

Предисловие

О руководстве

Люди, которые используют Calmira GNU/Linux-libre с недавнего времени, найдут, что в этом руководстве описаны далеко не все сведения о дистрибутиве, так как на данный момент руководство всё ещё находится в разработке. Однако, если вы хотите ускорить работу над Calmira GNU/Linux-libre Handbook, можете либо написать какую-либо из глав, либо исправить уже существующие страницы. Для этого посетите наш [GitLab](#) для получения соответствующих сведений. Очень полезным будет раздел "Issues" оттуда - там перечислено всё, что нуждается в исправлении, дополнении и написании.

Данное руководство кратко освещает идеи и традиции, на которых базируется GNU/Linux(-libre). Работа с этим руководством требует нечего большего, чем просто исследовать - важна возможность принимать к сведению и запоминать новые вещи.

Другая часть руководства является инструкцией по установке этого дистрибутива GNU/Linux-libre на свой компьютер или виртуальную машину.

Целевая аудитория

Так как дистрибутив Calmira GNU/Linux-libre (далее - *дистрибутив*) предназначен для опытных пользователей, либо для тех, кто хочет изучить строение GNU/Linux, то и руководство ориентируется именно на таких людей. Здесь не будет очень обширных сведений о банальных вещах, вроде записи образа на флешку, загрузки ПК с неё и пр. Предполагается, что читатель уже знаком с базовыми сведениями об управлении ПК, поэтому в данном руководстве они отсутствуют.

Если вам не понятны какие-либо сведения, то на это есть две причины:

1. Вам не хватает базовых знаний. Тогда добро пожаловать в [Google](#).
2. В руководстве эти сведения описаны некорректно. Тогда оставьте у нас на GitLab [ссылка выше] соответствующее issue с полным описанием проблемы.

Условные обозначения, принятые в руководстве

Курсивный шрифт

Используется для выделенного текста и первого применения технических терминов.

Полужирный шрифт

Используется для обозначения приложений, команд и параметров.

`Моноширинный шрифт`

Используется для обозначения команд, имён пользователей, групп, названий устройств, файлов, переменных и фрагментов кода.

Цитата

Просто какая-то цитата :)

Пользовательский ввод

Клавиши указываются **полужирным текстом** для выделения среди остального текста. Комбинации клавиш разделяются символом `+`:

`Ctrl` + `Alt` + `Delete`

Команды, которые начинаются со знака `#` означают, что вы должны запустить эту команду от имени пользователя `root` в дистрибутиве. Например:

```
# cport --install editors/gvim
```

Команды, которые начинаются со знака `%` означают, что вы должны запустить эту команду от имени обычного пользователя:

```
% cport list -a
```

Символы `#` и `%` вводить не нужно!

Иногда в командах встречаются комментарии. Они также начинаются с символа `#`. Для того, чтобы определить комментарий от команды, используйте следующие правила:

1. В руководстве комментарии выделяются серым цветом. А команды - нет.
2. Комментарии являются простыми подсказками пользователю и не несут никакой ценности дистрибутиву. Комментарии просто поясняют какие-то вещи, не более.

HELP WANTED!

- ☐ Мы ищем переводчиков на английский, немецкий, украинский и белорусский языки. Если вы хотите заниматься переводом этого руководства, напишите автору в [Telegram](#)
- ☐ Заполнение страниц документации:
 - Глава 3 - особенно важно

Часть 1. Введение

Эта часть руководства предназначена для пользователей и администраторов, которые являются новичками в данном дистрибутиве. Сведения отсюда:

- Введут вас в Calmira GNU/Linux-libre.
- Проведут вас по процессу установки дистрибутива.
- Обучат вас некоторым основам Linux.
- Покажут вам как устанавливать стороннее программное обеспечение дистрибутива.

Предназначение руководства

Мы написали это руководство для нескольких категорий пользователей. Во-первых, это те, кто изучает строение GNU/Linux. Таким людям нужен простой и быстрый дистрибутив этой ОС, который распространяется в исходных кодах, а также дистрибутив достаточно легко модифицируемый.

Во-вторых, данное руководство предназначено для разработчиков свободного программного обеспечения для GNU/Linux. Так как наша система достаточно минималистична, достаточно неплохим шагом будет тестирование ПО на ней.

Это общее руководство. Здесь рассматриваются базовые возможности и объясняются, как они работают воедино. В случае выбора между несколькими технологиями предлагается сравнение каждой из них.

Здесь собраны те сведения, которые необходимо знать каждому пользователю для выполнения типичных задач.

Основные обязанности пользователя операционной системы

В приведённых ниже разделах приведены основные задачи, которые должны выполнять пользователи операционных систем. Какие-то задачи достаточно специфичны для какой-либо IT-специальности, но мы их поместили сюда потому, что они не будут лишними и для обычных пользователей, а в некоторых случаях даже будут желательными.

Управление резервными копиями

Резервное копирование данных является достаточно важной задачей для любого пользователя. У каждого из нас есть жёсткий диск или SSD, на котором мы храним как операционную систему, так и свои документы и файлы (слава б-гам, что эра дискет, кассет и картриджей канула в лету). Кто знает, когда эти накопители уйдут в прошлое? Конечно, у нас есть такая вещь, как SMART, но часто ли мы туда заглядываем? А кто знает, когда во время работы выключат свет, и ваш компьютер выключится, в результате чего часть ваших файлов окажется повреждённой?

Резервные копии должны выполняться по определённому расписанию (каждый месяц, неделю, день, час, секунду, etc.), а их восстановление должно периодически проверяться, т.е. необходимо проверять корректность созданных копий.

Установка и обновление программного обеспечения

Дистрибутив Calmira GNU/Linux-libre содержит систему портов и МСП sport для управления программным обеспечением. Эти две вещи в значительной степени упрощают и автоматизируют сборку программного обеспечения из исходного кода, однако работа пользователя не минимализируется до нуля. Он должен управлять зависимостями ПО, проверять и тестировать обновлённое ПО и выполнять прочие действия, о которых пойдёт речь в главах далее.

Исправление проблем

Сбои в работе программного обеспечения и операционной системы происходят достаточно часто, поэтому пользователь должен, по крайней мере, иметь представление о том, как исправить наиболее часто встречающиеся ошибки. Очень кстати будет и обратная связь с разработчиками ПО: пользователь может отправлять им информацию о найденных ошибках или баг-репорты. Достаточно неплохой вещью же со стороны разработчиков является сбор телеметрии¹ - обезличенных данных о работе программы, её конфигурации и конфигурации ОС. На самом деле, такие данные очень хорошо помогают понять разработчикам причины ошибки и условия, в которых эта ошибка происходит. Но не следует опускаться до уровня той же Microsoft, которая собирает даже те данные, которые, кажется, и не должна. В эти данные входит история посещений веб-страниц, наиболее используемые приложения и файлы, личные данные, такие, как e-mail адреса, номера телефонов и прочее. Например, правительство Германии не так давно вообще запретило использовать продукцию Microsoft Office в учебных организациях из-за того, что их вредоносный офисный пакет собирал кучу самых разных данных даже из содержимого самих офисных документов!

¹ конечно, мы не выступаем за сбор абсолютно всех сведений, как это делают Microsoft, Apple, Google и прочие подобные корпорации. Тогда это называлось бы слежкой и шпионажем. Во-первых, сбор указанных мною данных должен быть обезличенным и добровольным. Во-вторых, пользователь, если он всё-таки хочет отправлять диагностические данные разработчикам, должен выбирать те сведения, которые следует передавать. Например, мне очень нравится реализация этого в рабочем окружении KDE Plasma: там есть несколько уровней от низшего к высшему, в зависимости от уровня меняется и количество отправляемых диагностических данных. По умолчанию сбор и передача телеметрии там отключена вообще.

