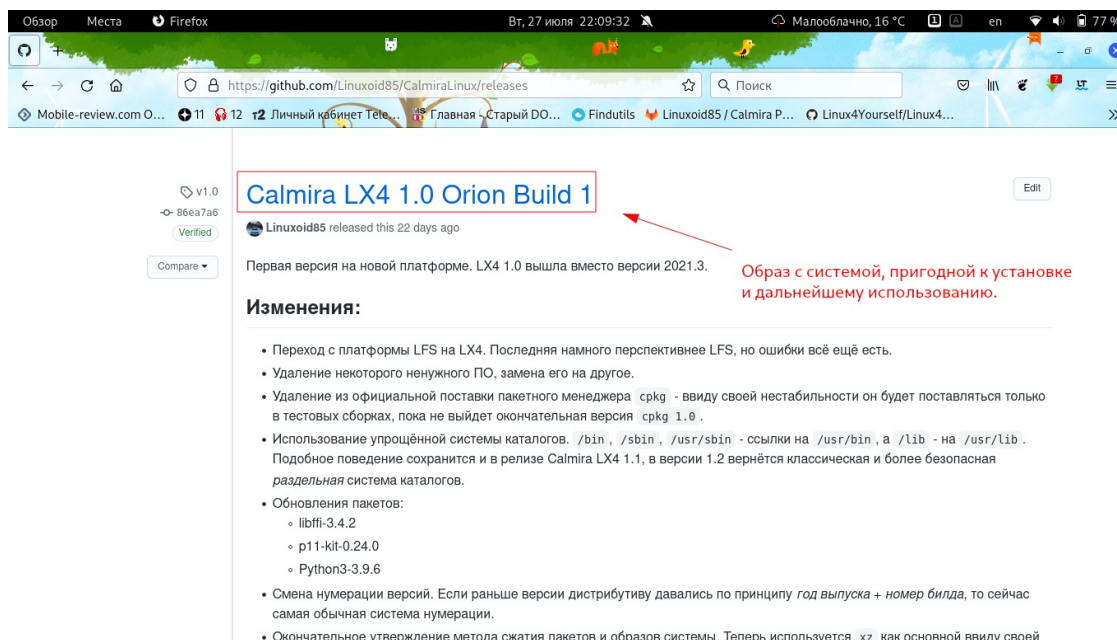
2.1. Скачивание дистрибутива из репозитория GitHub

На главной странице приведён раздел загрузок, но там приведены последние версии системы. А пользователь вправе выбирать нужную только ему.

Все релизы находятся в разделе [Releases](#). Здесь выложены как образы самой системы, так и временные инструменты для пересборки или восстановления Calmira GNU/Linux. Распознать одно от другого просто - у временного инструментария в заголовке написано "Calmira \$VERSION Cross Toolchain". Это архивы с временным инструментарием (кросс-компилятором и другими инструментами, необходимыми для восстановления или пересборки системы). ЭТО НЕ ЗАГРУЗОЧНЫЕ СИСТЕМЫ!



Всё остальное же - образы с системой. Вам следует скачать образ `sqsh`. Впоследствии, он будет распакован, а данные из него будут скопированы на определённый раздел жёсткого диска. Архивы `zip` и `tar`, подписанные как "Source code" являются копией данных из репозитория. Вам следует скачивать образ `sqsh`.



Нужные файлы, описанные в предыдущем параграфе, находятся в разделе "Assets":

Assets 3	
calmira-1.0.sqsh	207 MB
Source code (zip)	
Source code (tar.gz)	

[Назад](#)
Установка дистрибутива

[Домой](#)

[Далее](#)
Распаковка дистрибутива

2.2. Распаковка дистрибутива

После скачивания дистрибутива нужно его распаковать. Это не *raw*-образ, который можно записать утилитой *dd* или любыми другими аналогами. Тут нужно сделать всё самостоятельно.

2.2.1. Распаковка

Так как это *squashfs*-образ, то распаковка производится утилитой *unsquashfs* из состава пакета *squashfs-tools*. Пример использования:

```
sudo unsquashfs calmira-1.1.sqsh
```

Пакет *squashfs-tools* может быть скомпилирован в Calmira GNU/Linux из системы портов. На других системах может быть установлен из бинарного пакета. Учитывайте то, что, например, на Fedora Linux (и других *rpm*-дистрибутивах) этот пакет НЕ собирается с поддержкой формата сжатия *xz*. Вам необходимо будет пересобрать его с поддержкой этого формата сжатия, так как образ с дистрибутивом Calmira сжимается с помощью *xz*.

□ ВНИМАНИЕ!

squashfs очень сильно сжимает данные. Сжатый образ весит 207 Мб, а данные, которые распакуются из него - 1 Гб. Учитывайте это.

Файлы распакуются в директорию *squashfs-root*. Введите:

```
cd squashfs-root
```

2.3. Установка системы на ПК

В данной инструкции описаны действия по установке системы на реальное железо.

2.3.1. Форматирование раздела, на который установится Calmira GNU/Linux

Предполагается, что Calmira будет установлена на один раздел, а не несколько. Хотя вы без проблем можете выделить столько разделов под установку, сколько вам надо.

Минимальный раздел корневого раздела должен равняться 1.5 Гб. Этого вполне хватит для установки дистрибутива и хранения некоторых пользовательских файлов (например, документов). Отформатируйте раздел в файловую систему *ext4*:

```
sudo mkfs.ext4 /dev/sdX
```

Замените *sdX* на метку раздела.

Теперь смонтируйте этот раздел в */mnt/calmira_system*:

```
sudo mkdir /mnt/calmira_system  
sudo mount -v /dev/sdX /mnt/calmira_system
```

Замените *sdX* на метку раздела.

2.3.2. Копирование системы

После того, как смонтировали раздел, скопируйте на него данные из распакованного образа дистрибутива. Во-первых убедитесь, что находитесь в директории *squashfs-root*:

```
pwd
```

Если всё верно, приступайте к копированию:

```
sudo cp -rvx * /mnt/calmira_system/
```

Во время копирования на экран будут выведены скопированные файлы. Внимательно прочитайте вывод: в нём не должно содержаться *error*, *fail* и пр. Если вы не хотите видеть подробный результат о каждом скопированном файле, то уберите ключ *-v* из команды. В таком случае на экран будут выведены сообщения об ошибках и предупреждения (если они есть).

2.4. Установка системы в Qemu/KVM

Вы можете захотеть установить Calmira не на реальное железо, а в виртуальную машину Qemu/KVM. Создайте виртуальную машину со следующими параметрами:

1. Тип ОС: Generic Linux/Generic Default;
2. Процессор: qemu64/kvm64/копирующий конфигурацию ЦП хоста (в случае если на хосте процессор x86_64);
3. Оперативная память: 64 Мб;
4. Жёсткий диск: qcow2 1.5 Гб;
5. Шина диска: SATA.

Всё это создавалось в virt-manager. Параметры рекомендуемые.

2.4.1. Монтирование

Теперь нужно скопировать распакованный в [этой](#) инструкции дистрибутив в только что созданный образ жёсткого диска (образ сгенерировался во время создания виртуальной машины). Нужно смонтировать этот образ. Для этого подключите модуль qemu-nbd и смонтируйте образ ЖД:

```
# Подключение модуля ядра:
sudo modprobe nbd max_part=8

# Подключение образа:
sudo qemu-nbd --connect=/dev/nbd0 /var/lib/libvirt/images/$NAME.qcow2

sudo fdisk /dev/nbd0 -l
mkdir /mnt/calmira_system
```

Замените `$NAME` на имя виртуальной машины (и, по совместительству, имя диска).

2.4.2. Разметка разделов

Необходимо разметить новый "жёсткий диск". Для этого выполните:

```
sudo cfdisk /dev/nbd0
```

`cfdisk` запросит у вас, какую таблицу разделов выбрать. Выбирайте *mbr* (она же *dos*), либо новомодную *gpt*. В данном случае нет никакой разницы в выборе. Советуется выбрать *mbr* (*dos*).

Если вы всё-таки выбрали *gpt*, то создайте в начале диска раздел объёмом 1 Мб без файловой системы. Тип: *BIOS Boot*.

Потом (и для *gpt*, и для *mbr*) создайте корневой раздел. Он должен занимать всё оставшееся место. Тип: *Linux filesystem*. При желании можете создать ещё и раздел со *swap*. Его объём должен равняться половине от объёма оперативной памяти (к примеру, в виртуальной машине установлено значение ОЗУ, равное 64 Мб. Для подкачки тогда нужно выбрать объём 32 Мб). Тип раздела с подкачкой: *Linux swap*.

2.4.3. Форматирование разделов

После разметки отформатируйте нужные разделы. Для форматирования корневого раздела выполните:

```
sudo mkfs.ext4 /dev/nbd0pX
```

Замените *nbd0pX* на нужную метку раздела. Обычно это *nbd0p1*, если таблица разделов *mbr (dos)*, или *nbd0p2*, если таблица разделов - *gpt*.

Для форматирования подкачки (если вы её создали):

```
mkswap /dev/nbd0pX
```

Замените *nbd0pX* на нужную метку раздела. Обычно это *nbd0p2*, если таблица разделов *mbr (dos)*, или *nbd0p3*, если таблица разделов - *gpt*.

2.4.4. Копирование системы

Смонтируйте корневой раздел:

```
sudo mount /dev/nbd0pX /mnt/calmira_system
```

Замените *nbd0pX* на нужную метку раздела. Обычно это *nbd0p1*, если таблица разделов *mbr (dos)*, или *nbd0p2*, если таблица разделов - *gpt*.

После того, как смонтировали раздел, скопируйте на него данные из распакованного образа дистрибутива. Во-первых убедитесь, что находитесь в директории *squashfs-root*:

```
pwd
```

Если всё верно, приступайте к копированию:

```
sudo cp -rvxa * /mnt/calmira_system/
```

Во время копирования на экран будут выведены скопированные файлы. Внимательно прочитайте вывод: в нём не должно содержаться *error*, *fail* и пр. Если вы не хотите видеть подробный результат о каждом скопированном файле, то уберите ключ *-v* из команды. В таком случае на экран будут выведены сообщения об ошибках и предупреждения (если они есть).

[Назад](#)

Установка дистрибутива
на ПК

[Домой](#)

[Далее](#)

Первичная настройка
после копирования

2.5. Первичная настройка дистрибутива после копирования

Вы скопировали содержимое снимка себе на компьютер. Сейчас же необходимо настроить дистрибутив для корректной работы на нём. Без выполнения этих действий Calmira GNU/Linux не будет работать.

2.5.1. Смена корня

Первым делом необходимо загрузиться в только что скопированную Calmira. Сейчас же нельзя это сделать, так как система не загрузится - нужно настроить fstab и загрузчик. Поэтому на хост-системе произведём операцию смены корня.

Первым делом вы должны смонтировать раздел жёсткого диска или образ виртуального диска qcow2 в вашу систему. Точка монтирования: `/mnt/calmira_system`. О монтировании сказано в инструкциях по копированию системы на раздел жёсткого диска ([здесь](#)), либо в виртуальную машину ([здесь](#)).

```
for DIR in "dev" "dev/pts" "proc" "sys"; do
sudo mount -v --bind /$DIR /mnt/calmira_system/$DIR
done

if [ -h /mnt/calmira_system/dev/shm ]; then
mkdir -pv /mnt/calmira_system/$(readlink /mnt/calmira_system/dev/shm)
fi

chroot "/mnt/calmira_system" /usr/bin/env -i \
HOME=/root TERM="$TERM" \
PS1=(chroot) \u:w/$ \
PATH=/bin:/sbin:/usr/bin:/usr/sbin \
/bin/bash --login
```

Вы окажетесь в скопированной Calmira GNU/Linux. Вы изолированы от хост-системы, и можете настраивать дистрибутив так, как вам удобно.

2.5.2. Настройка file system table

fstab (file systems table) — один из конфигурационных файлов, который содержит информацию о различных файловых системах и устройствах хранения информации компьютера, описывает, как диск будет использоваться или как будет интегрирован в систему. Файл `/etc/fstab` делает возможным автоматическое монтирование определенных файловых систем, что особенно нужно при загрузке системы. Он содержит ряд строк, описывающих файловые системы, их точки монтирования и другие параметры.

Каждая строка содержит:

1. устройство монтируемой файловой системы;
2. точку монтирования;
3. тип файловой системы;
4. параметры монтирования;
5. флаг для `dump`, утилиты создания резервных копий;
6. порядок проверки для `'fsck'` (File System Check).

Здесь всегда есть запись о корневой файловой системе. Раздел `swar` является специальным, поэтому его не видно в древовидной структуре, и в поле точки монтирования для таких разделов всегда содержится ключевое слово `swar`.

Необходимо настроить `fstab`, чтобы система корректно загружалась. Настройка `fstab` будет примерно одинаковой как для способа установки системы на раздел жёсткого диска, так и для способа установки системы в виртуальную машину Qemu.

2.5.2.1. Настройка fstab

Создайте файл `fstab`:

```
cat > /etc/fstab << "EOF"
# Begin /etc/fstab

# file system mount-point type options dump fsck order

/dev/sdX / ext4 defaults 1 1
proc /proc proc nosuid,noexec,nodev 0 0
sysfs /sys sysfs nosuid,noexec,nodev 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
tmpfs /run tmpfs defaults 0 0
devtmpfs /dev devtmpfs mode=0755,nosuid 0 0
EOF
```

Замените `/dev/sdX` из первой строки `fstab` на метку корневого раздела, на который устанавливалась система. Например, `/dev/sda3`. Если вы устанавливаете систему в виртуальную машину, то метка корневого раздела будет называться примерно так: `/dev/nbd0pX`, где `X` - номер нужного раздела.

Если у вас UEFI, то добавьте соотв. запись в `fstab`:

```
echo "/dev/sdN /boot/efi vfat umask=0077 0 0" >> /etc/fstab
```

Замените `/dev/sdN` из команды выше на нужную метку раздела EFI. Повторимся, что его объём должен быть равен 256 Мб, файловая система - Fat32 (vfat).

Если вы используете `swap`-раздел, то и его пропишите в `fstab`:

```
echo "/dev/sdM swap swap pri=1 0 0" >> /etc/fstab
```

Замените `/dev/sdM` из команды выше на нужную метку раздела с подкачкой (swap).

2.5.2.2. Монтировать в ОЗУ или не монтировать? - вот в чём вопрос.

Как вы могли заметить в `/etc/fstab`, некоторые директории монтируются в оперативную память (посредством `tmpfs`). У такого способа главное достоинство в том, что при монтировании этих каталогов в `tmpfs`, система будет работать несколько быстрее. На современном оборудовании это не так заметно, но на старом и слабом может сделать систему немного быстрее.

Недостаток в том, что на несколько Мб система будет потреблять больше. Так, например, если система потребляла 28 Мб, то с монтированием в `tmpfs` будет потреблять 30 Мб. Потребление зависит от содержимого тех директорий.

Владельцам оборудования с небольшим объёмом ОЗУ (меньше 64 мегабайт) просьба учитывать этот факт и трезво оценить достоинства и недостатки выноса некоторых директорий в ОЗУ.

