ОС МСВСфера. База знаний

версия 1.0

Инв. № подл.	Подпись и дата	Взам. инв №	Инв. № дубл.	Подпись и дата

2025

Оглавление

Безопасность **4**
Технология одновременной многопоточности (SMT) 4
Добавление Secure Boot ключа для безопасной загрузки МСВСфера ОС на системах с UEFI 6
Использование алгоритмов ГОСТ 14
Установка веб-браузера Chromium ГОСТ 21
Установка обновлений 22
База данных уязвимостей МСВСфера ОС 24
Уровни опасности уязвимостей 29
Установка КриптоПро CSP и расширения ЭЦП для браузера 32

Стороннее ПО и совместимость **45**
Работа с Flatpak 45
Добавление в систему сторонних репозиториев 51
Мигратор 53
Переход на МСВСфера ОС с RHEL-подобных дистрибутивов в рамках одной версии 69

Настройка операционной системы **73**
Как перенести значок приложения на рабочий стол 73
Автоматизированная установка (Kickstart-установка) 76
Подключение учётной записи Microsoft Exchange 169
Обновление ядра системы до версий 6.1 или 6.6 180
Как перенести пользовательскую панель на экран, который назначен основным в настройках дисплея 181
Подключение репозиториев LibreOffice и Virtualization 184
Как создать ярлык на папку на рабочем столе 186
Автоматическое подключение сетевых ресурсов при входе в сеанс GNOME . 188

Практические рекомендации **190**
Как установить Telegram на МСВСфера 9 АРМ 190
Как установить и настроить серверную часть TrueConf на МСВСфера 9 Сервер 192
Как установить и настроить K3s Rancher (kubernetes) на МСВСфера 9 Сервер 194
Как установить Яндекс.Браузер в МСВСфера ОС 198
Как установить и настроить почтовый клиент Thunderbird 201
Как установить Zoom 206
Как установить Steam из репозитория Third-Party 208
Как установить VK Play через PortProton 211
Работа с Wine в ОС МСВ Сфера 214
Дополнительные инструменты Wine 222
Как установить сторонние браузеры в МСВСфера ОС 228
Альтернативные решения для работы с документами, таблицами и презентациями. 232
Как создать снимок экрана и записать видео с экрана в МСВСфера 236
Установка МСВСфера в Windows с помощью WSL 238

Текстовые редакторы в МСВСфера 239
Текстовые редакторы vi и vim 244
Утилиты для архивации и распаковки файлов 253
Работа с файлами и папками в МСВСфера 263
Организация удалённого доступа в МСВСфера 269
Добавление принтера с помощью сервиса CUPS в ОС МСВСфера 275
Файловые менеджеры в МСВСфера 278
Мониторинг производительности системы с помощью утилиты atop 292
Редактирование PDF-файлов в МСВСфера 300

Безопасность

Технология одновременной многопоточности (SMT)

Введение

Технология одновременной многопоточности (англ. Simultaneous Multithreading, далее **SMT**, также известна как технология Intel® Hyper-Threading) позволяет одновременно исполнять инструкции из нескольких (как правило двух) потоков на одном физическом ядре процессора.

На некоторых классах задач использование SMT позволяет более эффективно использовать ресурсы центрального процессора.

Проблемы с безопасностью SMT

За последние годы были выявлены различные уязвимости, так или иначе связанные с технологией SMT:

- **L1TF (L1 Terminal Fault,также используется название Foreshadow: https://www.securitylab.ru/news/495093.php, https://foreshadowattack.eu/):**
  - CVE-2018-3620
  - CVE-2018-3646
- **MDS (Microarchitectural Data Sampling: https://www.securitylab.ru/news/499094.php, https://mdsattacks.com/):**
  - CVE-2018-12130
  - CVE-2018-12126
  - CVE-2018-12127
  - CVE-2019-11091

Как правило, уязвимости удаётся исправить с помощью программных обновлений, в том числе и микрокода центрального процессора. Однако в некоторых случаях может потребоваться отключение SMT.

Отключение SMT

Существует несколько способов отключения поддержки SMT:

- в настройках BIOS или UEFI (аппаратный, перманентный)
- через параметры загрузки ядра (программный,перманентный)
- используя интерфейс управления SMT в Sysfs (программный, временный)

Аппаратное отключение SMT

Для аппаратного отключения SMT требуется перезагрузить в систему, зайти в настройки BIOS/UEFI, найти там опцию управления SMT (материнские платы для процессоров AMD) или Hyper Threading (материнские платы для процессоров Intel) и отключить её.

За дополнительной информацией обратитесь к документации от поставщика вашего оборудования.

Отключение SMT через параметры загрузки ядра

Для отключения SMT через параметры загрузки ядра необходимо добавить в строку `GRUB_CMDLINE_LINUX` в файле `/etc/default/grub` аргумент `nosmt`:

```
$ sudo sed -i 's/GRUB_CMDLINE_LINUX="[^"]*/& nosmt/' /etc/default/grub
```

и обновить конфигурацию загрузчика:

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

Для применения изменений необходимо перезагрузить компьютер. Проверить статус активности SMT можно с помощью следующей команды:

```
$ cat /sys/devices/system/cpu/smt/control
  off
```

`off` в выводе означает что технология SMT отключена, `on` — включена.

Отключение SMT через Sysfs

Отключение SMT без перезагрузки возможно через интерфейс управления в Sysfs, однако данное решение является временным и будет действовать только до следующей перезагрузки компьютера:

```
$ echo off | sudo tee /sys/devices/system/cpu/smt/control
```

Добавление Secure Boot ключа для безопасной загрузки МСВСфера ОС на системах с UEFI

Глоссарий

- BIOS — Basic Input/Output System, базовая система ввода/вывода;
- UEFI — Unified Extensible Firmware Interface, единый расширяемый интерфейс прошивки;
- Secure Boot — безопасная загрузка;
- MOK — Machine Owner Key, ключ владельца компьютера (пользовательский ключ).

Область применения

Технология UEFI Secure Boot применяется в основном на современных персональных компьютерах (ПК), автоматизированных рабочих местах (АРМ), серверах, планшетных компьютерах и других электронных вычислительных машинах (ЭВМ).

Загрузка с включенным режимом Secure Boot имеет некоторые преимущества — UEFI Secure Boot (в отличие от функционала BIOS) помогает защититься от вредоносных программ до загрузки операционной системы.

В случае загрузки с отключенным Secure Boot, после включения ЭВМ может произойти запуск вредоносного EFI-приложения с любого устройства, подключенного к USB-порту, например, с USB-накопителя. В результате чего в ЭВМ может проникнуть зловредное программное обеспечение (ПО), исполняемое в кольце защиты центрального процессора (ЦП), которое недоступно антивирусу в операционной системе.

Поэтому получается, что антивирус никогда не сможет обнаружить это зловредное ПО, так как фактически операционная система (ОС) и антивирус будут исполняться в среде этого ПО. Удаление такого зловредного ПО возможно только восстановлением прошивки с помощью программатора, иногда с выпаиванием микросхемы flash-памяти, что обычно возможно сделать только в специализированной мастерской.

Принцип работы технологии Secure Boot заключается в проверке электронной цифровой подписи (ЭЦП) EFI-приложения перед его выполнением при запуске ЭВМ относительно ключей, находящихся в хранилищах UEFI Secure Boot (PK, KEK, DB и DBX).

Управление Secure Boot

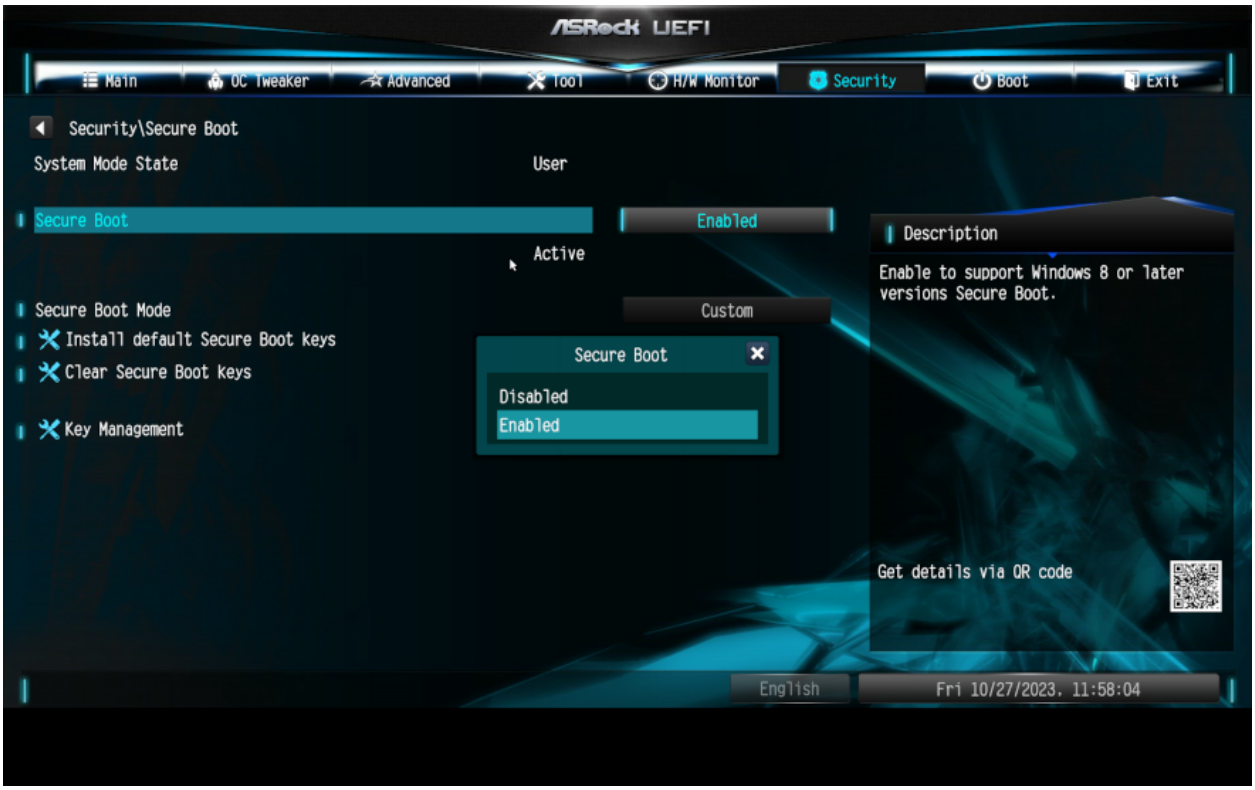
Отключение Secure Boot

Несмотря на то что Secure Boot предназначен для защиты системы и разрешает в процессе загрузки только аутентифицированные двоичные файлы, он является дополнительной функцией для большинства систем общего назначения. По умолчанию безопасная загрузка UEFI может быть отключена на большинстве компьютеров. Поставщики систем сами решают, какие системные политики будут реализованы на данной машине. Например, при развертывании киосков, банкоматов или субсидированных устройств по соображениям безопасности владелец этой системы не хочет, чтобы её меняли.

Включение Secure Boot

Для использования Secure Boot необходимо удостовериться, что значение переменной `Secure Boot Mode` установлено в `Standard`. Это необходимо сделать после того, как будет установлен MOK.

Проверить включение Secure Boot можно в **Setup Utility** в меню **Security**.



Режимы Secure Boot

В Setup Utility для Secure Boot можно задать два режима работы — **Standard** или **Custom**.

- Режим **Standard** служит для запуска загрузчиков, подписанных корпорацией Microsoft (запуск Windows), а также загрузчиков Shim для запуска Linux и других ОС, имеющих соответствующую подпись (МСВСфера ОС работает в режиме **Standard** после установки МОК).
- Режим **Custom** предназначен для управления работой Secure Boot вручную. В этом режиме необходимо установить МОК.

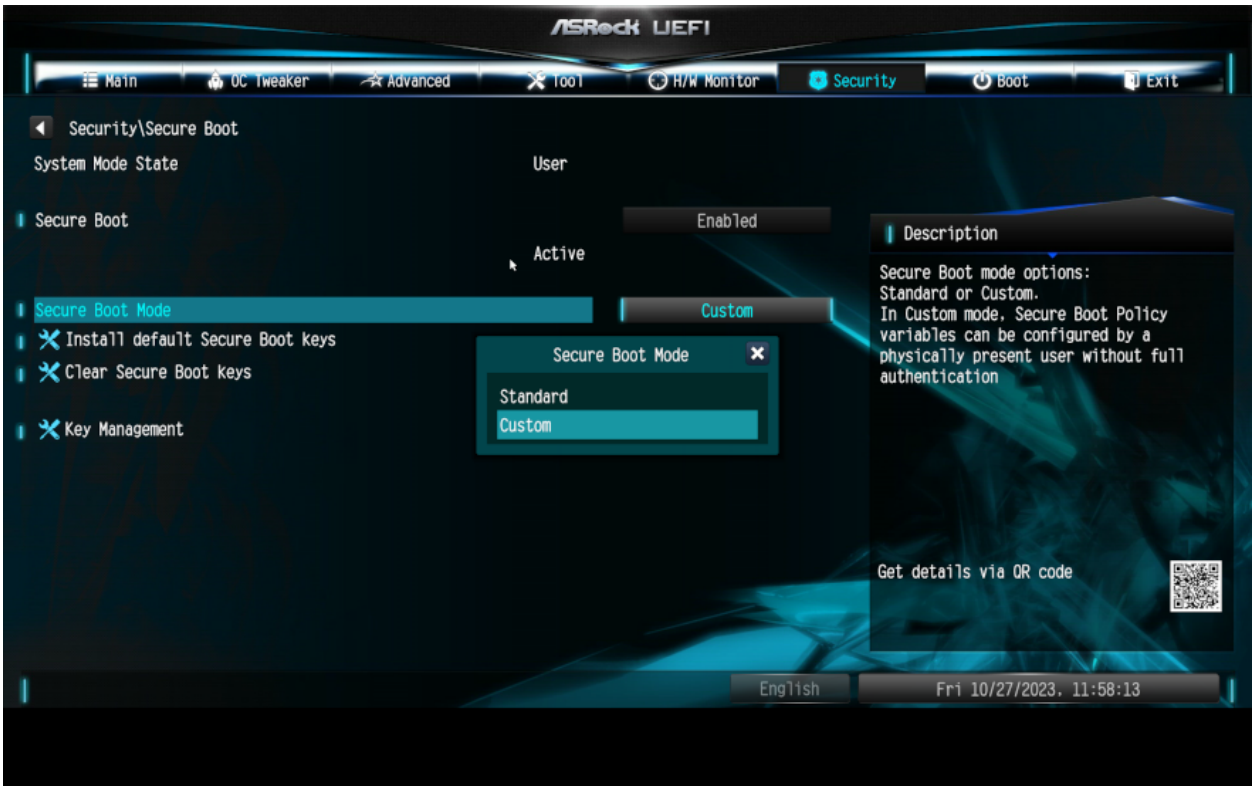


Рис. 1: Режимы работы Secure Boot

Примечание: Если вы планируете устанавливать и использовать в ОС проприетарные драйверы nVidia, загруженные с сайта nvidia.com вручную, то необходимо удалить ключи Secure Boot. В этом режиме Secure Boot можно оставить включенным, но проверка запускаемых загрузчиков производиться не будет и, соответственно, система не будет защищена. Использование проприетарных драйверов nVidia будет рассмотрено в отдельной статье.

Рис. 2: Удалить ключи Secure Boot

Добавление сертификатов МСВСфера ОС в хранилище ЭЦП Secure Boot

Добавить сертификат можно тремя различными способами:

- используя утилиту mokutil
- используя метод Enroll EFI Image в Setup Utility
- используя метод добавления сертификата в хранилище Secure Boot при запуске ОС

Создание резервной копии хранилища ЭЦП UEFI Secure Boot

Перед изменениями Secure Boot в Setup Utility рекомендуется создать резервную копию хранилищ, таких как Platform Key (PK), Key Exchange (KEK), Authorized Signatures (db) и Forbidden Signatures (dbx). Для этого можно использовать интерфейс Setup Utility (возможность зависит от конкретной реализации UEFI).

Для сохранения резервной копии необходимо иметь на диске раздел, на котором создана файловая система FAT32. В системах с UEFI обычно имеется раздел диска (таблица разделов GPT) с типом 1, EFI System, C12A7328-F81F-11D2-BA4B-00A0C93EC93B, который отформатирован FAT.

Хранилища ЭЦП Secure Boot

Secure Boot имеет несколько хранилищ сертификатов:

- Platform Key (PK) — содержит сертификат производителя оборудования;
- Key Exchange (KEK) — содержит сертификат корпорации Microsoft (для обновления DB и DBX);
- Authorized Signatures (DB) — содержит разрешенные сертификаты;
- Forbidden Signatures (DBX) — содержит отозванные сертификаты (чёрный список).

Сертификат МСВСфера ОС необходимо добавить как MOK.

Установка сертификата

С помощью mokutil

Установка производится с помощью утилиты mokutil, установленной в операционную систему. Если у вас уже установлена МСВСфера ОС и вы хотите настроить загрузку Secure Boot, то установить mokutil и RPM-пакет, содержащий сертификаты МСВСфера ОС, можно следующей командой:

```
$ sudo dnf install mokutil sphere-sb-certs
```

Если планируется устанавливать МСВСфера ОС в режиме Secure Boot **Standard**, то для импортирования сертификата можно воспользоваться загрузочным образом *MSVSphere-9.2-x86_64-arm.iso <https://repo1.msvsphere-os.ru/msvsphere/9/isos/x86_64/>*.

До загрузки с ISO необходимо включить режим Secure Boot **Custom** с включённым режимом запроса исполнения загружаемого файла (зависит от реализации UEFI Secure Boot) или произвести очистку всех ключей.

Далее импортировать MOK и установить ключи по умолчанию (MOK после этой процедуры должен остаться в UEFI). После чего можно включать режим Secure Boot **Standard** и проводить установку МСВСфера ОС в обычном режиме.

Если была произведена очистка ключей, желательно после установки МСВСфера ОС запустить следующую команду для обновления сигнатур в DB и DBX (для повышения безопасности системы):

```
$ sudo fwupdtool update
```

Установка сертификата производится следующей командой:

```
$ sudo mokutil --import /usr/share/pki/sb-certs/secureboot-ca-x86_64.cer
```

После чего следует перезагрузиться и проверить корректность импортирования сертификата командой:

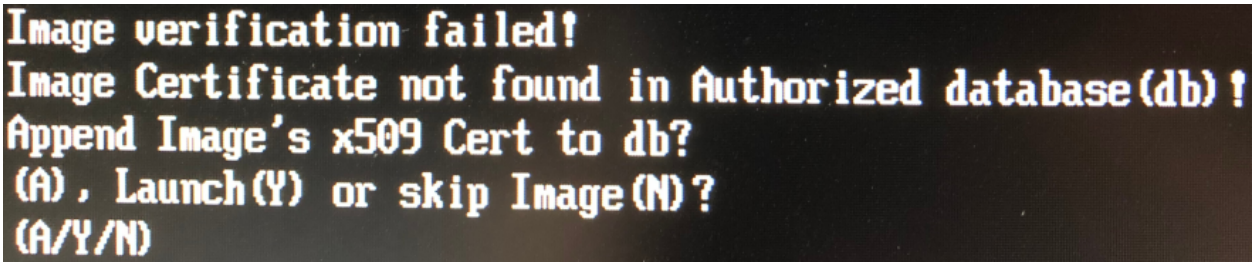
```
$ sudo mokutil --list-enrolled
```

С помощью метода Enroll EFI Image в Setup Utility

Установить сертификат можно методом импортирования сертификата из файла, который подписан сертификатом МСВСфера ОС (возможность использования этого метода зависит от конкретной реализации UEFI).

С помощью добавления сертификата в хранилище Secure Boot при запуске ОС

Установить сертификат также можно методом добавления сертификата в DB предварительно включив режим запроса исполнения загружаемого файла в Secure Boot (возможность использования зависит от конкретной реализации UEFI).



После нажатия клавиши **A** (Append Image`s Cert to DB) происходит добавление сертификата в DB и последующие загрузки системы (если не изменится сертификат) будут происходить без запросов.

Использование алгоритмов ГОСТ

В МСВСфера 9 Сервер вы можете включить поддержку российских алгоритмов хэширования, шифрования и электронной подписи (алгоритмы ГОСТ).

Пакет `crypto-policies-x.x.x` содержит библиотеки, необходимые для работы алгоритмов ГОСТ.

В пакет добавлены политики (**.pol**), реализующие необходимые алгоритмы ГОСТ, а также модули (**.mod**), котороые могут подключаться к любым политикам и в них добавлять поддержку алгоритмов ГОСТ.

Модуль GOST.mod

- ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 — алгоритмы электронной цифровой подписи.
- ГОСТ Р 34.11-94 — алгоритм хэширования. 256-битное значение хэша.
- ГОСТ Р 34.11-2012 — алгоритм хэширования. 256- и 512-битные значения хеша.
- ГОСТ 28147-89 — симметричное шифрование с 256-битным ключом. Доступны режимы CBC, CFB и CNT. Для усложнения статистического анализа поддерживается «key meshing».
- ГОСТ 28147-89 в режиме выработки имитовставки (MAC). Основан на алгоритме хэширования с использованием алгоритмов симметричного шифрования. Он имеет 256-битный симметричный ключ и разрядность от 8 до 64 (по умолчанию 32) бит.
- ГОСТ Р 34.13–2015 — симметричное шифрование «Кузнечик» («Grasshopper»).
- Алгоритмы базового профиля (т.е если базовый профиль `DEFAULT`, то к его алгоритмом добавляются указанные выше).

Модуль PAM-GOST.mod

- Задание хешей паролей в соответствии с ГОСТ Р 34.11-2012.
- Алгоритмы базового профиля (т.е если базовый профиль `DEFAULT`, то к его алгоритмом добавляются указанные выше).

Модуль SSSD-PAM-GOST.mod

- Задание хешей паролей в соответствии с ГОСТ Р 34.11-2012.
- Алгоритмы базового профиля (т.е если базовый профиль `DEFAULT`, то к его алгоритмом добавляются указанные выше).

Модуль PATCH-PAM-GOST.mod

- Задание хешей паролей в соответствии с ГОСТ Р 34.11-2012.

- Алгоритмы базового профиля (т.е если базовый профиль `DEFAULT`, то к его алгоритмом добавляются указанные выше).

Модуль TLSGOST.mod

- ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 — алгоритмы электронной цифровой подписи.
- ГОСТ Р 34.11-94 — алгоритм хэширования. 256-битное значение хэша.
- ГОСТ Р 34.11-2012 — алгоритм хэширования. 256- и 512-битные значения хеша.
- ГОСТ 28147-89 — симметричное шифрование с 256-битным ключом. Доступны режимы CBC, CFB и CNT. Для усложнения статистического анализа поддерживается «key meshing».
- ГОСТ 28147-89 в режиме выработки имитовставки (MAC). Основан на алгоритме хэширования с использованием алгоритмов симметричного шифрования. Он имеет 256-битный симметричный ключ и разрядность от 8 до 64 (по умолчанию 32) бит.
- ГОСТ Р 34.13–2015 — симметричное шифрование «Кузнечик» («Grasshopper»).
- Поддержка вышеперечисленных алгоритмов ГОСТ для gnutls.
- Набор алгоритмов оптимизирован для поддержки ГОСТ в OpenVPN.

Политика GOST-ONLY.pol

- ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 — алгоритмы электронной цифровой подписи.
- ГОСТ Р 34.11-94 — алгоритм хэширования. 256-битное значение хэша.
- ГОСТ Р 34.11-2012 — алгоритм хэширования. 256- и 512-битные значения хеша.
- ГОСТ 28147-89 — симметричное шифрование с 256-битным ключом. Доступны режимы CBC, CFB и CNT. Для усложнения статистического анализа поддерживается «key meshing».
- ГОСТ 28147-89 в режиме выработки имитовставки (MAC). Основан на алгоритме хэширования с использованием алгоритмов симметричного шифрования. Он имеет 256-битный симметричный ключ и разрядность от 8 до 64 (по умолчанию 32) бит.
- ГОСТ Р 34.13–2015 — симметричное шифрование «Кузнечик» («Grasshopper»).

Политика GOST-ONLY-PAM.pol

- ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 — алгоритмы электронной цифровой подписи.
- ГОСТ Р 34.11-94 — алгоритм хэширования. 256-битное значение хэша.
- ГОСТ Р 34.11-2012 — алгоритм хэширования. 256- и 512-битные значения хеша.

- ГОСТ 28147-89 — симметричное шифрование с 256-битным ключом. Доступны режимы CBC, CFB и CNT. Для усложнения статистического анализа поддерживается «key meshing».
- ГОСТ 28147-89 в режиме выработки имитовставки (MAC). Основан на алгоритме хэширования с использованием алгоритмов симметричного шифрования. Он имеет 256-битный симметричный ключ и разрядность от 8 до 64 (по умолчанию 32) бит.
- ГОСТ Р 34.13–2015 — симметричное шифрование «Кузнечик» («Grasshopper»).
- Задание хешей паролей в соответствии с ГОСТ Р 34.11-2012.

Выберите политику, которую необходимо применить или модуль, который необходимо подключить к политике и сделайте это из командной строки или в Cockpit (веб-интерфейс удалённого администрирования).

Здесь `GOST-ONLY-PAM`, `GOST-ONLY`, `PAM-GOST` и `GOST` — имена политик и модулей, которые используются в командах их подключения.

Требования к окружению для работы с алгоритмами ГОСТ

Для корректной работы алгоритмов ГОСТ с OpenSSL установите пакет `openssl-gost-engine` следующей командой:

```
$ sudo dnf install openssl-gost-engine -y
```

Применение политики с поддержкой ГОСТ из командной строки

1. Для просмотра текущей политики выполните следующую команду (в выводе команды показана текущая политика):

```
$ update-crypto-policies --show
  DEFAULT
```

2. Примените необходимую политику.

```
$ sudo update-crypto-policies --set ИМЯ_ПОЛИТИКИ
```

   Пример применения политики `GOST-ONLY`.

```
$ sudo update-crypto-policies --set GOST-ONLY
[sudo] пароль для user1:
Setting system policy to GOST-ONLY
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
Изменения успешно применены.
```

3. Проверьте, что текущая политика по умолчанию изменилась:

```
$ update-crypto-policies --show
  GOST-ONLY
```

Подключение модуля с поддержкой ГОСТ к существующей политике из командной строки

1. Для просмотра текущей политики выполните следующую команду (в выводе команды показана текущая политика):

```
$ update-crypto-policies --show
  DEFAULT
```

2. Подключите необходимый модуль. Он будет добавлен к текущей политике. Если текущая политика — `DEFAULT`, то общая команда подключения модуля выглядит следующим образом:

```
$ sudo update-crypto-policies --set DEFAULT:ИМЯ_МОДУЛЯ
```

   Пример подключения модуля `PAM-GOST` к политике `DEFAULT`.

```
$ sudo update-crypto-policies --set DEFAULT:PAM-GOST
[sudo] пароль для user1:
Setting system policy to DEFAULT:PAM-GOST
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies
to fully take place.
Изменения успешно применены.
```

   Для подключения нескольких модулей перечислите их через `:`:

```
$ sudo update-crypto-policies --set DEFAULT:PAM-GOST:GOST
```

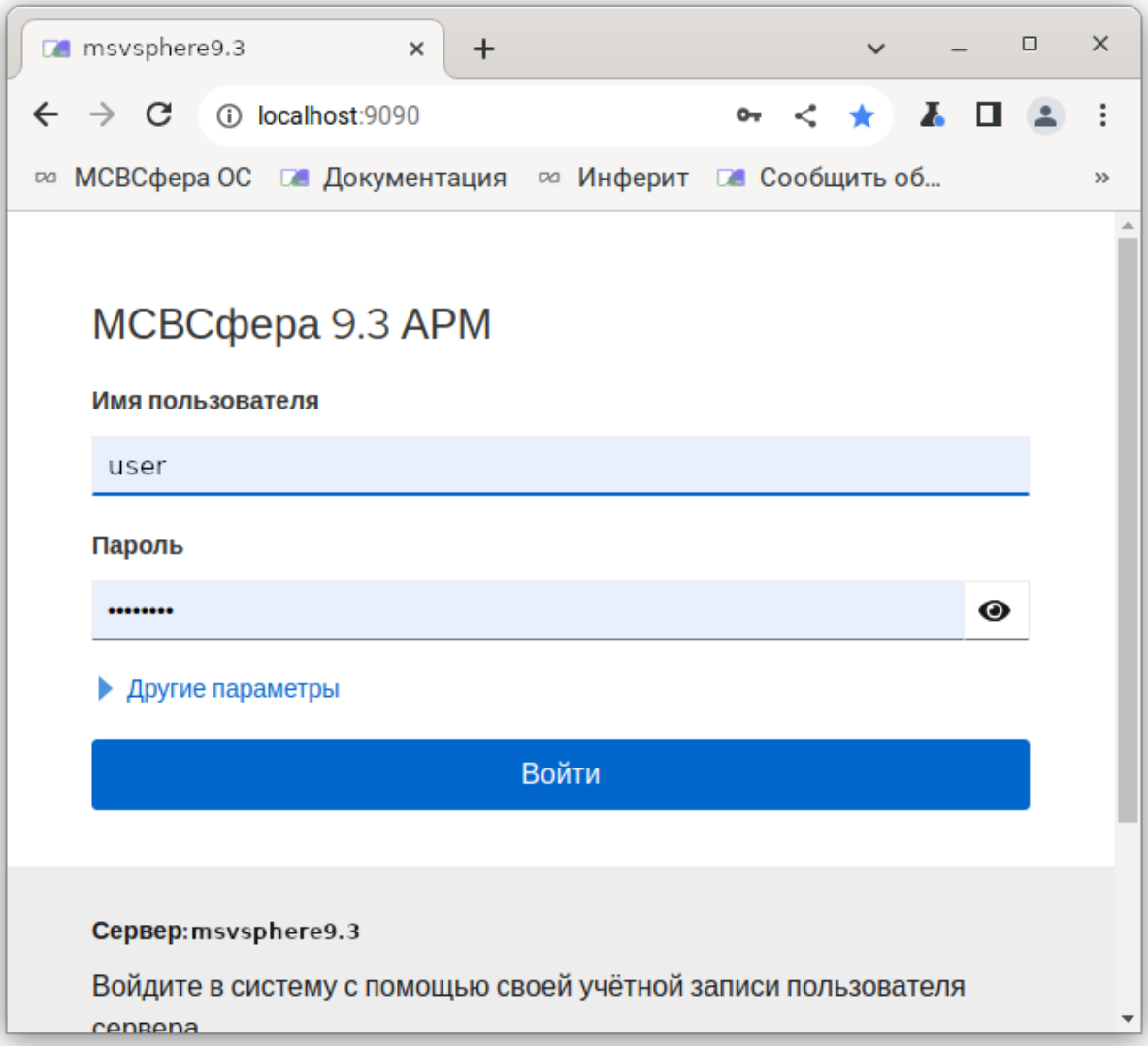
3. Проверьте, что текущая политика по умолчанию изменилась:

```
$ update-crypto-policies --show
DEFAULT:PAM-GOST
```

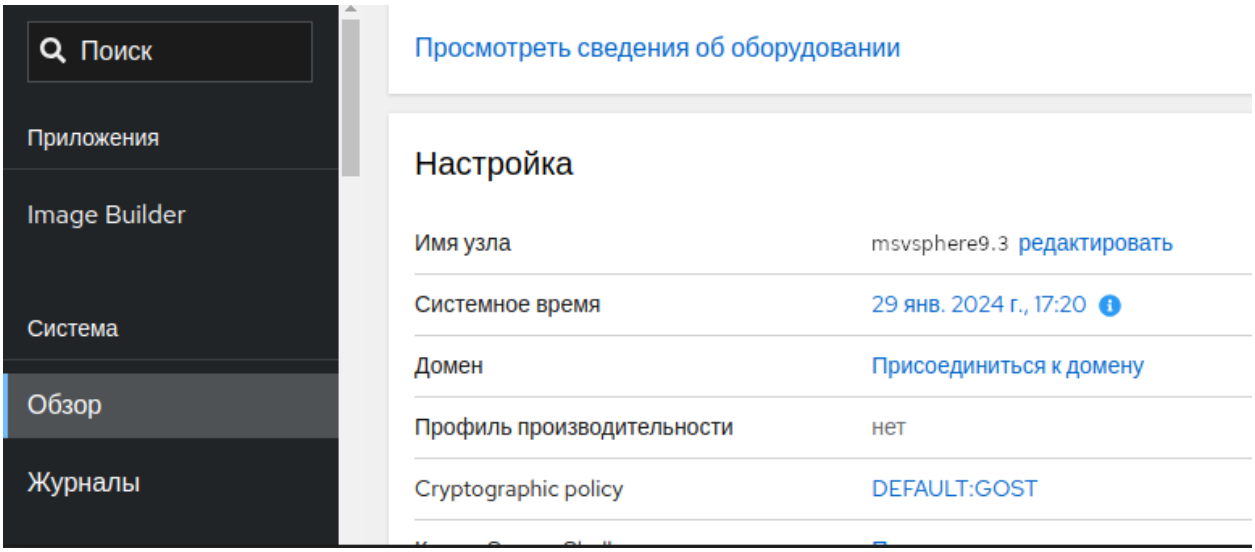
Применение политики с поддержкой ГОСТ в Cockpit

Для применения политик с поддержкой ГОСТ должна быть установлена последняя версия Cockpit с поддержкой ГОСТ.

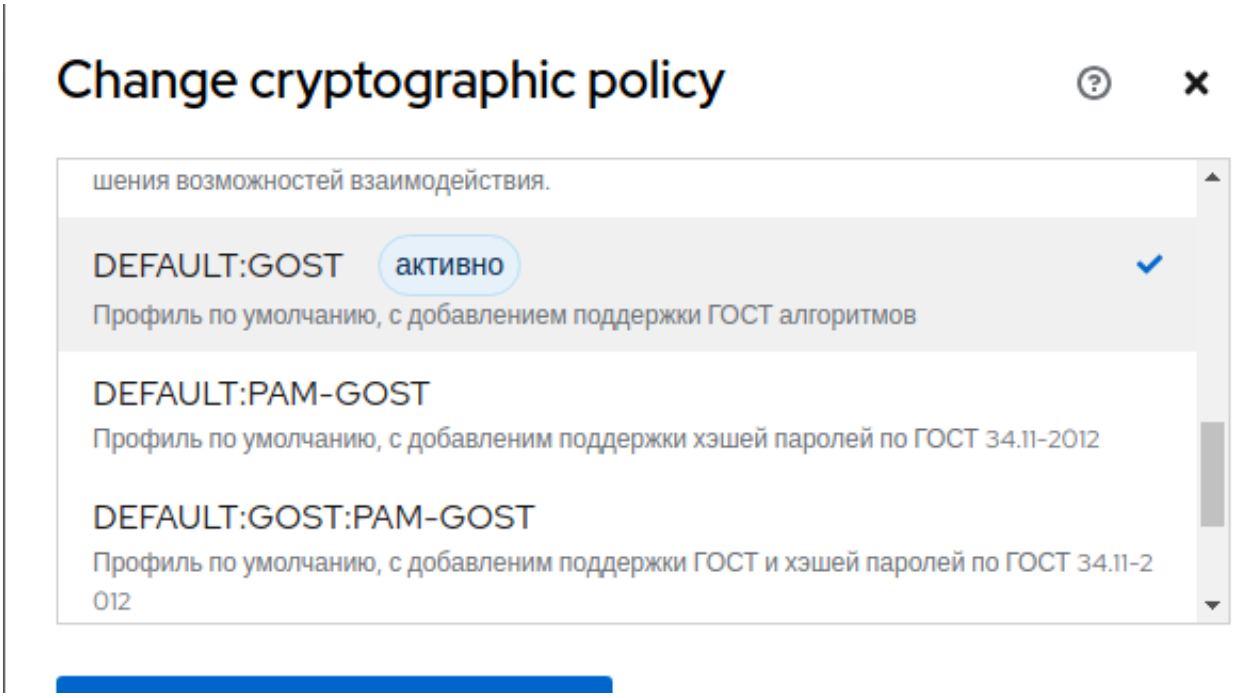
1. Откройте веб-интерфейс приложения по адресу `http://localhost:9090`.
2. Войдите под своей учётной записью с правами администратора.



3. Выберите пункт меню **Обзор** и перейдите в окно **Настройка**. В этом окне нажмите на поле **Cryptographic policy**.



4. Откроется окно выбора политики, где показана текущая политика и можно выбрать одну из политик ГОСТ, нажав на неё.



Пример генерации сертификата с подключённым ГОСТ

1. Проверьте, что используется политика с поддержкой ГОСТ:

```
$ update-crypto-policies --show
  DEFAULT:GOST
```

   Если установлена какая-либо другая политика, установите политику с поддержкой ГОСТ с помощью инструкции *Применение политики с поддержкой ГОСТ из командной строки*.

2. Генерация закрытого ключа и создание сертификата с подписью ГОСТ выполняется следующей командой:

```
$ openssl req -x509 -newkey gost2012_256 -pkeyopt paramset:A -nodes -keyout key.pem -out␣
↪cert.pem -md_gost12_256
```

3. Проверьте сгенерированный сертификат:

```
$ openssl x509 -in cert.pem -text -noout | grep GOST
     Signature Algorithm: GOST R 34.10-2012 with GOST R 34.11-2012 (256 bit)
         Public Key Algorithm: GOST R 34.10-2012 with 256 bit modulus
 Signature Algorithm: GOST R 34.10-2012 with GOST R 34.11-2012 (256 bit)
```

   То есть файл `cert.pem` должен существовать и вывод `grep` не должен быть пустым.

Хранение аутентификационной информации пользователей с использованием хеш-функций по ГОСТ Р 34.11-2012

При использовании политики `GOST-ONLY-PAM` или подключении к существующей политике модулей `PAM-GOST`, `SSSD-PAM-GOST` или `PATCH-PAM-GOST` хеширование паролей пользователей производится с использованием алгоритма ГОСТ Р 34.11-2012.

Убедиться в этом можно с помощью команды `passwd -S`.

Если пароль был установлен при активной политике `DEFAULT`, вывод команды будет следующий:

```
$ sudo passwd -S dev1
dev1 PS 2025-04-21 0 99999 7 -1 (Пароль задан, шифр SHA512)
```

После изменения политики и смены пароля вывод команды будет следующий:

```
$ sudo passwd -S dev1
dev1 PS 2025-04-21 0 99999 7 -1 (Пароль задан, шифр GOST Yescrypt)
```

Убедиться в том, что используется алгоритм ГОСТ Р 34.11-2012, также можно просмотрев файл `/etc/shadow`.

Зашифрованный пароль в `/etc/shadow` выглядит следующим образом:

```
пользователь: $идентификатор-алгоритма$соль$хэш
```

Где `идентификатор-алгоритма` — наименование алгоритма шифрования:

- `$5$` — SHA-256;
- `$6$` — SHA-512;
- `$gy$` — ГОСТ Р 34.11-2012.

Выполнив команду `sudo cat /etc/shadow | grep dev1` до и после применения политики и смены пароля можно убедиться, что поле идентификатора алгоритма меняется с `$6$` на `$gy$`.

До:

```
dev1:$6$ts9u0la8BdTaGaeK$...
```

После:

```
gostuser:$gy$j9T$...
```

Установка веб-браузера Chromium ГОСТ

Некоторые сайты, предоставляющие государственные услуги, используют шифрование по алгоритму ГОСТ. Для доступа к таким сайтам требуется установить программу КриптоПро CSP и веб-браузер, который проверяет, установлена ли она. https://gost.cryptopro.ru

1. Для установки браузера Chromium ГОСТ установите пакет `chromium-gost` с помощью следующей команды (от имени администратора):

   ```
   $ sudo dnf install chromium-gost
   ```

2. Скачайте с сайта КриптоПро последнюю версию программы КриптоПро CSP (https://www.cryptopro.ru/products/csp). Выбирайте версию для **Linux RPM**.

3. Распакуйте архив и запустите скрипт установки:

   ```
   $ cd Загрузки
   $ tar xavf linux-amd64.tgz
   $ cd linux-amd64
   $ sudo ./install.sh
   ```

4. Перезагрузите компьютер или запустите программу КриптоПро CSP следующей командой (от имени администратора):

```
$ sudo /etc/init.d/cprocsp start
```

5. Для проверки работы нового криптопровайдера запустите браузер Chromium ГОСТ и зайдите на страницу https://gost.cryptopro.ru. Параметр `ssl_cipher` должен быть GOST.

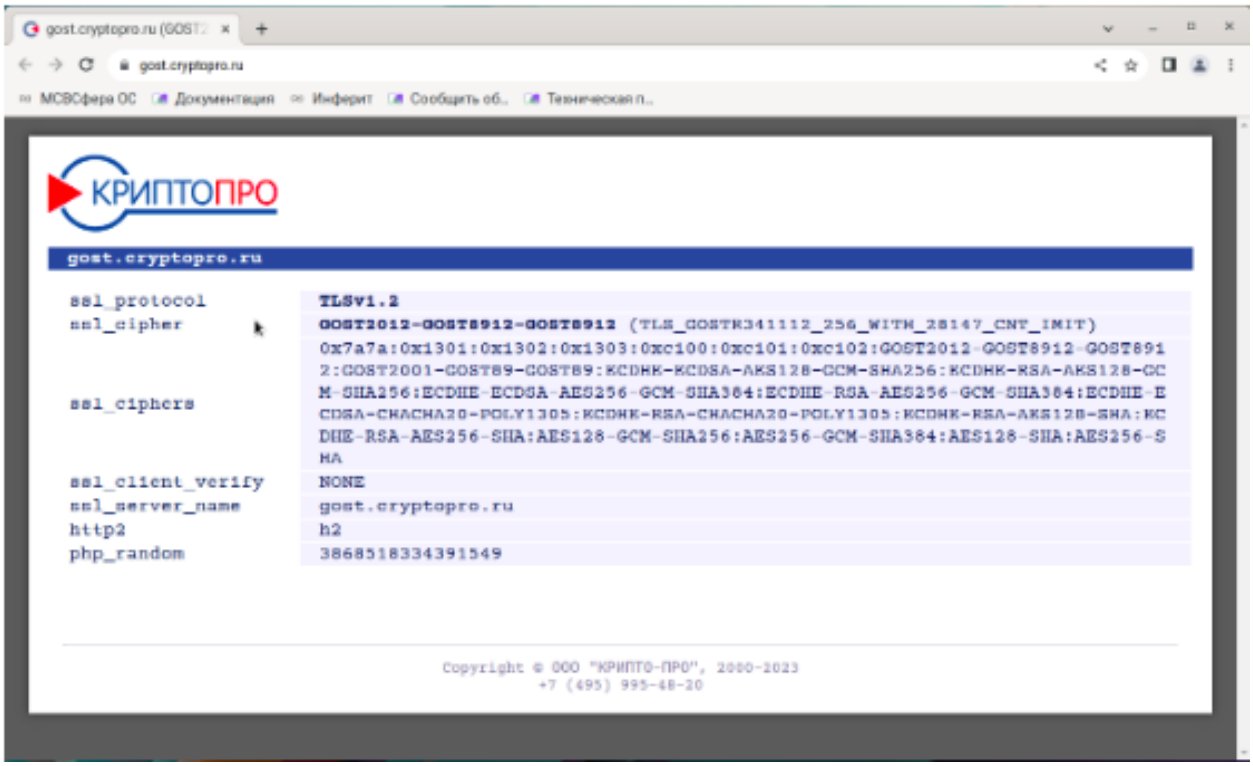


Рис. 3: Параметр `ssl_cipher` — GOST

Установка обновлений

МСВСфера ОС 9

Перед установкой обновления убедитесь, что все ранее выпущенные исправления, относящиеся к системе, установлены.

В графическом интерфейсе

1. Откройте «Центр приложений» и перейдите во вкладку «Обновления».
2. Нажмите соответствующие кнопки для загрузки обновлений, а также обновления системы и пакетов.

Примечание

Чтобы открыть «Центр приложений» из «Терминала», введите команду `gnome-software`.

В «Терминале»

Для обновления операционной системы используйте следующую команду:

```
$ sudo dnf upgrade
```

Для установки определённого пакета, например `vsftpd`, используйте следующую команду:

```
$ sudo dnf install vsftpd
```

Для обновления определённого пакета, например `bind`, используйте следующую команду:

```
$ sudo dnf upgrade bind
```

Для вывода дополнительной информации о параметрах командной строки, доступных для `dnf`, используйте следующую команду:

```
$ man dnf
```

МСВСфера ОС 8

Перед установкой обновления убедитесь, что все ранее выпущенные исправления, относящиеся к системе, установлены.

В графическом интерфейсе

1. Откройте «Центр приложений» и перейдите во вкладку «Обновления».
2. Нажмите соответствующие кнопки для загрузки обновлений, а также обновления системы и пакетов.

Примечание

Чтобы открыть «Центр приложений» из «Терминала», введите команду `gnome-software`.

В «Терминале»

Для обновления операционной системы используйте следующую команду:

```
$ sudo dnf upgrade
```

Для установки определённого пакета, например `vsftpd`, используйте следующую команду:

```
$ sudo dnf install vsftpd
```

Для обновления определённого пакета, например `bind`, используйте следующую команду:

```
$ sudo dnf upgrade bind
```

Для вывода дополнительной информации о параметрах командной строки, доступных для `dnf`, используйте следующую команду:

```
$ man dnf
```

База данных уязвимостей МСВСфера ОС

Для защиты системы от текущих и будущих угроз требуется выполнять регулярные обновления. В инструкции ниже вы найдёте информацию, необходимую для выполнения рекомендаций по обеспечению безопасности.

Бюллетени по безопасности

В базе данных уязвимостей МСВСфера ОС содержится информация об устранении уязвимостей в продуктах МСВСфера ОС.

База доступна для всех пользователей МСВСфера ОС по ссылке: https://errata.msvsphere-os.ru/

На главной странице вы увидите список бюллетеней в хронологическом порядке. Для каждого бюллетеня доступно описание всех устранённых уязвимостей, а также его тип (устранение уязвимостей, устранение недостатков и улучшение) и версия системы, для которой он применим.

При желании вы можете отфильтровать результаты по версии операционной системы или воспользоваться поиском (по идентификатору уязвимости или бюллетеня, а также по ключевым словам).

МСВСфера

МСВСфера 8 | МСВСфера 9 | Показать всё | Поиск | Найти | Язык: Русский

Бюллетень	Описание	Тип	Версия
INFSA-2024:8127	CVE-2023-48161 Уязвимость функции DumpSCreen2RGB в GifLib вызвана переполнением буфера на основе кучи. Эксплуатация уязвимости может позволить локальному нарушителю получить конфиденциальную информацию и использовать её для запуска дальнейших атак на уязвимую систему, убедив жертву открыть специально созданный файл. CVE-2024-21208 Уязвимость компонента Networking в Oracle Java SE, Oracle GraalVM для JDK, Oracle GraalVM Enterprise Edition (уязвимые версии: Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4, 23; Oracle GraalVM для JDK: 17.0.12, 21.0.4, 23; Oracle GraalVM Enterprise Edition: 20.3.15 и 21.3.11) связана с неправильной обработкой maxHeaderSize клиентом HTTP. Эксплуатация уязвимости может позволить нарушителю скомпрометировать систему и вызвать частичный отказ в обслуживании. CVE-2024-21210 Уязвимость компонента Hotspot в Oracle Java SE (уязвимые версии Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4 и 23.) связана с целочисленным переполнением при индексации массива. Эксплуатация уязвимости может позволить нарушителю получить несанкционированный доступ к данным. Показать всё...	устранение уязвимостей	8
INFSA-2024:8121	CVE-2023-48161 Уязвимость функции DumpSCreen2RGB в GifLib вызвана переполнением буфера на основе кучи. Эксплуатация уязвимости может позволить локальному нарушителю получить конфиденциальную информацию и использовать её для запуска дальнейших атак на уязвимую систему, убедив жертву открыть специально созданный файл. CVE-2024-21208 Уязвимость компонента Networking в Oracle Java SE, Oracle GraalVM для JDK, Oracle GraalVM Enterprise Edition (уязвимые версии: Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4, 23; Oracle GraalVM для JDK: 17.0.12, 21.0.4, 23; Oracle GraalVM Enterprise Edition: 20.3.15 и 21.3.11) связана с неправильной обработкой maxHeaderSize клиентом HTTP. Эксплуатация уязвимости может позволить нарушителю скомпрометировать систему и вызвать частичный отказ в обслуживании. CVE-2024-21210 Уязвимость компонента Hotspot в Oracle Java SE (уязвимые версии Oracle Java SE: 8u421, 8u421-perf, 11.0.24, 17.0.12, 21.0.4 и 23.) связана с целочисленным переполнением при индексации массива. Эксплуатация уязвимости может позволить нарушителю получить несанкционированный доступ к данным. Показать всё...	устранение уязвимостей	9

Для получения подробной информации о бюллетене, нажмите на его номер.

INFSA-2024:7262: Устранение уязвимостей osbuild-composer

Информация о бюллетене

Идентификатор: INFSA-2024:7262

Тип: устранение уязвимостей

Дата публикации: 2024-10-10 10:47:44 UTC

Информация о пакете

Служба для создания настраиваемых артефактов операционной системы, таких как образы ВМ и коммиты OSTree, которая использует osbuild. Помимо создания образов для локального использования, она также может загружать образы напрямую в облако. Совместима с клиентами composer-cli и cockpit-composer.

Описание уязвимостей

- **CVE-2024-1394**
  Уязвимость языка программирования Golang связана с ошибками освобождения памяти в коде шифрования/дешифрования RSA. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, вызвать отказ в обслуживании с помощью специально созданных данных.
- **CVE-2024-34156**
  Уязвимость функции Decoder.Decode языка программирования Go связана с неконтролируемой рекурсией. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, вызвать отказ в обслуживании.

Уровень опасности

CVE	Оценка CVSS 2.0	Оценка CVSS 3.x	Оценка CVSS 4.0
NIST — CVE-2024-1394	нет информации	7,5	нет информации
NIST — CVE-2024-34156	нет информации	7,5	нет информации

Критичный, средний, некритичный

Обновлённые пакеты

Пакет	Архитектура
▼ osbuild-composer-101-2.el8_10.inferit	
osbuild-composer-0:101-2.el8_10.inferit	x86_64
osbuild-composer-core-0:101-2.el8_10.inferit	x86_64
osbuild-composer-worker-0:101-2.el8_10.inferit	x86_64

Получение данных о бюллетенях в «Терминале»

В «Терминале» вы можете просматривать данные о бюллетенях и пакетах, содержащих устренённые уязвимости, с помощью утилиты `dnf`.

Показать список всех доступных бюллетеней и пакетов

```
$ sudo dnf updateinfo list updates security
...
INFSA-2024:6356 Важные/Безоп.    bubblewrap-0.4.1-7.el9_4.x86_64
INFSA-2024:3842 Низкие/Безоп.    c-ares-1.19.1-2.el9_4.x86_64
INFSA-2024:3843 Умеренные/Безоп. cockpit-311.2-1.el9_4.inferit.x86_64
...
```

Показать список всех установленных бюллетеней и пакетов

```
$ sudo dnf updateinfo list security --installed
...
INFSA-2024:2433 Умеренные/Безоп. avahi-libs-0.8-20.el9.x86_64
INFSA-2024:2551 Важные/Безоп.    bind-libs-32:9.16.23-18.el9_4.1.x86_64
INFSA-2024:5231 Важные/Безоп.    bind-libs-32:9.16.23-18.el9_4.6.x86_64
...
```

Если было установлено несколько обновлений для одного пакета, `dnf` покажет список всех бюллетеней для этого пакета. В примере для пакета `bind-libs` были установлены два обновления.

Показать содержимое бюллетеня

Если вы знаете номер бюллетеня, вы также можете просмотреть его содержимое в «Терминале». Например, чтобы показать содержимое бюллетеня с номером `INFSA-2024:2377`, необходимо выполнить следующую команду:

```
$ sudo dnf updateinfo info INFSA-2024:2377
Последняя проверка окончания срока действия метаданных: 1:32:45 назад, Чт 07 ноя 2024 14:35:08.
==============================================================================
INFSA-2024:2377: zziplib security update
==============================================================================
Идентификатор обновления: INFSA-2024:2377
                    Тип: безопасность
               Обновлено: 2024-08-13 15:35:36
        Уязвимости (CVE): CVE-2020-18770
                Описание: The zziplib is a lightweight library to easily extract data from zip␣
↪files.
                        :
                        : * CVE-2020-18770
                        : An invalid memory access flaw was found in the mmapped.c
                        files zzip_disk_entry_to_file_header function in Zziplib.
                        This issue could allow an attacker to entice a victim into opening a
                        specially crafted file, leading to a denial of service.
               Опасность: Moderate
              Установлен: истина
```

Установка пакетов с устранёнными уязвимостями в «Терминале»

В общем случае все пакеты с устранёнными уязвимостями устанавливаются автоматически во время обновления операционной системы. При необходимости вы можете установить все доступные пакеты с устранёнными уязвимостями, не дожидаясь обновления системы или же установить только пакеты, входящие в определённый бюллетень. А также настроить систему таким образом, чтобы она автоматически загружала и устанавливала все пакеты с устранёнными уязвимостями.

Автоматическая загрузка и установка всех пакетов с устранёнными уязвимостями

Для настройки автоматической загрузки и установки всех пакетов с устранёнными уязвимостями необходимо, чтобы был установлен пакет `dnf-automatic`.

1. Откройте файл `/etc/dnf/automatic.conf` и убедитесь, что в разделе `[commands]` для параметра `upgrade_type` задано значение `default` или `security`:

```
[commands]
#  What kind of upgrade to perform:
# default                            = all available upgrades
# security                           = only the security upgrades
upgrade_type = security
```

2. Включите и запустите модуль таймера `systemd`:

```
$ sudo systemctl enable --now dnf-automatic-install.timer
```

3. Убедитесь, что таймер включён:

```
$ sudo systemctl status dnf-automatic-install.timer
```

Установка всех доступных пакетов с устранёнными уязвимостями

1. Установите все пакеты с устранёнными уязвимостями с помощью утилиты `dnf`:

```
$ sudo dnf update --security
```

   При использовании параметра `--security` будут установлены только пакеты, содержащие устранённые уязвимости. Без этого параметра будет выполнено полное обновление системы.

2. Подтвердите и начните установку, нажав `Д`.

3. Необязательно: вы можете вывести на экран список процессов, требующих ручного перезапуска системы после установки обновленных пакетов. Например:

```
$ sudo dnf needs-restarting
1107 : /usr/sbin/rsyslogd -n
1199 : -bash
```

   Обратите внимание, что перечислены только **процессы**, требующие перезапуска, а не службы. То есть вы не сможете перезапустить перечисленные процессы с помощью утилиты `systemctl`. Например, процесс `bash` в выводе завершится, когда пользователь, которому принадлежит этот процесс, выйдет из системы.

Установка пакетов с устранёнными уязвимостями, входящих в определённый бюллетень

В некоторых ситуациях может потребоваться установить только определённые обновления. Например, если службу можно обновить без планового простоя, вы можете установить пакеты с устранёнными уязвимостями только для этой службы, а остальные обновления установить позже.

1. Установите все пакеты, входящие в определённый бюллетень, например:

```
$ sudo dnf update --advisory=INFSA-2024:7262
```

2. Чтобы применить обновление с минимальным изменением версии, используйте команду `dnf upgrade-minimal`, например:

```
$ sudo dnf upgrade-minimal --advisory=INFSA-2024:7262
```

3. Подтвердите и начните установку, нажав `Д`.

4. Необязательно: вы можете вывести на экран список процессов, требующих ручного перезапуска системы после установки обновленных пакетов. Например:

```
$ sudo dnf needs-restarting
1107 : /usr/sbin/rsyslogd -n
1199 : -bash
```

   Обратите внимание, что перечислены только **процессы**, требующие перезапуска, а не службы. То есть вы не сможете перезапустить перечисленные процессы с помощью утилиты `systemctl`. Например, процесс `bash` в выводе завершится, когда пользователь, которому принадлежит этот процесс, выйдет из системы.

Уровни опасности уязвимостей

Мы оцениваем серьёзность уязвимостей, обнаруженных в продуктах МСВСфера, по четырёхбалльной шкале:

- критический уровень опасности;
- высокий уровень опасности;
- средний уровень опасности;
- низкий уровень опасности.

Четырёхбалльная шкала показывает, насколько серьезной считается уязвимость, помогая вам оценить степень риска и определить наиболее важные обновления.

Шкала учитывает потенциальный риск на основе технического анализа уязвимости и её типа. Уровень опасности уязвимости не меняется, если позже для неё выпущен

программный код или модуль, используемый для атак на узлы, содержащие уязвимости («эксплойт» или «червь»). Уязвимость будет переоценена только в случае появления новых технических подробностей.

Уровни опасности в МСВСфера соответствуют классификации Common Vulnerability Scoring System v3.0 (CVSSv3), которая совпадает с классификацией Red Hat. Эти системы обеспечивают оценку опасности, позволяя вам принимать обоснованные решения о риске, который каждая уязвимость представляет для вашей уникальной среды.

Все бюллетени по безопасности МСВСфера вы можете найти в базе данных уязвимостей МСВСфера ОС, а также получить данные о бюллетенях в «Терминале». Для каждого бюллетеня доступно описание всех устранённых уязвимостей, а также его тип (устранение уязвимостей, устранение недостатков и улучшение), версия системы, для которой он применим, и список пакетов и архитектур. Каждая уязвимость в бюллетене имеет свой уровень опасности.

Классификация уровней опасности уязвимостей в МСВСфера ОС

Рассмотрим более подробно каждый уровень опасности.

Критический уровень

- Диапазон значений: **9,0 — 10,0**.
- Название:

МСВСфера ОС	Red Hat	CVSSv3
Критичный	Critical	Critical

- Описание: присваивается уязвимостям, которые могут быть легко использованы удалённым неаутентифицированным нарушителем и привести к компрометации системы (выполнению произвольного кода) без необходимости взаимодействия с пользователем или легко вызвать компрометацию системы через конечные точки вывода в системах с ИИ. Уязвимости, требующие аутентификации, локального или физического доступа к системе или маловероятной конфигурации, не классифицируются как критические (эти типы уязвимостей могут быть использованы «червями»).

Высокий уровень

- Диапазон значений: **7,0 — 8,9**.
- Название:

МСВСфера ОС	Red Hat	CVSSv3
Высокий	Important	High

- Описание: присваивается уязвимостям, которые могут легко поставить под угрозу конфиденциальность, целостность или доступность данных. Это такие типы уязвимостей, которые позволяют локальным или аутентифицированным нарушителям получать дополнительные привилегии, позволяют неаутентифицированным удалённым нарушителям получать доступ к защищаемым ресурсам или другими элементами управления, позволяют аутентифицированным удалённым нарушителям выполнять произвольный код, позволяют удалённым нарушителям вызывать отказ в обслуживании или могут скомпрометировать систему.

Средний уровень

- Диапазон значений: **4,0 — 6,9**.
- Название:

МСВСфера ОС	Red Hat	CVSSv3
Средний	Moderate	Medium

- Описание: присваивается уязвимостям, которые может быть сложно использовать, но которые всё равно могут привести к некоторому нарушению конфиденциальности, целостности или доступности данных при определённых обстоятельствах. Это такие типы уязвимостей, которые могли бы иметь иметь критический или важный уровень, но их сложнее использовать и/или конфигурация системы для их использования маловероятна.

Низкий уровень

- Диапазон значений: **0,1 — 3,9**.
- Название:

МСВСфера ОС	Red Hat	CVSSv3
Низкий	Low	Low

- Описание: присваивается уязвимостям, эксплуатация которых маловероятна, или же успешная эксплуатация которых приведёт к минимальным последствиям. Может включать ошибки в исходном коде, для которых не существует текущих или теоретически возможных векторов эксплуатации или которые были обнаружены в ходе технического анализа кода.

БДУ ФСТЭК

БДУ ФСТЭК — банк данных угроз безопасности информации Федеральной службы по техническому и экспортному контролю — отечественная база уязвимостей, содержащая все уязвимости и способы их устранения. В БДУ ФСТЭК используется похожая шкала для оценки серьёзности уязвимости.

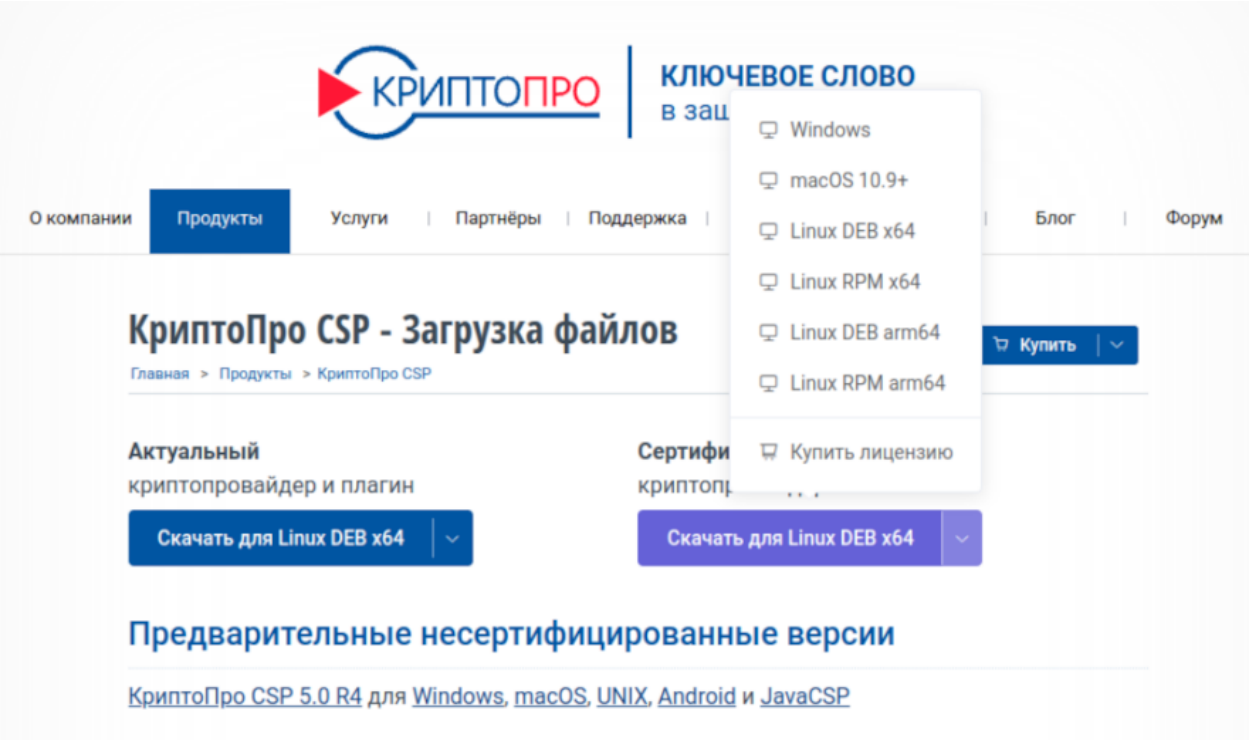
Установка КриптоПро CSP и расширения ЭЦП для браузера

В данной инструкции рассматривается установка криптопровайдера КриптоПро CSP, расширения ЭЦП для браузера, а также получение личного сертификата тестового удостоверяющего центра.

Установка КриптоПро CSP

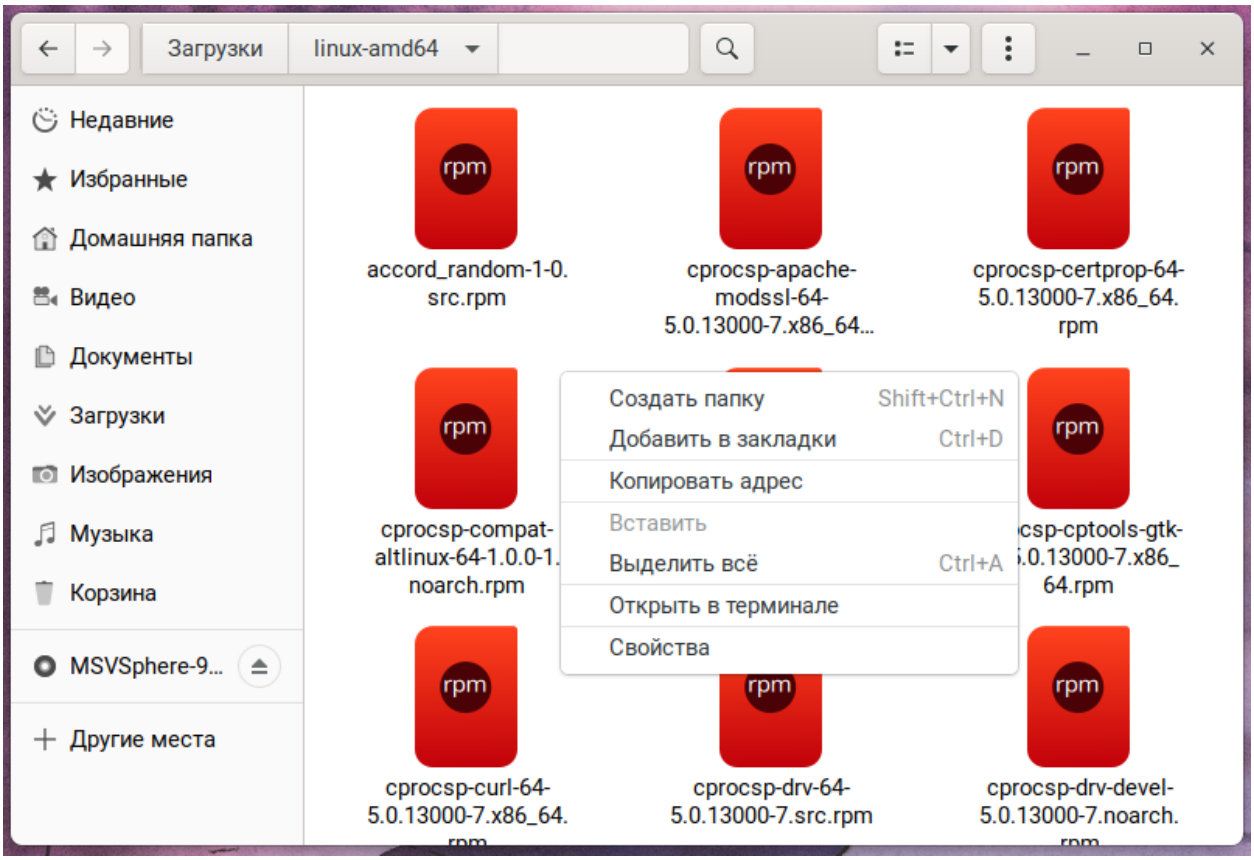
Перед скачиванием и установкой КриптоПро CSP вам необходимо создать учётную запись на сайте https://cryptopro.ru/user/register.

1. Загрузите актуальную сертифицированную версию КриптоПро CSP (https://cryptopro.ru/products/csp/downloads), при скачивании выберите «Linux RPM x64». В результате на ваш компьютер будет загружен архив с именем `linux-amd4.tgz`.



2. Перейдите в папку, в которую вы скачали архив с именем `linux-amd4.tgz` из шага 1 (обычно это папка «Загрузки»), и распакуйте его двойным щелчком мыши.

3. Перейдите в папку с именем «linux-amd4», нажмите правой кнопкой мыши на свободном пространстве и выберете пункт «Открыть в терминале».

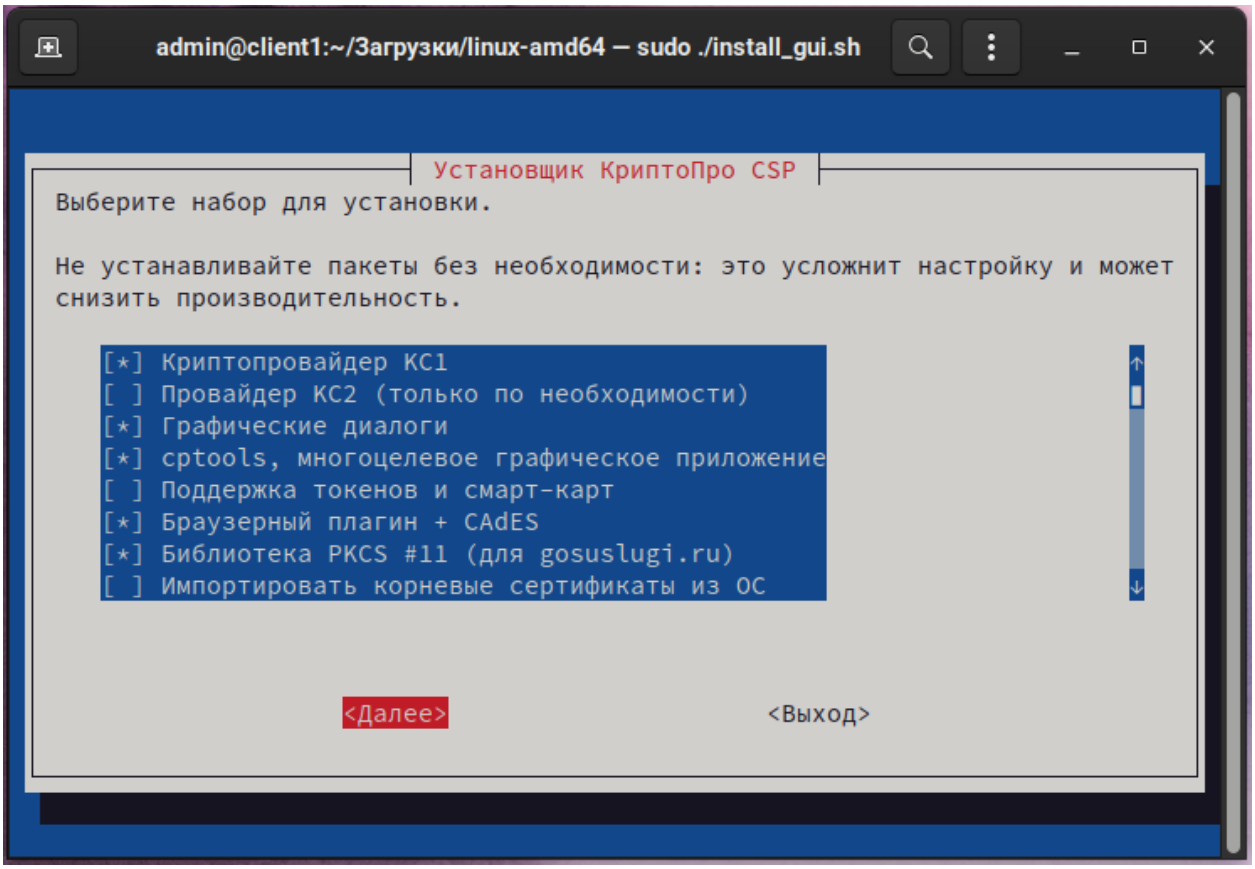


4. В открывшемся окне «Терминала» введите следующую команду:

```
$ sudo ./install_gui.sh
```

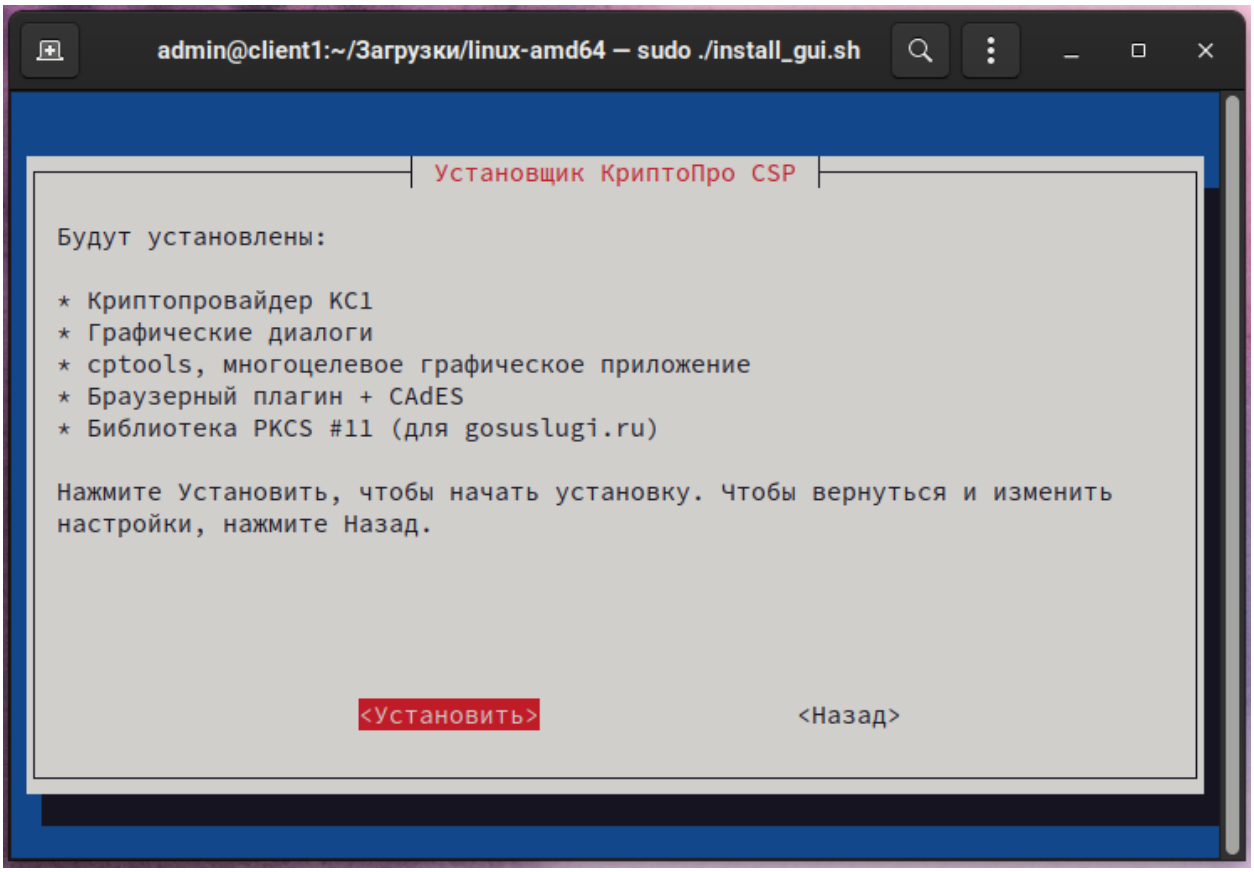
5. Откроется мастер установки КриптоПро CSP. Навигация по мастеру установки производится при помощи клавиши «Tab». В открывшемся окне нажмите выделенную по умолчанию кнопку «Далее» с помощью клавиши «Enter» («Ввод») на клавиатуре.

6. Откроется форма выбора устанавливаемых компонентов.



Рекомендуется выбрать предлагаемый по умолчанию набор компонентов. Затем перейдите к следующему этапу, нажав кнопку «Далее» с помощью клавиши «Enter» («Ввод») на клавиатуре.

7. Проверьте список устанавливаемых компонентов и нажмите кнопку «Установить» с помощью клавиши «Enter» («Ввод») на клавиатуре.