Оптимизация операционной системы для выполнения каких-то определённых задач

GNU/Linux, как и многие другие операционные системы, являются ОС общего назначения, а значит, они предназначены по умолчанию для выполнения самых разных задач. Пользователи могут адаптировать ОС

для выполнения каких-то конкретных вещей, тогда ОС либо научится их выполнять уже, либо просто будет выполнять их куда лучше.

Введение

```
root:~# about_sys
#####*,*.
=#DD-D#####DD#
#----#      #-#
#---#      ##
#---#
#QQQ#
#=Q=#      ##
#==#      ##
##==##==#

----- 0 системе -----
Имя:      Calmira
Версия:    LX4 1.1
Кодовое имя: Aquarius
Пользователь: root
Положение: /root

root:~#
```

Краткий обзор

Прежде всего, добро пожаловать в Calmira GNU/Linux-libre! Этот дистрибутив всегда даёт свободу выбора. При его установке вы не раз убедитесь в этом: пользователи самостоятельно могут выбирать программное обеспечение, которое будет устанавливаться на компьютер, способ установки и прочие подобные вещи.

Этот дистрибутив основан на свободном программном обеспечении и не скрывает от пользователя, как он устроен. `scport`, система управления ПО дистрибутива, написана на языке Python, что позволяет вам с лёгкостью просматривать и изменять его исходный код. Система сборки CABS написана на Rust и распространяется под лицензией GNU GPLv3. `scport` и CABS используют исходный код, а настройка дистрибутива производится с помощью текстовых конфигурационных файлов.

Области применения Calmira GNU/Linux-libre

Этот дистрибутив изначально несёт экспериментальную цель - тестирование новейших компонентов либо переход на использование чего-то экзотического. Кроме того, основной областью применения этой системы является *изучение* особенностей GNU/Linux, его строения и принципа работы. Свободная природа дистрибутива позволяет удалённым группам сотрудничать, обмениваться идеями и совместными разработками, не беспокоясь о наличии специальных лицензий или ограничений. Например, Calmira GNU/Linux-libre разрабатывает небольшая команда из 4 человек, живущих в России, Украине и Норвегии.

Однако никто вам не запрещает использовать эту систему и для других целей. К примеру, в качестве системы для слабого оборудования. Так как дистрибутив не требовательный к ресурсам ПК, его можно запустить на достаточно слабом железе. Главное, чтобы процессор был архитектуры x86_64 - поддержки 32-битных ЦП в Calmira GNU/Linux-libre никогда не было.

Варианты установки

Дистрибутив можно установить разными способами. Вы можете загрузить установочный iso-образ системы. Тогда установщик оттуда скопирует вам уже собранную систему на компьютер. Либо вы можете пойти более сложным путём - самостоятельно скомпилировать Calmira GNU/Linux-libre из исходного кода посредством системы сборки CABS.

Установка дистрибутива версии $\geq 2.X$

Дистрибутив версии 2.0 и выше поставляется с простой в использовании псевдографической программой установки. В релизах, предшествующих 2.0, такой программы нет, и вся установка производилась вручную. Кроме того, в этих релизах не формировались загрузочные iso-образы, а генерировались squashfs-снимки. Как вы уже поняли, процесс установки старых версий Calmira GNU/Linux-libre достаточно сложен, поэтому мы специально закончили их техподдержку после выхода версии 2.0, процесс установки которой сильно упрощён.

Прочтя эту страницу документации, вы узнаете:

- Как дистрибутив видит и делит на разделы жёсткие диски.
- Как запустить `calminstall`.
- Как работать с `calminstall`.

Системные требования

Пункт	Минимальные требования	Рекомендуемые требования
ЦП	x86_64 > 900 МГц	x86_64 1 ГГц
ОЗУ	64 Мб	128 Мб
Жёсткий диск	HDD 1.5 Гб	SSD 5 Гб
Сеть	--	Наличие сетевого адаптера ¹

¹ в дистрибутиве используется ядро Linux-libre, которое было очищено от несвободных драйверов, прошивок и блобов. Это значит, что некоторое

оборудование (как правило, это сетевые, звуковые и видеокарты) в ОС не поддерживаются, т.е. либо не будут определяться в системе вообще, либо просто не будут корректно работать. В том случае, если что-то из вашего оборудования не поддерживается, попробуйте переключиться на обычное ядро Linux с проприетарными несвободными компонентами [оно входит в минимальную поставку дистрибутива; переключиться можно, выбрав в меню загрузчика ОС соответствующий пункт]. Если не помогло и это, то отправьте нам issue об этом на нашем [GitLab](#). Опишите поведение дистрибутива, версии программного обеспечения (ядро Linux-libre и Linux), тип оборудования, с которым возникли проблемы, его модель и прочие подобные сведения.

Разделы диска для дистрибутива

Диск ПК делится на отдельные части, которые называются *разделами*. Каждая операционная система определяет разделы своим способом. К примеру, MS-DOS и её потомки, такие, как Windows, присваивает каждому главному и логическому разделу *букву диска*, например, **A:**, **B:**, **C:**, **D:** и т.д. А, например, операционные системы семейства GNU/Linux, именуют диски и разделы так: `/dev/XdYN`, где **X** и **Y** - определённая буква диска, а **N** - номер раздела на этом диске. **X** определяет тип диска (**h** - диск IDE, **s** - диск SATA), **y** - сам диск (**a** - главный диск, **b**, **c**, **d** ... - съёмные диски).

К примеру, **sda3** - это третий раздел (**3**) главного жёсткого диска (**a**) типа SATA (**s**). Таблица ниже поможет определить тип диска

Тип устройства	Дескриптор устройства	Примечания
SATA, SCSI, USB Flash	<code>sda</code> , <code>sdb</code> , <code>sdc</code> ...	Стал доступен где-то в 2007 году и встречается до сих пор. Наверно, на данный момент самый используемый в GNU/Linux тип.
NVMe	<code>nvme0n1</code>	Передовая технология накопителей. Устройства NVMe подключаются к шине PCI Express и обладают наиболее высокой скоростью передачи данных. Первый раздел на первом NVMe

Тип устройства	Дескриптор устройства	Примечания
		накопителе обозначается как <code>nvme0n1p1</code> . Поддержка таких устройств в ядре Calmira GNU/Linux-libre есть, но не протестирована. Если у вас есть возможность протестировать систему на таком накопителе, оставьте issue на GitLab CalmiraLinux .
MMC, eMMC, SD	<code>mmcblk0</code>	SD-карты, MMC и embedded MMC устройства. Используются для хранения данных. ВНИМАНИЕ: не все устройства могут иметь возможность загрузки с этих накопителей. ВНИМАНИЕ: цикл чтения/записи у таких устройств ограничен куда более, чем у накопителей выше. Поэтому настоятельно не рекомендуется использовать их для установки туда какой-либо операционной системы. Лучше используйте их для переноса файлов, хранения мультимедиа-контента или для кратковременного резервного копирования данных.