Дополнительно о метке корневого раздела с системой

Мы настоятельно советуем использовать вместо метки корневого раздела, на который установлена Calmira GNU/Linux (например, `/dev/sda1`, `/dev/hdc2`, etc.), его UUID. Если метка диска, прописанного в `/etc/fstab` изменится, то могут возникнуть проблемы с загрузкой ОС, либо же она не загрузится вообще.

Узнать UUID для нужного раздела можно, выполнив:

```
blkid /dev/sdX
```

Замените `/dev/sdX` на метку корневого раздела. Например, `/dev/sda3`.

Вывод будет примерно таким:

```
/dev/sda3: UUID="1ed63fa7-e4c9-43cc-96f4-9f951224a338" BLOCK_SIZE="4096" TYPE="ext4" PARTLABEL="Calmira data" PARTUUID="56ae06ac-dbd8-11eb-96fc-e8039ae29c93"
```

Вам понадобится что-то вроде:

```
UUID="1ed63fa7-e4c9-43cc-96f4-9f951224a338"
```

Запишите эту строку вместо `/dev/sdX` в `fstab` (но без кавычек). Например, `fstab` с UUID корневого раздела вместо его метки будет выглядеть так:

```
cat > /etc/fstab << "EOF"
# Begin /etc/fstab

# file system mount-point type options dump fsck order
UUID=1ed63fa7-e4c9-43cc-96f4-9f951224a338 / ext4 defaults 1 1
proc /proc proc nosuid,noexec,nodev 0 0
sysfs /sys sysfs nosuid,noexec,nodev 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
tmpfs /run tmpfs defaults 0 0
devtmpfs /dev devtmpfs mode=0755,nosuid 0 0
EOF
```

Замените `UUID=1ed63fa7-e4c9-43cc-96f4-9f951224a338` на полученное значение.

Этот способ является более надёжным, чем использование обычных меток дисков/разделов, которые могут в определённых ситуациях меняться.

2.5.3. Установка загрузчика GRUB

В Calmira используется классический загрузчик GRUB2. Без него система не сможет загрузиться. Процесс установки для MBR BIOS, Legacy BIOS/GPT BIOS, UEFI различаются. Здесь будут описаны все эти три случая.

2.5.3.1. Установка загрузчика в MBR

Для того, чтобы установить загрузчик, выполните:

```
grub-install /dev/sdX
```

Замените `sdX` на метку диска. Например, `/dev/sda` или `/dev/nbd0`.

ЗАМЕЧАНИЕ.

В предыдущих инструкциях нужно было заменить `sdX` на метку раздела (например, на `sda3`). Сейчас же нужно заменить на метку ДИСКА (например, `sda`). Полученная команда будет `grub-install /dev/sda`, если вы выбрали установку загрузчика на этот диск. НЕ ВЫБИРАЙТЕ установку в раздел (`sda3`) - ни к чему это не приведёт.

2.5.3.2. Установка загрузчика Legacy BIOS

Если вы по каким-то причинам используете Legacy BIOS, либо же обычный BIOS, но у жёсткого диска таблица разделов GPT, то для установки загрузчика вам необходимо иметь раздел объёмом 1 Мб без файловой системы в начале диска. Если у вас на ПК уже установлена какая-либо ОС, то этот раздел должен присутствовать.

Установите загрузчик командой:

```
grub-install /dev/sdX
```

Замените *sdX* на метку диска. Например, */dev/sda* или */dev/nbd0*.

❏ ЗАМЕЧАНИЕ.

В предыдущих инструкциях нужно было заменить *sdX* на метку раздела (например, на *sda3*). Сейчас же нужно заменить на метку ДИСКА (например, *sda*). Полученная команда будет `grub-install /dev/sda`, если вы выбрали установку загрузчика на этот диск. НЕ ВЫБИРАЙТЕ установку в раздел (*sda3*) - ни к чему это не приведёт.

2.5.3.3. Установка загрузчика EFI

На данный момент все современные ПК имеют UEFI, а не BIOS. Для установки системы на UEFI, вы должны иметь доп. раздел объёмом 256 Мб с файловой системой *fat32* и названием *ESP*. Если у вас на ПК уже установлена какая-либо ОС, то этот раздел уже имеется. Смонтируйте его в */boot/efi* командой `mount`. После чего необходимо установить несколько пакетов из системы портов для правильной работы загрузчика.

Установите из все порты из *base/grub-efi* в следующем порядке:

- `efivar`;
- `popt` (находится не в *base/grub-efi*, а в *general_libs*);
- `efibootmgr`;
- `grub`.

Для этого выполнить:

```
cd /usr/ports/$КАТЕГОРИЯ/$ПАКЕТ
./install
```

- *\$КАТЕГОРИЯ* - категория, в котором располагается порт. Например, *base/grub-efi* или *general_libs*;
- *\$ПАКЕТ* - имя нужного пакета. Например, ``efibootmgr``.

После чего можно устанавливать `grub`:

```
grub-install /dev/sdX
```

Замените *sdX* на метку диска. Например, */dev/sda* или */dev/nbd0*.

❏ ЗАМЕЧАНИЕ.

В предыдущих инструкциях нужно было заменить *sdX* на метку раздела (например, на *sda3*). Сейчас же нужно заменить на метку ДИСКА (например, *sda*). Полученная команда будет `grub-install /dev/sda`, если вы выбрали установку загрузчика на этот диск. НЕ ВЫБИРАЙТЕ установку в раздел (*sda3*) - ни к чему это не приведёт.

2.5.4. Создание конфига GRUB (актуально для установки загрузчика на MBR или Legacy BIOS)

Создайте конфиг следующей командой:

```
cat > /boot/grub/grub.cfg << "EOF"
# Begin /boot/grub/grub.cfg
set default=0
set timeout=5

insmod ext2
set root=(hd0,2)

menuentry "Calmira LX4 1.1 GNU/Linux, Linux" {
    linux /boot/vmlinuz root=/dev/sda2 ro
}
EOF
```

Строка `set timeout=5` устанавливает таймаут (время ожидания автоматической загрузки системы, если пользователь не выбрал ни одного элемента из загрузочного меню GRUB), равный пяти секундам. Измените это значение при необходимости.

Строка `set root=(hd0,2)` устанавливает нужный жёсткий диск. Если вы не знаете какой выбрать, наберите в консоли GRUB строки:

```
ls
ls ($DISK)/boot/grub
```

Первая команда выдаст список дисков, а вторая выдаст список файлов загрузчика, если выбран нужный диск с системой (замените *\$DISK* на нужный диск из вывода первой команды).

Строка `linux /boot/vmlinuz root=/dev/sda2 ro` указывает нахождение ядра Linux (`linux /boot/vmlinuz`), а `root=/dev/sda2` указывает ядру, какой корневой раздел ему использовать. Замените `/dev/sda2` на ваш раздел, куда копировалась система в предыдущей инструкции.

2.6. Завершение установки. Выход из chroot и перезагрузка

Установка и загрузка системы завершена. Теперь вы можете выйти из chroot и перезагрузить компьютер.

2.6.1. Выход из chroot

Для того, чтобы выйти из chroot и отмонтировать виртуальные ФС ядра, выполните:

```
logout

sudo umount /mnt/calmira_system/dev{/pts,}
sudo umount /mnt/calmira_system/{sys,proc,run}
```

2.6.2. Размонтирование

Теперь можно отмонтировать раздел/образ жёсткого диска из системы.

2.6.2.1. При установке в раздел

```
sudo umount /dev/sdX
```

Замените *sdX* на метку раздела.

2.6.2.2. При установке на виртуальную машину

1. Отмонтируйте образ виртуального жёсткого диска;
2. Отключите образ ВЖД от системы;
3. Отключите модуль ядра nbd.

```
sudo umount /mnt/calmira_system
sudo qemu-nbd --disconnect /dev/nbd0
sudo rmmod nbd
```

2.6.3. Перезагрузка

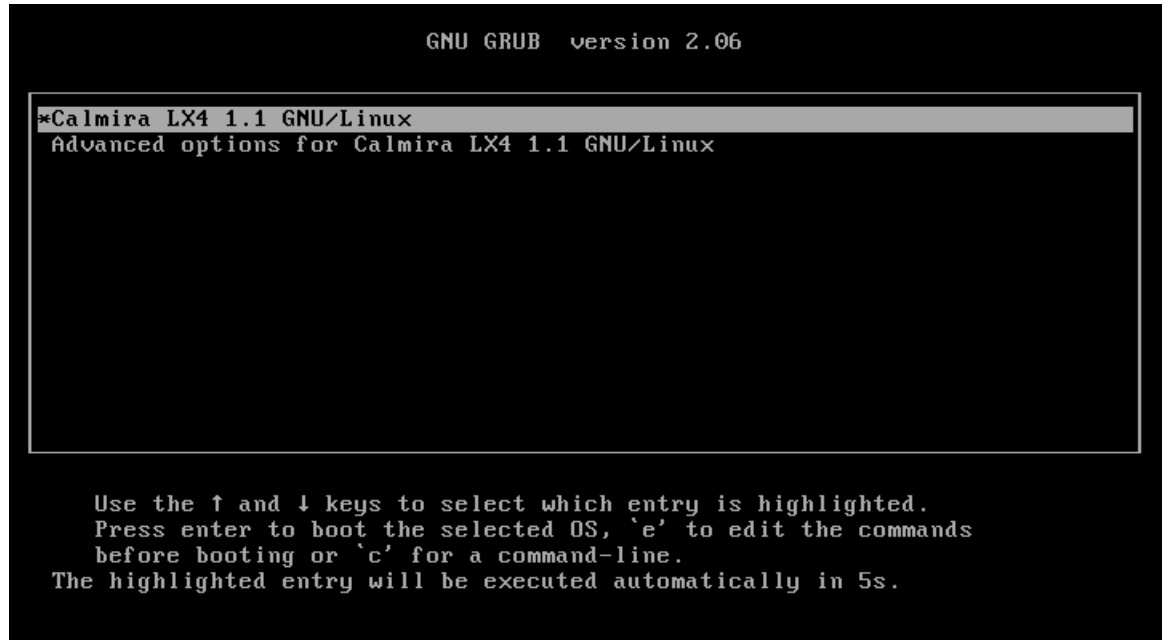
Введите:

```
sudo reboot
```

Для перезагрузки.

2.7. Вход в установленную систему

В предыдущей инструкции вы ввели команду `reboot` для перезагрузки. После того, как ваш ПК включится, вас будет приветствовать загрузочное меню GRUB:



Для загрузки Calmira выберите пункт `Calmira LX4 1.1 GNU/Linux`. Во время загрузки будут выведены сообщения ядра, а через какое-то время - сообщения загрузки сервисов `SysVinit`. Как только появится сообщение `Welcome to Calmira GNU/Linux!`, система будет загружена полностью. Она запросит у вас логин и пароль. Введите их, если во время настройки вы создавали пользователя, если нет, то зайдите из-под корневой учётной записи и продолжите дальнейшую настройку. Логин: `root`, пароль: `root`.



Примерное содержимое экрана после окончания загрузки системы. Вместо `calm-pc` может быть ваше имя хоста, которое вы указывали во время настройки дистрибутива.

Если вы ввели логин, а пароль не вводите на протяжении 60 секунд, то у вас снова отобразится приветственное сообщение `Welcome to...` и запрос к вводу логина и пароля. Учитывайте, что вы не увидите никаких символов при вводе пароля. Это правило актуально так же для утилит `su` и `sudo`.

После удачного входа в систему отобразится дата последнего входа в Calmira, число пропущенных e-mail сообщений (в случае отсутствия отобразится `No mail.`), а так же системное приглашение к вводу команды `(PS1):root:~#`. Вместо `root` будет имя пользователя, под которым вы вошли в систему, `:` разделяет имя пользователя и директорию в которой он находится, соответственно, вместо `~` будет отображаться та директория, в которой находится пользователь. Знак `~` означает, что вы находитесь в своей домашней директории. Если вы вошли от имени `root`, то помимо соответствующего имени в начале приглашения (до двоеточия), в конце приглашения (после рабочей директории) будет отображён знак `#`. Он означает, что вы имеете повышенные привилегии, так как зашли от имени `root`. Если вы вошли от имени обычного пользователя, то вместо этого знака будет `$`.

```

*   Checking file systems... [ OK ]
Remounting root file system in read-write mode...[ 2.493519] EXT4-fs (s
da2): re-mounted. Opts: (null). Quota mode: none.
*   [ OK ]
*   Mounting remaining file systems... [ OK ]
*   Cleaning file systems: /tmp [ OK ]
*   Retrying failed uevents, if any... [ OK ]
Setting up Linux console...[ 3.444894] urandom_read: 3 callbacks suppre
ssed
[ 3.444923] random: udevd: uninitialized urandom read (16 bytes read)
[ 3.444956] random: udevd: uninitialized urandom read (16 bytes read)
[ 3.444996] random: udevd: uninitialized urandom read (16 bytes read)
[ 3.544574] random: crng init done
*   [ OK ]
INIT: Entering runlevel: 3
*   Starting system log daemon... [ OK ]
*   Starting kernel log daemon... [ OK ]

Welcome to Calmira GNU/Linux!

test login: root
Password:
Last login: Sat Aug 21 01:14:43 +0300 2021 on /dev/tty1.
No mail.
root:~#

```

После входа в систему вы можете работать с ней: создавать, редактировать и перемещать файлы, устанавливать дополнительное ПО из системы портов и прочее. Но сейчас рекомендуем перейти к следующему пункту - "Настройка дистрибутива".

[Назад](#)

Завершение установки

[Домой](#)

[Далее](#)

Настройка системы

3.0. Настройка дистрибутива

В данном разделе приведены сведения о настройке системы Calmira GNU/Linux.

3.1 [Настройка окружения](#)

3.2 [Настройка часов](#)

3.3 [Настройка клавиатуры и шрифта в TTY](#)

3.4 [Настройка inittab](#)

3.5 [Установка дополнительных загрузочных скриптов](#)

3.1. Настройка окружения и конфигурационных файлов

В данном разделе содержится описание конфигурационных файлов Calmira.

3.1.1. Редактирование текста приветствия

```
* Activating all swap files/partitions... [ OK ]
Mounting root file system in read-only mode... [ 2.756238] EXT4-fs (sda
2): re-mounted. Opts: (null). Quota mode: none.
* [ OK ]
Checking file systems...[ 2.767473] random: crng init done
[ 2.767980] random: 3 urandom warning(s) missed due to ratelimiting
* [ OK ]
Remounting root file system in read-write mode...[ 2.853062] EXT4-fs (s
da2): re-mounted. Opts: (null). Quota mode: none.
* [ OK ]
Mounting remaining file systems... [ OK ]
Cleaning file systems: /tmp [ OK ]
Retrying failed uevents, if any... [ OK ]
Setting up Linux console... [ OK ]
INIT: Entering runlevel: 3
Starting system log daemon... [ OK ]
Starting kernel log daemon... [ OK ]

Welcome to Calmira GNU/Linux! Developer ONLY version!
Do not use this version permanently. Do not leak assembly data
to third parties!

Build 18.07.1.2021

test login:
```

Перед тем, как система запросит ваш логин и пароль, она выведет приветственное сообщение с базовой информацией о дистрибутиве. Если вам оно не нравится, либо вы хотите его заменить на нужное вам, отредактируйте файл `/etc/issue`:

```
# Очистка файла, если вам не надо, чтобы
# отображалось приветственное сообщение:
> /etc/issue

# Редактирование файла:
vim /etc/issue
```

3.1.2. Редактирование общесистемных настроек окружения

Основной упор в Calmira GNU/Linux на работу в TTY, а не графике. Поэтому пользователям было бы неплохо отредактировать настройки по умолчанию для более комфортной работы. Общесистемные настройки окружения находятся в файле `/etc/profile`.