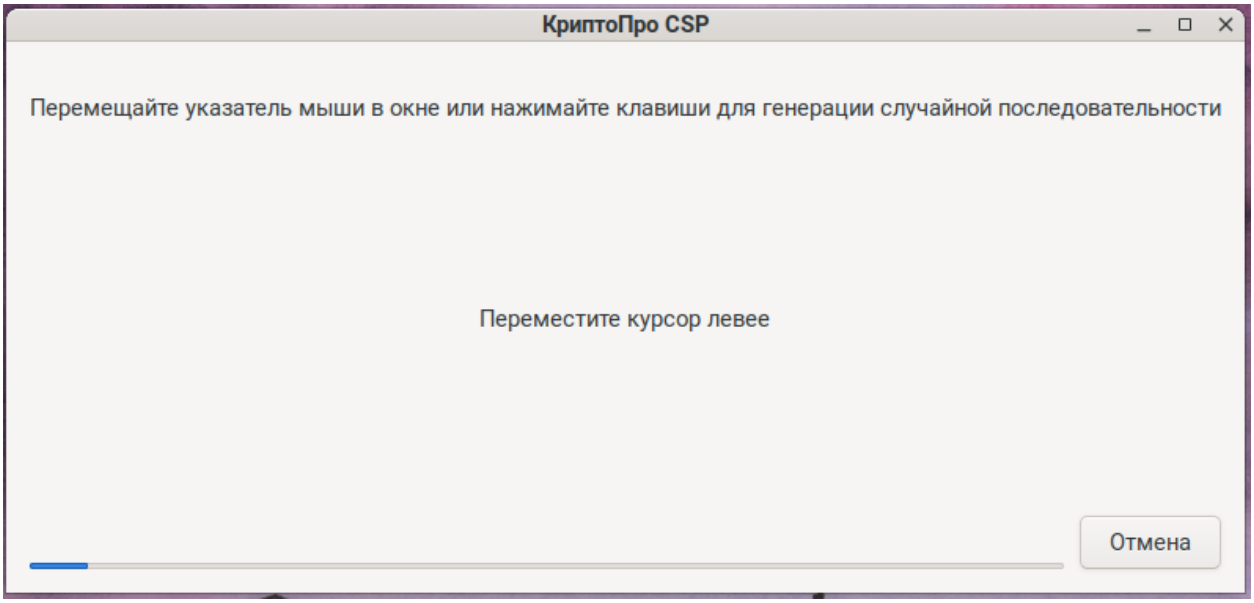


8. В окне «Терминала» подтвердите установку, нажав клавишу «д» или «у» на клавиатуре, а затем нажмите «Enter» («Ввод»):

```
admin@client1:~/Загрузки/linux-amd64 — sudo ./install_gui.sh
================================================================================
 Пакет                      Архитектура
                                        Версия          Репозиторий      Размер
================================================================================
Установка:
 cprocsp-cptools-gtk-64     x86_64     5.0.13000-7     @commandline     2.0 M
 cprocsp-curl-64            x86_64     5.0.13000-7     @commandline     356 k
 cprocsp-pki-cades-64       x86_64     2.0.15000-1     @commandline      13 M
 cprocsp-pki-plugin-64      x86_64     2.0.15000-1     @commandline     659 k
 cprocsp-rdr-gui-gtk-64     x86_64     5.0.13000-7     @commandline     103 k
 lsb-cprocsp-base           noarch     5.0.13000-7     @commandline     183 k
 lsb-cprocsp-ca-certs       noarch     5.0.13000-7     @commandline      20 k
 lsb-cprocsp-capilite-64    x86_64     5.0.13000-7     @commandline     1.9 M
 lsb-cprocsp-kc1-64         x86_64     5.0.13000-7     @commandline     910 k
 lsb-cprocsp-pkcs11-64      x86_64     5.0.13000-7     @commandline     182 k
 lsb-cprocsp-rdr-64         x86_64     5.0.13000-7     @commandline     431 k

Результат транзакции
================================================================================
Установка  11 Пакетов

Общий размер: 20 M
Объем изменений: 90 M
Продолжить? [д/Н]:
```

9. После завершения установки пакетов вам будет предложено ввести лицензию. Нажмите кнопку «Позже», а на следующем экране – «Выход».

10. Откроется окно генерации случайной последовательности. Перемещайте указатель мыши по этому окну пока индикатор прогресса не дойдёт до конца.



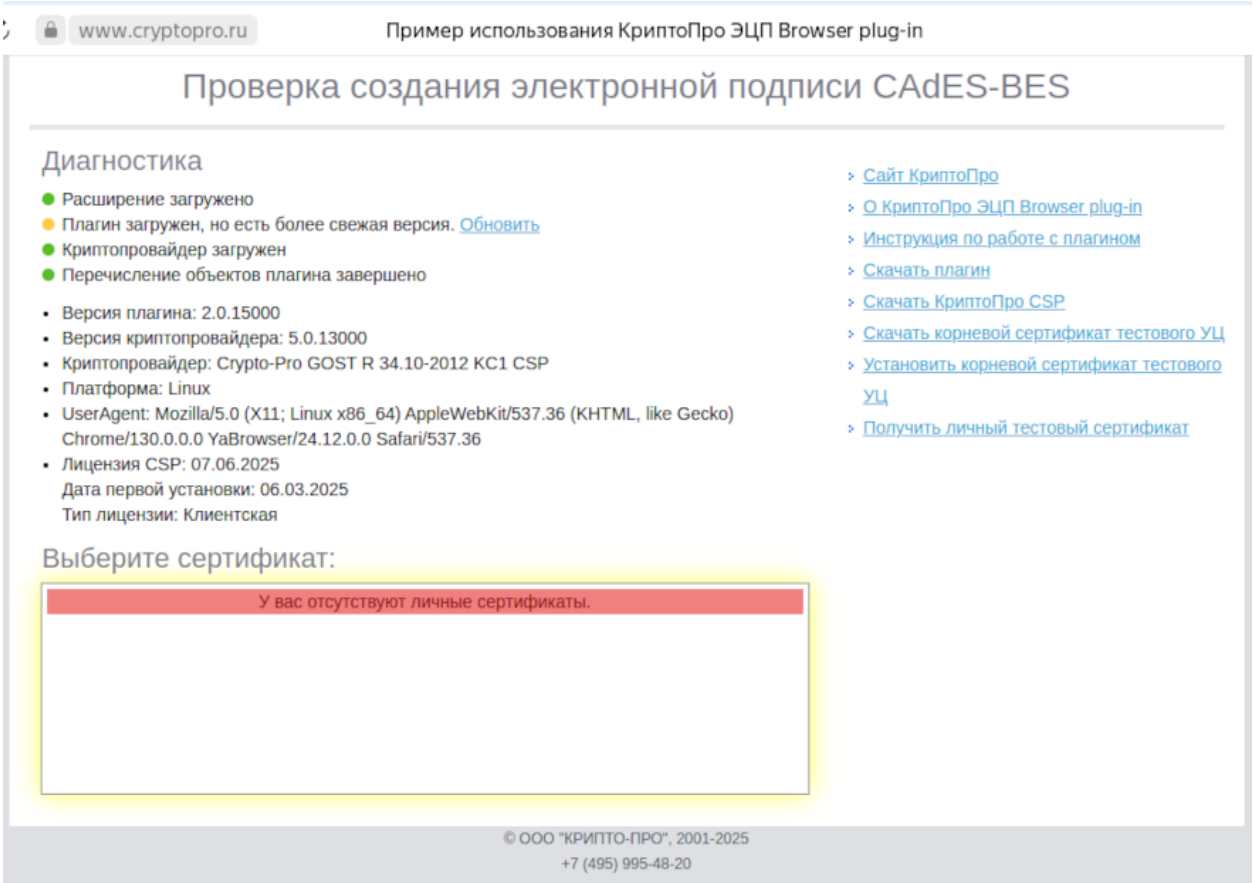
На этом процедура установки КриптоПро CSP завершена.

Установка и проверка расширения для браузера

Установите расширение в зависимости от используемого браузера. Варианты установки расширения для различных браузеров описаны в базе знаний КриптоПро: https://support.cryptopro.ru/index.php?/Knowledgebase/Article/View/467/0/ustnovk-rsshirenija-dlja-plgin-iz-fjjl.

После установки расширения необходимо проверить его работу. Для этого перейдите по ссылке: https://www.cryptopro.ru/sites/default/files/products/cades/demopage/cades_bes_sample.html.

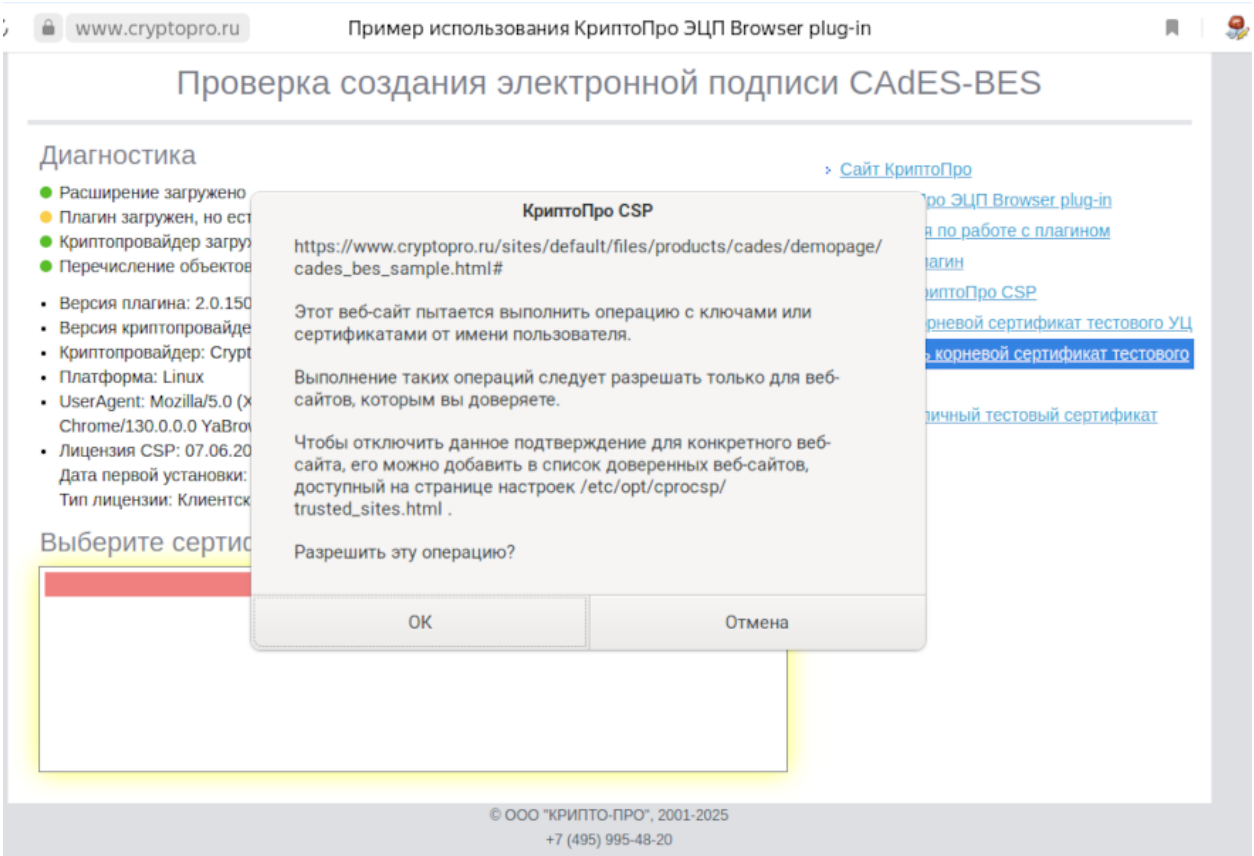
Если установка расширения выполнена успешно, то после подтверждения использования веб-страница будет выглядеть следующим образом.



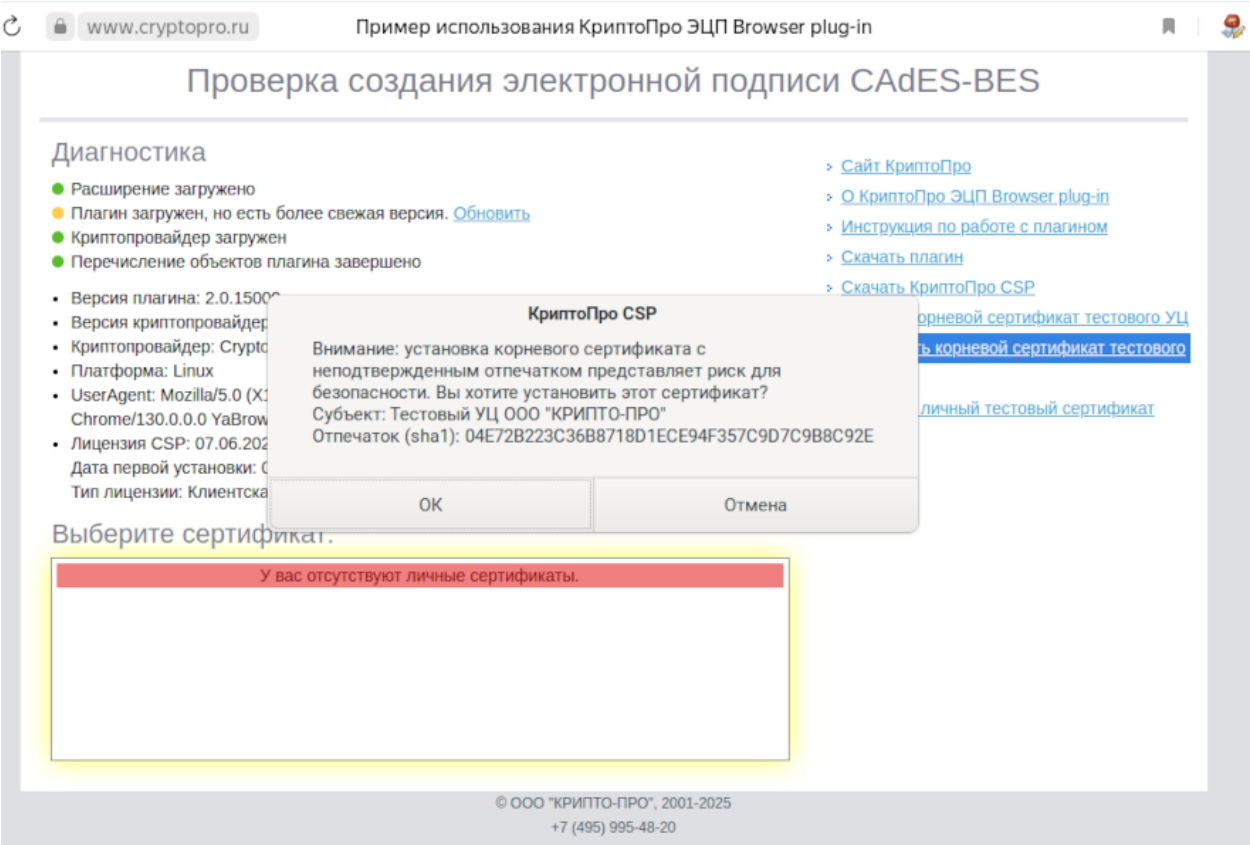
В идеальном случае все пункты в разделе «Диагностика» должны подсвечиваться зелёным. На рисунке выше вы можете увидеть предупреждение о том, что доступна более свежая версия плагина, это не является препятствием для дальнейших действий.

Следующий шаг – установка корневого сертификата тестового УЦ.

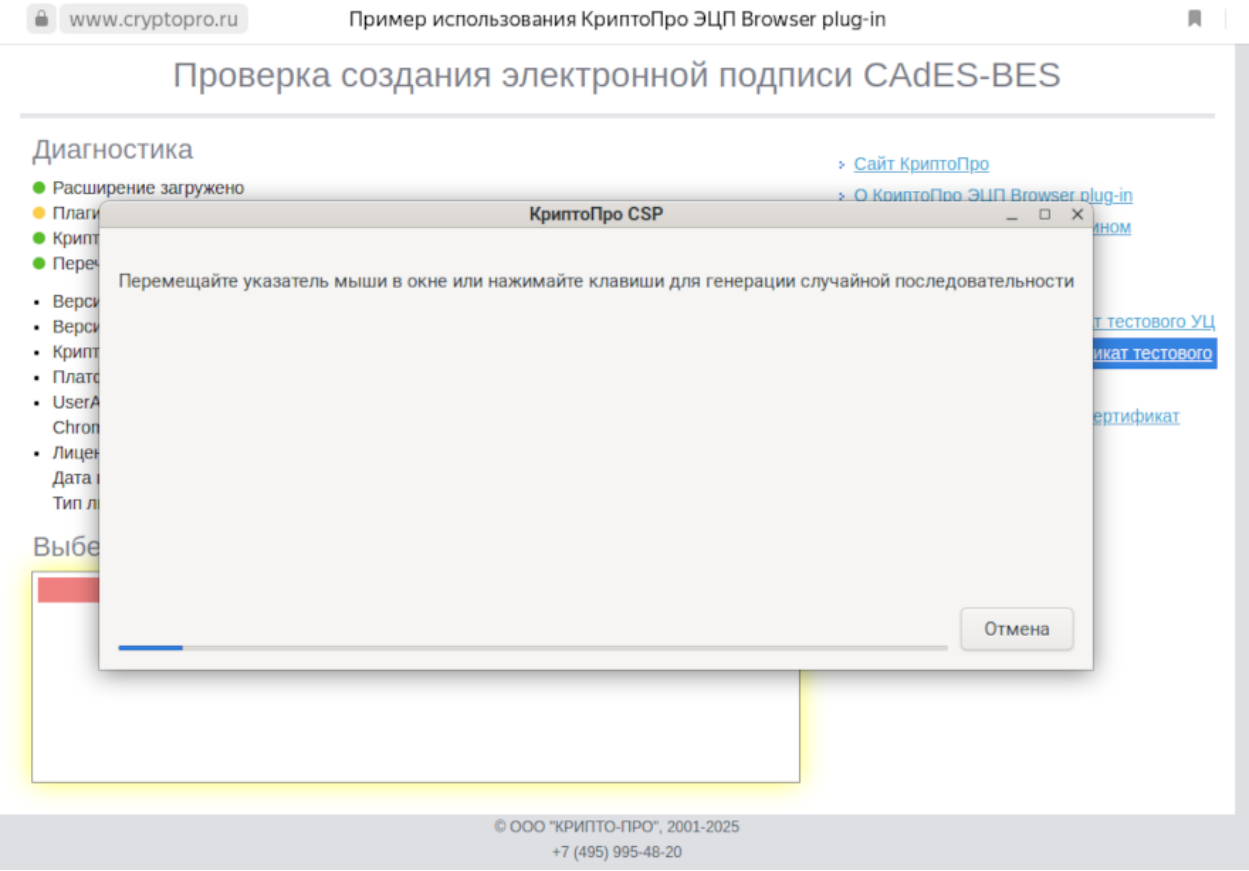
1. Выберите пункт меню «Установить корневой сертификат тестового УЦ» справа.



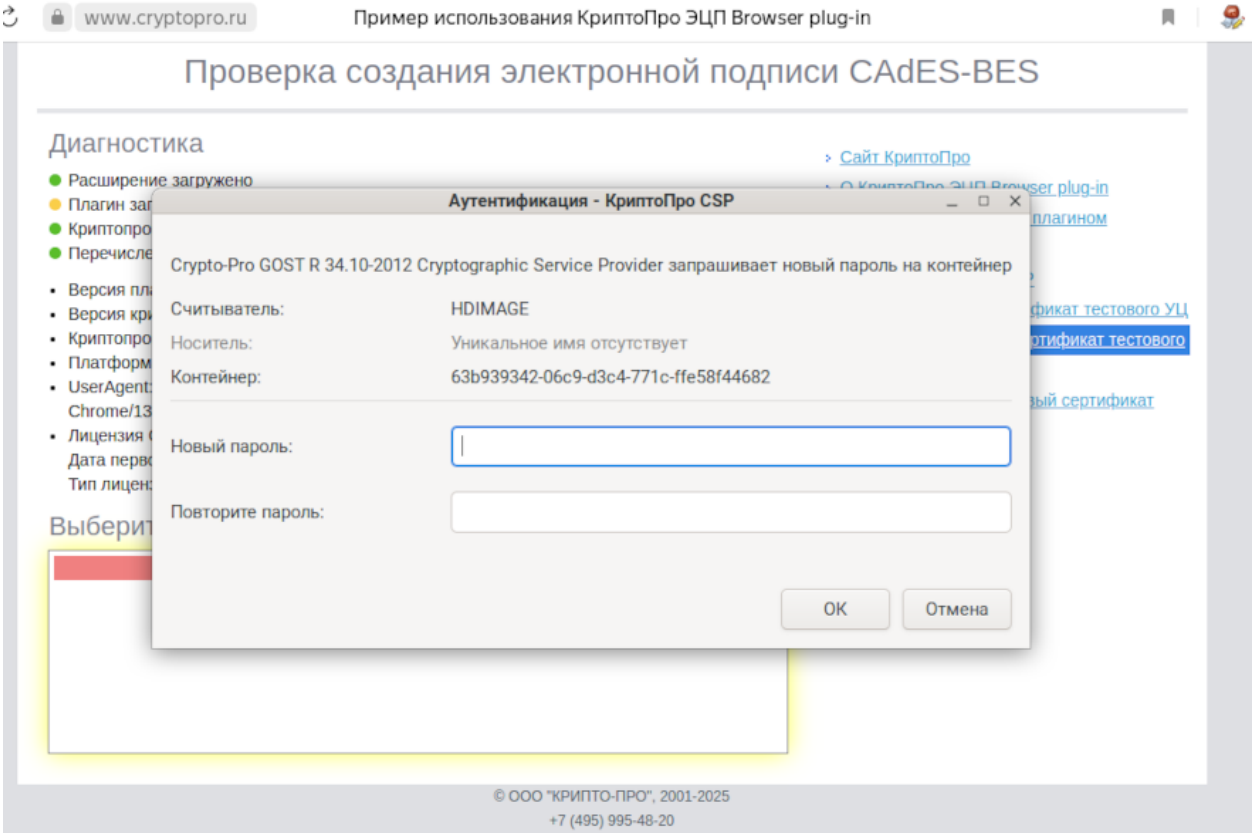
2. Нажмите «OK», чтобы разрешить операцию.
3. А затем снова «OK» для начала установки корневого сертификата.



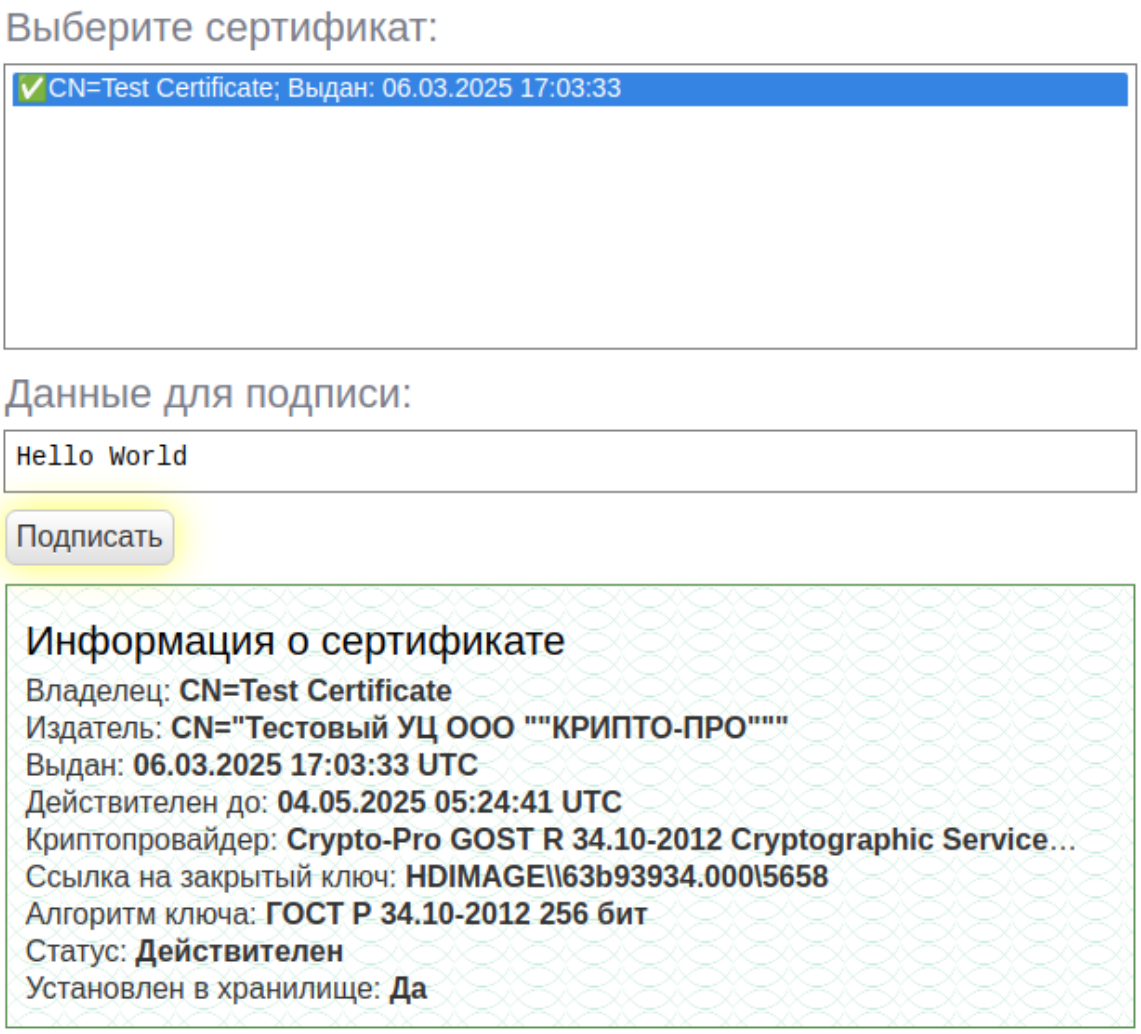
4. Следующий шаг – получение личного тестового сертификата. Для этого снова перемещайте указатель мыши для генерации случайной последовательности.



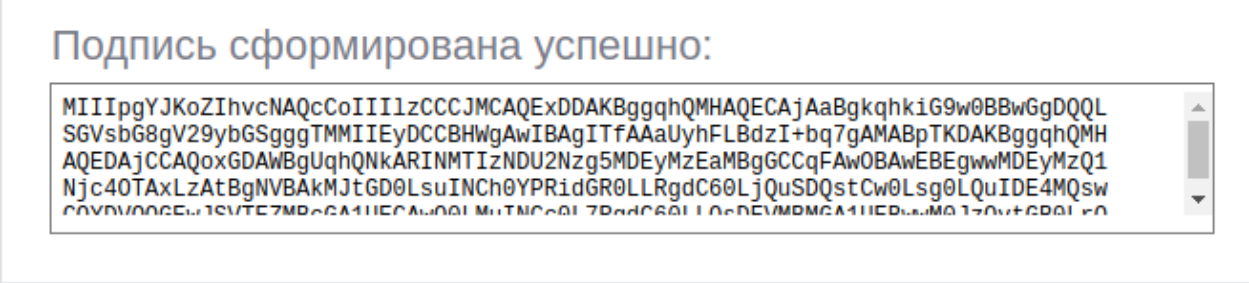
5. Задайте новый пароль на контейнер.



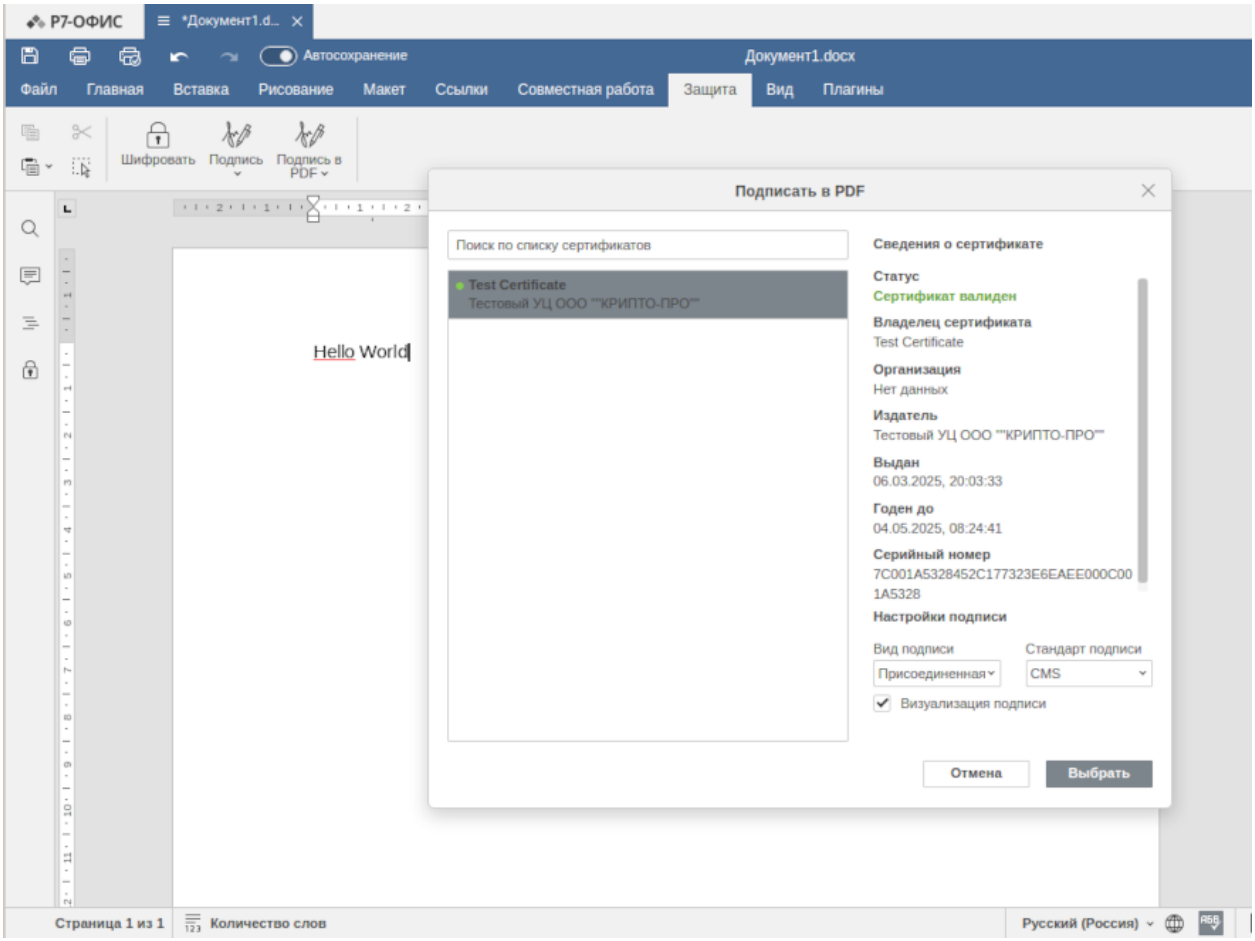
6. После успешного создания сертификата вы можете просмотреть информацию о нём. Для этого выделите его.



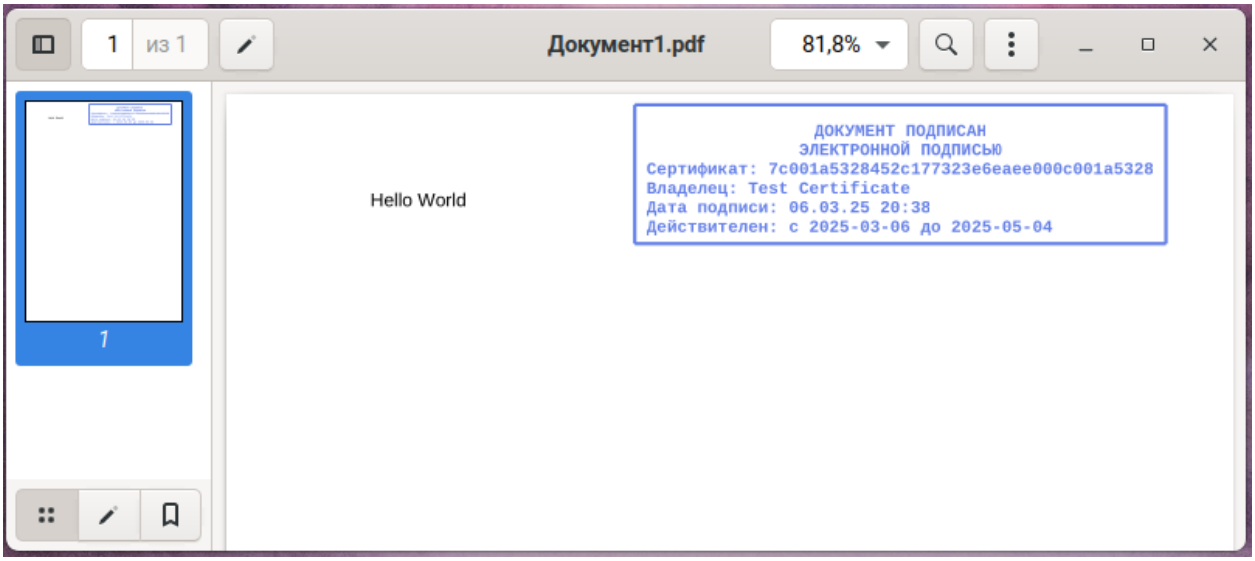
7. Если нажать кнопку «Подписать», будет отображено содержимое подписи.



Вы можете использовать этот же личный сертификат для проверки электронной подписи в текстовом редакторе офисного пакета «Р7-Офис».



В результате подписанный документ будет выглядеть следующим образом.



Стороннее ПО и совместимость

Работа с Flatpak

Flatpak — это система для создания, распространения и запуска десктопных приложений в Linux. Предоставляет пользователям удобный способ получения и установки приложений.

Flatpak предоставляет собой изолированную среду — так называемую «песочницу», в которой создаются и запускаются пользовательские приложения без влияния на основную систему.

Flatpak дедуплицирует библиотеки и другие файлы, используемые несколькими приложениями, что значительно экономит место на диске.

Узнать больше о Flatpak вы можете на официальном сайте: https://docs.flatpak.org/ru/latest/.

Для установки Flatpak выполните следующую команду (необходимо по запросу системы указать пароль суперпользователя root):

```
$ sudo dnf install flatpak
```

При необходимости перезагрузите устройство для завершения установки.

Управление репозиториями Flatpak

Основным поставщиком приложений для работы в системе Flatpak является сайт Flathub — здесь представлено наибольшее количество готовых самых популярных пакетов.

Подключение Flathub

Для подключения Flathub в качестве репозитория пакетов Flatpak выполните команду:

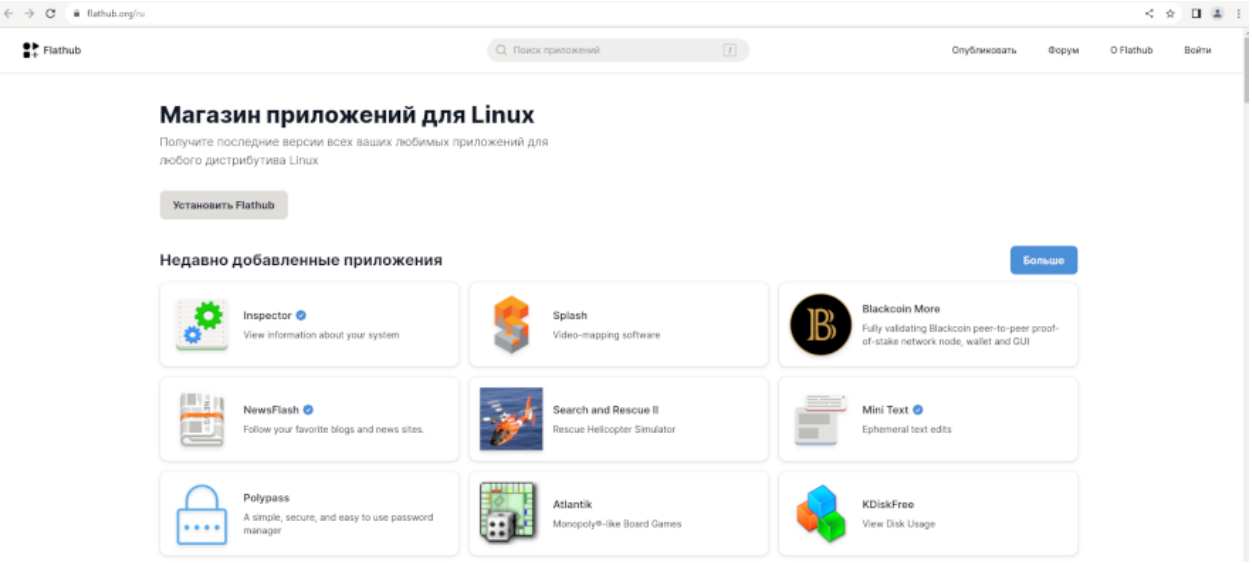
```
$ flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo
```

Кроме Flatpak есть и другие репозитории. Для просмотра всех подключённых репозиториев выполните команду:

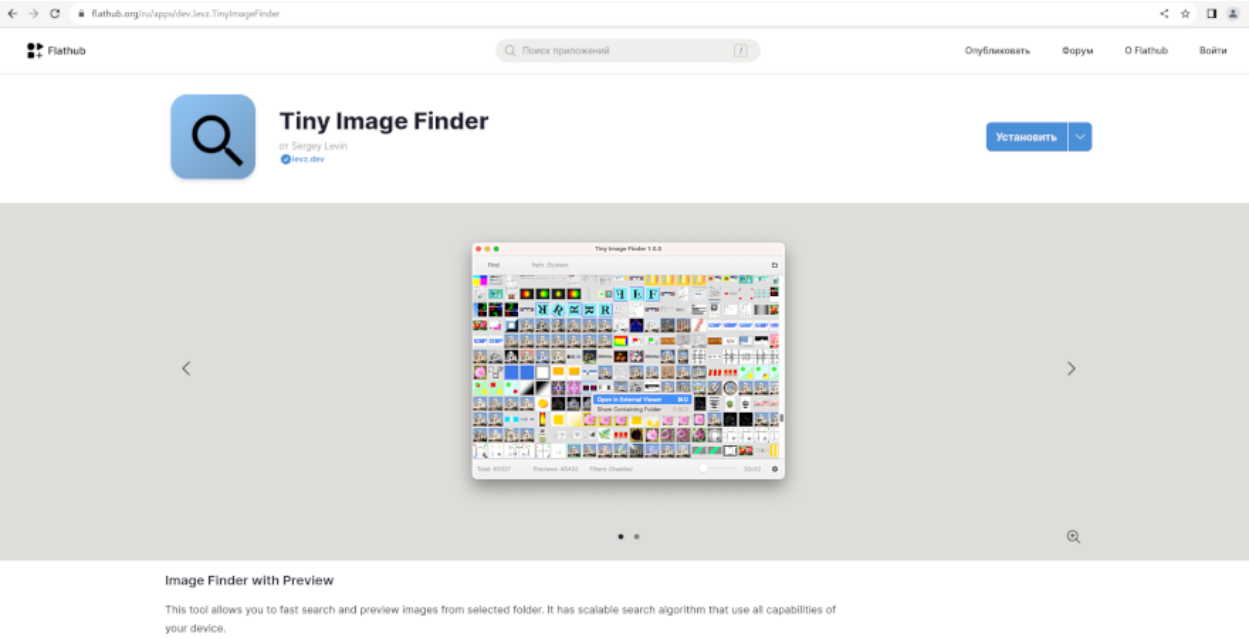
```
$ flatpak remotes
```

Установка приложений с сайта Flathub

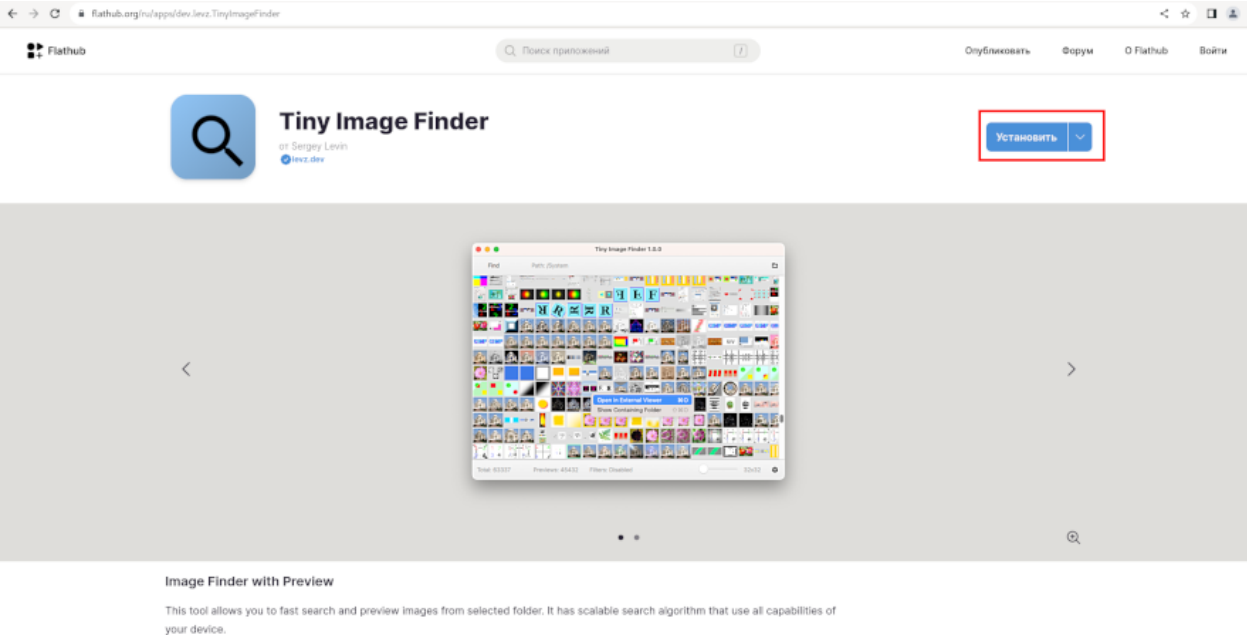
- Перейдите на сайт сайт Flathub.



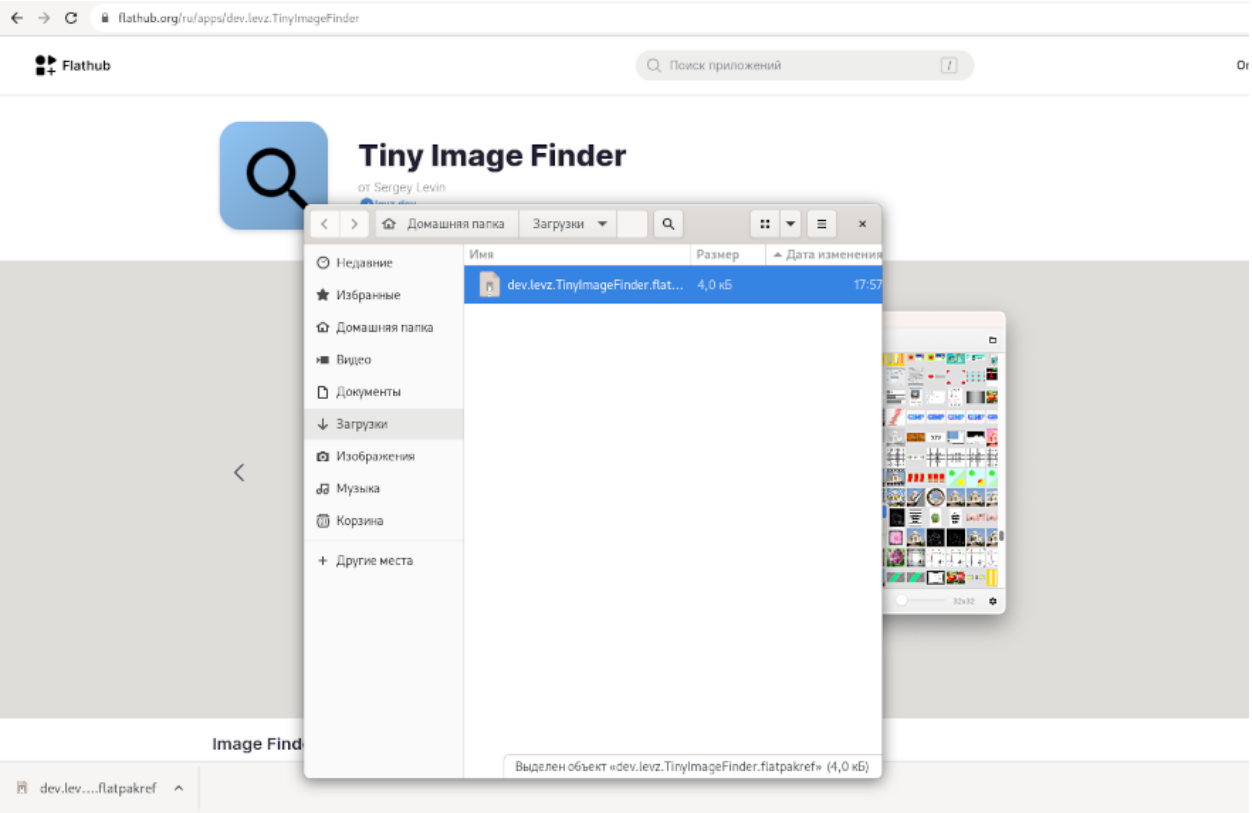
- Выберите необходимое приложение и нажмите на него для перехода к подробной информации о приложении.



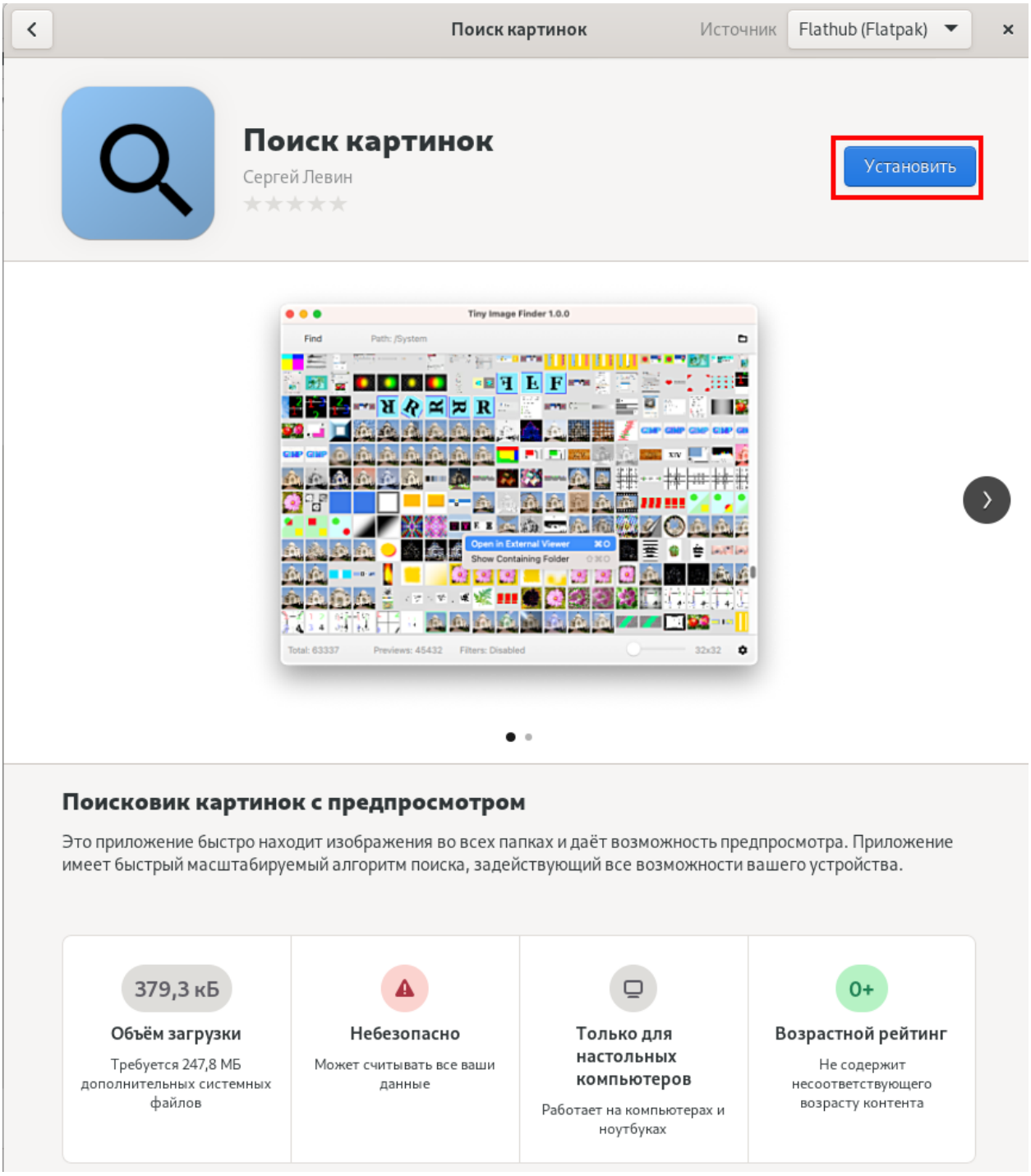
- Нажмите «**Установить**» для начала загрузки.



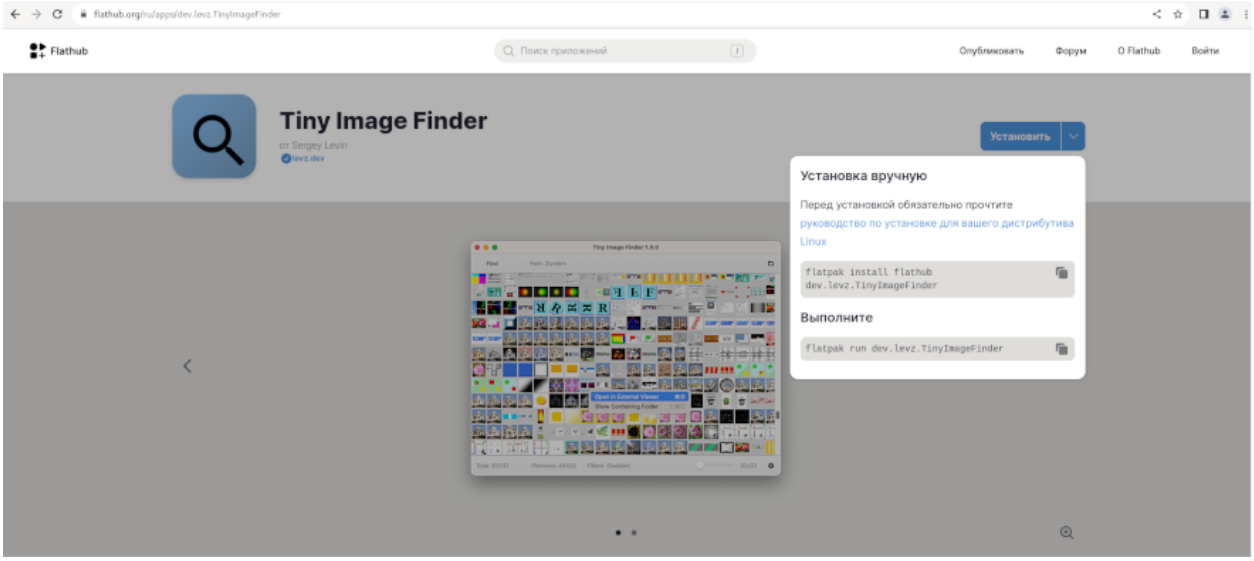
- Перейдите в папку с загруженным файлом и запустите его двойным щелчком левой кнопки мыши.



- Откроется «**Центр приложений**». Нажмите «**Установить**», начнётся автоматическая установка.



Вы также можете установить приложение с помощью терминала. Для этого на сайте Flathub нажмите на стрелку рядом с кнопкой «**Установить**» и выполните в терминале соответствующие команды для установки и запуска.



Поиск пакета Flatpak

Для просмотра полного списка пакетов, входящих в репозиторий Flathub, выполните следующую команду:

```
$ flatpak remote-ls flathub
```

Результат работы команды:

Имя	ID Приложения	Версия	Ветка
AuthPass	app.authpass.AuthPass	1.9.6_1904	stable
BlueBubbles	app.bluebubbles.BlueBubbles	1.12.2.1	stable
Cantara	app.cantara.Cantara	2.4.0	stable
Blurble	app.drey.Blurble	0.4.0	stable
Damask	app.drey.Damask	0.2.0	stable
Dialect	app.drey.Dialect	2.1.1	stable
Ear Tag	app.drey.EarTag	0.4.2	stable
Elastic	app.drey.Elastic	0.1.3	stable

Для поиска определённого пакета по его имени выполните следующую команду:

```
$ flatpak search имя пакета
```

Пример поиск приложения с именем `Pencil2D`:

```
$ flatpak search Pencil2D
```

Имя	Описание	ID Приложения	Версия	Ветка	Удалённые репозитории
Pencil2D	2D animation software supporting bitmap and vector graphics	org.pencil2d.Pencil2D	0.6.6	stable	flathub

Для установки приложения используйте имя пакета приложения из колонки «**ID Приложения**» и имя репозитория из колонки «**Удалённые репозитории**».

Установка приложения

Для установки приложения выполните следующую команду:

```
$ flatpak install имя_репозитория имя_пакета_приложения
```

Пример: установка приложения `Pencil2D` с ID `org.pencil2d.Pencil2D` из репозитория `flathub`:

```
$ flatpak install flathub org.pencil2d.Pencil2D
```

```
Поиск совпадений…
Требуемая среда исполнения для org.pencil2d.Pencil2D/x86_64/stable (runtime/org.kde.Platform/x86_64/5.15) найдена в удалённом репозитории flathub
Хотите установить её? [Y/n]: Y

Разрешения org.pencil2d.Pencil2D:
    ipc     network    pulseaudio    wayland    x11    dri    file access [1]    dbus access [2]

    [1] host, xdg-config/kdeglobals:ro
    [2] com.canonical.AppMenu.Registrar


        ID                                                    Ветка          Оп          Удалённый репозиторий          Загрузка
 1. [/] org.freedesktop.Platform.GL.default                   20.08          i           flathub                            7,1 МБ / 106,4 МБ
 2. [ ] org.freedesktop.Platform.VAAPI.Intel                  20.08          i           flathub                         < 11,6 МБ
 3. [ ] org.freedesktop.Platform.openh264                     2.0            i           flathub                          < 1,5 МБ
 4. [ ] org.kde.KStyle.Adwaita                                5.15           i           flathub                          < 6,6 МБ
 5. [ ] org.kde.Platform.Locale                               5.15           i           flathub                        < 341,5 МБ (частично)
 6. [ ] org.kde.PlatformTheme.QGnomePlatform                  5.15           i           flathub                          < 9,7 МБ
 7. [ ] org.kde.PlatformTheme.QtSNI                           5.15           i           flathub                          < 1,3 МБ
 8. [ ] org.kde.WaylandDecoration.QGnomePlatform-decoration   5.15           i           flathub                         < 10,2 МБ
 9. [ ] org.kde.Platform                                      5.15           i           flathub                        < 373,5 МБ
10. [ ] org.pencil2d.Pencil2D                                 stable         i           flathub                         < 12,0 МБ

Установка 1/10…                    14%  506,4 кБ/s  01:26
```

Для просмотра всех установленных приложений выполните следующую команду:

```
$ flatpak list
```

Пример: результат работы команды

```
Имя                                   ID Приложения                                   Версия        Ветка          Тип установки
QGnomePlatform                        org.kde.PlatformTheme.QGnomePlatform                          5.15           system
QGnomePlatform                        org.kde.PlatformTheme.QGnomePlatform                          5.15-22.08     system
QtSNI                                 org.kde.PlatformTheme.QtSNI                                   5.15           system
QGnomePlatform-decoration             org.kde.WaylandDecoration.QGnomePlatform-decoration           5.15           system
Pencil2D                              org.pencil2d.Pencil2D                           0.6.6         stable         system
```

Удаление приложений

Для удаления приложения выполните следующую команду:

```
$ flatpak uninstall имя из ID Приложения
```

Пример: удаление приложения Pencil2D:

```
$ flatpak uninstall org.pencil2d.Pencil2D
```

```
        ID                                          Ветка          Оп
 1. [-] org.pencil2d.Pencil2D                       stable         r
 2. [-] org.freedesktop.Platform.GL.default         20.08          r
 3. [-] org.freedesktop.Platform.VAAPI.Intel        20.08          r
 4. [-] org.kde.Platform.Locale                     5.15           r
 5. [-] org.kde.Platform                            5.15           r

Удаление завершено.
```

Вы также можете удалить неиспользуемые компоненты (чтобы освободить место на диске) с помощью следующей команды:

```
$ flatpak uninstall --unused
```

Обновление приложений

Для обновления всех установленных приложений пакета Flatpak до последней версии выполните следующую команду:

```
$ flatpak update
```

Добавление в систему сторонних репозиториев

Иногда возникает необходимость установить приложение, которого нет в имеющихся репозиториях. В этом случае есть возможность добавить в систему сторонний репозиторий.

Рассмотрим подключение стороннего репозитория на примере подключения двух самых популярных сторонних репозиториев: EPEL и RPM Fusion. Подобным образом можно добавить любой другой сторонний репозиторий.

Важно

Рекомендуем быть предельно осторожными при подключении сторонних репозиториев и тщательно соблюдать меры безопасности.

Подключение репозитория EPEL

EPEL (Extra Packages for Enterprise Linux) — это открытый и бесплатный проект репозитория, предоставляемый командой Fedora. Он включает пакеты дополнительного ПО для дистрибутивов Linux. Это самый популярный дополнительный репозиторий.

1. Включите репозиторий CRB (от имени администратора — `sudo`):

```
$ sudo dnf config-manager --set-enabled crb
```

2. Установите EPEL RPM (от имени администратора — `sudo`):

```
$ sudo dnf install epel-release epel-next-release
```

3. Теперь вы можете устанавливать приложения из репозитория EPEL (от имени администратора — `sudo`):

```
$ sudo dnf install имя_приложения
```

Вы можете вывести список пакетов, доступных для установки, следующей командой:

```
$ dnf repo-pkgs epel list
```

Подключение репозитория RPM Fusion

RPM Fusion содержит дополнительные пакеты, которые не могут быть распространены в репозиториях Fedora Project, в основном, по лицензионным причинам.

В RPM Fusion есть два типа репозиториев:

- RPM Fusion Free — бесплатный репозиторий, содержащий пакеты с открытым исходным кодом.
- RPM Fusion Non-Free — несвободный репозиторий, содержащий не Open Source-пакеты и пакеты, которые не являются бесплатными для коммерческих целей.

Вы можете добавить оба репозитория и использовать их одновременно или установить только один — Free или Non-Free.

1. Подключите EPEL-репозиторий (см. Подключение репозитория EPEL).
2. Для установки RPM Fusion Free выполните следующую команду (от имени администратора — `sudo`):

```
$ sudo dnf install --nogpgcheck https://mirrors.rpmfusion.org/free/el/rpmfusion-free-release-
↪$(rpm -E %rhel).noarch.rpm
```

3. Для установки RPM Fusion Non-Free выполните следующую команду (от имени администратора — `sudo`):

```
$ sudo dnf install --nogpgcheck https://mirrors.rpmfusion.org/nonfree/el/rpmfusion-nonfree-
↪release-$(rpm -E %rhel).noarch.rpm
```

4. Чтобы проверить, активен ли репозиторий RPM Fusion, выполните следующую команду (от имени администратора — `sudo`):

```
$ sudo dnf repolist rpmfusion-*
```

5. Чтобы вывести список всех пакетов, доступных в репозитории RPM Fusion, выполните следующую команду (от имени администратора — `sudo`):

```
$ sudo dnf --enablerepo=rpmfusion-* list available | more
```

6. Для установки пакета из репозитория RPM Fusion выполните следующую команду (от имени администратора — `sudo`):

```
$ sudo dnf --enablerepo=rpmfusion-* install имя_пакета
```

Вы также можете подключить сторонний репозиторий, если есть .repo-файл, с помощью следующей команды:

```
$ dnf config-manager --add-repo путь_к_.repo_файлу
```

Пример подключения .repo-файла docker.io:

```
$ dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
```

Мигратор

«Мигратор» — это утилита, позволяющая упростить миграцию RHEL-подобных (RHEL — Red Hat Enterprise Linux) дистрибутивов на МСВСфера ОС с сохранением настроек, данных и приложений.

Сценарии миграции на более высокую верию

На текущий момент доступны следующие сценарии миграции на более высокую версию.

- Миграция с CentOS 7/8/9 на МСВСфера ОС 8/9/10 соответственно;
- Миграция с RHEL 8/9 на МСВСфера ОС 9/10 соответственно;
- Миграция с AlmaLinux/Rocky Linux/Oracle Linux/Euro Linux 8/9 на МСВСфера ОС 9/10 соответственно;

- Миграция с МСВСфера ОС 8/9 на МСВСфера ОС 9/10 соответственно.

Инструкции по миграции

Миграция с CentOS 7 на МСВСфера 8 ОС

1. Если вы используете CentOS, то на данный момент большинство зеркал отключены. Для успешной миграции необходимо переключить репозитории на зеркало mirror.yandex.ru:

```
$ sudo sed -i 's@^mirrorlist@#mirrorlist@g;s@#baseurl=http://mirror.centos.org/centos/
↪$releasever@baseurl=https://mirror.yandex.ru/centos/7.9.2009/@g' \
 /etc/yum.repos.d/CentOS-Base.repo
```

   Далее обновите систему и перезагрузить компьютер:

```
$ sudo yum update -y
```

```
$ sudo reboot
```

2. Подключите репозиторий «Мигратора»:

```
$ sudo yum -y install https://repo1.msvsphere-os.ru/migrator/migrator-release-latest-el
↪$(rpm --eval %rhel).noarch.rpm
```

3. Установите «Мигратор» и пакеты с правилами для обновления:

```
$ sudo yum -y install leapp-upgrade leapp-data-msvsphere
```

4. Запустите preupgrade-проверку:

```
$ sudo leapp preupgrade
```

   Результатом работы утилиты будет отчёт, по которому можно определить возможность обновления системы.

   Подробный отчёт можно найти в файле `/var/log/leapp-report.txt`.

   Также будет сгенерирован файл `/var/log/leapp/answerfile` для подтверждения исправления найденных проблем, типичных для EL 7.

5. При возникновении ошибок их необходимо устранить. См. «*Известные проблемы при миграции с EL 7 на МСВСфера 8 ОС*».

6. Снова запустите preupgrade-проверку. Если результат будет помечен жёлтым или зелёным, то можно продолжать миграцию.

7. Запустите обновление:

```
$ sudo leapp upgrade
```

8. После удачного завершения обновления перезагрузите систему:

```
$ reboot
```

9. После перезагрузки в загрузчике операционной системы выберите пункт `Migrator-Upgrade-Initramfs`.

10. Дождитесь окончания обновления системы.

    В конце обновления система будет перезагружена и загрузится в обычном режиме.

    При первом запуске будет запущена процедура повторной маркировки меток SELinux. После чего система перезагрузится ещё раз.

11. Если вы до этого обновили EL 7 до МСВСфера 8 ОС, то вам необходимо сначала удалить пакеты, оставшиеся от предыдущей версии «Мигратора». Для этого выполните следующую команду:

```
$ sudo dnf remove '*leapp*'
```

И удалите из конфигурационного файла `/etc/yum.conf` следующую строку:

```
exclude=python2-leap...
```

Миграция с МСВСфера 8 на МСВСфера 9 ОС

1. Обновите МСВСфера 8 ОС и перезагрузите компьютер:

```
$ sudo dnf update -y

$ sudo reboot
```

2. Подключите репозиторий «Мигратора»:

```
$ sudo dnf -y install -y https://repo1.msvsphere-os.ru/migrator/migrator-release-latest-el
↪$(rpm --eval %rhel).noarch.rpm
```

3. Установите «Мигратор» и пакет с правилами для обновления:

```
$ sudo dnf -y install leapp-upgrade leapp-data-msvsphere
```

4. Запустите preupgrade-проверку:

```
$ sudo leapp preupgrade
```

Результатом работы утилиты будет отчёт, по которому можно определить возможность обновления системы. Подробный отчёт можно найти в файле

`/var/log/leapp-report.txt`. Также будет сгенерирован файл `/var/log/leapp/answerfile` для подтверждения исправления найденных проблем, типичных для МСВСфера 8 ОС.

5. При возникновении ошибок их необходимо устранить. См. «*Известные проблемы при миграции с EL 8 на МСВСфера 9 ОС*».

6. Снова запустите preupgrade-проверку. Если результат будет помечен жёлтым или зелёным, то можно продолжать миграцию.

7. Запустите обновление:

```
$ sudo leapp upgrade
```

8. После удачного завершения обновления перезагрузите систему:

```
$ reboot
```

9. После перезагрузки в загрузчике операционной системы выберите пункт `Migrator-Upgrade-Initramfs`.

10. Дождитесь окончания обновления системы. В конце обновления система будет перезагружена и загрузится в обычном режиме. При первом запуске будет запущена процедура повторной маркировки меток SELinux. После чего система перезагрузится ещё раз.

11. Если вы до этого обновили EL 8 до МСВСфера 9 ОС, то вам необходимо сначала удалить пакеты, оставшиеся от предыдущей версии «Мигратора». Для этого выполните следующую команду:

```
$ sudo dnf remove '*leapp*'
```

Миграция с AlmaLinux/Rocky Linux/Oracle Linux/Euro Linux 8 на МСВСфера 9 ОС

Процесс миграции с дистрибутивов AlmaLinux, Rocky Linux, Oracle Linux, Euro Linux 8 на МСВСфера 9 ОС идентичен процессу «*Миграция с МСВСфера 8 на МСВСфера 9 ОС*».

Миграция с МСВСфера 9 ОС на МСВСфера 10 ОС

1. Обновите МСВСфера 9 ОС и перезагрузите компьютер:

```
$ sudo dnf update -y

$ sudo reboot
```

2. Подключите репозиторий «Мигратора»:

```
$ sudo dnf -y install -y https://repo1.msvsphere-os.ru/migrator/migrator-release-latest-el
↪$(rpm --eval %rhel).noarch.rpm
```

3. Установите «Мигратор» и пакет с правилами для обновления:

```
$ sudo dnf -y install leapp-upgrade leapp-data-msvsphere
```

4. Запустите preupgrade-проверку:

```
$ sudo leapp preupgrade
```

   Результатом работы утилиты будет отчёт, по которому можно определить возможность обновления системы.

   Подробный отчёт можно найти в файле `/var/log/leapp-report.txt`.

   Также будет сгенерирован файл `/var/log/leapp/answerfile` для подтверждения исправления найденных проблем, типичных для МСВСфера 8 ОС.

5. При возникновении ошибок их необходимо устранить. См. «*Известные проблемы при миграции с EL 9 на МСВСфера 10 ОС*».

6. Снова запустите preupgrade-проверку. Если результат будет помечен жёлтым или зелёным, то можно продолжать миграцию.

7. Запустите обновление:

```
$ sudo leapp upgrade
```

8. После удачного завершения обновления перезагрузите систему:

```
$ reboot
```

9. После перезагрузки в загрузчике операционной системы выберите пункт `Migrator-Upgrade-Initramfs`.

10. Дождитесь окончания обновления системы.

    В конце обновления система будет перезагружена и загрузится в обычном режиме.

    При первом запуске будет запущена процедура повторной маркировки меток SELinux. После чего система перезагрузится ещё раз.

Известные проблемы

Известные проблемы при миграции с EL 7 на МСВСфера 8 ОС

1. Обратите внимание, что Oracle Linux должен быть загружен в ядро 3.10.
2. Скорее всего для продолжения обновления необходимо будет отключить модуль `pata_acpi` и PAM-модуль `pkcs11`:

```
$ sudo rmmod pata_acpi
```

```
$ sudo leapp answer --section remove_pam_pkcs11_module_check.confirm=True
```

3. Для Oracle Linux и Red Hat Linux необходимо будет удалить пакет `python-hwdata`:

```
$ sudo yum -y remove python-hwdata
```

4. Для Oracle Linux также будет необходимо удалить пакеты `uname26` и `iwlax2xx-firmware`:

```
$ sudo yum -y remove uname26 iwlax2xx-firmware
```

Эти и другие требуемые изменения будут описаны в файле `/var/log/leapp/leapp-report.txt`.

Известные проблемы при миграции с EL 8 на МСВСфера 9 ОС

1. Обратите внимание, что Oracle Linux должен быть загружен в ядро 4.18.
2. Если Oracle Linux не сможет загрузить метаданные некоторых репозиториев, отключите их:

```
$ sudo dnf config-manager --set-disabled <repo>
```

3. Для CentOS Stream необходимо изменить версию дистрибутива с `8` на `8.5` в файле `/etc/centos-release`. А также отключить репозитории, у которых невозможно загрузить метаданные:

```
$ sudo dnf config-manager --set-disabled <repo>
```

4. Скорее всего для продолжения обновления будет необходимо исправить зону `firewalld`:

```
$ sudo sed -i "s/^AllowZoneDrifting=.*/AllowZoneDrifting=no/" /etc/firewalld/firewalld.conf
```

А также отключить проверку vdo устройств:

```
$ sudo leapp answer --section check_vdo.confirm=True
```

Запретить доступ по `ssh` пользователю `root`:

```
$ echo PermitRootLogin no | sudo tee -a /etc/ssh/sshd_config
```

Эти и другие требуемые изменения будут описаны в файле `/var/log/leapp/leapp-report.txt`.

Известные проблемы при миграции с EL 9 на МСВСфера 10 ОС

1. Чтобы процесс обновления не оборвался из режима `Migrator-Upgrade-Initramfs`, нужно удалить пакет `alsa-sof-firmware`:

```
$ sudo dnf remove alsa-sof-firmware
```

2. Для устранения проблем с `dhclient` выполните следующую команду:

```
$ grep -q '^dhcp=internal' /etc/NetworkManager/NetworkManager.conf || \
 sed -i '/^\[main\]/a dhcp=internal' /etc/NetworkManager/NetworkManager.conf
```

3. Также возможна проблема, которая в процессе обновления не даёт создать `initramfs` для ядра `el10`. Чтобы решить её до запуска обновления и перезагрузки, выполните следующую команду:

```
$ sudo sed -i '/network-legacy/d' /etc/dracut.conf.d/*.conf
```

Диагностика

В некоторых случаях обновление системы из режима `Migrator-Upgrade-Initramfs` заканчивается неудачно и запускается отладочная консоль. Скорее всего ошибка происходит на этапе установки отдельных пакетов, поэтому `DNF` возвращает ошибку. При этом остальные пакеты устанавливаются успешно, так как все зависимости разрешаются ещё на этапе подготовки системы к обновлению.

В этом случае следует изучить системный журнал доступный через команду `journalctl`.

1. В `Initramsfs` находится обрезанная версия загрузочного образа, поэтому команды `less` и `more` там отсутствуют. Лучше всего перенаправить вывод `journalctl` в файл:

```
$ journalctl > system.log
```

2. Подключите корневой раздел основной операционной системы:

```
$ mkdir /mnt
$ mount /dev/sdXX /mnt
```

3. Для LVM подключите нужный том:

```
$ mount /dev/mapper/XX-root /mnt
```

4. Если вы используете отдельный раздел для каталога `/usr`, то его также необходимо подключить в каталог `/mnt/usr`.

5. Далее скопируйте `system.log` в `/mnt/tmp` и измените корневой каталог на `/mnt` с помощью команды:

```
$ chroot /mnt
```

   Теперь вы можете просматривать файл `system.log` с помощью команды `less`.

6. Если в системном журнале видно, что DNF действительно закончил устанавливать пакеты, но установка некоторых пакетов закончилась с ошибкой, то скорее всего систему можно будет загрузить.

7. Следует обязательно переразметить метки SELinux. Для этого необходимо создать в корневом разделе операционной системы файл `.autorelabel`:

```
$ touch /mnt/.autorelabel
```

8. Если вы изменяли корневой каталог командой `chroot`, то необходимо вернуться в корневой каталог `initramfs` при помощи команды `exit` или комбинации клавиш `Ctrl` + `d`.

9. После этого отключите все разделы, которые вы подключали выше:

```
$ umount /mnt/usr
$ umount /mnt
```

10. Перезагрузите систему:

```
$ reboot
```

Система начнёт загрузку в обычном режиме, переразметит метки SELinux и перезагрузится ещё раз.

Поддержка сторонних поставщиков

В настоящее время «Мигратор» поддерживает следующие сторонние репозитории:

- EPEL в настоящее время поддерживается только для обновлений до МСВСфера ОС.
- Docker CE — для всех поддерживаемых операционных систем.
- MariaDB — для всех поддерживаемых операционных систем.
- Microsoft Linux Package Repositories — для всех поддерживаемых операционных систем.
- nginx — для всех поддерживаемых операционных систем.
- PostgreSQL — для всех поддерживаемых операционных систем.

Процедура интеграции пакетов сторонних репозиториев в процесс обновления «Мигратора» состоит из следующих шагов.

1. Клонируйте репозиторий leapp-data.
2. Перейдите в ветку devel.
3. Перейдите в папку `vendors.d`. Она содержит необходимые файлы для каждого поддерживаемого стороннего репозитория.
4. Чтобы добавить новый сторонний репозиторий необходимо создать отдельные файлы. Эти файлы должны иметь одинаковую часть `<vendor_name>` в своих именах:
   - `<vendor_name>_map.json` — см. «*Файл соответствия репозиториев*».
   - `<vendor_name>.repo` — см. «*Информация о репозитории, в котором содержатся пакеты*».
   - `<vendor_name>.sigs` — см. «*Список подписей поставщика для пакетов*».
   - `<vendor_name>_pes.json` — см. «*Список событий миграции для пакета*».

Предупреждение

Обратите внимание, что все сторонние пакеты должны быть подписаны.

Файл соответствия репозиториев

Файл соответствия в формате `.json` содержит информацию о соответствии между репозиториями исходной системы (той, которую вы обновляете) и репозиториями целевой системы (теми, из которых будут устанавливаться пакеты при обновлении).

Структура файла соответствия:

- `datetime` — временная метка создания файла (например, `202306191741Z`);
- `version` — версия базы данных. Актуальную версию можно посмотреть в репозитории leapp-data.
- Разделы (описаны ниже): * `mapping` — определяет соответствия между репозиториями. * `repositories` — описывает все задействованные репозитории.

Важно

Файл соответствия определяет, участвуют ли пакеты от поставщика в процессе обновления. Если найден хотя бы один из указанных исходных репозиториев, поставщик считается активным, и его репозитории и события PES будут участвовавть в процессе обновления. Если ни один репозиторий не найден — поставщик будет проигнорирован при обновлении.

Раздел `mapping` — соответствие репозиториев

Устанавливает соответствие между версиями системы и связанными с ними репозиториями.

Содержит следующие поля:

- `source_major_version` — начальная основная (мажорная) версия системы (например, 7 или 8).
- `target_major_version` — целевая основная (мажорная) версия (например, 8 или 9).
- `entries` — список правил соответствия:
- `source` — ID исходный репозиторий из старой системы.
- `target` — ID одного или нескольких репозиториев, из которых будут устанавливаться обновлённые пакеты.
- Каждый исходный репозиторий может быть сопоставлен с одним или несколькими целевыми репозиториями.
- Все репозитории, указанные в `source` и `target`, должны быть также описаны в разделе `repositories` и в файле `.pes`. Их ID должны совпадать.

В качестве примера вы можете использовать файлы для MariaDB — mariadb_map.json.

Раздел `repositories` — описание репозиториев

Этот раздел содержит детальную информацию о каждом исходном и целевом репозитории.

Информация о каждом репозитории должна включать следующее:

- `pesid` — уникальный идентификатор, используемый в `.pes`-файлах. Если он не совпадёт, обновление завершится с ошибкой.
- `entries`:
  - `major_version` — основная (мажорная) версия системы.
  - `repoid` — ID репозитория. Должен совпадать с ID из `.repo`-файла. (Может не совпадать с `pesid`.)
  - `arch` — архитектура системы (например, `x86_64`).
  - `channel` — канал репозитория (актуально для Red Hat):
    * `ga` — стабильный (основной).
    * `beta` — тестовый.
    * `eus`, `e4s`, `aus`, `tus` — расширенные каналы поддержки.
  - `repo_type` — тип репозитория:
    * `rpm` — бинарные пакеты.
    * `srpm` — исходные пакеты.
    * `debuginfo` — отладочная информация.

В качестве примера вы можете использовать файлы для MariaDB — mariadb_map.json.

Информация о репозитории, в котором содержатся пакеты

Файл `.repo` определяет, какие именно репозитории поставщика будут использоваться при обновлении.

Файл репозитория имеет формат, типичный для файлов репозитория пакетов YUM/DNF:

```
[repository ID]
name = repository name
baseurl = repository baseurl
gpgkey = GPG Key directory
gpgcheck = 1
enabled = 1
```

В качестве примера вы можете использовать файлы для MariaDB — mariadb.repo.

Совет

Репозитории, перечисленные в этом файле, используются только во время обновления. В обновлённую систему репозитории добавляются как обычно.

Список подписей поставщика для пакетов

Файл `.sigs` должен содержать список публичных заголовков подписей, которыми подписаны пакеты.

Чтобы узнать подпись конкретного пакета, выполните команду:

```
$ rpm -qip <ИМЯ_ПАКЕТА> | grep Signature
```

Пример вывода:

```
Signature   : DSA/SHA1, Mon Aug 23 08:17:13 2021, Key ID 8c55a6628608cb71
```

Значение `Key ID` (в данном случае — `8c55a6628608cb71`) и есть публичный заголовок пакета, который необходимо добавить в файл подписей. Файл должен содержать одну подпись на строку.