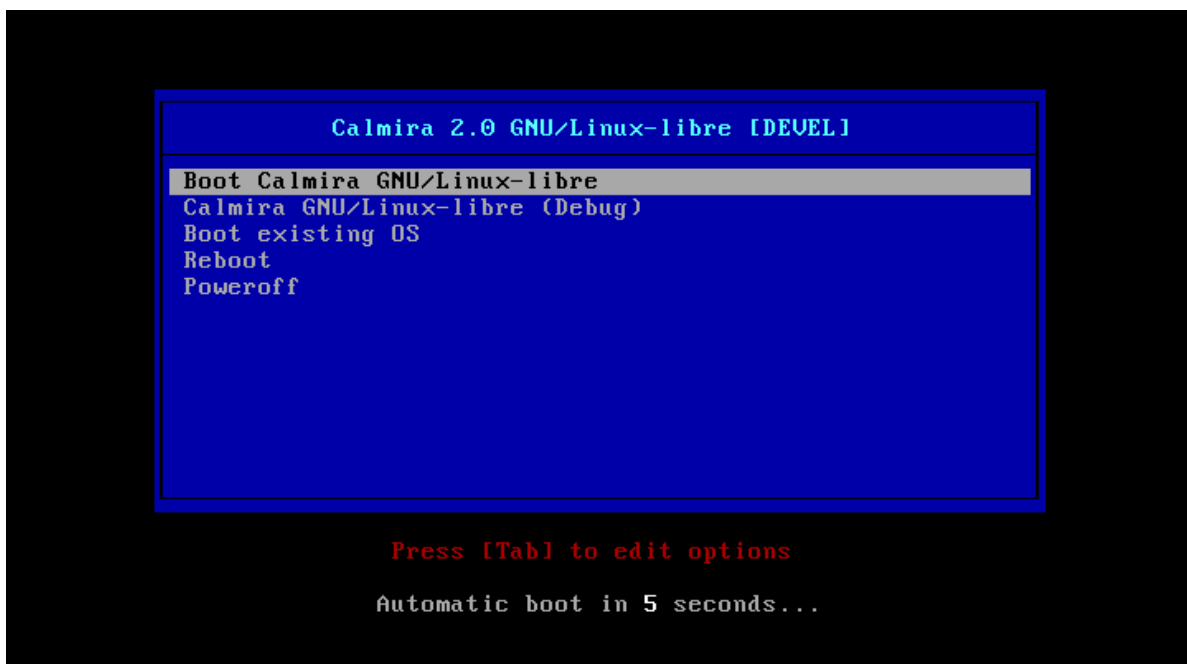
Принцип "всё есть файл"

Операционная система UNIX воспринимает многие вещи как файлы. Даже устройства. Файлы устройств содержатся в директории `/dev`. Поэтому если в данном руководстве указывается что-то вроде *раздел*

`sdc4` (или что-то вроде этого), то знайте, что здесь подразумевается раздел `/dev/sdc4`.

Начало установки

Вставьте загрузочный носитель в ПК и загрузитесь с него. Вас будет приветствовать меню загрузки:



Вам нужно выбрать пункт "Build Calmira GNU/Linux-libre". Для этого выберите данный пункт стрелками клавиш и нажмите `Enter`. Во время загрузки на экран будут выведены сообщения о процессе старта системы. Как правило, пользователю они нужны будут только в том случае, если во время загрузки системы произошла какая-либо ошибка. К примеру, система не загрузится, если вы загружаете её iso-образ с помощью программы Ventoy.

После окончания загрузки система запросит ваш логин и пароль:

```
[ 4.279744] sleep (90) used greatest stack depth: 14224 bytes left
[ 4.326518] mdev (91) used greatest stack depth: 14168 bytes left
[ 4.371148] findfs (92) used greatest stack depth: 13864 bytes left
[ 4.419812] loop0: detected capacity change from 0 to 482536
[ 4.429550] /dev/loop0: Can't open blockdev
created directory: '/mnt/rw/data'
created directory: '/mnt/rw/work'
[ 4.441539] overlayfs: "xino" feature enabled using 2 upper inode bits.
[ 5.718866] stty (111) used greatest stack depth: 13112 bytes left
[ 6.102886] mount (118) used greatest stack depth: 13064 bytes left
[ 6.880997] ip (175) used greatest stack depth: 13040 bytes left
[ 6.940991] S08localnet (163) used greatest stack depth: 12760 bytes left
[ 7.382675] udevd[200]: starting version 3.2.11
[ 7.869466] random: crng init done
[ 7.942993] udevd[201]: starting eudev-3.2.11
[ 18.017721] S30checkfs (245) used greatest stack depth: 12720 bytes left
[ 19.459011] rc (110) used greatest stack depth: 12232 bytes left

Welcome to Calmira GNU/Linux-libre 6.0.0-gnu (tty1)!

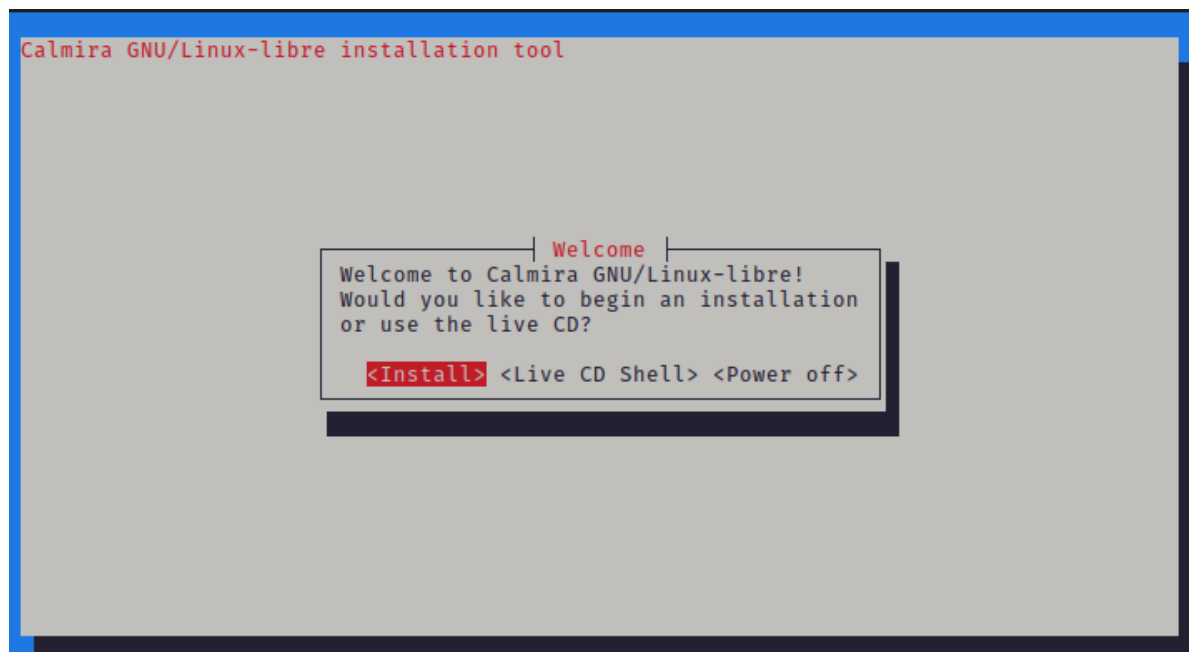
Date:   Sun Dec 18  2022 14:53:55
Kernel: 6.0.0-gnu
Host:   test-pc

test-pc login:
```

Введите следующие данные:

- **Логин:** root
- **Пароль:** root

После логина вас будет приветствовать диалоговое окно:



Вам будет предложено 3 пункта:

1. **Install** - запускает установщик системы;
2. **Live CD Shell** - это диалоговое окно закрывается и вы переходите в оболочку BASH загруженной ОС;
3. **Power off** - выключение/перезагрузка компьютера.

Для запуска установщика выберите пункт 1.

В том случае, если вы выбрали второй пункт, вы окажетесь в оболочке системы. Если вы захотите позже запустить установщик, введите команду `calminstall`.

Сбор начальных сведений

Выбор раскладки клавиатуры

Для начала установщик запросит у вас раскладку клавиатуры, которая будет использоваться в дистрибутиве после установки. Заметьте, что по умолчанию у вас будет всегда английская раскладка, а та, которую вы выбираете, будет второй.

Используя клавиши навигации найдите нужную вам раскладку и нажмите `Enter` для её выбора.

Установка имени хоста

Далее установщик запросит имя хоста для устанавливаемой системы.

Выбор устанавливаемых компонентов

Выберите дополнительные компоненты, которые вы хотите установить в систему. Заметьте, что менеджер системы портов `cport`, как и сама система портов, не входят в состав базовой системы и являются опциональными. Если вы не выберете пункт `ports.txz`, то вы не сможете управлять программным обеспечением в Calmira централизованным и простым путём. Однако, вы можете самостоятельно находить архивы с программным обеспечением (как исходный код, так и уже собранное ПО). Однако это достаточно сложный и небезопасный способ. Система портов - это простой и удобный способ управления программами. Она не содержит исходного кода ПО. Система портов состоит из множества файлов с информацией о ПО, а также со сборочными инструкциями для автоматизации загрузки, компиляции и установки программ.