Функции `pathremove`, `pathprepend`, `pathappend` предназначены для работы с PATH. Функция `ver` отображает информацию о релизе дистрибутива Calmira, а `system_welcome` показывает информацию о базовых командах и предназначен для вставки в `/etc/bashrc`, `/etc/skel/.bashrc` и/или `~/.bashrc`.

По умолчанию в PATH содержатся директории `/bin`, `/sbin`, `/usr/bin`, `/usr/sbin`. Если вы создаёте обычного пользователя без привилегий, специфичных пользователю `root`, то **обязательно** уберите из PATH директории `/sbin` и `/usr/sbin`. Обычному пользователю программы оттуда не понадобятся, так же

эта операция обезопасит вас и непривилегированного пользователя от всевозможных ошибок и вирусов. Для этого в `~/.bashrc` или `~/.profile` обычного (не root) пользователя добавьте строки:

```
unset PATH
PATH=/bin:/usr/bin
```

Если вы создали нового пользователя, то скопируйте все скрытые файлы из `/etc/skel/*`, если они не скопировались автоматически при создании этого пользователя. В конфигурационных файлах из `/etc/skel` уже произведены все настройки по умолчанию, в т.ч. настроен PATH для более безопасной и надёжной работы.

По умолчанию системное приглашение к вводу безцветное и непривлекательное. Если вы хотите переделать его, то отредактируйте значение переменной `PS1`. Возможно использование escape-последовательностей для изменения цвета приглашения. Так же дополнительные последовательности, из которых строится приглашение к вводу в консоль:

Последовательность	Значение	Последовательность	Значение
\a	Издать звуковой сигнал	\@	Текущее время в 12-часовом формате <i>AM/PM</i>
\d	Текущая дата в формате <i>день_недели</i> <i>месяц</i> <i>число</i> : Mon May 16	\A	Текущее время в 24-часовом формате <i>часы:минуты</i>
\h	Имя локальной машины -- имя домена	\r	Имя пользователя
\H	Полное имя хоста	\v	Номер версии командной оболочки
\j	Число заданий, действующих в текущем терминале	\V	Номер версии и выпуска командной оболочки
\l	Имя текущего устр-ва терминала	\w	Имя текущего рабочего каталога
\n	Переход на новую строку	\W	Последняя часть имени текущего рабочего каталога
\r	Возврат каретки	\!	Имя текущей команды в истории
\s	Имя программы в программной оболочке	\#	Число команд, введённых в текущем сеансе
\t	Текущее время в 24-часовом формате	\\$	Выводит \$, если пользователь не является root, если это root, то выводит #

\T	Текущее время в 12- часовом формате	[и \]	Отмечает начало и конец (соотв.) последовательности непечатаемых символов
----	-------------------------------------	--------	---

3.1.3. Редактирование настроек bash

Интерпретатором по умолчанию в Calmira является bash. Его настройки расположены в `/etc/bashrc`, `/etc/skel/.bashrc` и `~/.bashrc`. Рекомендуем вам редактировать файл `~/.bashrc`.

3.1.4. Редактирование настроек Vim

В Calmira GNU/Linux по умолчанию используется редактор Vim. Его настройки расположены в файле `/etc/vimrc`. Строки:

```
set nu
set lbr
```

Предназначены для отображения номеров строк сбоку и переноса строк по словам соотв.

3.1.5. Редактирование настроек file system table

В файле `/etc/fstab` приведены базовые настройки монтирования файловых систем. Не изменяйте следующие строки, без которых система не сможет загрузиться корректно:

```
proc      /proc      proc      nosuid,noexec,nodev 0 0
sysfs     /sys       sysfs     nosuid,noexec,nodev 0 0
devpts    /dev/pts   devpts    gid=5,mode=620 0 0
tmpfs     /run       tmpfs     defaults        0 0
devtmpfs  /dev       devtmpfs  mode=0755,nosuid 0 0
```

3.1.6. Установка имени хоста

```
echo "calmira_pc" > /etc/hostname
```

Замените `calmira_pc` на имя хоста.

[Назад](#)

Настройка дистрибутива

[Домой](#)

[Далее](#)

Настройка часов

3.2. Настройка часов

При загрузке считывается информация из аппаратных часов - CMOS. Программно нельзя определить, какой часовой пояс используют часы CMOS, однако вы можете выполнить команду `hwclock --localtime --show` и сравнить результат с местным временем. Если он не совпадает - ваши часы, скорее всего, используют UTC.

3.2.1. `/etc/sysconfig/clock`

Создайте нужный файл, определяющий, использует ли CMOS UTC. Если не использует, то замените значение переменной `UTC` на 0.

```
cat > /etc/sysconfig/clock << "EOF"
# Begin /etc/sysconfig/clock

UTC=1

# Set this to any options you might need to give to hwclock,
# such as machine hardware clock type for Alphas.
CLOCKPARAMS=

# End /etc/sysconfig/clock
EOF
```

3.2.2. Выбор часового пояса.

Для выбора нужного часового пояса выполните следующий скрипт:

```
tzselect
```

Для сохранения нужного часового пояса выполните:

```
In -svf /usr/share/zoneinfo/<xxx> /etc/localtime
```

Замените `<xxx>` на ваш часовой пояс. Это значение было получено из вывода `tzselect`.

3.3. Настройка клавиатуры (раскладки) и шрифта в ТТУ

Дистрибутив Calmira GNU/Linux предназначен для работы в ТТУ в первую очередь. Некоторым пользователям может понадобиться переключение раскладки клавиатуры с английской на свою родную. В системе Calmira уже предустановлен нужный софт для настройки раскладок.

3.3.1. Файлы

Настройка раскладок производится в файле `/etc/sysconfig/console`.

Раскладки клавиатуры содержатся в директории `/usr/share/keymaps`. Шрифты в `/usr/share/consolefonts`.

3.3.2. Параметры в конфигурационном файле

См. предыдущий пункт.

- `UNICODE` - при значении 1, yes или true переводит консоль в режим UTF-8, что важно для некоторых раскладок и шрифтов, например, русских.
- `KEYMAP` содержит аргументы для программы `loadkeys`, которая загружает раскладки клавиатуры. Значение этого параметра - имя загружаемой раскладки.
- `FONT` - имя шрифта консоли. Содержит аргументы для программы `setfont`. Значение этого параметра - имя загружаемого шрифта.

3.3.3. Установка раскладки клавиатуры

Для поиска нужной раскладки выполнить:

```
find /usr/share/keymaps --type f
```

Выберите нужную раскладку. Например, для русской qwerty-раскладки, это будут варианты:

1. `ru` - переключение раскладки не установлено автором (либо же вовсе отсутствует).
2. `ruwin_alt_sh-UTF-8` - переключение раскладки по Alt+Shift.
3. `ruwin_cplk-UTF-8` - переключение раскладки по Caps Lock.
4. `ruwin-ct_sh-UTF-8` - переключение раскладки по Ctrl+Shift.

Как вы могли заметить, значение раскладки берётся из вывода предыдущей команды, но без расширения на конце и полного пути. Подставьте полученный результат как значение переменной `KEYMAP`, например:

```
KEYMAP="ruwin_alt_sh-UTF-8"
```

3.3.4. Установка шрифта

Для корректного отображения символов в родной для пользователя локализации (в т.ч. и для смысла настройки раскладок клавиатуры) нужно установить шрифт, поддерживающий эту локализацию.

Для просмотра всех установленных раскладок клавиатуры выполните:

```
ls /usr/share/consolefonts
```

Из списка выберите нужный шрифт. Например, для корректного отображения кириллицы и латиницы неплохо подходит шрифт *cyr-sun16*. Подставьте нужное значение в переменную FONT, например:

```
FONT="cyr-sun16"
```

3.3.5. Пример готового конфигурационного файла

```
# Begin /etc/sysconfig/console  
  
UNICODE="1"  
KEYMAP="ruwin_alt_sh-UTF-8"  
FONT="cyr-sun16"  
  
# End /etc/sysconfig/console
```

[Назад](#)
Настройка часов

[Домой](#)

[Далее](#)
Настройка inittab

3.4. Настройка inittab

Система инициализации SystemVinit читает параметры из файла `/etc/inittab`. Так же там описан запуск программ `sulogin` (программа для логина пользователей), `agetty` (альтернатива `getty` - программа, управляющая доступом к TTY).

По умолчанию этот файл уже создан. Для того, чтобы отредактировать его, выполните:

```
vim /etc/inittab
```

ВНИМАНИЕ!

Не рекомендуется редактировать этот файл!

Редактирование этого файла может быть полезно, если нужно изменить число TTY, например. По умолчанию их 6:

```
1:2345:respawn:/sbin/agetty --noclear tty1 9600
2:2345:respawn:/sbin/agetty tty2 9600
3:2345:respawn:/sbin/agetty tty3 9600
4:2345:respawn:/sbin/agetty tty4 9600
5:2345:respawn:/sbin/agetty tty5 9600
6:2345:respawn:/sbin/agetty tty6 9600
```

Для добавления новых TTY добавьте строку:

```
$NUMBER:2345:respawn:/sbin/agetty tty$NUMBER 9600
```

Замените `$NUMBER` на номер TTY.

Переключаться между TTY можно сочетанием клавиш `Ctrl + Alt + Fx` (где `x` - номер функциональной клавиши, равный номеру TTY), или `Alt + Стрелки влево/вправо`.

3.5. Установка дополнительных загрузочных скриптов

Пакет `additional-bootscripts` содержит сценарии (скрипты) инициализации, используемые дополнительными пакетами из портов дистрибутива. Предполагается, что вы будете использовать этот пакет вместе с совместимым пакетом `calm-bootscripts`, который установлен по умолчанию. Без `calm-bootscripts` не будет корректно загружаться и функционировать Calmira GNU/Linux.

Введение

Рекомендуется хранить архив `additional-bootscripts.tgz` с дополнительными загрузочными скриптами ДО конца настройки Calmira для своих нужд. Некоторым пакетам для нормального функционирования требуются эти скрипты. Когда нужно установить загрузочные сценарии для какого-то пакета/порта, просто перейдите в директорию с распакованными скриптами и выполните:

```
make install-<имя_пакета/скрипта>
```

Эта команда установит и настроит нужные сценарии и создаст нужные ссылки (в случае необходимости). Больше не нужно заморачиваться с самостоятельным написанием скриптов для SysVinit!

Установка и примеры использования

В данном разделе будут рассмотрены примеры использования `calm-additional-bootscripts`

Для начала требуется скачать нужный пакет (в портах его не может быть в принципе). Важная деталь: это НЕ пакет `calm-bootscripts.tgz`, который устанавливается по умолчанию и не подлежит удалению. Это совершенно другой пакет.

Для того, чтобы скачать его, выполните:

```
wget https://raw.githubusercontent.com/CalmiraLinux/calm-bootscripts/master/additional/additional-calm-bootscripts.tgz
```

Теперь следует распаковать его и перейти в рабочую директорию:

```
tar -xf additional-calm-bootscripts.tgz
cd additional-bootscripts
```

❑ ВНИМАНИЕ!

Имя распакованной директории будет не `additional-calm-bootscripts`, как ожидалось, а `additional-bootscripts`

После чего выполняете команду:

```
make install-<имя_скрипта/пакета>
```

Заменив `<имя_скрипта/пакета>` на имя того пакета, загрузочные скрипты для

которого вам нужно установить. Имена вы можете посмотреть в самом файле Makefile, либо же, если вы устанавливаете какой-либо пакет из системы портов, в файле setup.html будут инструкции по доп. настройке пакета. Если там упоминаются дополнительные загрузочные скрипты, то вы можете установить их, заменив <имя_скрипта/пакета> на имя того пакета.

Например, для пакета dhcpd из порта network/connecting/dhcpd команда будет выглядеть так:

```
make install-service-dhcpd
```

В некоторых случаях, название опции будет отличаться. В документации к порту это сказано. Читайте файл setup.html для получения доп. информации.

[Назад](#)

Настройка inittab

[Домой](#)

[Далее](#)

Управление пакетами

4.0. Управление пакетами

В данном разделе пойдёт речь об управлении пакетами в дистрибутиве.

Содержание

- 4.1 [Введение в порты](#)
 - 4.2 [Введение в srkg](#)
 - 4.3 [Создание порта](#)
 - 4.4 [Создание бинарного пакета для srkg](#)
-

4.1. Введение в порты

Порты - относительно молодой (в Calmira), но, в тоже время, очень удобный инструмент управления ПО. На данный момент, с помощью портов можно только устанавливать и удалять ПО, но в скором времени будут доступны новые функции.

4.1.1. Введение

Система портов находится в */usr/ports*. Всё ПО разделено на категории:

1. base - базовые порты, входящие в минимальную поставку дистрибутива;
2. console - консольные утилиты;
 - editors - консольные текстовые редакторы;
3. kernel - ядро и его компоненты;
4. multimedia - мультимедиа утилиты;
5. network - программы для работы с сетью;
 - browsers - интернет-браузеры и менеджеры зачек;
 - connecting - утилиты для подключения сети;
6. optional - дополнительный софт, не вошедший в базовую поставку дистрибутива, но который может понадобится после установки системы;
 - libs - не вошедшие в минимальную поставку библиотеки, которые могут понадобится при сборке других пакетов;
7. programming - утилиты для программирования (компиляторы, IDE, библиотеки и пр.);
 - python-modules - модули ЯП Python;
8. xorg - сервер X и другое ПО, такое как оконные менеджеры и рабочие окружения;
 - x11-de - рабочие окружения для X11 (GNOME, KDE, Xfce);
 - x11-minimal - Xorg - минимальная реализация;
 1. xlibs - дополнительные библиотеки для Xorg;
 2. xdrivers - драйверы для Xorg;
 - x11-wm - оконные менеджеры для X11 (WMaker, Fluxbox);

В каждой категории несколько подкаталогов с портом программы. Первые версии портов включают в себя только файл *install* с инструкциями по сборке, либо ещё *info* с информацией о пакете. В новых портах вместо *info* используется файл *config.sh*, который предназначается ещё и для занесения пакета в базу данных cpkg.