В качестве примера вы можете использовать файл для MariaDB — mariadb.sigs.

Список событий миграции для пакета

Список событий миграции используется для описания действий, которые невозможно выполнить стандартными средствами DNF.

Как правило, эта информация хранится в файле:

- в GitHub: `leapp-data/vendors.d/<vendor_name>_pes.json`
- в обновляемой системе: `/etc/leapp/files/vendors.d/<vendor_name>_pes.json`

Предупреждение

Действия из PES-файлов применяются **только** к тем пакетам, которые подписаны и имеют подпись в файле `<vendor_name>.sigs`. Неподписанные пакеты будут обновлены **только** в том случае, если на них есть зависимость от подписанных пакетов.

Создание `pes.json`

Если дополнительные действия не требуются, используйте следующий базовый шаблон:

```
{
  "legal_notice": "",
  "packageinfo": [],
  "provided_data_streams": [
      "2.0"
  ],
  "timestamp": "202408081321Z"
}
```

Если требуются дополнительные действия, то добавьте информацию для обновления. Для этого создайте новую запись в массиве `packageinfo`. Поле `action` определяет одно из следующих действий:

- `0` (present) — сохранить пакеты в `in_packageset`, чтобы гарантировать включение репозитория, в котором они находятся в целевой системе.
- `1` (removed) — все указанные пакеты будут удалены из `in_packageset`.
- `2` (deprecated) — сохранить пакеты в `in_packageset`, чтобы гарантировать включение репозитория, в котором они находятся в целевой системе, но считать их устаревшими.
- `3` (replaced) — удалить все пакеты из `in_packageset`, установить отсутствующие в системе пакеты из `out_packageset`, а также сохранить установленные.
- `4` (split) — установить отсутствующие в системе пакеты из `out_packageset`, сохранить установленные из `out_packageset`. Удалить из `in_packageset` пакеты, отсутствующие в `out_packageset`. Если пакет X разделён на Y и Z, пакет X будет удалён. Если пакет X разделён на X и Y, пакет X не будет удалён.

  Пример:

  ```
  {
   "action": 4,
   "architectures": [
       "aarch64",
       "ppc64le",
       "s390x",
       "x86_64"
   ],
   "id": 23,
   "in_packageset": {
  ```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)


```
    "package": [
        {
            "modulestreams": [
                null
            ],
            "name": "ntp",
            "repository": "base"
        }
    ],
    "set_id": 29
},
"initial_release": {
    "major_version": 7,
    "minor_version": 7,
    "os_name": "CentOS"
},
"modulestream_maps": [
    {
        "in_modulestream": null,
        "out_modulestream": null
    }
],
"out_packageset": {
    "package": [
        {
            "modulestreams": [
                null
            ],
            "name": "chrony",
            "repository": "msvsphere8-baseos"
        },
        {
            "modulestreams": [
                null
            ],
            "name": "ntpstat",
            "repository": "msvsphere8-appstream"
        }
    ],
    "set_id": 30
},
"release": {
    "major_version": 8,
    "minor_version": 0,
    "os_name": "MSVSphere"
}
},
```


- 5 (merged) — установить отсутствующие в системе пакеты из `out_packageset`, удалить из `in_packageset` пакеты, которых нет в `out_packageset`. Если пакеты Y и Z объединены в X, пакеты Y и Z будут удалены. Если пакеты Y и Z не объединяются в X, пакеты Y и Z не будут удалены.

  Пример:


```
{
 "action": 5,
 "architectures": [
     "aarch64",
     "ppc64le",
     "s390x",
     "x86_64"
 ],
 "id": 93,
```


(продолжение на следующей странице)

(продолжение с предыдущей страницы)


```
"in_packageset": {
    "package": [
        {
            "modulestreams": [
                null
            ],
            "name": "infiniband-diags",
            "repository": "base"
        },
        {
            "modulestreams": [
                null
            ],
            "name": "libibmad",
            "repository": "base"
        }
    ],
    "set_id": 118
},
"initial_release": {
    "major_version": 7,
    "minor_version": 7,
    "os_name": "CentOS"
},
"modulestream_maps": [
    {
        "in_modulestream": null,
        "out_modulestream": null
    }
],
"out_packageset": {
    "package": [
        {
            "modulestreams": [
                null
            ],
            "name": "infiniband-diags",
            "repository": "msvsphere8-baseos"
        }
    ],
    "set_id": 9451
},
"release": {
    "major_version": 8,
    "minor_version": 0,
    "os_name": "MSVSphere"
}
},
```


- `6` (moved to new repository) — сохранить пакет, чтобы убедиться, что репозиторий, в котором он находится в целевой системе, включён. С `in_packageset` ничего не происходит, поскольку он всегда содержит один пакет — такой же, как и пакет «out».
- `7` (renamed) — удалить `in_packageset` и установить `out_packageset`, если он не установлен. Если он уже установлен, оставить `out_packageset` как есть.
- `8` (reinstalled) — переустановить пакет `in_packageset` во время обновления. Полезно, если версия пакета не меняется между системами.

Действия `1`, `3`, `4` и `5` используются наиболее часто.

Предупреждение

Исключения и особые случаи:

- Любое действие, кроме `present`, игнорируется, если хотя бы один из пакетов в `in_packageset` помечен на удаление.
- Любое действие, кроме `merged`, игнорируется, если пакеты из `in_packageset` не установлены и не помечены на установку.
- Для `merged` достаточно, чтобы хотя бы один пакет из `in_packageset` был установлен или помечен на установку.

Дополнительные поля

- `architectures` — архитектуры, для которых применяются действия.
- `id` — уникальный идентификатор записи.
- `in_packageset` — список исходных пакетов, подлежащих обновлению (поле `repository` указвает репозиторий, из которого пакет был установлен в исходной системе).
- `out_packageset` — список целевых пакетов (поле `repository` должно совпадать с полем «Target system repo name in PES» в файле соответствия для поставщика).
- `initial_release` — исходная система, включая основную (мажорную) и младшую (минорную) (например, `MSVSphere 8.9`).
- `release` — целевая система, включая основную (мажорную) и младшую (минорную) (например, `MSVSphere 9.3`).

Совет

Leapp устанавливает пакеты строго из заданного репозитория. Он лишь включает нужный репозиторий из `out_packageset`, а DNF устанавливает последнюю доступную версию.

Пример структуры `in_packageset` и `out_packageset`:

```
"in_packageset": {
  "package": [
    {
      "module_stream": null,
      "name": "PackageKit",
      "repository": "base"
    },
    {
      "module_stream": null,
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
      "name": "PackageKit-yum",
      "repository": "base"
    }
  ],
  "set_id": 1592
},
```

Переход на МСВСфера ОС с RHEL-подобных дистрибутивов в рамках одной версии

В статье описаны сценарии перехода с RHEL-подобных дистрибутивов на МСВСфера ОС в рамках одной версии. Ниже вы найдёте инструкции для перехода с EL8 (EL — Red Hat **Enterprise Linux**) на МСВСфера 8 ОС и с EL9 на МСВСфера 9 ОС.

Сценарии перехода на МСВСфера ОС в рамках одной версии

На текущий момент доступны следующие сценарии перехода.

- Переход с EL8 на МСВСфера 8 ОС;
- Переход с EL9 на МСВСфера 9 ОС.

Инструкции по переходу

Переход с EL8 на МСВСфера 8 ОС

Примечание

Минимальная поддерживаемая версия операционной системы EL8 — 8.4. Если версия вашей операционной системы ниже, обновите её. Также рекомендуется сделать резервную копию системы.

1. Обновите систему и перезагрузите компьютер:

```
$ sudo dnf update -y

$ sudo reboot
```

2. Загрузите скрипт msvsphere-deploy.sh с помощью следующей команды:

```
$ curl -O https://git.inferitos.ru/msvsphere/msvsphere-deploy/raw/branch/main/msvsphere-
↪deploy.sh
```

3. Запустите скрипт:

```
$ sudo bash msvsphere-deploy.sh
```

Проверьте вывод на наличие ошибок. Если конвертация прошла успешно, вы увидите следующее сообщение: «Migration to MSVSphere is completed».

4. Перезагрузите систему для загрузки ядра МСВСфера ОС:

```
$ sudo reboot
```

5. Убедитесь, что переход прошёл успешно.

   - Проверьте файл релиза:

   ```
   $ cat /etc/redhat-release
   MSVSphere release 8.10 (Server)
   ```

   - Проверьте, что система по умолчанию загружается с ядром МСВСфера ОС:

   ```
   $ sudo grubby --info DEFAULT | grep MSVSphere
   title="MSVSphere (4.18.0-553.16.1.el8_10.x86_64) 8.10 Server"
   ```

Переход с CentOS версий ниже 8.4

1. По состоянию на 31 января 2022 года зеркала CentOS 8 отключены. Для успешного обновления системы до рекомендуемой версии 8.5 вам необходимо обновить файлы конфигурации `dnf`, чтобы они указывали на допустимое зеркало. Рекомендуется использовать следующие команды `sed`:

```
$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[baseos\]/a baseurl=https://mirror.rackspace.com/centos-
↪vault/8.5.2111/BaseOS/$basearch/os' /etc/yum.repos.d/CentOS-Linux-BaseOS.repo

$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[appstream\]/a baseurl=https://mirror.rackspace.com/
↪centos-vault/8.5.2111/AppStream/$basearch/os' /etc/yum.repos.d/CentOS-Linux-AppStream.
↪repo

$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[cr\]/a baseurl=https://mirror.rackspace.com/centos-
↪vault/8.5.2111/ContinuousRelease/$basearch/os' /etc/yum.repos.d/CentOS-Linux-
↪ContinuousRelease.repo

$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[devel\]/a baseurl=https://mirror.rackspace.com/centos-
↪vault/8.5.2111/Devel/$basearch/os' /etc/yum.repos.d/CentOS-Linux-Devel.repo

$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[extras\]/a baseurl=https://mirror.rackspace.com/centos-
↪vault/8.5.2111/extras/$basearch/os' /etc/yum.repos.d/CentOS-Linux-Extras.repo
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[fasttrack\]/a baseurl=https://mirror.rackspace.com/
↪centos-vault/8.5.2111/fasttrack/$basearch/os' /etc/yum.repos.d/CentOS-Linux-FastTrack.
↪repo

$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[ha\]/a baseurl=https://mirror.rackspace.com/centos-
↪vault/8.5.2111/HighAvailability/$basearch/os' /etc/yum.repos.d/CentOS-Linux-
↪HighAvailability.repo

$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[plus\]/a baseurl=https://mirror.rackspace.com/centos-
↪vault/8.5.2111/centosplus/$basearch/os' /etc/yum.repos.d/CentOS-Linux-Plus.repo

$ sudo sed -i -e '/mirrorlist=http:\/\/mirrorlist.centos.org\/?release=$releasever&arch=
↪$basearch&repo=/ s/^#*/#/' -e '/baseurl=http:\/\/mirror.centos.org\/$contentdir\/
↪$releasever\// s/^#*/#/' -e '/^\[powertools\]/a baseurl=https://mirror.rackspace.com/
↪centos-vault/8.5.2111/PowerTools/$basearch/os' /etc/yum.repos.d/CentOS-Linux-PowerTools.
↪repo
```

Или же вы можете использовать флаг `-f` при запуске скрипта msvsphere-deploy.sh:

```
$ sudo bash msvsphere-deploy.sh -f
```

2. Обновите систему:

```
$ sudo dnf update -y
```

3. Загрузите скрипт msvsphere-deploy.sh с помощью следующей команды:

```
$ curl -O https://git.inferitos.ru/msvsphere/msvsphere-deploy/raw/branch/main/msvsphere-
↪deploy.sh
```

4. Запустите скрипт:

```
$ sudo bash msvsphere-deploy.sh
```

5. Убедитесь, что переход прошёл успешно.

   - Проверьте файл релиза:

     ```
     $ cat /etc/redhat-release
     MSVSphere release 8.10 (Server)
     ```

   - Проверьте, что система по умолчанию загружается с ядром МСВСфера ОС:

     ```
     $ sudo grubby --info DEFAULT | grep MSVSphere
     title="MSVSphere (4.18.0-553.16.1.el8_10.x86_64) 8.10 Server"
     ```

Переход с EL9 на МСВСфера 9 ОС

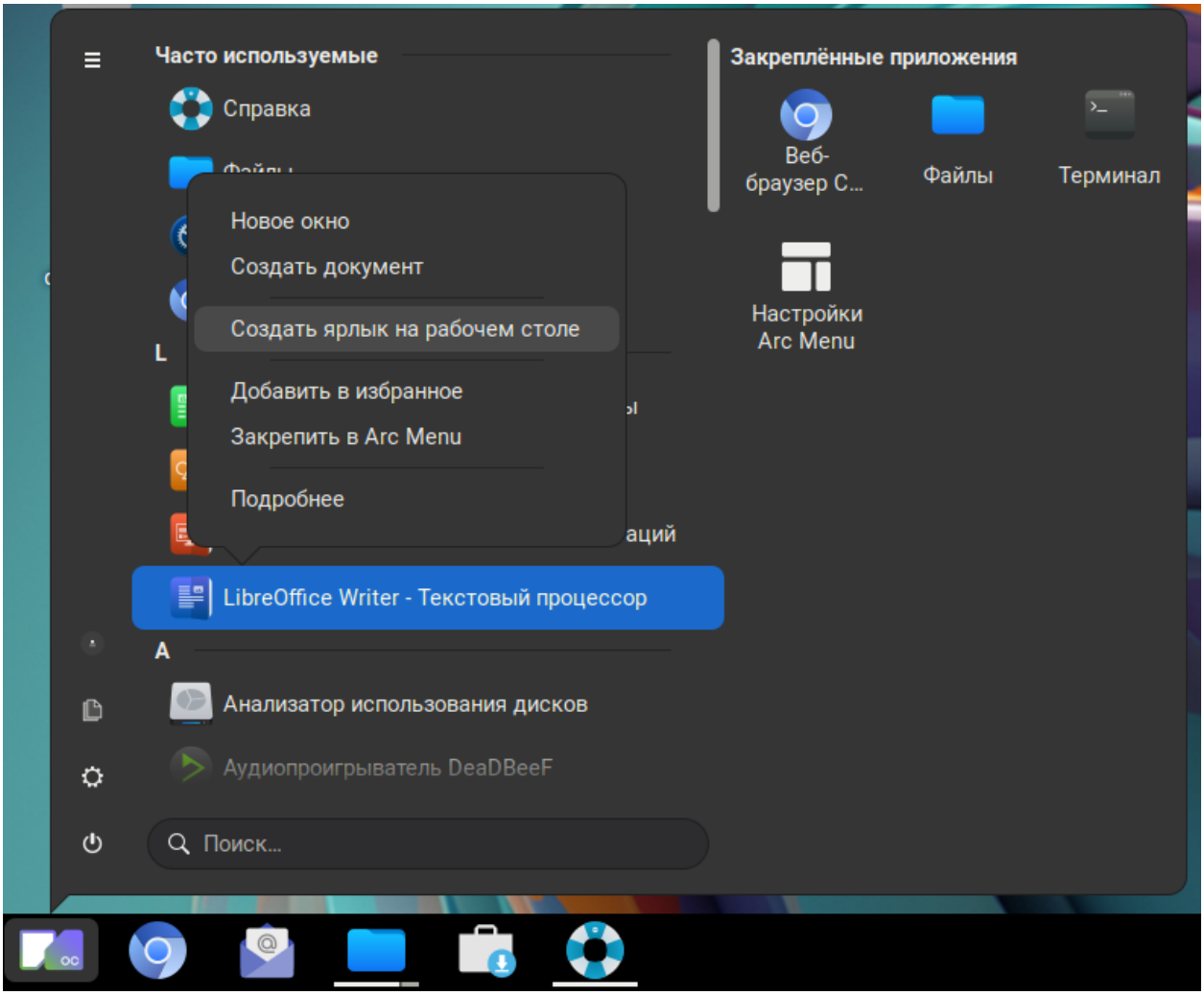
Процесс перехода с дистрибутивов с EL9 на МСВСфера 9 ОС идентичен процессу *Переход с EL8 на МСВСфера 8 ОС*.

Настройка операционной системы

Как перенести значок приложения на рабочий стол

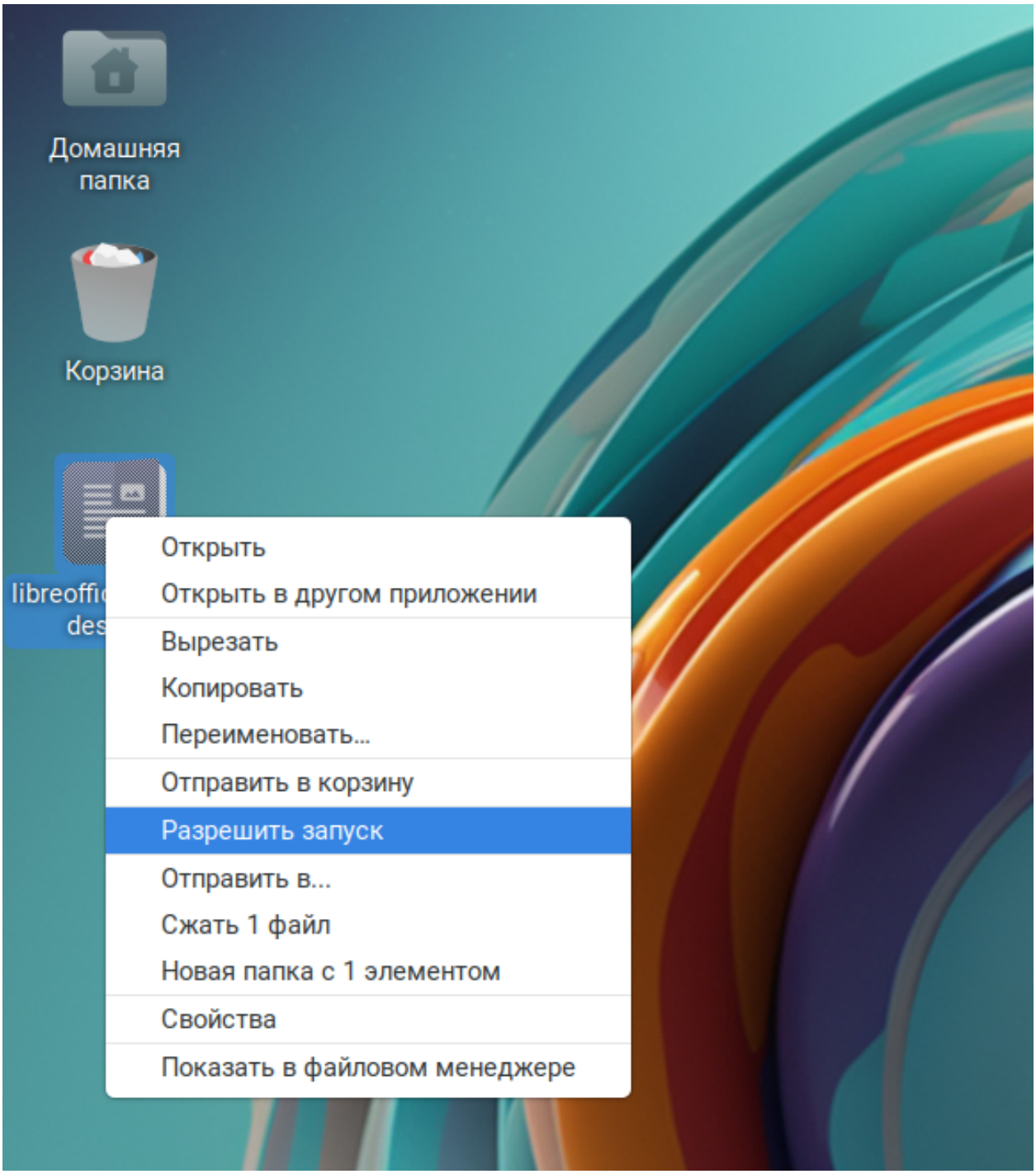
Для того, чтобы перенести значок приложения на рабочий стол, выполните следующие действия.

1. Нажмите на кнопку **Пуск**, откроется меню приложений.
2. Выберите приложение, значок которого вы хотите перенести на рабочий стол и нажмите на него правой клавишей мыши. В контекстном меню выберите **Создать ярлык на рабочем столе**.



3. Значок приложения будет добавлен на рабочий стол, но будет неактивным, а при запуске будет выдаваться предупреждение.

4. Чтобы разрешить запуск приложения, нажмите на его значок на рабочем столе правой клавишей мыши. В контекстном меню выберите **Разрешить запуск**.



5. Теперь значок приложения стал активным и вы можете запускать его с рабочего стола.

Автоматизированная установка (Kickstart-установка)

Установка с помощью Kickstart-файла

Поддерживается следующая архитектура:

- 64-битная архитектура AMD и Intel

Ограничения

- При загрузке с образа диска МСВСфера ОС АРМ Kickstart-установка не поддерживается. Для установки МСВСфера ОС АРМ с помощью Kickstart-файла необходимо использовать или полный DVD (для локальной установки), или Netinstall (для установки по сети).

Программа установки Anaconda

Anaconda — программа установки операционной системы, использующаяся в МСВСфера ОС. **Anaconda** представляет собой набор модулей и Python-скриптов с дополнительными файлами, такими как виджеты **Gtk** (написанные на языке программирования `C`), модули **systemd** и библиотеки **dracut**. Вместе они образуют инструмент, позволяющий задавать параметры устанавливаемой операционной системы. В документации МСВСфера ОС термин «программа установки» обозначает программу установки **Anaconda**.

Методы установки

Вы можете установить МСВСфера ОС несколькими способами.

- Установка в графическом режиме. Необходимо скачать ISO-образ МСВСфера ОС и воспользоваться инструкцией по установке.
- Установка с помощью Kickstart-файла. Для частичной или полной автоматизации процесса установки МСВСфера ОС можно использовать Kickstart-файл, в котором содержатся параметры установки, например, часовой пояс, разделы диска и устанавливаемые пакеты. Использование Kickstart-файла позволяет установить МСВСфера ОС одновременно на большом количестве устройств без вмешательства пользователя.

Также использование Kickstart-файла даёт больше возможностей выбора программного обеспечения. При установке МСВСфера ОС вручную в графическом интерфейсе выбор программного обеспечения ограничивается предварительно заданными средами и надстройками. Используя Kickstart-файл, вы можете устанавливать или удалять отдельные пакеты.

Kickstart-файлы могут храниться на одном компьютере, а другие компьютеры могут читать их во время установки. То есть, один Kickstart-файл используется для установки МСВСфера ОС на нескольких компьютерах, что делает его идеальным для сетевых и системных администраторов.

Все Kickstart-сценарии и файлы журналов их выполнения хранятся в каталоге `/tmp` вновь установленной системы. Kickstart-файл, используемый для установки, а также выходные Kickstart-файлы, сгенерированные Anaconda, хранятся в каталоге `/root` в целевой системе, журналы выполнения Kickstart хранятся в каталоге `/var/log/anaconda`.

Примечание

Ранее с помощью Kickstart-файла также можно было настроить обновление системы. Теперь обновление выполняется с помощью специализированных инструментов.

Порядок автоматизированной установки

Автоматизированная установка может быть запущена с локального жёсткого диска, локального DVD или с сервера NFS, FTP, HTTP и HTTPS.

Порядок автоматизированной установки.

1. Создание Kickstart-файла.

2. Запись Kickstart-файла на локальный загрузочный носитель (DVD, USB, жёсткий диск) (или обеспечение доступа к нему по сети (NFS, FTP, HTTP(S)).
3. Предоставление доступа к установочной структуре.
4. Начало установки.

Создание Kickstart-файла

Вы можете создать Kickstart-файл, следующими способами:

- Использовать как есть или изменить Kickstart-файл, созданный в результате установки операционной системы.
- Создать Kickstart-файл вручную.

Примечание

Обратите внимание, что некоторые весьма специфические параметры установки можно настроить только путём создания или редактирования Kickstart-файла вручную.

Использование как есть или изменение Kickstart-файла, созданного в результате установки операционной системы

После завершения установки операционной системы все параметры, заданные во время установки, сохраняются в Kickstart-файле с именем `anaconda-ks.cfg`, расположенном в каталоге `/root` установленной системы. Вы можете использовать этот файл для установки операционной системы с точно такими же параметрами. Либо вы можете скопировать этот файл и внести необходимые изменения, а затем использовать его для установки операционной системы.

Порядок действий

1. Установите МСВСфера ОС. Во время установки создайте пользователя с правами администратора.
2. Перезагрузите устройство и войдите в установленную систему с правами администратора.
3. Скопируйте файл `/root/anaconda-ks.cfg`. Будьте внимательны — файл содержит информацию о пользователях и паролях!
    - Чтобы отобразить содержимое файла в терминале, выполните следующую команду:

```
$ sudo cat /root/anaconda-ks.cfg
```

Вы можете скопировать вывод команды и сохранить его в другой файл.

- Чтобы скопировать файл в другое место, используйте файловый менеджер. Вы можете изменить права доступа к копии, чтобы файл мог быть прочитан пользователями без полномочий `root`.

Создание Kickstart-файла вручную

Kickstart-файл — это обычный текстовый файл, содержащий параметры установки. Его можно создать в любом текстовом редакторе, способном сохранять файлы в формате ASCII.

При создании Kickstart-файла вручную рекомендуется руководствоваться разделами *Формат Kickstart-файла* и *Синтаксис команд*.

Проверка Kickstart-файла

После создания Kickstart-файла рекомендуется проверить его формат. Для этого используется программа `ksvalidator` в составе пакета `pykickstart`.

Установите пакет `pykickstart` с помощью следующей команды:

```
$ dnf install pykickstart
```

Чтобы проверить формат файла, выполните следующую команду:

```
$ ksvalidator /путь/к/kickstart.ks
```

Вместо `/путь/к/kickstart.ks` укажите действительный путь к Kickstart-файлу.

Предупреждение

Возможности такой проверки ограничены. Структура файла может быть достаточно сложной и, несмотря на то что `ksvalidator` проверяет синтаксис файла и помогает убедиться в отсутствии устаревших параметров, это не гарантирует успех установки. Более того, разделы `%pre`, `%post` и `%packages` вообще не проверяются.

Запись Kickstart-файла на локальный или удалённый носитель

В таблице перечислены порты, которые должны быть открыты на сервере для каждого типа сетевой установки.

Таблица 5: Порты для сетевой установки

Протокол	Порт
HTTP	80
HTTPS	443
FTP	21
NFS	2049, 111, 20048
TFTP	69

Основным преимуществом размещения Kickstart-файла в сети является то, что это позволяет установить МСВСфера ОС на несколько устройств без необходимости использовать физический носитель.

Запись и обеспечение доступа к Kickstart-файлу по NFS

Предварительные условия

- У вас есть доступ с правами администратора к серверу с МСВСфера ОС 9 в локальной сети.
- Устанавливаемая система может подключаться к серверу.
- Межсетевой экран на сервере разрешает подключения из устанавливаемой системы.

Порядок действий

1. Установите пакет `nfs-utils`, выполнив следующую команду от имени пользователя root.

   ```
   $ sudo dnf install nfs-utils
   ```

2. Положите Kickstart-файл в директорию на NFS-сервере.
3. Откройте файл `/etc/exports` в текстовом редакторе и добавьте следующую строку.

   ```
   /exported_directory/ clients
   ```

4. Здесь `/exported_directory/` — полный путь к каталогу, в котором находится Kickstart-файл.

   Вместо `clients` укажите имя хоста или IP-адрес компьютера, на который устанавливается МСВСфера ОС, подсеть, из которой все компьютеры

должны иметь доступ к ISO-образу или звёздочку (*), если хотите, чтобы любой компьютер с доступом к NFS-серверу мог использовать ISO-образ.

Пример: сделать каталог `/msvsphere9-install` доступным только для чтения для всех клиентов:

```
/msvsphere9-install/ *
```

5. Сохраните файл `/etc/exports`.
6. Запустите службу `nfs`.

```
$ sudo systemctl start nfs-server.service
```

Если служба была запущена до изменения файла `/etc/exports`, выполните следующую команду для перезагрузки конфигурации работающего NFS-сервера:

```
$ sudo systemctl reload nfs-server.service
```

Теперь Kickstart-файл доступен по NFS и готов к использованию для установки.

При указании источника установки как Kickstart-файл используйте следующие значения:

- `nfs:` в качестве протокола
- затем имя хоста или IP-адрес сервера
- знак двоеточия (:)
- путь внутри каталога, в котором находится файл

Например, если имя сервера — `myserver.example.ru` и вы сохранили файл в каталоге `/msvsphere9-install/my-ks.cfg`, то источник установки будет выглядеть следующим образом:

```
inst.ks=nfs:myserver.example.ru:/msvsphere9-install/my-ks.cfg
```

Запись и обеспечение доступа к Kickstart-файлу по HTTP(S)

Предварительные условия

- У вас есть доступ с правами администратора к серверу с МСВСфера ОС 9 в локальной сети.
- Устанавливаемая система может подключаться к серверу.
- Межсетевой экран на сервере разрешает подключения из устанавливаемой системы.

Порядок действий

1. Для HTTP-сервера установите пакет `httpd`, выполнив следующую команду от имени пользователя root.

```
$ sudo dnf install httpd
```

2. Для HTTPS-сервера установите пакеты `httpd` и `mod_ssl`, выполнив следующую команду от имени пользователя root.

```
$ sudo dnf install httpd mod_ssl
```

Предупреждение

Если в конфигурации веб-сервера Apache включена безопасность SSL, убедитесь, что включён только протокол TLSv1, а протоколы SSLv2 и SSLv3 отключены. Это связано с уязвимостью SSL POODLE (CVE-2014-3566).

Если HTTPS-сервер подписан самоподписанным сертификатом, используйте параметр `inst.noverifyssl` при установке.

3. Положите Kickstart-файл в директорию `/var/www/html/`.

Теперь Kickstart-файл доступен по HTTP(S) и готов к использованию для установки.

При указании источника установки как Kickstart-файл используйте следующие значения:

- `http://` или `https://` в качестве протокола
- затем имя хоста или IP-адрес сервера
- путь к Kickstart-файлу относительно корня HTTP(S)-сервера

Например, если имя сервера — `myserver.example.ru` и вы сохранили файл как `/var/www/html/msvsphere9-install/my-ks.cfg`, то источник установки будет выглядеть следующим образом:

```
http://myserver.example.com/msvsphere9-install/my-ks.cfg
```

Запись и обеспечение доступа к Kickstart-файлу по FTP

Предварительные условия

- У вас есть доступ с правами администратора к серверу с МСВСфера ОС 9 в локальной сети.
- Устанавливаемая система может подключаться к серверу.

- Межсетевой экран на сервере разрешает подключения из устанавливаемой системы.

Порядок действий

1. Установите пакет `vsftpd`, выполнив следующую команду от имени пользователя root.

```
$ sudo dnf install vsftpd
```

2. Откройте в текстовом редакторе конфигурационный файл `/etc/vsftpd/vsftpd.conf` и отредактируйте его следующим образом.

   - Измените строку `anonymous_enable=NO` на `anonymous_enable=YES`;
   - Измените строку `line write_enable=YES` на `write_enable=NO`;
   - Добавьте строки `pasv_min_port=min_port` и `pasv_max_port=max_port`. Замените `min_port` и `max_port` диапазоном номеров портов, используемых FTP-сервером в пассивном режиме, например, 10021 и 10031. Этот шаг может потребоваться в сетевых средах с различными настройками межсетевого экрана/NAT.
   - При желании вы можете добавить другие изменения в конфигурацию. Доступные параметры вы можете найти в руководстве `vsftpd.conf(5)`. Здесь же предполагается, что используются параметры по умолчанию.

> **Предупреждение**
>
> Если настроена безопасность SSL/TLS в файле `vsftpd.conf`, убедитесь, что включен только протокол TLSv1, а протоколы SSLv2 и SSLv3 отключены. Это связано с уязвимостью SSL POODLE (CVE-2014-3566).

3. Настройте межсетевой экран сервера.

- Включите межсетевой экран:

```
$ sudo systemctl enable firewalld
$ sudo systemctl start firewalld
```

- Включите в межсетевом экране FTP-порт и диапазон портов из предыдущего шага:

```
$ sudo firewall-cmd --add-port min_port-max_port/tcp --permanent
$ sudo firewall-cmd --add-service ftp --permanent
$ sudo firewall-cmd --reload
```

Замените `min_port-max_port` номерами портов, которые вы указали в файле конфигурации `/etc/vsftpd/vsftpd.conf`.

4. Скопируйте Kickstart-файл на FTP-сервер в каталог `/var/ftp/` или его подкаталог.

5. Убедитесь, что для файла установлен правильный контекст SELinux и режим доступа:

```
$ sudo restorecon -r /var/ftp/your-kickstart-file.ks
$ sudo chmod 444 /var/ftp/your-kickstart-file.ks
```

6. Запустите службу `vsftpd`.

```
$ sudo systemctl start vsftpd.service
```

Если служба была запущена до изменения файла `/etc/vsftpd/vsftpd.conf`, выполните следующую команду для перезапуска службы и загрузки отредактированного файла:

```
$ sudo systemctl restart vsftpd.service
```

Включите службу `vsftpd` для запуска во время процесса загрузки:

```
$ sudo systemctl enable vsftpd
```

Теперь Kickstart-файл доступен по FTP и готов к использованию для установки.

При указании источника установки как Kickstart-файл используйте следующие значения:

- `ftp://` в качестве протокола
- затем имя хоста или IP-адрес сервера
- путь к Kickstart-файлу относительно корня FTP-сервера

Например, если имя сервера — `myserver.example.ru` и вы сохранили файл в каталоге `/var/ftp/my-ks.cfg`, то источник установки будет выглядеть следующим образом:

```
ftp://myserver.example.ru/my-ks.cfg
```

Запись Kickstart-файла на локальный носитель и обеспечение доступа к нему

Категории локальных носителей:

- DVD, USB — установочный диск, созданный на основе ISO-образа.
- Жёсткий диск — установочный ISO-образ может располагаться на жёстком диске.

Предварительные условия

- У вас есть локальный носитель, который можно подключить к компьютеру для установки, например USB-накопитель или DVD-диск.
- На диске есть раздел, который может быть прочитан программой установки. Поддерживаемые типы: `ext2`, `ext3`, `ext4`, `xfs` и `fat`.
- Диск (локальный носитель) подключён к системе и его тома примонтированы.

Порядок действий

1. Узнайте информацию о томе и запишите UUID тома, на который хотите скопировать Kickstart-файл:

```
$ sudo lsblk -l -p -o name,rm,ro,hotplug,size,type,mountpoint,uuid
```

2. Перейдите к файловой системе тома.
3. Скопируйте туда Kickstart-файл.
4. Запишите строку, которая будет использоваться позже с опцией `inst.ks=`. Эта строка имеет вид `hd:UUID=volume-UUID:path/to/kickstart-file.cfg`. Обратите внимание, что путь указывается относительно корня файловой системы, а не `/` — корня иерархии файловой системы. Замените `Volume-UUID` на UUID, который вы узнали ранее.
5. Отмонтируйте все тома дисков:

```
$ sudo umount /dev/xyz ...
```

   Добавьте в команду все тома, разделённые пробелами.

Запись Kickstart-файла на локальный носитель и обеспечение доступа к нему для автоматической загрузки

Kickstart-файл с особым именем может располагаться в корне тома со специальным именем в устанавливаемой системе. В этом случае нет необходимости в другой системе, так как программа установки автоматически загрузит файл.

Предварительные условия

- У вас есть локальный носитель, который можно подключить к компьютеру для установки, например USB-накопитель или DVD-диск.
- На диске есть раздел, который может быть прочитан программой установки. Поддерживаемые типы: `ext2`, `ext3`, `ext4`, `xfs` и `fat`.
- Диск (локальный носитель) подключён к системе и его тома примонтированы.

Порядок действий

1. Узнайте информацию о томе, на который хотите скопировать Kickstart-файл:

   ```
   $ sudo lsblk -l -p
   ```

2. Перейдите к файловой системе тома.
3. Скопируйте Kickstart-файл в корневой каталог.
4. Переименуйте Kickstart-файл как `ks.cfg`.
5. Переименуйте том как `OEMDRV`:
   - Для файловой системы ext2, ext3 и ext4:

     ```
     $ sudo e2label /dev/xyz OEMDRV
     ```

   - Для файловой системы XFS:

     ```
     $ sudo xfs_admin -L OEMDRV /dev/xyz
     ```

     Где `/dev/xyz` — путь к устройству.
6. Отмонтируйте все тома дисков:

   ```
   $ sudo umount /dev/xyz ...
   ```

   Добавьте в команду все тома, разделённые пробелами.

Предоставление доступа к установочной инфраструктуре

Процесс установки должен иметь доступ к установочному образу МСВСфера ОС или к дереву установки, которое представляет собой копию МСВСфера ОС на DVD или USB-носителе.

Прежде чем приступить к установке, вставьте DVD-диск USB-носитель (см. раздел Установка и настройка системы). При выполнении установки с жёсткого диска надо будет разместить ISO-образ диска МСВСфера ОС на жёстком диске компьютера (см. раздел Установка и настройка системы). При выполнении сетевой установки (NFS, FTP, HTTP) потребуется открыть доступ к установочному образу DVD.

Начало установки

Запустить Kickstart-установку можно несколькими способами:

- Вручную, войдя в меню загрузки программы установки и указав там параметры использования Kickstart-файла.
- Автоматически, путём редактирования параметров загрузки при PXE-загрузке.
- Автоматически, поместив файл с определённым имененм в локальный том.

Запуск Kickstart-установки вручную

Запуск Kickstart-установки вручную означает, что администратору необходимо добавить параметр загрузки `inst.ks=location` в `boot:`, заменив `location` на путь до вашего Kickstart-файла.

Точный способ указания варианта загрузки и параметры загрузки зависят от архитектуры вашей системы.

Предварительные условия

- Готовый Kickstart-файл лежит в папке, доступной из устанавливаемой системы.

Порядок действий

1. Загрузите систему с локального носителя (DVD или USB).
2. Перейдите в параметры загрузки. Для этого при появлении экрана графической загрузки нажмите `Tab` при выбранном пункте меню **Установить МСВСфера 9.3= Сервер**. Внизу экрана графической установки появится строка параметров загрузки.
3. В конце строки параметров загрузки укажите необходимые сведения:

- Если Kickstart-файл или необходимый репозиторий находятся в сетевой папке, вам может потребоваться настроить сеть с помощью параметра `ip=`. Если параметр не используется, то по умолчанию программа установки будет настраивать все сетевые устройства с использованием протокола DHCP.
- Добавьте параметр загрузки `inst.ks=` и местоположение Kickstart-файла.
- Чтобы получить доступ к источнику программного обеспечения, из которого будут установлены необходимые пакеты, вам может потребоваться добавить параметр `inst.repo=`. Если вы не укажете этот параметр, то необходимо указать источник установки в Kickstart-файле.

4. Подтвердите добавление параметров и запустите установку. Установка начнётся с использованием параметров, указанных в Kickstart-файле. Если Kickstart-файл составлен правильно и содержит все необходимые команды, с этого момента установка будет полностью автоматизирована.

Примечание

При установке МСВСфера ОС в системах с включённой безопасной загрузкой UEFI, добавьте открытый ключ МСВСфера ОС в список ключей владельца компьютера (MOK).

Автоматический запуск Kickstart-установки с помощью PXE

Системы AMD64 и Intel 64 (а также 64-битные ARM и серверы IBM Power Systems) могут загружаться с помощью PXE-сервера. При настройке PXE-сервера вы можете добавить параметр загрузки в файл конфигурации загрузчика, что, в свою очередь, позволит запускать установку автоматически. Используя такой подход, можно полностью автоматизировать установку, включая процесс загрузки.

Инструкция приводится для общего ознакомления, так как действия различаются в зависимости от архитектуры системы, и не все параметры доступны для всех архитектур.

Предварительные условия

- Готовый Kickstart-файл лежит в папке, доступной из устанавливаемой системы.
- Имеется PXE-сервер, который можно использовать для загрузки системы и начала установки.

Порядок действий

1. Откройте файл конфигурации загрузчика на PXE-сервере и добавьте параметр загрузки `inst.ks=` в соответствующую строку. Имя файла и его синтаксис зависят от архитектуры и оборудования системы:

   - В системах AMD64 и Intel 64 с BIOS имя файла может быть задано по умолчанию или основано на IP-адресе системы. В этом случае добавьте параметр `inst.ks=` в строку добавления установки. Пример строки в файле конфигурации:

     ```
     append initrd=initrd.img inst.ks=http://10.32.5.1/mnt/archive/msvsphere-9/9.x/x86_64/
     ↪kickstarts/ks.cfg
     ```

   - В системах с загрузчиком GRUB2 (AMD64 и Intel 64, а также 64-битные ARM с UEFI и IBM Power Systems) имя файла будет `grub.cfg`. В этом файле добавьте параметр `inst.ks=` к строке ядра в записи установки. Пример строки ядра в файле конфигурации:

     ```
     kernel vmlinuz inst.ks=http://10.32.5.1/mnt/archive/msvsphere-9/9.x/x86_64/
     ↪kickstarts/ks.cfg
     ```

2. Запустите установку с сетевого сервера. Установка начнётся с использованием параметров, указанных в Kickstart-файле. Если Kickstart-файл составлен правильно и содержит все необходимые команды, установка будет полностью автоматизирована.

Примечание

При установке МСВСфера ОС в системах с включённой безопасной загрузкой

UEFI, добавьте открытый ключ МСВСфера ОС в список ключей владельца компьютера (MOK).

Автоматический запуск Kickstart-установки с использованием тома на локальном диске

Вы можете запустить Kickstart-установку, поместив Kickstart-файл с определённым именем в специально помеченный том на локальном диске.

Предварительные условия

- Определён том с меткой `OEMDRV` и в его корне лежит Kickstart-файл с именем `ks.cfg`.
- Диск с этим томом доступен в системе при загрузке программы установки.

Порядок действий

1. Загрузите систему с локального носителя (DVD или USB).
2. Перейдите в параметры загрузки. Для этого при появлении экрана графической загрузки нажмите `Tab` при выбранном пункте меню **Установить МСВСфера 9 Сервер**. Внизу экрана графической установки появится строка параметров загрузки.
3. В конце строки параметров загрузки укажите необходимые сведения:

- Если необходимый репозиторий находятся в сетевой папке, вам может потребоваться настроить сеть с помощью параметра `ip=`. Если параметр не используется, то по умолчанию программа установки будет настраивать все сетевые устройства с использованием протокола DHCP.
- Чтобы получить доступ к источнику программного обеспечения, из которого будут установлены необходимые пакеты, вам может потребоваться добавить параметр `inst.repo=`. Если вы не укажете этот параметр, то необходимо указать источник установки в Kickstart-файле.

4. Подтвердите добавление параметров и запустите установку. Kickstart-файл будет обнаружен автоматически и будет использоваться для Kickstart-установки.

Примечание

При установке МСВСфера ОС в системах с включённой безопасной загрузкой UEFI, добавьте открытый ключ МСВСфера ОС в список ключей владельца компьютера (MOK).

Ведение журнала событий во время установки

Программа установки МСВСфера ОС использует терминальный мультиплексор tmux для отображения и управления несколькими окнами в дополнение к основному интерфейсу. Каждое из этих окон служит разным целям: они отображают несколько разных журналов, которые можно использовать для устранения проблем в процессе установки. В одном из окон отображается интерактивное приглашение оболочки с правами `root` (если только это приглашение не было специально отключено с помощью параметра загрузки или команды в Kickstart-файле).

Примечание

Если нет необходимости диагностировать проблемы установки, то нет причин покидать графическую среду установки по умолчанию.

Терминальный мультиплексор работает в виртуальной консоли 1. Чтобы переключиться из фактической среды установки в tmux, нажмите `Ctrl` + `Alt` + `F1`. Чтобы вернуться к основному интерфейсу установки, который работает в виртуальной консоли 6, нажмите `Ctrl` + `Alt` + `F6`.

Примечание

Если вы проводите установку в текстовом режиме, то запускаете виртуальную консоль 1 (tmux), а при переключении на консоль 6 вместо графического интерфейса откроется приглашение оболочки.

Консоль, на которой запущен tmux, имеет пять доступных окон. Они описаны в таблице ниже вместе с сочетаниями клавиш. Обратите внимание, что сочетания клавиш состоят из двух частей: сначала вы нажимаете `Ctrl` + `b`, затем отпускаете обе клавиши и нажимаете цифровую клавишу для окна, которое хотите использовать.

Вы также можете использовать `Ctrl` + `b n`, `Alt` + `Tab` и `Ctrl` + `b p` для переключения на следующее или предыдущее окно tmux соответственно.

Таблица 6: Окна tmux

Сочетание клавиш	Окно
`Ctrl` + `b 1`	Главное окно программы установки. Содержит текстовые подсказки (во время установки в текстовом режиме или при использовании прямого режима VNC), а также некоторую отладочную информацию.
`Ctrl` + `b 2`	Интерактивная командная строка с правами `root`.
`Ctrl` + `b 3`	Журнал установки; отображает сообщения, хранящиеся в `/tmp/anaconda.log`.
`Ctrl` + `b 4`	Журнал устройств хранения; отображает сообщения, относящиеся к устройствам хранения и конфигурации и хранящиеся в `/tmp/storage.log`.
`Ctrl` + `b 5`	Журнал выполнения программ; отображает сообщения от утилит, выполняющих действия в процессе установки и хранящиеся в `/tmp/program.log`.

Формат Kickstart-файла

Kickstart-файл — это обычный текстовый файл, содержащий параметры установки, каждый из которых определяется ключевым словом (см. *Синтаксис команд*). Его можно создать в любом текстовом редакторе, способном сохранять файлы в формате ASCII — для этой цели подойдет блокнот в Windows, Gedit и vim в Linux или во встроенном редакторе в macOS.

Обычно вы можете задать любое имя для Kickstart-файла, однако рекомендуется использовать простое имя, так как это имя нужно будет указывать в других файлах или диалоговых окнах.

Команда — это ключевое слово, которое даёт указания программе установки. Каждая команда должна находиться в одной строке. Команда может быть как с параметрами, так и без. Указание команд и параметров аналогично использованию команд Linux в оболочке.

Некоторые специальные команды, начинающиеся с символа процента `%`, начинают **разделы**. Интерпретация команд в разделах отличается от интерпретации команд, размещённых вне их. Каждый раздел должен быть завершён директивой `%end`.

Типы разделов:

- Раздел надстроек. Начинается с директивы `%addon addon_name`.
- Раздел устанавливаемых пакетов. Начинается с директивы `%packages` В этом разделе можно указать отдельный пакет, группу или целое окружение.
- Разделы сценариев. Начинаются с директив `%pre`, `%pre-install`, `%post` и `%onerror`. Это необязательные разделы.

Разделы команд — этот термин используется для обозначения команд в Kickstart-файле, которые не являются частью какого-либо раздела сценария или раздела `%packages`.

Количество и порядок разделов скрипта

Все разделы, кроме раздела команд, являются необязательными и могут присутствовать несколько раз. Когда необходимо обработать раздел скрипта определённого типа, все разделы этого типа, представленные в Kickstart-файла, обрабатываются в порядке их появления: два раздела `%post` обрабатываются один за другим, в порядке их появления. Порядок следования разделов не важен — вы можете размещать их в любом порядке, например, не имеет значения, есть ли разделы `%post` перед разделами `%pre`.

Строки, начинающиеся с символа `#`, интерпретируются как **комментарии** и игнорируются программой установки.

Элементы, не отмеченные как обязательные, могут быть опущены. Если обязательный параметр не определён, программа установки предложит его ввести, так же как и

во время обычной установки. После получения ответа установка будет продолжена без вмешательства пользователя (если снова не столкнётся с отсутствующим параметром). С помощью команды `cmdline` можно прервать выполнение скрипта, если обязательные параметры не определены. В этом режиме любой отсутствующий ответ прерывает процесс установки.

Если в процессе Kickstart-установки в текстовом или графическом режиме требуется взаимодействие с пользователем, используйте только те окна, где взаимодействие обязательно для завершения установки. Чрезмерное взаимодействие может привести к сбросу Kickstart-конфигурации. В частности, сброс конфигурации применяется к Kickstart-командам, связанным с хранилищем, после входа в окно «Место установки».

Выбор пакетов

Команда *%packages`* отмечает начало разделов устанавливаемых пакетов. Здесь вы можете указать отдельный пакет, группу или целое окружение. Список групп и окружений можно найти в файле `repodata/*-comps-вариант.архитектура.xml` на установочном диске МВССфера ОС. Раздел `%packages` должен завершаться директивой `%end`.

В файле `*-comps-вариант.архитектура.xml` окружения отмечены тегом `<environment>`, а группы — тегом `<group>`. Каждая запись имеет свой идентификатор, признак отображения для пользователей, имя, описание и список пакетов. Пакеты, отмеченные как обязательные (`mandatory`), будут всегда устанавливаться при установке группы, пакеты `default` автоматически выбираются при выборе группы, если они специально не исключены, а пакеты `optional` должны быть заданы явно, даже если группа выбрана.

Группы и окружения можно выбрать по имени (тег `<name>`) и идентификатору (тег `<id>`).

Если вы не уверены, какой пакет следует установить, рекомендуется выбирать минимальную установку. В этом случае будут установлены только те пакеты, которые необходимы для работы МСВСфера ОС. Это существенно снизит вероятность возникновения уязвимости в системе. При необходимости дополнительные пакеты можно будет добавить после установки. Обратите внимание, что если в процессе установки из Kickstart-файла не была установлена система X Window и среда рабочего стола, а также не была включена графическая авторизация, то после перезагрузки этап первоначальной настройки будет пропущен.

Примечание

Чтобы установить 32-битный пакет в 64-битной системе, необходимо указать его имя и архитектуру, например `glibc.i686`. Также потребуется добавить параметр `--multilib` для раздела `%packages`.

Выбор окружения

Раздел `%packages` может содержать не только пакеты, но и целые окружения. Окружение необходимо указать в виде строки, начинающейся с символов `@^`.

Пример:

```
%packages
@^Infrastructure Server
%end
```

В примере будет устанвлено окружение `Infrastracture Server` и, соответственно, все пакеты, входящие в него. Список доступных окружений можно найти в файле `repodata/*-comps-вариант.архитектура.xml` на установочном диске МСВСфера ОС. Kickstart-файл должен содержать только одно окружение, если указано несколько окружений, то будет установлено последнее в списке.

Установка МСВСфера ОС АРМ

Для установки МСВСфера ОС АРМ с помощью Kickstart-файла (при выполнении условий, перечисленных в разделе *Ограничения*) необходимо указать следующее окружение:

```
%packages
@^workstation-product-environment
%end
```

Выбор групп

В каждой строке определяется по одной группе, начиная со знака `@`. Затем указываются полные имена или идентификаторы групп как в файле `*-comps-вариант.архитектура.xml`.

Пример:

```
%packages
@X Window System
@Desktop
@Sound and Video
%end
```

Основные группы пакетов (`Core`) будут выбраны по умолчанию, поэтому их не надо указывать в разделе `%packages`.

Выбор отдельных пакетов

Отдельные пакеты указываются по одному в строке по имени. Допускается использование шаблонов (*).

Пример:

```
%packages
sqlite
curl
aspell
docbook*
%end
```

Так, запись `docbook*` включает пакеты `docbook-dtds`, `docbook-style`, `docbook-slides` и т.п.

Выбор профилей с модулями

Профили для модулей указываются по одной записи в строке, с помощью синтаксиса профилей.

Пример:

```
%packages
@module:stream/profile
%end
```

Будут установлены все пакеты, перечисленные в указанном профиле модуля.

- Если для модуля указан поток по умолчанию, вы можете его не указывать. Если поток по умолчанию не указан, его необходимо указать.
- Если для потока модуля указан профиль по умолчанию, вы можете его не указывать. Если профиль по умолчанию не указан, его необходимо указать.
- Установка модуля несколько раз с разными потоками невозможна.
- Возможна установка нескольких профилей одного и того же модуля и потока.

Синтаксис для модулей и групп одинаковый, строка должна начинаться с символа `@`. Если модуль и группа пакетов имеют одинаковые имена, модуль имеет приоритет.

В МСВСфера ОС 9 модули есть только в репозитории `AppStream`. Чтобы просмотреть список доступных модулей, выполните следующую команду в установленной системе:

```
$ dnf module list
```

Также можно включить потоки модулей с помощью команды Kickstart `module`, а затем установить пакеты, содержащиеся в потоке модуля, указав их напрямую.

Исключение окружений, групп и пакетов

Добавление дефиса (-) перед именами пакетов и групп исключает их из списка установки.

Пример:

```
%packages
-@Graphical Administration Tools
-autofs
-ipa*compat
%end
```

Установка всех доступных пакетов с использованием только шаблона * в Kickstart-файле не поддерживается.

Вы можете контролировать поведение раздела `%packages` с помощью параметров. Некоторые параметры применимы для всех пакетов, другие — только с определёнными группами.

Общие параметры выбора пакетов

Ниже перечислены параметры раздела `%packages`. Параметры добавляются в конец заголовка раздела.

Пример:

```
%packages --multilib --ignoremissing
```

`--default`
: Установить набор пакетов по умолчанию. Соответствует набору пакетов, который был бы установлен, если бы во время обычной установки на экране выбора пакетов не было бы выбрано ничего.

`--excludedocs`
: Исключить документацию пакетов. В большинстве случаев это означает исключение файлов из `/usr/share/doc`.

`--ignoremissing`
: Игнорировать недостающие пакеты, группы, потоки модулей и окружения вместо того, чтобы приостановить установку и ожидать подтверждения продолжения.

`--inst-langs`
: Определяет список языков для установки. Обратите внимание, что это отличается от выбора на уровне группы пакетов. Этот параметр не описывает, какие группы пакетов следует установить; вместо этого он устанавливает макросы RPM, управляющие тем, какие файлы перевода из отдельных пакетов следует устанавливать.

--multilib

Разрешает установку 32-битных пакетов в 64-битных системах. Обычно на платформах AMD64 и Intel 64 по умолчанию устанавливаются пакеты, предназначенные для заданной архитектуры (например, x86_64) и стандартные пакеты (noarch). Этот параметр разрешает установку 32-битных пакетов (i686). Этот подход применим только к пакетам, явно указанным в разделе `%packages`. Если же пакет участвует в решении зависимостей и не указан в Kickstart-файле, будет установлена только необходимая версия. Вы можете настроить Anaconda для установки пакетов в режиме `--multilib` при установке системы. Вы можете сделать это одним из следующих способов:

- Добавьте в Kickstart-файл следующие строки:

```
%packages --multilib --default
%end
```

- Добавьте параметр загрузки `inst.multilib` во время загрузки установочного образа.

--nocore

Отключить установку группы пакетов `@Core`, которая всегда устанавливается по умолчанию. Отключение группы пакетов `@Core` с помощью `--nocore` следует использовать только для создания контейнеров; установка МСВСфера АРМ и Сервер с `--nocore` приведёт к непригодности системы.

- Невозможно исключить пакеты из группы пакетов `@Core` с помощью `-@Core`. Единственный способ исключить группу пакетов `@Core` — использовать опцию `--nocore`.
- В группу пакетов `@Core` входит минимальный набор пакетов, необходимых для установки работающей системы. Он никак не связан с основными пакетами.

--exclude-weakdeps

Отключить установку пакетов со слабыми зависимостями. Это пакеты, связанные с выбранным пакетом с помощью флагов `Recommends` и `Supplements`. По умолчанию будут установлены слабые зависимости.

--retries=

Задать число повторных попыток DNF загрузить пакеты. По умолчанию — 10. Этот параметр применяется только во время установки и не влияет на конфигурацию DNF в установленной системе.

--timeout=

Задать таймаут DNF в секундах. По умолчанию — 30. Этот параметр применяется только во время установки и не влияет на конфигурацию DNF в установленной системе.

Параметры для отдельных групп

Чтобы определить параметры для конкретной группы пакетов, надо добавить их после названия группы.

Пример:

```
%packages
@Graphical Administration Tools --optional
%end
```

`--nodefaults`
: Установить только обязательные пакеты, а не стандартный набор.

`--optional`
: Установить дополнительные пакеты группы согласно списку в `*-comps-вариант.архитектура.xml`. Обратите внимание, что для некоторых групп пакетов, таких как `Scientific Support`, не указаны обязательные пакеты или пакеты по умолчанию — только дополнительные пакеты. В этом случае всегда необходимо использовать опцию `--optional`, иначе пакеты из этой группы не будут установлены.

  Параметры `--nodefaults` и `--optional` нельзя использовать вместе. Вы можете установить только обязательные пакеты во время установки, используя `--nodefaults`, а дополнительные пакеты установить в систему после завершения установки.

Сценарии в Kickstart-файле

Kickstart-файл может включать следующие сценарии:

- `%pre`
- `%pre-install`
- `%post`

Ниже вы найдёте следующие сведения о сценариях:

- Время исполнения
- Типы команд, которые можно включить в сценарий
- Цель сценария
- Параметры сценария

Сценарий %pre

В разделе `%pre` определяются команды, которые будут выполняться сразу после обработки Kickstart-файла, но до начала установки. Этот раздел должен располагаться ближе к концу файла, начинаться с директивы `%pre` и заканчиваться директивой `%end`.

Сценарий `%pre` можно использовать для активации и настройки сетевых устройств и устройств хранения данных. Также можно запускать сценарии, используя интерпретаторы, доступные в среде установки. Добавление сценария `%pre` может оказаться полезным, если у вас есть сеть и хранилище, требующие специальной настройки перед продолжением установки, или если у вас есть сценарий, который, например, настраивает дополнительные параметры журналирования или переменные окружения.

Отладка проблем со сценариями `%pre` может быть затруднена, поэтому рекомендуется использовать их только при необходимости.

Раздел `%pre` Kickstart-файла выполняется на этапе установки, который происходит после получения образа установщика (`inst.stage2`), то есть после переключения root на среду установщика (образ установщика) и после запуска самого установщика Anaconda. Затем применяется конфигурация `%pre`, и её можно использовать для получения пакетов из установочных репозиториев, настроенных, например, по URL-адресу в Kickstart-файле. Однако его нельзя использовать для настройки сети для получения образа (`inst.stage2`) из сети.

Команды, связанные с сетью, хранилищем и файловыми системами, можно использовать в сценарии `%pre` в дополнение к большинству утилит в каталогах среды установки `/sbin` и `/bin`.

В этом разделе можно обращаться к сети, но так как на этом этапе служба имён еще не настроена, следует использовать IP-адреса.

Примечание

Сценарий `%pre` не выполняется в окружении `chroot`.

Ниже перечислены параметры `%pre`. Они добавляются в конец заголовка раздела:

```
%pre --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end
```

`--interpreter=`

Указать другой язык сценариев, например Python. Можно использовать любой язык, доступный в системе; в большинстве случаев это `/usr/bin/sh`, `/usr/bin/bash` и `/usr/libexec/platform-python`.

Обратите внимание, что интерпретатор платформы Python использует Python версии 3.9. Если вы используете Python-скрипты из предыдущих версий системы, необходимо изменить путь и версию. Кроме того, Python-платформа предназначена для системных инструментов: используйте пакет `python36` вне среды установки.

`--erroronfail`
: При сбое сценария показать сообщение об ошибке и прервать установку. Сообщение об ошибке показывает где зарегистрирована причина сбоя. Установленная система может стать нестабильной и не загружаться. Вы можете использовать параметр `inst.nokill` для отладки скрипта.

`--log=`
: Вывод сообщений сценария в заданный файл:

```
%pre --log=/tmp/ks-pre.log
```

Сценарий %pre-install

Команды сценария `%pre-install` запускаются после выполнения следующих задач:

- Система разбита на разделы.
- Файловые системы созданы и смонтированы в `/mnt/sysroot`.
- Сеть настроена в соответствии с параметрами загрузки и командами Kickstart-файла.

Раздел должен начинаться с директивы `%pre-install` и заканчиваться директивой `%end`.

Сценарий `%pre-install` можно использовать для изменения установки, а также для добавления пользователей и групп с заданными идентификаторами перед установкой пакетов.

Для изменений, необходимых в процессе установки, рекомендуется использовать сценарии `%post`. Используйте сценарий `%pre-install` только в том случае, если сценарий `%post` не позволяет внести необходимые изменения.

Примечание

Сценарий `%pre-install` не выполняется в окружении `chroot`.

Ниже перечислены параметры `%pre-install`. Они добавляются в конец заголовка раздела:

```
%pre-install --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end
```

Может быть несколько разделов `%pre-install` с одинаковыми или разными интерпретаторами. Они обрабатываются в порядке их появления в Kickstart-файле.

`--interpreter=`

Указать другой язык сценариев, например Python. Можно использовать любой язык, доступный в системе; в большинстве случаев это `/usr/bin/sh`, `/usr/bin/bash` и `/usr/libexec/platform-python`.

Обратите внимание, что интерпретатор платформы Python использует Python версии 3.6. Если вы используете Python-скрипты из предыдущих версий системы, необходимо изменить путь и версию. Кроме того, Python-платформа предназначена для системных инструментов: используйте пакет `python36` вне среды установки.

`--erroronfail`

При сбое сценария показать сообщение об ошибке и прервать установку. Сообщение об ошибке показывает где зарегистрирована причина сбоя. Установленная система может стать нестабильной и не загружаться. Вы можете использовать параметр `inst.nokill` для отладки скрипта.

`--log=`

Вывод сообщений сценария в заданный файл:

```
%pre-install --log=/mnt/sysroot/root/ks-pre.log
```

Сценарий %post

В разделе `%post` определяются команды, которые будут выполняться после установки, но до перезагрузки системы. Этот раздел должен располагаться после раздела команд ближе к концу файла, начинаться с директивы `%post` и заканчиваться директивой `%end`.

В этом разделе можно установить дополнительные программы или настроить сервер имён. Сценарий `%post` выполняется в окружении `chroot`, поэтому некоторые задачи (такие как копирование сценариев и пакетов с установочных носителей) выполнить не удастся. Это поведение можно изменить с помощью `--nochroot`. Тогда сценарий `%post` будет выполняться в среде установки, а не в `chroot` установленной системы.

Поскольку сценарий `%post` выполняется в среде `chroot`, большинство команд `systemctl` не выполняются.

Обратите внимание, что во время выполнения сценария `%post` установочный носитель должен быть вставлен.

Ниже перечислены параметры `%post`. Параметры добавляются в конец заголовка раздела:

```
%post --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end
```

`--interpreter=`

Указать другой язык сценариев, например Python. Можно использовать любой язык, доступный в системе; в большинстве случаев это `/usr/bin/sh`, `/usr/bin/bash` и `/usr/libexec/platform-python`.

Обратите внимание, что интерпретатор платформы Python использует Python версии 3.6. Если вы используете Python-скрипты из предыдущих версий системы, необходимо изменить путь и версию. Кроме того, Python-платформа предназначена для системных инструментов: используйте пакет `python36` вне среды установки. Пример:

```
%post --interpreter=/usr/libexec/platform-python
```

`--nochroot`

Указать команды, запускаемые вне среды `chroot`. В примере файл `/etc/resolv.conf` копируется в только что установленную файловую систему:

```
%post --nochroot
cp /etc/resolv.conf /mnt/sysroot/etc/resolv.conf
%end
```

`--erroronfail`

При сбое сценария показать сообщение об ошибке и прервать установку. Сообщение об ошибке показывает где зарегистрирована причина сбоя. Установленная система может стать нестабильной и не загружаться. Вы можете использовать параметр `inst.nokill` для отладки скрипта.

`--log=`

Сохраняет протокол выполнения сценария в файл. Путь должен учитывать наличие `--nochroot`. Пример без `--nochroot`:

```
%post --log=/root/ks-post.log
```

Пример с `--nochroot`:

```
%post --nochroot --log=/mnt/sysroot/root/ks-post.log
```

Ниже приведён пример сценария `%post`. В этом примере будет подключен внешний NFS-ресурс и запущен сценарий `runme` из `/usr/new-machines/`. Аргумент `-o nolock` в этом случае является обязательным, так как NFS-блокирование файлов в Kickstart-режиме не поддерживается.

```
# Начало раздела %post, журнал в /root/ks-post.log
%post --log=/root/ks-post.log

# Монтирование NFS-ресурса
mkdir /mnt/temp
mount -o nolock 10.10.0.2:/usr/new-machines /mnt/temp
openvt -s -w -- /mnt/temp/runme
umount /mnt/temp

# Конец раздела %post
%end
```

Раздел обработки ошибок

Начиная МСВСфера ОС 8, Kickstart-установка может содержать пользовательские сценарии, которые запускаются, когда программа установки обнаруживает фатальную ошибку. Например, ошибка в пакете, который был запрошен для установки, сбой при запуске VNC или ошибка при сканировании устройств хранения данных — после таких ошибок установка не может быть продолжена.

Сценарии `%onerror` выполняются в порядке следования в Kickstart-файле. Кроме того, сценарии `%onerror` выполняются в случае обратной отладки.

Раздел должен начинаться с директивы `%onerror` и заканчиваться директивой `%end`.

Параметры:

`--erroronfail`
: При сбое сценария показать сообщение об ошибке и прервать установку. Сообщение об ошибке показывает где зарегистрирована причина сбоя. Установленная система может стать нестабильной и не загружаться. Вы можете использовать параметр `inst.nokill` для отладки скрипта.

`--interpreter=`
: Указать другой язык сценариев, например Python. Можно использовать любой язык, доступный в системе; в большинстве случаев это `/usr/bin/sh`, `/usr/bin/bash` и `/usr/libexec/platform-python`. Обратите внимание, что интерпретатор платформы Python использует Python версии 3.6. Если вы используете Python-скрипты из предыдущих версий системы, необходимо изменить путь и версию. Кроме того, Python-платформа предназначена для системных инструментов: используйте пакет `python36` вне среды установки. Пример:

```
%onerror --interpreter=/usr/libexec/platform-python
```

`--log=`
: Вывод сообщений сценария в заданный файл.

Раздел надстроек

Начиная МСВСфера ОС 8, в Kickstart-установке поддерживаются надстройки. Такие надстройки могут довольно сильно расширить базовую функциональность как Kickstart-установки, так и в Anaconda.

Для подключения надстроек в Kickstart-файле раздел должен начинаться с директивы `%addon addon_name options` и заканчиваться директивой `%end`.

Например, если вы хотите использовать надстройку `Kdump`, которая по умолчанию поставляется вместе с Anaconda, используйте следующие команды:

```
%addon com_redhat_kdump --enable --reserve-mb=auto
%end
```

Команда `%addon` не имеет собственных параметров — все параметры определяются подключаемой надстройкой.

Синтаксис команд

Изменения синтаксиса

В целом, структура Kickstart-файла не изменилась с момента его первоначального выхода, но синтаксис команд и параметров время от времени подвергается изменениям.

Устаревшие команды и параметры

Команда `auth` (`authconfig`) устарела ещё в 8 версии МСВСфера ОС поскольку инструмент и пакет `authconfig` были удалены.

Подобно `authconfig` в командной строке, команды `authconfig` в Kickstart-сценариях теперь используют инструмент `authselect-compat` для запуска нового `authselect`. Описание совместимости и известных проблем можно прочитать в руководстве `authselect-migration(7)`. Программа установки автоматически обнаруживает устаревшие команды и устанавливает пакет `authselect-compat` для обеспечения совместимости.

Вы также можете адаптировать Kickstart-файлы, использовавшиеся в предыдущей версии операционной системы.

Следующие команды и параметры Kickstart-файла устарели в версии МСВСфера ОС 9.

- `timezone --ntpservers` (вместо неё используется команда `timesource`)
- `timezone --nontp`

- `logging --level`
- `%packages --excludeWeakdeps` (вместо этого используется `--exclude-weakdeps`)
- `%packages --instLangs` (вместо этого используется `--inst-langs`)
- `%anaconda`
- `pwpolicy` (вместо этого используйте файлы конфигурации Anaconda)
- `syspurpose` (вместо этого используется `subscription-manager syspurpose`)

Если указаны только определённые параметры, базовая команда и другие её параметры по-прежнему доступны и не устарели. При использовании устаревших команд в Kickstart-файле выводится предупреждение в журнале регистрации. Вы можете выводить предупреждения об ошибках с помощью параметра загрузки `inst.ksstrict`.

Удалённые команды и параметры

Следующие команды и параметры полностью удалены в МСВСфера ОС 9. Их использование в Kickstart-файле приводит к ошибкам.

- `device`
- `deviceprobe`
- `dmraid`
- `install` (используйте подкоманды или методы непосредственно как команды)
- `multipath`
- `bootloader --upgrade`
- `ignoredisk --interactive`
- `partition --active`
- `harddrive --biospart`
- `autostep`

Если указаны только определённые параметры и значения, базовая команда и другие её параметры по-прежнему доступны и не были удалены.

Команды настройки программы установки и управления потоком

cdrom (дополнительная)

Выполнить установку с первого оптического привода в системе.

Синтаксис

```
cdrom
```

Замечания

- У этой команды нет параметров.
- Чтобы запустить установку, необходимо указать один из следующих дисков: `cdrom`, `harddrive`, `hmc`, `nfs`, `liveimg`, `ostreesetup`, `rhsm`, или `url`, если в командной строке ядра не указана опция `inst.repo`.

cmdline (дополнительная)

Выполнить установку без взаимодействия с пользователем (в командной строке). При любом запросе на взаимодействие с пользователем установка будет прервана.

Синтаксис

```
cmdline
```

Замечания

- У этой команды нет параметров.
- Для полностью автоматической установки необходимо либо указать один из доступных режимов (`graphical`, `text`, `cmdline`). в Kickstart-файле, либо использовать параметр загрузки `console=`. Если режим не указан, система будет устанавливаться в графическом режиме, если это возможно, или предложит выбрать `VNC` или текстовый режим.

driverdisk (дополнительная)

Предоставить программе установки дополнительные драйверы. Диски с драйверами можно использовать во время Kickstart-установки для предоставления дополнительных драйверов, не включённых по умолчанию. Содержимое дисков с драйверами необходимо скопировать в корневой каталог раздела на жёстком диске системы. Затем вы должны использовать команду `driverdisk`, чтобы указать, что программа установки должна искать диск с драйверами и его местоположение.

Синтаксис

```
driverdisk [partition|--source=url|--biospart=biospart]
```

Параметры

Местоположение диска с драйверами необходимо указать одним из следующих способов:

- *partition* — раздел, содержащий диск с драйверами. Обратите внимание, что для раздела необходимо указать полный путь (например, `/dev/sdb1`), а не только имя раздела (например, `sdb1`).
- `--source=` — URL-адрес диска с драйверами. Пример:

  ```
  driverdisk --source=ftp://path/to/dd.img
  driverdisk --source=http://path/to/dd.img
  driverdisk --source=nfs:host:/path/to/dd.img
  ```

- `--biospart=` — раздел BIOS, содержащий диск с драйверами (например, `82p2`).

Замечания

Диски с драйверами также можно загружать с жёсткого диска или аналогичного устройства вместо загрузки по сети или из `initrd`. Порядок действий:

- Загрузите диск с драйверами на жёсткий диск, USB-накопитель или любое подобное устройство;
- Задайте метку этому устройству, например, `DD`;
- Добавьте следующую строку в Kickstart-файл:

  ```
  driverdisk LABEL=DD:/e1000.rpm
  ```

Вместо `DD` укажите метку, вместо `e1000.rpm` укажите имя. Используйте всё, что поддерживается командой `inst.repo`, вместо `LABEL`, чтобы указать ваш жёсткий диск.

eula (дополнительная)

Принять лицензионное соглашение с конечным пользователем (EULA) без взаимодействия с пользователем. Если параметр указан, то при первоначальной установке пользователю не будет предлагаться принять лицензионное соглашение после завершения установки и первой перезагрузки системы.

Синтаксис

```
eula [--agreed]
```

Параметры

- `--agreed` (обязательный) — принять лицензионное соглашение. Эту опцию необходимо использовать всегда, иначе команда `eula` не имеет смысла.

firstboot (дополнительная)

Определяет, запускать ли приложение начальной настройки при первой загрузке системы. Если параметр включён, необходимо установить пакет начальной установки. Если ничего не указано, опция по умолчанию отключена.

Синтаксис

```
firstboot OPTIONS
```

Параметры

- `--enable` или `--enabled` — начальная установка запускается при первой загрузке системы.
- `--disable` или `--disabled` — начальная установка не запускается при первой загрузке системы.
- `--reconfig` — включить начальную настройку для запуска во время загрузки в режиме реконфигурации. В этом режиме в дополнение к настройкам по умолчанию доступны параметры конфигурации пароля root, времени и даты, а также сети и имени хоста.

graphical (дополнительная)

Выполнять устновку в графическом режиме. Значение по умолчанию.

Синтаксис

```
graphical [--non-interactive]
```

Параметры

- `--non-interactive` — выполнять установку без взаимодействия с пользователем. Если требуется вмешательство пользователя, установка прекратится.

Замечания

- Для полностью автоматической установки без взаимодействия с пользователем необходимо указать в Kickstart-файле один из доступных режимов (`graphical`, `text`, `cmdline`) или использовать параметр загрузки `console=`. Если режим не указан, система будет использовать графический режим, если это возможно, или предложит выбрать VNC или текстовый режим.

halt (дополнительная)

Остановить систему после успешного завершения установки. Поведение команды похоже на поведение при установке вручную, когда Anaconda отображает сообщение и ожидает нажатия клавиши для перезагрузки. Если во время Kickstart-установки метод завершения не указан, то этот метод используется по умолчанию.

Синтаксис

```
halt
```

Замечания

- Команда `halt` аналогична команде `shutdown -H`.
- За дополнительной информацией обратитесь к описанию команд `poweroff`, `reboot` и `shutdown`.
- У этой команды нет параметров.

harddrive (дополнительная)

Выполнить установку из дерева установки или полного установочного ISO-образа на локальном диске. Диск должен быть отформатирован в файловой системе, которую может смонтировать программа установки: `ext2`, `ext3`, `ext4`, `vfat` и `xfs`.

Синтаксис

```
harddrive OPTIONS
```

Параметры

- `--partition=` — раздел с каталогом установки (например, `sdb2`).
- `--dir=` — DVD, ISO-образ или каталог, содержащий дерево установки.

Пример

```
harddrive --partition=hdb2 --dir=/tmp/install-tree
```

Замечания

- Ранее команда `harddrive` использовалась совместно с командой `install`. Команда `install` устарела и теперь команду `harddrive` можно использовать отдельно, поскольку она подразумевает установку.
- Чтобы фактически запустить установку, необходимо указать одно из значений: `cdrom`, `harddrive`, `hmc`, `nfs`, `liveimg`, `ostreesetup`, `rhsm`, `url`, если в командной строке ядра не указана опция `inst.repo`.

liveimg (дополнительная)

Выполнить установку из образа вместо пакетов.

Синтаксис

```
liveimg --url=SOURCE [OPTIONS]
```

Обязательные параметры

- `--url=` — путь к образу. Поддерживаемые протоколы: `HTTP`, `HTTPS`, `FTP` и `file`.
- `--dir=` — DVD, ISO-образ или каталог, содержащий дерево установки.

Дополнительные параметры

- `--proxy=` — определяет прокси `HTTP`, `HTTPS` или `FTP`.
- `--checksum=` (дополнительный параметр) — контрольная сумма SHA256.
- `--noverifyssl` — отключает проверку SSL при подключении к серверу `HTTPS`.

Пример

```
liveimg --url=file:///images/install/squashfs.img --
↪checksum=03825f567f17705100de3308a20354b4d81ac9d8bed4bb4692b2381045e56197 --noverifyssl
```

Замечания

- Образом может быть файл `squashfs.img` из живого ISO-образа, архив (`.tar`, `.tbz`, `.tgz`, `.txz`, `.tar.bz2`, `.tar.gz` или `.tar.xz`), или любая файловая система, которую может смонтировать установочный носитель. Поддерживаемые файловые системы: `ext2`, `ext3`, `ext4`, `vfat` и `xfs`.
- При использовании режима установки `liveimg` с диска с драйверами, находящиеся на диске драйверы не будут автоматически включены в устанавливаемую систему. При необходимости эти драйверы следует установить вручную или определить в разделе `%post` сценария Kickstart.
- Для фактического запуска установки необходимо указать один из следующих параметров: `cdrom`, `harddrive`, `hmc`, `nfs`, `liveimg`, `ostreesetup`, `rhsm` или `url`, если в командной строке ядра не указана опция `inst.repo`.

logging (дополнительная)

Отвечает за ведение записей в журнале ошибок Anaconda. Не влияет на установленную систему.