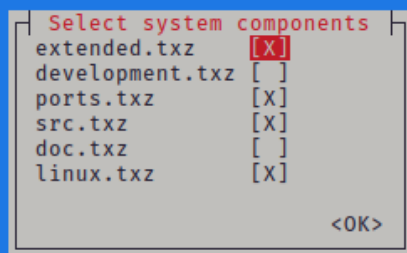
Если вы устанавливаете редакцию Extended (расширенную), то обязательно выберите пункт `extended.txz`, иначе ничего из расширенной редакции установлено не будет.

В зависимости от содержимого установочного iso-образа, а также от редакции дистрибутива, содержимое меню будет различаться. Все пункты приведены в списке ниже:

- `development.txz` - программное обеспечение для разработчиков;
- `doc.txz` - документация Calmira GNU/Linux-libre;
- `ports.txz` - система портов и `cport`;
- `src.txz` - исходный код системы;
- `linux.txz` - обычное ядро Linux (по умолчанию устанавливается Linux-libre);
- `extended.txz` - расширенная редакция дистрибутива. Включает в себя сервер Xorg и рабочее окружение Anomura;

Внимание!

Учтите, что программа установки не проверяет наличие свободного места на диске. Поэтому, когда вы выбираете тот или иной пункт, будьте уверены, что вам хватит для него места. К примеру, исходный код системы может занимать от 1 до 3 Гб дискового пространства, а пересборка всей системы может затребовать от 5 до 20 Гб.



Работа с дисками

Есть два способа осуществить разбиение дискового пространства для установки дистрибутива. *Автоматическое* разбиение самостоятельно настроит разделы диска, *ручное* разбиение даст возможность пользователю разметить жёсткий диск согласно своим требованиям. Требуется учесть, что автоматическое разбиение **уничтожит все существующие разделы, создав на их месте необходимые для дистрибутива**. Если на жёстком диске уже установлена какая-то операционная система, то она будет уничтожена.

В случае использования ручного разбиения запускается программа `/sbin/cfdisk`, которая позволяет пользователю самостоятельно создавать, удалять и изменять разделы на жёстком диске. В таком случае после разметки диска установщик запросит у вас некоторые данные вроде корневого раздела (`/`), раздела `/home`, раздела подкачки и пр.

После выбора ручной разметки вы увидите следующее окно:

```

Disk: /dev/vda
Size: 10 GiB, 10737418240 bytes, 20971520 sectors
Label: dos, identifier: 0x90909090

Device      Boot      Start        End    Sectors    Size    Id Type
>> /dev/vda1  *          64      20971519   20971456    10G    a5 FreeBSD

Partition type: FreeBSD (a5)
Attributes: 80
Filesystem UUID: 7651023417054215565
Filesystem LABEL: zroot
Filesystem: zfs_member

[Bootable] [ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ]
[ Write ] [ Dump ]

Quit program without writing changes
```

Внимание!

На данный момент функция автоматического разбиения диска нестабильна, поэтому по умолчанию отключена. Дождитесь следующих версий Calmira GNU/Linux-libre для использования этой опции.

Завершение сбора сведений и разметки дисков

Следующий шаг - ваш последний шанс прервать установку. Далее последует монтирование созданных разделов, создание необходимых файлов и каталогов и, наконец, распаковка архивов с системой. Как только все выбранные архивы распакуются, `calminstall` начнёт исполнять скрипты для настройки установленной системы.

Установка пароля `root`

Установка `root`-пароля обязательна. Заметьте, что во время ввода пароля для всех пользователей набираемые символы не отображаются на экране. После ввода пароля он будет запрошен ещё раз для предотвращения опечаток при наборе.

Создание пользователя

Далее установщик запросит у вас имя обычного пользователя. После ввода имени создайте пароль нового пользователя. Всё также, как и в предыдущем пункте: символы пароля не отображаются на экране, ввод пароля повторится ещё один раз для исключения возможности опечататься при вводе.

Часть 2

Данная часть содержит сведения о настройке дистрибутива Calmira GNU/Linux-libre после установки. Информация здесь предназначена для быстрого входа в систему. В части 3 приведены более конкретные и объёмные сведения о строении и принципе работы GNU/Linux.

Авторизация в системе

После того, как только что установленная Calmira GNU/Linux-libre запустилась, вас будет приветствовать примерно такое системное приглашение:

```
Welcome to Calmira GNU/Linux-libre 6.0.0-gnu!
```

```
Date:   Mon Dec 26   2022 23:23:40
```

```
Kernel: 6.0.0-gnu
```

```
Host:   test-pc
```

```
test-pc login:
```

- **Date** - дата и время полной загрузки системы.
- **Kernel** - версия ядра Linux(-libre).
- **Host** - имя компьютера. У каждого компьютера в UNIX есть своё имя.

Если во время установки вы указали имя хоста, то вместо `test-pc` у вас будет отображаться это имя. `6.0.0-gnu` - это версия ядра Linux, которое используется конкретно в вашей версии Calmira GNU/Linux-libre. В зависимости от версии системы будет изменяться и версия ядра.

Строка `test-pc login:` указывает вам, что необходимо ввести имя пользователя. На данный момент мы не создавали никаких пользователей, но у нас есть пользователь `root` - что-то вроде пользователя "Администратор" в MS-Windows.

Введите имя пользователя: `root`.

```
test-pc login: root
```

После чего система запросит у вас его пароль. Пароль по-умолчанию: `root`. При вводе пароля на экране не будут показываться его символы - это сделано ради безопасности!

```
test-pc login: root
Password:
```

В том случае, если вы правильно ввели логин и пароль, у вас загрузится оболочка `/bin/bash`:

```
Welcome to Calmira GNU/Linux-libre 6.0.0-gnu!

Date:   Mon Dec 26   2022 23:23:40
Kernel: 6.0.0-gnu
Host:   test-pc

test-pc login:
Password:
Last login: Sun Sep 25 20:54:53 +0300 2022 on /dev/tty1
No mail.
root:~#
```

Строка `Last login ...` показывает информацию о том, когда был предыдущий вход в систему:

- Дата входа
- В каком ТТУ был произведён вход¹

¹ разговор о ТТУ будет в главах далее. А пока просто считайте, что это консоль.

`root:~#` - это приглашение к вводу команды. Наличие этой строки означает то, что система загрузилась и готова к выполнению ваших команд.

Первичная настройка дистрибутива

После того, как система полностью загрузилась, вам необходимо выполнить некоторые действия по её первичной настройке.

Создание пользователей

Первым делом вам нужно создать обычного пользователя, из-под которого вы будете работать по умолчанию. Для создания пользователей используется программа `useradd`. Введите команду:

```
# useradd -m USERNAME -s /bin/bash
```

В том случае, если программа `useradd` завершилась без ошибок (а значит, нужный пользователь создан), в терминал не будет ничего выведено. Если `useradd` вывела в терминал какое-либо сообщение, то проанализируйте его и выполните те действия, которые там, возможно, указаны.

После чего вам нужно дать ему пароль:

```
# passwd USERNAME
```

Программа выведет в терминал:

```
Изменение пароля для USERNAME
Введите новый пароль (минимальная длина 5 символов)
Используйте комбинацию из символов в верхнем и нижнем регистре и цифры
Новый пароль:
Повторите новый пароль:
passwd: пароль изменён.
```

В том случае, если по умолчанию у вас не русский язык, а английский, то вывод будет таким:

```
Changing password for USERNAME
```

```
Enter the new password (minimum of 5 characters)
```

```
Please use a combination of upper and lower case letters and numbers.
```

```
New password:
```

```
Re-enter new password:
```

```
passwd: password changed.
```

`USERNAME` замените на имя пользователя. Например, `michail`. Имя пользователя должно быть на английском и содержать только буквы и, возможно, цифры. Все остальные символы использовать не рекомендуется во избежание ошибок в работе системы и стороннего программного обеспечения.