4.1.2. Установка программы из портов

Для того, чтобы установить какую-либо программу из порта, нужно выполнить:

```
# Переход в нужную директорию
cd /usr/ports/$КАТЕГОРИЯ/$ПРОГРАММА

# Выполнение инструкций сборки и установки
./install
```

Так же, в директории с портом могут находиться следующие файлы:

- install_doc.sh - для установки дополнительной документации;
- setup.html - инструкции по дальнейшей настройке пакета, которые трудно автоматизировать (нужно активное участие пользователя/администратора для выбора и изменения параметров, например), либо автоматизировать невозможно/не рекомендуется. Пригоден для открытия во всех известных браузерах. Типографские условные обозначения и прочее содержимое такое же, как и у этой документации.

- `setup.sh` (**не путать с предыдущим `setup.html`!**) - содержит инструкции для автоматизированной настройки пакета после установки для КОНКРЕТНОГО пользователя, от чьего имени запускается этот скрипт. Обычно создаёт нужные директории и конфигурационные файлы в домашней папке (`~`) того пользователя.
- `info` (устаревший) - содержит описание пакета и его версию. На данный момент заменён на `config.sh`, предоставляющий более обширные данные в более удобном формате [срkg](#).

Всё ПО из портов собирается из исходного кода. В некоторых случаях сборка может занять довольно много времени (так, например, пакет `gcc` на очень слабом железе может собираться до трёх дней). Однако, у сборки очень много преимуществ:

- Возможность оптимизации программы под своё железо;
- Большой контроль при установке;
- Выбор нужных функций и удаление ненужных (если программа это поддерживает).

Иногда пакеты предоставляют к установке дополнительную документацию. Она не всем нужна, поэтому мы оставили в портах возможность выбора пользователем: ставить её или нет. Система портов всегда спрашивает пользователя об этом. Если вы отказались от установки такой документации в будущем, но она вам пригодилась позже - перейдите в директорию с портом и выполните:

```
chmod +x install_doc.sh
./install_doc.sh
```

Этим командами вы установите документацию пакета.

Настоятельно советуется использовать порты вместо установки бинарных пакетов посредством `срkg`.

Кстати, портами скоро можно будет управлять и с помощью пакетного менеджера `срkg`. Ко второй версии `Calmira` планируется добавить просмотр информации и удаление пакета, установленного из порта, а так же множество других функций.

4.1.3. port-пакеты `срkg`

Пакетный менеджер `срkg` умеет устанавливать не только бинарные пакеты, но и собирать ПО из исходного кода. Но у такого подхода несколько минусов:

- Отсутствие контроля. Если в портах каждую инструкцию можно изменить так, как нужно человеку, то в случае с port-пакетами `срkg` так не случится. Для редактирования инструкций сборки придётся распаковывать пакет, править его, а потом запаковывать обратно.
- Нестабильность. Несмотря на то, что механизм port-пакетов уже достаточно стабилен, могут возникать ошибки.
- Небольшое число пакетов. Для пакетного менеджера `срkg` очень мало port-пакетов. Выбор очень невелик, а единственный репозиторий с port-пакетами было принято решение закрыть после выхода полноценной системы портов.

Рекомендуется использовать систему портов, а не сборку port-пакетов с помощью `срkg`.

4.2. Введение в срkg

срkg - написанный с нуля пакетный менеджер для Calmira GNU/Linux. Впервые он вошёл в состав Calmira GNU/Linux 2021.2 на смену менее надёжному и функциональному срkgi-tools. Из состава Calmira LX4 1.0 GNU/Linux он был убран ввиду своей нестабильности, но вернулся в Calmira LX4 1.1.

4.2.1. Обзор функций

1. Установка;
2. Удаление;
3. Просмотр информации.

Эти 3 функции - основные в любом пакетном менеджере (далее - ПМ). Так же есть несколько дополнительных:

1. Создание локального репозитория с пакетами срkg. Уже неактуально, так как из этого ПМ удалили скачивание пакетов из репозиториев. В следующих релизах это будет удалено.
2. Вывод списка установленных пакетов;
3. Очистка. Очистка кеша: `/usr/src`, `/var/cache/cpkg/archives`.

Все остальные функции предназначены для разработчиков и не могут быть использованы для работы с пакетами.

4.2.2. Функции

1. `install` - установить пакет;
2. `remove` - удалить пакет;
3. `info` - просмотр информации о пакете;
4. `help` - просмотр справки об использовании.

4.2.3. Установка пакета

Для того, чтобы установить пакет, введите:

```
срkg install package.txz
```

Для краткости можно заменить опцию `install` на ключ `-i`.

Пакет копируется в `/var/cache/cpkg/archives` и распаковывается там. После чего данные пакета копируются в нужные папки и добавляются в базу данных срkg.

4.2.3.1. Удовлетворение зависимостей пакета

Учитывайте то, что срkg НЕ обрабатывает зависимости. Он может отображать зависимости определённого пакета, но устанавливать их вы должны сами. срkg выводит список зависимостей пакета перед тем, как установить его. Без наличия *необходимых* зависимостей пакет не будет работать (либо будет работать некорректно). Так же пакет не будет работать без некоторых *рекомендуемых* зависимостей. *Опциональные* зависимости нужны для добавления каких-либо новых функций в исходный пакет. Если вы не хотите их ставить (или вам они не нужны), то опциональные зависимости можно не устанавливать.

4.2.4. Удаление пакета

Для удаления выполните:

```
сpkg remove package
```

Для краткости можно заменить опцию `remove` на ключ `-r`.

Читается `config.sh` пакета из базы данных `сpkg`, и на основе данных из этого файла удаляется пакет. После чего он выносится из базы данных `сpkg`.

Просмотр информации о пакете

Для того, чтобы узнать версию, сборщика пакета, зависимости и другую информацию, выполните:

```
сpkg info package
```

Для краткости можно заменить опцию `info` на `-l`.

4.2.5. Дополнительная информация

Для просмотра доп. справки выполните:

```
сpkg help
```

Для просмотра бОльших технических особенностей и характеристик зайдите в репозиторий `сpkg` на [GitHub](#).

[Назад](#)

Введение в порты

[Домой](#)

[Далее](#)

Создание порта

4.3. Создание порта

От помощи в создании порта для сборки какого-либо пакета мы никогда не откажемся. Если есть какие-либо заинтересованные в помощи люди - вы нам нужны.

4.3.1. Начальный этап

Создайте у себя на GitHub форк [этого](#) репозитория, после чего приступайте к работе. Советуем вам клонировать этот репозиторий себе на ПК для удобной работы:

```
git clone https://github.com/USERNAME/Ports
cd Ports
```

Замените *USERNAME* на имя пользователя GitLab.

После этих действий вы можете приступать к созданию порта. Хотим обратить ваше внимание на то, что строение порта должно повторять строение других портов.

4.3.2. Структура системы портов

Просим вас соблюдать структуру системы портов:

```
КАТЕГОРИЯ      <название категории брать из Введения в порты>
├── makeport.sh  <скрипт для автоматизации создания порта>
├── ИМЯ_ПАКЕТА  <указывается как аргумент скрипту makeport.sh>
│   ├── install <обязательно>
│   ├── install_doc.sh <опционально>
│   ├── setup.html <опционально>
│   ├── setup.sh <опционально>
│   └── config.sh <опционально>
```

Здесь приведена общая структура системы портов. КАТЕГОРИЯ и ИМЯ_ПАКЕТА - директории. КАТЕГОРИЯ - имя категории, в которую входит пакет с именем ИМЯ_ПАКЕТА.

Файл *makeport.sh* нужен для автоматизации создания порта. Он создаёт директорию с именем порта, а так же файл *install* и открывает его в вашем текстовом редакторе, установленном по умолчанию. Наличие этого файла необязательно. Синтаксис:

```
./makeport.sh $ИМЯ_ПАКЕТА
```

Файл *install* необходим. Он содержит инструкции по сборке ПО из исходного кода. Файл *install_doc.sh* опционален. Если у пакета есть дополнительная документация, которая не устанавливается по умолчанию, то в файле *install_doc.sh* содержатся инструкции по установке документации. Запускать его или нет - зависит от выбора пользователя. Он выбирает нужный ответ в диалоговом окне.

Файл *config.sh* содержит информацию о пакете. Она может быть добавлена в базу данных cpkg для удобного управления пакетом (так как система портов не умеет отображать информацию о пакете и удалять его, для таких целей используется cpkg). Наличие этого файла **ОБЯЗАТЕЛЬНО**, так же как и файла

install.

setup.html содержит дополнительные инструкции о дальнейшей настройке пакета после установки. Описаны команды и примеры файлов. Открывается в любом браузере. Файл *setup.sh* предназначен для автоматизированной настройки порта после установки. Как правило, настройка производится только для того пользователя, от имени которого был запущен этот скрипт

▣ **Смотрите также:**

Для установки какого-либо текстового редактора по умолчанию (в ТТУ) экспортируйте переменную `EDITOR`, значение которой - путь до исполняемого бинарного файла редактора.

4.3.3. Создание нужных файлов

Перейдите в одну из нужных директорий, выполните:

```
./makeport.sh $PACKAGE_NAME
```

Замените `$PACKAGE_NAME` на имя порта. Скрипт `makeport.sh` создаёт нужную директорию и записывает в файл нужную информацию, которая должна содержаться в файле порта.

Если вы не нашли этот скрипт, скопируйте его из любой другой директории.

Будет создан файл *install* в директории, имеющей имя порта, который вы хотите создать. Этот файл будет открыт для редактирования в вашем текстовом редакторе, установленном по умолчанию. Замените строки

```
# Port created by Linuxoid85  
#  
# (C) 2021 Michail [Linuxoid85] Krasnov
```

на строки с вашим именем.

Строки

```
wget  
tar -xf  
cd
```

Предназначены для того, чтобы подставить в их конец нужную информацию. `wget` скачивает архив с исходным кодом, `tar -xf` распаковывает его, а `cd` переходит в распакованную директорию.

Запишите в конец файла нужные инструкции для сборки. Доступны все команды `bash`.

Если можно установить дополнительную документацию, то в конце открытого в редакторе файла вызовите функцию `print_document_dial`:

```
print_document_dial
```

Он будет запрашивать у пользователя, ставить ли дополнительную документацию пакета.

Запишите нужные инструкции для установки документации в файл `install_doc.sh` и поместите его в ту же директорию, что и файл `install`.

Последняя строка файла `install` предназначена для добавления установленного порта в базу данных `cpkg` для дальнейшей удобной работы с ним (просмотра информации о порте, его удаление и пр.). Эта строка имеет вид:

```
package_add_port_db
```

После того, как вы сохраните изменения и выйдете из окна текстового редактора, будет открыт файл `config.sh` - в него запишите в него информацию о пакете. Структура этого файла аналогично тому, что используется в пакетном менеджере `crkg`. О создании этого файла читайте [в этой](#) странице руководства.

Так же можете написать `bash`-скрипт `setup.sh` для автоматизированной настройки порта (как правило, для создания основных конфигов и директорий) после установки. Эта настройка производится для конкретного пользователя, от чьего имени был запущен скрипт.

Если у пакета есть дополнительные настройки, которые трудно автоматизировать (нужно активное участие пользователя/администратора для выбора и изменения параметров, например), либо автоматизировать невозможно/не рекомендуется, то поместите рекомендации по настройке в файл `setup.html`. Формат такого файла должен быть примерно таким:

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Configuring $PACKAGE_NAME</title>
</head>
<link rel="stylesheet" href="/usr/share/update-ports/css/styles.css">
<body style="margin: 0px 150px 0px 150px">

<h1>Configuring $PACKAGE_NAME</h1><hr>

<h2>Заголовок второго уровня</h2>

<p>Абзац</p>

<table border="1" width="100%" cellpadding="5"><colgroup>
<col style="background-color:#eeddcc">
</colgroup>
<tr><td>
<pre>
Команда, которую нужно ввести в терминал, либо же пример кода
</pre></td></tr></table>

<table border="1" width="100%" cellpadding="5">
<tr><td>
<pre>
Пример текстового файла, либо вывода команды в терминал
</pre></td></tr></table>

</body></html>
```

4.3.4. Отправка изменений на github

Если вы клонировали репозиторий себе на ПК, выполните:

```
# Фиксирование изменений
git add .
git commit -m "Добавление порта с пакетом $PACKAGE_NAME"

# Отправка изменений себе на GitLab
git push
```

Замените `$PACKAGE_NAME` на имя порта с пакетом.

Если вы делали всё в Web, то отметьте изменения с помощью графического интерфейса GitLab.

Создайте запрос на слияние (pull-request). Создание порта завершено!

[Назад](#)

Введение в сpkg

[Домой](#)

[Далее](#)

Создание бинарного
пакета для сpkg

4.4. Создание бинарного пакета для cpkg

За время использования ОС вы можете захотеть сделать бинарный пакет для cpkg с какой-либо программой для её удобной установки. В данной статье рассказано, как сделать это. Несмотря на то, что основной упор в Calmira делается на сборку ПО из исходного кода, разработчики не препятствуют установке готовых бинарных пакетов, хотя их официально не предоставляют.

4.4.1. Введение

Пакет для cpkg представляет собой tar архив, сжатый методом xz. В архиве находится директория PKG, в которой находятся файлы config.sh, preinst.sh/reremove.sh (опционально), postinst.sh/postremove.sh (опционально), а так же директория pkg, в которой находится сам пакет. Дерево каталогов пакета должно совпадать с деревом каталогов операционной системы.

Назначение файлов и директорий:

- PKG - в этой директории находятся сам пакет и файлы описания, а так же файлы пред- и послеустановочных настроек.
- PKG/pkg - директория с пакетом
- PKG/config.sh - описание пакета
- PKG/preinst.sh - скрипт, выполняющийся до установки пакета. Может быть полезен для настройки системы или окружения перед установкой пакета. Наличие этого файла **опционально**.
- PKG/postinst.sh - скрипт, выполняющийся после установки пакета. Может быть полезен для настройки пакета. Наличие этого файла **опционально**.
- PKG/preremove.sh - скрипт, выполняющийся до удаления пакета. Может быть полезен для настройки системы или окружения.
- PKG/postremove.sh - скрипт, выполняющийся после удаления пакета. Может быть полезен для окончательной настройки системы/окружения, а так же удаления некоторых конфигурационных (и других) файлов, не вошедших в список файлов пакета.
- PKG/metadata.xz - в данном архиве содержится информация о совместимости пакета с дистрибутивом Calmira.

Тогда дерево каталогов пакета будет примерно таким:

```
some_package.tgz # <-- Архив с пакетом
├── PKG           # <-- Директория, в котором находится пакет
│   ├── config.sh
│   ├── {post,pre}inst.sh
│   ├── {post,pre}remove.sh
│   ├── metadata.xz
│   └── pkg       # <-- Директория, в котором находятся данные пакета
│       ├── usr
│       ├── etc
│       ├── var
│       └── ...
```

В случае, если это пакет с исходным кодом (port-пакет), в директории PKG наличие файла port.sh **обязательно**. В нём описываются инструкции по сборке и установке пакета.

Наличие шебанга обязательно:

```
#!/bin/bash
```

Пример:

```
#!/bin/bash

cd /var/cache/cpkg/archives # Переход в рабочую директорию
wget https://www.some.site/some_package.tar.gz # Скачивание пакета some_package.tar.gz с сайта https://www.some.site
tar -xvf some_package.tar.gz # Распаковка пакета с исходным кодом

cd some_package
./configure --prefix=/usr \
--bindir=/usr/bin \
--sysconfdir=/etc # Конфигурирование пакета
make # Сборка
make install # Установка
```

❑ ВНИМАНИЕ!

Мы **НАСТОЯТЕЛЬНО НЕ** рекомендуем использовать port-пакеты.
Если хотите автоматизировать сборку ПО из исходного кода, то
используйте [систему портов Calmira GNU/Linux](#).

4.4.1.1. Строение файла config.sh

В этом файле описывается сам пакет. В нём указывается имя пакета, версия, майнтейнер, описание и файлы пакета. Пример такого файла:

```
NAME=some
VERSION=1.0
MAINTAINER="Linuxoid85 <linuxoid85@gmail.com>"
REQ_DEPS="bash"
RECOM_DEPS="openssl freetype"
OPT_DEPS="coreutils"
TEST_DEPS="expect"
BEF_DEPS="cpkg"
CON_DEPS="wget make-ca"
PRIORITY=user
DESCRIPTION="Some package for test cpkg package manager"
FILES="/usr/bin/{some_pkg,test_cpkg} /usr/share/some_pkg/"
```

Описание переменных:

1. NAME - имя пакета;
2. VERSION - версия пакета/программы;
3. MAINTAINER - сборщик (сопровождающий) пакета;
4. REQ_DEPS - необходимые для работы пакета зависимости;
5. RECOM_DEPS - рекомендуемые зависимости;
6. OPT_DEPS - опциональные зависимости;
7. TEST_DEPS - необходимые для тестирования пакета зависимости (указывать только в port-пакетах);
8. BEF_DEPS - какие пакеты требуют, чтобы указанный пакет был собран и установлен до их сборки и установки;
9. CON_DEPS - зависимости, с которыми конфликтует данный пакет;
10. PRIORITY - приоритет пакета;
11. DESCRIPTION - описание пакета;
12. SITE - сайт пакета, либо ссылка, откуда можно скачать оригинал/архив с исходным кодом, из которого собирался пакет;
13. FILES - все файлы пакета.

Если в одной директории (чаще всего это /bin, /usr/bin, /etc и пр.) несколько файлов из пакета, то нет смысла дублировать пути до этих файлов. Проще объединить их в массив. К примеру, есть директория /usr/bin, в которую устанавливаются файлы some_pkg и test_cpkg. В строке FILES="..." они объединены в массив /usr/bin/{some_pkg,test_cpkg}. То есть, файлы перечисляются в фигурных скобках {}.

Так же, все отдельные файлы разделяются между собой пробелами.

В версии cpkg 1.0ra4 была добавлена функция просмотра зависимостей пакета (при установке, удалении и просмотра информации о пакете).

Типов зависимостей несколько:

Наименование	Объяснение	В какой строке вывода
REQ_DEPS	Необходимые зависимости. Если вы устанавливаете бинарный пакет, то необходимые зависимости можно установить после, но если вы ставите port-пакет (либо порт), то поставьте сначала необходимые зависимости, а только потом пакет.	Необходимые:
RECOM_DEPS	Рекомендуемые зависимости. Советуется устанавливать их.	Рекомендуемые:
OPT_DEPS	Опциональные зависимости. Устанавливать их необязательно, они лишь служат для добавления определённого функционала. Порядок установки таких зависимостей в большинстве случаев не важен (кроме port-пакетов, где опциональные должны быть установлены पहले).	Опциональные:
TEST_DEPS	Зависимости, необходимые для тестирования пакета.	Для тестирования:
CON_DEPS	Зависимости, которые конфликтуют с данным пакетом. Их желательно удалить во избежание сбоев в работе как их самих, так и (в некоторых случаях) Calmira GNU/Linux	Конфликтует
BEF_DEPS	Необязательные зависимости, определяющие, что указанный port-пакет должен собран ДО установки/сборки тех зависимостей	Установлен перед ними:

Помимо этого, есть приоритет пакета - `PRIORITY`. Только два значения: `system`, если пакет системный и `user` - если пользовательский. Т.е., в первую группу входят все пакеты, которые обеспечивают корректную работу МИНИМАЛЬНОЙ системы. Системные пакеты удалить нельзя. Их можно лишь обновлять до новой версии и просматривать информацию о них. А с пользовательскими пакетами (приоритет `user`) можно делать всё, что угодно.

Настоятельно рекомендуем использовать пользовательский (`user`) пакета! Все системные пакеты собираются одним мейнтейнером для поддержания работоспособности системы. Как правило, большинство пакетов от сторонних сборщиков с приоритетом `system` не только не являются системными, но и при удалении этих пакетов в системе не произойдёт ничего.

4.4.2. Сборка пакета

Создайте некую директорию `/usr/PKG/pkg`, в которую, впоследствии, будет установлен пакет. И выполните процедуру сборки. Но тут загвоздка. Использовать альтернативный префикс `/usr/PKG/pkg` в процессе конфигурирования (напр., `configure`) сделает пакет нерабочим. Будут сделаны неправильные ссылки, файлы будут скопированы не в те директории. В каталоге `/usr/PKG/pkg` нужно создать зеркальную копию `/usr`, либо же корня (`/`) - в случае, если пакет системный. Поэтому при конфигурировании пакета (напр., создании `Makefile`) используйте префикс `/usr` или `/` (по умолчанию). А потом с помощью `make DESTDIR=/usr/PKG/pkg install` сфальсифицировать установку пакета в корень, либо же в указанные директории (когда как на самом деле пакет будет установлен в `/usr/PKG/pkg`).

В Calmira GNU/Linux LX4 1.1 используется стандартная структура директорий, где `/bin`, `/sbin`, `/usr/bin` и `/usr/sbin` - разные каталоги. Поэтому если бинарные файлы должны быть установлены в `/bin`, бинарные файлы для суперпользователя в `/sbin`, и пр. При необходимости (если некоторые файлы пакета должны быть установлены в `/bin`, `/sbin`, `/lib`) используйте ключи `--bindir=/bin`, `--sbindir=/sbin`, `--libexecdir=/lib`.

Выполните процедуру конфигурирования:

```
./configure --prefix=$PREFIX \  
--bindir=$BINDIR \  
--sbindir=$SBINDIR
```

Если вы используете систему сборки `meson`, то замените процедуру конфигурирования на корректную для этой системы сборки.

Используйте нужные вам ключи. В любом случае, процедура конфигурирования должна быть точно такой, какой она является по умолчанию.

Если конфигурирование прошло успешно, выполните сборку:

```
make
```

Если используете `meson`, то сборка будет такой:

```
ninja
```

После чего установите пакет, только не в те директории, что были указаны `configure/meson`, а в `/usr/PKG/pkg`:

```
make DESTDIR=/usr/PKG/pkg install
```

Вместо `DESTDIR` могут быть переменные `INSTALL_DIR`, `prefix` и пр. Или может не быть вообще, тогда файлы придётся находить и копировать вручную. О подробной информации смотрите в документации к установке пакета.

Теперь создайте файл `config.sh` в директории `/usr/PKG` и запишите в него информацию о пакете:

1. Название пакета;
2. Версия пакета;
3. Описание пакета;
4. Сборщик пакета;
5. Зависимости пакета;
6. Файлы пакета.

И прочую информацию. Запишите её в соответствии с пунктом "Введение".

Для того, чтобы записать список файлов в раздел `FILE`, перейдите в директорию `pkg` и выведите список файлов:

```
cd pkg  
  
find
```

Запишите вывод в раздел `FILE`:

```
FILE="/usr/bin/file  
....  
....  
"
```

Удалите из вывода все точки в начале, а так же все файлы, относящиеся к другим пакетам. Так же удалите все ``/bin``, ``/usr/bin``, ``/usr/share`` и пр. Оставьте только файлы. Одна ошибка - и при удалении пакета можно сломать либо какой-то отдельный пакет, либо всю систему.

Если у пакета должны быть *preinstall* и *postinstall* скрипты для настройки окружения и пакета соотв., то создайте их. Назовите preinst.sh и postinst.sh соотв.

И соберите пакет:

```
# Переход из /usr/PKG на уровень ниже (в /usr):  
cd ..  
  
# Сборка пакета  
tar -cf PKG $PACKAGE_NAME.txz -J
```

Замените \$PACKAGE_NAME на имя. Оно должно включать в себя имя и версию пакета, например: some_1.0.txz.

[Назад](#)

Создание порта

[Домой](#)

[Далее](#)

Администрирование
системы

Администрирование Calmira GNU/Linux

В данном разделе приведены базовые сведения об администрировании
системы Calmira GNU/Linux.

Содержание

5.1 [Права доступа](#)

5.2 [Управление процессами](#)

5.1. Права доступа

В данной инструкции приведены базовые сведения об управлении правами доступа и привилегиях суперпользователя. Под *управлением доступом* подразумевается механизм принятия решений, связанных с безопасностью, в первую очередь.

Работа с правами доступа на файлы

Введение

В отличие от DOS и ей подобных систем, UNIX проектировалась как многопользовательская система. Поэтому в ней должны быть хотя бы примитивные подсистемы для управления правами доступа к файлам, чтобы, например, *обычный* пользователь <user1> не удалил случайно (либо по злому умыслу) файлы из /bin, либо любые другие системные файлы или файлы других пользователей. На самом деле, примеров можно привести много.

Стандартная модель управления доступом в UNIX в течение десятилетий практически не изменилась. С некоторыми изменениями она доступна нам и сегодня и применяется как в Linux, так и в других UNIX-подобных/образных системах. Ниже представлена модель доступа.

- Решение по управлению доступом зависят от того, какой пользователь пытается выполнить операцию.
- Объекты (например, файлы, процессы и пр.) имеют владельцев, которые имеют широкий (но не обязательно неограниченный) контроль над своими объектами.
- Вы являетесь владельцем создаваемых вами объектов.
- Только пользователь root может действовать как владелец любого объекта и выполнять важные административные операции с системой.

Пользователь root

Пользователь root имеет больше всего привилегий в системе. Его так же называют "*суперпользователем*". Идентификатор UID равен нулю.

Суперпользователю (т.е. любому процессу, для которого текущий идентификатор UID = 0) позволяется выполнять любую допустимую операцию для любого файла или процесса (но некоторые операции, такие как исполнение файла как программы, для которого не установлен бит разрешения на запуск, запрещены даже для root). Как только процесс root изменит своего владельца и станет обычным пользовательским процессом, он не сможет восстановить прежнее привилегированное состояние.

Вход в учётную запись пользователя root

Несмотря на то, что во многих дистрибутивах Linux эта учётная запись по умолчанию отключена, в Calmira GNU/Linux этого не планируется. Однако мы вам **не советуем** пользоваться этой учётной записью для решения определённых задач, которые решаются куда более безопасными методами.

Лучше всего получать доступ к учётной записи (УЗ) root с помощью команды su. При вызове без аргументов она запросит у вас пароль от УЗ root. Пароль по умолчанию: root (т.е. такой же, как и название УЗ). **ОБЯЗАТЕЛЬНО** поменяйте его (об этом в следующем пункте). Так же, с помощью su можно получать доступ к УЗ и других пользователей. Для этого выполните:

```
su $USER
```


Заменяв `$USER` на имя нужного пользователя.

Использование утилиты `sudo`

Программа `sudo` принимает в качестве аргумента командную строку, выполняемую с правами `root` (или правами любого другого ограниченного пользователя). Она проверяет файл `/etc/sudoers`, в котором перечислены пользователи, имеющие разрешение использовать `sudo` и команды, которые им можно запускать на каждом хосте. Если предлагаемая команда разрешена, `sudo` запрашивает собственный пароль пользователя и выполняет команду.

По умолчанию, `sudo` позволяет выполнять команды без введения пароля пользователя в течение пяти минут, хотя время тайм аута можно менять (но не советуется).

Пример записи в `sudoers`

В данном пункте представлены сведения по настройке `sudo`

По умолчанию, эта программа не установлена. Установите её из порта `optional/sudo`. После сборки и установки Система портов спросит у вас, настраивать ли этот порт или нет. Выберите "Да". Будет создана базовая конфигурация. Но вам может понадобиться перенастроить порт под свои нужды.

Настройки хранятся в `/etc/sudoers`. Редактировать файл `sudoers` "вручную" — плохая практика. Для редактирования есть специальная команда `visudo`. Она сама знает, где расположен файл `sudoers`, и отслеживает, чтобы вы редактировали копию этого файла во избежание ошибок, а основной файл всегда был целостным.

Программа `visudo` использует тот редактор, который прописан в переменной окружения `EDITOR`. В Calmira GNU/Linux по умолчанию это `/usr/bin/vim`, но вы можете установить свой, выполнив:

```
export EDITOR=ваш_редактор
```

После чего выполните от имени `root`:

```
visudo
```

Если вы допустите ошибку, то, при выходе из редактора получите сообщение вида:

```
visudo: >>> /etc/sudoers: syntax error near line 73 <<<
What now?
```

Полномочия для каждого пользователя описываются в следующем виде:

- Пользователь, к которому относится запись;
- Хосты, на которые распространяется запись;
- Разрешённые команды;

Формат текстовой записи будет такой:

```
[пользователь] [хост]=([кем может стать]) [что может сделать]
```

Например:

```
user1 ALL=(root) /bin/mount
```

Эта строка означает, что пользователь user1 может выполнять с правами root команду /bin/mount. **Обязательно** указывайте *полный путь* до команды!

❏ ВНИМАНИЕ!

К сожалению, очень большое число пользователей отключают запрос пароля (NOPASS). Они это делают ради лени, так как не хотят вводить длинные и сложные пароли. Мы настоятельно **НЕ РЕКОМЕНДУЕМ** использовать такой подход, так как он очень небезопасный. Если у ПК окажется злоумышленник, то он может привести ваши данные или всю ОС в неработоспособное состояние.

Владельцы объектов

Чтобы получить доступ к определённому объекту (далее будет употребляться термин "*файл*" и всё про них, другие объекты затронуты не будут) в Linux, используются разрешения, которые назначаются трём объектам: *файлу*, *группе* и другому объекту (т.е. всем остальным). Но перед этим нужно знать про владельца файла (директории не упоминаю специально, так как это тоже файл - концепция "Всё есть файл" в UNIX-подобных/образных).

В Linux у каждого файла есть два владельца: *пользователь* и *группа*. Они устанавливаются при создании файла. Каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа:

1. **Владелец** - набор прав для владельца файла - пользователя, который создал его или сейчас установлен его владельцем. Обычно владелец имеет все права: чтение, запись и исполнение;
2. **Группа** - любая группа пользователей, существующая в ОС и привязанная к файлу;
3. **Остальные** - все пользователи (кроме владельца и юзеров, входящих в группу файла);

Только пользователь *root* (он же суперпользователь) может работать со всеми файлами независимо от набора их полномочий.

Пользователь, создавший файл, становится его владельцем, так же как и первичная группа, в которую входит этот юзер. Чтобы определить, есть ли у вас, как у пользователя, права доступа к файлу, оболочка проверяет владение им. Принцип работы таков:

1. Оболочка проверяет, являетесь ли вы владельцем файла, к которому запрашивается доступ. Если владельцем являетесь, то оболочка прекращает проверку и вы получаете разрешения. Если вы не являетесь владельцем, но входите в группу, у которой есть доступ к файлу, то вы получаете доступ к файлу с теми же разрешениями, что и у той группы.
2. Если же вы не являетесь ни пользователем, ни владельцем группы, то вы получаете права других пользователей (``other``).

Каждый пользователь может получить полный доступ к файлу в том случае, если доступ ему разрешён. Либо же, если он является владельцем файла.

Чтобы увидеть пользователя и группу-владельца файла, выполните:

```
ls -l
```

Чтобы просмотреть тоже самое, но для конкретного файла:

```
ls -l $FILE
```

Замените *\$FILE* на имя нужного файла.

Изменение владельца файла

Очень часто системным администраторам (и обычным пользователям, но реже) требуется по каким-то причинам изменить владельца файла. Для смены владельца используется программа `chown` (change owner).

Синтаксис `chown`

```
chown пользователь опции /путь/к/файлу
```

Ключи и опции `chown`

- `-c --changes` - подробный вывод всех выполняемых операций;
- `-v --verbose` - описание действий и вывод подробных данных о каждом обработанном файле;
- `-R --recursive` - рекурсивная обработка всех подкаталогов;
- `-f --silent --quiet` - минимум информации, выводимой на экран, даже сообщения об ошибках;
- `--dereference` - изменять права для файла, к которому ведёт символическая ссылка вместо самой ссылки (*по умолчанию*);
- `--no-dereference -h` - изменять права симлинков, но оставить неизменными файлы, на которые указывают эти симлинки;
- `-L` - переходить по всем симлинкам на каталоги;
- `-H` - если передан симлинк на каталог, перейти по нему;
- `-P` - не переходить по символическим ссылкам на каталоги (*по умолчанию*);

Примеры использования `chown`

Допустим, есть два пользователя: `root` и `liveuser`. В директории `/root` создана поддиректория `/root/liveuser`. Там ещё несколько файлов. Эта поддиректория принадлежит пользователю `root`, как и все файлы в ней. А надо сделать так, чтобы принадлежала пользователю `liveuser`. Выполнить в терминале:

```
sudo chown liveuser /root/liveuser
```

Если вы хотите видеть подробную информацию о проделанном действии, выполните:

```
sudo chown -v liveuser /root/liveuser
```

Но вы сменили владельца только для каталога `/root/liveuser`. А все подпапки и другие файлы, которые находятся в нём, так же принадлежат пользователю `root`. В тех подкаталогах ещё какие-то каталоги находятся. Можно, конечно, для каждого файла/каталога изменить владельца персонально, но это муторно, а если файлов много, то никто не будет с этим заморачиваться. Но проще же выполнить `chown` с ключом `-R`, дабы эта утилита рекурсивно обошла все подкаталоги и файлы, сменив их владельца:

```
sudo chown -R liveuser /root/liveuser/
```

Опять возвращаем всё в то состояние, которое было до экспериментов с `chown`:

```
sudo chown -Rv root /root/liveuser
```

Если вы хотите поменять не только владельца, но и группу файла, то запишите имя пользователя и имя группы через двоеточие:

```
sudo chown -v liveuser:liveuser /root/liveuser
```

Теперь измените группу и владельца на liveuser только для тех файлов, у которых владелец и группа root в каталоге /root/liveuser/:

```
sudo chown --from=root:root liveuser:liveuser ./
```

Изменение группы файла с помощью chgrp

Для изменения группы файла используется команда `chgrp` (*change group*). В отличие от предыдущей описанной команды `chown`, для `chgrp` требуется только имя группы, имя пользователя не нужно.

Синтаксис chgrp

```
chgrp опции имя_группы /путь/к/директории
```

Вместо имени группы можно указать её *GID* (идентификатор группы)

Ключи и опции chgrp

- `-c --changes` - подробно описывать действия для каждого файла, чья группа изменяется;
- `-f --silent --quiet` - не выдавать сообщения об ошибке для файлов, чья группа не может быть изменена;
- `-h --no-dereference` - работать с символическими ссылками, а не файлами, на которые они указывают. Данная опция доступна, только если используется `lchown`;
- `-v --verbose` - подробно описывать действие или отсутствие для **каждого** файла;
- `-R` - рекурсивно изменить группы для каталогов и их содержимого, а возникающие ошибки не прекратят работу программы;
- `-L` (**используется вместе с -R**)- для каждого файла, указанного или пользователем, или встреченного при обходе дерева каталогов, если этот файл является симлинком на каталог, изменить группу самого этого каталога и всех файлов в его иерархии;
- `-h` - для каждого файла, являющегося символической ссылкой, изменить группу самой этой ссылки, а не объекта, на который она указывает, а если система не поддерживает группы для симлинков, то ничего не делать;
- `--` - завершение списка опций.

Примеры использования chgrp

Опять же, перед изменением группы-владельца файла, проверьте с помощью `ls` текущую группу:

```
ls -l 'ИМЯ_ФАЙЛА'
```

Теперь добавьте некую группу `mygroup`:

```
sudo groupadd mygroup
```

И сделайте группу `mygroup` владельцем директории `/root/liveuser`:

```
sudo chgrp mygroup /root/liveuser
```

Как и в случае `schown`, группа стала владельцем только каталога `/root/liveuser`, но не файлов и подкаталогов этой директории. Чтобы изменить группу-владельца файла, можно, конечно, сделать вручную, а можно воспользоваться рекурсивным способом. За это отвечает ключ `-R`:

```
sudo chgrp -R mygroup /root/liveuser/
```

Однако, рекурсивный метод не назначает прав на симлинки, поэтому у них сохраняется прежняя группа. Чтобы установить новую группу и на символические ссылки, добавьте ключ `-h`:

```
sudo chgrp -Rh mygroup /root/liveuser
```

Для того, чтобы скопировать группу владельцев директории, воспользуйтесь ключом `--reference`:

```
sudo chgrp --reference /home/liveuser /root/liveuser
```

Специальные права доступа

Для того, чтобы позволить *обычным* пользователям выполнять программы от имени суперпользователя, не зная его пароль, были созданы биты SUID и SGID.

1. Если установлен бит SUID, то при выполнении программы, ID пользователя меняется на ID владельца файла. *Фактически*, это позволяет обычным пользователям запускать программы от имени суперюзера.
2. SGID работает аналогичным способом, но разница в том, что пользователь считается членом группы, с которой связан этот файл, а не групп, которым он действительно принадлежит. Если флаг SGID установлен на каталог, все файлы, созданные в нём, будут связаны с группой каталога, а не пользователя. Такое поведение используется для организации общих папок.
3. Бит Sticky-bit так же используется для создания общих директорий. Когда он установлен, пользователь может только создавать, читать и выполнять файлы, но не может удалять их, если они принадлежат другим пользователям.

Условные обозначения

Так же как и в предыдущих пунктах, опять просмотрите список файлов в подробном режиме:

```
ls -l
```

В первой колонке вывода `ls` выведены права на файл по следующей схеме:

Табл. 1 - буквенные обозначения прав на файлы

Обозначение	Расшифровка
---	Совсем нет прав
--x	Разрешено выполнение файла как программы, но не чтение и запись (в файл)
-w-	Разрешена запись в файл, но не чтение и не исполнение его как программы

-wx	Разрешена запись (изменение) в файл и его исполнение как программы, но не чтение
r--	Разрешено только чтение файла
r-x	Разрешено чтение и исполнение файла, но не запись в него (т.е., его изменение)
rw-	Разрешено чтение и запись в файл, но не его исполнение как программы
rwX	Разрешено всё: чтение, запись и исполнение

□ Примечание

Запись означает здесь *запись в файл*. Это поведение ещё можно охарактеризовать как *изменение файла*

Табл. 2 - буквенные обозначения прав

Буква	Обозначение
r	Чтение (Read)
w	Запись (Write)
x	Исполнение (eXecute)
s	Исполнение от имени суперпользователя (Superuser) - дополнительный

Табл. 3 - буквенные обозначения категорий пользователей

Буква	Обозначение
u	Владелец файла (User)
g	Группа файла (Group)
o	Все остальные (Other)

Действия так же можно записывать и с помощью цифр. Первая цифра указывает права для пользователя, вторая - для группы и третья для всех остальных.

Табл. 4 - цифровые обозначения прав доступа к файлам

Цифра	Обозначение
744	Разрешить всё для владельца, остальным только чтение
755	Разрешить всё для владельца, остальным только запуск (исполнение)
764	Разрешить всё для владельца, чтение и запись для группы, остальным только чтение
777	Всем разрешено всё

Изменение прав доступа к файлам с помощью **chmod**

chmod - программа для изменения прав доступа к файлам.

Синтаксис **chmod**

chmod опции <категория><действие><флаг> файл

Ключи и опции chmod

- -c - выводить информацию обо всех изменениях;
- -f - не выводить сообщения об ошибках;
- -v - выводить максимум сообщений о ходе работы chmod;
- -R - рекурсивный метод;
- --reference - взять маску прав из указанного файла;
- --preserve-root - не выполнять рекурсивные операции для корня файловой системы;

Примеры использования chmod

Разрешить выполнение определённого скрипта или бинарника пользователю (что лично мне надо очень часто). Допустим, у нас есть некий ELF файл, который называется binary. И его надо сделать исполняемым. Для начала выполните `ls -l binary`, дабы посмотреть его текущие права.

```
chmod u+x binary
```

Теперь запускаем его:

```
./binary
```

Тоже самое, но с помощью цифр:

```
chmod 766 binary
```

А теперь посмотрите, как изменились права на файл:

```
ls -l binary
```

Отберите все права у binary:

```
chmod ugo-rwx binary
```

Файлы с правами 000 недоступны никаким пользователям, кроме суперпользователя и владельца. Вернём права обратно:

```
chmod 755 binary
```

Для применения расширенных прав так же используется chmod. Нужно указать четырёхзначный аргумент в chmod, первая цифра относится к специальному разрешению, например:

```
sudo mkdir binary.d  
sudo chmod 2755 binary.d
```

Эта команда добавит разрешение SGID на каталог binary.d.

Напомним цифровые обозначения SUID, SGID и Sticky bit :

Табл. 5 - цифровые обозначения SUID, SGID и Sticky bit.

Число	Обозначение

1	Sticky bit
2	SGID
4	SUID

Примеры использования chmod:

- SUID:

```
chmod u+s
```

- SGID:

```
chmod g+s
```

- Sticky bit:

```
chmod +t
```

Дополнительная информация про SGID, SUID и Sticky bit

SUID

Рассмотрим обычный для администратора пример. Нужно поменять пароль своей/чужой учётной записи. Для этого нужно отредактировать файл `/etc/shadow`. Но он доступен только суперпользователю. Однако, в утилите `/bin/passwd` применяется разрешение SUID. Это означает, что пользователь **временно** получит права `root`, что позволяет ему без проблем и ограничений редактировать `/etc/shadow`. Так ведь, если `passwd` именно это и делает? - редактирует нужный файл. Вот в этом и опасность разрешения SUID: с одной стороны, оно, однозначно, полезно и удобно, но довольно опасно. **Поэтому пользуйтесь им с осторожностью.**

SGID

SGID - идентификатор группы. SGID даёт пользователю, который исполняет определённый файл, разрешения владельца группы этого файла, что означает, что SGID позволяет выполнить примерно тоже самое, что и SUID. Но, как ни странно, SGID для этой цели если используется, то очень редко, но как в случае с SUID, он применяется к некоторым системным файлам по умолчанию.

Однако, SGID может быть полезен тогда, когда он применяется к каталогу: вы можете использовать его для установки владельца группы по умолчанию для файлов и подкаталогов, созданных в этом каталоге. По умолчанию, когда пользователь создает файл, его эффективная первичная группа устанавливается как владелец группы для этого файла.

Sticky bit

Это разрешение полезно для защиты файлов от *случайного* удаления в среде, где несколько пользователей имеют права на запись в одну и ту же директорию; если применяется закреплённый sticky-bit, пользователь может удалить файл, только если он является пользователем-владельцем файла/каталога, в котором содержится файл. Именно поэтому он применяется, скажем, в `/tmp`.

Без sticky bit, если пользователь может создавать файлы в каталоге, он также может удалять файлы из этого каталога. В общедоступной групповой среде это может раздражать. Представьте себе пользователей `user1` и `user2`, которые оба имеют права на запись в каталог `/data/account` и получают эти

разрешения благодаря участию в группе `account`. Поэтому `user1` может удалять файлы, созданные `user2`, и наоборот.

Когда вы применяете `sticky bit`, пользователь может удалять файлы, только если выполняется одно из следующих условий:

1. пользователь является владельцем файла;
2. пользователь является владельцем директории с этим файлом.

В первой колонке вывода `ls -l` у файлов с разрешением `Sticky bit` будет буква `t` в той позиции, где вы обычно видите разрешение на выполнение для других.

[Назад](#)

Администрирование
системы

[Домой](#)

[Далее](#)

Управление процессами

5.2. Управление процессами

Введение

Процесс в Linux (как и в UNIX) - это программа, которая выполняется в отдельном виртуальном адресном пространстве. Такая абстракция, с помощью которой можно управлять памятью, временем работы процессора и ресурсами ввода/вывода.

Процессы защищены друг от друга и крах одного процесса не повлияет на работу других и всей системы в целом. Один процесс не может прочитать (или записать) данные другого процесса без его разрешения на это, а санкционированное взаимодействие друг с другом разрешено системой.

Процесс состоит из адресного пространства и набора структур данных внутри ядра. Адресное пространство - набор страниц памяти, выделенных ядром для использования процессу.

Для создания процессов используются два системных вызова: `fork()` и `exec`. `fork()` создаёт новое адресное пространство, которое полностью идентично адресному пространству другого процесса. После выполнения `fork()` получается два абсолютно одинаковых процесса - *основной* и *порождённый*.

Каждый процесс в системе имеет уникальный идентификатор - *Process ID*. Этот номер используется ядром и специализированными утилитами для работы с процессами.

Пользователь может управлять поведением процессов, запущенных от его имени. Но только `root` имеет право управления всеми процессами - и его, и других пользователей. Управление процессами производится специально предназначенными для этого утилитами, а так же некоторыми командами оболочки (shell)

Выполнение процессов

Есть несколько типов процессов:

1. **Процессы переднего плана (foreground)**. На переднем плане в каждый момент для текущего терминала может выполняться только один процесс. Однако пользователь может перейти в другой TTY и запустить ещё один. ППП получает информацию с клавиатуры (`stdin`) и выводит результат на экран (`stdout`). Такие процессы иногда называют *интерактивными*;
2. **Фоновые (автоматические) процессы** отвязаны от терминала и не ожидают пользовательского ввода, а командная оболочка не ожидает окончания запущенного процесса, что позволяет пользователю запустить ещё один процесс;
3. **Отложенные процессы** в данный момент не выполняются и в данный момент остановлены. Их можно восстановить как на переднем плане, так и в фоне. Возобновление приостановленного процесса не изменит его состояния - он продолжит свою работу с того места, с которого был восстановлен;
4. **Демоны** - особый тип фоновых процессов, которые запускаются при старте системы и продолжают работу в виде службы;

PID, PPID

Ядро назначает каждому процессу уникальный идентификатор (PID - Process Identification). На сегодня Linux использует концепцию пространства имён процесса, которая ещё больше ограничивает способность процессов видеть и влиять друг на друга. Недостаток в том, что процесс может иметь разные PID в зависимости от пространства имён наблюдателя.

Для того, чтобы создать новый процесс, существующий должен клонировать сам себя. Клон может заменить выполняемую программу другой.

Исходный процесс называется *родительским*, клон - *дочерним*. Каждый дочерний процесс имеет Parent Process ID (PPID).

Другие идентификаторы

Существует ещё несколько идентификаторов процессов, которые пригодятся при администрировании Calmira и других дистрибутивов Linux.

UID (User ID) - это идентификатор пользователя, создавшего данный процесс - копия UID родительского процесса. Менять атрибуты могут только его владелец (тот, кто его создал) и суперпользователь.

EUID (Effective User ID) - текущий пользовательский идентификатор процесса, предназначенный для определения, к каким ресурсам и файлам есть право доступа в данный момент. У большинства процессов значения UID и EUID одинаковые, кроме тех, у которых установлен бит смены идентификатора пользователя `setuid`.

Так же может встречаться довольно редкий и нестандартный параметр FSUID, определяющий возможности работы с ФС, но он используется очень редко.

GID (Group ID) - идентификационный номер группы данного процесса. EGID связан с GID так же, как EUID с UID.

Жизненный цикл процесса

Все процессы, кроме `init`, создаются с помощью системного вызова `fork()`. Вызывая эту функцию, процесс создаёт свой дубликат, называемый дочерним процессом (см. предыдущие пункты). Дочерний процесс - копия родительского, но с отличиями:

1. У дочернего процесса свой PID;
2. PPID дочернего процесса равен PPID'у родительского.

После вызова `fork()` может посредством системного вызова `wait()/waitpid()` приостановить своё выполнение до завершения дочернего процесса, или продолжать свою работу независимо от него. Процесс завершает своё выполнение вызовом функции `exit()`

Использование программ для работы с процессами

kill для отправки сигналов; pidof для определения PID процессов

Два основных сигнала - `SIGTERM` и `SIGKILL`. Первый запрашивает остановку процесса, который работает. Этот сигнал может быть проигнорирован. Процессу даётся некоторое время, чтобы он корректно завершился. Второй (`SIGKILL`) заставляет процесс прекратить своё выполнение немедленно. Программа не может проигнорировать этот сигнал.

Синтаксис kill