> **Примечание**
>
> Ведение журнала поддерживается только через TCP. Для удалённого ведения журнала убедитесь, что номер порта, указанный в опции `--port=`, открыт на удалённом сервере. Порт по умолчанию — 514.

Синтаксис

```
logging OPTIONS
```

Дополнительные параметры

- `--host=` — узел, на котором ведётся журнал. Чтобы использовать функции удалённого журналирования, необходимо, чтобы на узле выполнялся процесс `syslogd`.
- `--port=` — порт для подключения удалённого процесса `syslogd` (если не используется порт по умолчанию).

mediacheck (дополнительная)

Выполнить проверку носителя до начала установки. Так как проверка должна проводиться под наблюдением пользователя, по умолчанию параметр отключён.

Синтаксис

```
mediacheck
```

Замечания

- Команда `mediacheck` аналогична параметру загрузки `rd.live.check`.
- У этой команды нет параметров.

nfs (дополнительная)

Выполнить установку с указанного NFS-сервера.

Синтаксис

```
nfs OPTIONS
```

Параметры

- `--server=` — имя или IP-адрес сервера, с которого будет выполняться установка.
- `--dir=` — каталог дерева установки.
- `--opts=` (дополнительный) — параметры монтирования.

Пример

```
nfs --server=nfsserver.example.com --dir=/tmp/install-tree
```

Замечания

- Для фактического запуска установки необходимо указать один из следующих параметров: `cdrom`, `harddrive`, `hmc`, `nfs`, `liveimg`, `ostreesetup`, `rhsm` или `url`, если в командной строке ядра не указана опция `inst.repo`.

ostreesetup (дополнительная)

Настроить установку на основе OStree.

Синтаксис

```
ostreesetup --osname=OSNAME [--remote=REMOTE] --url=URL --ref=REF [--nogpg]
```

Обязательные параметры

- `--osname=OSNAME` — корень управления для установки ОС.
- `--url=URL` — URL-адрес репозитория для установки.
- `--ref=REF` — имя ветки репозитория, которая будет использоваться для установки.

Дополнительные параметры

- `--remote=REMOTE` — метостоположение репозитория.
- `--nogpg` — отключить проверку ключа GPG.

Замечания

- Дополнительную информацию об инструментах OStree вы можете найти документации: https://ostreedev.github.io/ostree/

poweroff (дополнительная)

После установки завершает работу системы и отключает питание компьютера. Обычно при установке вручную Anaconda выводит на экран сообщение и ожидает нажатия клавиши для перезагрузки.

Синтаксис

```
poweroff
```

Замечания

- Команда `poweroff` аналогична команде `shutdown -P`.
- Другие команды завершения: `halt`, `reboot` и `shutdown`. Если метод завершения явно не указан, по умолчанию выбирается `halt`.
- Функции `poweroff` сильно зависят от оборудования компьютера. В частности, необходимо, чтобы BIOS, APM (Advanced Power Management) и ACPI (Advanced Configuration and Power Interface) могли правильно взаимодействовать с ядром системы. Подробную информацию об APM и ACPI можно найти в документации оборудования.
- У этой команды нет параметров.

reboot (дополнительная)

Перезагрузить компьютер после успешного завершения установки (без аргументов). Обычно после Kickstart-установки на экране будет показано сообщение, и пользователь должен будет нажать клавишу для перезагрузки.

Синтаксис

```
reboot OPTIONS
```

Параметры

- `--eject` — прежде чем установленная система будет перезагружена, эта команда попытается извлечь установочный носитель (DVD, USB, и др.).
- `--kexec` — вместо полной перезагрузки использовать системный вызов `kexec`, который немедленно загружает установленную систему в память, минуя инициализацию оборудования, обычно выполняемую BIOS или прошивкой.

Важно

Эта команда признана устаревшей и доступна только в виде функции Technology Preview.

Важно

При использовании `kexec` регистры устройств (которые обычно очищаются во время полной перезагрузки системы) могут оставаться заполненными данными, что потенциально может создать проблемы для некоторых драйверов.

Замечания

- Команда `reboot` может привести к зацикливанию установки в зависимости от выбранного носителя и способа установки.
- Команда `reboot` аналогична команде `shutdown -r`.
- Другие методы завершения работы системы включают `halt`, `poweroff` и `shutdown`. Если в Kickstart-файле явно не указан режим завершения, по умолчанию будет использоваться `halt`.

shutdown (дополнительная)

Завершить работу системы после успешной установки.

Синтаксис

```
shutdown
```

Замечания

- Kickstart-команда `shutdown` аналогична команде `shutdown`.
- Другие методы завершения работы включают `halt`, `poweroff` и `reboot`. Если метод явно не указан, будет выбран `halt`.
- У этой команды нет параметров.

sshpw (дополнительная)

За ходом установки можно следить, удалённо подключившись к системе по `SSH`. С помощью команды `sshpw` можно создать временные учётные записи для подключения. Каждый вызов команды создает новую учётную запись в среде установки. После установки временные записи будут удалены.

Синтаксис

```
sshpw --username=name [OPTIONS] password
```

Обязательные параметры

- `--username=name` — обязательное имя пользователя.

- *password* — обязательный пароль для пользователя.

Дополнительные параметры

- `--iscrypted` — сообщает, что пароль зашифрован. Параметры `--plaintext` и `--iscrypted` являются взаимоисключающими. Зашифровать пароль можно следующим образом:

```
$ python3 -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if␣
↪(pw==getpass.getpass("Confirm: ")) else exit())'
```

  Эта команда создаст шифр sha512 с использованием заданной случайной строки (соли).

- `--plaintext` — сообщает, что пароль определён в открытом виде. `--plaintext` и `--iscrypted` являются взаимоисключающими.
- `--lock` — блокирует учётную запись, то есть пользователи не смогут авторизоваться из консоли.
- `--sshkey` — интерпретитровать строку `<password>` как значение ключа `ssh`.

Замечания

- Во время установки `ssh` по умолчанию не запускается. За активацию `ssh` отвечает параметр `inst.sshd`.
- Чтобы ограничить ssh-доступ для root (разрешив другим пользователям) на время установки, выполните следующую команду:

```
sshpw --username=example_username example_password --plaintext

sshpw --username=root example_password --lock
```

- Чтобы отключить ssh-доступ для root на время установки, выполните следующую команду:

```
sshpw --username=root example_password --lock
```

text (дополнительная)

Выполнить установку в текстовом режиме. По умолчанию установка выполняется в графическом режиме.

Синтаксис

```
text [--non-interactive]
```

Параметры

- `--non-interactive` — выполнив установку без взаимодействия с пользователем. Установка прекратится, когда потребуется вмешательство пользователя.

Замечания

- Для полностью автоматической установки необходимо либо указать один из доступных режимов (`graphical`, `text`, `cmdline`) в Kickstart-файле, либо использовать параметр загрузки `console=`. Если режим не указан, система будет устанавливаться в графическом режиме, если это возможно, или предложит выбрать VNC или текстовый режим.

url (дополнительная)

Выполнить установку из образа дерева установки на удаленном сервере с использованием протокола FTP, HTTP или HTTPS. Можно указать только один URL. Необходимо указать один из параметров `--url`, `--metalink` и `--mirrorlist`.

Синтаксис

```
url --url=FROM [OPTIONS]
```

Параметры

- `--url=FROM` — адрес файла установки или протокол: HTTP, HTTPS, FTP.
- `--mirrorlist=` — адрес зеркала.
- `--proxy=` — прокси-сервер HTTP, HTTPS или FTP.
- `--noverifyssl` — отключает проверку SSL при подключении к серверу HTTPS.
- `--metalink=URL` — URL-адрес metalink. Подстановка переменных производится для `$releasever` и `$basearch` в `URL`.

Примеры

- Установить с HTTP-сервера:

  ```
  $ url --url=http://server/path
  ```

- Установить с FTP-сервера:

  ```
  $ url --url=ftp://username:password@server/path
  ```

Замечания

- Чтобы запустить установку, вы должны указать одно из значений: `cdrom`, `hard-drive`, `hmc`, `nfs`, `liveimg`, `ostreesetup`, `rhsm`, `url`, если в командной строке ядра не указана опция `inst.repo`.

vnc (дополнительная)

Разрешает удалённое подключение к графической установке с помощью VNC, что более предпочтительно по сравнению с вынужденным переходом в текстовый режим в силу его ограничений по размеру и языку. Если команда используется без параметров, VNC-сервер будет запущен без пароля, и на экране появится сообщение с информацией о подключении к удалённому компьютеру.

Синтаксис

```
vnc [--host=host_name] [--port=port] [--password=password]
```

Параметры

- `--host=` — подключение к клиенту VNC, прослушивающему заданный узел.
- `--port=` — порт, который будет прослушиваться клиентом VNC. Если не задан, Anaconda использует порт VNC по умолчанию — 5900.
- `--password=` — пароль доступа к сеансу VNC. Этот параметр не является обязательным, но рекомендуется его задать.

%include (дополнительная)

Включить содержимое другого файла в Kickstart-файл, как если бы это содержимое находилось в месте указания команды `%include`. Такие включения будут обрабатываться после раздела `%pre`, что может использоваться для включения файлов, созданных сценариями, в разделы `%pre`.

Для включения файлов до обработки раздела `%pre` используется команда `%ksappend`.

Синтаксис

```
%include path/to/file
```

%ksappend (дополнительная)

Включить содержимое другого файла в Kickstart-файл, как если бы это содержимое находилось в месте указания команды `%ksappend`. Такие включения будут обрабатываться перед разделом `%pre`, в отличие от включения с помощью команды `%include`.

Синтаксис

```
%ksappend path/to/file
```

Команды настройки системы

Команды используются для настройки дополнительных сведений о системе, таких как пользователи, репозитории или службы.

auth или authconfig (устарела)

Предупреждение

Вместо устаревшей команды `auth` или `authconfig` используйте команду `authselect`. Команды `auth` и `authconfig` доступны в ограниченной функциональности для обеспечения обратной совместимости.

Определяет параметры аутентификации в системе. Аналогичные функции выполняет команда `authconfig`, которую можно выполнить после установки.

Синтаксис

```
authconfig [OPTIONS]
```

Замечания

- Раньше команды `auth` и `authconfig` были инструментом `authconfig`. Этот инструмент устарел. Эти команды теперь используют инструмент `authselect-compat` для вызова нового инструмента `authselect`. Программа установки автоматически обнаружит использование устаревших команд и установит в системе пакет `authselect-compat` для обеспечения совместимости.
- По умолчанию пароли скрыты.
- При использовании OpenLDAP с протоколом SSL в целях безопасности убедитесь, что протоколы SSLv2 и SSLv3 отключены в конфигурации сервера. Это связано с уязвимостью SSL POODLE (CVE-2014-3566).

authselect (дополнительная)

Определяет параметры аутентификации в системе. Команду `authselect` также можно запустить из командной строки после завершения установки.

Синтаксис

```
authselect [OPTIONS]
```

Замечания

- Передаёт все параметры команде `authselect`. Дополнительную информацию можно получить, выполнив команду `authselect --help`.
- Заменяет устаревшие команды `auth` и `authconfig` (вместе с инструментом `authconfig`).
- По умолчанию пароли скрыты.
- При использовании OpenLDAP с протоколом SSL в целях безопасности убедитесь, что протоколы SSLv2 и SSLv3 отключены в конфигурации сервера. Это связано с уязвимостью SSL POODLE (CVE-2014-3566).

firewall (дополнительная)

Отвечает за настройку межсетевого экрана.

Синтаксис

```
firewall --enabled|--disabled [тип] [OPTIONS]
```

Обязательные параметры

- `--enabled`, `--enable` — отклоняет входящие подключения, которые не отвечают на исходящие запросы, например ответы DNS или запросы DHCP. Если необходим доступ к службам, запущенным на этом компьютере, можно его включить в настройках межсетевого экрана.
- `--disabled`, `--disable` — отключает настройку правил iptables.

Дополнительные параметры

- `--trust=` — доверенное устройство, через которое может проходить трафик. Чтобы указать несколько устройств, используйте формат `--trust em1 --trust em2`. Не разделяйте устройства запятыми. Неверное указание: `--trust em1, em2`.
- `--remove-service` — не разрешать службам проходить через межсетевой экран.
- *тип* — это значение можно заменить типом соединений, которые будут пропускаться через экран:
  - `--ssh`
  - `--smtp`
  - `--http`
  - `--ftp`
- `--port=` — список открытых портов в формате `порт:протокол`. Например, чтобы разрешить доступ IMAP через межсетевой экран, укажите `imap:tcp`. Номер порта можно указать явно — например, чтобы разрешить прохождение

пакетов UDP через порт 1234, введите `1234:udp`. Несколько портов разделяются запятыми.

- `--service=` — пропускает трафик указанной службы через межсетевой экран. Некоторые службы, такие как `cups` и `avahi`, используют несколько портов, поэтому вместо перечисления списка портов и других специальных параметров можно указать имя службы и открыть для неё все порты.

  Допустимые значения включают службы, которые распознаёт `firewall-offline-cmd`. Если в системе выполняется firewalld, для получения списка служб выполните `firewall-cmd --get-services`.

- `--use-system-defaults` — вообще не настраивать межсетевой экран. Этот параметр предписывает Anaconda ничего не делать и позволяет системе полагаться на значения по умолчанию, предоставленные пакетом или `ostree`. Если параметр используется совместно с другими параметрами, то все другие будут игнорироваться.

group (дополнительная)

Позволяет создать новую группу пользователей.

Синтаксис

```
group --name=name [--gid=gid]
```

Обязательные параметры

- `--name=` — имя группы.

Дополнительные параметры

- `--gid=` — идентификатор группы. Если не задан, по умолчанию используется следующий свободный идентификатор.

Замечания

- Если группа с заданным именем или идентификатором уже существует, команда завершится ошибкой.
- Чтобы создать группу для нового пользователя, можно использовать эту команду совместно с `user`.

keyboard (обязательная)

Выбор раскладки клавиатуры.

Синтаксис

```
keyboard --vckeymap|--xlayouts OPTIONS
```

Дополнительные параметры

- `--vckeymap=` — определяет раскладку VConsole. В качестве значения можно выбирать имена файлов из директории `/usr/lib/kbd/keymaps/xkb/` без расширения `.map.gz`.
- `--xlayouts=` — список раскладок X через запятую без пробелов. Значения можно определить в форме `раскладка` (например, `cz`) или `раскладка (вариант)` (например, `cz (qwerty)`), что аналогично формату `setxkbmap(1)`. Полный список раскладок можно найти в справке `xkeyboard-config(7)`, в разделе `Layouts`.
- `--switch=` — комбинации клавиш для переключения между раскладками в формате, аналогичном `setxkbmap(1)`. Несколько значений разделяются запятыми без пробелов. Описание дополнительных параметров можно найти в справке `xkeyboard-config(7)`, в разделе `Options`.

Замечания

- `--vckeymap=` и `--xlayouts=` не должны использоваться вместе.

Пример

- Ниже будут настроены две раскладки — английская (США) и русская — переключение между которыми осуществляется с помощью `Alt+Shift`:

  ```
  $ keyboard --xlayouts=us,'ru (qwerty)' --switch=grp:alt_shift_toggle
  ```

lang (обязательная)

Определяет язык, который будет использоваться в ходе установки и по умолчанию — в установленной системе.

Синтаксис

```
lang language [--addsupport=language,...]
```

Обязательные параметры

- `language` — определяет язык установки и по умолчанию в установленной системе.

Дополнительные параметры

- `--addsupport=` — список дополнительных языков через запятую без пробелов. Пример:

```
lang en_US --addsupport=ru_RU,de_DE,en_UK
```

Замечания

- Для вывода списка поддерживаемых языков выполните следующие команды `locale -a | grep _` или `localectl list-locales | grep _`.
- Некоторые языки, в частности китайский, японский, корейский и языки индийской группы, не поддерживаются в текстовом режиме установки. Если такого рода язык задан с помощью команды `lang`, установка будет продолжена на английском, а указанный язык будет включён по умолчанию в готовой системе.

Пример

Выбор русского языка в Kickstart-файле:

```
lang ru_RU
```

module (дополнительная)

Включает поток модуля пакета в Kickstart-сценарий.

Синтаксис

```
module --name=NAME [--stream=STREAM]
```

Обязательные параметры

- `--name=` — имя подключаемого модуля. Замените NAME реальным именем.

Дополнительные параметры

- `--stream=` — имя подключаемого потока модуля. Замените STREAM реальным именем. Вам не нужно указывать этот параметр для модулей с определённым потоком по умолчанию. Для модулей без потока по умолчанию этот параметр является обязательным, и его отсутствие приведёт к ошибке. Включение модуля несколько раз с разными потоками невозможно.

Замечания

- Использование комбинации этой команды и раздела `%packages` позволяет устанавливать пакеты, предоставляемые включённой комбинацией модуля и потока, без явного указания модуля и потока. Модули должны быть включены перед установкой пакета. После включения модуля с помощью команды `module` вы можете установить пакеты, включённые этим модулем, перечислив их в разделе `%packages`.

- Одна команда `module` может включить только одну комбинацию модуля и потока. Чтобы включить несколько модулей, используйте команды для нескольких модулей. Включение модуля несколько раз с разными потоками невозможно.
- Модули располагаются только в репозитории AppStream. Чтобы просмотреть список доступных модулей, используйте команду `dnf module list` в установленной операционной системе.

repo (дополнительная)

Позволяет настроить дополнительные репозитории dnf. Вы можете добавить несколько строк `repo`.

Синтаксис

```
repo --name=repoid [--baseurl=url|--mirrorlist=url|--metalink=url] [OPTIONS]
```

Обязательные параметры

- `--name=` — обязательный идентификатор репозитория. Если репозиторий уже был добавлен, он будет пропущен. Программа установки использует заранее подготовленный список репозиториев, поэтому имена дополнительных репозиториев не должны с ними совпадать.

URL-параметры

Эти параметры являются взаимоисключающими и необязательными. Переменные, которые можно использовать в файлах конфигурации репозитория dnf, здесь не поддерживаются. Вы можете использовать строки `$releasever` и `$basearch`, которые заменяются соответствующими значениями в URL-адресе.

- `--baseurl=` — URL репозитория.
- `--mirrorlist=` — URL списка зеркал для репозитория.
- `--metalink=` — URL с металинком для репозитория.

Дополнительные параметры

- `--install` — сохранить конфигурацию репозитория в установленной системе в каталоге `/etc/yum.repos.d/`. Без использования этой опции репозиторий, настроенный в Kickstart-файле, будет доступен только во время установки, а не в установленной системе.
- `--cost=` — целое значение, определяющее приоритет репозитория. Используется для выбора репозитория, если интересующий пакет входит в состав нескольких репозиториев. В этом случае будет выбран репозиторий с минимальным значением.

- `--excludepkgs=` — список пакетов через запятую, которые будут исключены при загрузке из репозитория. Обычно используется, если один пакет входит в состав нескольких репозиториев, и вы хотите исключить одну из версий. В качестве значений принимаются полные имена пакетов (например, `publican`) и шаблоны (например, `gnome-*`).
- `--includepkgs=` — список пакетов и шаблонов их имен через запятую, которые будут загружены из репозитория. Используется, если один пакет входит в состав нескольких репозиториев, и вы хотите загрузить конкретную версию из определенного репозитория.
- `--proxy=[protocol://][username[:password]@]host[:port]` — прокси-сервер HTTP, HTTPS или FTP для доступа к репозиторию. Этот параметр не меняет способ загрузки `install.img` при выполнении установки с HTTP-сервера и не влияет на другие репозитории.
- `--noverifyssl` — отключает проверку SSL при подключении к серверу HTTPS.

Замечания

- Состояние репозиториев должно оставаться неизменным во время установки, так как их модификация может нарушить ход установки.

rootpw (обязательная)

Позволяет установить пароль root.

Синтаксис

```
rootpw [--iscrypted|--plaintext] [--lock] password
```

Обязательные параметры

- `password` — пароль. Может быть как обычный текст, так и зашифрованная строка. См. `--iscrypted` и `--plaintext` ниже.

Дополнительные параметры

- `--iscrypted` — сообщает, что пароль зашифрован. Параметры `--plaintext` и `--iscrypted` являются взаимоисключающими. Зашифровать пароль можно следующим образом:

  ```
  $ python -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if␣
  ↪(pw==getpass.getpass("Confirm: ")) else exit())'
  ```

  Эта команда создаст шифр SHA512 с использованием заданной случайной строки (соли).
- `--plaintext` — сообщает, что пароль определён в открытом виде. Параметры `--plaintext` и `--iscrypted` являются взаимоисключающими.

- `--lock`— блокирует учётную запись. Авторизация пользователя `root` из консоли будет недоступна. Также будут недоступны экраны пароля `root` как при графической, так и при текстовой установке вручную.
- `--allow-ssh` — пользователь `root` может войти в систему по SSH с паролем. Этот параметр доступен только в МСВСфера ОС 9.1 и более поздних версиях. Чтобы включить root-вход по SSH на основе пароля, добавьте следующую строку в Kickstart-файл. Параметр `--allow-ssh` недоступен в МСВСфера ОС 9.0.

```
%post
echo "PermitRootLogin yes" > /etc/ssh/sshd_config.d/01-permitrootlogin.conf
%end
```

selinux (дополнительная)

Определяет режим SELinux (по умолчанию будет выбран строгий режим).

Синтаксис

```
selinux [--disabled|--enforcing|--permissive]
```

Дополнительные параметры

- `--enforcing` — строгий режим SELinux.
- `--permissive` — режим предупреждений.
- `--disabled` — полностью отключает SELinux.

services (дополнительная)

Список служб, выполняемых в рамках `systemd`. Список отключённых служб проверяется первым, а включённых — вторым. Таким образом, если служба присутствует в обоих списках, она будет включена.

Синтаксис

```
services [--disabled=list] [--enabled=list]
```

Дополнительные параметры

- `--disabled=` — отключает службы, перечисленные через запятую.
- `--enabled=` — включает службы, перечисленные через запятую.

Замечания

- Службы должны быть перечислены через запятую без пробелов. Если строка содержит пробелы, будут обработаны только элементы до первого пробела. Пример:

```
services --disabled=auditd, cups,smartd, nfslock
```

В этом примере будет отключена только служба `auditd`. Для отключения всех служб удалите пробел:

```
services --disabled=auditd,cups,smartd,nfslock
```

skipx (дополнительная)

Пропускает настройку системы X.

Если вы устанавливаете дисплейный менеджер среди параметров выбора пакета, этот пакет создаёт конфигурацию X, а установленная система по умолчанию использует `graphical.target`. Это переопределяет эффект параметра `skipx`.

Синтаксис

```
skipx
```

Замечания

- У этой команды нет параметров.

sshkey (дополнительная)

Добавить ключ SSH в файл `authorized_keys` указанного пользователя в установленной системе.

Синтаксис

```
sshkey --username=user "ssh_key"
```

Обязательные параметры

- `--username=` — пользователь, для которого будет установлен ключ.
- `ssh_key` — полный отпечаток ключа SSH. Должен быть в кавычках.

timezone (обязательная)

Устанавливает часовой пояс.

Синтаксис

```
timezone timezone [OPTIONS]
```

Обязательные параметры

- `timezone` — часовой пояс.

Дополнительные параметры

- `--utc` — если задан, подразумевается, что системные часы установлены по Гринвичу.
- `--nontp` — отключает автоматический запуск NTP. Параметр признан устаревшим.
- `--ntpservers=` — список NTP-серверов через запятую без пробелов. Параметр признан устаревшим, вместо него используется команда `timesource`.

Замечания

- В МСВСфера ОМ 9 названия часовых поясов проверяются с помощью списка `pytz.common_timezones`, предоставляемого пакетом `pytz`.

timesource (дополнительная)

Команда используется для настройки NTP, серверов NTS и пулов, которые предоставляют данные о времени, а также для управления включением или отключением служб NTP в системе.

Синтаксис

```
timesource [--ntp-server NTP_SERVER | --ntp-pool NTP_POOL | --ntp-disable] [--nts]
```

Обязательные параметры

При использовании команды `timesource` обязательно укажите один из следующих параметров:

- `--ntp-server` — добавляет один NTP-сервер в качестве источника времени. Этот параметр можно добавить только один раз в одну команду, чтобы добавить один NTP-сервер. Чтобы добавить несколько источников, добавьте несколько команд `timesource``с параметром ``--ntp-server` или `--ntp-pool`. Например, добавим несколько источников для России:

  ```
  timezone Russia
  timesource --ntp-server ntp.msk-ix.ru
  timesource --ntp-server ntp1.vniiftri.ru
  timesource --ntp-server ntp.sstf.nsk.ru
  ```

- `--ntp-pool` — добавляет пул NTP-серверов в качестве источника времени. Этот параметр можно добавить только один раз, чтобы добавить один NTP-пул. Чтобы добавить несколько источников, добавьте несколько команд `timesource` с параметром.
- `--ntp-disable` отключает источники времени NTP для установленной системы.

Дополнительные параметры

- `--nts` — сервер или пул, добавленный с помощью этой команды, использует протокол NTS. Обратите внимание, что этот параметр можно добавить даже с `--ntp-disable`, но он не даст никакого эффекта.

Замечания

- Параметр `--ntpservers` устарел. Рекомендуется использовать команду `timesource`.
- Только команда `timesource` может пометить серверы и пулы как использующие `NTS` вместо обычного протокола `NTP`.

user (дополнительная)

Создание пользователя.

Синтаксис

```
user --name=username [OPTIONS]
```

Обязательные параметры

- `--name=` — имя пользователя.

Дополнительные параметры

- `--gecos=` — строка параметров GECOS через запятую. Типичные параметры включают полное имя пользователя, номер офиса, и т.п. Подробную информацию можно найти в справке `passwd(5)`.
- `--groups=` — список групп пользователя, разделённых запятой. Список должен содержать существующие группы (см. описание команды `group`).
- `--homedir=` — домашний каталог пользователя. По умолчанию используется `/home/пользователь`.
- `--lock` — блокирует учётную запись, то есть пользователи не смогут авторизоваться из консоли. Параметр также отключает экран «Создание пользователя» как при графической, так и при текстовой установке вручную.
- `--password=` — пароль пользователя. Если не задан, учётная запись будет заблокирована.
- `--iscrypted` — сообщает, что пароль зашифрован. Параметры `--plaintext` и `--iscrypted` являются взаимоисключающими. Зашифровать пароль можно следующим образом:

```
$ python -c 'import crypt,getpass;pw=getpass.getpass();print(crypt.crypt(pw) if␣
↪(pw==getpass.getpass("Confirm: ")) else exit())'
```

Эта команда создаст шифр SHA512 с использованием заданной случайной строки (соли).

- `--plaintext` — сообщает, что пароль определён в открытом виде. `--plaintext` и `--iscrypted` являются взаимоисключающими.
- `--shell=` — оболочка пользователя. По умолчанию используются системные настройки.
- `--uid=` — идентификатор пользователя. По умолчанию будет использоваться следующий свободный идентификатор.
- `--gid=` — идентификатор группы. По умолчанию используется следующий свободный идентификатор.

Замечания

- Рассмотрите возможность использования параметров `--uid` и `--gid` для установки идентификаторов пользователей и их групп по умолчанию в диапазоне, начиная с `5000` вместо `1000`. Это связано с тем, что диапазон, зарезервированный для системных пользователей и групп, `0–999`, может увеличиться в будущем и, таким образом, совпасть с идентификаторами обычных пользователей.
- Файлы и каталоги создаются с различными полномочиями, определяемыми приложением, используемым для их создания. Например, команда `mkdir` создаёт каталоги со всеми включёнными разрешениями. Однако приложениям не разрешается предоставлять определённые разрешения вновь создаваемым файлам, как указано в настройке `user file-creation mask`.

  Настройкой `user file-creation mask` можно управлять с помощью команды `umask`. Настройка по умолчанию `user file-creation mask` для новых пользователей определяется переменной `UMASK` в файле конфигурации `/etc/login.defs` в установленной системе. Если этот параметр не указан, по умолчанию используется значение `022`. Это означает, что по умолчанию, когда приложение создает файл, ему не разрешается предоставлять разрешение на запись пользователям, кроме владельца файла. Однако это поведение может быть переопределено другими настройками или сценариями.

xconfig (дополнительная)

Настраивает X Window System.

Синтаксис

```
xconfig [--startxonboot]
```

Дополнительные параметры

- `--startxonboot` — графическая авторизация в установленной системе.

Замечания

- Так как в пакет поставки не входит среда рабочего стола KDE, не используйте параметр `--defaultdesktop=`.

Команды настройки сети

Команды используются для настройки сети в системе.

network (дополнительная)

Настройка сетевого окружения. По умолчанию будет включено первое указанное в команде устройство. Активацию конкретного устройства можно явно запросить с помощью `--activate`.

Синтаксис

```
network OPTIONS
```

Параметры

- `--activate` — активировать устройство в среде установки. Если этот параметр указан применительно к уже активному устройству (например, интерфейс, настроенный с параметрами загрузки, чтобы система могла получить Kickstart-файл), будет выполнена его повторная активация согласно настройкам в Kickstart-файле.

  Используйте параметр `--nodefroute`, чтобы запретить устройству использовать маршрут по умолчанию.

- `--no-activate` — запрет активации устройства в среде установки. По умолчанию Anaconda активирует первое сетевое устройство в Kickstart-файле независимо от параметра `--activate`. Вы можете отключить настройку по умолчанию, используя опцию `--no-activate`.

- `--bootproto=` — может принимать одно из следующих значений: `dhcp` (используется по умолчанию), `bootp`, `ibft`, `static`. Параметры `dhcp` и `bootp` обрабатываются одинаково. Для отключения настройки `ipv4` устройства используйте параметр `--noipv4`. Этот параметр настраивает `ipv4` устройства. Для настройки `ipv6` используйте параметры `--ipv6` и `--ipv6gateway`.

  В режиме DHCP для получения сетевых настроек используется DHCP-сервер. Как можно догадаться, в случае с BOOTP для получения сетевых настроек подобным образом используется сервер BOOTP. Чтобы настроить функции DHCP, добавьте:

```
network --bootproto=dhcp
```

Чтобы настроить получение сетевой конфигурации с помощью BOOTP, добавьте в Kickstart-файл:

```
network --bootproto=bootp
```

Чтобы использовать конфигурацию из iBFT, добавьте строку:

```
network --bootproto=ibft
```

Для метода `static` в Kickstart-файле необходимо определить как минимум IP-адрес и маску сети (но также рекомендуется определить шлюз и сервер имён). Эти данные не изменяются и используются во время и после установки.

Все данные статической конфигурации должны быть указаны в одной строке, вы не можете переносить строки с помощью символа `\`, как в командной строке.

```
network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.
↪0.2.254 --nameserver=10.0.2.1
```

Здесь же можно настроить несколько серверов имён одновременно с помощью параметра `--nameserver=`, указав их IP-адреса через запятую:

```
network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.
↪0.2.254 --nameserver=192.168.2.1,192.168.3.1
```

- `--device=` — сетевое устройство, которое необходимо настроить (и, в конечном итоге, активировать в Anaconda). Если в первом вхождении команды `network` не указан параметр `--device=`, по умолчанию будет использоваться значение параметра загрузки `inst.ks.device=`. Это поведение считается устаревшим, поэтому рекомендуется явно определить значение `--device=`.

> **Примечание**
>
> Агрегирование (Team) устарело в МСВСфера ОС 9. В качестве альтернативы рассмотрите возможность использования Агрегирования (Bond).

Для всех последующих команд `network` необходимо явно определить `--device=`. Устройства могут быть заданы любым из следующих способов:

  – по имени интерфейса, например `em1`

  – по MAC-адресу, например `01:23:45:67:89:ab`

  – по ключевому слову `link`, которое определяет первый интерфейс со статусом `up`

  – по ключевому слову `bootif` с целью наследования MAC-адреса из

переменной `BOOTIF`. Чтобы `pxelinux` смог определить переменную `BOOTIF`, включите `IPAPPEND 2` в файл `pxelinux.cfg`.

Пример:

```
network --bootproto=dhcp --device=em1
```

- `--ipv4-dns-search` / `--ipv6-dns-search` — задать домены поиска DNS вручную. Эти параметры используются вместе с параметрами `--device` и отражают соответствующие свойства NetworkManager, например:

  ```
  network --device ens3 --ipv4-dns-search domain1.example.com,domain2.example.com
  ```

- `--ipv4-ignore-auto-dns` / `--ipv6-ignore-auto-dns` — игнорировать настройки DNS от DHCP. Эти параметры используются вместе с параметрами `--device` и не требуют каких-либо аргументов.
- `--ip=` — адрес устройства.
- `--ipv6=` — IPv6-адрес устройства в формате `address[/prefix length]`, например `3ffe:ffff:0:1::1/128`. Если префикс не задан, по умолчанию будет использоваться `64`. Другие допустимые значения включают `auto` (автоматическая конфигурация) и `dhcp` (только для DHCPv6, без маршутизатора).
- `--gateway=` — IPv4-адрес шлюза.
- `--ipv6gateway=` — IPv6-адрес шлюза.
- `--nodefroute` — определяет, что выбранный интерфейс не может использоваться по умолчанию. Обычно устанавливается при активации дополнительных устройств с помощью `--activate=` — например для сетевой карты в другой подсети или цели iSCSI.
- `--nameserver=` — IP-адрес сервера имён. Несколько серверов разделяются запятыми.
- `--netmask=` — маска сети.
- `--hostname=` — имя устанавливаемой системы. Имя может быть либо полным доменным именем (FQDN) в формате `hostname.domainname`, либо коротким именем без домена. Во многих сетях имеется служба DHCP, которая автоматически предоставляет подключённым системам доменное имя. Чтобы разрешить службе DHCP назначать этому компьютеру имя домена, укажите только короткое имя.

  При использовании статического IP-адреса и конфигурации имени выбор использования короткого или полного доменного имени зависит от планируемого варианта использования системы. Обычно полное доменное имя настраивается во время подготовки, но для некоторых сторонних продуктов может потребоваться короткое имя. В любом случае, чтобы обеспечить доступность

обеих форм во всех ситуациях, добавьте запись для хоста в `/etc/hosts` в формате `IP FQDN short-alias`.

Имена хостов могут содержать только буквы и цифры, а также символы `-` и `.`. Имя хоста должно быть не более 64 символов. Имена хостов не могут начинаться или заканчиваться символами `-` и `.`. Для совместимости с DNS каждая часть полного доменного имени должна быть равна или меньше 63 символов, а общая длина полного доменного имени, включая точки, не должна превышать 255 символов.

Если вы хотите настроить только имя хоста целевой системы, используйте параметр `--hostname` и не используйте никаких других параметров.

Если при настройке имени хоста вы используете дополнительные параметры, команда `network` настроит устройство, используя указанные параметры. Если вы не указываете, какое устройство настраивать с помощью параметра `--device`, используется значение по умолчанию `--device link`. Кроме того, если вы не укажете протокол с помощью параметра `--bootproto`, устройство по умолчанию будет настроено на использование DHCP.

- `--ethtool=` — дополнительные параметры сетевого устройства для передачи ethtool.
- `--onboot=` — активация устройства при загрузке.
- `--dhcpclass=` — класс DHCP.
- `--mtu=` — MTU устройства.
- `--noipv4` — отключает IPv4.
- `--noipv6` — отключает IPv6.
- `--bondslaves=` — определяет подчинённые устройства, которые войдут в состав устройства `--device=`:

```
network --device=bond0 --bondslaves=em1,em2
```

  Эта команда создаст устройство `bond0` на основе интерфейсов `em1` и `em2`.

- `--bondopts=` — список дополнительных параметров для интерфейса агрегации, указанного в параметрах `--bondslaves=` и `--device=`. Параметры могут быть отделены запятой (`,`) или точкой с запятой (`;`). Если сам параметр содержит запятую, используйте точку с запятой для разделения. Пример:

```
network --bondopts=mode=active-backup,balance-rr;primary=eth1
```

Примечание

> `--bondopts=mode=` принимает полное имя режима, такое как `balance-rr` или `broadcast`, а не его числовое представление, такое как `0` или `3`.

- `--vlanid=` — идентификатор VLAN (тег `802.1q`) подчинённого устройства, которое будет создано для устройства `--device`. Так, например, `network --device=em1 --vlanid=171` создаст виртуальное устройство LAN `em1.171`.
- `--interfacename=` — позволяет изменить имя VLAN-устройства, присвоенное в соответствии с `--vlanid=`. ? Пример:

  ```
  network --device=em1 --vlanid=171 --interfacename=vlan171
  ```

  Эта команда создаст виртуальный интерфейс `vlan171` для устройства `em1` с ID `171`.

  Обозначение интерфейса может быть любым, например `my-vlan`, но в некоторых случаях следует придерживаться определенных рекомендаций:

  – Если имя содержит точку (`.`), то обозначение интерфейса должно иметь формат `NAME.ID`, где имя может быть произвольным, а ID содержит идентификатор VLAN. Примеры: `em1.171`, `my-vlan.171`.

  – Если имя начинается с `vlan`, то интерфейсу будет присвоено имя в формате `vlanID`, например `vlan171`.

- `--teamslaves=` — список подчинённых устройств, которые войдут в состав устройства `--device=`, через запятую. За именем устройства может следовать JSON-строка параметров в одинарных кавычках, где каждый параметр заключён в двойные кавычки, которым предшествует знак `\`. Пример:

  ```
  network --teamslaves="p3p1'{\"prio\": -10, \"sticky\": true}',p3p2'{\"prio\":␣
  ↪100}'"
  ```

  Также смотрите описание параметра `--teamconfig=`.

  > **Примечание**
  >
  > Агрегирование (Team) устарело в МСВСфера ОС 9. В качестве альтернативы рассмотрите возможность использования Агрегирования (Bond).

- `--teamconfig=` — JSON-строка конфигурации созданного устройства агрегации в кавычках, где каждый параметр тоже заключён в двойные кавычки с предшествующим им знаком `\`. Список подчинённых устройств, которые войдут в состав устройства `--device=`, заданный параметром `--teamslaves=`. Пример:

  ```
  network --device team0 --activate --bootproto static --ip=10.34.102.222 --
  ↪netmask=255.255.255.0 --gateway=10.34.102.254 --nameserver=10.34.39.2 --
  ```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
↪teamslaves="p3p1'{\"prio\": -10, \"sticky\": true}',p3p2'{\"prio\": 100}'" --
↪teamconfig="{\"runner\": {\"name\": \"activebackup\"}}"
```

> **Примечание**
>
> Агрегирование (Team) устарело в МСВСфера ОС 9. В качестве альтернативы рассмотрите возможность использования Агрегирования (Bond).

- `--bridgeslaves=` — создать сетевой мост с устройством, указанным в `--device=`; к мосту будут добавлены устройства, определённые в `--bridgeslaves=`. Пример:

  ```
  network --device=bridge0 --bridgeslaves=em1
  ```

- `--bridgeopts=` — дополнительные параметры мостового интерфейса, разделённые запятыми: `stp`, `priority`, `forward-delay`, `hello-time`, `max-age`, `ageing-time`.
- `--bindto=mac` — привязать файл конфигурации устройства в установленной системе к MAC-адресу устройства (HWADDR) вместо привязки по умолчанию к имени интерфейса (DEVICE). Обратите внимание, что этот параметр не зависит от параметра `--device=` и будет применяться, даже если команда `network` пределяет имя устройства, `link` или `bootif`.

Замечания

- Имена устройств `ethN`, такие как `eth0`, больше недоступны из-за изменений в схеме именования.
- Если установочный репозиторий в сети был указан в Kickstart-файле или как параметр загрузки, но в начале установки сеть недоступна, программа установки покажет окно «Настройка сети» для настройки сетевого подключения перед отображением окна «Обзор установки».

realm (дополнительная)

Настраивает подключение к домену IPA или Active Directory.

Синтаксис

```
realm join [OPTIONS] domain
```

Обязательные параметры

- `domain` — домен.

Дополнительные параметры

- `--computer-ou=OU=` — уникальное имя подразделения. Формат определяется программным обеспечением клиента. Составляющую root DSE можно опустить.
- `--no-password` — автоматическое подключение без ввода пароля.
- `--one-time-password=` — при входе необходимо ввести пароль один раз. Подходит только для некоторых областей.
- `--client-software=` — подключаться только к областям, где могут выполняться выбранные программы клиента. Допустимые значения включают `sssd` и `winbind`. Не все области поддержтвают все значения. По умолчанию программное обеспечение клиента выбирается автоматически.
- `--server-software=` — подключаться только к областям, где могут выполняться выбранные программы сервера. Допустимые значения включают `active-directory` и `freeipa`.
- `--membership-software=` — программа, выбираемая при подключении к области. Допустимые значения включают `samba` и `adcli`. Не все области поддерживают все значения. По умолчанию выбирается автоматически.

Команды управления хранилищем

Команды используются для настройки устройств, дисков, разделов, LVM и файловых систем.

Формат `sdX` (или `/dev/sdX`) не гарантирует согласованность имён устройств при перезагрузках, что может усложнить использование некоторых Kickstart-команд. Если команде требуется имя узла устройства, в качестве альтернативы вы можете использовать любой элемент из `/dev/disk`. Например, вместо использования следующего имени устройства:

```
part / --fstype=xfs --onpart=sda1
```

Вы можете использовать запись, подобную одной из следующих:

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

При таком подходе команда всегда нацелена на одно и то же устройство хранения. Это особенно полезно в больших средах хранения. Чтобы просмотреть имена доступных устройств в системе, вы можете использовать команду `ls -lR /dev/disk` во время интерактивной установки.

autopart (дополнительная)

Автоматически создает следующие разделы: корневой раздел `/` (не меньше 1 ГБ), `swap` и `/boot`. На больших дисках (больше 50 ГБ) также будет создан раздел `/home`.

Синтаксис

```
autopart OPTIONS
```

Дополнительные параметры

- `--type=` — позволяет выбрать готовую схему разделов. Возможные значения:
  - `lvm` — LVM-схема;
  - `plain` — стандартные разделы (не LVM);
  - `thinp` — динамическая схема LVM.
- `--fstype=` — выбрать тип файловой системы. Доступные значения: `ext2`, `ext3`, `ext4`, `xfs` и `vfat`. Файловая система по умолчанию — `xfs`.
- `--nohome` — отключает автоматическое создание раздела `/home`.
- `--nolvm` — запрещает выбор LVM при автоматическом создании разделов. Эквивалентно `--type=plain`.
- `--noboot` — не создавать раздел `/boot`.
- `--noswap` — не создавать раздел `swap`.
- `--encrypted` — включает шифрование разделов с помощью единого ключа Linux (LUKS). Аналогично установке флажка шифрования в окне создания разделов программы установки.

  При шифровании одного или нескольких разделов Anaconda пытается собрать 256 бит энтропии, чтобы обеспечить безопасное шифрование разделов. Сбор энтропии может занять некоторое время — процесс остановится максимум через 10 минут, независимо от того, было ли собрано достаточно энтропии. Процесс можно ускорить, взаимодействуя с системой установки (набирая текст на клавиатуре или перемещая мышь). Если вы устанавливаете систему на виртуальную машину, вы также можете подключить к гостевой машине устройство `virtio-rng` (виртуальный генератор случайных чисел).
- `--luks-version=LUKS_VERSION` — версия LUKS для шифрования файловой системы. Используется только если указан параметр `--encrypted`.
- `--passphrase=` — пароль доступа к зашифрованным устройствам.
- `--escrowcert=URL_of_X.509_certificate` — сохранить ключи шифрования данных всех зашифрованных томов в виде файлов в каталоге `/root`, зашифрованных с использованием сертификата X.509 по URL-адресу,

указанному в `URL_of_X.509_certificate`. Ключи хранятся в виде отдельного файла для каждого зашифрованного тома. Используется только если указан параметр `--encrypted`.

- `--backuppassphrase` — случайная парольная фраза для зашифрованного тома. Файлы паролей хранятся в `/root` и зашифрованы с помощью сертификата X.509. Этот параметр используется вместе с `--escrowcert`.
- `--cipher=` — тип шифрования, который будет выбран, если стандартного значения `aes-xts-plain64` оказалось недостаточно. Этот параметр используется вместе с `--encrypted` и сам по себе силы не имеет. Рекомендуется выбрать тип шифрования `aes-xts-plain64` или `aes-cbc-essiv:sha256`.
- `--pbkdf=PBKDF` — задаёт алгоритм PBKDF для ключей LUKS. См. также справку `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`
- `--pbkdf-memory=PBKDF_MEMORY` — задаёт затраты памяти при использовании PBKDF. Этот параметр используется вместе с `--encrypted` и сам по себе силы не имеет.
- `--pbkdf-time=PBKDF_TIME` — задаёт время в миллисекундах на обработку кодовой фразы PBKDF. Этот параметр используется вместе с `--encrypted` и сам по себе силы не имеет. Является взаимоисключающим с `--pbkdf-iterations`.
- `--pbkdf-iterations=PBKDF_ITERATIONS` — задаёт число итераций на обработку кодовой фразы PBKDF напрямую. Этот параметр используется вместе с `--encrypted` и сам по себе силы не имеет. Является взаимоисключающим с `--pbkdf-time`.

Замечания

- Команда `autopart` не может использоваться вместе с `part/partition` и командами `raid`, `logvol` и `volgroup` в одном Kickstart-файле.
- Команда `autopart` — дополнительная, но её необходимо включить в Kickstart-сценарий, если не используются команды `part` и `mount`.
- При установке на один FBA DASD типа CMS рекомендуется использовать `autopart --nohome`. Это гарантирует, что программа установки не создаст отдельный раздел `/home`. После этого установка проходит успешно.
- При утере парольной фразы LUKS, все зашифрованные разделы и их данные станут полностью недоступными. Восстановить утерянную парольную фразу невозможно. Однако вы можете сохранить кодовые фразы с помощью `--escrowcert` и создать резервные кодовые фразы с помощью `--backuppassphrase`.
- При использовании `autopart`, `autopart --type=lvm` и `autopart=thinp` убедитесь, что размеры секторов диска соответствуют требованиям.

bootloader (обязательная)

Определяет порядок установки загрузчика.

Синтаксис

```
bootloader [OPTIONS]
```

Дополнительные параметры

- `--append=` — содержит список параметров ядра через запятую, например:

  ```
  bootloader --location=mbr --append="hdd=ide-scsi ide=nodma"
  ```

  Параметры `rhgb` и `quiet` добавляются автоматически при установке пакета `plymouth`, даже если они явно не заданы, или строка `--append=` вовсе не определена. Чтобы отключить такое поведение, явно запретите установку `plymouth`:

  ```
  %packages
  -plymouth
  %end
  ```

  Этот параметр полезен для отключения механизмов устранения уязвимостей Meltdown и Spectre, обнаруженных в большинстве современных процессоров (CVE-2017-5754, CVE-2017-5753 и CVE-2017-5715). В некоторых случаях эти механизмы могут быть не нужны, а их включение привести к снижению производительности без улучшения безопасности. Чтобы отключить эти механизмы, добавьте соответствующие команды в Kickstart-файл, например для AMD64/Intel 64, `bootloader --append="nopti noibrs noibpb"`.

  Опасно

  Прежде чем отключать механизмы устранения уязвимостей, убедитесь, что система не будет подвергнута риску!

- `--boot-drive=` — диск, на который будет записан загрузчик, и с которого будет загружаться система. При использовании многопутевого устройства в качестве загрузочного диска, укажите устройство, используя его имя `disk/by-id/dm-uuid-mpath-WWID`.

- `--leavebootorder` — добавить систему в список установленных систем в UEFI. Установленная система не добавляется в порядок загрузки. Все существующие загрузочные записи, а также их порядок сохраняются. Применяется для Power и UEFI.

- `--driveorder=` — определяет порядок загрузки в BIOS, например:

```
bootloader --driveorder=sda,hda
```

- `--location=` — место расположения загрузочной записи. Допустимые значения:
  - `mbr` — (используется по умолчанию). Его поведение определяется наличием MBR или GPT.
    - На диске с GPT код загрузчика стадии 1.5 будет установлен в раздел BIOS.
    - На диске с MBR код загрузчика стадии 1.5 будет установлен в свободное пространство между MBR и первым разделом.
  - `partition` — установка загрузчика в первый сектор раздела, содержащего ядро.
  - `none` — не устанавливать загрузчик.

  В большинстве случаев этот параметр можно не определять.
- `--nombr` — не устанавливать загрузчик в MBR.
- `--password=` — пароль GRUB2. Рекомендуется установить пароль для ограничения доступа к оболочке GRUB2 во избежание несанкционированной передачи параметров ядру.

  Если пароль задан, будет предложено ввести имя пользователя. По умолчанию будет выбран пользователь `root`.
- `--iscrypted` — разрешает использование зашифрованного пароля, который изначально был создан помощью параметра `--password=`. В противном случае пароль хранится в открытой форме.

  Пароль можно зашифровать с помощью команды `grub2-mkpasswd-pbkdf2`. Скопируйте код из строки вывода, начинающейся с `grub.pbkdf2`, и сохраните его в Kickstart-файл.

  Пример записи с зашифрованным паролем:

```
bootloader --iscrypted --password=grub.pbkdf2.sha512.10000.
↪5520C6C9832F3AC3D149AC0B24BE69E2D4
FB0DBEEDBD29CA1D30A044DE2645C4C7A291E585D4DC43F8A4D82479F8B95CA4BA4381F8550510B75E8E0BB2938990.
↪
C688B6F0EF935701FF9BD1A8EC7FE5BD2333799C98F28420C5CC8F1A2A233DE22C83705BB614EA17F3FDFDF4AC2161CEA3384E56EB
```

- `--timeout=` — время ожидания в секундах, по истечении которого начнётся загрузка в соответствии с параметром `--default=`.
- `--default=` — определяет загрузочный образ, который будет использоваться по умолчанию.
- `--extlinux` — выбирает extlinux вместо GRUB2. Используется только в системах с поддержкой extlinux.

- `--disabled` — расширенная версия `--location=none`. В то время как `--location=none` просто отключает установку загрузчика, `--disabled` отключает установку загрузчика, а также отключает установку пакета, содержащего загрузчик, тем самым экономя место.

Замечания

- Рекомендуется установить пароль загрузчика в каждой системе. Незащищённый загрузчик может позволить потенциальному злоумышленнику изменить параметры загрузки системы и получить несанкционированный доступ к системе.
- В некоторых случаях для установки загрузчика в системах AMD64, Intel 64 и 64-битных ARM требуется специальный раздел. Тип и размер этого раздела зависит от того, использует ли диск, на который устанавливается загрузчик, MBR или GPT.
- Формат `sdX` (или `/dev/sdX`) не гарантирует согласованность имён устройств при перезагрузках, что может усложнить использование некоторых команд. Если команде требуется имя узла устройства, в качестве альтернативы можно использовать любой элемент из `/dev/disk`. Например, вместо использования следующего имени устройства:

  ```
  part / --fstype=xfs --onpart=sda1
  ```

  Вы можете использовать запись, подобную одной из следующих:

  ```
  part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-
  ↪part1

  part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
  ```

  При таком подходе команда всегда нацелена на одно и то же устройство хранения. Это особенно полезно в больших средах хранения. Чтобы просмотреть имена доступных устройств в системе, можно использовать команду `ls -lR /dev/disk` во время интерактивной установки.

clearpart (дополнительная)

Удаляет разделы с диска, прежде чем создавать новые. По умолчанию разделы не удаляются.

Синтаксис

```
clearpart OPTIONS
```

Дополнительные параметры

- `--all` — удаляет все разделы, к которым может получить доступ программа установки, включая любые подключённые сетевые хранилища. Используйте с

осторожностью! Запретить очищать необходимое хранилище можно используя параметр `--drives=` и указав только те диски, которые вы хотите очистить. Вы можете подключить сетевое хранилище позже (например, в разделе `%post`). Также вы можете внести в чёрный список модули ядра, используемые для доступа к сетевому хранилищу.

- `--drives=` — удаляет разделы на заданных дисках. Например, следующая команда очистит все разделы на первых двух дисках основного IDE-контроллера:

```
clearpart --drives=hda,hdb --all
```

Чтобы очистить многопутевое устройство, используйте формат `disk/by-id/scsi-WWID`. Так, очистка диска с идентификатором `58095BEC5510947BE8C0360F604351918` будет выглядеть так:

```
clearpart --drives=disk/by-id/scsi-58095BEC5510947BE8C0360F604351918
```

Чтобы очистить многопутевое устройство, не использующее LVM, следуйте формату `disk/by-id/dm-uuid-mpath-WWID`. Так, команда очистки диска с идентификатором `2416CD96995134CA5D787F00A5AA11017` будет выглядеть так:

```
clearpart --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

Не используйте имена устройств в виде `mpatha`, так как они не привязаны к конкретным дискам, и могут измениться в процессе установки. Это может привести к тому, что `clearpart` выберет не тот диск для освобождения.

- `--initlabel` — метка диска, которая будет выбираться по умолчанию для заданной архитектуры (например, msdos для x86). Используется вместе с `--all`. Поскольку `--initlabel` видит все диски, важно убедиться, что подключены только те диски, которые необходимо отформатировать. На дисках, очищенных с помощью `clearpart`, будет создана метка, даже если `--initlabel` не используется.

```
clearpart --initlabel --drives=names_of_disks
```

Пример:

```
clearpart --initlabel --drives=dasda,dasdb,dasdc
```

- `--list=` — список разделов для удаления. Допускается выбор разделов на разных дисках. Этот параметр переопределяет `--all` и `--linux`.

```
clearpart --list=sda2,sda3,sdb1
```

- `--disklabel=LABEL` — метка диска по умолчанию. Будут приняты только метки дисков, поддерживаемые данной платформой. Например, на 64-битных архитектурах Intel и AMD принимаются метки `msdos` и `gpt`, но не принимаются `dasd`.

- `--linux` — удаляет разделы Linux.
- `--none` (по умолчанию) — не удалять разделы.
- `--cdl` — переформатировать все LDL DASD в CDL.

Замечания

- Формат `sdX` (или `/dev/sdX`) не гарантирует согласованность имён устройств при перезагрузках, что может усложнить использование некоторых команд. Если команде требуется имя узла устройства, в качестве альтернативы можно использовать любой элемент из `/dev/disk`. Например, вместо использования следующего имени устройства:

  ```
  part / --fstype=xfs --onpart=sda1
  ```

  Вы можете использовать запись, подобную одной из следующих:

  ```
  part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-
  ↪part1

  part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
  ```

  При таком подходе команда всегда нацелена на одно и то же устройство хранения. Это особенно полезно в больших средах. Чтобы просмотреть имена доступных устройств, используйте команду `ls -lR /dev/disk` во время интерактивной установки.

- `clearpart` и `part --onpart` не могут использоваться вместе по отношению к одному и тому же логическому разделу.

fcoe (дополнительная)

Определяет устройства FCoE, которые будут активированы автоматически помимо обнаруженных службами EDD (Enhanced Disk Drive Services).

Синтаксис

```
fcoe --nic=name [OPTIONS]
```

Дополнительные параметры

- `--nic=` (обязательный) — имя устройства.
- `--dcb=` — параметры DCB (Data Center Bridging).
- `--autovlan` — автоматическое определение VLAN. По умолчанию включено.

ignoredisk (дополнительная)

Позволяет пропустить некоторые диски. Это, в частности, поможет исключить диски при автоматическом разбиении. К примеру, попытка установки без параметра `ignoredisk` в SAN-кластере завершится неудачей вследствие того, что установщик будет находить пассивные пути к SAN без таблицы разделов.

Синтаксис

```
ignoredisk --drives=drive1,drive2,... | --only-use=drive
```

Дополнительные параметры

- `--drives=driveN,…` — `driveN` может принимать значения `sda`, `sdb`,..., `hda` и т.п.
- `--only-use=driveN,…`— список дисков, которые будет использовать программа установки. Все остальные диски игнорируются. Например, чтобы использовать диск `sda` во время установки и игнорировать все остальные диски:

  ```
  ignoredisk --only-use=sda
  ```

  Чтобы включить многопутевое устройство, не использующее LVM:

  ```
  ignoredisk --only-use=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
  ```

  Чтобы включить многопутевое устройство, использующее LVM:

  ```
  ignoredisk --only-use==/dev/disk/by-id/dm-uuid-mpath-
  ```

  ```
  bootloader --location=mbr
  ```

Можно использовать только один из `--drives` или `--only-use`.

Замечания

- Чтобы пропустить многопутевое устройство, не использующее LVM, выберите формат `disk/by-id/dm-uuid-mpath-WWID`. Так, чтобы пропустить диск с идентификатором `2416CD96995134CA5D787F00A5AA11017` выполните:

  ```
  ignoredisk --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
  ```

- Не используйте имена устройств в виде `mpatha`, так как они не привязаны к конкретным дискам, и могут измениться в процессе установки. Это может привести к тому, что `clearpart` выберет не тот диск для освобождения.
- Формат `sdX` (или `/dev/sdX`) не гарантирует согласованность имен устройств при перезагрузках, что может усложнить использование некоторых команд. Если команде требуется имя узла устройства, в качестве альтернативы можно использовать любой элемент из `/dev/disk`. Например, вместо использования следующего имени устройства:

```
part / --fstype=xfs --onpart=sda1
```

Вы можете использовать запись, подобную одной из следующих:

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1

part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

При таком подходе команда всегда нацелена на одно и то же устройство хранения. Это особенно полезно в больших средах хранения. Чтобы просмотреть имена доступных устройств, вы можете использовать команду `ls -lR /dev/disk` во время интерактивной установки.

iscsi (дополнительная)

Определяет дополнительный iSCSI-накопитель, который будет подключён во время установки.

Синтаксис

```
iscsi --ipaddr=address [OPTIONS]
```

Обязательные параметры

- `--ipaddr=` (обязательный) — IP-адрес ресурса.

Дополнительные параметры

- `--port=` (обязательный) — номер порта. Если не указан, по умолчанию используется `--port=3260`.
- `--target=` — IQN-имя ресурса (iSCSI Qualified Name).
- `--iface=` — привязка соединения к конкретному сетевому интерфейсу. Определив таким образом интерфейс, убедитесь, что остальные экземпляры `iscsi` тоже содержат его обозначение.
- `--user=` — имя пользователя для доступа к ресурсу.
- `--password=` — пароль доступа.
- `--reverse-user=` — имя пользователя для авторизации на инициаторе из целевого объекта, использующего обратное шифрование CHAP.
- `--reverse-password=` — пароль пользователя инициатора.

Замечания

- Если задан параметр `iscsi`, надо присвоить имя узла iSCSI при помощи `iscsiname`. В Kickstart-файле `iscsi` должен следовать за командой `iscsiname`.

- Так как параметр `iscsi` требует дополнительной конфигурации, более предпочтительным вариантом является настройка iSCSI-накопителей в BIOS или встроенном ПО (или iBFT на Intel). Anaconda автоматически определяет и использует диски, настроенные в BIOS или во встроенном ПО, и не требует специального изменения Kickstart-файла.
- Если все же необходимо использовать `iscsi`, проверьте работоспособность сетевого подключения и убедитесь, что в Kickstart-файле эта команда предшествует остальным командам iSCSI-дисков, таким как `clearpart` и `ignoredisk`.

iscsiname (дополнительная)

Позволяет определить имя узла iSCSI, заданного с помощью `iscsi`.

Синтаксис

```
iscsiname iqname
```

Дополнительные параметры

- `iqname` — имя узла iSCSI.

Замечания

- В Kickstart-файле команда `iscsiname` должна предшествовать команде `iscsi`.

logvol (дополнительная)

Помогает создать логический том LVM.

Синтаксис

```
logvol mntpoint --vgname=name --name=name [OPTIONS]
```

Обязательные параметры

- `mntpoint` — точка монтирования. Формат точки монтирования:
  - `/path` — путь. Например, `/` или `/home`.
  - `swap` — Раздел подкачки. Чтобы автоматически выбрать размер раздела подкачки, добавьте аргумент `--recommended`:

    ```
    swap --recommended
    ```

    Будет выбран подходящий, но не обязательно оптимальный размер. Чтобы автоматически выбрать размер с выделением дополнительного пространства для перехода в спящий режим, добавьте аргумент `--hibernation`:

```
swap --hibernation
```

Общая формула расчёта: размер пространства подкачки = (`--recommended`) + (размер ОЗУ)

- `--vgname=name` — имя группы томов.
- `--name=name` — имя логического тома.

Дополнительные параметры

- `--noformat` — использует существующий логический том без форматирования.
- `--useexisting` — использует существующий логический том и повторно его форматирует
- `--fstype=` — тип файловой системы: `xfs`, `ext2`, `ext3`, `ext4`, `swap` или `vfat`.
- `--fsoptions=` — строка параметров монтирования файловой системы, заключённая в кавычки. В готовой системе эта строка будет записана в `/etc/fstab`. В разделе EFI (`/boot/efi`) Anaconda жёстко кодирует значение и игнорирует указанные пользователем значения в `--fsoptions`.
- `--mkfsoptions=` — указывает дополнительные параметры, которые необходимо передать программе, создающей файловую систему в этом разделе. Список аргументов не обрабатывается, поэтому их необходимо предоставить в формате, который можно передать непосредственно в программу `mkfs`. Это означает, что несколько параметров должны быть разделены запятыми или заключены в двойные кавычки, в зависимости от файловой системы. Например,

```
part /opt/foo1 --size=512 --fstype=ext4 --mkfsoptions="-O ^has_journal,^flex_bg,^metadata_
↪csum"

part /opt/foo2 --size=512 --fstype=xfs --mkfsoptions="-m bigtime=0,finobt=0"
```

  Подробности см. в руководствах создаваемых файловых систем. Например, `mkfs.ext4` или `mkfs.xfs`.

- `--fsprofile=` — указывает тип использования, который будет передан программе, создающей файловую систему в этом разделе. Тип использования определяет множество параметров настройки, которые будут использоваться при создании файловой системы. Чтобы этот параметр работал, файловая система должна поддерживать концепцию типов использования и должен существовать файл конфигурации, в котором перечислены допустимые типы. Для `ext2`, `ext3` и `ext4` этот файл — `/etc/mke2fs.conf`.
- `--label=` — метка логического тома.
- `--grow` — увеличить размер логического тома до заданной величины или заполнить все свободное место. Этот параметр следует использовать только в том случае, если вы заранее выделили минимальное пространство для хранения в образе диска и хотите, чтобы том увеличивался и занимал

доступное пространство. В физической среде это одноразовое действие. Однако в виртуальной среде размер тома увеличивается по мере того, как виртуальная машина записывает какие-либо данные на виртуальный диск.

- `--size=` — минимальный размер тома в мегабайтах. Нельзя использовать вместе с `--percent=`.
- `--percent=` — размер увеличения логического тома в процентной доле от доступного места в группе томов. Нельзя использовать вместе с `--size=`.

  При создании нового логического тома необходимо указать его размер статически с помощью параметра `--size=` или в процентах от оставшегося свободного пространства с помощью параметра `--percent=`. Вы не можете использовать оба параметра применительно к одному и тому же логическом томе.

- `--maxsize=` — максимально допустимый размер тома в мегабайтах. Укажите целое значение без единиц, например 500.
- `--recommended` — автоматический выбор размера тома.
- `--resize` — изменение размера тома. Также надо будет определить параметры `--useexisting` и `--size`.
- `--encrypted` — включает шифрование логического тома с использованием парольной фразы LUKS из параметра `--passphrase=`. Если парольная фраза не определена, Anaconda использует стандартный пароль из строки `autopart --passphrase`, а если и он не задан — будет предложено ввести пароль во время установки.

  При шифровании одного или нескольких разделов Anaconda пытается собрать 256 бит энтропии, чтобы обеспечить безопасное шифрование разделов. Сбор энтропии может занять некоторое время — процесс остановится максимум через 10 минут, независимо от того, было ли собрано достаточно энтропии. Процесс можно ускорить, взаимодействуя с системой установки (набирая текст на клавиатуре или перемещая мышь). Если вы устанавливаете на виртуальную машину, вы также можете подключить к гостю устройство `virtio-rng` (виртуальный генератор случайных чисел).

- `--passphrase=` — парольная фраза для доступа к логическому тому. Используется вместе с `--encrypted`.
- `--cipher=` — тип шифрования, который будет выбран, если стандартного значения `aes-xts-plain64`, используемого в Anaconda, оказалось недостаточно. Этот параметр используется вместе с `--encrypted` и сам по себе силы не имеет. Рекомендуется выбрать `aes-xts-plain64` или `aes-cbc-essiv:sha256`.
- `--escrowcert=URL_of_X.509_certificate` — сертификат X.509, с помощью которого зашифрованы ключи томов в `/root`. Для каждого тома создается

отдельный файл. Этот параметр имеет смысл только при наличии аргумента `--encrypted`.

- `--luks-version=LUKS_VERSION` — версия LUKS для шифрования файловой системы. Этот параметр имеет смысл только если указан `--encrypted`.
- `--backuppassphrase` — случайная парольная фраза для зашифрованного тома. Файлы паролей хранятся в `/root` и зашифрованы с помощью сертификата X.509. Этот параметр используется вместе с `--escrowcert`.
- `--pbkdf=PBKDF` — задаёт алгоритм PBKDF для ключей LUKS. См. также справку `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`.
- `--pbkdf-memory=PBKDF_MEMORY` — задаёт затраты памяти для PBKDF. См. также справку `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`.
- `--pbkdf-time=PBKDF_TIME` — задаёт количество миллисекунд, затрачиваемое на обработку кодовой фразы PBKDF. См. также `--iter-time` в руководстве `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`, и является взаимоисключающей с `--pbkdf-iterations`.
- `--pbkdf-iterations=PBKDF_ITERATIONS` — задаёт количество итераций напрямую и не использует тест PBKDF. См. также `--pbkdf-force-iterations` в руководстве `cryptsetup(8)`. Этот параметр имеет смысл только в том случае, если указан `--encrypted`, и является взаимоисключающим с `--pbkdf-time`.
- `--thinpool` — создание логического тома в динамическом пуле. В качестве точки монтирования укажите `none`.
- `--metadatasize=size` — размер области метаданных (в МБ).
- `--chunksize=size` — размер секции для нового устройства динамического пула (в КиБ).
- `--thin` — создание динамического логического тома. Используется вместе с `--poolname`.
- `--poolname=name` — пул, в котором будет создан динамический том. Используется вместе с `--thin`.
- `--profile=name` — имя профиля конфигурации, который будет использоваться с динамическими логическими томами. Если используется, имя также будет включено в метаданные данного логического тома. По умолчанию доступны профили `default` и `thin-performance`, которые определены в `/etc/lvm/profile/` . Дополнительную информацию смотрите в руководстве `lvm(8)`.
- `--cachepvs=` — список физических томов, разделенных запятыми, которые следует использовать в качестве кэша для этого тома.

- `--cachemode=` — режим, который следует использовать для кэширования этого логического тома: `writeback` или `writethrough`.

  Дополнительную информацию о кэшированных логических томах и их режимах см. в руководстве `lvmcache(7)`.

- `--cachesize=` — размер кэша для логического тома в МБ. Для этого параметра требуется `--cachepvs=`.

Замечания

- Имена логических томов и их групп не должны содержать дефис – в противном случае в списке `/dev/mapper/` они будут включать по два дефиса. То есть если группа `volgrp-01` содержит том `logvol-01`, его имя будет иметь вид `/dev/mapper/volgrp--01-logvol--01`. Это ограничение накладывается только на новые группы и тома. Имена существующих томов (см. `--noformat`) не изменятся.

- При утере парольной фразы LUKS, все зашифрованные разделы и их данные станут полностью недоступными. Восстановить утерянную парольную фразу невозможно. Однако вы можете сохранить кодовые фразы шифрования с помощью `--escrowcert` и создать резервные кодовые фразы с помощью `--backuppassphrase`.

Примеры

- Создать раздел, группу логических томов, а затем логический том:

```
part pv.01 --size 3000
volgroup myvg pv.01
logvol / --vgname=myvg --size=2000 --name=rootvol
```

- Создать раздел, группу логических томов, а затем логический том, который займёт 90 % оставшегося места в группе томов:

```
part pv.01 --size 1 --grow
volgroup myvg pv.01
logvol / --vgname=myvg --name=rootvol --percent=90
```

mount (дополнительная)

Задать точку монтирования существующему блочному устройству и при необходимости переформатировать его в заданный формат.

Синтаксис

```
mount [OPTIONS] device mountpoint
```

Обязательные параметры

- `device` — блочное устройство.

- `mountpoint` — действительная точка монтирования устройства, такая как `/` или `/usr`. Если устройство невозможно смонтировать (например, в swap), точки монтирования не будет.

Дополнительные параметры

- `--reformat=` — задаёт формат для переформатирования (например, `ext4`).
- `--mkfsoptions=` — дополнительные параметры для передаче команде, которая создает новую файловую систему, указанную в `--reformat=`. Представленные здесь параметры не обрабатываются, поэтому их необходимо указать в формате, который можно передать непосредственно в программу `mkfs`. Список параметров должен быть разделён запятыми или заключён в двойные кавычки, в зависимости от файловой системы.
- `--mountoptions=` — задаёт строку произвольной формы, содержащую параметры, которые будут использоваться при монтировании файловой системы. Строка будет скопирована в файл `/etc/fstab` установленной системы и должна быть заключена в двойные кавычки.

Замечания

- В отличие от большинства других команд настройки хранилища, команда `mount` не требует от вас описания всей конфигурации хранилища в Kickstart-файле. Вам нужно только убедиться, что описанное блочное устройство существует в системе. Однако если вы хотите создать стек хранилища со всеми подключёнными устройствами, для этого необходимо использовать другие команды, например `part`.
- Команда `mount` не может использоваться вместе с другими командами, связанными с хранилищем, такими как `part`, `logvol` или `autopart`, в одном Kickstart-файле.

nvdimm (дополнительная)

Выполнить действие на NVDIMM-устройстве.

Синтаксис

```
nvdimm action [OPTIONS]
```

Действия

- `reconfigure` — перенастроить NVDIMM-устройство в заданный режим. Кроме того, указанное устройство неявно помечается как используемое, поэтому следующая команда `nvdimm use` для того же устройства не нужна. Формат:

  ```
  nvdimm reconfigure [--namespace=NAMESPACE] [--mode=MODE] [--sectorsize=SECTORSIZE]
  ```

  - `--namespace=` — устройство в пространстве имён. Пример:

    ```
    nvdimm reconfigure --namespace=namespace0.0 --mode=sector --sectorsize=512
    ```

  - `--mode=` — режим. Сейчас доступен только `sector`.
  - `--sectorsize=` размер сектора для режима `sector`. Пример:

    ```
    nvdimm reconfigure --namespace=namespace0.0 --mode=sector --sectorsize=512
    ```

    Поддерживаемые размеры секторов — 512 и 4096 байт.

- `use` — указать NVDIMM-устройство в качестве целевого устройства для установки. Устройство уже должно быть настроено в режим `sector` с помощью команды `nvdimm reconfigure`. Формат:

  ```
  nvdimm use [--namespace=NAMESPACE|--blockdevs=DEVICES]
  ```

  - `--namespace=` — устройство в пространстве имён. Пример:

    ```
    nvdimm use --namespace=namespace0.0
    ```

- `--blockdevs=` — разделённый запятыми список блочных устройств, соответствующих используемым NVDIMM-устройствам. Поддреживается подстановочный знак звёздочка `*`. Например:

  ```
  nvdimm use --blockdevs=pmem0s,pmem1s
  nvdimm use --blockdevs=pmem*
  ```

Замечания

- По умолчанию все NVDIMM-устройства игнорируются программой установки. Разрешите установку на эти устройства командой `nvdimm`.

part или partition (обязательная)

Отвечает за создание раздела.

Синтаксис

```
part|partition mntpoint [OPTIONS]
```

Параметры

- `mntpoint` — каталог, в который будет смонтирован раздел. Формат точки монтирования:
  - `/path`. Примеры: `/`, `/usr`, `/home`
  - `swap`. Раздел подкачки. Чтобы автоматически выбрать размер раздела подкачки, добавьте аргумент `--recommended`:

```
swap --recommended
```

Будет выбран подходящий, но не обязательно оптимальный размер. Чтобы автоматически выбрать размер с выделением дополнительного пространства для перехода в спящий режим, добавьте аргумент `--hibernation`:

```
swap --hibernation
```

Общая формула расчёта: размер пространства подкачки = (`--recommended`) + (размер ОЗУ).

- `raid.id` — программный RAID-массив (см. `raid`).
- `pv.id` — раздел LVM (см. `logvol`).
- `biosboot` — загрузочный раздел BIOS. На платформах AMD64 и Intel 64 с BIOS, использующих таблицу разделов GPT, необходимо создать загрузочный раздел размером 1 МБ. Именно в этот раздел будет установлен загрузчик. В системах с UEFI этот раздел не требуется. Также смотрите описание команды `bootloader`.
- `/boot/efi` — системный раздел EFI. На платформах AMD64 и Intel 64 с UEFI, а также 64-битных ARM необходимо создать раздел с минимальным размером 50 МБ (рекомендуется 200 МБ). В системах с BIOS этот раздел не требуется. Также смотрите описание команды `bootloader`.

- `--size=` — минимальный размер раздела в мегабайтах. Введите целое значение без указания единиц, например `500`.

Важно

Если значение `--size` недостаточно велико, установка будет прекращена. Установите значение `--size` как минимальный объём требуемого места.

- `--grow` — увеличить размер раздела до заданной величины или заполнить все свободное место.

Примечание

Если для раздела подкачки определен параметр `--grow=` без `--maxsize=`, Anaconda ограничит его максимальный размер, руководствуясь следующими правилами: в системах с физической памятью <2 ГБ максимальный размер будет равен двойному объему памяти, а в системах с памятью >2 ГБ максимальный размер будет равен объему памяти плюс 2 ГБ.

- `--maxsize=` — максимальный размер раздела в мегабайтах. Укажите целое значение без единиц, например `500`.
- `--noformat` — использует существующий логический том без форматирования. Используется вместе с `--onpart`.
- `--onpart=` или `--usepart=` — устройство, где будет создан раздел. Использует существующее пустое устройство и форматирует его. Пример:

```
partition /home --onpart=hda1
```

  Эта команда создаст раздел `/home` на `/dev/hda1`.

  Пример создания раздела в пределах логического тома:

```
partition pv.1 --onpart=hda2
```

  Устройство уже должно существовать, так как `--onpart` не может создать его.

  Также можно указать весь диск, а не раздел, и в этом случае Anaconda отформатирует и будет использовать диск без создания таблицы разделов. Однако обратите внимание, что установка GRUB2 не поддерживается на устройстве, отформатированном таким образом, и его необходимо разместить на диске с таблицей разделов.

```
partition pv.1 --onpart=hdb
```

- `--ondisk=` или `--ondrive=` — позволяет выбрать диск, на котором будет создан раздел. Например, `--ondisk=sdb` размещает раздел на втором SCSI-диске.

  Чтобы выбрать многопутевое устройство, не использующее LVM, следуйте формату `disk/by-id/dm-uuid-mpath-WWID`. Так, для устройства с идентификатором `2416CD96995134CA5D787F00A5AA11017` выполните:

```
part / --fstype=xfs --grow --asprimary --size=8192 --ondisk=disk/by-id/dm-uuid-mpath-
↪2416CD96995134CA5D787F00A5AA11017
```

> **Предупреждение**
>
> Не используйте имена устройств в виде `mpatha`, так как они не привязаны к конкретным дискам, и могут измениться в процессе установки. Это может привести к тому, что `part` выберет не тот диск.

- `--asprimary` — отмечает раздел как первичный. Если это невозможно — что вероятно при наличии большого числа первичных разделов — процесс разбиения завершится неудачей. Этот параметр имеет смысл только для дисков с MBR, для дисков с GPT этот параметр не имеет смысла.
- `--fsprofile=` — определяет профиль для передачи программе создания

файловой системы. Профиль содержит набор параметров файловой системы. Этот аргумент требует наличия файла конфигурации. Так, в файловых системах `ext2`, `ext3`, `ext4` файл расположен в `/etc/mke2fs.conf`.

- `--mkfsoptions=` — дополнительные параметры, которые необходимо передать программе, создающей файловую систему. Похож на `--fsprofile`, но работает для всех файловых систем, а не только для тех, которые поддерживают концепцию профиля. Список аргументов не обрабатывается, поэтому их необходимо предоставить в формате, который можно передать непосредственно в программу `mkfs`. Это означает, что несколько параметров должны быть разделены запятыми или заключены в двойные кавычки, в зависимости от файловой системы. Например:

```
part /opt/foo1 --size=512 --fstype=ext4 --mkfsoptions="-O ^has_journal,^flex_bg,^metadata_
↪csum"

part /opt/foo2 --size=512 --fstype=xfs --mkfsoptions="-m bigtime=0,finobt=0"
```

  Подробнее см. руководства создаваемых файловых систем. Например, `mkfs.ext4` или `mkfs.xfs`.