Объяснение команд

- `useradd` - программа для создания новых пользователей.
- `-m` в команде `useradd ...` - указывает программе `useradd` создавать домашнюю директорию пользователя в `/home/`.
- `-s /bin/bash` - закрепляет за создаваемым пользователем оболочку `/bin/bash`. Она будет использоваться для него по умолчанию. Других командных оболочек в Calmira GNU/Linux по умолчанию нет.
- `USERNAME` - вместо этих символов подставьте имя пользователя.
- `passwd` - устанавливает новый/изменяет пароль пользователя.

Для того, чтобы войти в систему от имени созданного пользователя, можно использовать два способа:

Способ 1:

```
# su USERNAME
```

Способ 2:

```
# exit
```

После выполнения данной команды вас будет приветствовать приглашение к входу в систему как на предыдущей странице. Введите имя созданного пользователя и его пароль.

Внимание!

Почему не рекомендуется работать от имени пользователя root?

`root` - это системный привилегированный пользователь. Если вы работаете от его имени, то у вас есть широкие возможности для модификации и изменении операционной системы GNU/Linux-libre. Вы можете случайно удалить какой-либо системный файл, в результате чего работа системы будет либо некорректной, либо вообще невозможной. Кроме того, ваша система становится уязвимой для не самых хороших личностей, которые могут либо поломать её, либо украсть какие-либо данные, либо даже отправить в систему вирус или троян.

Вопреки известному мнению о том, что "в GNU/Linux(-libre) нет вирусов" - это совсем не так. Вирусы, трояны и прочая гадость эксплуатирует уязвимости в программном обеспечении, в том числе и в ОС. А в любом ПО так или иначе есть баги, ошибки и уязвимости. Однако можно отметить тот факт, что для UNIX-подобных систем существует несколько меньше вирусов, чем для той же MS-Windows. Да и на компьютерах обычных пользователей вирусы - это редкое явление. Однако не следует пренебрегать своей безопасностью.

Используйте учётную запись `root` исключительно в крайних случаях. В том случае, если вам нужно выполнить команду/программу с повышенными привилегиями, то используйте либо программу `su -c "команда"`, либо программу `sudo` (порт `general/sudo`). Информация об этой программе будет в главе 3. Установить её легко:

```
# cport -i general/sudo
```

Настройка GRUB

Первичный конфиг GRUB находится в файле `/boot/grub/grub.cfg`:

```
# Begin /boot/grub/grub.cfg
set default=0
set timeout=3

insmod ext2
set root=(hd0,2)

menuentry "Calmira v2.0 Core Edition" {
    linux /boot/vmlinuz-6.0-gnu root=/dev/sda2 ro quiet
}
```

Параметр `default` определяет выбранный по умолчанию пункт меню GRUB.

Параметр `timeout` определяет время (в секундах), по прошествии которого, в случае бездействия пользователя, будет выбран стандартный пункт меню, указанный в параметре `default`.

TODO: больше конкретики

Редактирование файлов

TODO:

- ☐ Добавить информацию по работе с `x`, `dd`
 - ☐ Добавить информацию о работе в визуальном режиме
 - ☐ Добавить информацию о конфигурировании текстового редактора
-

Неотъемлимой частью работы с любой операционной системой или дистрибутивом операционной системы является редактирование простых текстовых файлов. Это могут быть конфигурационные файлы [а в Calmira GNU/Linux-libre их вы будете редактировать достаточно часто], код или что-то в этом роде. Следовательно, в системе должен присутствовать удобный и функциональный текстовый редактор, предназначенный для выполнения разных задач.

В релизах Calmira GNU/Linux 2021.x и 1.x по умолчанию был установлен текстовый редактор Vim, однако в Calmira GNU/Linux-libre 2.0 и далее Vim был заменён на NeoVim.

Здесь приводятся только самые базовые сведения об этом текстовом редакторе. Для получения развёрнутой информации прочтите документацию этого текстового редактора. Её вы можете найти на официальном сайте [neovim](https://neovim.org).

Для того, чтобы открыть в редакторе файл, введите:

```
% nvim FILENAME
```

`FILENAME` замените на имя файла, который нужно создать/отредактировать.

Сейчас вы находитесь в командном режиме. Для того, чтобы начать редактировать файл, нажмите клавишу `i`. После того, как работа над

редактированием файла завершилась, нажмите `Esc` для перехода в обычный режим. Для сохранения файла введите:

```
:w
```

Для выхода из редактора:

```
:q
```

Эквивалентная команда:

```
:wq
```

Сохранит файл и выйдет из редактора

Управление программным обеспечением

TODO

- ☐ Зависимости
 - ☒ Установка ПО
 - ☒ Удаление ПО
 - ☐ Обновление ПО
 - ☐ Обновление системы портов
 - ☒ Просмотр информации о ПО
 - ☒ Список портов
-

Всё программное обеспечение в Calmira GNU/Linux-libre собирается из исходного кода. Для многих людей это достаточно сложный и утомительный процесс, поэтому в системе присутствует средство для автоматизации этого процесса: система портов и программа, которая с ней взаимодействует.

Эта вещь называется "системой портов" потому, что она может функционировать на различных архитектурах процессора.

Система портов

Система портов представляет собой набор файлов в `/usr/ports`. В этой директории существует ряд поддиректорий, которые являются категориями ПО, сортирующее это ПО по предназначению. К примеру, поддиректория `base/` содержит в себе все системные порты, поддиректория `editors/` содержит в себе порты текстовых редакторов.

В категориях расположены ещё одни поддиректории - это либо каталоги с самими портами (имя каталога = имя порта), либо подкатегории.

Например, подкатегория `base/efi` содержит в себе порты для обеспечения корректной загрузки системы на UEFI.

Как различать каталог с портом от каталога с подкатегорией портов? Очень просто. В подкатегориях содержатся ещё одни каталоги, в которых расположены порты, принадлежащие данной подкатегории. В директории порта же находятся текстовые файлы, основные из них: `install`, `port.toml` и `files.list`. Об их предназначении далее.

Строение порта

Нахождение порта в файловой системе

Порт - это директория в одной из (под)категорий в `/usr/ports`. Когда мы называем имя порта, мы указываем его по следующему примеру:

категория/имя. Т.е. указываем путь до директории порта, опуская `/usr/ports`. Например, порт, находящийся по адресу `/usr/ports/base/ac1` в руководстве и прочей документации будет указываться как `base/ac1`. Тоже самое нужно указывать и при работе с менеджером системы портов `port`, речь о котором будет в следующей части этой страницы.

Файлы в каталоге порта

Порт - это набор определённых файлов, содержащих сведения о портируемом ПО, инструкции по сборке этого ПО из исходного кода и список файлов, которые этот порт устанавливает в систему:

- `port.toml` - информация о портируемом ПО. Содержит сведения об имени пакета, его версии, краткое описание, информация о сборщике данного порта, список зависимостей и прочие подобные сведения.
- `install` - исполняемый BASH-скрипт, содержащий инструкции по сборке ПО из исходного кода.

- `files.list` - список файлов, которые будут установлены в систему этим портом.

Кроме того, есть ряд необязательных файлов:

- `README.md` - Markdown-файл, содержащий сведения из файла `port.toml`. Предназначен для формирования этой документации.
- `files/` - директория, содержащая дополнительные файлы порта, такие, как сторонние рекомендуемые конфигурационные файлы или патчи.
- `description` - файл с расширенным описанием порта (т.к. описание в `port.toml` ограничено 1й строкой, файл `description` может содержать произвольный объем информации).

Обратите внимание!

Глава 2 является вводной, поэтому здесь не будет расширенных сведений о системе портов и управлении программным обеспечением. Для получения дополнительных сведений см. [главу 3](#).