```
kill <сигнал> PID
```

Если сигнал не указан, то будет использоваться тот, что по умолчанию - `SIGTERM`. Если это не помогает, что можно использовать одну из следующих команд для принудительного завершения процесса:

```
kill SIGKILL PID
```

```
# Или  
kill -9 PID
```

Наиболее часто используемые сигналы

1. **1** (HUP) - перезапустить процесс;
2. **9** (KILL) - убить процесс;
3. **15** (TERM) - остановить процесс;

Дополнительные сведения

Вы можете не знать PID нужного процесса. В таком случае можете использовать системные мониторы `top`, `htop`, которые предоставляются в портах, либо использовать "изкоробочные" решения, такие как `pidof`:

```
pidof vim
```

Команда выше выведет вам PID текстового редактора Vim.

Можно поступить проще. Если вы не знаете PID нужного процесса, то можно не выполнять последовательно `pidof` и `kill` - можно объединить их в одну команду:

```
kill $(pidof vim)
```

Использование ps для мониторинга текущих работающих процессов

Программа ps один из основных инструментов системного администратора для контроля процессов. Эта программа будет доступна даже тогда, когда не будет смонтирован раздел /usr (либо его вообще не будет). С помощью этой программы можно получить информацию об имени процесса, его PID, UID, приоритете, используемой памяти и пр.

Для просмотра списка всех процессов используется опция aux. а означает, что требуется вывести все процессы, x - даже те процессы, отсоединённые от управляющего терминала, u - фильтрация по имени или идентификатору пользователя.

На скриншоте ниже показан вывод ps aux |less (команда ps aux объединена в конвейер с утилитой less для возможности прокрутки вывода, если он не умещается на экран).

```
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.2  4.6  2584  1620 ?        Ss   07:56   0:00 init [3]
root         2  0.0  0.0      0      0 ?        S    07:56   0:00 [kthreadd]
root         3  0.0  0.0      0      0 ?        I<   07:56   0:00 [rcu_gp]
root         4  0.0  0.0      0      0 ?        I<   07:56   0:00 [rcu_par_gp]
root         5  0.0  0.0      0      0 ?        I    07:56   0:00 [kworker/0:0-pm
]
root         6  0.0  0.0      0      0 ?        I<   07:56   0:00 [kworker/0:0H-e
vents_highpril
root         7  0.0  0.0      0      0 ?        I    07:56   0:00 [kworker/u2:0-e
vents_unbound]
root         8  0.0  0.0      0      0 ?        I<   07:56   0:00 [mm_percpu_wq]
root         9  0.0  0.0      0      0 ?        S    07:56   0:00 [ksoftirqd/0]
root        10  0.0  0.0      0      0 ?        I    07:56   0:00 [rcu_sched]
root        11  0.0  0.0      0      0 ?        S    07:56   0:00 [migration/0]
root        12  0.0  0.0      0      0 ?        S    07:56   0:00 [cpuhp/0]
root        13  0.0  0.0      0      0 ?        S    07:56   0:00 [kdevtmpfs]
root        14  0.0  0.0      0      0 ?        I<   07:56   0:00 [netns]
root        15  0.0  0.0      0      0 ?        I<   07:56   0:00 [inet_frag_wq]
root        16  0.0  0.0      0      0 ?        S    07:56   0:00 [kauditd]
root        17  0.0  0.0      0      0 ?        S    07:56   0:00 [oom_reaper]
root        18  0.0  0.0      0      0 ?        I<   07:56   0:00 [writeback]
root        19  0.0  0.0      0      0 ?        S    07:56   0:00 [kcompactd0]
root        39  0.0  0.0      0      0 ?        I<   07:56   0:00 [kblockd]
:
```

Табл. 1 - колонки таблицы вывода ps aux

Колонка	Объяснение
USER	Владелец процесса
PID	Process ID
%CPU	Доля времени ЦП, выделенная процессу (в процентах)
%MEM	Сколько реальной памяти занято процессом (в процентах)
VSZ	Виртуальный размер процесса
RSS	Кол-во страниц памяти (размер резидентного набора)
TTY	В каком TTY запущено (идентификатор управляющего терминала) - выделено красным

```
root      381  0.0  4.2  5620  1492 tty2      Ss+   07:56   0:00 /sbin/agetty tt
y2 9600
root      419  0.0  5.7  3428  2024 tty1      R+    08:00   0:00 ps aux
root      420  0.0  2.5  3164   884 tty1      R+    08:00   0:00 less
```

STAT	<p>Текущий статус процесса:</p> <ol style="list-style-type: none"> 1. R - выполняется; 2. D - ожидает записи на диск; 3. S - неактивен; 4. T - приостановлен; 5. Z - зомби. <p>Дополнительные:</p> <ol style="list-style-type: none"> 1. W - выгружен на диск; 2. < - имеет повышенный приоритет; 3. N - имеет пониженный приоритет; 4. L - некоторые страницы заблокированы в ядре; 5. s - процесс является лидером сеанса.
TIME	Кол-во времени ЦП, затраченное на выполнение процесса
COMMAND	Имя аргумента и команды. Программы могут модифицировать эту информацию, поэтому в этой колонке не для всех процессов может быть получена достоверная информация. Команды в скобках (в колонке <i>COMMAND</i>) являются потоками ядра, запланированными в качестве процессов.

Пользователям и администраторам часто приходится искать нужный процесс среди большого списка процессов из вывода `ps`. Тогда объедините эту программу `sgrep` в пайп:

```
ps aux |grep "vim"
```

Замените `vim` на имя нужного вам процесса. Но обратите внимание, что вывод будет включать в себя и программу `grep`, так как этот процесс был активен во время выполнения `ps` (мы же объединили эти две программы в пайп). Чтобы этого избежать, используйте `grep -v`:

```
ps aux |grep -v grep |grep "vim"
```

Так же (для определения PID процесса) можно использовать `pgrep`:

```
pgrep vim
```

```
root:~# ps aux |grep vim
root      441  0.7 16.5 10332 5824 tty1    T   08:38   0:00 vim
root      446  0.0  1.8  2780  652 tty1    S+  08:38   0:00 grep vim
root:~# pgrep vim
441
root:~#
```

Красным выделены значения PID нужного процесса `vim`

Использование `top` для интерактивного мониторинга процессов

```
top - 08:51:52 up 55 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 55 total, 1 running, 53 sleeping, 1 stopped, 0 zombie
%Cpu0 :  0.7/1.3      2[          ]
GiB Mem : 65.3/0.0    [          ]
GiB Swap:  0.0/0.0    [          ]

  PID USER      PR  NI    VIRT    RES    %CPU    %MEM     TIME+ S COMMAND
  1 root        20   0    2.4m    0.5m    0.0    1.6   0:00.75 S init [3]
162 root        20   0    4.1m    1.1m    0.0    3.2   0:00.10 S ` - /sbin/udev +
353 root        20   0    2.5m    0.4m    0.0    1.1   0:00.02 S ` - /sbin/syslog+
373 root        20   0    5.3m    3.4m    0.0    9.9   0:00.06 S ` - /sbin/klogd
380 root        20   0    4.6m    2.7m    0.0    7.8   0:00.55 S ` - -bash
441 root        20   0   10.1m    5.4m    0.0   15.8   0:00.07 T ` - vim
449 root        20   0    3.9m    2.5m    2.0    7.2   0:00.21 R ` - top
381 root        20   0    5.5m    0.4m    0.0    1.2   0:00.00 S ` - /sbin/agetty+
  2 root        20   0    0.0m    0.0m    0.0    0.0   0:00.00 S [kthreadd]
  3 root         0 -20    0.0m    0.0m    0.0    0.0   0:00.00 I ` - [rcu_gp]
  4 root         0 -20    0.0m    0.0m    0.0    0.0   0:00.00 I ` - [rcu_par_gp]
  6 root         0 -20    0.0m    0.0m    0.0    0.0   0:00.00 I ` - [kworker/0:0+
  8 root         0 -20    0.0m    0.0m    0.0    0.0   0:00.00 I ` - [mm_percpu_w+
  9 root        20   0    0.0m    0.0m    0.0    0.0   0:00.00 S ` - [ksoftirqd/0]
10 root        20   0    0.0m    0.0m    0.0    0.0   0:00.06 I ` - [rcu_sched]
11 root        rt   0    0.0m    0.0m    0.0    0.0   0:00.00 S ` - [migration/0]
12 root        20   0    0.0m    0.0m    0.0    0.0   0:00.00 S ` - [cpuhp/0]
13 root        20   0    0.0m    0.0m    0.0    0.0   0:00.00 S ` - [kdevtmpfs]
```

На скриншоте выше виден вывод утилиты top. По умолчанию, информация из вывода обновляется каждые 10 секунд. На первых строках вывода в сжатом виде представлена информация о нагрузке системы, кол-ве процессов, использовании процессора и пр.

Так же есть более удобная утилита htop, так же предназначенная для работы с процессами. Установите его из портов (console/htop

[Назад](#)

[Домой](#)

[Далее](#)

Права доступа

Управление пользователями

Управление пользователями

Даже если за компьютером работает только один человек, необходимо понимать принципы управления пользователями в Linux. Для повседневной работы за ПК с Linux вполне хватит непривилегированного пользователя, а для настройки и восстановления системы понадобится пользователь с другим уровнем доступа. Так же очень важен аспект контроля других пользователей, а они идентифицируются в системе по своим именам.

В Linux существует три типа пользователей:

- `root` - суперпользователь. Имеет право на выполнение всех операций без исключения. В Calmira Linux включен по умолчанию.
- Системные пользователи - создаются системой автоматически. Предназначены для некоторых процессов для управления привилегиями и правами доступа к файлам.
- Обычные пользователи - создаются системным администратором. Предназначены для работы обычных людей.

Каждый пользователь имеет своё имя в системе. Имена нескольких пользователей не могут повторяться (т.е. не может быть двух юзеров с одинаковым именем).

Каждый пользователь имеет числовой идентификатор UID - User Identifier (об этом можно просить и в предыдущей инструкции). `root` имеет идентификатор 0. Системные пользователи - от 1 до 100. Обычные пользователи - от 100.

Пользователи могут объединяться в группы. Каждый пользователь *обязательно* входит в ту, или иную группу. Группы так же имеют числовой идентификатор (GID - Group Identifier).

Введение

Система регистрирует следующую информацию о пользователе:

1. Имя пользователя (username). Как уже упоминалось выше, имя должно быть уникальным в системе. В именах могут быть использованы **только** английские буквы, а так же цифры и символы `_` (нижнее подчёркивание) и `.` (точка).
2. Полное имя. Помимо обычного имени пользователя, заносится имя и "реального". Например, иногда полное имя содержит в себе имя и фамилию реального человека, использующего ПК.
3. Идентификационный номер пользователя (UID). По этому номеру система отслеживает пользователя (а не по его имени).
4. Идентификационный номер группы (GID). Этот номер означает группу, в которую входит пользователь. Группы позволяют регулировать доступ юзеров к различным ресурсам. Принадлежность к определённой группе устанавливает системный администратор.
5. Пароль (password). Каждый пользователь для доступа к своему аккаунту должен иметь пароль (он зашифрован). Без пароля не будет доступа к аккаунту.
6. Домашний каталог. Это название каталога, в который попадает пользователь после того, как зарегистрируется в системе (с помощью программы `login`), и где хранятся его собственные файлы (документы, музыка и прочее). Обычно такой каталог находится в `/home`, однако такое поведение можно изменить (но не рекомендуется, чтобы не нарушать стандарт FHS, которому следует Calmira Linux).
7. Начальная оболочка (`login shell`). Командная оболочка, которая запускается после входа в систему. По умолчанию это `/bin/bash`, однако при установке другой (например, `zsh`, `tcsh` и пр.), стандартный `bash` можно заменить на другую.

Вся информация находится в `/etc/passwd`. Строки имеют следующий формат:

```
имя пользователя:зашифрованный пароль:UID:GID:полное имя:домашний каталог:оболочка
```

Например:

```
administrator:x:1000:1000:Михаил Краснов:/home/administrator:/bin/bash
```

Для обеспечения безопасности пароли не хранятся в "читабельном" виде, а зашифрованы. Для того, чтобы расшифровать пароль, нужно его знать. Такая форма является достаточно надёжной.

Так же пароли могут храниться не в `/etc/passwd`, а в `/etc/shadow` (такие называются *теньевыми паролями*). Права на чтение у него ограничены, когда как `/etc/passwd` может читать почти каждый. Вынос паролей в `/etc/shadow` является более безопасной. Теневые пароли так же обеспечивают дополнительные функции, например, ограниченный срок действия пароля.

Дополнительные материалы

Здесь представлена дополнительная информация: тексты лицензий, манифестов и пр.

Содержание

5.1 [Манифест разработчиков Calmira](#)

5.2 [GNU GPLv3](#)

5.3 [Лицензия свободной документации GNU](#)

Манифест разработчиков Calmira

Автор: Михаил Linuxoid85 Краснов <linuxoid85@gmail.com>

В данном манифесте пойдёт речь о принципах создания системы Calmira GNU/Linux. Ни для кого не секрет, что современные дистрибутивы Linux стали слишком жирными, сложными и медленными, по сравнению с тем, что было раньше. Раньше -- это конец девяностых - начало двухтысячных. Конечно, за эти 20-30 лет многое изменилось. Системы (дистрибутивы Linux) стали более совершенными: более простыми, надёжными и красивыми. Однако не надо забывать истинное предназначение ОС -- быть миниатюрной прослойкой между ПК и его пользователем. Ключевое слово здесь -- миниатюрной. Дистрибутив должен потреблять минимум ресурсов ПК, чтобы максимум было доступно для пользовательского ПО. Но дистрибутив, не смотря на свою миниатюрность, должен отвечать базовым требованиям:

1. быть надёжным и стабильным, а так же быстрым;
2. выполнять поставленные пользователем задачи.

Из этого следует, что не надо делать максимально маленький дистрибутив (как, например, Floppinux, который помещается всего лишь на одну дискету). Система должна быть пригодной для использования. Например, Calmira GNU/Linux потребляет 25 Мб ОЗУ и занимает на жёстком диске 1.2 Гб. Конечно, потребление оперативной памяти во многом зависит от её количества. Например, на ПК с двухгигабайтной плашкой ОЗУ система потребляет 25 Мб, а на компьютере с восьмигигабайтной - уже 36. Поэтому при формировании системных требований дистрибутива используются усреднённые значения.

Этот документ обязаны прочитать все те, кто хочет войти в команду разработчиков дистрибутива, либо же хочет сделать его форк.

Особое внимание хочу уделить на третий принцип, в котором описано самое главное зло - проприетарное и несвободное программное обеспечение.

ПРИНЦИП 1. Делайте систему максимально компактной, но чтобы она оставалась пригодной к использованию.

Несмотря на то, что в персональных компьютерах всё лучше и лучше их железо -- не надо жиреть. Система должна максимально меньше расходовать ресурсы ПК, но оставаться юзабельной (пригодной к использованию). Ведь невозможно предугадать, на каком железе пользователь запустит дистрибутив. Если на мощном железе - хорошо, у него останется больше ресурсов для запуска тяжеловесного ПО, такого как игры, IDE, видео- и фоторедакторы, etc. Если же железо пользователя слабое, то у него останутся ресурсы для запуска того ПО, которое ему необходимо. А легковесные дистрибутивы несколько урезаны, да и не очень удобны.

Хотя я (создатель дистрибутива) не всегда следую первому принципу создания ОС. В Calmira на данный момент ещё есть какое-то число ненужного мусора, которое можно (даже нужно) окончательно вычистить.

Только представьте -- у вас есть старенький компьютер, который ещё год назад выполнял возложенные на него задачи просто отлично. Но вдруг, он те же самые задачи начал выполнять медленно, а некоторые вообще разучился. В чём беда? Только в том, что компьютеры с каждым днём становятся всё более мощными, а программисты не умеют экономить, и делают свои программы неоптимизированными, тяжёлыми и жирными. В этом и беда -- программисты избалованы мощными ПК.

А что же делать тем, у кого не самый мощный компьютер, который выполнял (и мог бы выполнять!!!) возложенные на него задачи просто отлично, он надёжный, а выбрасывать его не только жалко, но и расточительно? Вопрос остаётся в силе.

Теперь про использование опционального софта, который рекомендуется устанавливать после установки дистрибутива. Он должен соответствовать первому принципу.

Включайте в систему только самое необходимое, без чьего пользователь обойтись не может. Система не должна представлять из себя швейцарский нож. В конце концов -- Calmira GNU/Linux старается следовать принципу KISS!

Базовая система не должна потреблять более 40 Мб оперативной памяти и занимать более 1.5 Гб на жёстком диске. Так как всё начинается с базы. Если бы не было базовой ОС, не было и программ, установленных на ней. И потребление ресурсов ПК начинается именно с базовой системы. Поэтому чем меньше ресурсов ПК она потребляет -- тем лучше. На то она и "базовая", ведь предоставляет минимальный инструментарий, необходимый для работы всего остального ПО.

Дистрибутив Calmira GNU/Linux с установленным сервером Xorg и каким-либо оконным менеджером должна потреблять не более 90 Мб ОЗУ и занимать на жёстком диске не более 1.2-2 Гб. Чем меньше -- тем лучше.

В идеале, система Calmira 2021.1 с оконным менеджером Fluxbox потребляла 55-60 Мб ОЗУ. От этого и нужно отталкиваться.

В других случаях, Calmira GNU/Linux не должна потреблять более 512 Мб оперативной памяти. "Другие случаи" включают в себя:

1. установка рабочего окружения;
2. сборка специальной редакции дистрибутива с определённым набором программного обеспечения.

ПРИНЦИП 2. Делайте систему простой и понятной как для разработчика, так и для пользователя. Однако, ориентируйте систему на хакера, либо продвинутого пользователя.

Я (разработчик дистрибутива Calmira GNU/Linux) был вдохновлён принципом KISS (Keep it simple, stupid!). В переводе с английского это значит: "Не усложняй, тупица!". То есть, дистрибутив должен быть простым. Всё должно быть логичным. Да, на первый взгляд, Calmira -- достаточно сложный дистрибутив, но это только на первый взгляд. Немного разобравшись, понимаешь, что всё достаточно просто. На первый взгляд, сложность заключается в том, что только конечный пользователь дистрибутива решает то, каким он должен быть. Только пользователь решает, что в Calmira устанавливать, а что нет. Кстати, именно из-за этого в её пакетном менеджере `srkg` нет обработки зависимостей.

Однако, если производится замена привычных инструментов на более новые и функциональные аналоги, либо же полнофункциональные замены, то следует убедиться в том, что они максимально совместимы с предыдущими инструментами и не вызовут проблем с использованием у пользователей, либо же в определённых скриптах и программах. Должен быть обеспечен либо плавный переход на другой инструмент, либо же максимально незаметный. Если одно из этих высказываний реализовать невозможно, то переход не может быть выполнен. Более того, дистрибутив с этими инструментами не будет пропущен, так как он нарушает второй принцип.

Если обычному пользователю, откровенно говоря, наплевать на инструменты работы с системой, то продвинутым юзерам и хакерам -- нет. Они работают только с теми инструментами, к которым привыкли, эти инструменты они хорошо знают. А переучивание вызывает у них раздражение; помимо этого, переучивание может занять довольно большое кол-во времени, а если изучать новый инструмент в процессе работы, можно совершить немало ошибок как в изучении, так и, непосредственно, в работе.

Защитите дистрибутив от резкого, молниеносного и быстрого перехода с одних инструментов на другие!

Документируйте своё ПО, а так же предлагайте установить дополнительную документацию, если вы делаете порт с пакетом. Документация позволит новым пользователям узнать больше всего информации о вашем ПО.

В документацию входят файлы `/usr/share/man` и `/usr/doc`.

ПРИНЦИП 3. Обеспечьте свободу пользователю.

Только пользователь должен контролировать поведение ОС, а она, в свою очередь, должна выполнять то, что решил пользователь. В рамках разумного. К большому сожалению, большинство современных дистрибутивов не могут похвастаться свободой действий. Например, Ubuntu, Fedora, etc.

Помимо свободы действий, есть ещё свобода в ПО. Не включайте в дистрибутив несвободное и/или проприетарное ПО, ибо это зло. А так же в Calmira не должно быть никакого ПО с т.н. "тайм-бомбами". Это ПО оказывается неработоспособным по прошествии некоторого времени. Чтобы вернуть работоспособность, нужно либо заплатить автору, либо выполнить какое-либо действие.

Распространение несвободного или проприетарного ПО есть нарушение программной этики, а так же -- нарушение свободы пользователя. Вспомните случай со Столлманом в Карнеги-Меллон. А ведь ему понадобились исходные коды программы для принтера -- ведь в этой программе была серьёзная недоработка. И Ричарду нужно было доделать программу для себя и для МТИ.

Проприетарный софт может следить за пользователем, может удалять какие-либо данные, либо же производить какие-либо несанкционированные действия в системе. Из этого следует, что под видом какой-либо программы ("всего-навсего" проприетарной) вам могут подсунуть вирус. Остерегайтесь этого.

Несвободный софт создан лишь для насыщения разработчика деньгами, не более. Несвободный софт прямо-таки требует у пользователя: "заплатите разработчикам денег, а не то мы сделаем всё возможное, чтобы наша программа не работала". И ведь вправду -- разработчики несвободного софта делают всё, чтобы пользователь заплатил денег. Они буквально требуют их! Например, пока пользователь не заплатил разработчикам, урезают функционал, либо же вообще делают программу неработоспособной, встраивая в неё т.н. "тайм-бомбы", о чём говорилось ранее.

Сравните определённую программу с рецептом какого-либо блюда. Ведь эти две вещи предоставляют пошаговые инструкции о том, как достичь наилучшего результата или желаемой цели. И то, и другое можно без проблем изменить так, как нужно конечному пользователю. Вы же не обязаны точно следовать кулинарному рецепту! Вы вправе отбросить какие-то ингредиенты, либо же добавить чего-либо. Только потому что вам это надо. Вам либо это нравится, либо необходимо. Так же и с программой для компьютера. Но самое важное -- то, что и программу, и рецепт можно очень легко распространять. В случае рецепта, вам будет достаточно ручки и листа бумаги, чтобы начертать на нём этот рецепт и дать своему гостю. А с программами ещё проще -- нужно лишь пару кликов мыши или ввод одной-двух команд на клавиатуре.

В этих двух случаях пользу получают все -- и тот, кто поделился, и тот, кто принял. Первый укрепляет вашу дружбу, либо взаимоотношения и повышает шансы того, что с ним поделится так же. А у второго очень большое преимущество -- он может подстроить программу для своих нужд. Либо же кулинарный рецепт. В котором, к примеру, содержатся ингредиенты, которые нельзя употреблять человеку, который принял этот рецепт. И он без каких либо преград меняет тот "запрещённый" ингредиент на то, что ему разрешено. И никаких проблем от этого нет! Либо же, этому человеку не нравится вкус блюда. Тогда он так же без каких-либо проблем и ограничений меняет, убирает или добавляет в рецепт какие-то продукты, чтобы блюдо получилось вкуснее. И ведь тоже самое с программой!

Но есть и обратная ситуация. Когда рецепт закрыт. И вы не можете узнать, что в него входит. Более того, вы не можете делиться им с другими людьми! Однако, если вы совершите это, вы станете самым настоящим вором, пиратом и пипидастром. Это будет очень неприятным.

Вот поэтому стоит остерегаться несвободного и/или проприетарного программного обеспечения. И никто из команды разработчиков Calmira GNU/Linux (пока там только я) не допустит попадания проприетарного и несвободного ПО в состав дистрибутива.

ПРИНЦИП 4. Не использовать наработки Calmira GNU/Linux во вред.

Наработки дистрибутива Calmira GNU/Linux распространяются под лицензией GNU GPLv3. И, несмотря на то, что в GNU GPL действительно используется понятие "свобода", всё-таки, ряд ограничений присутствует и там. Для того, чтобы понимать, о чём речь, прочитайте эту лицензию самостоятельно.

Так значит ли это, что GNU GPL и этот список принципов можно считать полностью свободными? Либо же правильно называть лицензию и ПО, которое распространяется под этой лицензией не свободным, а условно-свободным? Нет. Давайте отталкиваться от значения слова "свобода". Что это? С одной стороны, свобода -- это неограниченность в действиях. Человек или программа может сделать всё, что хочет. А давайте подумаем. Если бы люди были полностью свободными, то каким бы был наш мир? Люди могли бы свободно убивать друг друга, грабить, причинять увечья. Это ли свобода? Это ли то, что мы обычно подразумеваем под словом "свобода"? Наверно, нет. Всё-таки, под "свободой" мы имеем в виду все действия в рамках разумного. Именно это и есть свобода. Человек может делать всё, что считает нужным, но только если это не противоречит или не нарушает писанные и неписанные законы и принципы.

Тоже самое и с ПО. И Calmira может считаться свободным дистрибутивом GNU/Linux. Но в рамках дозволенного. И команда разработчиков не позволит использовать наработки дистрибутива во вред кому-то.

Дистрибутив Calmira GNU/Linux разрабатывается для решения следующих задач:

1. предоставление пользователям чистой, простой и быстрой системы, из которой они, впоследствии, соберут то, что хотят;
2. предоставление свободного и открытого дистрибутива.

Ну а так же, этот дистрибутив создаётся "для души". И создавался с благими намерениями!

ПРИНЦИП 5. Делись наработками с сообществом (совет).

Если вы хотите сделать дистрибутив Calmira GNU/Linux лучше, то обязательно делитесь наработками с его сообществом и командой разработчиков! Это традиция в мире UNIX. Нарушать её -- значит тормозить развитие, а это не есть хорошо.

Раньше было принято писать программы в одиночку, так как считалось, что это намного быстрее, чем в команде. Но на самом деле, это не так. Давайте рассмотрим простой пример. Кто быстрее сварит 10 литров супа: один повар или несколько? За одинаковое время. Конечно же, несколько. Так же и здесь. И дело не только в скорости, дело в обмене опытом. Когда между собой взаимодействуют несколько программистов, они учатся друг у друга, перенимают какие-то навыки, приёмы. Это полезно как для самих программистов, так и для их программы, ведь код становится более аккуратным, грамотным, продуманным и надёжным. А писать красивый и понятный, а самое главное - надёжный код сейчас, как я понял, очень немодно, вот и появляются на свет всякие Ubuntu Linux...

Помимо этого, активно принимайте участие в жизни сообщества и системы! Помогайте людям, отвечайте на вопросы, помогайте разработчикам. Опять же

-- это внесёт огромный вклад в развитие дистрибутива.

[Назад](#)

Дополнительные
материалы

[Домой](#)

[Далее](#)

Лицензия GNU GPLv3

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such

permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its

Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called

the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to

the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see [<https://www.gnu.org/licenses/>](https://www.gnu.org/licenses/).

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read [<https://www.gnu.org/licenses/why-not-lgpl.html>](https://www.gnu.org/licenses/why-not-lgpl.html).

[Назад](#)

Манифест разработчиков
Calmira

[Домой](#)

[Далее](#)

Лицензия свободной
документации GNU

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this

License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.