- `--fstype=` — тип файловой системы: `xfs`, `ext2`, `ext3`, `ext4`, `swap`, `vfat`, `efi`, `biosboot`.

- `--fsoptions` — строка параметров монтирования файловой системы, заключённая в кавычки. В готовой системе эта строка будет записана в `/etc/fstab`.

> **Примечание**
>
> В системном разделе EFI (`/boot/efi`) Anaconda жёстко кодирует значение и игнорирует указанные пользователем значения `--fsoptions`.

- `--label=` — метка раздела.

- `--recommended` — автоматический выбор размера.

> **Важно**
>
> Параметр можно использовать только для разделов, в которых используется такая файловая система, как `/boot` и `swap`. Его нельзя использовать для создания физических томов LVM или RAID.

- `--onbiosdisk` — создание раздела на заданном диске (в соответствии с обозначенным в BIOS).

- `--encrypted` — включает шифрование раздела LUKS с использованием парольной фразы из `--passphrase=`. Если парольная фраза не задана, Anaconda

использует стандартный пароль из строки `autopart --passphrase`, а если и он не задан — будет предложено его ввести во время установки.

При шифровании одного или нескольких разделов Anaconda пытается собрать 256 бит энтропии, чтобы обеспечить безопасное шифрование разделов. Сбор энтропии может занять некоторое время — процесс остановится максимум через 10 минут, независимо от того, было ли собрано достаточно энтропии. Процесс можно ускорить, взаимодействуя с системой установки (набирая текст на клавиатуре или перемещая мышь). Если вы устанавливаете систему на виртуальную машину, вы также можете подключить к гостевой машине устройство `virtio-rng` (виртуальный генератор случайных чисел).

- `--luks-version=LUKS_VERSION` — версия LUKS для шифрования файловой системы. Используется только если указан параметр `--encrypted`.
- `--passphrase=` — парольная фраза для доступа к разделу. Используется вместе с `--encrypted`.
- `--cipher=` — тип шифрования, который будет выбран, если стандартного значения `aes-xts-plain64` оказалось недостаточно. Этот параметр используется вместе с `--encrypted` и сам по себе силы не имеет. Рекомендуется выбирать тип `aes-xts-plain64` или `aes-cbc-essiv:sha256`.
- `--escrowcert=URL_of_X.509_certificate` — сертификат X.509, с помощью которого зашифрованы ключи разделов в `/root`. Для каждого тома создается отдельный файл. Этот параметр имеет смысл только при наличии аргумента `--encrypted`.
- `--backuppassphrase` — случайная парольная фраза для зашифрованного раздела. Файлы паролей хранятся в `/root` и зашифрованы с помощью сертификата X.509. Этот параметр используется вместе с `--escrowcert`.
- `--pbkdf=PBKDF` — задаёт алгоритм PBKDF для ключей LUKS. См. также справку `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`.
- `--pbkdf-memory=PBKDF_MEMORY` — задаёт затраты памяти для PBKDF. См. также справку `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`.
- `--pbkdf-time=PBKDF_TIME` — задаёт количество миллисекунд, затрачиваемое на обработку кодовой фразы PBKDF. См. также `--iter-time` в руководстве `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`, и является взаимоисключающей с `--pbkdf-iterations`.
- `--pbkdf-iterations=PBKDF_ITERATIONS` — задаёт количество итераций напрямую и не использует тест PBKDF. См. также `--pbkdf-force-iterations` в руководстве `cryptsetup(8)`. Этот параметр имеет смысл только в том случае, если указан `--encrypted`, и является взаимоисключающим с `--pbkdf-time`.

- `--resize=` — изменение размера раздела (в МБ). Используется в комбинации с `--size=` и `--onpart=`.

Замечания

- Команда `part` не является обязательной, но в Kickstart-сценарий должна быть включена одна из команд `part`, `autopart` или `mount`.
- Если по какой-либо причине разбиение не было успешно, сообщения диагностики будут выведены на виртуальную консоль 3.
- Все создаваемые разделы форматируются в процессе установки, если не указан `--noformat` и `--onpart`.
- Формат `sdX` (или `/dev/sdX`) не гарантирует согласованность имён устройств при перезагрузке, что может усложнить использование некоторых команд. Если команде требуется имя узла устройства, в качестве альтернативы можно использовать любой элемент из `/dev/disk`. Например, вместо использования следующего имени устройства:

  ```
  part / --fstype=xfs --onpart=sda1
  ```

  Вы можете использовать запись, подобную одной из следующих:

  ```
  part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-
  ↪part1

  part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
  ```

  При таком подходе команда всегда нацелена на одно и то же устройство хранения. Это особенно полезно в больших средах хранения. Чтобы просмотреть имена доступных устройств в системе, можно использовать команду `ls -lR /dev/disk` во время интерактивной установки.
- При утере парольной фразы LUKS, все зашифрованные разделы и их данные станут полностью недоступными. Восстановить утерянную парольную фразу невозможно. Однако вы можете сохранить их с помощью `--escrowcert` и создать резервные кодовые фразы шифрования с помощью `--backuppassphrase`.

raid (дополнительная)

Создание программного RAID-массива.

Синтаксис

```
raid mntpoint --level=level --device=device-name partitions*
```

Параметры

- `mntpoint` — каталог, в который будет смонтирована файловая система RAID. Если это — `/`, то уровень RAID должен быть равен `1`. Если определён

загрузочный раздел, `/boot` будет иметь уровень `1`, а корневой раздел `/` может быть любого типа. Параметр `partitions*` определяет идентификаторы разделов, которые будут добавлены в RAID-массив. Можно указать несколько разделов.

Если на платформах IBM Power Systems RAID-устройство не было отформатировано во время установки, убедитесь, что метаданные RAID имеют версию `0.90` или `1.0`. Это является обязательным условием для размещения разделов `/boot` и PReP в пределах RAID-массива. Версия метаданных mdadm, `1.1` and `1.2`, для разделов `/boot` и PReP не поддерживается. Загрузочные разделы PReP не требуются в системах PowerNV.

- `--level=` — уровень RAID (0, 1, 4, 5, 6, or 10).
- `--device=` — имя устройства, например, `--device=root`.

  Не используйте имена `mdraid` в форме `md0` — постоянство этих имен не гарантируется. Вместо этого используйте осмысленные имена, такие как `root` или `swap`. Использование осмысленных имён создает символическую ссылку из `/dev/md/name` на любой узел `/dev/mdX`, назначенный массиву.

  К старым версиям RAID-массивов (с метаданными версии 0.90), которым нельзя присвоить имя, можно обращаться по метке файловой системы или UUID. Пример: `--device=LABEL=root` и `--device=UUID=93348e56-4631-d0f0-6f5b-45c47f570b88`.

  Вы можете использовать UUID файловой системы на RAID-устройстве или UUID самого RAID-устройства. UUID RAID-устройства должен быть в формате `8-4-4-4-12`. UUID, сообщаемый `mdadm`, имеет формат `8:8:8:8`, который необходимо изменить. Например, `93348e56:4631d0f0:6f5b45c4:7f570b88` следует изменить на `93348e56-4631-d0f0-6f5b-45c47f570b88`.

- `--chunksize=` — размер фрагмента RAID-хранилища в КБ. В определенных ситуациях использование размера фрагмента, отличного от размера по умолчанию (512 КБ), может повысить производительность RAID.
- `--spares=` — число резервных дисков. Резервные диски используются для восстановления массива в случае сбоя одного из дисков.
- `--fsprofile=` — профиль файловой системы, который определяет набор параметров файловой системы. Этот параметр требует наличия файла конфигурации. Так, в файловых системах `ext2`, `ext3`, `ext4` файл расположен в `/etc/mke2fs.conf`.
- `--fstype=` — тип файловой системы. Допустимые значения: `xfs`, `ext2`, `ext3`, `ext4`, `swap`, `vfat`.
- `--fsoptions=` — строка параметров монтирования файловой системы, заключённая в кавычки. В готовой системе эта строка будет записана в `/etc/fstab`.

В системном разделе EFI (`/boot/efi`) Anaconda жёстко кодирует значение и игнорирует указанные пользователем `--fsoptions`.

- `--mkfsoptions=` — дополнительные параметры, которые необходимо передать программе, создающей файловую систему в этом разделе. Список аргументов не обрабатывается, поэтому их необходимо предоставить в формате, который можно передать непосредственно в программу mkfs. Это означает, что параметры должны быть разделены запятыми или заключены в двойные кавычки, в зависимости от файловой системы. Пример:

```
part /opt/foo1 --size=512 --fstype=ext4 --mkfsoptions="-O ^has_journal,^flex_bg,^metadata_
↪csum"

part /opt/foo2 --size=512 --fstype=xfs --mkfsoptions="-m bigtime=0,finobt=0"
```

  Подробнее см. в руководствах файловых систем. Например, `mkfs.ext4` или `mkfs.xfs`.

- `--label=` — метка файловой системы. Если метка уже используется, будет создана новая метка.

- `--noformat` — использует существующее устройство без форматирования RAID-массива.

- `--useexisting` — использует существующее устройство и повторно его форматирует.

- `--encrypted` — включает шифрование RAID-устройства с использованием парольной фразы LUKS из `--passphrase`. Если парольная фраза не задана, Anaconda использует стандартный пароль из строки `autopart --passphrase`, а если и он не задан — будет предложено его ввести во время установки.

  При шифровании одного или нескольких разделов Anaconda пытается собрать 256 бит энтропии, чтобы обеспечить безопасное шифрование разделов. Сбор энтропии может занять некоторое время — процесс остановится максимум через 10 минут, независимо от того, было ли собрано достаточно энтропии. Процесс можно ускорить, взаимодействуя с системой установки (набирая текст на клавиатуре или перемещая мышь). Если вы устанавливаете систему на виртуальную машину, вы также можете подключить к гостевой машине устройство `virtio-rng` (виртуальный генератор случайных чисел).

- `--luks-version=LUKS_VERSION` — версия LUKS для шифрования файловой системы. Используется только если указан параметр `--encrypted`.

- `--cipher=` — ип шифрования, который будет выбран, если стандартного значения `aes-xts-plain64`, используемого в Anaconda, оказалось недостаточно. Этот параметр используется вместе с `--encrypted` и сам по себе силы не имеет. Рекомендуется выбирать тип `aes-xts-plain64` или `aes-cbc-essiv:sha256`.

- `--passphrase=` — парольная фраза для доступа к RAID-устройству. Используется вместе с `--encrypted` и сам по себе силы не имеет.

- `--escrowcert=URL_of_X.509_certificate` — сертификат X.509, с помощью которого зашифрован ключ устройства в `/root`. Этот параметр имеет смысл только при наличии аргумента `--encrypted`.
- `--backuppassphrase` — случайная парольная фраза для устройства. Файлы паролей хранятся в `/root` и зашифрованы с помощью сертификата X.509. Этот параметр используется вместе с `--escrowcert`.
- `--pbkdf=PBKDF` — задаёт алгоритм PBKDF для ключей LUKS. См. также справку `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`.
- `--pbkdf-memory=PBKDF_MEMORY` — задаёт затраты памяти для PBKDF. См. также справку `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`.
- `--pbkdf-time=PBKDF_TIME` — задаёт количество миллисекунд, затрачиваемое на обработку кодовой фразы PBKDF. См. также `--iter-time` в руководстве `cryptsetup(8)`. Этот параметр имеет смысл только если указан `--encrypted`, и является взаимоисключающей с `--pbkdf-iterations`.
- `--pbkdf-iterations=PBKDF_ITERATIONS` — задаёт количество итераций напрямую и не использует тест PBKDF. См. также `--pbkdf-force-iterations` в руководстве `cryptsetup(8)`. Этот параметр имеет смысл только в том случае, если указан `--encrypted`, и является взаимоисключающим с `--pbkdf-time`.

Пример

В следующем примере будут созданы разделы `/` (RAID1) и `/home` (RAID5). Предполагается, что всего установлено три диска SCSI. Также будут созданы три раздела подкачки, по одному на каждом диске.

```
part raid.01 --size=6000 --ondisk=sda
part raid.02 --size=6000 --ondisk=sdb
part raid.03 --size=6000 --ondisk=sdc
part swap --size=512 --ondisk=sda
part swap --size=512 --ondisk=sdb
part swap --size=512 --ondisk=sdc
part raid.11 --size=1 --grow --ondisk=sda
part raid.12 --size=1 --grow --ondisk=sdb
part raid.13 --size=1 --grow --ondisk=sdc
raid / --level=1 --device=rhel8-root --label=rhel8-root raid.01 raid.02 raid.03
raid /home --level=5 --device=rhel8-home --label=rhel8-home raid.11 raid.12 raid.13
```

Замечания

- При утере парольной фразы LUKS, любые зашифрованные разделы и их данные станут полностью недоступными. Восстановить утерянную парольную фразу невозможно. Однако вы можете сохранить кодовые фразы шифрования с помощью `--escrowcert` и создать резервные кодовые фразы шифрования с помощью `--backuppassphrase`.

reqpart (дополнительная)

Автоматически создаёт разделы, требующиеся для аппаратной платформы:

- `/boot/efi` для систем с UEFI
- `biosboot` для систем с BIOS и GPT
- `PRePBoot` для IBM Power Systems.

Синтаксис

```
reqpart [--add-boot]
```

Дополнительные параметры

- `--add-boot` — создать отдельный раздел `/boot` в дополнение к разделу для конкретной платформы, созданному базовой командой.

Замечания

- Нельзя использовать вместе с `autopart`, поскольку `autopart` делает всё то же, что и `reqpart` и, кроме того, создаёт другие разделы или логические тома, такие как `/` и `swap`. В отличие от `autopart`, эта команда создаёт только разделы для конкретной платформы и оставляет остальную часть диска пустой, что позволяет вам создать разделы по вашему желанию.

snapshot (дополнительная)

Создаёт снимки динамических томов LVM во время процесса установки. Это позволяет создавать резервные копии логического тома до или после установки.

Чтобы создать несколько снимков, добавьте команду `snaphost` в Kickstart-файл несколько раз.

Синтаксис

```
snapshot vg_name/lv_name --name=snapshot_name --when=pre-install|post-install
```

Дополнительные параметры

- `vg_name/lv_name` — задаёт имя группы томов и логического тома, из которого создается снимок.
- `--name=snapshot_name` — имя снимка. Это имя должно быть уникальным в пределах группы томов.
- `--when=pre-install|post-install` — создавать снимок до начала установки или после её завершения.

volgroup (дополнительная)

Создание группы LVM.

Синтаксис

```
volgroup name [OPTIONS] [partition*]
```

Обязательные параметры

- `name` — имя новой группы томов.

Дополнительные параметры

- `partition` — разделы физических томов, которые будут использоваться в качестве резервного хранилища для группы томов.
- `--noformat` — использует существующую группу томов без форматирования.
- `--useexisting` — использует существующую группу томов и повторно ее форматирует. Не указывайте раздел при использовании этого параметра. Пример:

  ```
  volgroup rhel00 --useexisting --noformat
  ```

- `--pesize=` — размер физических экстентов в КБ. Значение по умолчанию — 4096 (4 МБ), минимальное значение — 1024 (1 МБ).
- `--reserved-space=` — размер пространства в мегабайтах, которое должно быть зарезервировано в группе томов. Применимо только к новым группам томов.
- `--reserved-percent=` — процентная доля пространства, зарезервированного в группе томов. Применимо только к новым группам томов.

Замечания

- Пример создания раздела, группы логических томов и отдельного тома:

  ```
  part pv.01 --size 10000
  volgroup my_volgrp pv.01
  logvol / --vgname=my_volgrp --size=2000 --name=root
  ```

- Имена логических томов и их групп не должны содержать дефис – в противном случае в списке `/dev/mapper/` они будут включать по два дефиса. То есть если группа `volgrp-01` содержит том `logvol-01`, его имя будет иметь вид `/dev/mapper/volgrp--01-logvol--01`. Это ограничение накладывается только на новые группы и тома. Имена существующих томов (см. `--noformat`) не изменятся.

zerombr (дополнительная)

Команда `zerombr` заново инициализирует недействительные таблицы разделов. Это приведет к потере данных на дисках с этими таблицами.

Эта команда требуется при выполнении установки в 64-битной IBM System Z с неотформатированными DASD-дисками, в противном случае неотформатированные диски не форматируются и не используются во время установки.

Синтаксис

```
zerombr
```

Замечания

- Если в ходе выполнения установки на платформах IBM System Z были обнаружены неотформатированные DASD-устройства (Direct Access Storage Device), они будут автоматически отформатированы с помощью `dasdfmt`. Эта команда подавляет запросы подтверждения пользователя в ходе интерактивной установки.
- Если `zerombr` не задан и программа установки обнаружила неотформатированное DASD-устройство, Kickstart-установка завершится неудачей.
- Если `zerombr` не указан и программа установки обнаружила хотя бы одно неотформатированное DASD-устройство, но пользователь не подтвердил форматирование дисков, интерактивная установка завершит работу. Чтобы это предотвратить, следует подключать только те диски, которые будут использоваться в процессе установки. Остальные устройства можно будет добавить позднее.
- У этой команды нет параметров.

zfcp (дополнительная)

Применимо к IBM System Z и определяет FC-устройство (Fibre Channel).

Синтаксис

```
zfcp --devnum=devnum [--wwpn=wwpn --fcplun=lun]
```

Дополнительные параметры

- `--devnum=` — номер устройства (идентификатор шины адаптера zFCP).
- `--wwpn=` — шестнадцатеричный номер WWPN (World Wide Port Name) с префиксом `0x`.

- `--fcplun=` — шестнадцатеричный номер LUN (Logical Unit Number) с префиксом `0x`

Замечания

Достаточно указать идентификатор шины устройства FCP, если доступно автоматическое сканирование LUN и при установке версии 9 и более поздних. В противном случае обязательны все три параметра. Автоматическое сканирование LUN доступно для устройств FCP, работающих в режиме NPIV, если оно не отключено через параметр модуля `zfcp.allow_lun_scan` (включено по умолчанию). Он обеспечивает доступ ко всем устройствам SCSI, обнаруженным в сети хранения данных, подключённым к устройству FCP с указанным идентификатором шины.

Пример

```
zfcp --devnum=0.0.4000 --wwpn=0x5005076300C213e9
--fcplun=0x5022000000000000
zfcp --devnum=0.0.4000
```

Команды для надстроек, поставляемых с программой установки

Команды, относящиеся к надстройкам, поставляемым по умолчанию с программой установки: Kdump и OpenSCAP.

%addon com_redhat_kdump (дополнительная)

Команда для настройки механизма kdump.

Синтаксис

```
%addon com_redhat_kdump [OPTIONS]
%end
```

> **Примечание**
>
> Синтаксис этой команды необычен, поскольку это надстройка, а не встроенная Kickstart-команда.

Параметры

- `--enable` — включить kdump в установленной системе.
- `--disable` — выключить kdump в установленной системе.
- `--reserve-mb=` — объём памяти, зарезервированный для kdump, в МБ. Пример:

```
%addon com_redhat_kdump --enable --reserve-mb=128
%end
```

Вы также можете указать `auto` вместо числового значения. В этом случае программа установки определит объём памяти автоматически. Если вы включите kdump и не укажете опцию `--reserve-mb=`, будет использоваться значение `auto`.

- `--enablefadump` — включите дамп с помощью встроенного ПО в системах, которые это позволяют (в частности, на серверах IBM Power Systems).

Замечания

- Kdump — это механизм дампа ядра, который позволяет сохранять содержимое системной памяти для последующего анализа. Он опирается на `kexec`, который можно использовать для загрузки ядра Linux из контекста другого ядра без перезагрузки системы и сохранения содержимого памяти первого ядра, которое в противном случае было бы потеряно.
- В случае сбоя системы `kexec` загружается во второе ядро (ядро захвата). Это ядро захвата находится в зарезервированной части системной памяти. Затем Kdump захватывает содержимое памяти повреждённого ядра (аварийный дамп) и сохраняет его в указанном месте. Местоположение невозможно настроить с помощью описываемой Kickstart-команды; его необходимо настроить после установки, отредактировав файл конфигурации `/etc/kdump.conf`.

%addon com_redhat_oscap (дополнительная)

Надстройка программы установки **OpenSCAP** используется для применения содержимого SCAP (протокол автоматизации управления данными безопасности) — политик безопасности — в установленной системе. Это дополнение включено по умолчанию. Если параметр включён, пакеты, необходимые для обеспечения этой функциональности, будут установлены автоматически. Однако по умолчанию никакие политики не применяются, а это означает, что никакие проверки не выполняются ни во время, ни после установки, если это специально не настроено.

Примечание

Применение политик безопасности необходимо не во всех системах. Эту команду следует использовать только в том случае, если определённая политика предусмотрена правилами организации или государством.

В отличие от большинства других команд, эта команда не принимает обычные параметры, а вместо этого использует пары ключ-значение в теле определения `%addon`. Эти пары не зависят от наличия пробелов. Значения могут быть дополнительно

заключены в одинарные (') или двойные кавычки (").

Синтаксис

```
%addon com_redhat_oscap
key = value
%end
```

Ключи

- `content-type` — тип содержимого безопасности. Возможные значения: `datastream`, `archive`, `rpm` и `scap-security-guide`. Если `content-type` определён как `scap-security-guide`, надстройка будет использовать содержимое, предоставленное пакетом `scap-security-guide`, который есть на загрузочном носителе. Это означает, что все остальные ключи, кроме `profile`, не будут иметь никакого эффекта.
- `content-url` — расположение содержимого безопасности. Доступно по протоколу HTTP, HTTPS или FTP; расположение в локальном хранилище в настоящее время не поддерживается. Для доступа к определениям контента в удалённом месте должно быть доступно сетевое соединение.
- `datastream-id` — идентификатор потока данных, указанный в значении `content-url`. Используется только в том случае, если `content-type` указан как `datastream`.
- `xccdf-id` — идентификатор критерия, который вы хотите использовать.
- `content-path` — путь к потоку данных или файлу XCCDF, который следует использовать; указывается как относительный путь в архиве.
- `profile` — идентификатор применяемого профиля. Используйте `default`, чтобы применить профиль по умолчанию.
- `fingerprint` — контрольная сумма MD5, SHA1 или SHA2 контента, на который ссылается `content-url`.
- `tailoring-path` — путь к используемому файлу адаптации; указывается как относительный путь в архиве.

Примеры

- Раздел `%addon com_redhat_oscap`, в котором используется содержимое **scap-security-guide** на установочном носителе:

  ```
  %addon com_redhat_oscap
  content-type = scap-security-guide
  profile = xccdf_org.ssgproject.content_profile_pci-dss
  %end
  ```

- Более сложный пример загрузки пользовательского профиля с веб-сервера:

```
%addon com_redhat_oscap
content-type = datastream
content-url = http://www.example.com/scap/testing_ds.xml
datastream-id = scap_example.com_datastream_testing
xccdf-id = scap_example.com_cref_xccdf.xml
profile =  xccdf_example.com_profile_my_profile
fingerprint = 240f2f18222faa98856c3b4fc50c4195
%end
```

Команды, используемые в Anaconda

Команда `pwpolicy` специфична для пользовательского интерфейса Anaconda, её можно использовать только в разделе `%anaconda` Kickstart-файла.

pwpolicy (дополнительная, устарела)

Команда `pwpolicy` используется, чтобы применить пользовательскую политику паролей во время установки. Политика требует от вас создания паролей для учетных записей root, пользователей или пользователей luks. Такие факторы, как длина и надёжность пароля, определяют его достоверность.

Синтаксис

```
pwpolicy name [--minlen=length] [--minquality=quality] [--strict|--notstrict] [--emptyok|--
↪notempty] [--changesok|--nochanges]
```

Обязательные параметры

- `name` — замените на `root`, `user` или `luks`, чтобы применить политику для пароля root, паролей пользователей или кодовой фразы LUKS соответственно.

Дополнительные параметры

- `--minlen=` — минимальная допустимая длина пароля в символах. По умолчанию — `6`.
- `--minquality=` — минимально допустимая сложность пароля, определённая библиотекой `libpwquality`. Значение по умолчанию — `1`.
- `--strict` — включает строгое соблюдение пароля. Пароли, не соответствующие требованиям, указанным в `--minquality=` и `--minlen=`, не принимаются. По умолчанию отключена.
- `--notstrict` — пароли, которые не соответствуют минимальным требованиям, указанным в `--minquality=` и `--minlen=`, будут приняты после двойного нажатия кнопки «Готово» в графическом интерфейсе. Для текстового режима используется аналогичный механизм.
- `--emptyok` — разрешить использование пустых паролей. Включено по умолчанию для паролей пользователей.

- `--notempty` — запретить использование пустых паролей. Включено по умолчанию для пароля root и кодовой фразы LUKS.
- `--changesok` — изменить пароль в пользовательском интерфейсе, даже если в Kickstart-файле он уже указан. По умолчанию отключено.
- `--nochanges` — запретить изменение пароля, установленного в Kickstart-файле. Включено по умолчанию.

Замечания

- Команду `pwpolicy` можно использовать только в разделе `%anaconda` Kickstart-файла.
- Для проверки минимальных требований к паролю (длина и сложность) используется библиотека `libpwquality`. Вы можете использовать команды `pwscore` и `pwmake`, входящие в пакет `libpwquality` для проверки сложности пароля или для создания случайного пароля с заданным рейтингом.

Команды восстановления системы

Команды, использующиеся для восстановления установленной системы.

rescue (дополнительная)

Команда предоставляет среду оболочки с правами root и набор инструментов управления системой для восстановления установки и устранения таких проблем, как:

- монтирование файловых систем только для чтения
- чёрный список или добавление драйвера, имеющегося на диске
- установка или обновление системных пакетов
- управление разделами

Примечание

Режим восстановления Kickstart отличается от режима восстановления и аварийного режима, которые предоставляются как часть systemd и диспетчера служб.

Команда `rescue` сама по себе не модифицирует систему. Она только устанавливает среду восстановления, монтируя систему в `/mnt/sysimage` в режиме чтения-записи. Вы можете не монтировать систему или монтировать ее в режиме только для чтения.

Синтаксис

```
rescue [--nomount|--romount]
```

Дополнительные параметры

- `--nomount` или `--romount` — контролирует способ монтирования установленной системы в среде восстановления. По умолчанию программа установки находит систему и монтирует её в режиме чтения-записи, сообщая вам, где она выполнила это монтирование. При желании вы можете ничего не монтировать (опция `--nomount`) или монтировать в режиме только для чтения (опция `--romount`). Можно использовать только один из двух параметров.

Замечания

Для запуска режим восстановления, сделайте копию Kickstart-файла и включите в него команду `rescue`.

При использовании команды `rescue` установщик будет выполнять следующие шаги:

1. Запуск сценария `%pre`.
2. Настройка среды для режима восстановления. Будут выполнены следующие Kickstart-команды:
   - updates
   - sshpw
   - logging
   - lang
   - network
3. Настройка расширенной среды хранения. Будут выполнены следующие Kickstart-команды:
   - fcoe
   - iscsi
   - iscsiname
   - nvdimm
   - zfcp
4. Монтирование системы:

   ```
   rescue [--nomount|--romount]
   ```

5. Запуск сценария `%post`. Этот шаг выполняется только в том случае, если установленная система смонтирована в режиме чтения-записи.
6. Запуск оболочки.

7. Перезагрузка системы.

Подключение учётной записи Microsoft Exchange

В МСВСфера ОС вы можете подключить учётную запись Microsoft Exchange несколькими способами:

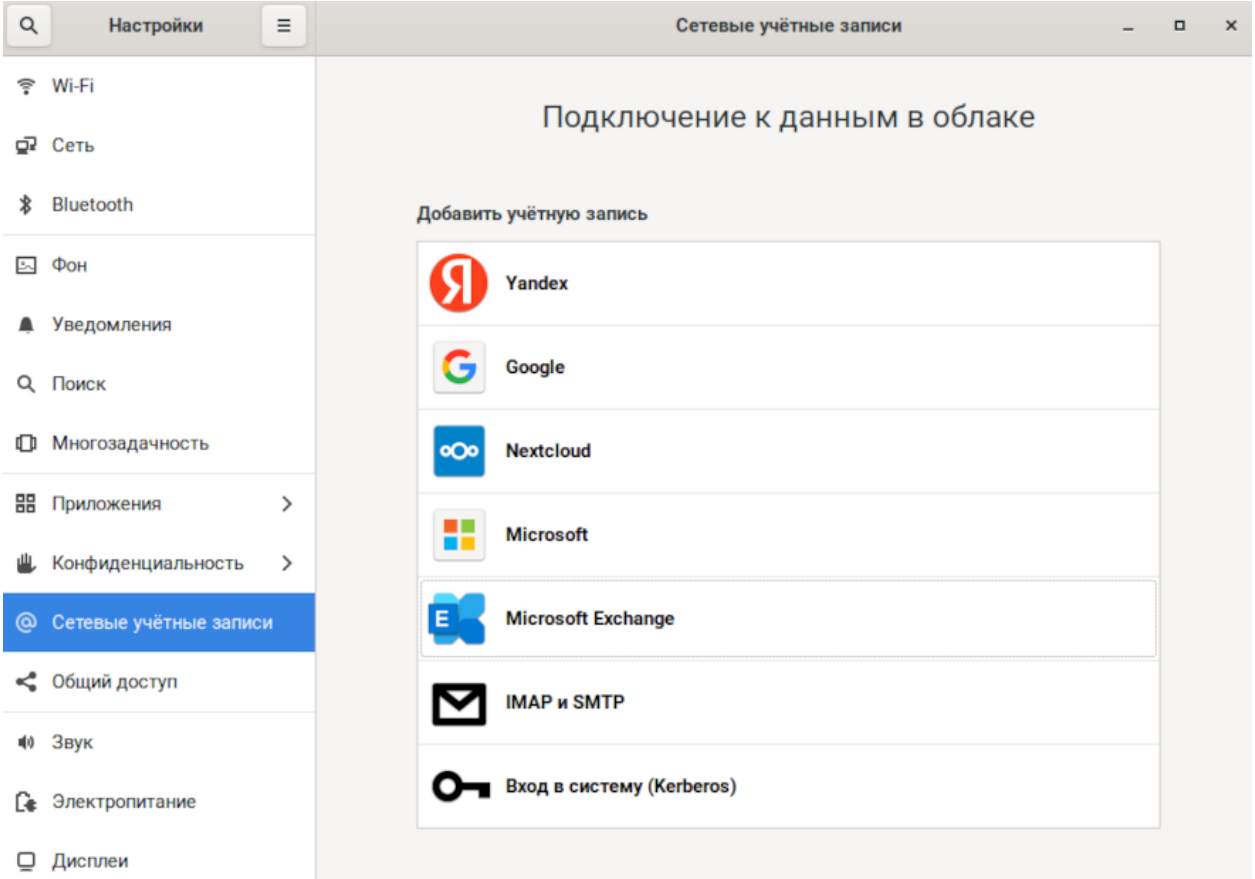
- Через «Сетевые учётные записи».
- В почтовом клиенте Evolution.

Подключение учётной записи Microsoft Exchange через «Сетевые учётные записи»

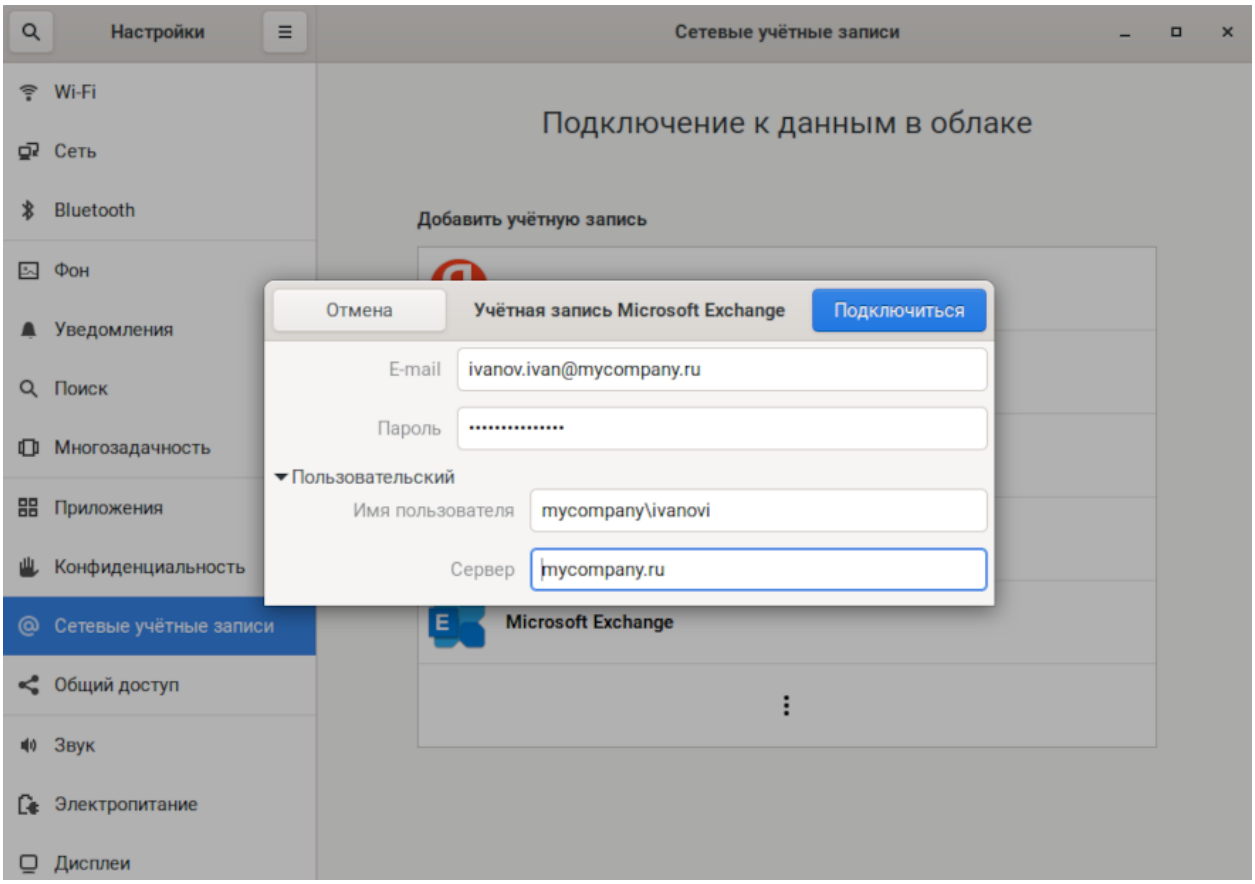
Для подключения учётной записи Microsoft Exchange вам могут понадобиться дополнительные настройки. Если нужные настройки неизвестны, обратитесь к вашему системному администратору или поставщику услуг электронной почты.

Затем выполните следующие действия.

1. Перейдите в «Настройки» → «Сетевые учётные записи», затем нажмите «Microsoft Exchange».

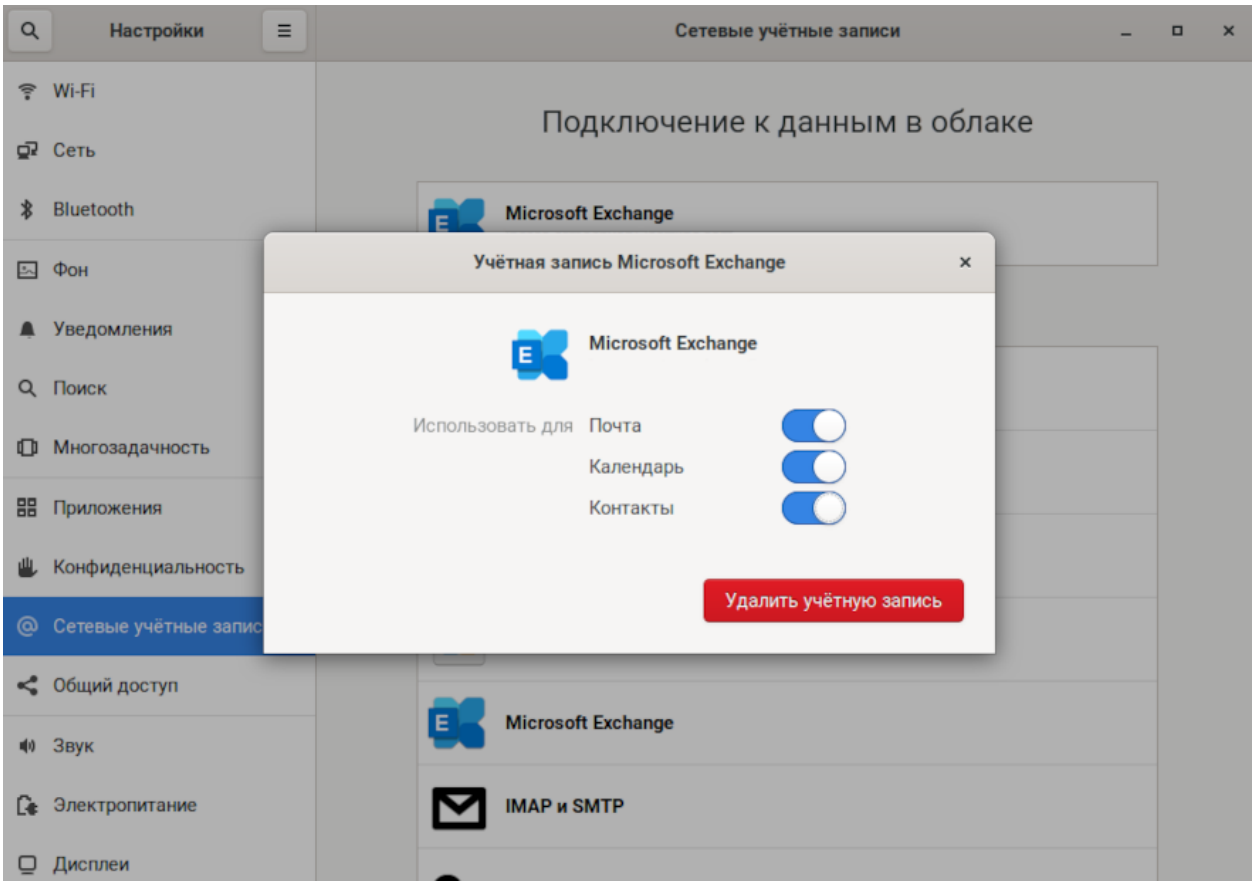


2. Укажите адрес электронной почты и пароль.
3. Затем нажмите на «Пользовательский».
4. Имя пользователя и сервер подставляются автоматически. Если требуются другие настройки, укажите их. Если нужные настройки неизвестны, обратитесь к поставщику услуг или вашему системному администратору.



Здесь `mycompany` — имя домена Microsoft Exchange.

5. Нажмите «Подключиться».
6. Выберите сервисы, которые будут подключены: «Почта», «Календарь», «Контакты».



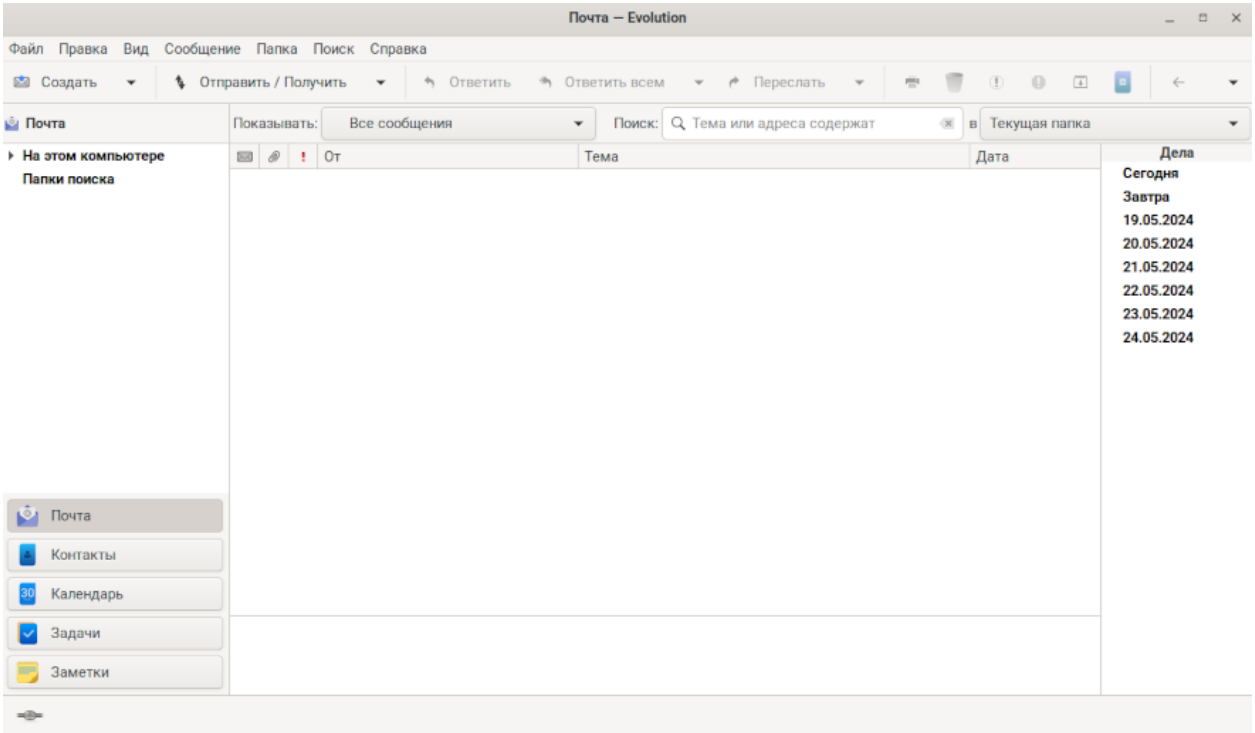
7. Закройте окно. Ваша учётная запись добавлена! Также автоматически будут подключены выбранные сервисы.

Подключение учётной записи Microsoft Exchange в почтовом клиенте Evolution

Вы также можете подключить учётную запись Microsoft Exchange сразу в почтовом клиенте Evolution. Вам также могут понадобиться настройки, которые вы можете узнать у вашего системного администратора или поставщика услуг электронной почты.

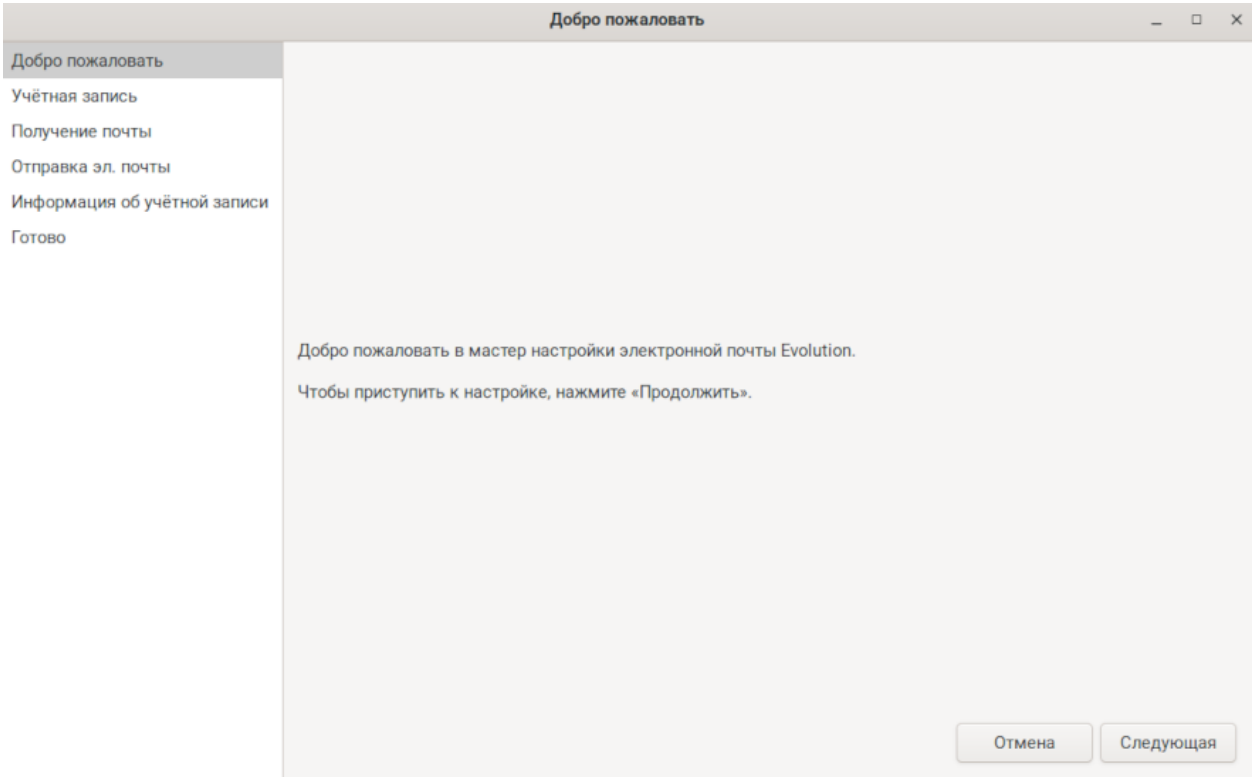
Затем выполните следующие действия.

1. Откройте приложение «Почта Evolution». Если не добавлено ни одной учётной записи, «Мастер настройки» запустится автоматически.

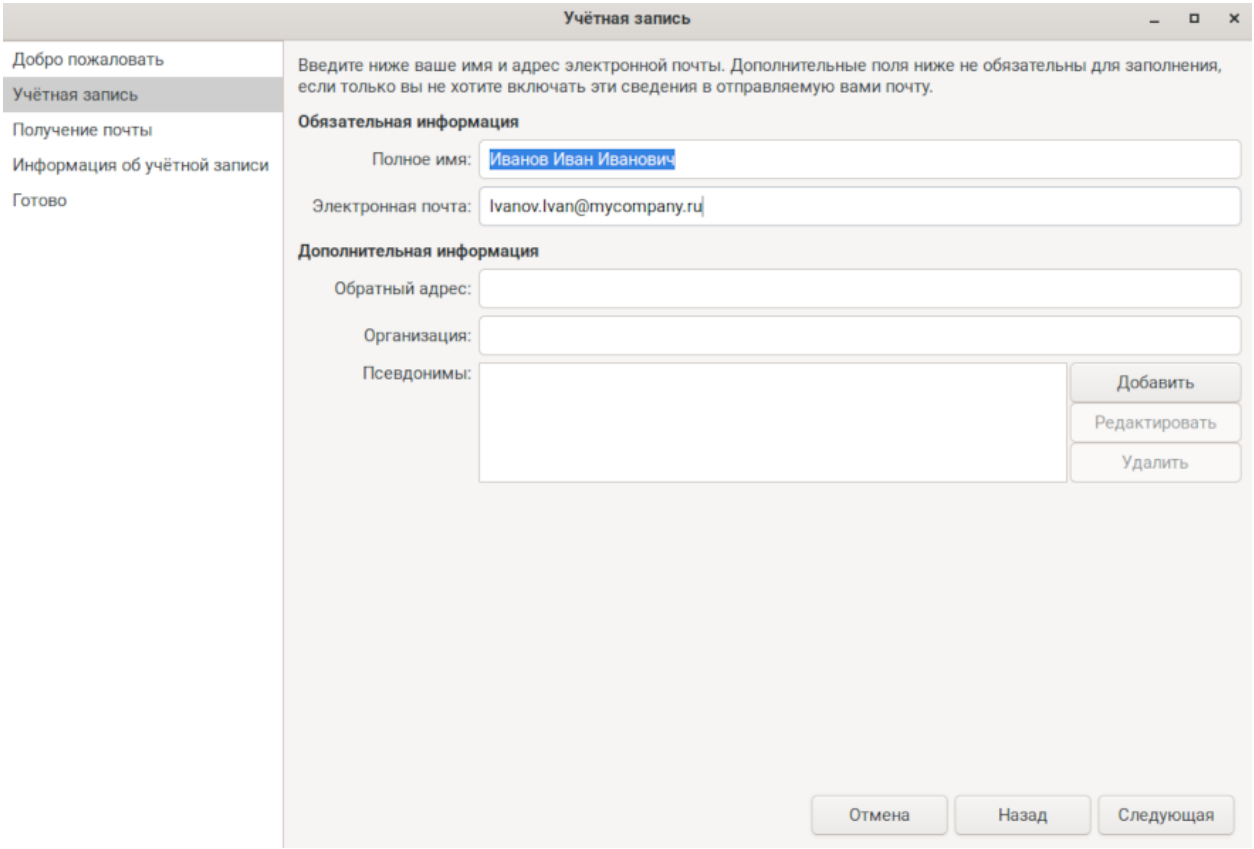


2. Если уже есть добавленная учётная запись, то выберите «Файл» → «Создать» → «Учётная запись почты».

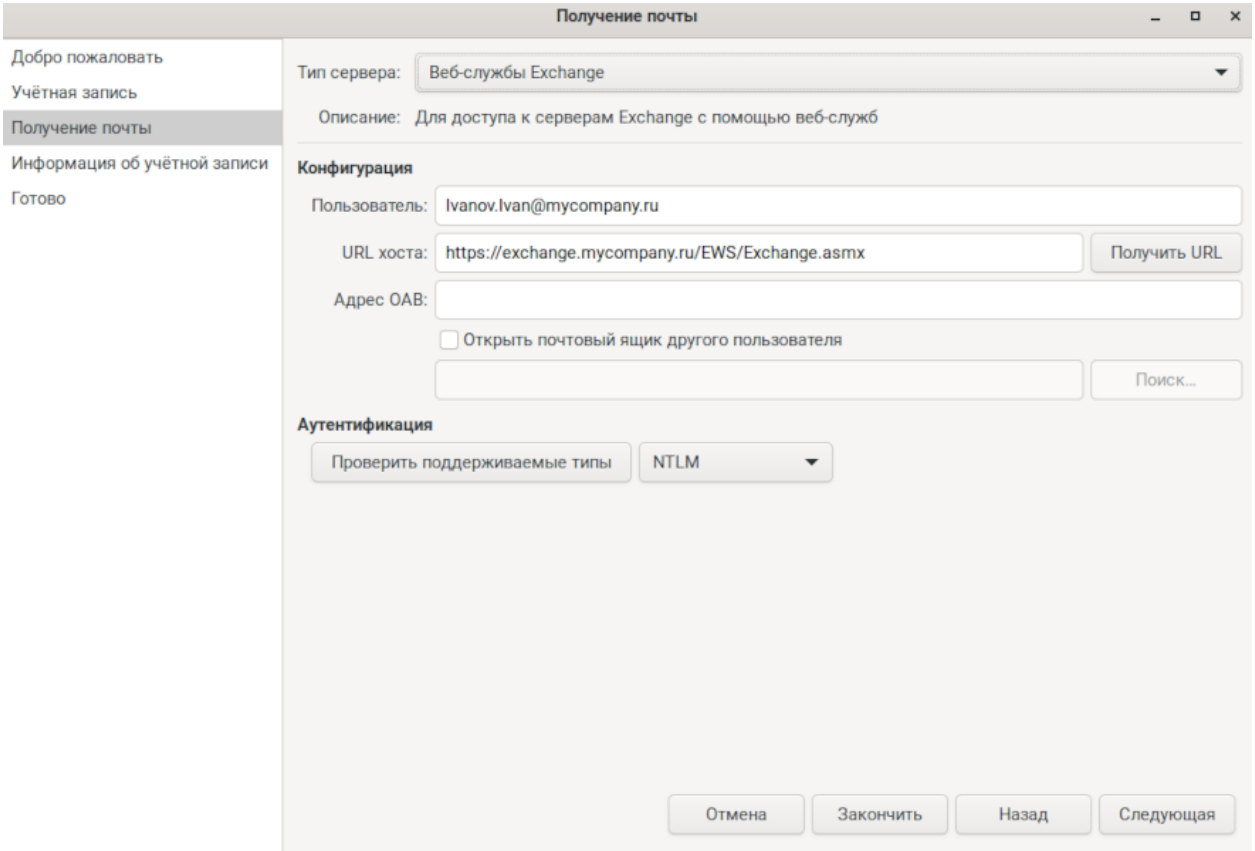
3. Откроется «Мастер настройки». Чтобы приступить к настройке, нажмите «Следующая».



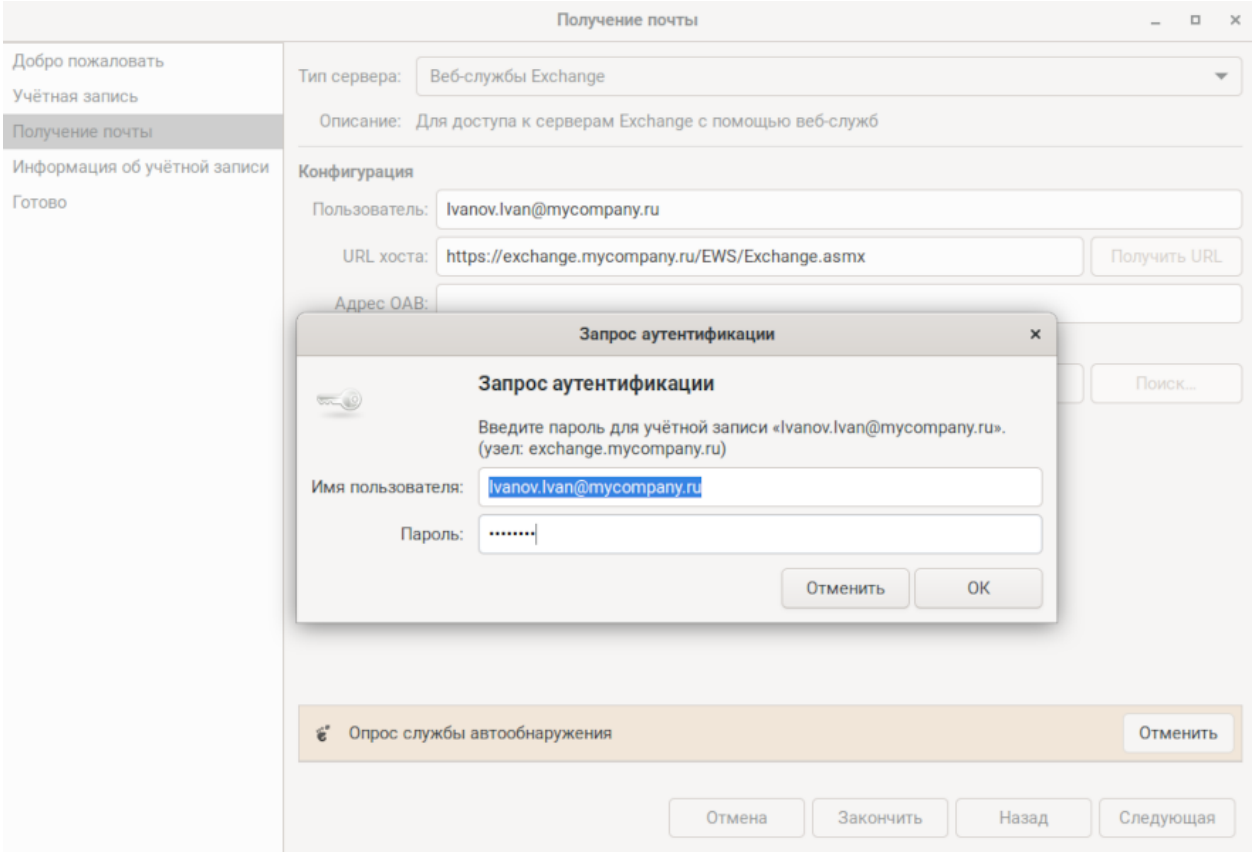
4. Укажите имя и адрес электронной почты и нажмите «Следующая».



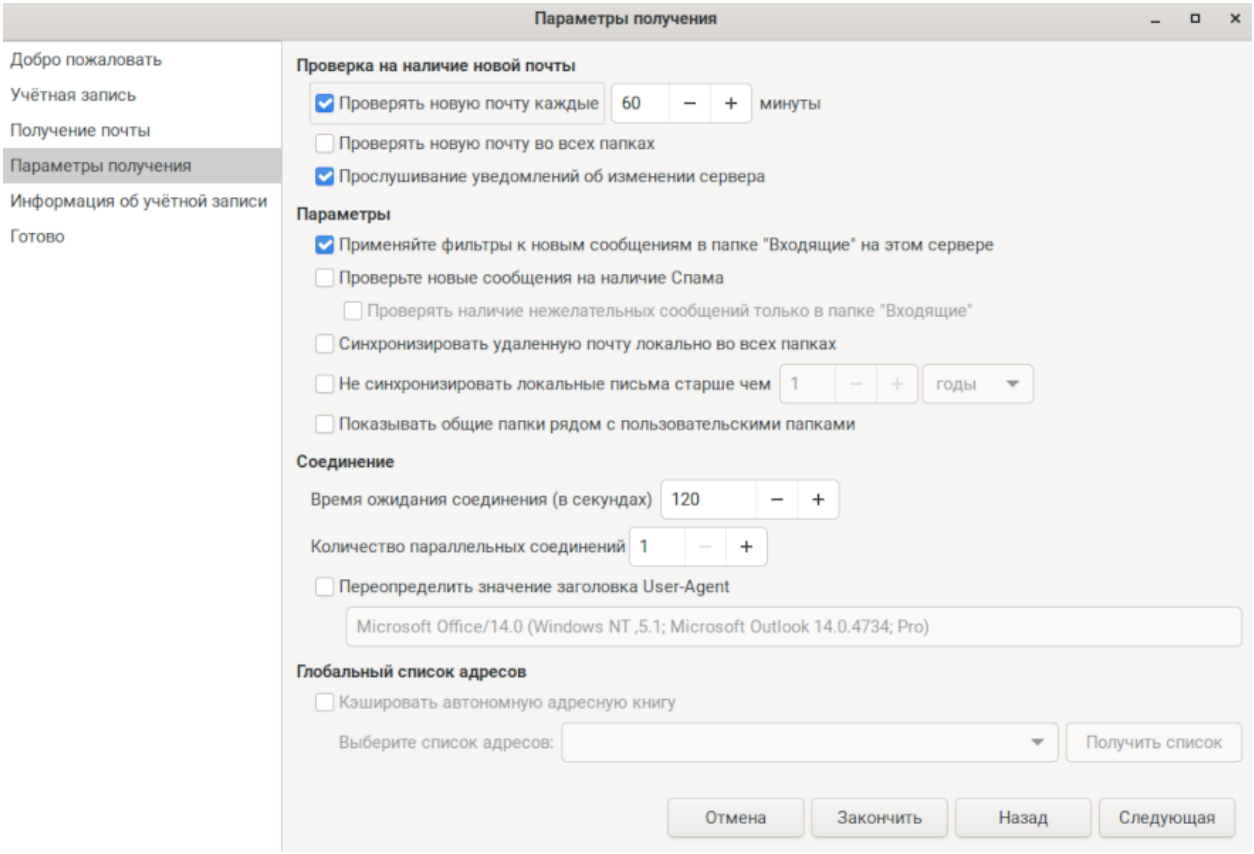
5. Выберете тип сервера «Веб-службы Exchange». Укажите адрес и нажмите кнопку «Получить URL».



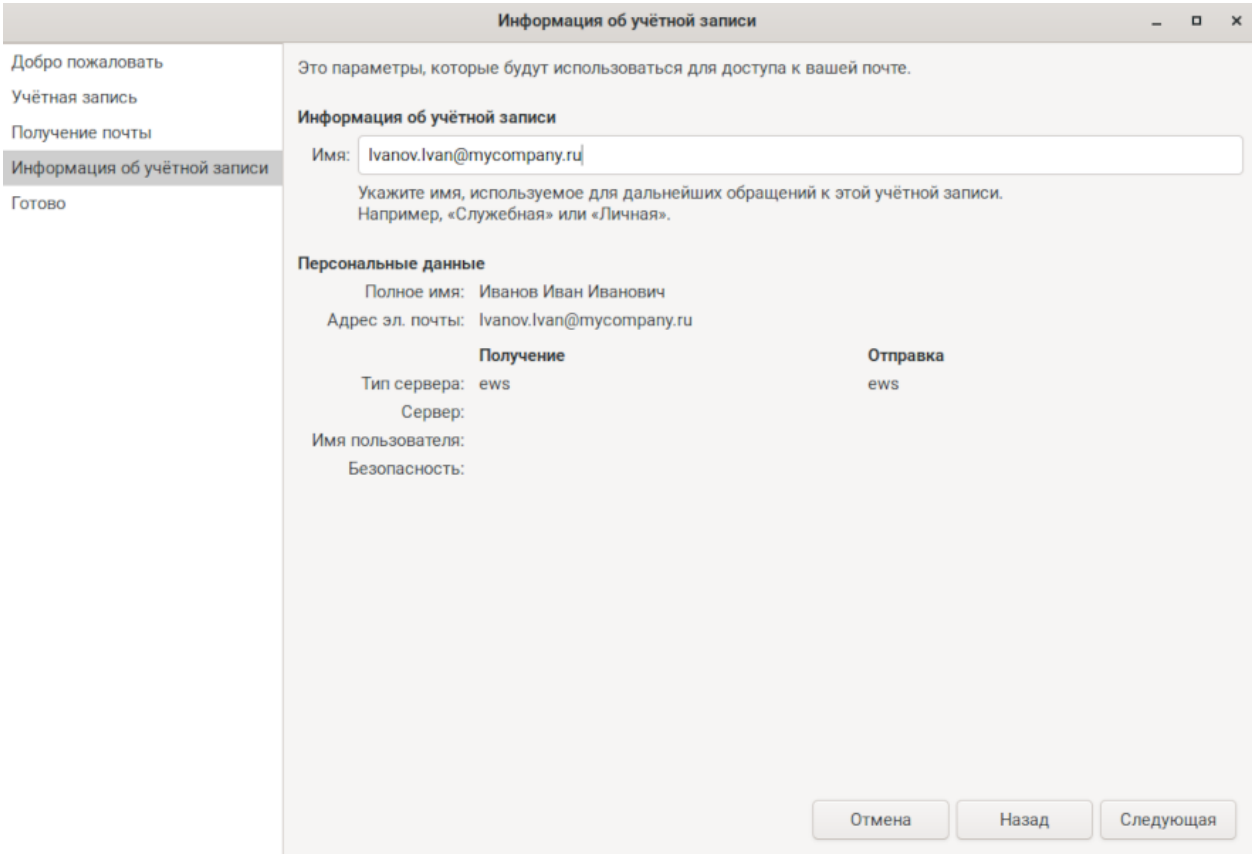
6. В открывшемся окне укажите пароль к вашей учётной записи и нажмите кнопку «OK». Если пароль указан верно, сервер вернёт «Адрес OAB». Затем нажмите на «Следующая».



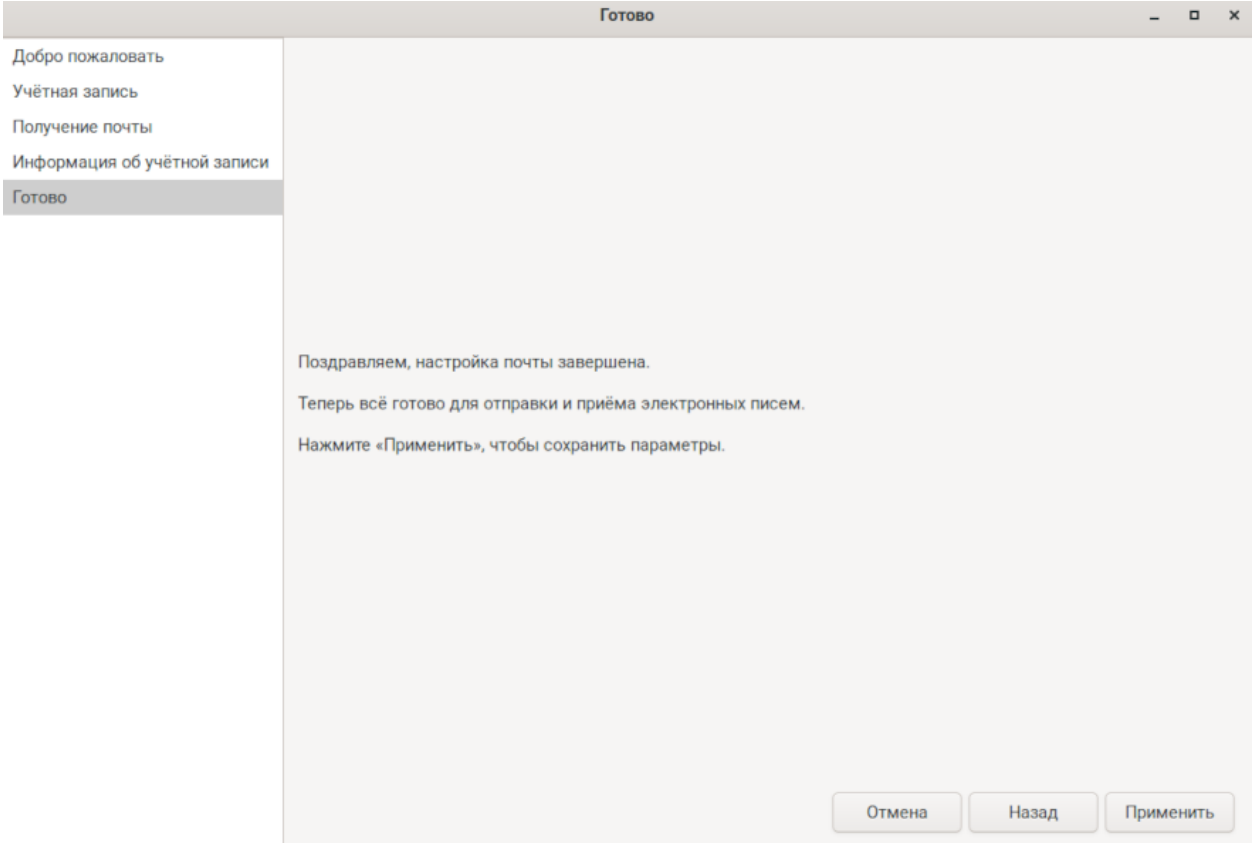
7. Далее настройте параметры получения или оставьте значения по умолчанию. Затем нажмите кнопку «Следующая».



8. Укажите имя для учётной записи или оставьте значение по умолчанию. Затем нажмите кнопку «Следующая».



9. Нажмите кнопку «Применить».



10. Снова введите пароль.
11. Ваша учётная запись добавлена! Также автоматически будут подключены адресная книга, календарь, задачи и заметки.

Обновление ядра системы до версий 6.1 или 6.6

Рассмотрим процедуру обновления ядра системы до версий 6.1 или 6.6. Все команды необходимо выполнять в «Терминале» от имени администратора.

1. Проверьте текущую версию ядра, выполнив следующую команду:

```
$ uname -r
```

2. При необходимости выполните обновление системы:

```
$ sudo dnf update
```

3. Установите пакет для подключения репозитория, содержащего ядра 6.1 и 6.6:

```
$ sudo dnf install sphere-repos-kernelextra
```

4. Для установки нового ядра выполните одну из следующих команд.

- Для установки ядра 6.1 выполните:

```
$ sudo dnf install kernel-lt
```

- Для установки ядра 6.6 выполните:

```
$ sudo dnf install kernel-lt66
```

5. При необходимости выполните обновление зависимых пакетов:

```
$ sudo dnf update
```

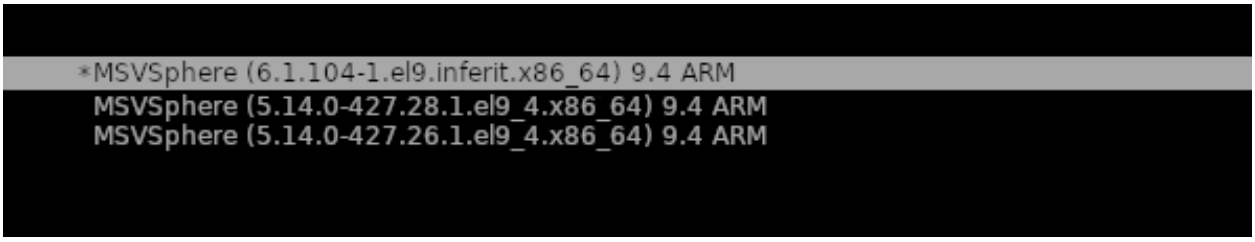
6. Проверьте текущую версию ядра:

```
$ uname -r
```

7. Перезагрузите компьютер:

```
$ reboot
```

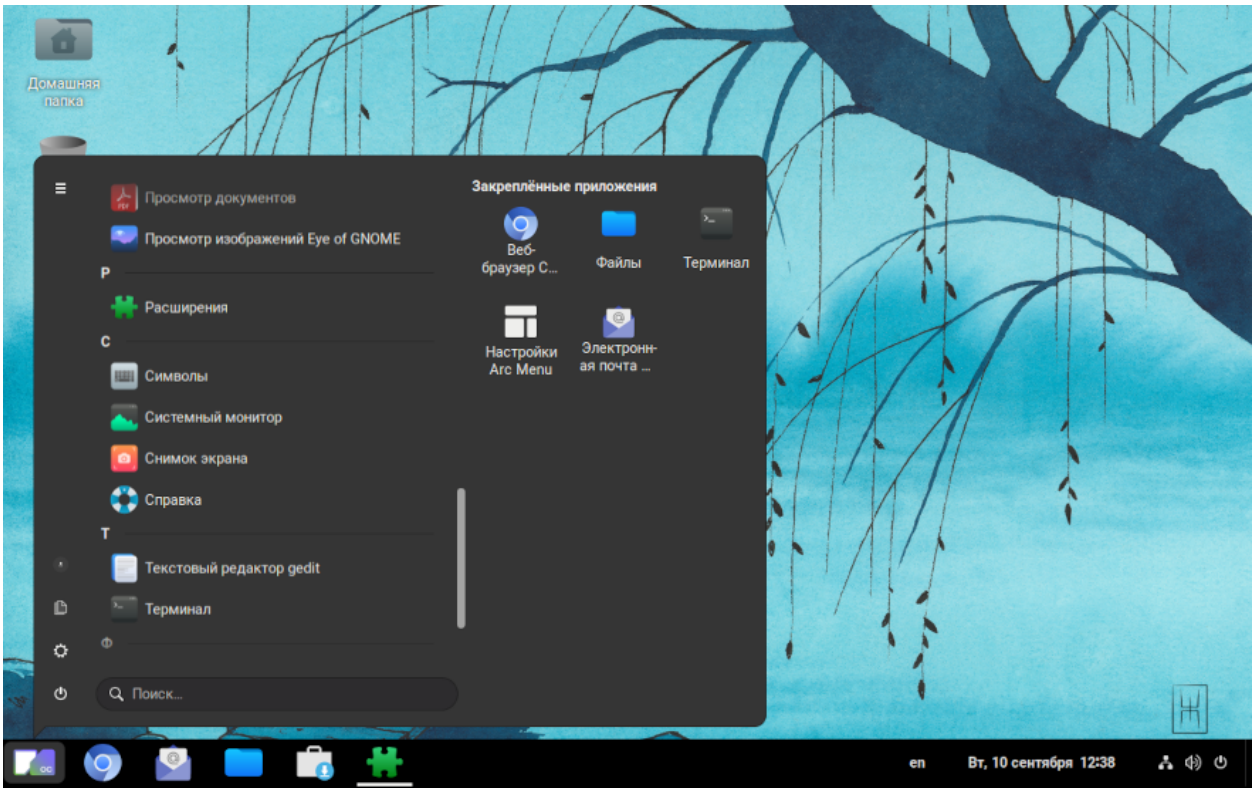
Теперь система будет загружаться с новой версией ядра. Вы также можете выбрать старую версию ядра при необходимости.



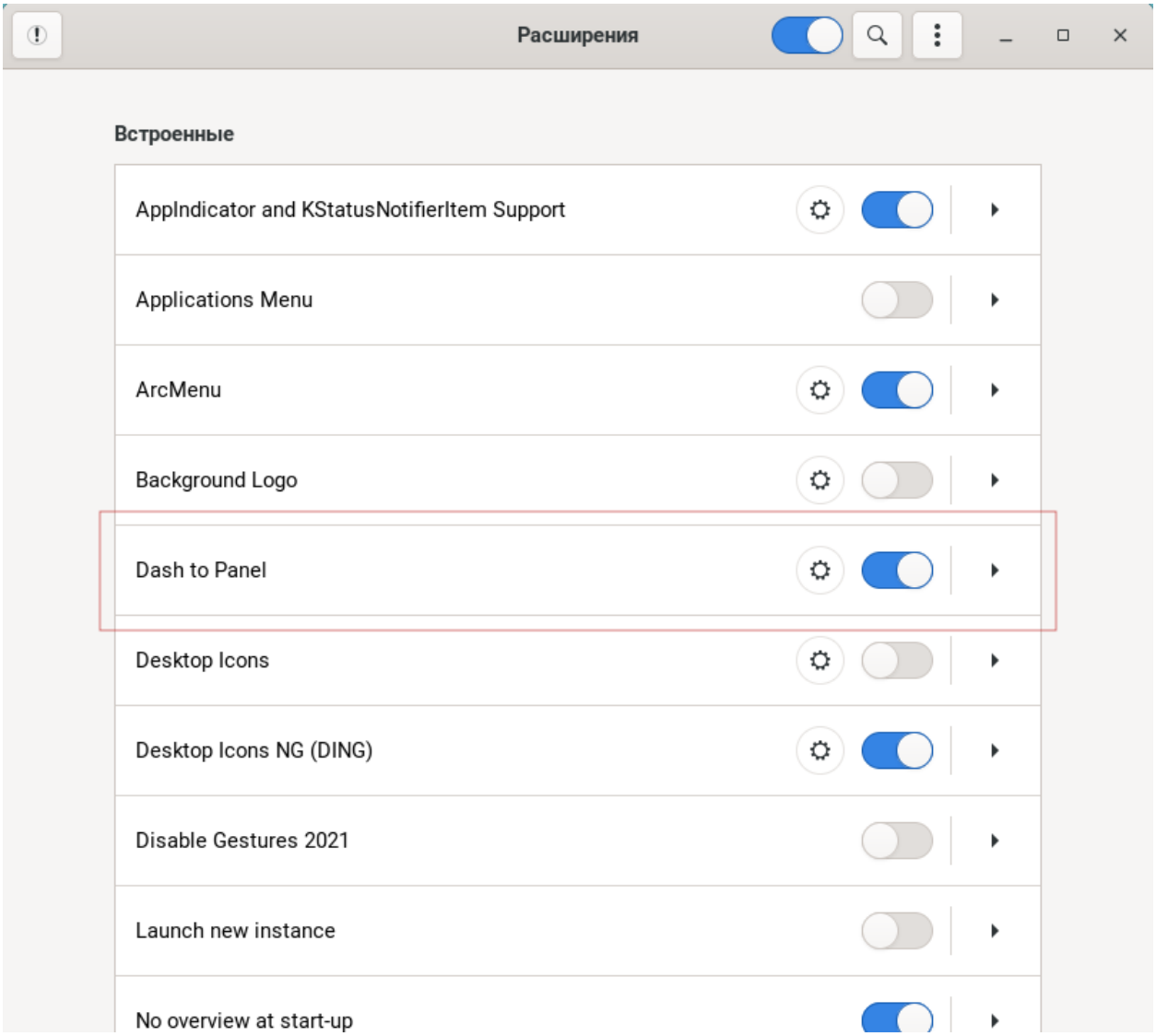
Как перенести пользовательскую панель на экран, который назначен основным в настройках дисплея

Если у вас подключено несколько мониторов, например, к ноутбуку подключён большой монитор, то вы можете настроить систему так, чтобы ваша пользовательская панель, на которой расположена область уведомлений, индикатор раскладки клавиатуры и пр. отображалась на том мониторе, который назначен основным в настройках дисплея. Для этого выполните следующие действия.