Менеджер системы портов `sport`

`sport` предназначен для управления системой портов. Без него СП - просто набор ненужных никому файлов. `sport` же выполняет основные действия по работе с СП: чтение конфигурационного файла порта, скачивание и распаковка архивов с исходным кодом, исполнение сборочных инструкций порта и ведение локальной базы данных установленного в систему ПО.

Без `sport` пользователю самостоятельно приходилось бы собирать и хранить сведения об установленном в систему программном обеспечении, самостоятельно скачивать и распаковывать архив с исходниками ПО, а также запускать сборочные инструкции ПО. Данный

метод не является безопасным, а `cport` выполняет множество проверок портов и дистрибутива перед исполнением каких-либо действий.

Иначе говоря, `cport` автоматизирует рутинные действия пользователей и администраторов, к тому же, избавляя их от выполнения проверок и тестов на совместимость с дистрибутивом Calmira GNU/Linux-libre, например.

Возможности `cport`

- Установка ПО
- Удаление ПО
- Сбор, хранение и просмотр информации о ПО
- Ведение локальной базы данных установленного в систему ПО
- Обслуживание системы портов (проверка на ошибки, поддержание её в актуальном состоянии)
- Обновление ПО до новой версии
- etc.

Установка программного обеспечения

Для того, чтобы установить какой-либо порт, используется ключ `-i` (полная версия: `--install`):

```
# cport -i PORT_NAME
```

Вместо `PORT_NAME` подставьте имя нужного порта, например, `base/ac1`. Кроме того, вы можете указать несколько портов. Они будут собраны в указанной вами последовательности.

Обратите внимание!

`cport` не обрабатывает зависимости! Поэтому, если у программы есть зависимости, их вам нужно установить первыми, а уже потом собрать исходный порт.

Перед началом установки будет выведено следующее сообщение:

```
Selected ports:
  llvm-15.0.1

How much space will be taken up on the disk: 3891.2 Mb

:: Continue? [Y/n] y
```

В секции `Selected ports` отображается список выбранных портов. Например, вы устанавливаете порт `general/llvm` командой `spport -i general/llvm`. Вывод в секции выбранных портов будет таким же, как в показано выше: отображается не имя порта(ов), а имя пакета (указано в `port.toml` порта) и его версия (версия указывается через дефис: `-`).

Далее показывается место, которое будет занято портом(портами) на диске.

После чего вас поприветствует приглашение к вводу ответа:

```
:: Continue? [Y/n]
```

Если вы согласны продолжить работу, введите `y` и нажмите `Enter`. В противном случае введите `n` и нажмите `Enter`.

В том случае, если у порта есть какие-либо **необходимые** (`required`) зависимости, которые не установлены в систему, вывод будет немного другим:

```
Selected ports:
  xterm-370

Ports that are not installed on the system:
  xorg/x11-minimal/xapps-0.1 xorg/x11-minimal/xcursor-0.4.0
  xorg/x11-minimal/xbitmaps-1.1.2 xorg/x11-minimal/mesa-21.3.6
  general/libpng-1.6.37

How much space will be taken up on the disk: 836.2 Mb

:: Continue? [Y/n] y
```

Здесь появилась новая секция: `Ports that are not installed on the system`. В том случае, если вы видите эту секцию со списком необходимых портов, прервите сборку исходного порта и установите все недостающие зависимости из этого списка. Только после их установки устанавливайте искомый порт.

После того, как вы ответили утвердительно (`y` + `Enter`) на вопрос о продолжении сборки, `spport` начнёт скачивание и распаковку архива с исходным кодом:

```
==> Download port 'editors/gvim'...
100% [.....] 15996348 /
15996348
[✓] Download OK
==> Validating source file with md5sums... Passed
==> Extracting 'editors/gvim'...
```

Обратите внимание!

При скачивании порта будет показываться прогресс операции.

После скачивания архива будет произведена проверка корректности скачивания. Для этого сравниваются MD5-контрольные суммы и SHA256-контрольные суммы, указанные в `port.toml` порта и вычисленные у скачанного файла.

В том случае, если ошибок нет, `spport` приступит к распаковке архива. После распаковки он начнёт исполнение сборочных инструкций порта:

```
==> Building port 'editors/gvim'...
---== WARNING! ===--
Attention! It is not possible to build this package in multiple
threads, a single threaded build will be used.
configure: creating cache auto/config.cache
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no

...
```

Первая строка (`Building port '...`) сигнализирует о том, что началось исполнение сборочных инструкций порта. Все следующие за ней строки - это информация, которую выводит файл `install` - наши сборочные инструкции. Эта информация отображает прогресс выполнения и конкретные действия, которые выполняются. В случае возникновения ошибки эта информация будет неплохим средством для поиска причин возникновения неудачи и способа её решения.

Внимание!

В терминал выводится довольно много информации. Это нормально.

В том случае, если порт установлен корректно, в конце вы получите примерно следующее сообщение:

```
[✓] Build OK!
==> Updating the database... OK
==> Checking the port 'general/llvm' for correct installation... [✓]
Checking completed successfully
==> Clearing the cache...
[✓] 161 files was deleted!
```

Оно говорит вам о том, что:

1. Сборка завершена успешно.

2. В базу данных добавлена информация об установленном(ых) порте(ах).
3. Проверка на корректность установки порта(ов) завершилась без ошибок (проверяется наличие установленных в систему файлов).
4. Очищен кеш sport с файлами исходного кода и копиями установленных в систему файлов.

В том же случае, если во время сборки порта произошла ошибка, будет выведено следующее сообщение:

```
[!] Build instructions was returned a value other than 0!
```

Оно означает, что работа сборочных инструкций завершилась с ошибкой. Если вы видите это сообщение, то выше обязательно должна быть отображена причина. Прочитайте вывод выше этого сообщения для поиска и устранения неисправности.

В случае ошибки не будут выполнены следующие действия:

1. Устанавливаемый порт не будет добавлен в базу данных.
2. Следовательно, не будет проверки на корректность установки порта.
3. Не будет очищен кеш - это сделано специально для того, чтобы пользователю были доступны все файлы исходного кода, логи и прочие подобные вещи, которые могут понадобиться для исправления ошибки.

Обратите внимание!

Распакованные файлы архива с исходным кодом порта содержатся в директории `/usr/src`. Все логи сборки и прочие файлы содержатся именно там. Но скачанные *архивы* с исходным кодом ПО содержатся в директории `/var/cache/cport/downloads`.

Удаление программного обеспечения

Для удаления используется ключ `-r` (`--remove`):

```
# cport -r PORT_NAME
```

МСП пройдёт по файлу `files.list` порта. Каждый файл, который указанный порт устанавливает в систему, указывается на новой строке `files.list`. На основе содержимого этого файла `cport` составляет список файлов. После чего проходит по этому списку и удаляет из него те файлы, которые в системе отсутствуют. По завершению этого процесса все существующие в системе файлы удаляются.

`cport` также спросит у вас продолжение работы:

```
===== Removing port 'editors/gvim'
=====

:: Continue? [y/N]
```

Обратите внимание!

На данный момент `cport` **не** выводит информацию об удалении. Это скоро будет исправлено.

В случае согласия он приступит к удалению:

```
==> Remove the 'editors/gvim' files from the system...
==> System check...
==> Remove port 'editors/gvim' from database... OK
```

Просмотр информации о ПО

Для того, чтобы посмотреть информацию о порте(ах) используется ключ `-I` (`--info`):

```
# cport -I PORT_NAME
```

Будет прочитано содержимое файла `port.toml` указанного порта, после чего результат парсинга будет выведен на экран:

```
===== Information about port 'base/attr'
=====
Base information:
name       : attr
version    : 2.5.1
description : Commands for Manipulating Filesystem Extended Attributes.
maintainer  : Sergey Gaberer <nordic.dev@pm.me>
releases   : v2.0a1 v2.0a2 v2.0a3
priority   : system
usage      : 4.4
upgrade_mode : soft
build_time  : 0.1
installed   : True

Dependencies:
required    : base/libtool
```

В терминал будет выведена основная информация, взятая из файла `port.toml`:

- **name** - имя **программы** (а не порта - имя порта и так уже известно)
- **version** - версия порта/программы
- **description** - краткое описание порта
- **maintainer** - информация о создателе и/или сопровождающем порта
- **releases** - список релизов Calmira GNU/Linux-libre, с которыми совместим данный порт
- **priority** - приоритет порта (`system` - системный, `user` - пользовательский; системные порты нельзя удалять из системы во избежание ошибок в её работе)
- **usage** - сколько занимает место порт в файловой системе после установки (в мегабайтах)
- **upgrade_mode** - режим обновления (`soft` - порт просто переустанавливается, `hard` - все файлы порта удаляются, после чего устанавливается новая версия порта)
- **build_time** - примерное время сборки порта (в единицах, аналогичных SBU из руководства "Linux From Scratch")

- **installed** - установлен ли порт в систему (`True` - порт установлен, `False` - порт не установлен)

Далее идёт секция с описанием зависимостей порта:

- **required** - необходимые зависимости
- **recommend** - рекомендуемые зависимости
- **optional** - опциональные зависимости

Обратите внимание!

Какого-то из пункта с описанием типов зависимостей в выводе может и не быть (ровным счётом как и у ПО может не быть определённого типа зависимостей). Иногда возникают такие ситуации, когда секция `Dependencies` вообще пуста - ничего страшного или необычного в этом нет.

Просмотр списка портов

`sort` позволяет выводить и список всех портов, которые присутствуют в СП. Это нужно, к примеру, для поиска нужного ПО, либо для просмотра всех портов, которые установлены в системе. Для просмотра списка портов используется опция `list`.

В том случае, если она передаётся МСП **без** указания дополнительных аргументов, будет выведен неформатированный список всех портов в СП:

```
% cport list
base/acl
base/attr
base/autoconf
base/automake
base/bash
general/vagina
general/valgrind
general/wf-recorder
general/which_orig
general/which_script
general/wl-clipboard
general/wlr-randr
general/wofi
net/curl
net/libpsl
net/links
net/lynx
postcpl/btrfs-progs
postcpl/cracklib
postcpl/dosfstools
postcpl/fuse

...
```

Далее, вы можете передавать опции `list` дополнительные ключи:

- `-a` - форматированный вывод всех портов;
- `-i` - вывод только установленных в систему портов;
- `-I` - вывод только **НЕ** установленных в систему портов;
- `-b` - вывод портов с ошибками (отсутствует какой-то файл, либо статус порта (установлен/не установлен) определить невозможно).

Форматированный вывод всех портов

Вывод поделён на две колонки: первая колонка состоит из одного символа, который символизирует статус порта. Вторая колонка (напротив первой) - имя порта.

- Если первая колонка зелёного цвета - порт установлен в систему.
- Если первая колонка серого цвета - порт **не** установлен.
- Если первая колонка красного цвета - порт с ошибкой.


```
Терминал - admin@test-pc:~/calmira/cport
Файл  Правка  Вид  Терминал  Вкладки  Справка
admin:cport% cport list -a
█ - installed ports; ░ - not installed ports; █ - broken ports
-----
base/acl
base/attr
base/autoconf
base/automake
base/bash
base/bc
base/binutils
base/bison
base/bzip2
base/check
base/cmake
base/coreutils
base/cport
base/curl
base/diffutils
base/e2fsprogs
base/eudev
base/expat
base/file
base/findutils
base/flex
base/gawk
base/gcc
base/gdbm
base/gettext
base/glibc
base/gmp
base/gperf
base/grep
```

Вывод только установленных в систему портов

Вывод поделён на две колонки: в первой колонке имя порта, во второй - дата его установки в систему, либо дата его добавления в базу данных установленного ПО. Иногда возникают ситуации, когда необходимо пересобрать базу данных, в результате чего информация о дате установки портов может потеряться, в результате чего она заменяется информацией о времени добавления порта в БД.

```
Терминал - admin@test-pc:~/calmira/cport
Файл  Правка  Вид  Терминал  Вкладки  Справка

admin:cport% cport list -i
port name                installation date
-----
base/acl                  Wed Dec 21 22:52:52 2022
base/attr                  Wed Dec 21 22:52:52 2022
base/autoconf              Wed Dec 21 22:52:52 2022
base/automake              Wed Dec 21 22:52:52 2022
base/bash                  Wed Dec 21 22:52:52 2022
base/bc                    Wed Dec 21 22:52:52 2022
base/binutils              Wed Dec 21 22:52:52 2022
base/bison                 Wed Dec 21 22:52:52 2022
base/bzip2                 Wed Dec 21 22:52:52 2022
base/check                 Wed Dec 21 22:52:52 2022
base/cmake                 Wed Dec 21 22:52:52 2022
base/coreutils             Wed Dec 21 22:52:52 2022
base/cport                 Wed Dec 21 22:52:52 2022
base/curl                  Wed Dec 21 22:52:52 2022
base/diffutils             Wed Dec 21 22:52:52 2022
base/e2fsprogs             Wed Dec 21 22:52:52 2022
base/eudev                 Wed Dec 21 22:52:52 2022
base/expat                 Wed Dec 21 22:52:52 2022
base/file                  Wed Dec 21 22:52:52 2022
base/findutils             Wed Dec 21 22:52:52 2022
base/flex                  Wed Dec 21 22:52:52 2022
base/gawk                  Wed Dec 21 22:52:52 2022
base/gcc                   Wed Dec 21 22:52:52 2022
base/gdbm                  Wed Dec 21 22:52:52 2022
base/gettext               Wed Dec 21 22:52:52 2022
base/gmp                   Wed Dec 21 22:52:52 2022
base/gperf                 Wed Dec 21 22:52:52 2022
base/grep                  Wed Dec 21 22:52:52 2022
base/groff                 Wed Dec 21 22:52:52 2022
```

Вывод только НЕ установленных в систему портов

Здесь просто выводится список не установленных в систему портов.

```
Терминал - admin@test-pc:~/calmira/cport
Файл  Правка  Вид  Терминал  Вкладки  Справка
admin:cport% cport list -I
base/glibc
base/m4
base/meson
base/ncurses
base/ninja
base/openssl
base/p11-kit
base/perl
base/procps-ng
base/readline
base/shadow
base/syslogd
base/wheel
base/xml-parser
bs/cross_compiler/binutils-p1
bs/cross_compiler/gcc-p1
bs/cross_compiler/glibc
bs/cross_compiler/libstdcxx
bs/cross_compiler/linux-api-headers
bs/toolchain/after_chroot_preparations
bs/toolchain/bash
bs/toolchain/before_chroot_preparations
bs/toolchain/binutils-p2
bs/toolchain/bison
bs/toolchain/coreutils
bs/toolchain/diffutils
bs/toolchain/file
bs/toolchain/findutils
bs/toolchain/gawk
bs/toolchain/gcc-p2
bs/toolchain/gettext
```

Работа с системой инициализации

Обратите внимание!

Версии Calmira 2021.x, 1.x и тестовые версии 2.0 используют систему инициализации SysVInit. Стабильные версии Calmira v2.0 GNU/Linux-libre перешли на использование OpenRC.

Далее в этом руководстве пойдут сведения об OpenRC.

О системе инициализации

После того, как запустилось ядро Linux(-libre), запускается система инициализации. Это программа, запускающая процессы и демоны, необходимые для работы ОС GNU/Linux(-libre).

Существует много разных СИ. Ранее была очень популярна SysVInit, которая и по сей день считается классической СИ в GNU/Linux. Она была очень проста (если не сказать - примитивна). В начале 2010-х были совершены попытки заменить её на что-либо более удобное и "красивое" с точки зрения дизайна. В рамках Ubuntu GNU/Linux был разработан `upstart` (который более не применяется в этом дистрибутиве), а проектом Gentoo GNU/Linux была представлена система OpenRC. Но настоящую популярность завоевала СИ `systemd` от компании Red Hat.

У `systemd` было много недостатков, список, на наш взгляд самых противных, выложен в "Приложении" к этому руководству. Отдельно стоит отметить то, что эта система инициализации монолитна (а значит, не следует UNIX Way и не совместима с философией Calmira GNU/Linux-libre), а

также, из-за особенностей строения нашей системы, systemd там работает некорректно.

Исходя из всего этого (SysVinit устарел, systemd не подходит), мы выбрали систему инициализации OpenRC как неплохой вариант.

Использование

- **Список состояний запущенных служб:**

```
% rc-status
```

- **Проверка отказавших служб:**

```
% rc-status --crashed
```

- **Отображение всех доступных служб:**

```
% rc-update -v show
```

- **Изменить состояние службы:**

```
# rc-service daemon start/restart/stop/status
```

- **Включить/выключить службу:**

```
# rc-update add/del service
```

Часть 3

Процесс загрузки системы

Обратите внимание!

В данной статье будет пропущена информация о процессе загрузки самого ПК - BIOS, UEFI и пр.

Загрузчик

Задача загрузчика - инициализировать ядро Linux [в нашем случае]. В помощь ядру обычно используется образ начальной загрузки: `initrd` или `initramfs`, представляющий собой архив с образом файловой системы, разворачивающийся в ОЗУ в начале загрузки. В `initrd` / `initramfs` находятся нужные драйверы, скрипты и прочее, что необходимо для инициализации оборудования и прочих целей.

Ядро

После этого загружается ядро. Находится в `/boot`, в своём названии имеет `vmlinuz`:

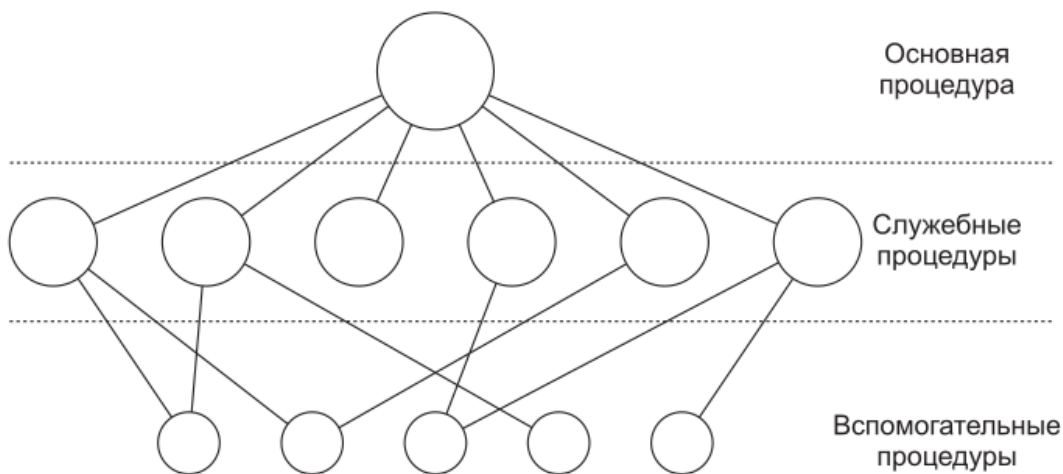
- `vm` - поддержка виртуальной памяти;
- `linu`;
- `z` - указывает на то, что файл сжат (формат сжатия, как правило, `zlib`, однако можно встретить `LZMA`, `BZIP2` и пр.).

Обратите внимание!

Ядро Linux является *монолитным*.

Монолитные ядра

Такая организация ядра является самой распространённой. Здесь всё ядро представлена единой программой - оно написано в виде набора процедур, связанных вместе в один большой исполняемый файл. При использовании этого решения каждая процедура может свободно вызывать любую другую процедуру, если та выполняет какое-либо полезное действие, в котором нуждается первая процедура. Наличие нескольких тысяч таких процедур нередко приводит к громоздкой и непонятной системе.



Очень примерное строение монолитного ядра; Э.Таненбаум, "Современные операционные системы"

Плюсы монолитного ядра

- Более прямой доступ к аппаратным средствам.
- Проще обмен данными между процессами.
- Процессы реагируют быстрее.

Минусы монолитного ядра

- Большой размер.
- Требуется больше ресурсов ПК.

- Менее безопасно.
- Все функции работают в одном адресном пространстве.

Особенности ядра Linux

У ядра Linux есть достоинство - это модули ядра. Подключать новые модули можно буквально на лету:

```
# modprobe MODULE_NAME
```

Предположим ситуацию. Вы подключили к компьютеру какое-то новое оборудование, а его поддержки в монолитном ядре нет. В итоге вам потребуется пересобрать ядро. Это может быть достаточно долгим и утомительным процессом. Поэтому, в случае ядра Linux, вы просто подключите нужный модуль.

Система инициализации

После ядра стартует `init` - подсистема инициализации в UNIX-системах, которая запускает все остальные процессы и демоны. Имеет PID = 1. Раньше де-факто стандартным инитом в GNU/Linux был SysVinit. Однако в самом начале 10-х годов были попытки заменить его на что-то более удобное, простое и быстрое. В итоге компания Canonical представила `upstart`, а в Gentoo появился OpenRC. Но настоящую популярность завоевал `systemd` от компании Red Hat, на который перешло большинство дистрибутивов GNU/Linux.

Достоинства systemd

- Сервисы запускаются параллельно, в результате чего вся система запускается быстрее.
- Возможность запуска сервисов по умолчанию.
- Работа с логами - `journald`.

- Простой и лаконичный синтаксис служб.
- Удобный контроль над службами.
- Анализ времени загрузки служб и всей системы в целом.

Недостатки systemd

- Сложная и монолитная система, заменяющая собой систему инициализации, планировщик, менеджер сети, пр.
- Достаточно тяжеловесная система по сравнению с другими инициаторами.
- Достаточно токсичный автор, вводящий в systemd много ненужного кала или удаляющий какие-либо кажущиеся ему ненужными функции.

TODO

- ☐ Добавить информацию о загрузке оболочки системы;
 - ☐ Добавить информацию о DM;
-

Ядро системы

Оболочка системы

Основные программы

BASH

Множественные консоли

Операционная система UNIX задумывалась как многозадачная и многопользовательская ОС. GNU/Linux перенимает и это.

Одной из составляющих многопользовательской и многозадачной системы UNIX являются множественные виртуальные консоли - там можно заходить в систему от имени разных пользователей и запускать от их имени всевозможные процессы.

При запуске системы открывается меню логина (см. [страницу](#) об авторизации в системе) в TTY под номером 1. Всего в системе активировано 6 виртуальных консолей (TTY). Переключаться между ними вы можете, используя сочетания клавиш **Alt** + **F1**, **F2**, **F3** ... **F6**.

При переключении от одной консоли к другой, GNU/Linux заботится о сохранении и восстановлении вывода на экран. Результатом является "иллюзия" наличия множества виртуальных экранов, которые могут быть использованы для ввода и запуска команд. Программы, которые вы запускаете в одном TTY, не прекращают своего выполнения, когда вы переключаетесь на другой TTY.

Перемещение по директориям

Работа с файлами

Переменные окружения

Пути до исполняемых файлов

Получение интерактивной справки

Ввод и вывод

Основные ошибки

Управление процессами

Компоненты процесса

PID и PPID

UID и EUID

GID и EGID

Жизненный цикл процесса

Сигналы

Использование kill

Использование ps

Использование top

Изменение приоритета выполнения

Файловая система /proc

Управление пользователями

Учётная запись root

**Замена идентификатора
пользователя: su**

Использование sudo

Прочие системные учётные записи

Группы пользователей

Создание пользователей

Управление правами доступа

Стандартное управление доступом

Расширенное управление доступом

Установка флагов setuid и setgid

ACL

Управление программным обеспечением

Часть 4

Продвинутое использование Neovim

Продвинутое использование sudo

Продвинутое использование системы портов

Изменение параметров загрузки системы