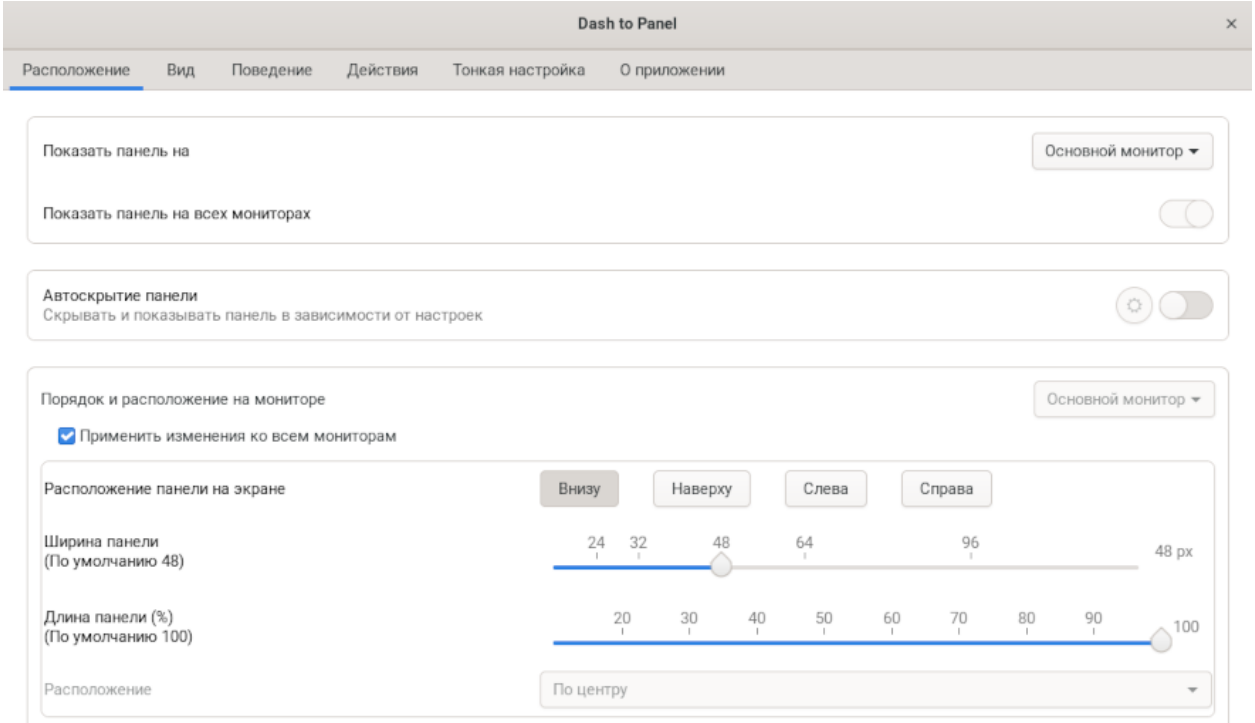
1. Откройте приложение «Расширения» (например, из меню «Пуск»).



2. Найдите среди прочих расширений «Dash to Panel» и нажмите на шестерёнку.



3. Во вкладке «Расположение» выключите «Показать панель на всех мониторах».
4. В «Показать панель на» выберите «основной монитор».



Если по какой-то причине у вас нет расширения «Dash to Panel», то установите его следующей командой:

```
$ sudo dnf install gnome-shell-extension-dash-to-panel
```

Перезапустите систему и снова выполните указанные выше действия.

Также вы можете попробовать сделать основной монитор единственным в настройках, а затем включить дополнительный монитор.

Подключение репозиториев LibreOffice и Virtualization

Репозиторий `LibreOffice` содержит актуальную версию офисного пакета LibreOffice со всеми последними обновлениями. Для того, чтобы подключить репозиторий `LibreOffice`, выполните следующую команду (от имени пользователя `root`):

```
$ sudo dnf install sphere-repos-libreoffice
```

При подключении репозитория `Virtualization` будет добавлена поддержка протокола `SPICE` в виртуализацию QEMU. Для того, чтобы подключить репозиторий `Virtualization`, выполните следующую команду (от имени пользователя `root`):

```
$ sudo dnf install sphere-repos-virtualization
```

Рассмотрим пример подключения репозитория `LibreOffice`.

1. Выполнить команду подключения репозитория:

```
$ sudo dnf install sphere-repos-libreoffice
```

2. Система попросит указать пароль для пользователя `root`:

```
[sudo] пароль для administrator:
```

3. После проверки правильности пароля система проверит зависимости:

```
MSVSphere 9 - AppStream   6.2 MB/s |  15 MB     00:02
MSVSphere 9 - BaseOS      9.7 kB/s | 4.3 kB     00:00
MSVSphere 9 - Extras      4.7 MB/s | 5.8 MB     00:01

Зависимости разрешены.
=================================================================================================
Пакет                           Архитектура     Версия         Репозиторий        Размер
=================================================================================================
Установка:
sphere-repos-libreoffice        noarch          9.4-2.el9      extras             16 k

Результат транзакции
=================================================================================================
Установка  1 Пакет

Объем загрузки: 16 k
Объем изменений: 65 k
```

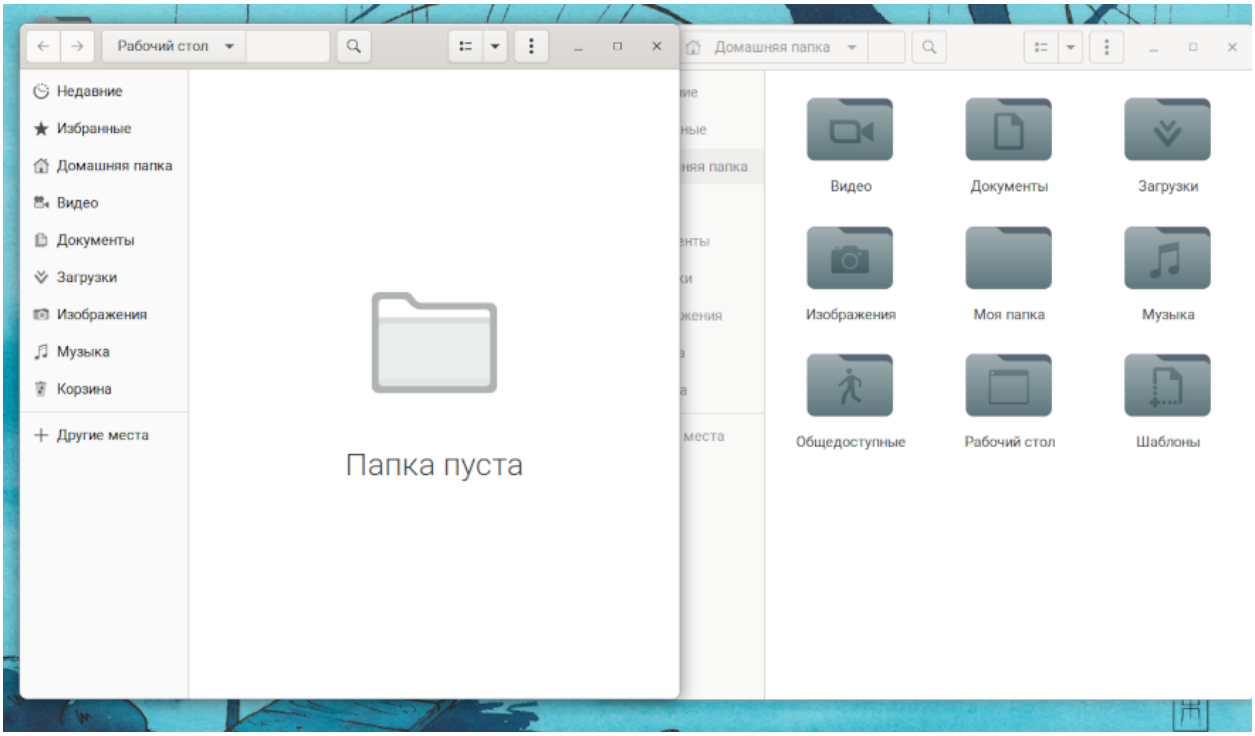
4. И попросит подтверждения загрузки. Введите `д` для подтверждения. После этого начнётся загрузка пакетов:

```
Продолжить? [д/Н]: д
Загрузка пакетов:
sphere-repos-libreoffice-9.4-2.el9.noarch.rpm                      61 kB/s |  16 kB    ␣
↪00:00
-------------------------------------------------------------------------------------------
Общий размер                                                       23 kB/s |  16 kB    ␣
↪00:00
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка       :                                                 1/1
Установка        : sphere-repos-libreoffice-9.4-2.el9.noarch       1/1
Проверка         : sphere-repos-libreoffice-9.4-2.el9.noarch       1/1

Установлен:
sphere-repos-libreoffice-9.4-2.el9.noarch

Выполнено!
```

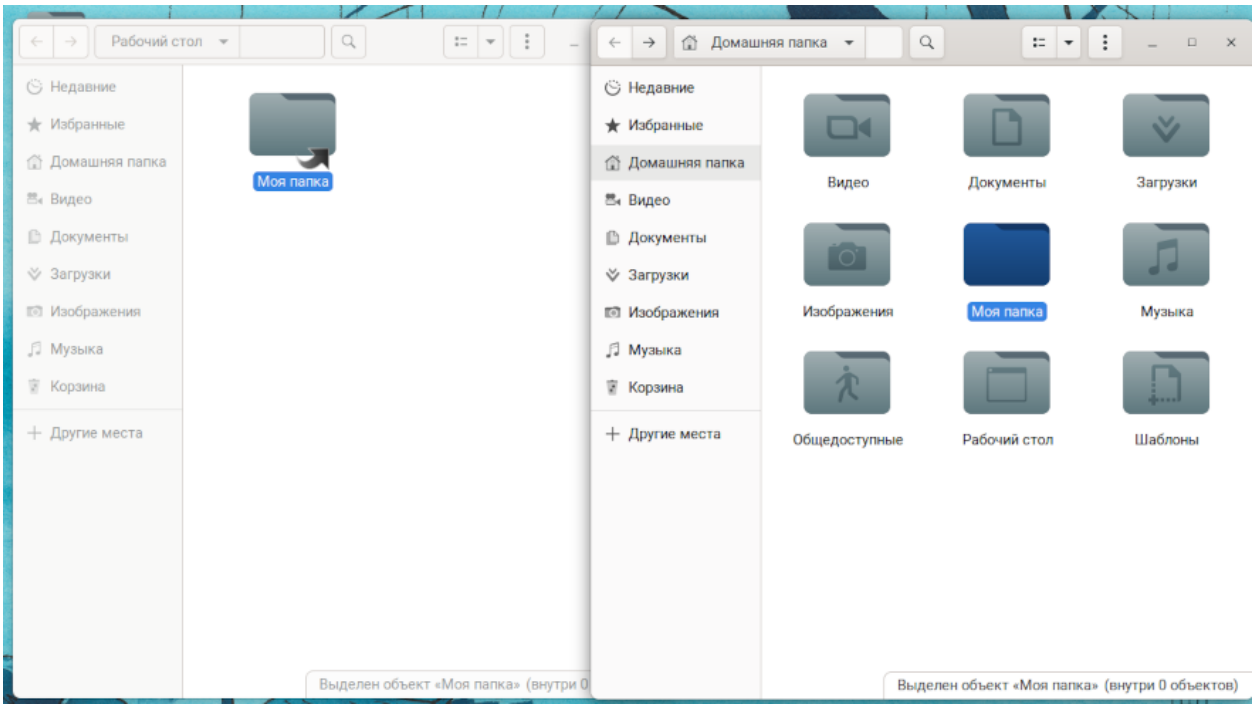
Как создать ярлык на папку на рабочем столе

Для того, чтобы создать ярлык на папку на рабочем столе, выполните следующие действия.

1. Откройте папку «**Рабочий стол**»: «Домашняя папка» → «Рабочий стол».
2. Откройте папку, в которой находится папка, на которую нужно сделать ярлык.



3. Замжмите на клавиатуре клавиши «**ctlr**» и «**shift**» (можно использовать обе клавиши справа), затем перетащите нужную папку левой кнопкой мыши в папку «Рабочий стол». Вы увидите, что в папке «Рабочий стол» появился ярлык на вашу папку.



4. При закрытии окон вы также увидите ярлык на вашу папку на рабочем столе. Теперь вы можете запустить её прямо с рабочего стола.

Автоматическое подключение сетевых ресурсов при входе в сеанс GNOME

Введение

GNOME использует GVfs (GNOME Virtual file system) — подсистему среды рабочего стола GNOME. Это виртуальная файловая система, которая предоставляет команду `gio mount` для монтирования ресурсов.

Порядок действий

Для автоматического подключения сетевых ресурсов при входе в сеанс GNOME выполните следующие действия.

1. Создайте скрипт, например `~/mount-smb.sh`:

   ```
   #!/bin/bash
   gio mount "smb://USERNAME:PASSWORD@server/share"
   ```

   В целях безопасности не рекомендуется указывать пароль в скрипте, лучше использовать хранилище GNOME Keyring. См. «»*Использование хранилища GNOME Keyring*».

2. Сделайте скрипт исполняемым:

   ```
   $ chmod +x ~/mount-smb.sh
   ```

3. Добавьте его в автозагрузку GNOME. Для этого создайте файл `~/.config/autostart/mount-smb.desktop`:

   ```
   [Desktop Entry]
   Type=Application
   Exec=/home/ваш_пользователь/mount-smb.sh
   Hidden=false
   NoDisplay=false
   X-GNOME-Autostart-enabled=true
   Name=Mount SMB Share
   Comment=Automatically mount SMB share on login
   ```

Использование хранилища GNOME Keyring

Чтобы избежать хранения пароля в скрипте, выполните следующие действия:

1. Один раз вручную подключитесь к сетевой папке через «Файлы»:
   - Откройте приложение «Файлы» → «Другие места» → «Подключиться к серверу».
   - Введите `smb://server/share`.

- Укажите имя пользователя и пароль.
- Поставьте галочку «Запомнить навсегда» — это сохранит пароль в GNOME Keyring.

2. После этого путь сохранится в истории GVfs. При запуске GNOME можно использовать следующую команду без указания пароля — он будет взят из Keyring:

```
$ gio mount "smb://server/share"
```

Добавление ярлычка сетевой папки на рабочий стол

Для добавление ярлычка сетевой папки на рабочий стол выполните следующие действия.

1. Найдите, как сетевая папка отображается в файловой системе. Обычно эта информация содержится в `/run/user/$UID/gvfs`:

```
$ ls -l /run/user/$UID/gvfs
итого 0
drwx ------. 1 sphere-user sphere-user 0 июл 25 15:30 'smb-share:server=192.168.10.1,
↪share=resort'
```

   Сетевой ресурс с именем `resort` на сервере `192.168.10.1`.

2. Создайте символическую ссылку на свой рабочий стол:

```
$ ln -s /run/user/$UID/gvfs'smb-share:server=192.168.10.1,share=resort' ~/'Рабочий стол'/
↪сетевая папка
```

   На рабочем столе будет создан ярлык с именем «Сетевая папка».

Практические рекомендации

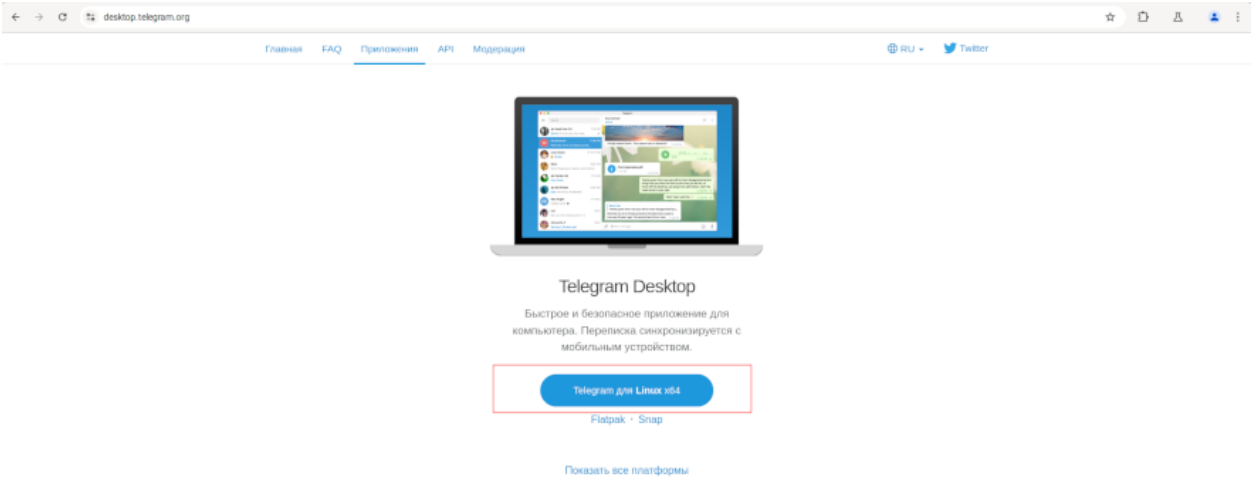
Как установить Telegram на МСВСфера 9 АРМ

Окружение

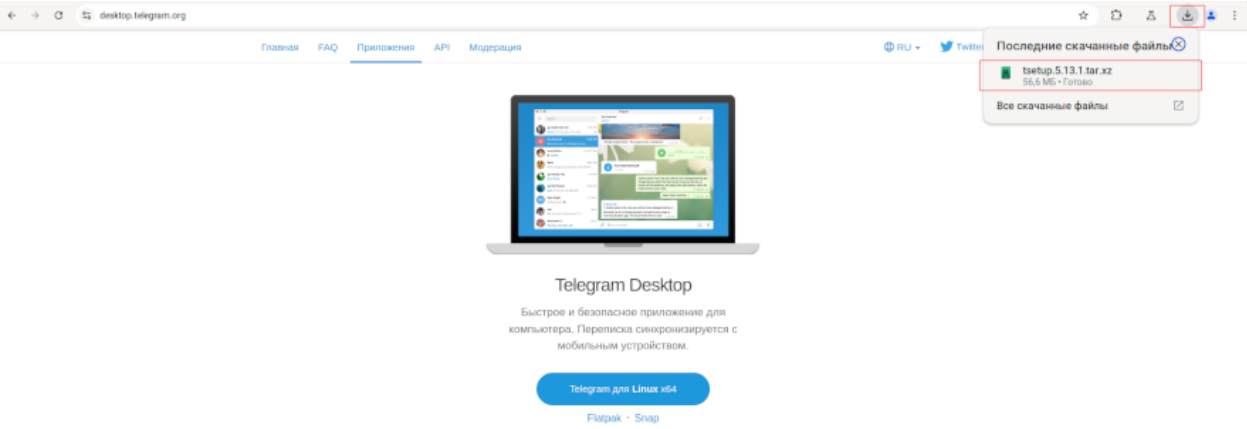
- МСВ Сфера АРМ версии 8.x
- МСВ Сфера АРМ версии 9.x

Установка Telegram

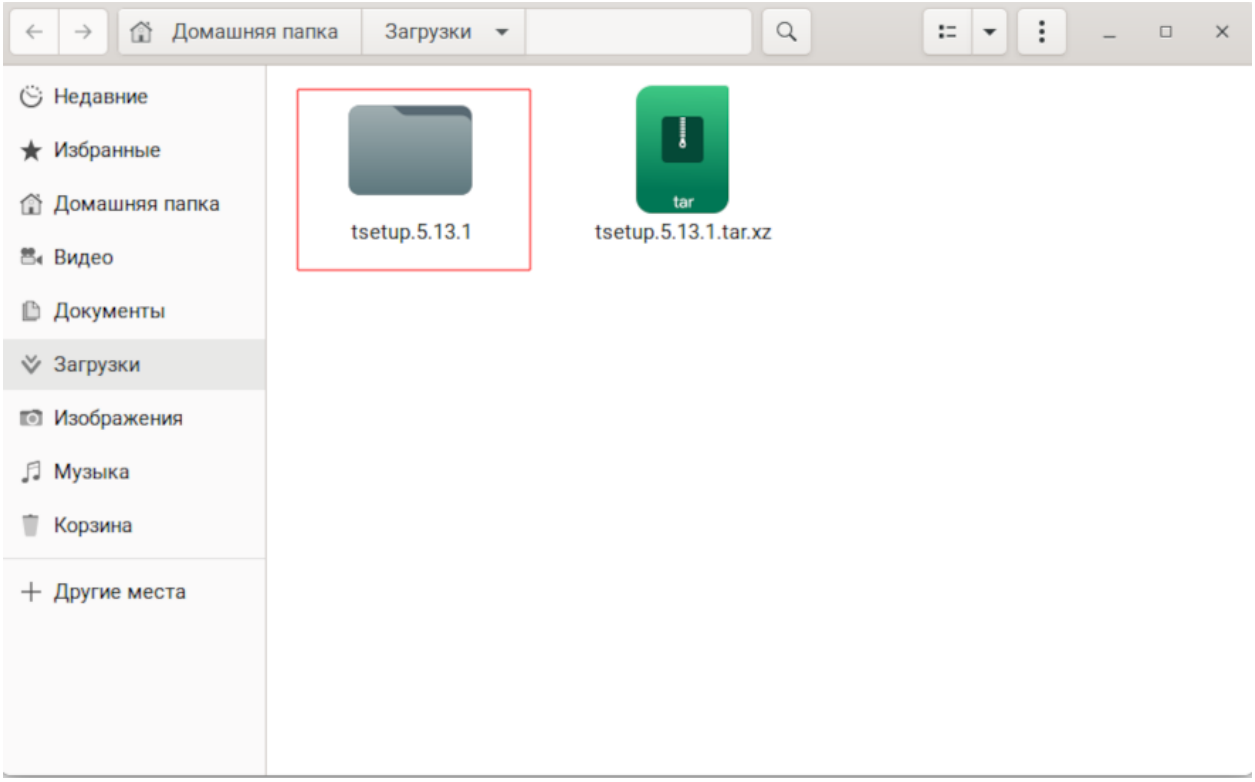
1. Авторизуйтесь в системе.
2. Откройте в браузере страницу скачивания настольного приложения.
3. Нажмите кнопку «**Telegram для Linux x64**».

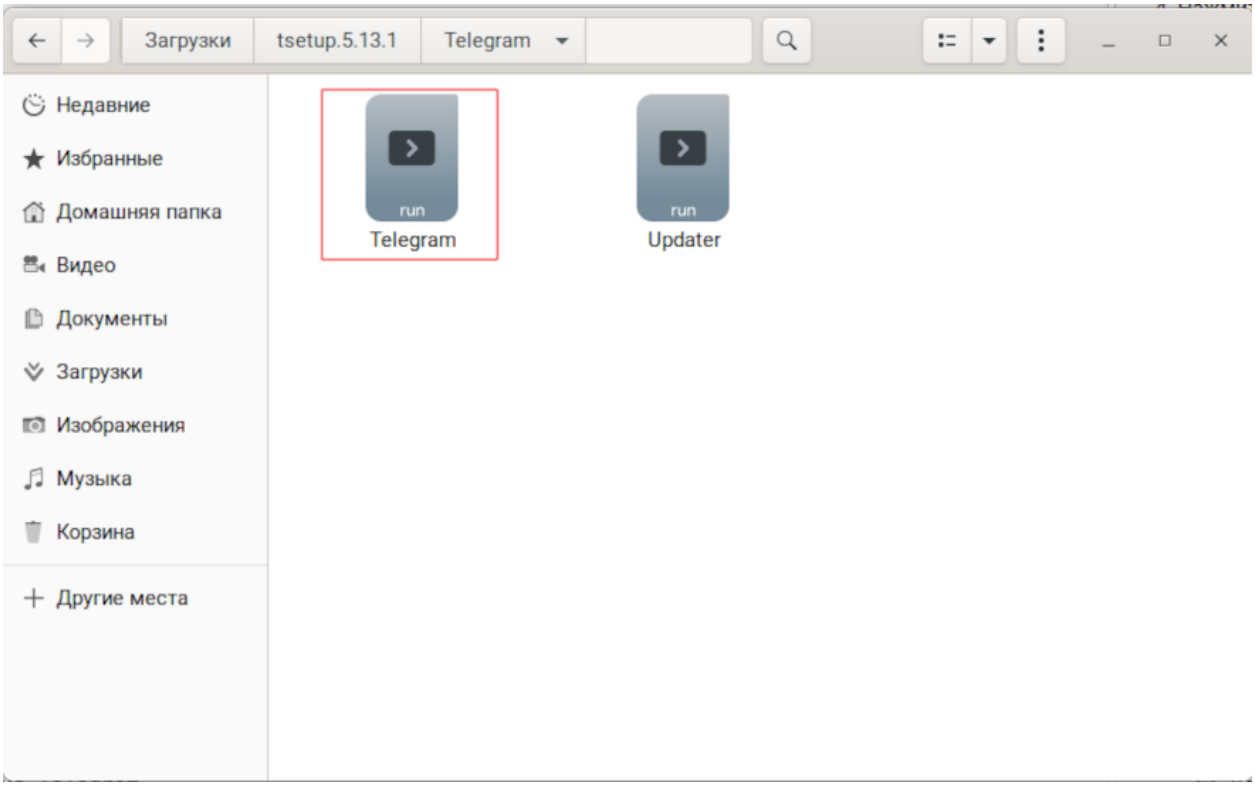


4. Нажмите кнопку «**Скачанные файлы**»» в правом верхнем углу окна браузера и кликните на скачанный файл `tsetup.X.XX.X.tar.xz`.



5. В открывшемся окне дождитесь, когда появится папка с именем вида `tsetup.X.XX.X`, откройте папку и запустите приложение Telegram.





Как установить и настроить серверную часть TrueConf на МСВСфера 9 Сервер

Инструкция по установке и настройке серверной части TrueConf на МСВСфера 9 Сервер.

Окружение

- МСВ Сфера Сервер версии 9.X

Установка и настройка

1. Обновите систему и перезагрузите сервер:

```
$ sudo dnf update -y && reboot
```

2. Отключите `selinux` командой:

```
$ sed -i 's/^SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
```

3. Создайте пользователя для администрирования TrueConf:

```
$ sudo adduser tc-admin
```

4. Установите пароль для созданного пользователя:

```
$ sudo passwd tc-admin
```

5. Подключите репозиторий `epel`:

```
$ sudo dnf install epel-release -y
```

6. Подключите репозиторий `trueconf` путём создания файла `/etc/yum.repos.d/trueconf.repo`:

```
[trueconf]
name=trueconf
baseurl=https://mirror.trueconf.ru/centos-stream/$releasever/$basearch/release
gpgcheck=1
enabled=1
```

7. Установите GPG-ключ репозитория:

```
$ rpm --import https://mirror.trueconf.ru/rpm-v3.trueconf.pub
```

8. Установите пакет `trueconf-server`:

```
$ sudo dnf install trueconf-server
```

Предупреждение

На сервере не должно быть других панелей управления, поскольку это может привести к некорректной работе приложения или к её полной неработоспособности.

> **Примечание**
>
> В процессе установки будет запрошено имя (имена) администраторов. Необходимо указать пользователя, созданного ранее.

9. Предоставьте расширенные права доступа к файлу `/etc/shadow` группе администраторов `trueconf`:

```
$ setfacl -m u:trueconf:r /etc/shadow
```

После установки можно войти в панель администратора в браузере по адресу `http://IPADDRESS`, используя данные для авторизации из п. 3 и 4.

Здесь `IPADDRESS` — IP-адрес сервера, на котором установлена серверная часть TrueConf.

Инструкции для клиентской части TrueConf и инструкцию по администрированию можно найти в официальной документации TrueConf.

Как установить и настроить K3s Rancher (kubernetes) на МСВСфера 9 Сервер

Ниже представлена инструкция для установки K3s Rancher (kubernetes) на МСВСфера 9 Сервер. В приведённом примере система состоит из главного узла (master-нода) и рабочего узла (worker-нода).

Окружение

- МСВСфера 9 Сервер

Установка и настройка

1. Обновите систему и перезагрузите сервер:

```
$ sudo dnf update -y && reboot
```

2. Загрузите необходимые модули ядра:

```
$ sudo modprobe overlay
$ sudo modprobe br_netfilter
```

   Чтобы модули загружались автоматически после перезагрузки сервера необходимо создать конфигурационный файл:

```
$ sudo cat <<EOF | sudo tee /etc/modules-load.d/containerd.conf
overlay
br_netfilter
EOF
```

3. Включите дополнительные настройки системы:

```
$ sudo cat <<EOF | sudo tee /etc/sysctl.d/99-kubernetes-cri.conf
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
net.bridge.bridge-nf-call-ip6tables = 1
EOF
```

```
$ sudo sysctl --system
```

4. Установите docker следующими командами:

```
$ sudo yum install -y yum-utils
$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.
↪repo
$ sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
↪compose-plugin
$ sudo systemctl enable docker --now
$ sudo systemctl enable containerd --now
```

5. Откройте файл `/etc/containerd/config.toml` для редактирования, найдите секцию `[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]` и под ней поменяйте значение «SystemdCgroup» на `true`. В результате содержимое файла должно выглядеть следующим образом:

```
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
    BinaryName = ""
    CriuImagePath = ""
    CriuPath = ""
    CriuWorkPath = ""
    IoGid = 0
    IoUid = 0
    NoNewKeyring = false
    NoPivotRoot = false
    Root = ""
    ShimCgroup = ""
    SystemdCgroup = true
```

6. Перезапустите `containerd`:

```
$ sudo systemctl restart containerd
```

7. Отключите SELinux и swap:

```
$ sudo setenforce 0
$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
$ sudo swapoff -a
$ sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

8. Добавьте файл `/etc/yum.repos.d/kubernetes.repo` со следующим содержимым:

```
[kubernetes]
name=Kubernetes
```

(продолжение на следующей странице)

(продолжение с предыдущей страницы)

```
baseurl=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.30/rpm/repodata/repomd.xml.key
```

9. Установите `kubectl`, `kubeadm` и `kubelet`:

```
$ sudo dnf install kubectl kubeadm kubelet
```

10. Установите `helm`:

```
$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/
↪get-helm-3
$ chmod 0700 get_helm.sh
$ ./get_helm.sh
```

11. Установите имя хоста на обоих узлах:

    Master:

```
$ sudo hostnamectl set-hostname k3s-master && exec bash
```

    Worker:

```
$ sudo hostnamectl set-hostname k3s-worker && exec bash
```

12. Добавьте необходимые правила в файервол на обоих узлах:

    Master:

```
$ sudo firewall-cmd --permanent --add-port=6443/tcp
$ sudo firewall-cmd --permanent --add-port=8472/udp
$ sudo firewall-cmd --permanent --add-port=10250/tcp
$ sudo firewall-cmd --permanent --add-port=51820/udp
$ sudo firewall-cmd --permanent --add-port=51821/udp
$ sudo firewall-cmd --permanent --zone=trusted --add-source=10.42.0.0/16
$ sudo firewall-cmd --permanent --zone=trusted --add-source=10.43.0.0/16
$ sudo firewall-cmd —reload
```

    Worker:

```
$ sudo firewall-cmd --permanent --add-port=8472/udp
$ sudo firewall-cmd --permanent --add-port=10250/tcp
$ sudo firewall-cmd --permanent --add-port=51820/udp
$ sudo firewall-cmd --permanent --add-port=51821/udp
$ sudo firewall-cmd --permanent --zone=trusted --add-source=10.42.0.0/16
$ sudo firewall-cmd --permanent --zone=trusted --add-source=10.43.0.0/16
$ sudo firewall-cmd --reload
```

13. Установите `wget` на главном узле (master) следующей командой:

```
$ sudo dnf install wget -y
```

14. Установите `k3s` на главном узле (master) следующей командой:

```
$ sudo curl -sfL https://get.k3s.io | sh -
```

15. Настройте `k3s` на главном узле (master):

```
$ mkdir ~/.kube
$ sudo cp /etc/rancher/k3s/k3s.yaml ~/.kube/config
$ sudo chown $USER ~/.kube/config
$ sudo chmod 600 ~/.kube/config
$ export KUBECONFIG=~/.kube/config
```

16. Выполните следующую команду на главном узле, чтобы вывести на экран токен доступа:

```
$ sudo cat /var/lib/rancher/k3s/server/node-token
```

17. На рабочем узле (worker) добавьте следующую строку в файл `/etc/hosts`:

```
MASTER_IP k3s-master
```

    где `MASTER_IP` — IP-адрес главного узла (master-ноды).

18. На рабочем узле (worker) выполните следующую команду:

```
$ curl -sfL https://get.k3s.io | K3S_URL=https://k3s-master:6443 K3S_TOKEN=<token>␣
↪sh -
```

    где `<token>` — токен доступа из п. 16.

На данном этапе кластер K3s готов к работе. Дополнительно можно установить панель управления кластером Rancher. Для этого выполните следующие команды:

```
$ sudo kubectl create namespace cattle-system
$ sudo helm repo add rancher-latest https://releases.rancher.com/server-charts/latest
$ sudo kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.15.3/
↪cert-manager.crds.yaml
$ sudo helm repo add jetstack https://charts.jetstack.io
$ sudo helm repo update
$ sudo helm install cert-manager jetstack/cert-manager   --namespace cert-manager   --create-
↪namespace
$ sudo helm install rancher rancher-latest/rancher   --namespace cattle-system   --set␣
↪hostname=MASTER_IP.sslip.io  --set replicas=1   --set bootstrapPassword=admin
```

где MASTER_IP — IP-адрес главного узла (master).

Более подробную инструкцию и детали установки вы можете найти в официальной документации Rancher.

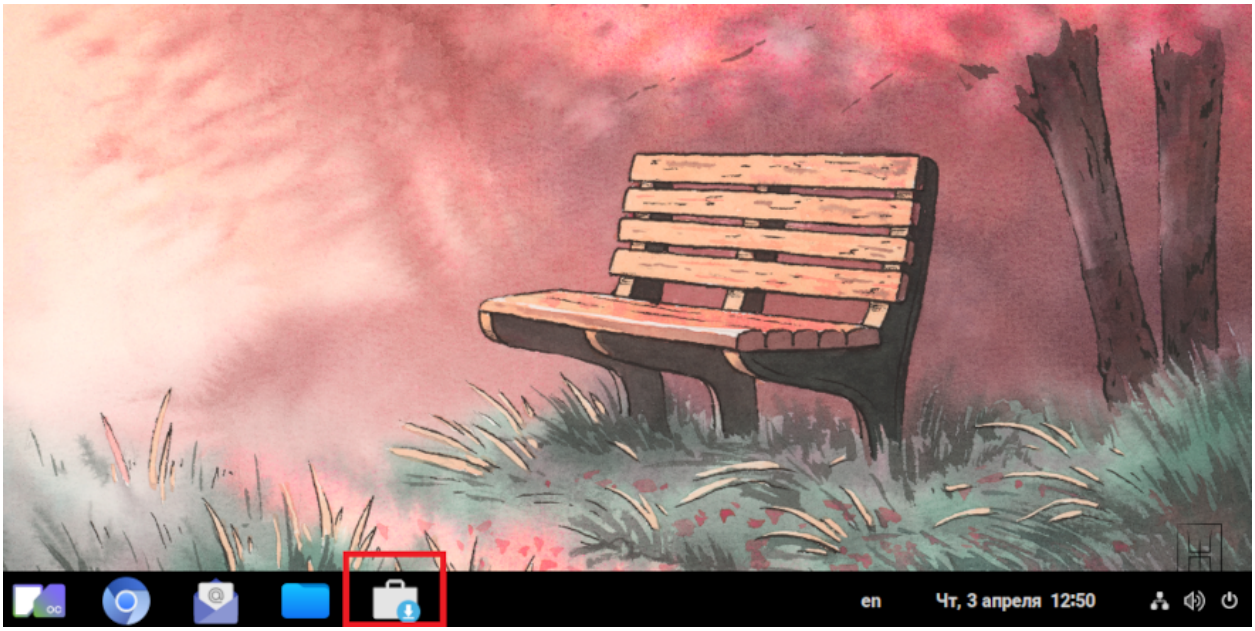
Как установить Яндекс.Браузер в МСВСфера ОС

Окружение

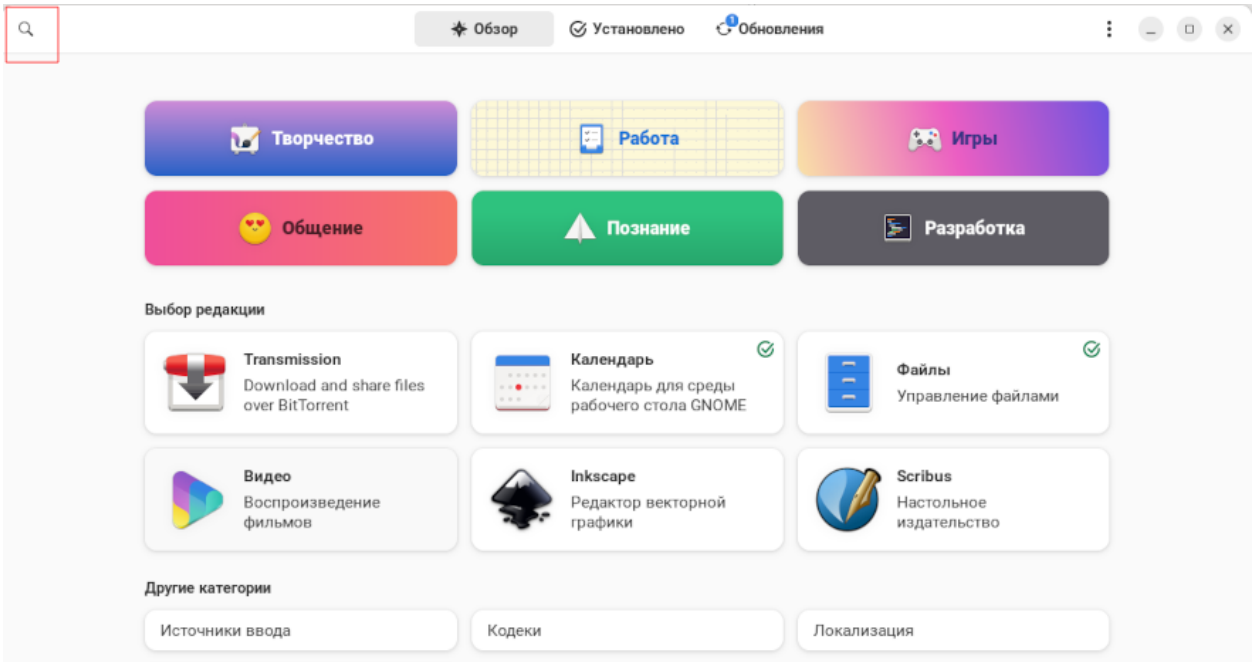
- МСВСфера Сервер версии 8.X/9.X
- МСВСфера АРМ версии 8.X/9.X

Установка через «Центр Приложений»

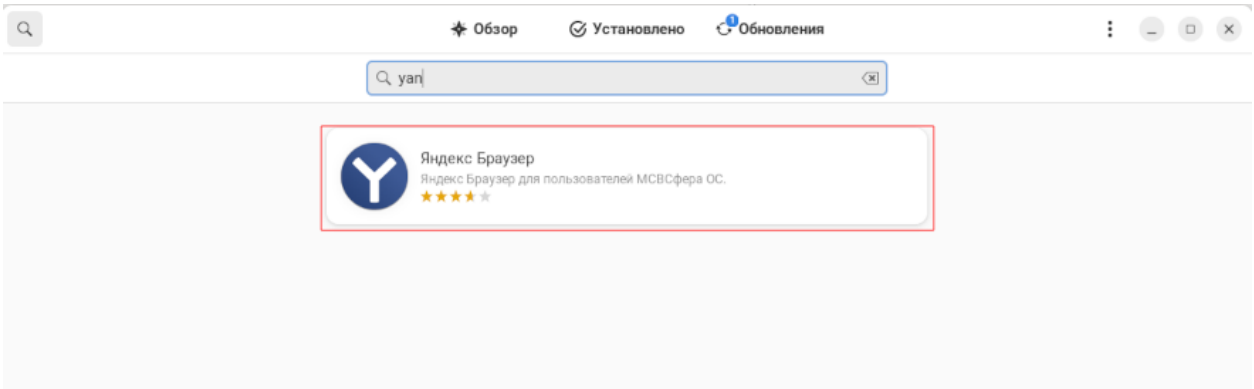
1. Откройте «Центр Приложений».



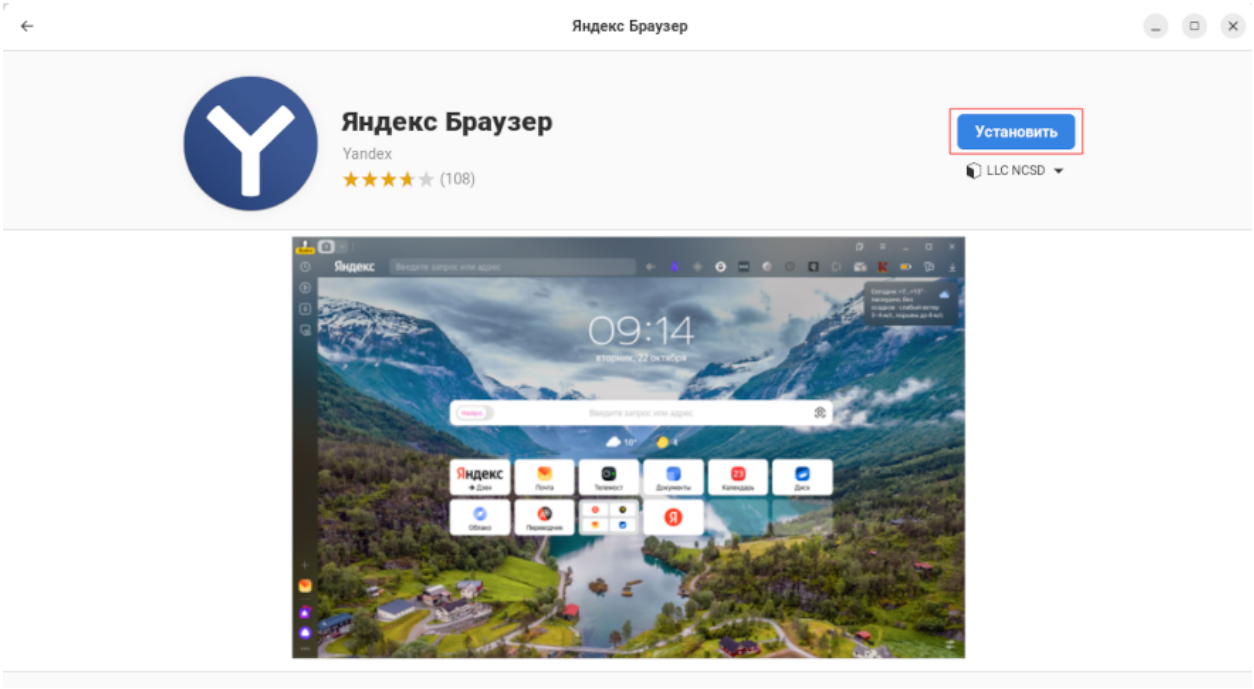
2. Нажмите кнопку поиска в левом верхнем углу.



3. Наберите «yandex» в поле поиска и нажмите на «Яндекс Браузер» в результатах поиска.

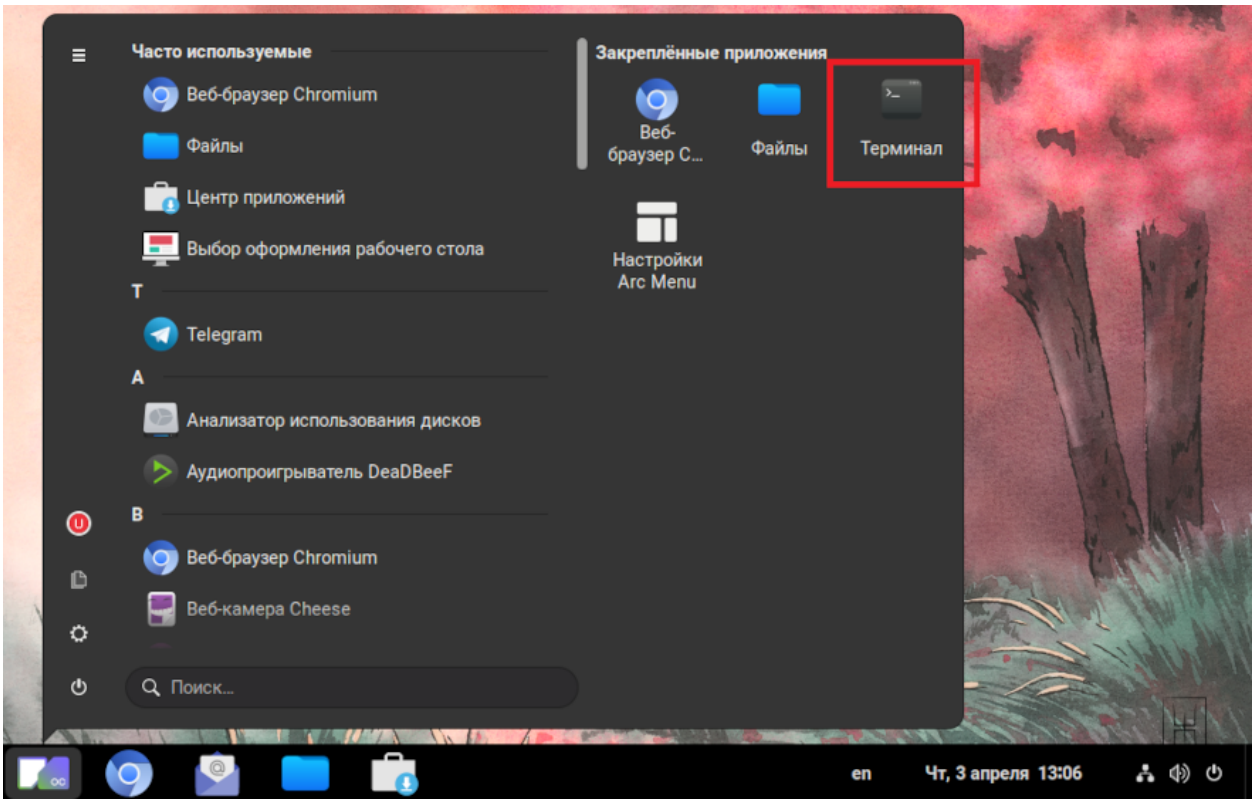


4. В окне приложения нажмите кнопку «Установить».



Установка через «Терминал»

1. Откройте «Терминал».



2. Выполните следующую команду:

```
$ sudo dnf install yandex-browser-corporate
```

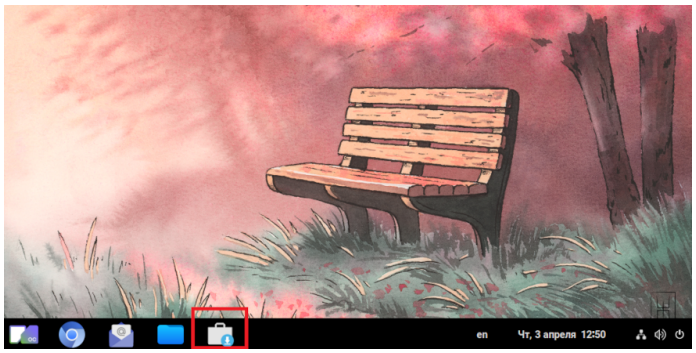
Как установить и настроить почтовый клиент Thunderbird

Окружение

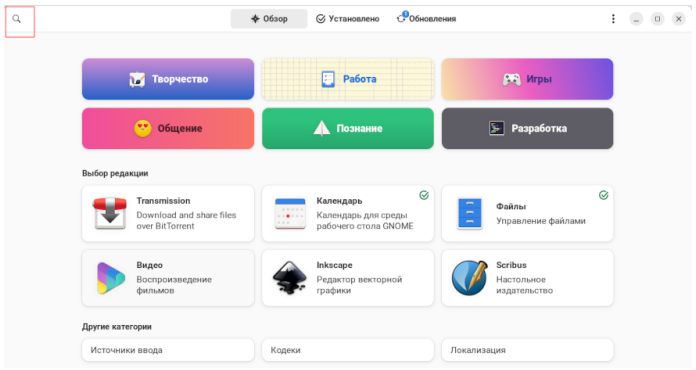
- МСВ Сфера Сервер 8.X/9.X
- МСВ Сфера АРМ 8.X/9.X

Установка через «Центр приложений»

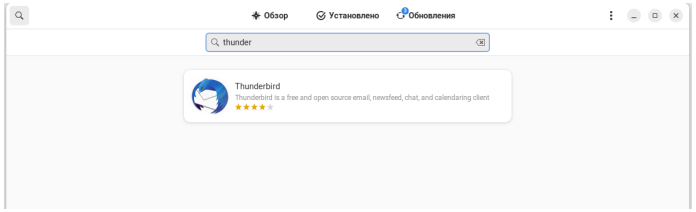
1. Откройте «Центр приложений».



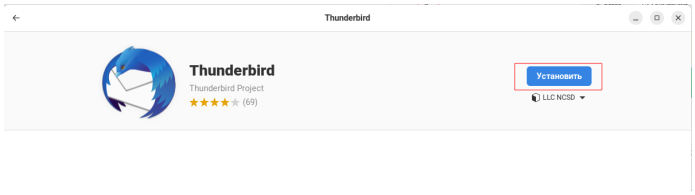
2. Нажмите кнопку поиска в левом верхнем углу.



3. Наберите «thunder» в поле поиска и нажмите на «Thunderbird» в результатах поиска.

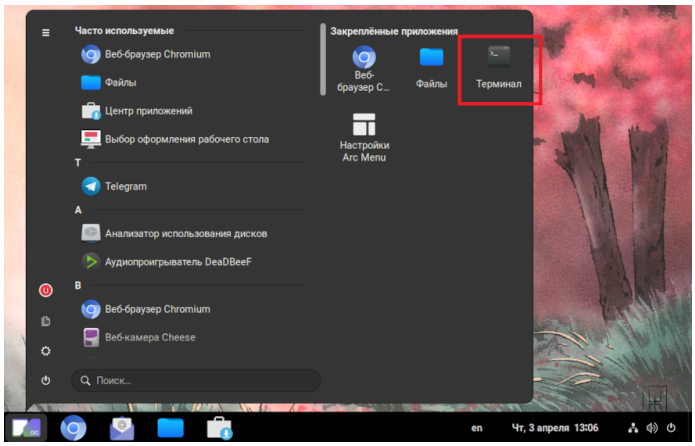


4. Нажмите кнопку «Установить».



Установка через «Терминал»

1. Откройте «Терминал».

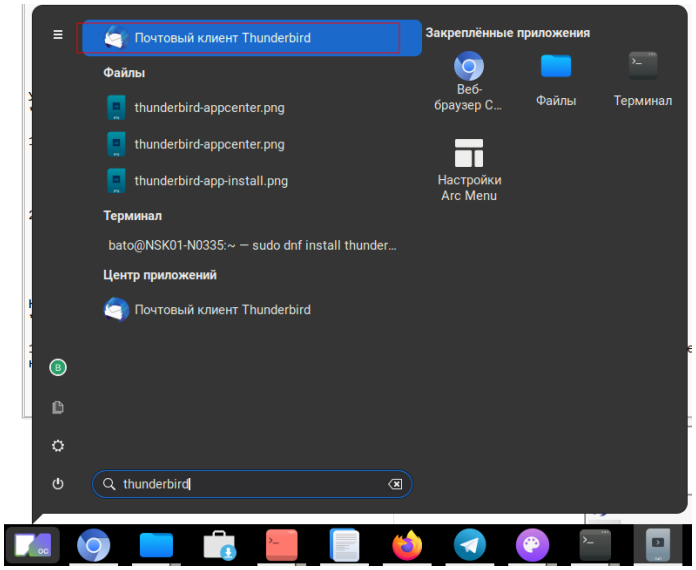


2. Выполните следующую команду:

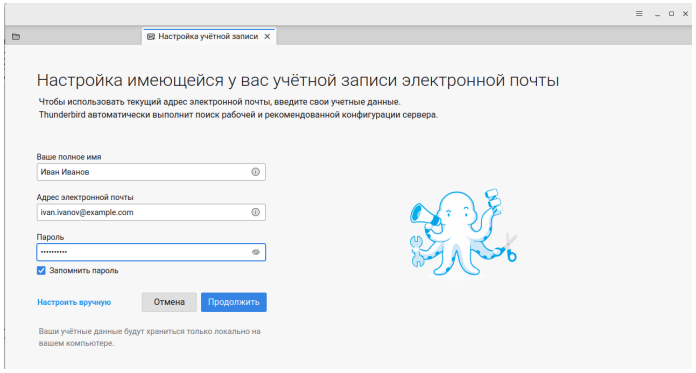
```
$ sudo dnf install thnuderbird
```

Настройка Thunderbird

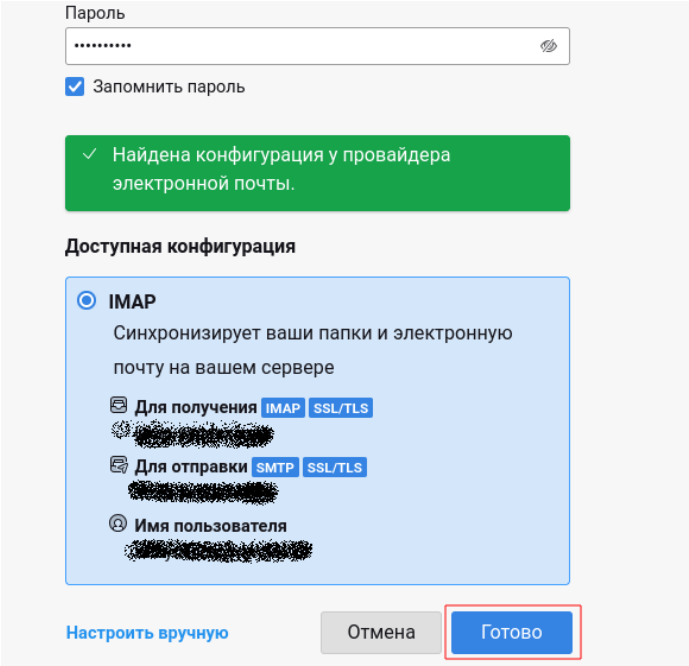
1. Откройте почтовый клиент «Thunderbird». Для этого откройте главное меню и в строке поиска наберите «thunderbird».



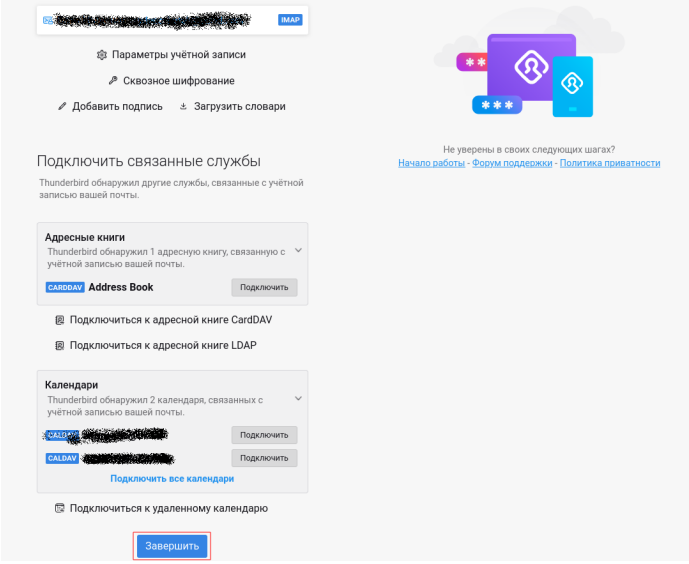
2. В открывшемся окне введите данные для входа в почту и нажмите «Продолжить».



3. Проверьте правильность данных, полученных от провайдера и нажмите «Готово».



4. Подключите дополнительные службы (при необходимости) и нажмите кнопку «Завершить».



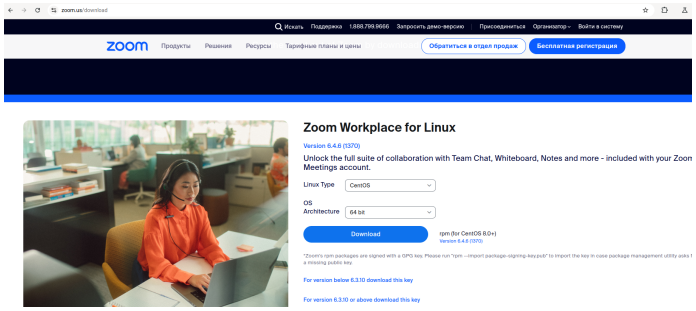
Как установить Zoom

Окружение

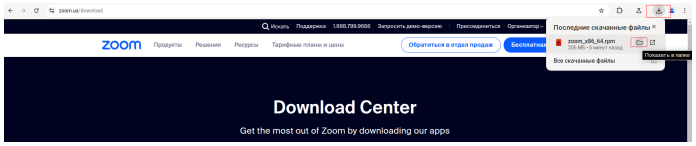
- МСВ Сфера Сервер 8.X/9.X
- МСВ Сфера АРМ 8.X/9.X

Установка

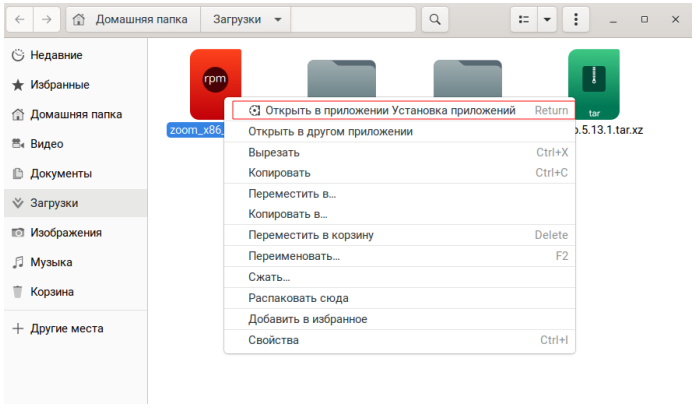
1. Перейдите на официальный сайт Zoom https://zoom.us/download.
2. В выпадающем списке «Linux Type» выберите «CentOS», после этого появится ещё один выпадающий список «OS Architecture». В нём выберите «64 bit».
3. Нажмите «Download». Начнётся загрузка rpm-файла.



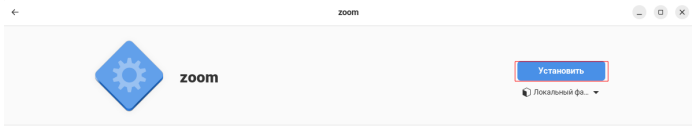
4. Откройте «Загрузки» в правом верхнем углу браузера, найдите «Zoom» и нажмите «Показать в папке».



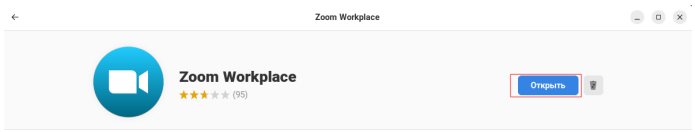
5. Нажмите на файл `zoom_x86_64.rpm` правой кнопкой мыши и выберите «Открыть в приложении Установка приложений».



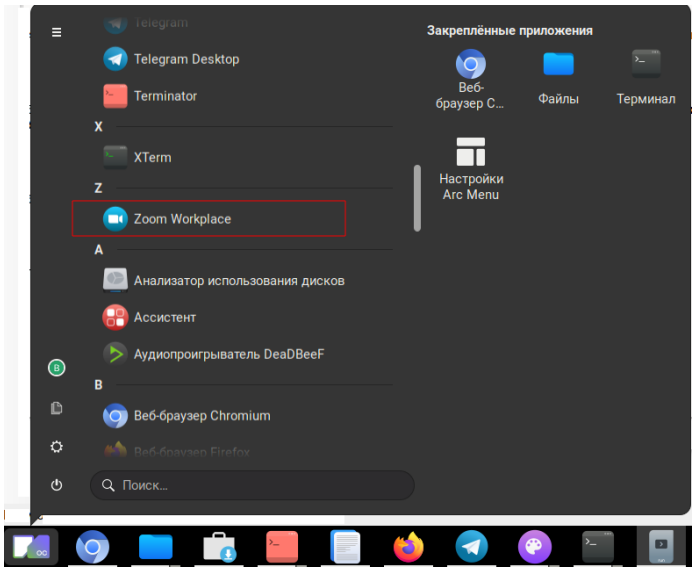
6. В окне «Центра приложений» нажмите «Установить».



После этого Zoom можно открыть из «Центра приложений».



Или через главное меню ОС.



Как установить Steam из репозитория Third-Party

Окружение

- МСВСфера ОС 9.

Подключение репозитория

В МСВСфера ОС 9 вы можете установить игровую платформу «Steam». Установка осуществляется из репозитория `Third-Party`.

Для того, чтобы подключить репозиторий `Third-Party`, выполните следующую команду:

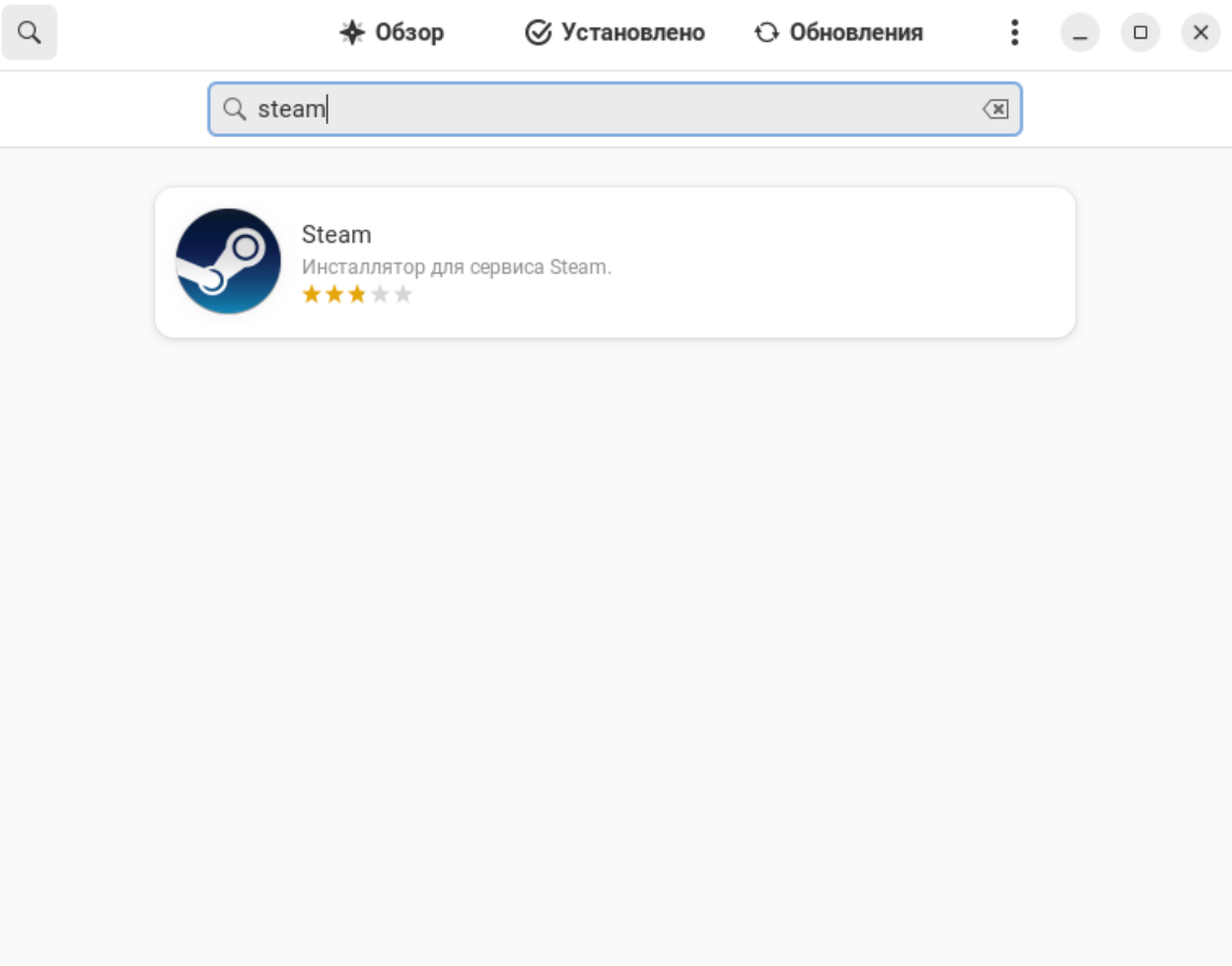
```
$ sudo dnf install sphere-repos-thirdparty
```

Перезагрузка не требуется.

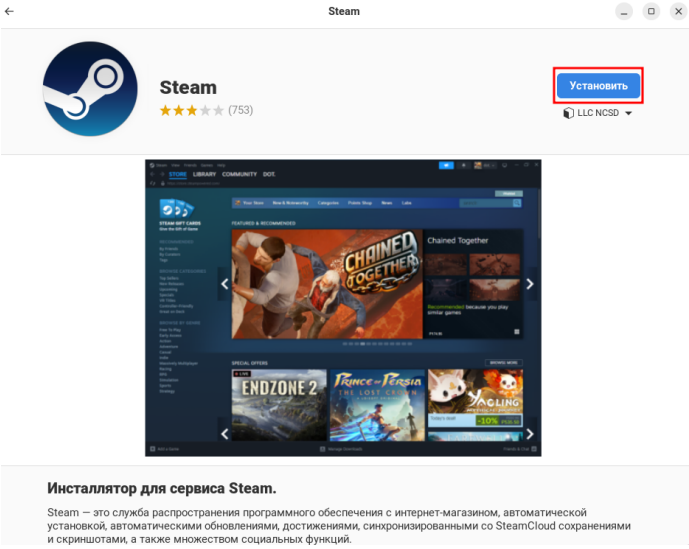
Установка «Steam»

Теперь вы можете установить приложение «Steam», для этого выполните следующие действия.

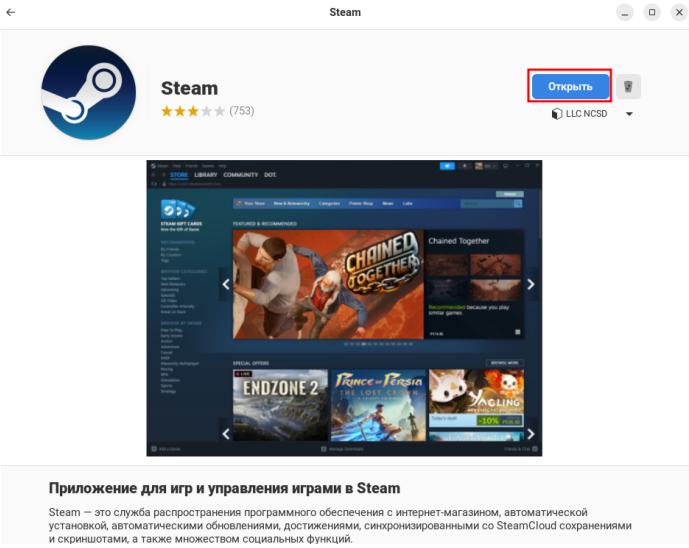
1. Перейдите в «Центр приложений» любым удобным способом.
2. В строке поиска введите «Steam».



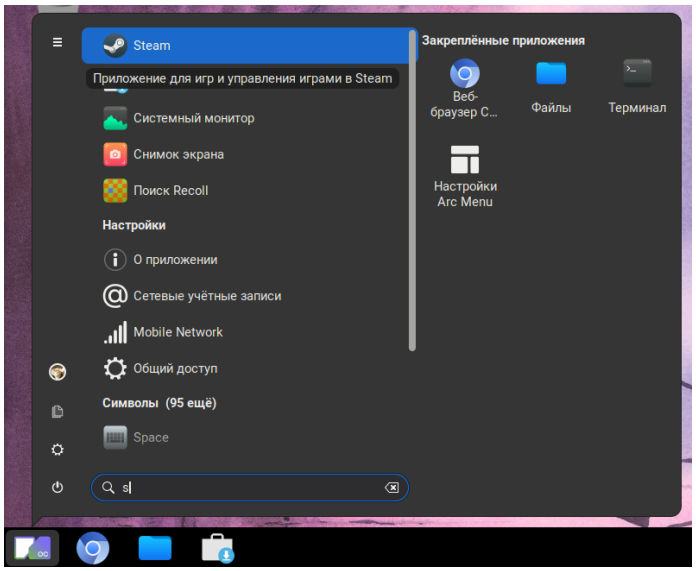
3. Нажмите на кнопку «Установить» в правом верхнем углу экрана. Сразу же начнётся установка.



Вы можете открыть приложение «Steam» сразу после установки из «Центра приложений».



Или же открыть его в любое время из главного меню.



Как установить VK Play через PortProton

Окружение

- МСВСфера ОС 9.

Введение

VK Play — российская площадка для любителей игр, разработчиков и авторов контента. Объединяет стриминг, киберспорт, место для общения с единомышленниками, каталог игр и облачный гейминг. Зарегистрироваться и получить доступ к площадке можно на официальном сайте.

PortProton — это проект, разработанный командой linux-gaming, позволяющий запускать игры на Linux-платформах в режиме эмуляции Windows.

В состав МСВСфера ОС 9 входит пакет `portproton`, который позволит вам подключить PortProton и установить VK Play на ваше устройство.

Подключение PortProton

1. Установите PortProton из репозитория `extras` (репозиторий подключён по умолчанию). Для этого выполните следующую команду:

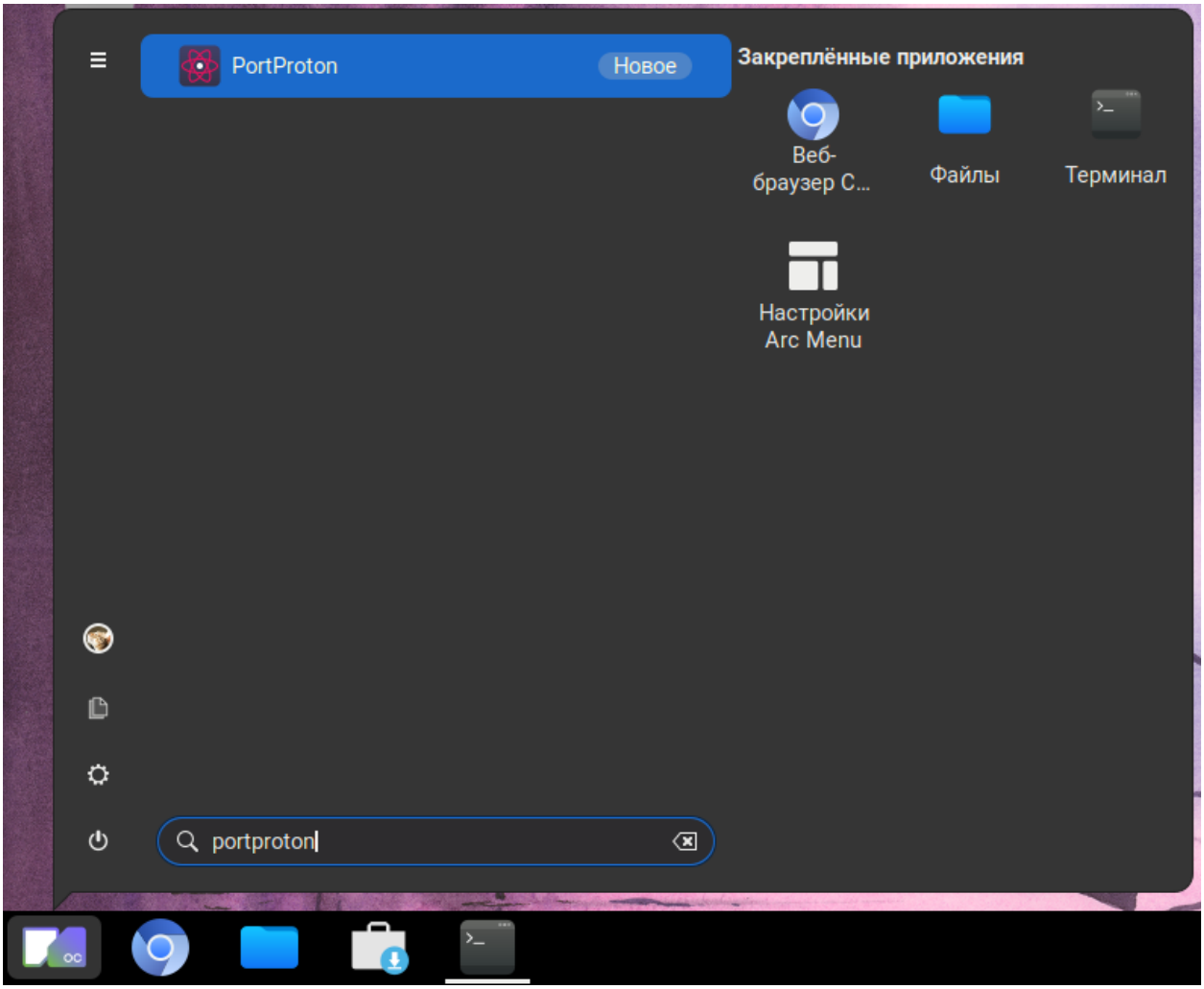
```
$ sudo dnf install portproton
```

Перезагрузка как правило не требуется.

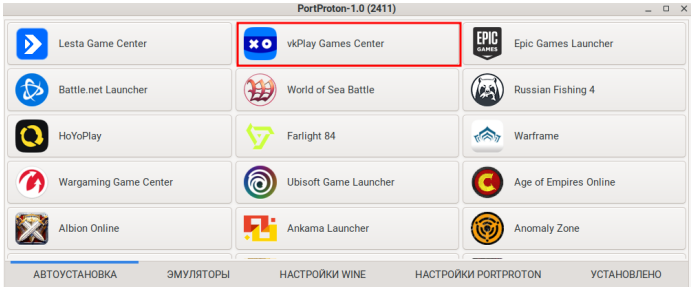
Установка VK Play

Теперь вы можете установить приложение VK Play, для этого выполните следующие действия.

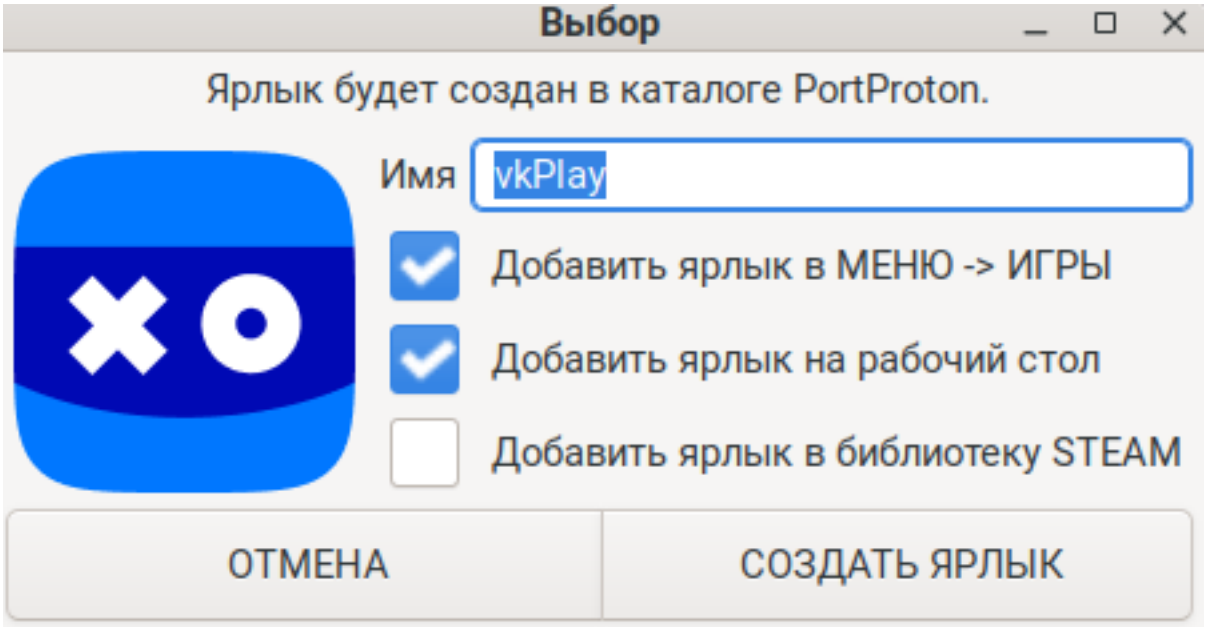
1. Запустите PortProton. Для этого перейдите в «Центр приложений» любым удобным способом. И в строке поиска введите «PortProton». При первом запуске будет выполнена установка и обновление приложения.



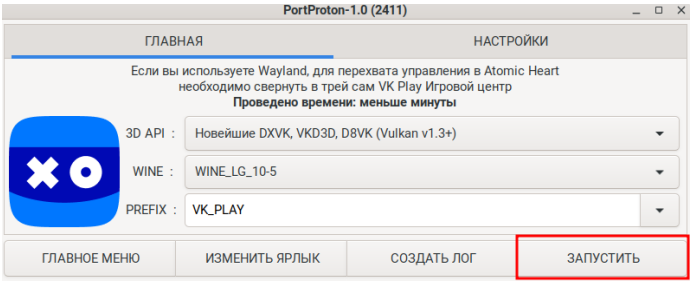
2. По умолчанию откроется вкладка «Автоустановка». Нажмите на кнопку «**vkPlay Games Center**». Начнётся загрузка приложения и установка обновлений.



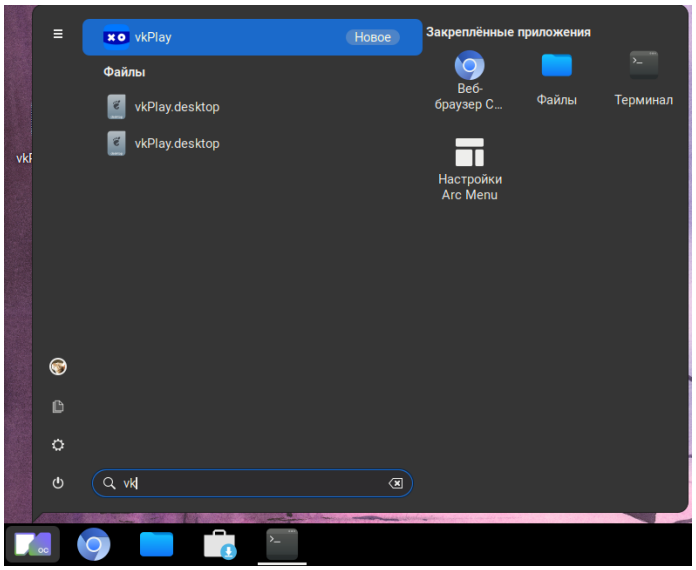
Выберите какие ярлыки создавать и имя приложения (или оставьте значения по умолчанию).



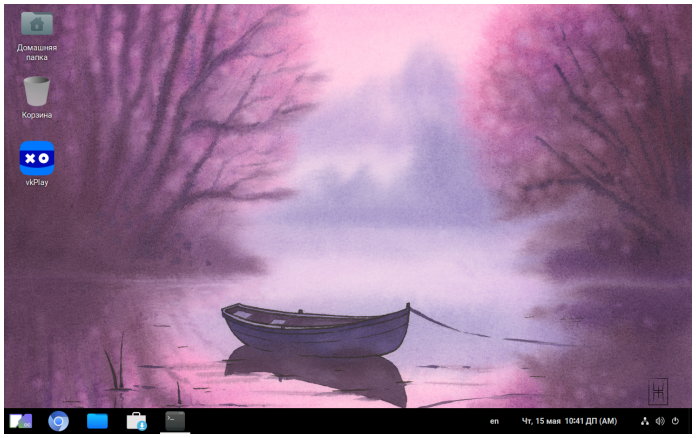
Вы можете запустить VK Play сразу же, нажав на кнопку «**ЗАПУСТИТЬ**».



А также в любое время из главного меню.



Или нажав на значок на рабочем столе.



Работа с Wine в ОС МСВ Сфера

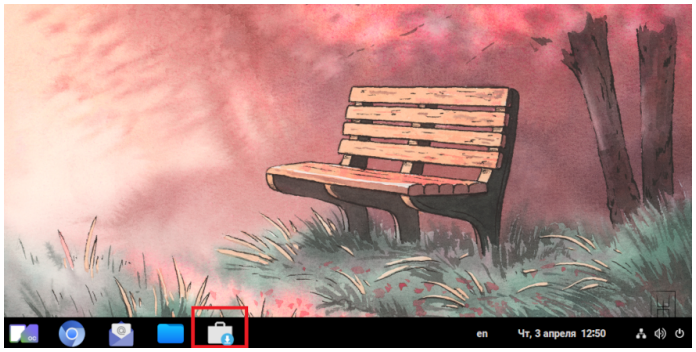
Wine — это свободное программное обеспечение для запуска Windows-приложений на нескольких POSIX-совместимых операционных системах, включая Linux, macOS и BSD. Название W.I.N.E. — рекурсивный акроним и расшифровывается «Wine Is Not Emulator» — «W.I.N.E. — это не эмулятор». Имеется в виду, что Wine — это не эмулятор компьютера или виртуальная машина, как, например, qemu, VirtualBox и другие подобные им. Вместо этого Wine — это альтернативная, свободная реализация Windows API (WIndows Native Environment).

Окружение

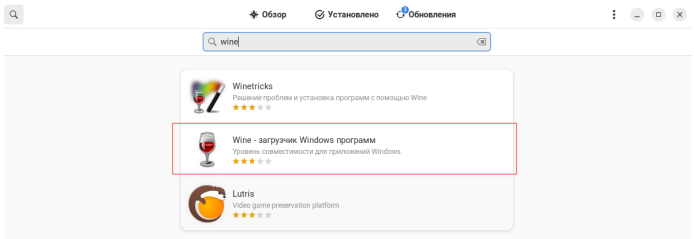
- МСВСфера 9 Сервер.
- МСВСфера 9 АРМ.

Установка Wine

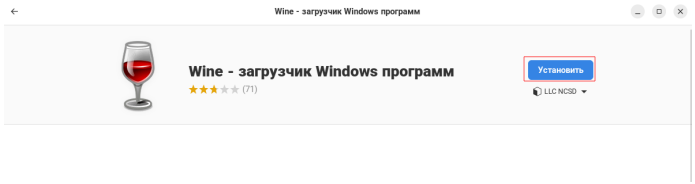
1. Откройте «Центр Приложений».



2. Наберите в поиске «wine» и нажмите на «Wine - загрузчик Windows программ».

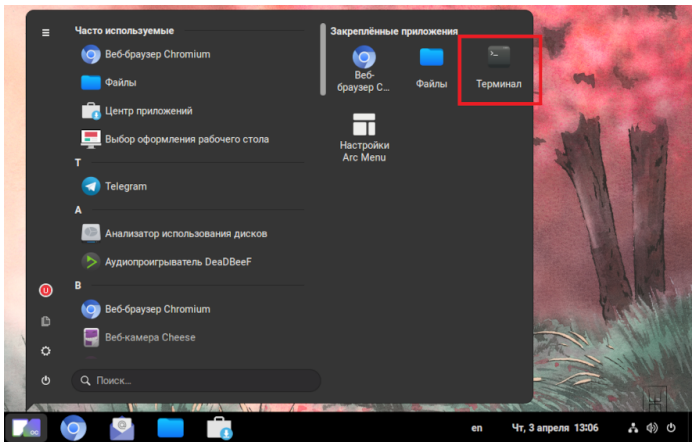


3. Нажмите «Установить».



Также Wine можно установить через «Терминал». Для этого выполните следующие действия.

1. Откройте «Терминал».



2. Выполните команду:

```
$ sudo dnf install wine
```

3. После установки запустите Wine с помощью команды:

```
$ winecfg
```

> **Примечание**
>
> При первом запуске команды `winecfg` создается префикс `.wine` — 64-разрядный по умолчанию.

Установка программ с помощью Wine

Для установки легковесных программ достаточно выполнить команду:

```
$ wine filename.exe
```

Однако в большинстве случаев необходимо настроить префикс `wine`. Например:

```
$ WINEPREFIX="/home/$USER/.wine_prefix" wine "/home/$USER/Downloads/filename.exe"
```

Если префикс не указан, то используется префикс, настроенный по умолчанию (x64).

Для установки из файла с расширением `.msi`, используется команда:

```
$ wine msiexec /i filename.msi
```

Запуск программ

Для запуска программы необходимо указать полный путь к исполняемому файлу:

```
$ wine /путь-к-файлу/program.exe
```

Примечание

Первый запуск программы может занять некоторое время.

Запуск файла `.bat` производится командой:

```
$ wine start filename.bat
```

Удаление программ

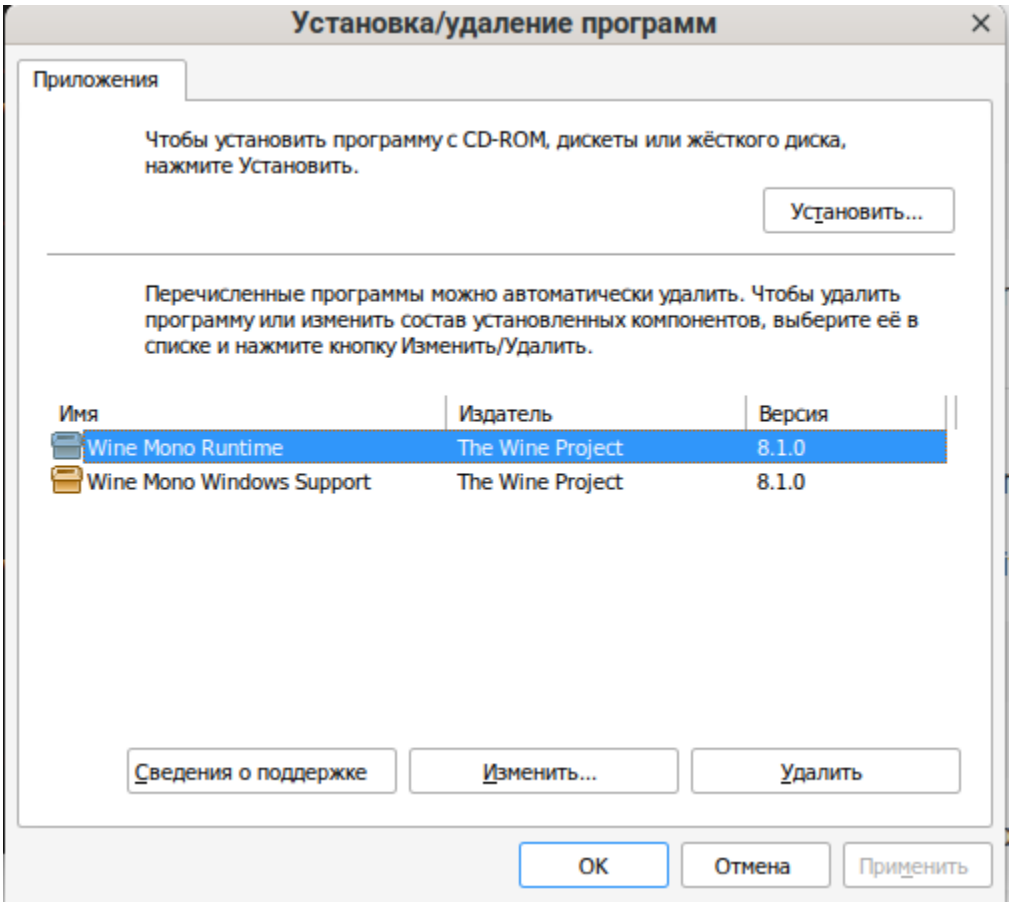
Удалить программу можно через графическое приложение «uninstaller». Для запуска приложения выполните команду:

```
$ wine uninstaller
```

Также можно удалить программу из определённого префикса:

```
$ WINEPREFIX="/home/$USER/.wine" wine uninstaller
```

При помощи «Wine uninstaller» можно устанавливать программы, нажав на кнопку «Установить» и выбрав необходимую для установки программу.



Для удаления программ без использования графического интерфейса используйте команду:

```
$ wine uninstaller --remove <название_программы>
```

Или из префикса:

```
$ WINEPREFIX="/home/$USER/.wine_prefix" wine uninstaller --remove <название_программы>
```

Также есть возможность удалить программу, используя её собственную программу удаления:

```
$ wine /<путь_к_папке_с_программой>/uninstall.exe
```

Префиксы Wine

Для разных программ могут потребоваться разные дополнительные компоненты, разные настройки Wine и т. п. Для этого существуют префиксы Wine. Используя для каждой программы свой префикс, вы можете обеспечить им индивидуальное окружение.

Чтобы создать 64-разрядный префикс с именем `.wine_prefix`, надо выполнить следующую команду:

```
$ WINEARCH=win64 WINEPREFIX="/home/$USER/.wine_prefix" winecfg
```

где `WINEARCH=win64` указывает, что префикс создается под архитектуру `amd64`, что позволит запускать в нём 64-разрядные программы, а также 32-разрядные. Для создания 32-разрядного префикса необходимо заменить `64` на `32`.

`WINEPREFIX="/home/$USER/.wine_prefix"` указывает путь расположения создаваемого префикса.

Примечание

Кавычки используются в случаях, если в имени папок содержится пробел, например `"/home/user/Рабочий стол"`.

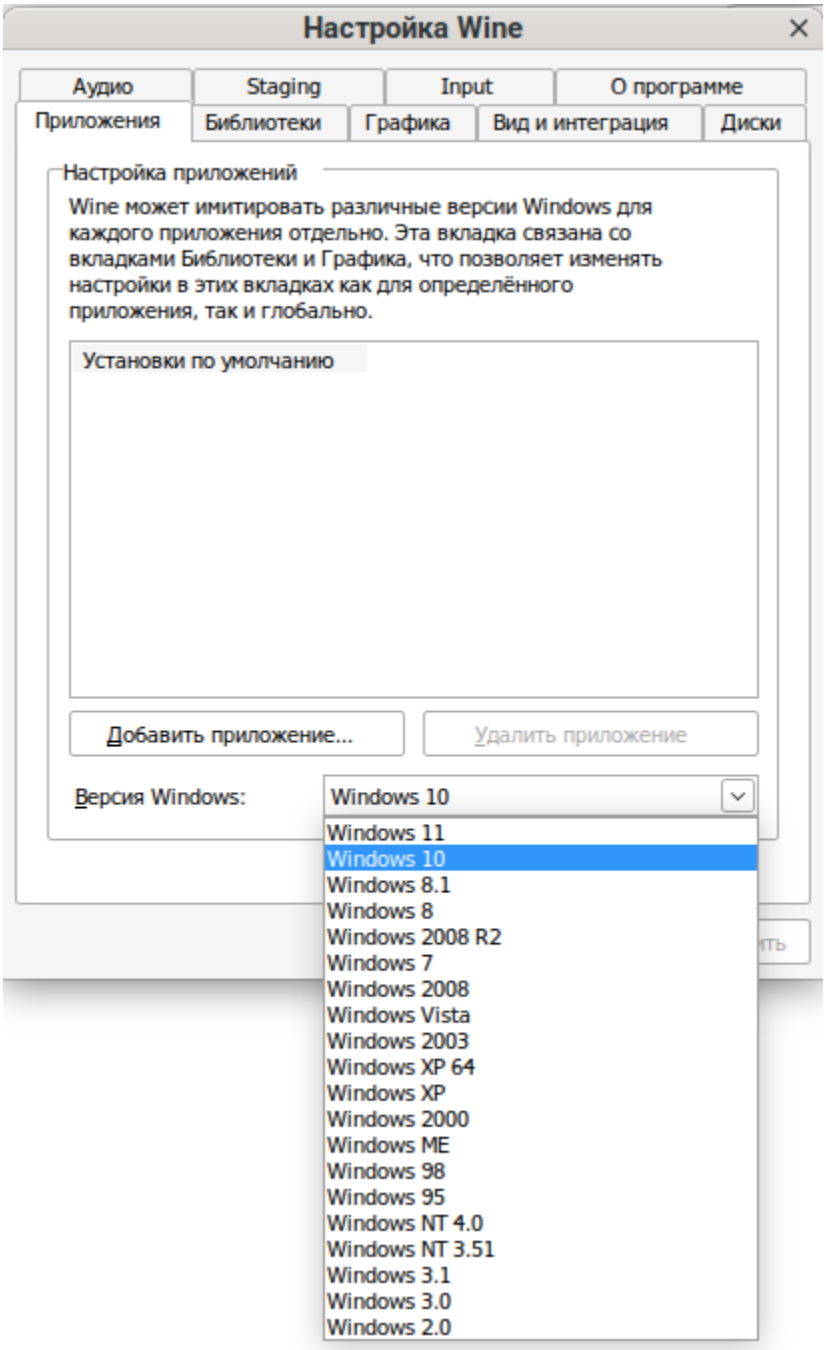
`winecfg` запускает программу конфигурирования префикса. Для запуска программ с префиксом каждый раз указывать архитектуру не требуется, поэтому после создания параметр `WINEARCH=win64` прописывать не нужно.

Пример запуска программы «Internet Explorer» с префиксом:

```
$ WINEPREFIX=/home/$USER/.wine_prefix wine /home/$USER/.wine/drive_c/Program\ Files/Internet\␣
↪Explorer/iexplore.exe
```

Настройка префикса Wine

Версия Windows позволяет выбирать режим запуска с необходимой версией Windows (по умолчанию — Windows 10).



Вкладка «Диски» позволяет управлять виртуальными и физическими разделами. В стандартной конфигурации подключено 2 раздела: `C` и `Z`.

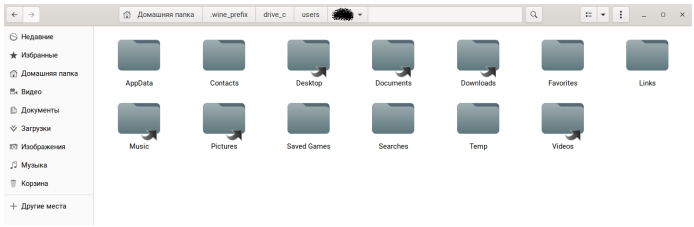
- `C` — является системным каталогом префикса;

- `Z` — ведёт в корень файловой системы Linux, в некоторых случаях такое поведение нежелательно, поэтому данный диск рекомендуется удалить, тем самым изолировав префикс от основной системы.

Примечание

Вредоносные программы для Windows потенциально способны нанести вред Linux-системе.

Вкладка «Вид и интеграция» даёт возможность выбора ассоциаций каталогов Windows-пользователя с домашними каталогами Linux-пользователя. Располагается в блоке «Папки» в нижней половине окна. В префиксе стандартные каталоги Windows-пользователя располагаются по пути `/home/user/.wine_prefix/drive_c/users/$USER/`.



Ссылки в данном случае указывают на домашние каталоги Linux-пользователя, который создал префикс.

Остальные вкладки подробно описаны в соответствующих руководствах и интуитивно понятны.

Файловый менеджер Winefile

Winefile — аналог «Проводника» в Windows.

Для запуска Winefile необходимо выполнить команду:

```
$ WINEPREFIX="/home/$USER/.wine_prefix" wine winefile
```

Работа с реестром

Для открытия реестра используется инструмент «regedit»:

```
$ wine regedit
```

Чтобы перемещаться по реестру, нажмите на ключи в левой части панели, чтобы перейти на следующий уровень глубины.

Чтобы удалить ключ, нажмите на него в меню «Edit» и выберите «Delete».

Чтобы добавить ключ или значение, определите место, где вы хотите поместить его, и выберите «New» из меню «Edit».

Чтобы изменить существующий ключ, выделите его в правой панели и выберите «Modify» из меню «Edit». Другой способ проделать то же самое, это щелчок правой кнопкой мыши на ключе или значении.

Файлы реестра располагаются в папке префикса, например:

```
/home/$USER/.wine_prefix/
```

- **system.reg** — содержит ветвь `HKEY_LOCAL_MACHINE`;
- **user.reg** — `HKEY_CURRENT_USER`;
- **userdef.reg** — `HKEY_HKEY_USERS\.Default` (содержит настройки Windows пользователя по умолчанию).

Дополнительные инструменты Wine

Wine — это свободное программное обеспечение для запуска Windows-приложений на нескольких POSIX-совместимых операционных системах, включая Linux, macOS и BSD. Название W.I.N.E. — рекурсивный акроним и расшифровывается «Wine Is Not Emulator» — «W.I.N.E. — это не эмулятор». Имеется в виду, что Wine — это не эмулятор компьютера или виртуальная машина, как, например, qemu, VirtualBox и другие подобные им. Вместо этого Wine — это альтернативная, свободная реализация Windows API (Windows Native Environment).

Окружение

- МСВСфера 9 Сервер.
- МСВСфера 9 АРМ.

Winetricks

Winetricks — вспомогательный скрипт для загрузки и установки различных распространяемых библиотек, необходимых для запуска некоторых программ в Wine, а также настройки окружения.

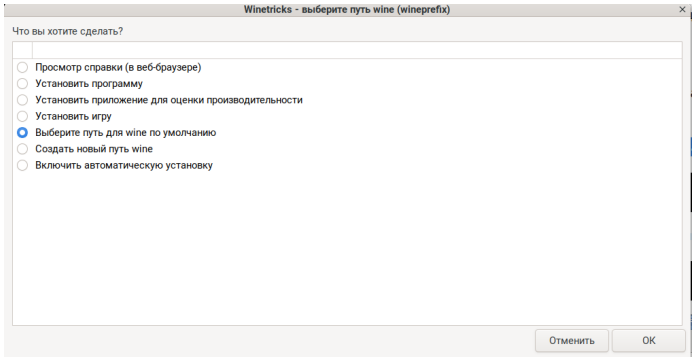
Для установки `winetricks` через «Терминал» выполните команду:

```
$ sudo dnf install winetricks
```

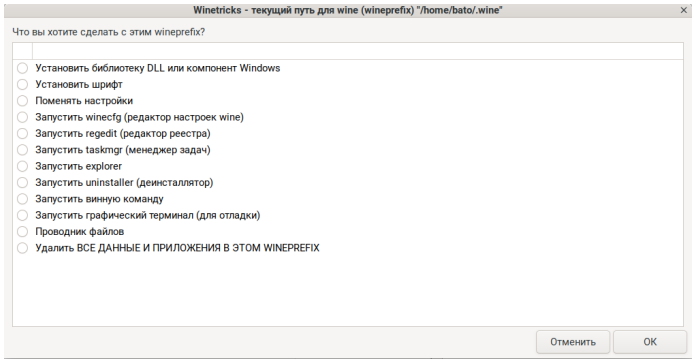
Чтобы открыть диалоговое окно программы для работы в графическом режиме, необходимо выполнить команду:

```
$ winetrick
```

Окно Winetricks выглядит так:



При выборе пункта меню, установленного по умолчанию, открывается следующее окно, которое содержит список доступных над wine-префиксом операций, где можно доустановить библиотеки, шрифты, вызвать редактор реестра или окно настроек wine (`winecfg`).



Чтобы установить пакеты через «Терминал» нужно выполнить следующую команду:

```
$ winetricks <название пакета>
```

Чтобы отобразить все установленные пакеты введите команду:

```
$ winetricks dlls list
```

Для вывода списка всех компонентов, которые уже установлены в текущем Wine-префиксе, выполните команду:

```
$ winetricks list-installed
```

Для вывода списка доступных категорий компонентов (например, приложения, шрифты, библиотеки DLL и т.п.) выполните команду:

```
$ winetricks list
```

Для вывода доступных компонентов в категории «Приложения»» выполните команду:

```
$ winetricks apps list
```

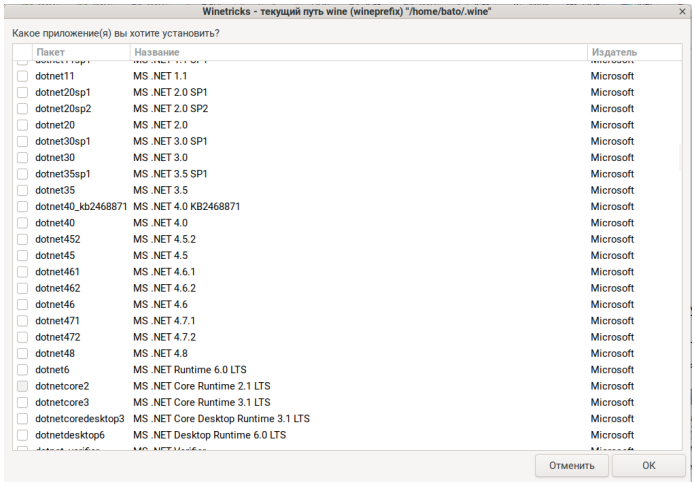
Для получения полного списка опций для команды `winetricks`, их описания и названий выполните:

```
$ winetricks --help
```

Пакет Dotnet

Для установки пакета в графическом режиме необходимо запустить `winetricks`.

При открытии окна выбрать пункт меню, установленный по умолчанию, затем необходимо выбрать пункт «**Установить библиотеку DLL или компонент Windows**». В списке приложений следует выбрать пакет для установки в зависимости от того, какая версия необходима, и нажать «**OK**».



Пакет Mono

Mono — программная платформа, позволяющая разработчикам достаточно просто создавать развитые кроссплатформенные приложения.

В базовом варианте Mono состоит из нескольких компонентов:

- **Компилятор языка C#**;

- **Среда исполнения Mono** — состоит из среды исполнения (CLI) непосредственно, компилятора среды исполнения (Just-In-Time, JIT), генератора машинного кода (Ahead-Of-Time, AOT), загрузчика сборок, сборщика мусора, подсистемы управления многопоточностью и компонентов поддержки взаимодействия между сборками и COM (или аналогами в Linux, например XCOM);
- **Базовая библиотека классов** — набор стандартных классов, совместимых с классами Microsoft .NET Framework;
- **Библиотека классов Mono** — набор классов, поддерживающих дополнительные функции для разработки Linux-приложений. Например, классы для поддержки Zip, OpenGL, Cairo, Gtk+.

Данный пакет устанавливается автоматически при первом запуске `winecfg`.

Пакет Gecko

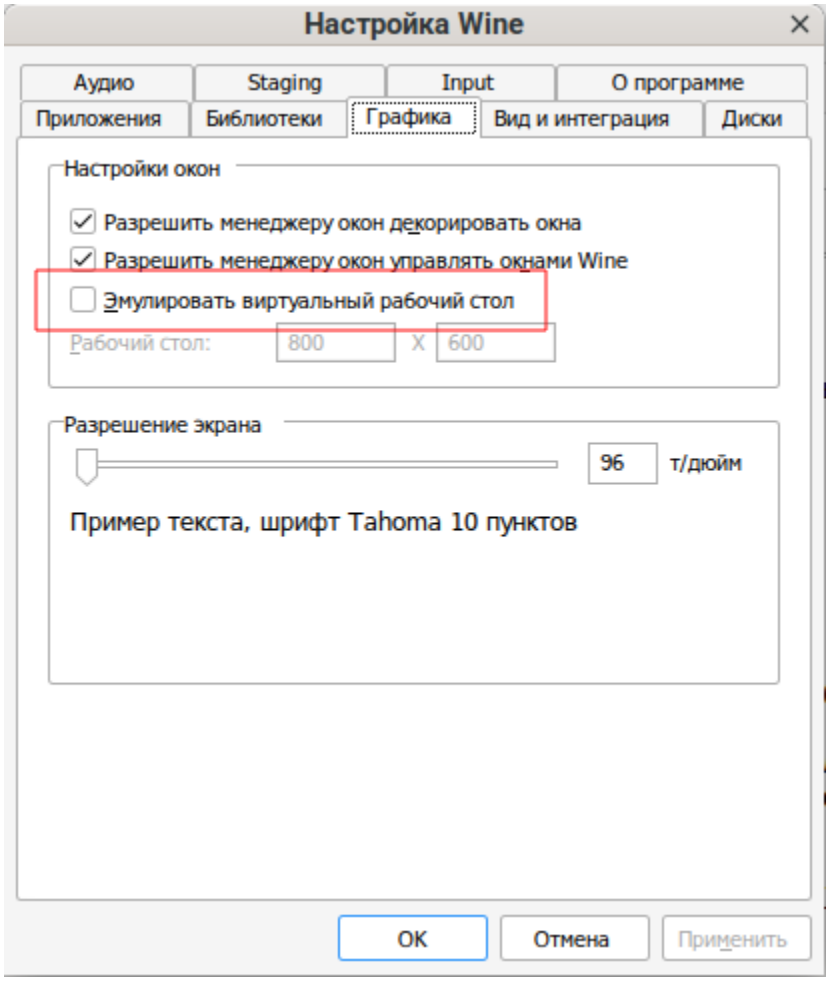
Gecko – это свободный движок вывода веб-страниц для браузера при работе Wine в Интернете. Если Wine не работает с Интернетом и вы не запускаете браузер `windows.exe`, то Gecko не нужен.

Обычно при запуске приложения для работы с веб-страницами Wine автоматически ищет Gecko сначала в `/usr/share/wine/gecko`. Если Wine не находит архив с Gecko там, то он автоматически загружает его из Интернета, но некоторые приложения не любят ждать, пока загрузиться Gecko и зависают или вылетают.

Виртуальный рабочий стол Wine

Имитация рабочего стола Windows в Wine необходима для решения проблем, связанных с разрешением экрана, а также при сворачивании программ в Wine.

Рабочий стол можно включить через `winecfg`.



Запущенная программа размещается в рамках виртуального рабочего стола.

Также есть возможность запустить рабочий стол, не используя интерфейс:

```
$ wine explorer /desktop="desk",800x600 regedit
```

- `explorer` — проводник Windows;
- `/desktop="desk",800x600 regedit` — опции для создания рабочего стола с именем `desk` и разрешением 800x600, затем запустить программу `regedit` (редактор реестра).

Дополнительный функционал Wine

`Wineconsole` позволяет запускать консоль `windows`, в основном используется для отладки. Запускается командой:

```
$ wineconsole cmd
```

`Wineserver` — служба, исполняющая роль ядра Windows. Для каждого префикса запускается своя служба, что позволяет выполнять одновременно несколько программ из разных префиксов. В случаях, когда программа зависла, поможет перезапуск `wineserver` («жёсткая перезагрузка»):

```
$ wineserver -k
```

Или с префиксом:

```
$ WINEPREFIX="/home/$USER/.wine_prefix" wineserver -k
```

- **`wineserver -d[n]:` — выводит отладочную информацию, имеет 3 уровня:**
  - 0 — не выводит информацию;
  - 1 — выводит основную информацию;
  - 2 — выводит полную информацию.
- `wineserver -f:` — выводит сервер на передний план для упрощения отладки;
- `wineserver -p[n]:` — определяет время до остановки сервера, когда все клиентские процессы завершились, что позволяет быстрее запускать приложения. По умолчанию 3 секунды, если значение `n` не указано, сервер будет работать вечно;
- `wineserver -w:` — ожидает завершения предыдущей сессии ядра для корректного старта новой сессии.

`Wineboot` выполняет инициализацию, запуск и перезагрузку префикса. Также выполняет симуляцию перезапуска и завершения работы программ («мягкая перезагрузка»).

```
$ wineboot -r
```

Создание префикса через `wineboot`:

```
$ WINEPREFIX="/home/$USER/.wine_prefix" wineboot -r
```

- `-i` — инициализировать префикс.

Как установить сторонние браузеры в МСВСфера ОС

Окружение

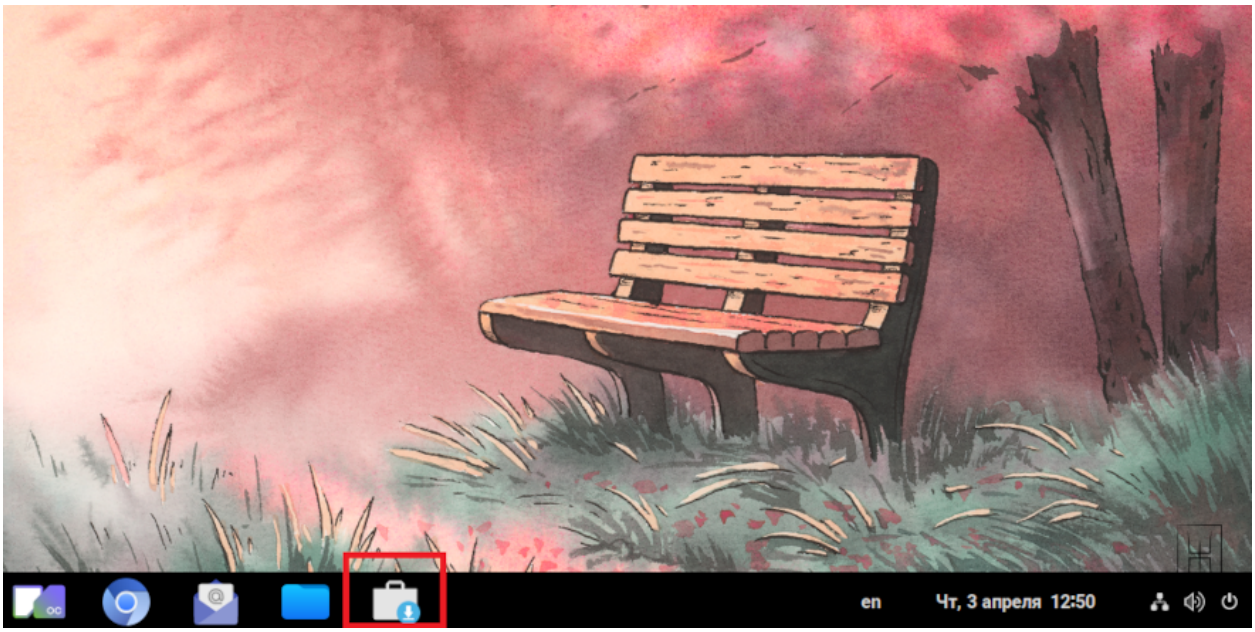
- МСВСфера Сервер версии 8.X/9.X.
- МСВСфера АРМ версии 8.X/9.X.

Браузер Firefox

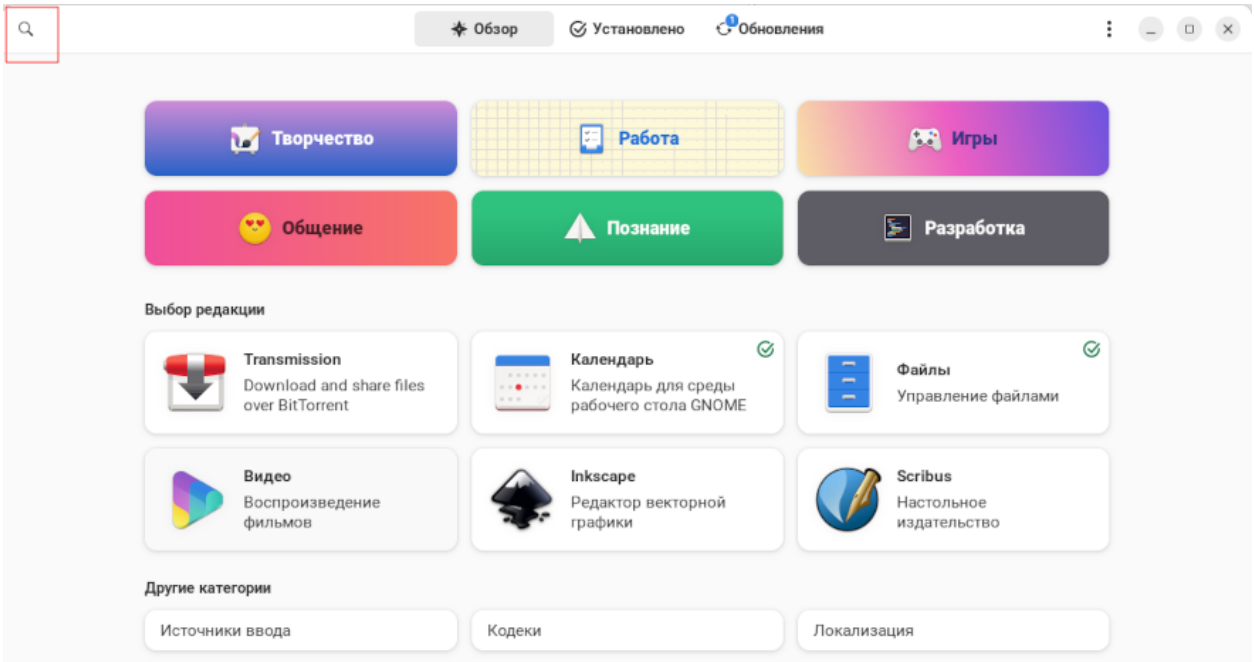
Firefox — бесплатный браузер, разрабатываемый и распространяемы компанией **Mozilla Corporation**. Это один из самых популярных браузеров в мире.

Для установки данного браузера сделайте следующее:

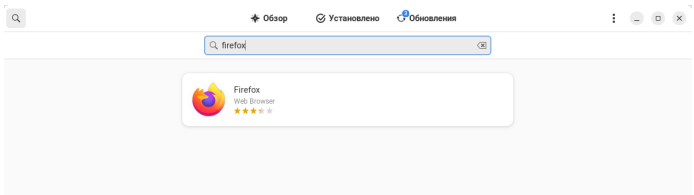
1. Откройте **Центр Приложений**.



2. Нажмите кнопку поиска в левом верхнем углу.



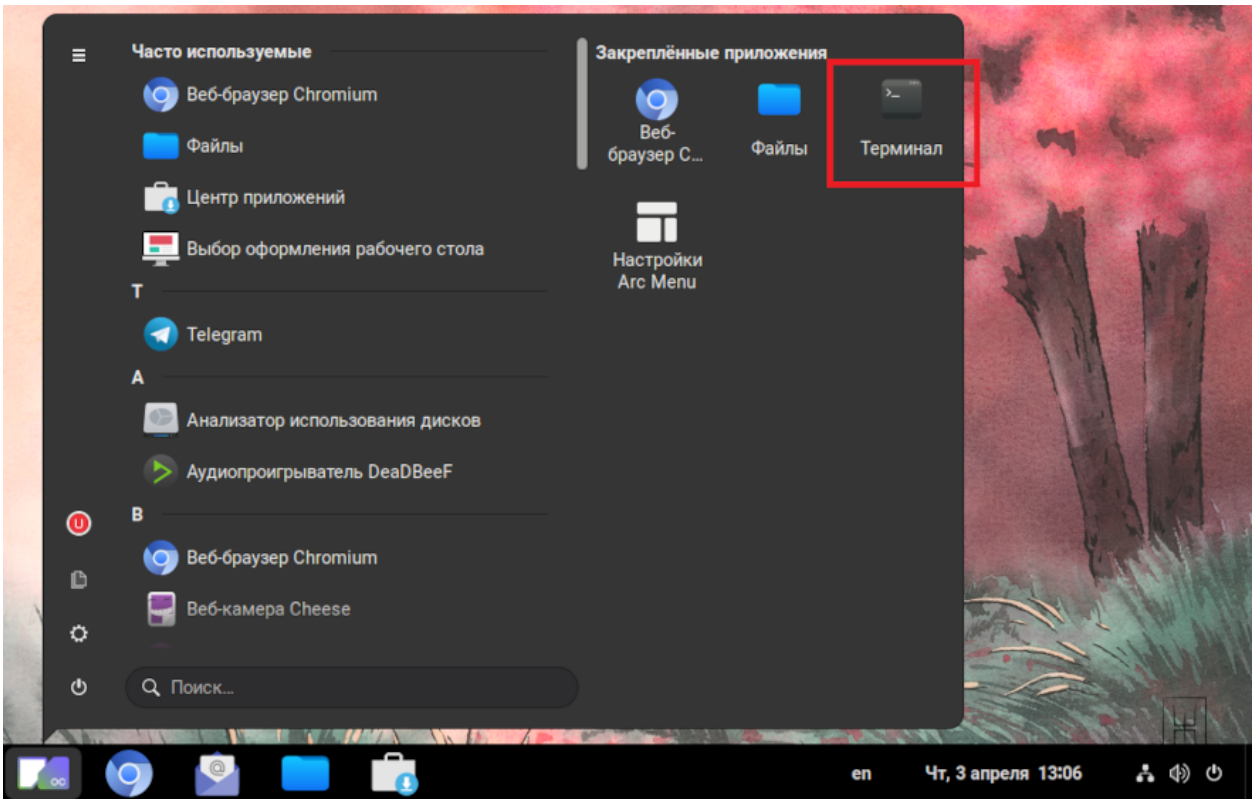
3. Наберите «firefox» в поле поиска и нажмите на «Firefox» в результатах.



4. В окне приложения нажмите кнопку **Установить**.

Также можно установить браузер через «Терминал». Для этого выполните следующие действия.

1. Откройте «Терминал».



2. Выполните следующую команду:

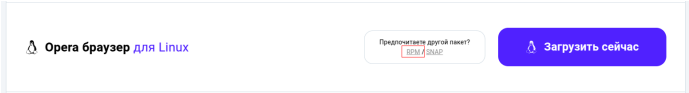
```
$ sudo dnf install firefox
```

Браузер Opera

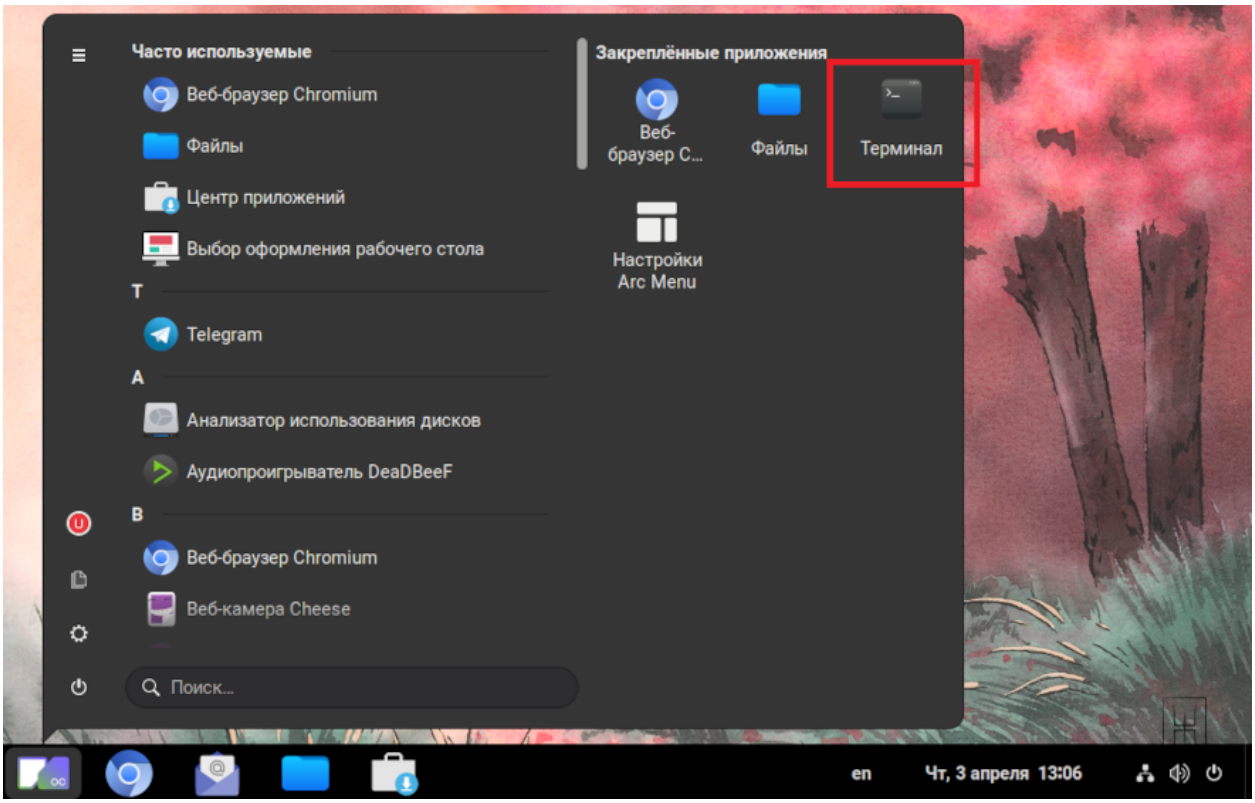
Браузер **Opera** — один самых распространённых браузеров в России.

Для установки выполните следующие действия.

1. Откройте в браузере официальный сайт разработчика.
2. В секции **Opera браузер для Linux** нажмите на **RPM**.



3. После загрузки откройте «Терминал».



4. Перейдите в папку «Загрузки» и установите браузер командами:

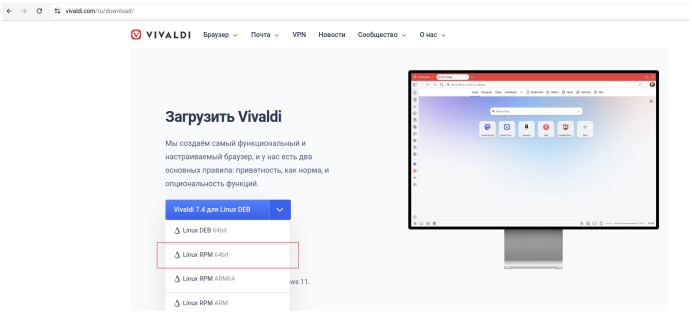
```
$ cd Загрузки/
$ sudo dnf install opera-stable*.rpm
```

Браузер Vivaldi

Vivaldi — браузер, созданный компанией Vivaldi Technologies на основе движка Blink.

Для установки браузера Vivaldi:

1. Перейдите на страницу загрузки официального сайта разработчика.
2. Скачайте версию `Linux RPM 64 bit`.



3. Перейдите в папку «Загрузки» и установите браузер командами:

```
$ cd Загрузки/
$ sudo dnf install vivaldi-stable*.rpm
```

Альтернативные решения для работы с документами, таблицами и презентациями.

В данном документе описаны инструкции для установки аналогов Microsoft Office:

- МойОфис
- WPSOffice
- OnlyOffice
- Р7-Офис

Окружение

- МСВ Сфера АРМ версии 8.x
- МСВ Сфера АРМ версии 9.x

МойОфис

МойОфис — отечественное решение для работы с текстовыми документами, таблицами и презентациями.

Стандартная версия состоит из следующих компонентов:

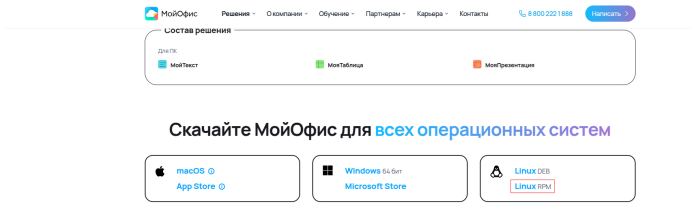
МойОфис Текст — приложение для работы с документами.

МойОфис Таблица — приложение для работы с таблицами.

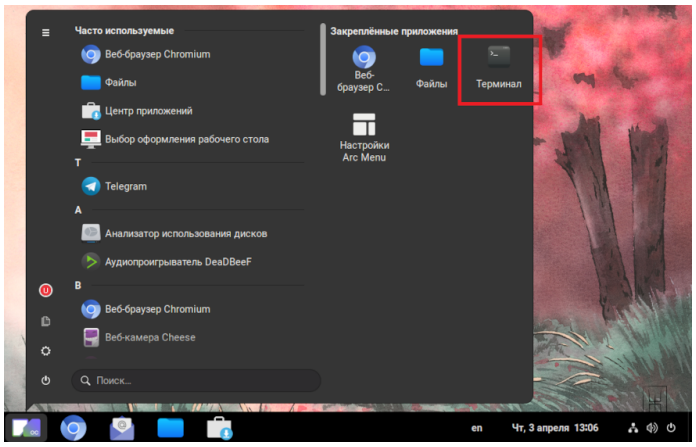
МойОфис Презентация — приложение для создания и редактирования презентаций.

Для установки сделайте следующее:

1. Перейдите на Страницу Стандартной версии на официальном сайте продукта.
2. Скачайте RPM пакет:



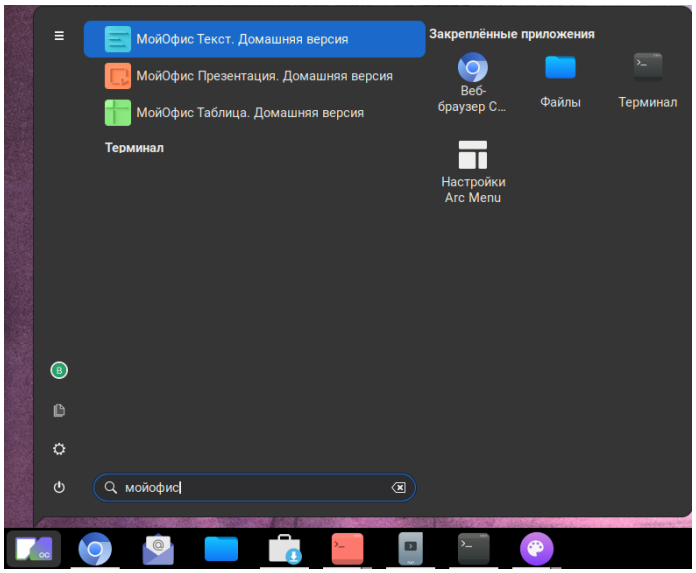
3. Откройте Терминал:



4. Перейдите в папку `Загрузки` и установите **Мой Офис** командой:

```
$ cd ~/Загрузки/ && sudo dnf install ./MyOfficeStandardHomeEdition.rpm
```

Чтобы запустить приложение **Мой Офис**, необходимо открыть Главное меню и набрать в поиске `МойОфис`:



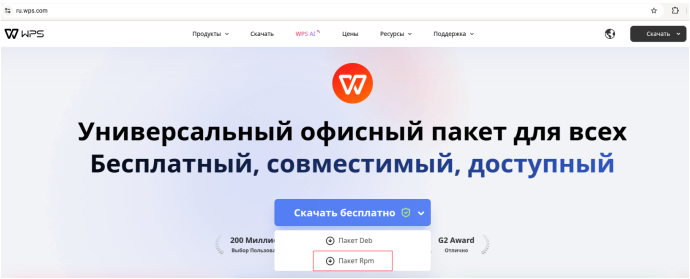
WPS Office

WPS Office — офисный пакет, созданный как альтернатива Microsoft Office компанией KingSoft. Пакет состоит из следующих компонентов:

- Writer — для текстовых документов
- Spreadsheets — для таблиц
- Presentation — для презентаций
- PDF — для PDF-файлов

Для установки сделайте следующее:

1. Перейдите на Официальный сайт WPS Office и скачайте RPM пакет:



2. Откройте Терминал
3. Перейдите в папку `Загрузки` и установите `WPS Office` командой:

```
$ cd ~/Загрузки/ && sudo dnf install ./wps-office*.rpm
```

Запустить необходимый компонент можно через Главное Меню.

OnlyOffice

OnlyOffice — бесплатный пакет для работы с офисными документами (текстовыми, таблицами и презентациями). Также приложение может подключиться к облачной/серверной версии **OnlyOffice**. Редакторы **OnlyOffice** полностью совместимы с **MS Office**

Для установки сделайте следующее:

1. Откройте Терминал
2. Установите репозиторий **OnlyOffice**:

```
$ sudo dnf install https://download.onlyoffice.com/repo/centos/main/noarch/onlyoffice-repo.
↪noarch.rpm
```

3. Установите пакет `onlyoffice-desktopeditors`:

```
$ sudo dnf install onlyoffice-desktopeditors
```

Запуск приложения производится из Главного меню

Р7-Офис

Р7-Офис — отечественный офисный пакет, ориентированный на совместную работу с документами и корпоративное взаимодействие.

> **Примечание**
>
> Устанавливаемый пакет имеет пробную лицензию на 30 дней. По истечении 30 дней будет необходимо приобрести и установить лицензию на официальном сайте разработчика для дальнейшего использования.

Для установки сделайте следующее:

1. Откройте Терминал
2. Добавьте репозиторий `Р7-Офис`:

```
$ sudo cat <<EOF | sudo tee /etc/yum.repos.d/r7.repo
[r7]
name=r7
baseurl=https://downloads.r7-office.ru/repository/r7-desktop-yum/
enabled=1
gpgcheck=1
gpgkey=https://download.r7-office.ru/repos/RPM-GPG-KEY-R7-OFFICE.public
sslverify=0
username=desktop
password=gyxiLab84FByn7sCTd5JY
EOF
```

3. Установите офисный пакет:

```
$ sudo dnf install r7-office
```

Запуск приложения производится из Главного меню.

Как создать снимок экрана и записать видео с экрана в МСВСфера

Операционная система МСВСфера позволяет делать снимок и запись экрана в графической среде `Gnome`.

Окружение

- МСВСфера АРМ/Сервер версии 8.x
- МСВСфера АРМ/Сервер версии 9.x

Снимок экрана

Чтобы сделать снимок экрана, активного окна или выделенной части экрана и сохранить в буфер обмена:

- Часть экрана — нажмите **Shift** + **Ctrl** + **PrtSc** и выделите нужную часть экрана;
- Активное окно — нажмите **Ctrl** + **Alt** + **PrtSc**;
- Весь экран — нажмите **Ctrl** + **PrtSc**.

Также можно сделать снимок экрана, активного окна или выделенной части экрана и сохранить в виде файла в папке **Изображения**:

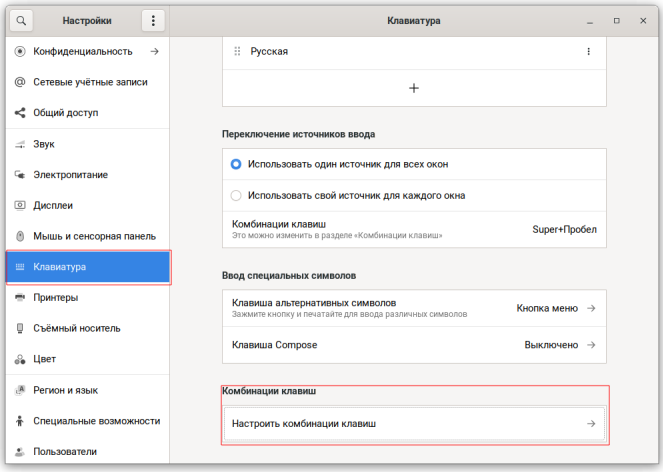
- Часть экрана — нажмите **Shift** + **PrtSc** и выделите нужную часть экрана;
- Активное окно — нажмите **Alt** + **PrtSc**;
- Весть экран — нажмите **PrtSc**.

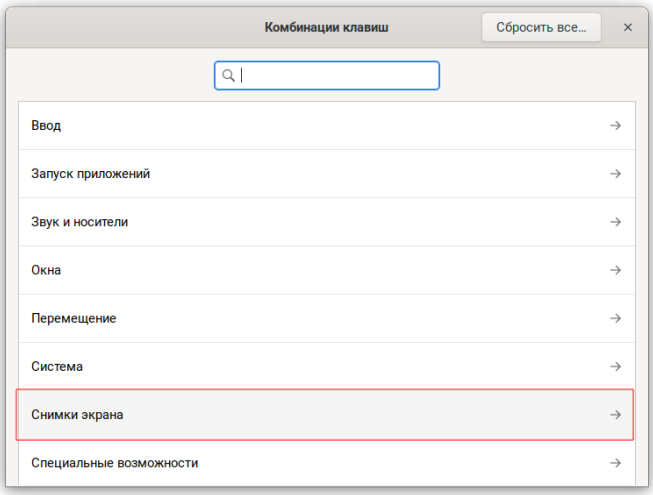
Запись экрана

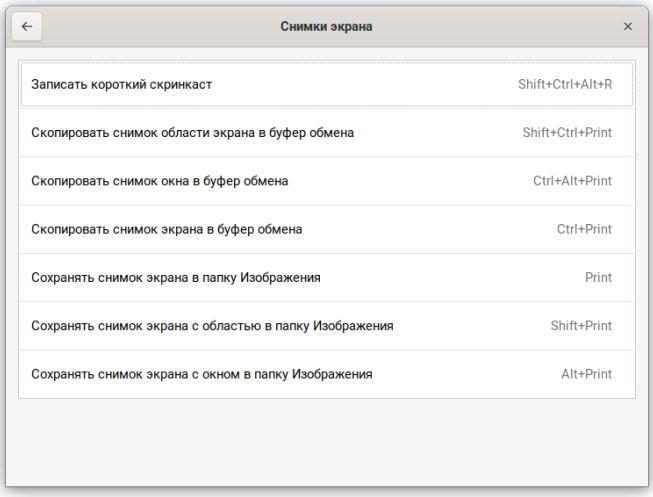
Для того чтобы начать запись с экрана, нужно нажать **Shift** + **Ctrl** + **Alt** + **R**. После этого в верхнем правом углу экрана появится индикация записи в виде красной точки. Чтобы остановить запись, нажмите **Shift** + **Ctrl** + **Alt** +**R** ещё раз. Запись будет сохранена в виде файла в папке **Видео**.

Дополнительные настройки

Комбинацию клавиш можно переназначить в **Настройки -> Клавиатура -> Настроить комбинации клавиш -> Снимки Экрана**:







Установка МСВСфера в Windows с помощью WSL

1. Авторизуйтесь в системе Windows.
2. Откройте терминал (`cmd`) и выполните команду:

```
$ wsl --update
```

Примечание

Данный шаг имеет большое значение для пользователей на архитектуре `ARM Windows`, поскольку импорт манифеста на более старых версиях wsl может быть некорректным.

3. Скачайте скрипт, который поможет импортировать манифест. Скрипт содержит в себе ссылку на манифест, а также команду, которая позволяет импортировать его в систему.
4. Откройте `PowerShell` от имени администратора и введите команду, которая разрешит выполнение сценариев:

```
$ Set-ExecutionPolicy -Scope Process -ExecutionPolicy Unrestricted
```

Во время выполнения команды система запросит подтверждение. Нажмите «Y».

5. Запустите скрипт `append_manifest.ps1`:

```
$ .\append_manifest.ps1
```

Во время выполнения команды система запросит подтверждение. Нажмите «R», что означает запуск скрипта один раз.

6. Верните политику в состояние `default` с помощью команды:

```
$ Set-ExecutionPolicy -Scope Process -ExecutionPolicy Default
```

Во время выполнения команды система запросит подтверждение. Нажмите «Y».

7. Манифест импортирован. Теперь возвращаемся в обычную консоль `cmd` и смотрим на список доступных для скачивания дистрибутивов:

```
$ wsl --list --online
```

Если миграция прошла успешно, вы увидите:

```
C:\Users\edbas>wsl --list --online
Ниже приведен список допустимых дистрибутивов, которые можно установить.
Для установки воспользуйтесь командой "wsl.exe --install <Distro>".

NAME                            FRIENDLY NAME
MSVSphere-10                    MSVSphere OS 10
MSVSphere-9.5                   MSVSphere OS 9
MSVSphere-9.6                   MSVSphere OS 9
MSVSphere-8                     MSVSphere OS 8
AlmaLinux-8                     AlmaLinux OS 8
AlmaLinux-9                     AlmaLinux OS 9
AlmaLinux-Kitten-10             AlmaLinux OS Kitten 10
Debian                          Debian GNU/Linux
FedoraLinux-42                  Fedora Linux 42
SUSE-Linux-Enterprise-15-SP5    SUSE Linux Enterprise 15 SP5
SUSE-Linux-Enterprise-15-SP6    SUSE Linux Enterprise 15 SP6
Ubuntu                          Ubuntu
Ubuntu-24.04                    Ubuntu 24.04 LTS
archlinux                       Arch Linux
kali-linux                      Kali Linux Rolling
openSUSE-Tumbleweed             openSUSE Tumbleweed
openSUSE-Leap-15.6              openSUSE Leap 15.6
Ubuntu-18.04                    Ubuntu 18.04 LTS
Ubuntu-20.04                    Ubuntu 20.04 LTS
Ubuntu-22.04                    Ubuntu 22.04 LTS
OracleLinux_7_9                 Oracle Linux 7.9
OracleLinux_8_7                 Oracle Linux 8.7
OracleLinux_9_1                 Oracle Linux 9.1
```

7. Чтобы установить ОС MSVSphere, выберите версию и введите команду:

```
$ wsl --install MSVSphere-9.6
```

Текстовые редакторы в МСВСфера

В операционных системах МСВСфера имеется множество инструментов для создания и редактирования текстовых документов. В данной статье будут рассмотрены самые популярные текстовые редакторы:

- GNU Emacs
- nano
- pluma

Окружение

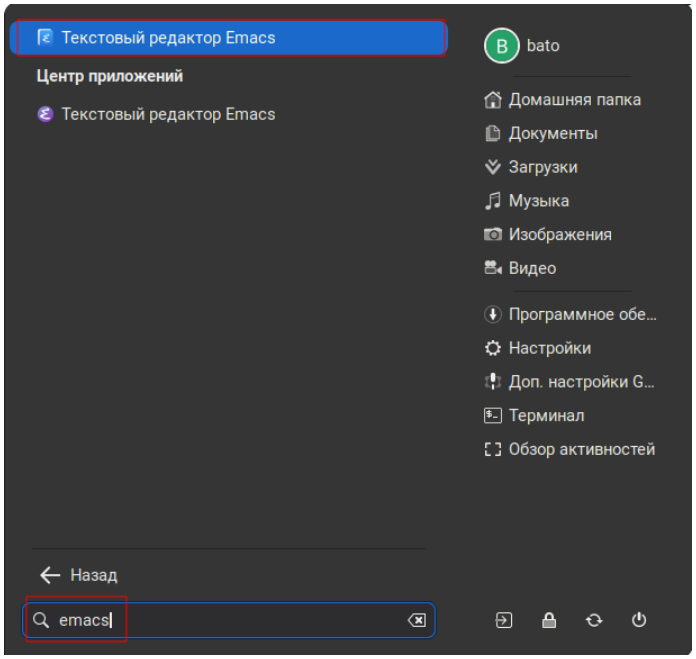
- МСВСфера АРМ/Сервер версии 8.x
- МСВСфера АРМ/Сервер версии 9.x

GNU Emacs

Emacs — это один из самых распространённых текстовых редакторов. Его особенностью является наличие как консольной, так и графической версий. **Emacs** предустановлен в ОС МСВСфера.

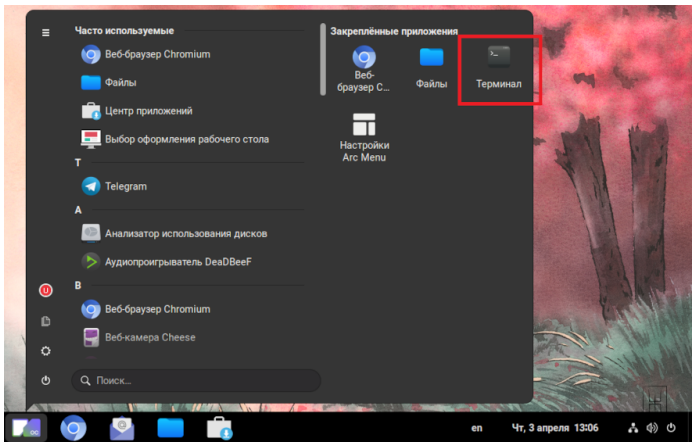
Примечание
GNU Emacs имеет только английскую версию интерфейса.

Для запуска графической версии необходимо открыть **Главное Меню** и в строке поиска набрать `emacs`:



Для запуска консольной версии:

1. Откройте **Терминал**:



2. Выполните команду:

```
$ emacs -nw [имя файла]
```

Если файла с таким именем нет, то будет создан новый файл с этим именем.

Для выполнения действий в редакторе необходимо нажимать определённые комбинации клавиш. Например:

- Сохранить — `Ctrl+X`, затем `Ctrl+S`;
- Сохранить как — `Ctrl+X`, затем `Ctrl+W`, затем необходимо указать имя нового файла;
- Выйти из редактора — `Ctrl+X`, затем `Ctrl+C`;
- Открыть руководство по командам — `Ctrl+H`.

Для более подробной информации по консольному **Emacs** откройте руководство пользователя с помощью команды:

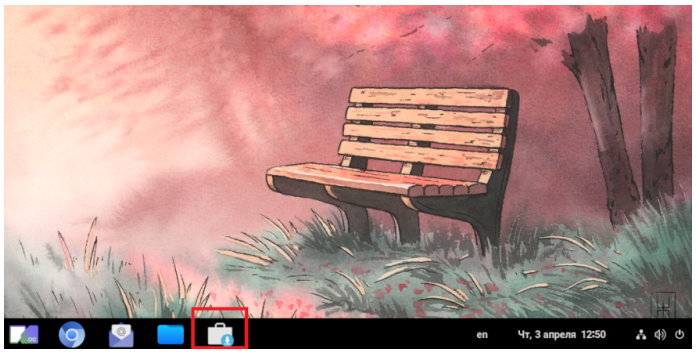
```
$ man emacs
```

Pluma

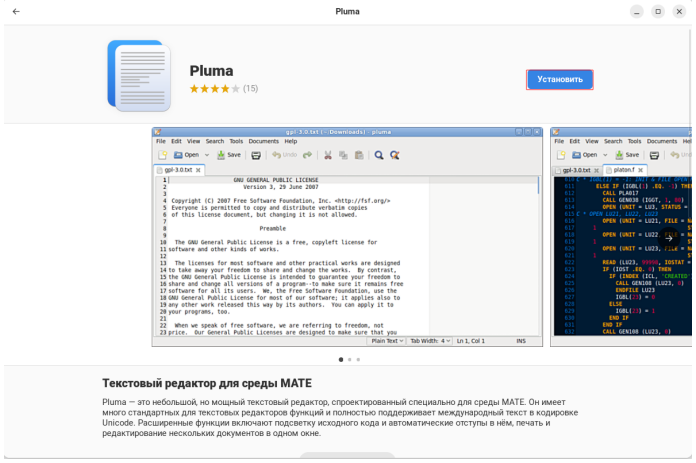
Pluma — это небольшой, но мощный графический текстовый редактор, спроектированный специально для среды MATE. Он имеет много стандартных для текстовых редакторов функций и полностью поддерживает международный текст в кодировке Unicode. Расширенные функции включают подсветку исходного кода и автоматические отступы в нём, а также печать и редактирование нескольких документов в одном окне.

Для того чтобы установить его, сделайте следующее:

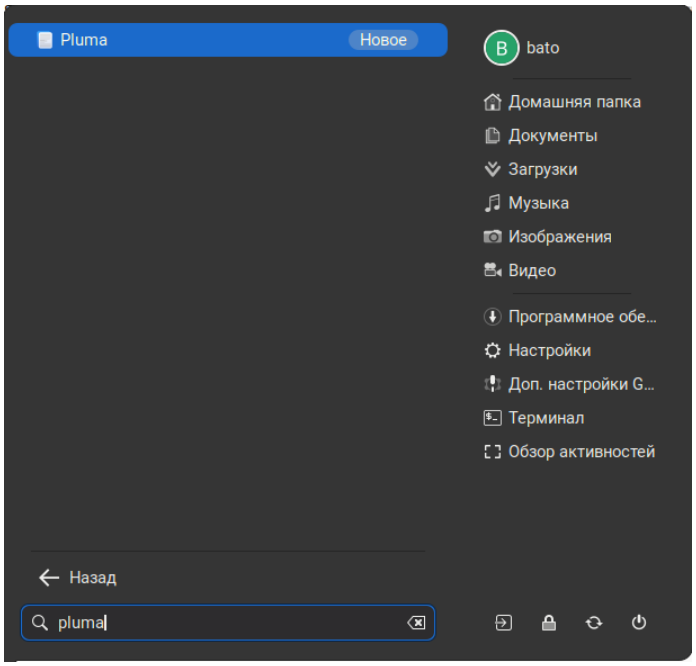
1. Откройте **Центр Приложений**:



2. В поиске наберите `Pluma` и нажмите на соответствующий результат;
3. Нажмите кнопку **Установить**:



Запустить редактор можно через **Главное меню**:



nano

nano — это простой и удобный консольный текстовый редактор. Редактор предустановлен в МСВСфера по умолчанию.

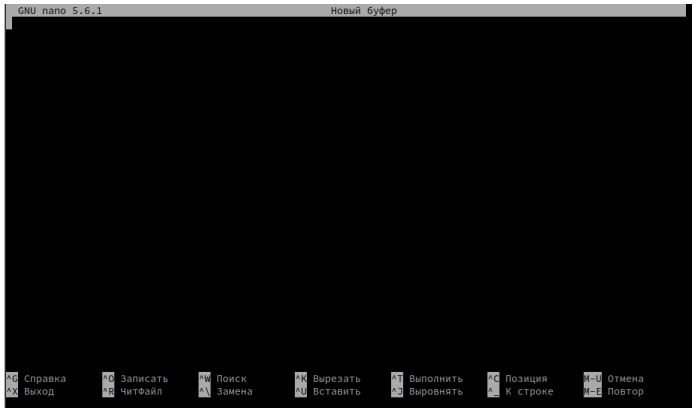
Для того чтобы открыть редактор, сделайте следующее:

1. Откройте **Терминал**
2. Выполните команду:

```
$ nano [имя файла]
```

Если файла с таким именем нет, то будет создан новый файл с этим именем.

Интерфейс редактора простой и интуитивно понятный:



Сверху отображена версия редактора, имя файла и состояние (изменён или не изменён). Снизу отображена подсказка по действиям. Здесь `^` обозначает нажатие клавиши `Ctrl` на клавиатуре. Например, для того чтобы выйти из редактора, надо нажать комбинацию клавиш `Ctrl+X`.

Для более подробной информации по использованию **nano** откройте руководство пользователя с помощью команды:

```
$ man nano
```

Текстовые редакторы vi и vim

Редактор **vi** — универсальный полноэкранный текстовый редактор в среде UNIX. Универсальность означает, что, во-первых, этот редактор есть во всех UNIX-подобных ОС и, во-вторых, этот редактор работает с практически любым видом терминала (ANSI, VT100, VT220, VT320, DEC и т. д.). Эта универсальность выражается непривычным (для пользователей DOS) пользовательским интерфейсом: для управления редактором используются лишь «обычные» кнопки клавиатуры (алфавитно-цифровые символы и знаки препинания).

Имеющиеся на многих типах терминалов функциональные клавиши практически не используются. Если клавиатура терминала имеет стрелочные клавиши, то они используются, но в ограниченном контексте.

Окружение

- МСВСфера Сервер версии 8.X/9.X
- МСВСфера АРМ версии 8.X/9.X

Редактор vi

Данный редактор установлен в ОС МСВСфера по умолчанию.

Режимы работы

- **Ввод текста**

  В этом режиме всё, что набирается на клавиатуре, отображается на экране терминала и запоминается в буфере редактора.

- **Командный режим**

В этом режиме символы клавиатуры выполняют специальные функции (перемещение курсора, стирание текста, и т. д.), то есть функции редактирования. В данном режиме набираемые команды не отображаются на экране.

- **Режим командной строки**

  Режим командной строки позволяет производить более глобальные операции с текстом: записывать отредактированный текст в файл, читать новый файл, выходить из vi, производить настройку редактора, поиск по шаблону, а также выполнять некоторые функции редактирования. Команды отображаются в нижней части экрана (в командной строке редактора).

Ввод текста

- **Return** или **Enter** — создает пустую строку и переводит курсор в её начало;
- **Ctrl+H** — удаляет последний введённый символ (это действие не отображается на экране до выхода в командный режим);
- **Ctrl+[** или клавиша **Esc** — переводит редактор в командный режим.

Командный режим

- `i` — переход в режим набора текста методом вставки перед текущим символом;
- `a` — переход в режим набора текста методом вставки за текущим символом;
- `R` — переход в режим набора текста методом замены.

Перемещение курсора

- `h, j, k, l` — на один символ (одну строку), как показано стрелками;
- `^` или `0` — в начало текущей строки;
- `$` — в конец текущей строки;
- `w` — на слово вправо;
- `b` — на слово влево;
- `}` — на параграф вперед (параграф — это блок текста, отделённый пустой строкой);
- `{` — на параграф назад;
- `[[` — в начало текста;

-]] — в конец текста.

Кнопки стрелочной клавиатуры также позволяют перемещаться по тексту.

Таблица 7: **Другие полезные команды**

Команда	Описание команды
R, i	Переход в режим ввода текста с заменой — обычный ввод текста
I	Переход в режим ввода с начала текущей строки
o	Переход в режим ввода с новой строки под курсором
O	Переход в режим ввода с новой строки над курсором
a	Переход в режим ввода после курсора
x	Стирание символа после курсора
X	Стирание символа перед курсором
dd	Стирание текущей строки
d<число>d	Стирание числа строк, начиная с текущей
yy	Копирование текущей строки в неименованный буфер
y<число>y	Копирование числа строк, начиная с текущей, в неименованный буфер
p	Вставка строки из неименованного буфера под курсором
P	Вставка строки из неименованного буфера над курсором
J	Слияние текущей строки со следующей
u	Отмена последней команды
.	Повтор последней команды
:	Переход в режим командной строки

Режим командной строки

Режим командой строки позволяет производить действия с файлом или с отображением файла. Для вызова того или иного действия в режиме командной строки, необходимо начать команду с символа :. Ниже представлен список наиболее полезных команд.

- :q или :q! — выход из редактора без сохранения изменений;
- :x — выход из редактора с записью, если файл был модифицирован;
- :w или w <имя_файла> или w! <имя_файла> — запись файла и возвращение в командный режим;
- :e <имя_файла> или :e! <имя_файла> — загрузка указанного файла;
- :r <имя_файла> — добавить содержимое указанного файла к редактируемому сразу за текущей строкой;

- `:set number` — включить нумерацию строк;
- `:set nonumber` — отключить нумерацию строк;
- `:!command` — выполнить команду UNIX, не покидая редактора;
- `:/word` — выполнить поиск слова `word` в тексте;
- `:/` — повторить поиск слова `word` далее по тексту.

Повторители, буферы и окна редактирования

Повторители

Командам и движениям курсора можно давать повторители (числа), например:

- `2w` — передвинуть курсор на два слова вперёд;
- `10l` — передвинуть курсор на десять символов вправо;
- `d10l` — стереть десять символов справа от курсора;
- `2d10l` — стереть двадцать символов справа от курсора;
- `5J` — слить пять последующих строк в одну;
- `4.` — повторить последнюю введённую команду четыре раза.

Буферы

Редактор имеет три типа буферов: буфер стирания (0-9), неименованный буфер и именованные буферы (a-z).

В буферы стирания автоматически заносятся стираемые элементы. В буфере 0 хранится последний стёртый элемент, в буфере 1 — предпоследний и т.д.

Вот некоторые команды для операций с буфером:

- `yy` — занести текущую строку в неименованный буфер;
- `y<движение_курсора>` — занести указанный движением курсора блок текста в неименованный буфер;
- `ayy` — занести текущую строку в именованный буфер «a»;
- `Ayy` — добавить текущую строку к содержимому именованного буфера «a»;
- `by10j` — занести последующие 10 строк в именованный буфер «b».
- `p` — вставить в текущую позицию содержимое неименованного буфера;
- `ap` — вставить в текущую позицию содержимое именованного буфера «a»;
- `1p` — вставить в текущую позицию содержимое буфера стирания «1».

Окна редактирования

Редактировать сразу несколько файлов можно, указав все необходимые файлы в командной строке при вызове редактора (например: `vi <файл_1> <файл_2> <файл_3>`). Примеры команд для перехода к другому файлу:

- `:n` — переход к следующему файлу в списке;
- `:rew` — возврат к редактированию первого файла в списке.
- `:e <имя файла>` — создание нового или открытие существующего файла.

Именованные буферы сохраняют своё содержимое при переходе к редактированию другого файла.

Дополнительные опции

Таблица 8: **Дополнительные команды редактора**

Команда	Описание команды
`G`	Переход в конец файла
`<номер строки> G`	Переход на конкретную строку в файле
`<число>`	Переход на `<число>` строк вперед
`:set wrap`	Перенос длинной строки
`:syntax on/off`	Включить/выключить подсветку синтаксиса
`/<слово>`	Поиск в тексте по определённому слову
`n`	Переход к следующему результату поиска
`:h`	Отображение руководства по редактору

Расширенный редактор vim

Редактор **vim** расшифоровывается как **vi improved**. Имеет расширенный функционал по сравнению с редактором **vi**.

Основные отличия от **vi**.

- Работа с несколькими файлами одновременно. Разбиение окон редактирования может производиться многократно как по горизонтали, так и по вертикали.
- Поддержка Unicode.
- Поддержка визуального режима, который позволяет, например, выполнять операции над блоками текста.
- Неограниченная глубина отмены (undo) и возврата (redo) действий.

- Режим сравнения двух файлов, перенос отдельных изменений из одного файла в другой.
- Широкая файловая поддержка (файл со справкой и более 200 файлов с описанием синтаксиса).
- Подсветка синтаксиса, автоматическое определение величины отступа для каждой строки в зависимости от языка программирования (изначально поддерживает более 200 языков программирования и форматов конфигурационных файлов).
- Интеграция с операционной системой, дающая возможности, близкие к интегрированным средам разработки, такие как поиск ошибки по сообщению компилятора, автодополнение идентификаторов и др..
- Поддержка языка сценариев; возможность написания модулей расширения — плагинов.
- Автоматическое продолжение команд, слов, строк целиком и имён файлов.
- Автоматический вызов внешних команд (например, автоматическая распаковка файла перед редактированием).
- Распознавание и преобразование файлов различных форматов.
- Удобный механизм истории команд, поисковых слов и т. д..
- Запись и исполнение макросов.
- Возможность сохранения настроек и сеанса.
- Возможна интеграция с языками программирования Perl, Tcl, Python и Ruby.
- Поддержка языков с письмом справа налево (арабских и других).
- Сворачивание (folding) текста для лучшего обзора.
- Возможность использования графического интерфейса в специальных версиях (GTK, Motif, и др.).
- Хорошо конфигурируется и настраивается под нужды пользователя.
- Для программистов: поддержка цикла разработки «редактирование — компиляция — исправление» программ. Автоматическое выполнение сборки/компиляции, обнаружение и распознавание ошибок, переход к строкам ошибок в тексте программы.

Данный редактор также установлен по умолчанию.

Режимы работы редактора vim

В **vim** существуют 3 режима работы:

- **Основной** — предназначен для просмотра файла, ввода команд и перехода из него в другие режимы. Из любого режима в командный можно попасть, нажав **ESC**. При нажатии клавиши `:` становится доступна командная строка vim, в которой можно вводить команды. Основные команды — команда выхода `quit` (vim понимает сокращения, поэтому можно давать команду одной буквой `q`), команда сохранения `write` (или `w`), параметром которой может быть имя файла и вызов справки по `help` (или `h`). На остальные клавиши (и их последовательности) можно присвоить любое действие, либо использовать значения по умолчанию.
- **Визуальный** — предназначен в первую очередь для выделения блоков текста. Предлагаются 3 варианта перехода в этот режим – клавишей `v` для посимвольного выбора, `Shift+v` для построчного и `Ctrl+v` для блочного. В нормальном режиме (при переходе по `v`) можно оперировать следующими сущностями: слово (`w`), предложение (`s`), параграф (`p`) и блок (`b`). Выделение при этом необходимо начинать с позиции курсора (`a`), или же с начала блока (`i`). Например, выделение текущего блока (участка, ограниченного парными элементами) можно выполнить комбинацией `Esc+vib`. Копирование в буфер выделенного текста осуществляется по `y`, вырезание по `d`, а вставка по `p`.
- **Режим редактирования** — переход в режим редактирования осуществляется нажатием клавиши `Ins`.

Основные возможности

Переходы

Для перехода на строку с номером `n` воспользуйтесь командой `G`. Так, для перехода к началу текста наберите `0G`, для сотой строки — `100G`, а для конца документа — `$G`. Для перехода на `n` символов в нужную вам сторону вы можете использовать клавиши со стрелками. То есть для перехода на 1000 символов вниз наберите `1000` и нажмите стрелку вниз.

- Для перемещения по тексту используйте следующие команды: `(`, `)`;
- для перемещения по предложениям — `{`, `}`;
- для параграфов — `[[`, `]]`;
- для функций: `%` — переход к парной скобке, `CTRL+O`, `CTRL+I` — переход назад и вперед соответственно по истории переходов.

Метки

Используются для отметки позиции `^<метка>`, где `<метка>` — любая буква для быстрого к ней перехода. Метки нижнего регистра действительны в пределах данного файла, метки верхнего регистра действуют во всех открытых файлах. Список всех меток можно получить командой `marks`. Для того, чтобы установить метку на текущей строке используйте клавиши **M** и **X**. Для того, чтобы перейти на метку **X** введите команду: `' X`.

Фолды

Предназначены для скрытия ненужных в данный момент данных, чтобы те не отвлекали внимания. Например, кода подпрограммы, с которой вы в данный момент не работаете. По умолчанию фолды активированы в режиме ручной расстановки. Если вы хотите их автоактивации по отношению к табуляции, то добавьте в конфигурационный файл следующую строку:

```
set foldmethod=indent
```

Все команды для работы с фолдами начинаются с `z`. Открытие фолда производится, например, по `zo` (или стрелке вправо), закрытие кода в фолд — по `zc`.

Сессии

При ведении группы проектов нередко требуется сохранить текущее состояние и настройки редактора, чтобы затем продолжить работу с того же места. Для этого и предназначены сессии, которые создаются командой `:mksession /path/to/Session.vim`, а читаются командой `:so /path/to/Session.vim`. Гораздо чаще возникает необходимость сохранить не всю сессию, а только текущий контекст (сюда входит, например, положение курсора в коде, текущая расстановка фолдов и многое другое). Это действие выполняет команда `:mkview`, чтение — `:loadview`. Очень удобно сделать сохранение и чтение контекста автоматическим при начале и окончании редактирования файла. В vim команда `au` используется для создания автоматических команд. Это может быть реализовано следующим кодом (применяется для всех файлов, имеющих точку в имени). Нажмите клавишу `:` и введите следующие команды по одной (нажимая **Enter** после каждой):

```
au BufWinLeave *.* mkview
au BufWinEnter *.* silent loadview
```

Поиск

Поиск осуществляется командой `/` для поиска (по регулярному выражению) вперед и `?` в обратном направлении. Для продолжения поиска используйте `n`, а для прошлого варианта `N`. Для поиска слова под курсором используются соответственно `#` и `*`.

Автодополнение

Производится по содержимому данного файла, а также указанных в переменной `dictionary` по нажатию клавиш `''`.

Отмена

Нажмите:

- `u` для отмены и `''` для смены регистра.
- `~` для выделенного участка (или буквы под курсором).
- `U` — принудительно установить верхний регистр, а `u` — нижний регистр.

Повторение

Нажмите `.` для повтора действия.

Конфигурация

Основным конфигурационным файлом является `~/.vimrc`. Активация русского шрифта в GUI-режиме плюс выбор темы для обоих режимов осуществляется, например, следующим кодом:

```
if has("gui_running")
colorscheme candy
set guifont=-cronyx-courier-medium-r-normal-*-*-120-*-*-m-*-koi8-r endif
if !has("gui_running") colorscheme elflord endif
```

Перечень наиболее используемых «горячих» клавиш:

Выход по **F10**:

```
nmap <F10> :q<CR>
imap <F10> <ESC>:q<CR>
```

Сохранение по **F2**:

```
nmap <F2> :w<CR>
imap <F2> <ESC>:w<CR>i<Right>
```

Компиляция по **F9**:

```
nmap <F9> :make<CR>
imap <F9> <ESC>:make<CR>
```

Утилиты для архивации и распаковки файлов

В данной инструкции рассмотрены популярные утилиты для архивации и распаковки файлов, доступные в ОС МСВСфера:

- tar
- bzip2
- gzip
- xz
- zstd
- zip
- unrar

Окружение

- МСВСфера АРМ/Сервер версии 8.x.
- МСВСфера АРМ/Сервер версии 9.x.

tar

tar — один из самых популярных инструментов архивации файлов в ОС семейства Linux. Утилита **tar** установлена в МСВСфера по умолчанию.

Синтаксис

```
$ tar [параметры] <файл_архива> [файлы]
```

Параметры

- `-A`, `--concatenate` — присоединить существующий архив к другому;
- `-c`, `--create` — создать новый архив;
- `-t`, `--list` — вывести список содержимого архива, если файлы не указаны, то вывести информацию обо всех файлах;
- `-x`, `--extract` — распаковать файлы из архива;
- `-f` — вывести результат в file;
- `-C <dir>` — изменить директорию на `<dir>` перед выполнением операции;
- `-j`, `--bzip2` — использовать сжатие **bzip2** при создании файла `tar`, в результате получается сжатый архив с расширением `*.tar.bz2`;
- `-v`, `--verbose` — вывести всю информацию о текущем процессе и показать сведения об уже завершённом процессе.

Примеры использования

Архивация файла в формат `tar` без сжатия:

```
$ tar -cf file.tar file.txt
```

Архивация каталога в формат `tar` без сжатия:

```
$ tar -cf files.tar ~/files
```

Распаковка архива в текущий каталог:

```
$ tar -xf files.tar
```

Распаковка архива в указанный каталог:

```
$ tar -xf files.tar -C /home/folder
```

Просмотр содержимого архива без распаковки:

```
$ tar -t files.tar
```

Дополнительную информацию о работе утилиты вы можете получить из руководства:

```
$ man tar
```

bzip2

bzip2 — это утилита командной строки, предназначенная для сжатия и распаковки файлов. Она использует алгоритм сжатия текста с блочной сортировкой Барроуза-Уилера и кодирования Хаффмана. Принимает список имён файлов в качестве входных данных и заменяет каждый файл его сжатой версией. Утилита установлена в ОС МСВСфера по умолчанию.

> **Примечание**
>
> Утилита **bzip2** может работать только с одиночными файлами. Для работы с несколькими файлами или папками можно использовать эту утилиту вместе с **tar**. Пример такого варианта использования приведён ниже.

Синтаксис

```
$ bzip2 [параметры] <файл>
```

Параметры

- `-d`, `--decompress` — разархивировать файл;
- `-t`, `--test` — проверить целостность архива;
- `-z`, `--compress` — принудительно сжать;
- `-k`, `--keep` — сохранить исходные файлы при архивации или разархивации;
- `-s` — использовать меньше оперативной памяти за счёт снижения скорости архивации, рекомендуется применять на компьютерах с малым объемом ОЗУ (8 Мбайт и меньше);
- `-v`, `--verbose` — выводить дополнительную информацию в процессе работы;
- `-V`, `--version` — показать версию программного обеспечения и лицензию;
- `-h`, `--help` — вывести справку;
- `-1` — `-9` — установить степень сжатия (чем больше число, тем выше степень сжатия).

Примеры использования

Архивация одного файла с наивысшей степенью сжатия и с сохранением исходного файла. В результате будет создан новый файл с расширением `.bz2`:

```
$ bzip2 -9k file.txt
```

Распаковка файла в текущую папку с сохранением исходного архива:

```
$ bzip2 -dk file.txt.bz2
```

Архивация файла с использованием **tar** для сжатия нескольких файлов:

```
$ tar -cjf file.tar.bz2 file1 file2 file3
```

Распаковака архива `.tar.bz2` в текущую папку:

```
$ tar -xjf file.tar.bz2
```

Дополнительную информацию о работе утилиты вы можете получить из руководства:

```
$ man bzip2
```

gzip

gzip — один из самых популярных алгоритмов сжатия, который позволяет уменьшить размер файла и сохранить исходный режим файла, владельца и отметку времени. Утилита установлена в ОС МСВСфера по умолчанию.

> **Примечание**
>
> Для запаковки нескольких файлов в один архив можно использовать эту утилиту вместе с **tar**. Пример такого варианта использования приведён ниже.

Синтаксис

```
$ gzip [опция] [файл]
```

Параметры

- `-c`, `--stdout` — печатать в стандартный вывод, сохранять исходные файлы без изменений;
- `-d`, `--decompress` — распаковать архив;
- `-k`, `--keep` — сохранить исходный файл;
- `-l`, `--list` — отобразить содержимое архива;
- `-t`, `--test` — проверить целостность архива;
- `-1` — `-9` — установить степень сжатия (чем больше число, тем выше степень сжатия);
- `-r`, `--recursive` — рекурсивная работа с каталогом;
- `-h`, `--help` — вывести справку по использованию утилиты.

Примеры использования

Архивация файла с сохранением исходного файла:

```
$ gzip -k file.txt
```

Распаковка архива с сохранением исходного файла (архива):

```
$ gzip -dk file.txt.gz
```

Архивация нескольких файлов:

```
$ gzip file1.txt file2.txt
```

Примечание

В результате будет создано 2 архива: `file1.txt.gz` и `file2.txt.gz`.

Архивация нескольких файлов в один архив:

```
$ tar -czf files.tar.gz file1.txt file2.txt
```

Распаковка архива с несколькими файлами в текущую папку:

```
$ tar -xzf files.tar.gz
```

Дополнительную информацию о работе утилиты вы можете получить из руководства:

```
$ man gzip
```

xz

xz — формат контейнера для архивов, использующий алгоритм сжатия `LZMA2` и призванный заменить формат контейнера `lzma`. Как и форматы `gzip` и `bzip2`, он является контейнером для единственного файла, поэтому его обычно используют совместно с форматом `tar`. Утилита установлена в ОС МСВСфера по умолчанию.

Синтаксис

```
$ xz [параметры] <файл>
```

Параметры

- `-d` — принудительная разархивация;
- `-z` — сжимает входной файл (используется по умолчанию);
- `-t`, `--test` — проверить целостность архива `*.xz`;
- `-k`, `--keep` — сохранять исходные файлы при архивации или разархивации;
- `-c`, `--stdout` — направить сжатые данные в стандартный вывод и не удалять входные файлы;
- `-v`, `--verbose` — отображать процесс сжатия или распаковки файлов;
- `-0` – `-9`, — установить степень сжатия (чем больше число, тем выше степень сжатия).

Примеры использования

Архивация файла с сохранением исходного файла с максимальной степенью сжатия:

```
$ xz -9k file.txt
```

Распаковка архива:

```
$ xz -d file.txt.xz
```

Архивация нескольких файлов в один архив с помощью **tar**:

```
$ tar -cJf files.tar.xz file1.txt file2.txt
```

Распаковка сжатого `.tar.xz` архива в текущую папку:

```
$ tar -xJf files.tar.xz
```

Проверка целостности без распаковки:

```
$ xz -t files.tar.xz
```

Дополнительную информацию о работе утилиты вы можете получить из руководства:

```
$ man xz
```

zstd

zstd — алгоритм сжатия данных без потерь. Алгоритм сочетает словарный алгоритм сжатия данных типа LZ77 и эффективное энтропийное кодирование типа ANS. Утилита установлена в ОС МСВСфера по умолчанию.

Синтаксис

```
$ zstd [параметры] <файл>
```

Параметры

- `-d` — принудительная разархивация;
- `-t` — проверка целостности указанного(ых) файла(ов) без разархивации;
- `-k` — сохранять исходные файлы при архивации или разархивации;
- `-s` — снижает требования к объёму используемой оперативной памяти за счет снижения скорости архивации. Эту опцию рекомендуется применять на компьютерах с малым объемом ОЗУ (8 Мбайт и меньше);
- `-v` — выводить дополнительную информацию в процессе работы (представляет интерес в диагностических целях);
- `-1` – `-9` — установить степень сжатия (чем больше число, тем выше степень сжатия).

Примеры использования

Архивация файла с сохранением исходного файла с максимальной степенью сжатия:

```
$ zstd -9k file.txt
```

Распаковка архива с сохранением исходного файла:

```
$ zstd -dk file.txt.zst
```

Использование zstd вместе с tar для сжатия нескольких файлов:

```
$ tar --zstd -cf files.tar.zst file1.txt file2.txt
```

Распаковка архива `.tar.zst`:

```
$ tar --zstd -xf files.tar.zst
```

Проверка целостности архива без распаковки:

```
$ zstd -t file.txt.zst
```

Дополнительную информацию о работе утилиты вы можете получить из руководства:

```
$ man zstd
```

zip

zip — утилита для создания сжатых архивов. Zip-архивы очень часто используются для обмена файлами в Интернете. С помощью этой утилиты можно сжимать как файлы, так и папки. Утилита установлена в ОС МСВСфера по умолчанию.

Синтаксис

```
$ zip [параметры] <имя_архива> <файлы>
```

Параметры

- `-d` — удалить файл из архива;
- `-r` — рекурсивное архивирование, то есть с вхождением во вложенные каталоги;
- `-0` — только архивировать, без сжатия;
- `-9` — наилучшая степень сжатия;
- `-F` — исправить zip-файл;

- `-e` — шифровать файлы.

Примеры использования

Архивация каталога:

```
$ zip archive.zip -r /home/folder
```

Архивация нескольких файлов:

```
$ zip archive.zip file1.txt file2.txt
```

Архивация файла с максимальной степенью сжатия:

```
$ zip -9 archive.zip file1.txt
```

Распаковка архива в текущую папку:

```
$ unzip archive.zip
```

Распаковка в определённую папку:

```
$ unzip archive.zip -d /tmp
```

Извлечение определённого файла из архива:

```
$ unzip archive.zip file2.log
```

Проверка содержимого архива без распаковки:

```
$ unzip -l archive.zip
```

Проверка целостности архива без распаковки:

```
$ unzip -t archive.zip
```

Дополнительную информацию о работе утилиты вы можете получить из руководства:

```
$ man zip
```

unrar

unrar — это инструмент на базе «Терминала», разработанный для извлечения и открытия RAR-архивов в системах семейства Linux.

Для установки выполните команду в «Терминале»:

```
$ sudo dnf install unrar
```

Синтаксис

```
$ unrar [параметры] <имя_архива> [<путь_разархивирования>]
```

Параметры

- `e` — распаковать архив;
- `l` — вывести список файлов внутри архива;
- `p` — вывести распакованный файл в стандартный вывод;
- `t` — проверить архив;
- `v` — вывести подробную информацию про архив;
- `x` — извлечь файлы, сохраняя полный путь внутри архива.
- `-ad` — добавить имя архива к пути распаковки;
- `-ai` — игнорировать атрибуты файлов;
- `-inul` — не выводить сообщения об ошибках;
- `-p` — указать пароль архива;
- `-sl` — распаковывать только файлы, меньше указанного размера;
- `-u` — обновить уже распакованные файлы;
- `-y` — отвечать «y» на все запросы;
- `-x` — не распаковывать указанные файлы;
- `-ts` — сохранять временную метку исходных файлов.

Примеры использования

Распаковка архива в текущую папку:

```
$ unrar x archive.rar
```

Распаковка архива в определенную папку:

```
$ unrar x archive.rar /home/folder
```

Дополнительную информацию о работе утилиты вы можете получить из руководства:

```
$ man unrar
```

Работа с файлами и папками в МСВСфера

В этой инструкции разобраны основные операции с файлами и папками в МСВСфера:

- копирование;
- перемещение;
- удаление;
- создание ссылок (ярлыков).

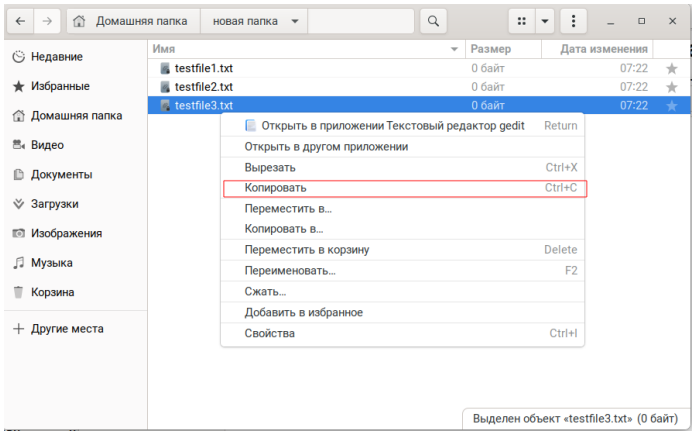
Окружение

- МСВСфера АРМ/Сервер версии 8.x.
- МСВСфера АРМ/Сервер версии 9.x.

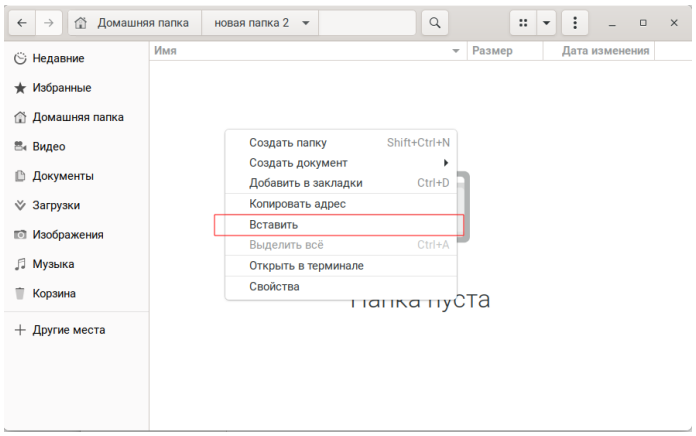
Копирование файлов

Графический интерфейс

1. Откройте папку, в которой находится файл.
2. Нажмите на нужный файл левой кнопкой мыши или выделите несколько файлов (нажимайте на нужные файлы с зажатой кнопкой **CTRL**, либо с зажатой кнопкой **SHIFT** нажмите на первый и последний файл, таким образом выделятся все файлы от начального до конечного. Также можно выделить все файлы в папке комбинацией клавиш **CTRL** + **A**).
3. Нажмите **CTRL** + **C** или нажмите правой кнопкой на выделенный файл и в появившемся меню нажмите **Копировать**.



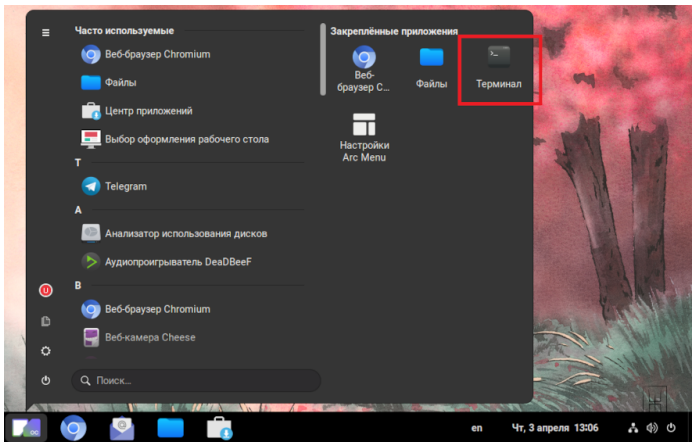
4. Откройте папку, в которую нужно скопировать файл, и нажмите **CTRL + V**, либо нажмите на пустую область в папке и нажмите **Вставить**.



Терминал

Для копирования файлов в «Терминале» используется утилита `cp`.

1. Откройте «Терминал»:



2. Для копирования одного файла, например, `testfile.txt` в папку `/home/target/directory` выполните следующую команду:

```
$ cp testfile.txt /home/target/directory/
```

Для копирования нескольких файлов необходимо указать все файлы поочерёдно:

```
$ cp testfile1.txt testfile2.txt testfile3.txt /home/target/directory/
```

Для копирования всех файлов из текущей папки используйте следующую команду:

```
$ cp * /home/target/directory/
```

Для копирования папки со всем содержимым (папки и файлы) используется атрибут `-R` или `--recursive`:

```
$ cp -R directory1 /home/target/directory/
```

Перемещение файлов

Графический интерфейс

1. Откройте папку, в которой находится файл.
2. Нажмите на нужный файл левой кнопкой мыши или выделите несколько файлов (нажимайте на нужные файлы с зажатой кнопкой **CTRL**, либо с зажатой кнопкой **SHIFT** нажмите на первый и последний файл, таким образом выделятся все файлы от начального до конечного. Также можно выделить все файлы в папке комбинацией клавиш **CTRL + A**).
3. Нажмите **CTRL + X** или нажмите правой кнопкой на выделенный файл и в появившемся меню нажмите **Вырезать**.
4. Откройте папку, в которую нужно переместить файл, и нажмите **CTRL + V**, либо нажмите на пустую область в папке и нажмите **Вставить**.

Терминал

Для перемещения файлов и папок в «Терминале» используется утилита `mv`.

1. Откройте «Терминал».
2. Для перемещения одного файла из текущей папки `testfile.txt` в папку `/home/target/directory` выполните следующую команду:

```
$ mv testfile.txt /home/target/directory/
```

Для копирования нескольких файлов необходимо указать все файлы поочерёдно:

```
$ mv testfile1.txt testfile2.txt testfile3.txt /home/target/directory/
```

Для копирования всех файлов из текущей папки используйте следующую команду:

```
$ mv * /home/target/directory/
```

Удаление файлов и папок

Графический интерфейс

1. Откройте папку, в которой находится файл.
2. Нажмите на нужный файл левой кнопкой мыши или выделите несколько файлов (нажимайте на нужные файлы с зажатой кнопкой **CTRL**, либо с зажатой кнопкой **SHIFT** нажмите на первый и последний файл, таким образом выделятся все файлы от начального до конечного. Также можно выделить все файлы в папке комбинацией клавиш **CTRL** + **A**).
3. Нажмите кнопку **Delete** или **SHIFT** + **Delete** (для безвозвратного удаления) на клавиатуре или нажмите правой кнопкой мыши на выделенном файле и нажмите **Переместить в корзину** или **Удалить безвозвратно...**.

> **Примечание**
>
> При перемещении файла или папки в корзину, соответствующий файл или папку можно восстановить. При безвозвратном удалении восстановить файл или папку невозможно.

Терминал

1. Откройте «Терминал».
2. Для удаления файла используйте следующую команду:

```
$ rm file.txt
```

Для удаления папки со всем содержимым выполните следующую команду:

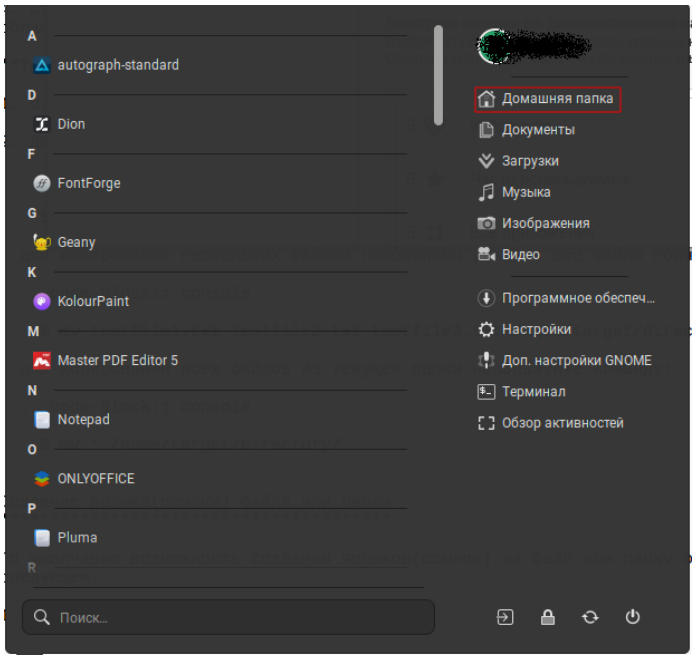
```
$ rm -rf /home/user/directory
```

Создание ярлыка (ссылки) на файл или папку

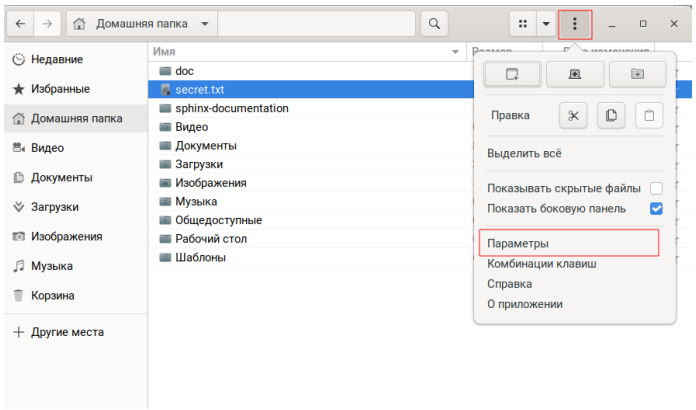
Графический интерфейс

По умолчанию возможность создания ярлыков (ссылок) на файл или папку отключена. Для включения выполните следующие действия.

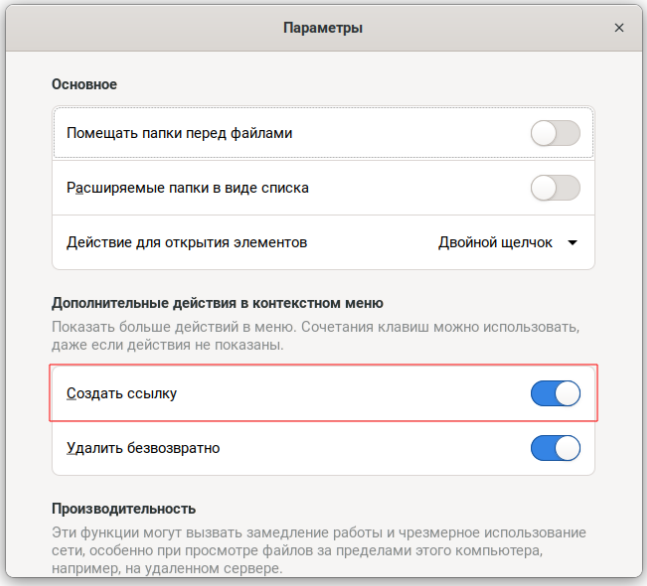
1. Откройте **Домашнюю папку**:



2. Откройте **Настройки** > **Параметры**:



3. Включите опцию **Создать ссылку**:



4. Откройте папку, в которой находится нужный файл или папка.
5. Нажмите правой кнопкой мыши на нужный файл или папку и нажмите **Создать ссылку** или нажмите комбинацию клавиш **CTRL** + **SHIFT** + **M**.

   После этого созданный ярлык можно перенести в требуемое место (например, на **Рабочий стол**), как описано выше.

Терминал

1. Откройте «Терминал».
2. Для создания ссылки (ярлыка) на файл или папку используйте утилиту `ln` со следующим синтаксисом:

```
$ ln -s [путь_целевого_файла] [путь_ссылки]
```

   Например, для создания ссылки на файл `/home/user/testfile.txt` на рабочем столе команда будет выглядеть следующим образом:

```
$ ln -s /home/user/testfile.txt /home/user/'Рабочий стол'/testfile.txt
```

Организация удалённого доступа в МСВСфера

В данном документе описаны инструкции для организации удалённого доступа в ОС МСВСфера:

- SSH
- xRDP

Окружение

- МСВСфера АРМ/Сервер версии 8.x.
- МСВСфера АРМ/Сервер версии 9.x.

SSH

SSH (Secure Shell «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий удалённо подключаться к операционной системе и туннелировать TCP-соединения. Весь трафик при этом шифруется. **SSH** работает по модели «клиент — сервер». В данном случае есть две стороны: «клиент» — устройство, с которого производится удалённое подключение, и «сервер» — удалённое устройство, к которому подключается клиент.

По умолчанию, «сервер» и «клиент» **SSH** установлены в ОС МСВСфера.

Настройка

Для того, чтобы включить «сервер», используется команда:

```
$ sudo systemctl enable sshd --now
```

Для остановки и запуска «сервера» используются команды:

```
$ sudo systemctl stop sshd
$ sudo systemctl start sshd
```

Основной файл конфигурации SSH-сервера — файл `/etc/ssh/sshd_config`, доступный для чтения или редактирования только суперпользователю. После каждого изменения этого файла необходимо перезапустить ssh-сервер для применения изменений командой:

```
$ sudo systemctl restart sshd
```

Основные директивы, используемые в конфигурационном файле «сервера»:

- `Port` — определяет порт, используемый для подключения. По умолчанию используется порт `22`.
- `PasswordAuthentication` — включает или отключает возможность аутентификации по паролю. По умолчанию включено.
- `PubkeyAuthentication` — включает или отключает возможность аутентификации по публичному ключу. По умолчанию включено.
- `PermitRootLogin` — позволяет или запрещает авторизацию под пользователем `root`. Поведение по умолчанию определяется в процессе установки ОС.
- `ListenAddress` — IP-адрес, на котором «сервер» ожидает подключения. По умолчанию, `0.0.0.0/0`, т.е. подключение возможно на всех IP-адресах, настроенных в системе.
- `AddressFamily` — определяет семейство сетевых адресов (IPv4 или IPv6), разрешённых для использования. По умолчанию разрешены оба.
- `AllowUsers` — определяет имена пользователей, которым разрешено подключение. По умолчанию разрешено любому пользователю.
- `AllowGroups` — определяет имена групп, пользователям которых разрешено подключение. По умолчанию разрешено пользователям любых групп.

Более подробное описание директив конфигурационного файла можно найти в руководстве:

```
$ sudo man sshd_config
```

Например, если требуется, чтобы использовался порт `2333`, авторизация пользователя `root` была запрещена, и подключение было возможно только через адрес `192.168.0.10`. То в таком случае необходимо прописать следующие директивы в конфигурационном файле `/etc/ssh/sshd_config`:

```
Port 2333
PermitRootLogin no
ListenAddress 192.168.0.10
```

Подключение к SSH-серверу

К SSH-серверу можно подключиться из любой ОС:

- Linux
- Windows
- MacOS

Подключение из систем Linux

Подключение из систем **Linux** возможно из «Терминала». Синтаксис команды:

```
$ ssh [ОПЦИИ] ПОЛЬЗОВАТЕЛЬ@АДРЕС
```

Примеры команд:

```
$ ssh -p 2333 user@192.168.0.10
```

В данном примере пользователь `user` подключается к SSH-серверу с IP-адресом `192.168.0.10` через порт `2333`.

```
$ ssh -i ~/.ssh/id_rsa user@192.168.0.10
```

В данном примере, пользователь `user` подключается с SSH-серверу с IP-адресом `192.168.0.10` через порт по умолчанию (`22`) с помощью приватного SSH-ключа `~/.ssh/id_rsa`. Для успешного подключения необходимо, чтобы публичный SSH-ключ был добавлен на удалённом SSH-сервере.

Подключение из систем Windows

Подключение к SSH-серверу из **Windows** возможно из командной строки с использованием встроенного SSH-клиента. Синтаксис команд точно такой же как и в системах **Linux**.

Также для подключения к удалённому SSH-серверу из систем Windows можно использовать специальные приложения. В сети Интернет есть множество приложений, позволяющих устанавливать SSH-соединение. Вот наиболее распространённые из них:

- PuTTY

  Классический, бесплатный SSH-клиент, поддерживающий разные протоколы.

- WinSCP

  Бесплатный клиент, специализирующийся на передаче файлов по протоколам SFTP, SCP и FTP, также поддерживает SSH.

- MobaXterm

  Многофункциональный терминал, который включает в себя SSH, FTP, RDP, X11, VNC и другие инструменты. Есть платная версия, в которой, в отличие от бесплатной, можно сохранить бесконечное число сессий.

Подключение из систем MacOS

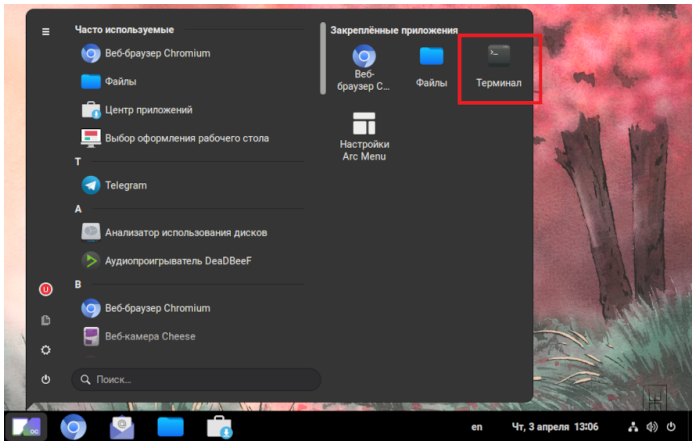
Подключение возможно из «Терминала». Синтаксис команд такой же, как и в случае с системами **Linux**.

xRDP

xRDP — это сервер протокола **Microsoft Remote Desktop Protocol (RDP)** с открытым исходным кодом. Он обеспечивает графический вход на удалённые машины через **RDP**, что важно для операционных систем, отличных от **Microsoft Windows**, например **Linux** и **MacOS**.

Установка

1. Откройте «Терминал»:



2. Добавьте **Epel**-репозиторий:

```
$ sudo dnf install epel-release -y
```

3. Установите **xRDP**:

```
$ sudo dnf install xrdp
```

4. Включите сервис и запустите **xRDP**-сервер:

```
$ sudo systemctl enable xrdp --now
```

5. Откройте порт `3389` в фаерволе:

```
$ sudo firewall-cmd --permanent --add-port=3389/tcp && sudo firewall-cmd --reload
```

После этого можно подключаться к серверу по **RDP**-протоколу.

Подключение к xRDP-серверу

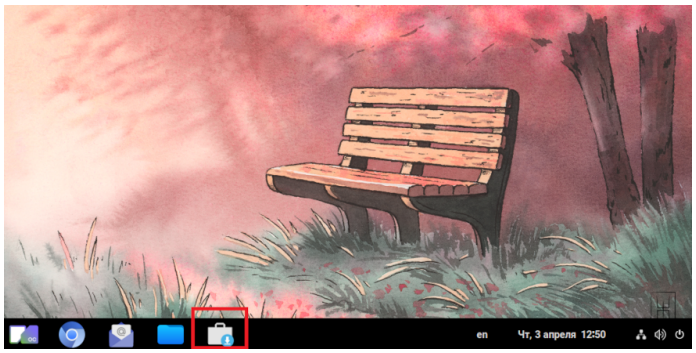
Также как и в случае SSH к xRDP-серверу можно подключится из любой ОС:

- Linux
- Windows
- MacOS

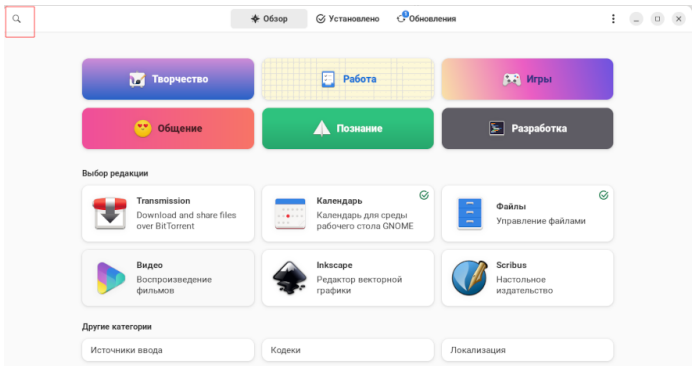
Подключение из систем Linux

Для подключения по протоколу **RDP** из систем Linux используется специальный клиент **Remmina**. Установка клиента в ОС МСВСфера производится следующим образом.

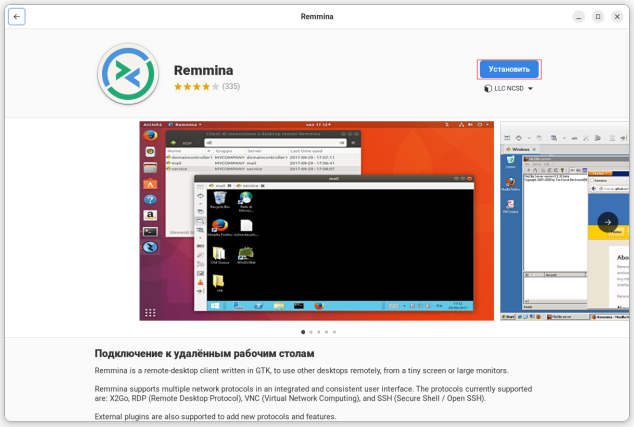
1. Откройте **Центр Приложений**.



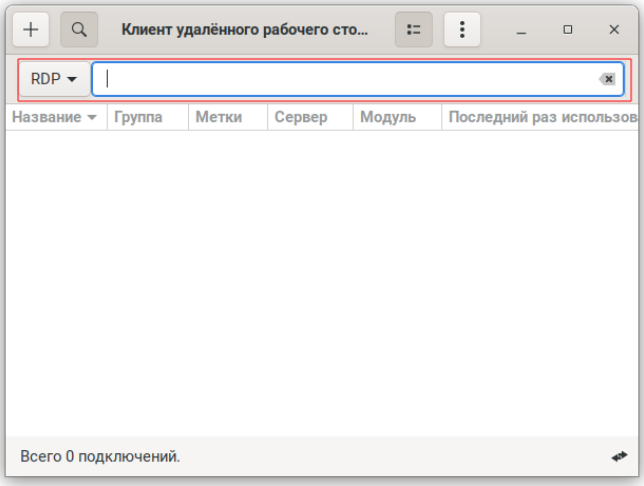
2. Нажмите кнопку поиска в левом верхнем углу.



3. Наберите в поиске `remmina` и нажмите на соответствующий результат.
4. Нажмите **Установить**:



Для подключения к удалённому серверу **xRDP** откройте приложение **Remmina**, в поле RDP укажите IP-адрес и имя хоста удалённого **xRDP**-сервера. Затем нажмите **ENTER**:



Подключение из систем Windows

Для подключения к удалённому xRDP-серверу из Windows можно использовать встроенную утилиту **Remote Desktop Connection**.

Подробную инструкцию по использованию **Remote Desktop Connection** вы можете найти на официальном сайте Microsoft.

Подключение из систем MacOS

Для подключения к удалённому xRDP-серверу из MacOS можно использовать приложение **Microsoft Remote Desktop**.

Подробную инструкцию по установке приложения вы можете найти здесь.

Добавление принтера с помощью сервиса CUPS в ОС МСВСфера

CUPS (Common UNIX Printing System) — сервер печати для UNIX-подобных операционных систем, включая Linux. Это модульная система, которая позволяет компьютеру выступать в роли сервера печати. Управление CUPS осуществляется через веб-интерфейс.

Окружение

- МСВСфера АРМ/Сервер версии 8.x.
- МСВСфера АРМ/Сервер версии 9.x.

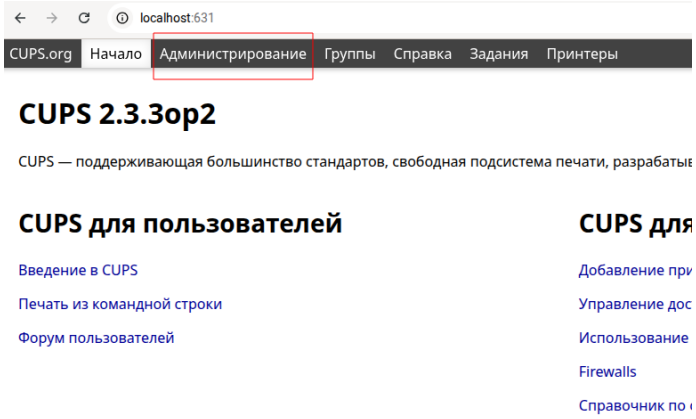
Установка

В ОС МСВСфера сервер **CUPS** установлен по умолчанию.

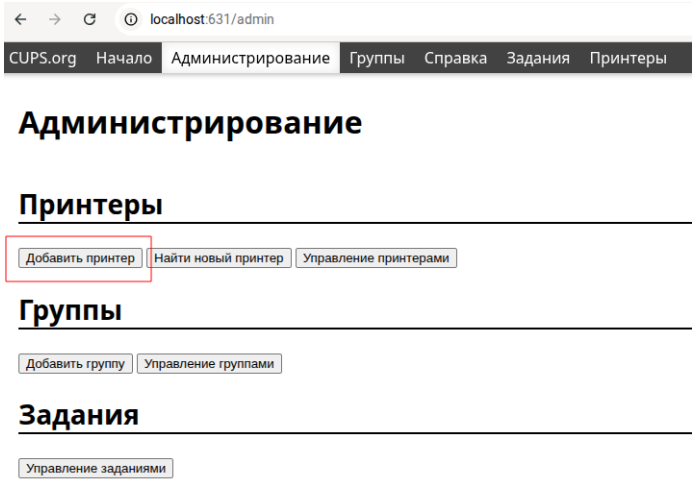
Панель управления доступна в браузере по адресу `http://localhost:631`. Для авторизации используется имя/пароль пользователя с правами администратора (`su`).

Добавление принтера

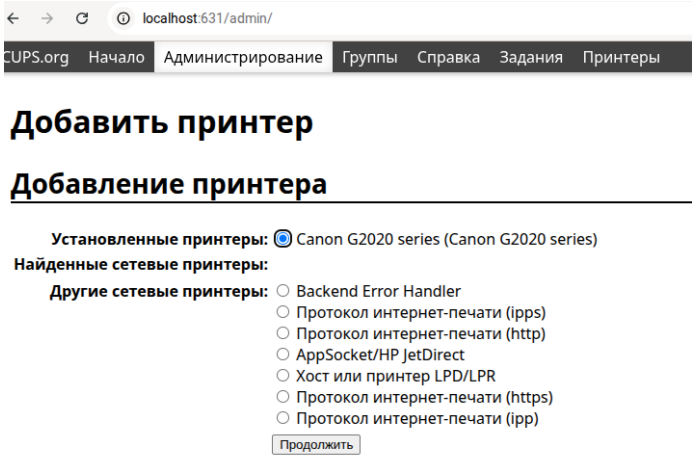
1. Авторизуйтесь в CUPS и перейдите во вкладку «**Администрирование**».



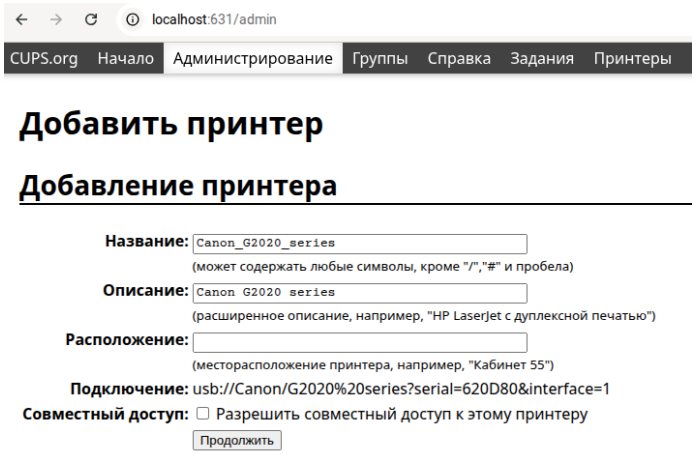
2. Нажмите на кнопку «**Добавить принтер**».



3. Выберите принтер и нажмите «**Продолжить**».



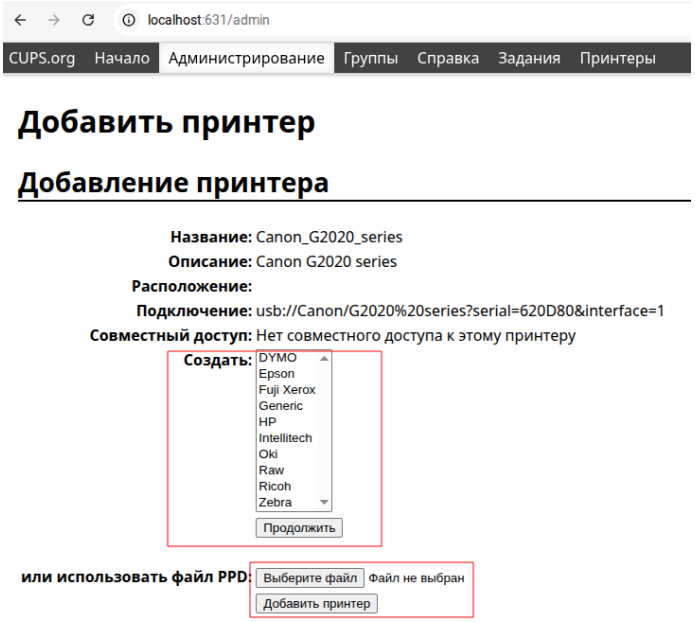
4. При необходимости заполните поля: «**Название**», «**Описание**», «**Расположение**». При необходимости включите опцию «**Разрешить совместный доступ к этому принтеру**» и нажмите «**Продолжить**».



5. Если имеется драйвер принтера в виде файла `PPD`, выберите файл `PPD` (драйвер принтера) и нажмите «**Добавить принтер**».

Как правило, файл драйвера можно скачать на официальном сайте производителя принтера.

Если же файл драйвера отсутствует, то вы можете выбрать драйвер из списка имеющихся драйверов, затем нажмите «**Продолжить**».



Теперь вы можете использовать принтер по назначению.

Файловые менеджеры в МСВСфера

В данном документе рассмотрены аналоги файлового менеджера **Total commander**, доступные в ОС МСВСФера:

- **Midnight commander**
- **Double commander**

Окружение

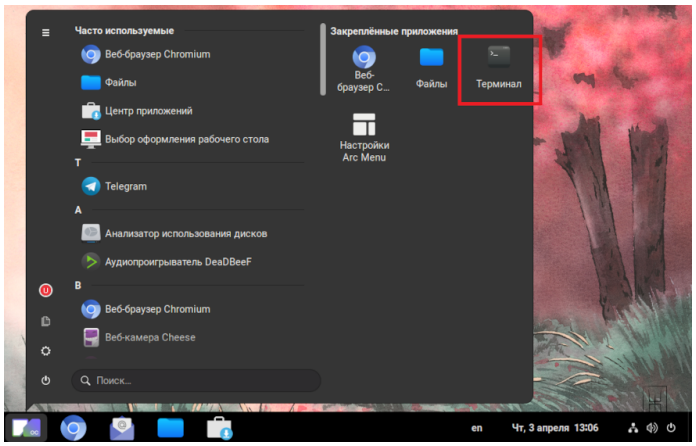
- МСВСфера АРМ/Сервер версии 8.x.
- МСВСфера АРМ/Сервер версии 9.x.
- МСВСфера АРМ/Сервер версии 10.x.

Midnight commander

Midnight Commander (**mc**, **MC**) – текстовая полнофункциональная программа, которая позволяет пользователю копировать, перемещать и удалять файлы и директории, производить поиск файлов и запускать на выполнение команды оболочки. Также в него встроен редактор и программа для просмотра файлов.

Установка

1. Откройте «Терминал»:



2. Выполните команду:

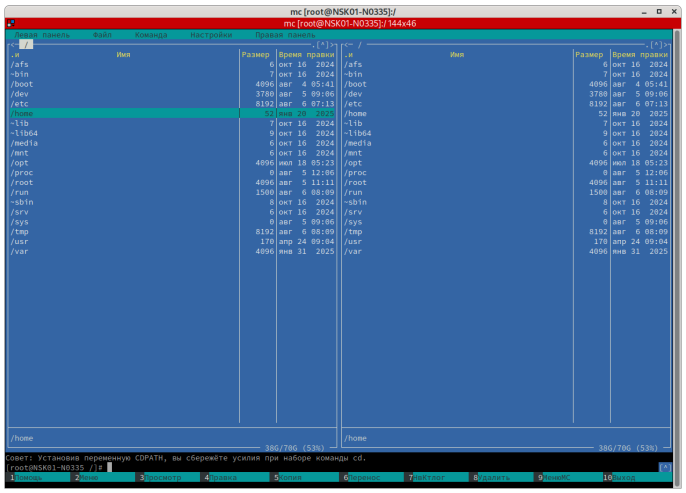
```
$ sudo dnf install mc
```

Запуск и использование

Для запуска **MC** необходимо выполнить команду:

```
$ mc
```

Интерфейс **MC** выглядит следующим образом:



Экран **MC** делится на несколько частей. Самая верхняя строка – строка основного меню программы. Она может быть не видна, но при нажатии клавиши `F9` или при щелчке мыши эта строка отображается в верхней части экрана.

В центральной части расположены левая и правая панели. Внизу находится строка подсказки и полезных советов.

Вторая строка снизу является командной, а в самой нижней части отображается строка с назначением функциональных клавиш.

MC поддерживает управление мышью. При щелчке левой кнопкой мыши на файле, этот файл становится текущим. При щелчке правой кнопкой – файл помечается (или метка снимается, в зависимости от предыдущего состояния).

При двойном щелчке на файле, будет произведена попытка его запуска, если это запускаемый файл. Если расширение файла связано с какой-либо программой, то эта программа будет запущена.

Также можно выполнять команды, связанные с метками функциональных клавиш, щёлкнув на них мышью.

При щелчке мышью на верхней границе панели, список файлов прокрутится на одну страницу назад. Соответственно, щелчок на нижней границе приведёт к прокрутке на страницу вперёд. Этот метод также работает с просмотром помощи и деревом каталогов.

По умолчанию значение параметра автоповтора для клавиш мыши равно 400 миллисекундам. Это значение может быть изменено путём редактирования параметра `mouse_repeat_rate` в файле `.mc.ini`.

Если **MC** сконфигурирован с поддержкой мыши, то вы можете вырезать и склеивать фрагменты текста, используя клавишу **Shift**.

В остальном управление аналогично управлению в **Total Commander** и интуитивно понятно.

Горячие клавиши и комбинации клавиш в Midnight commander

Ниже представлены наиболее распространённые горячие клавиши и комбинации клавиш, доступные в **MC**.

Таблица 9: **Горячие клавиши и сочетания клавиш**

Клавиши/комбинацы клавиш	Описание
`F3`	Просмотр файла
`F4`	Редактирование файла
`Shift+F4`	Создание нового файла
`+`	Вызов меню для выделения группы (только файлы, с учётом регистра, с метасимволами shell)
`\`	Снятие выделения с группы
`Alt+Enter`	Вставка файла или каталога, на котором установлен курсор, в командную строку
`Alt+p`	Возврат предыдущей выполненной команды в командную строку
`Ctrl + x,t`	Вставка всех выделенных объектов в командную строку
`Ctrl + x, Ctrl + p`	Вставка в командную строку полного пути, установленного на неактивной панели
`Alt+t`	Смена режима отображения панели
`Alt+.`	Скрытие или показ скрытых файлов и каталогов, которые начинаются на точку
`Alt+a`	Вставка в командную строку полного пути, установленного на активной панели
`Alt+c`	Смена текущего каталога
`Alt+h`	Просмотр последних выполненных команд
`Alt+i`	Смена каталога на неактивной панели на каталог, в котором находится пользователь
`Alt+o`	Переход неактивной панели в каталог, на который указывает активная панель
`Ctrl+x, q`	Просмотр содержимого файла на второй панели
`Ctrl+x,i`	Просмотр информации о текущем объекте на второй панели
`Ctrl+o`	Скрытие или показ панели
`Ctrl+r`	Перечитывание содержимого каталога
`Ctrl+Space`	Показ размера текущего каталога
`Alt+Shift+?`	Расширенный поиск
`Ctrl+s`	Поиск файла или каталога
`Ctrl+x,o`	Просмотр или изменения владельца, группы
`Ctrl+x,c`	Просмотр или изменение прав доступа
`Ctrl+x,l`	Создание жёсткой ссылки
`Ctrl+x,s`	Создание символической ссылки (symlink)
`Ctrl+x,Ctrl+s`	Редактирование символической ссылки

Горячие комбинации клавиш для редактора Midnight Commander (MCEdit)

Таблица 10: **Горячие клавиши и сочетания клавиш для редактора**

Клавиши/комбинац клавиш	Описание
`Alt+n`	Отображение или скрытие нумерации строк
`Ctrl+s`	Включение/выключение подсветки синтаксиса
`Alt+Shift+-`	Включение/выключение отображения табуляций и пробелов
`Alt+Shift++`	Включение/выключение автовыравнивания возвратом каретки
`F3`	Выделение текста, повторное нажатие `F3` закончит выделение
`Shift+F3`	Выделение блока текста, повторное нажатие `F3` закончит выделение
`F5`	Копирование выделенного текста
`F6`	Перемещение выделенного текста
`F8`	Удаление выделенного текста
`Ctrl+k`	Удаление части строки до конца строки
`Ctrl+y`	Удаление строки
`Ctrl+x`	Переход в конец слова
`Ctrl+z`	Переход на начало предыдущего слова
`Alt+l`	Переход на строку по номеру строки
`Alt+u`	Выполнение команды и вставка вывода результата команды в позицию под курсором
`Alt+t`	Сортировка строк выделенного текста
`Tab`	Сдвиг вправо выделенного текста
`Ctrl+u`	Отмена действия
`Ctrl+n`	Создание нового файла
`Ctrl+f`	Копирование выделенного текста в файл
`Shift+F5`	Вставка текста из файла

Double Commander

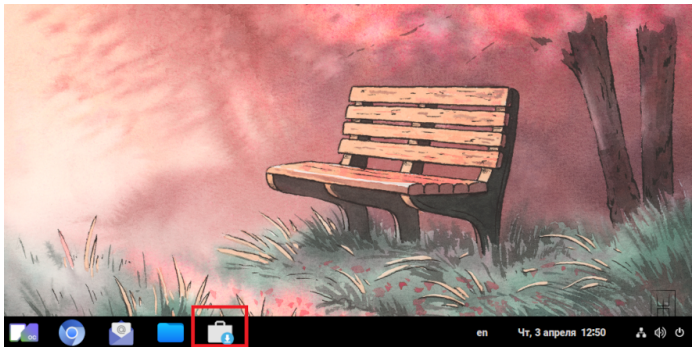
Double Commander – двухпанельный кроссплатформенный файловый менеджер с открытым исходным кодом. Аналогичен по функциональности Total Commander и совместим с его плагинами.

Double Commander работает под ОС Windows, macOS, FreeBSD, а также GNU/Linux с графическими интерфейсами **GTK** и **Qt**, выбор между которыми нужно производить в зависимости от используемой графической оболочки (среды рабочего стола).

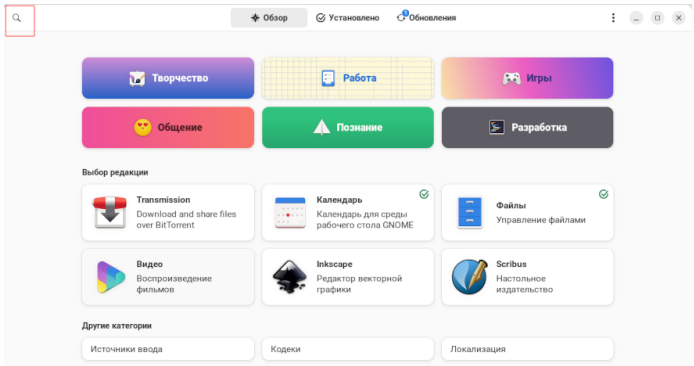
МСВСфера поддерживает оба интерфейса, поэтому данные требования необязательны.

Установка

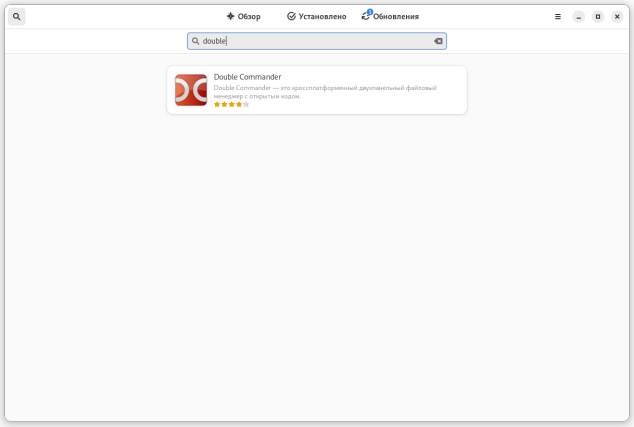
1. Откройте «Центр Приложений».



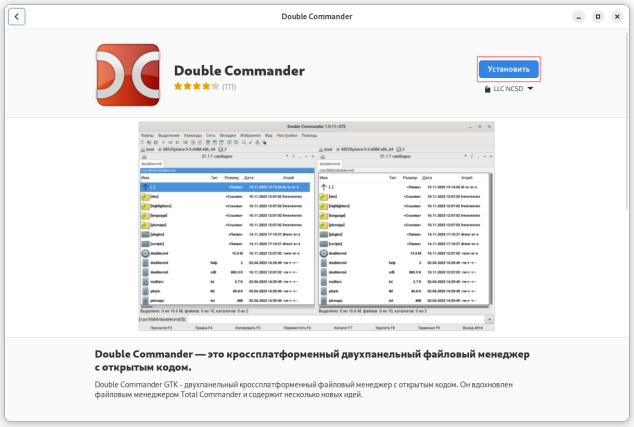
2. Нажмите кнопку поиска в левом верхнем углу.



3. Начните набирать «double» и нажмите на «**Double Commander**».



4. Нажмите «**Установить**».



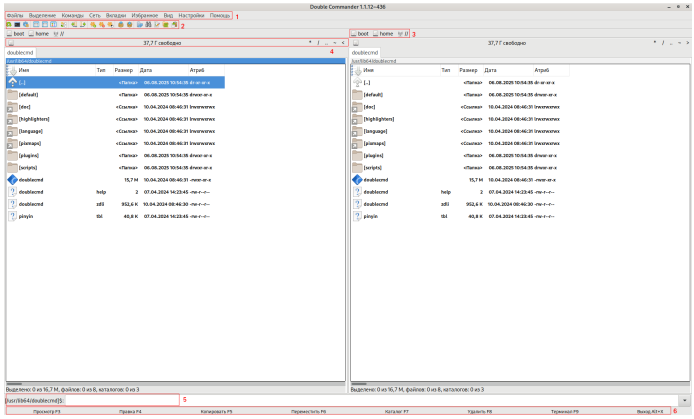
Работа в Double commander

Основные характеристики **Double Commander**:

- встроенный редактор с подсветкой синтаксиса (`F4`);
- встроенный просмотрщик файлов (`F3`) для просмотра файлов любых размеров в шестнадцатеричном, двоичном или текстовом формате;
- работа с архивами так же, как с подкаталогами. Доступно копирование файлов в архив и из него. Поддерживаются следующие типы архивов: **ZIP, TAR, GZ, BZ2, XZ, LZMA, 7Z**, а также **RPM, CPIO, DEB, RAR, ZIPX**;

- поддержка **Unicode**;
- поддержка консольных архиваторов, аналогично плагину **MultiArc** для **Total Commander**, со схожим форматом файла настроек;
- поддержка вкладок файловых панелей;
- настраиваемые колонки;
- инструмент синхронизации каталогов;
- инструмент группового переименования;
- протоколирование файловых операций;
- настраиваемая панель инструментов с кнопками для запуска внешних программ или внутренних команд меню;
- поддержка **WCX**, **WDX**, **WFX** и **WLX** плагинов от **Total Commander**.

Интерфейс Double commander выглядит следующим образом:



Главное меню (1) включает в себя следующие пункты:

- Файлы.
- Выделение.
- Команды.
- Сеть.
- Вкладки.
- Избранное.
- Вид.
- Настройки.

- Помощь.

Панель инструментов (2) с функциональными кнопками для быстрой и удобной работы с содержимым и настройками самой программы. Добавлять и удалять необходимые функциональные кнопки можно через пункт меню «**Настройки**» → «**Параметры**» → «**Панель инструментов**».

Кнопки **быстрого перехода** к разделам (3).

Над файловыми панелями доступны следующие действия с каталогами (4):

- список разделов;
- переход в корень файловой системы (`/`);
- переход в домашний каталог (`~`);
- управление избранными каталогами (`*`);
- переход на уровень выше (`..`);
- дублирование панелей (`>` и `<`);

Ниже располагается строка-заголовок с полным адресом текущего каталога. Данная строка также выполняет роль быстрых ссылок: с её помощью можно простым кликом перейти в любой из родительских каталогов.

Внизу окна Double Commander расположена «**Командная строка**» (5), используемая для ввода команд вручную, а также панель функциональных клавиш **F3** — **F9** и **Выход** — **Alt+X** (6).

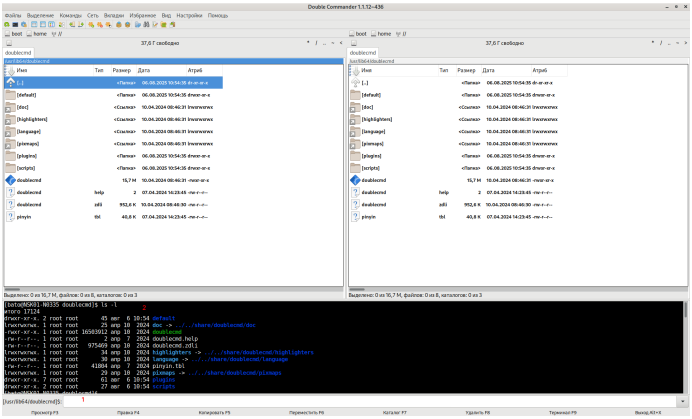
Для отображения результатов выполнения набранных команд из «**Командной строки**», а также действий в файловых панелях, во встроенном окне терминала **Double Commander** необходимо выставить флаг в чекбоксе «**Окно терминала**» в пункте меню «**Настройки**» → «**Параметры**» → «**Вид окна**».

По умолчанию используется **xterm**, но в системе он не установлен по умолчанию. Для установки выполните следующую команду в «Терминале» (с правами пользователя `root`):

```
$ sudo dnf install xterm
```

Изменить терминал, используемый по умолчанию, можно в пункте меню «**Настройки**» → «**Параметры**» → «**Инструменты**» → «**Терминал**», в полях «**Команда**».

Для передачи команды в терминал выполните её в текстовом поле (1), результат выполнения будет отображён в поле терминала (2). Команды выполняются в активной файловой панели.



Избранные каталоги

Избранные каталоги («*» или **Ctrl+D**) – это список каталогов и инструменты его изменения для формирования требуемого перечня быстрого доступа. Данный функционал также доступен из пункта меню «**Команды**» → «**Избранные каталоги**».

При активации нажатием на кнопку «*» или вызовом сочетания клавиш **Ctrl+D** появляется контекстное меню.

В списке будут отображены добавленные каталоги, ниже представлены пункты добавления текущего или выделенного каталога в перечень. Для того, чтобы добавить нужный каталог в избранные, необходимо выделить его курсором мыши, нажать «**Space**», вызвать функцию «**Избранные каталоги**»» одним из перечисленных выше способов и нажать «**Добавить выделенный: <путь_к_каталогу>**».

Редактирование разделов доступно через это же контекстное меню в пункте «**Настройка избранных каталогов**». Там можно добавить, удалить, экспортировать, импортировать, создать резервную копию и восстановить из резервной копии нужные каталоги.

Файлы

Пункт меню «**Файлы**»» в **Double Commander** позволяет:

- создавать символические или жёсткие ссылки на файлы;
- изменять атрибуты файлов;
- создавать каталоги;
- распаковывать / запаковывать файлы и каталоги;
- разрезать файлы или объединять их;

- подсчитывать и проверять контрольные суммы;
- сравнивать файлы по содержимому;
- применять групповое переименование.

Инструмент группового переименования позволяет присвоить нескольким файлам разные наименования с одной маской (например: «picture 1» , «picture 2», «picture 3» и т.д.).

В пункте меню «**Выделение**»» можно быстро выделить файлы или папки, сгруппировав файлы по расширению или выделив группу.

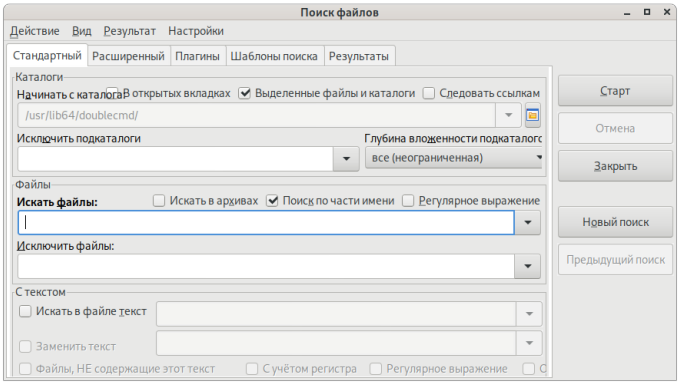
Пункт «**Сохранить выделение**» сохраняет выделенный список в буфер обмена.

Пункт «**Восстановить выделение**» восстанавливает ранее сохранённый выделенный список из буфера обмена.

Double Commander даёт возможность сохранить наименования выделенных файлов в текстовый документ, с возможностью сохранения как полных имён с абсолютным путём к файлам, так и сокращённых.

Поиск

Широкие возможности поиска в **Double Commander** представлены в виде отдельно открывающегося окна с вкладками. Поиск доступен из пункта меню «**Команды**» → «**Поиск...**».



Параметры места поиска:

- **В открытых вкладках** – поиск будет осуществляться в открытых вкладках;
- **Выбранные файлы и каталоги** – поиск ведётся только среди выделенных файлов или каталогов активной панели;
- **Следовать ссылкам** – если будет найдена символьная ссылка на каталог, то целевой каталог будет прочитан и использован, иначе эта символьная ссылка будет считаться обычным файлом.

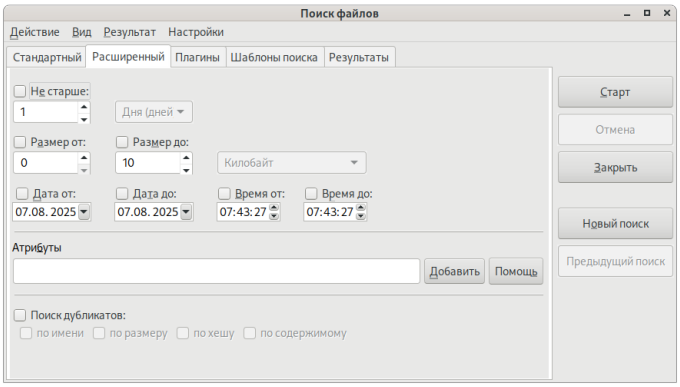
По умолчанию в строке «**Начинать с каталога**» используется текущий каталог активной панели. Также вы можете исключить подкаталоги и указать глубину вложенности.

В строке «**Искать файлы**» задаётся маска, по которой будет произведён поиск.

Дополнительные параметры этого поиска: «**Искать в архивах**», «**Поиск по части имени**», «**Регулярное выражение**», строка «**Исключить файлы**».

Если отметить «**Искать в файле текст**», то во время поиска будет происходить просмотр содержимого файлов, а в результат попадут только файлы, содержащие запрашиваемый текст.

Вкладка «**Расширенный**» позволяет указать дополнительные условия поиска файла: дату, время и возраст файла (по дате модификации), минимальный и максимальный размер файла, а также атрибуты.



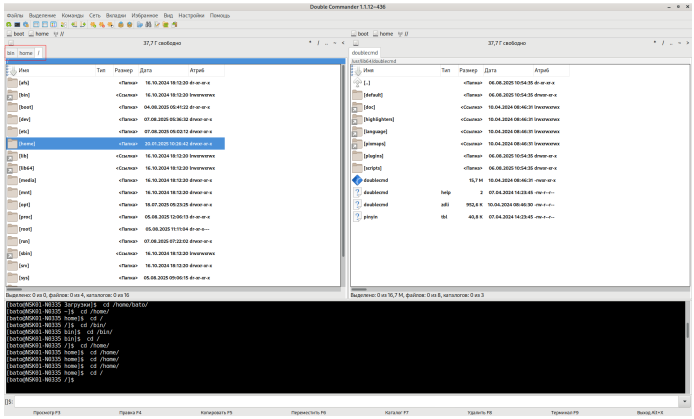
Вкладка «**Плагины**» задаёт поиск с помощью информационных (контентных, **WDX**) или поисковых (**DSX**) плагинов.

Вкладка «**Шаблоны поиска**» сохраняет параметры поиска для повторного использования.

Вкладки

Меню «**Вкладки**» предназначено для создания дополнительных вкладок в панелях и управления ими: создания, загрузки из файла, переименования. Во вложенном разделе «**Опции вкладки**» изменяется состояние вкладок.

Вкладку можно создать с помощью сочетания клавиш **Ctrl+T** или через пункт меню «**Вкладки**».



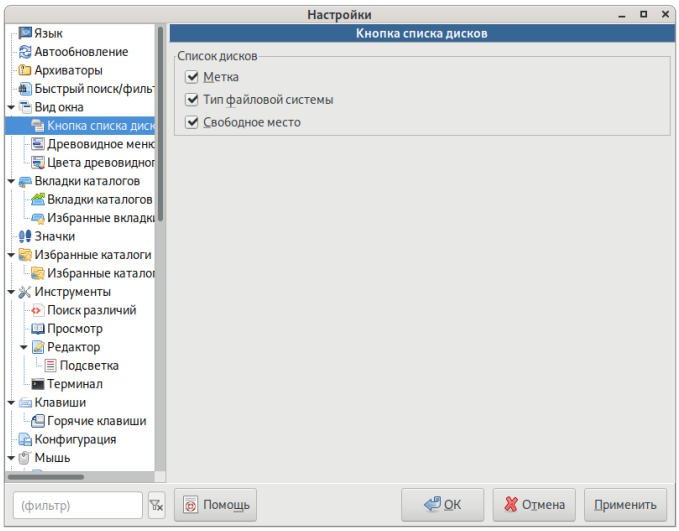
Подменю «**Опции вкладки**» позволяет назначить вкладку обычной или заблокированной, причём у заблокированной может быть несколько статусов:

- **Заблокировать вкладку** – блокирует смену каталога вкладки и предлагает при этом открыть этот же каталог в новой вкладке.
- **Заблокировать с возможностью смены каталога** – в этом случае можно сменить каталог, однако, если переключиться на другую вкладку, а потом обратно, в ней будет открыт каталог, с которым она была заблокирована;
- **Заблокировать и открывать каталоги в новых вкладках** – при попытке войти в любой другой каталог в панели, он откроется в новой вкладке.

Заблокированные вкладки помечаются значком «*».

Настройка параметров

Диалоговое окно, доступное через пункт меню «**Настройки**» → «**Параметры**», позволяет устанавливать и выбирать нужные пользователю параметры программы, такие как «**Вид окна**», «**Панель инструментов**», «**Шрифты**»» и пр..



Мониторинг производительности системы с помощью утилиты atop

Atop — популярный интерактивный монитор производительности. Является аналогом утилиты **top**, отличие состоит в том, что в **atop** выводятся только новые изменения об активных системных процессах, однако при этом есть функция ведения логов.

Окружение

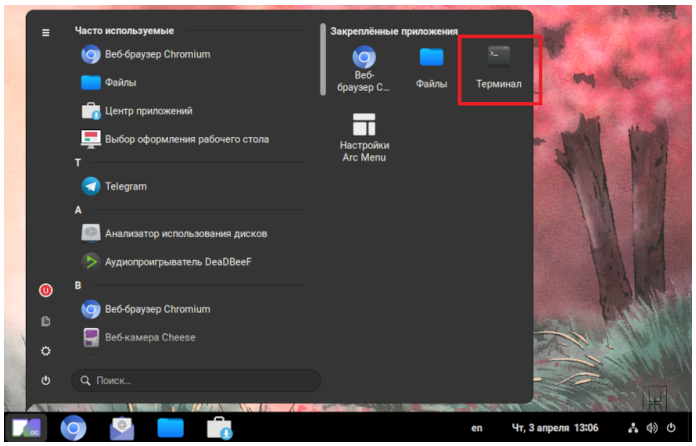
- МСВСфера АРМ/Сервер версии 8.x.
- МСВСфера АРМ/Сервер версии 9.x.
- МСВСфера АРМ/Сервер версии 10.x.

Установка и запуск atop

Примечание

Утилита **atop** доступна в репозитории **EPEL**. Поэтому для установки утилиты необходимо добавить *упомянутый репозиторий*.

1. Откройте «Терминал»:



2. Выполните команду:

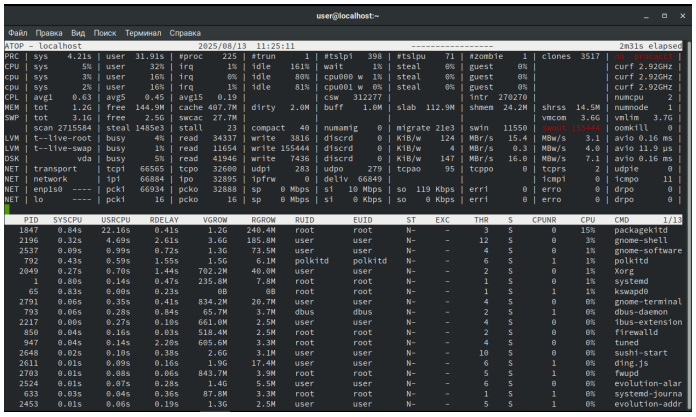
```
$ sudo dnf install atop
```

Для запуска утилиты используйте команду:

```
$ atop
```

Описание интерфейса atop

Интерфейс **atop** выглядит как показано на снимке экрана:



Статистика процессов

Верхняя строка **PRC** отображает статистику процессов:

- `sys` — время, затраченное процессами в режиме ядра;
- `user` — время, затраченное процессами в режиме пользователя;
- `proc` — общее количество процессов в системе;
- `trun` — количество процессов, находящихся в состоянии выполнения;
- `tslpi` — количество процессов, ожидающих ресурсов в спящем режиме;
- `tslpu` — количество процессов, ожидающих ресурсов в интерактивном режиме;
- `tidle` — количество процессов, находящихся в состоянии простоя;
- `zombie` — количество процессов-зомби;
- `clones` — количество процессов, созданных с момента последнего замера;
- `exit` — количество завершённых процессов.

Мониторинг загруженности центрального процессора компьютера/сервера

Строка **CPU** отображает статистику производительности всего процессора. В случае, когда в компьютере/сервере используется многоядерный процессор, в вывод добавляются строки **cpu**, которые отображают статистику производительности каждого отдельного ядра.

- `sys` — использование процессора в режиме ядра (системные задачи);
- `user` — использование процессора в режиме пользователя;
- `irq` — использование процессора на обработку прерываний;
- `idle` — время простоя процессора;
- `wait` — время ожидания ввода/вывода;
- `steal` — процессорные затраты гипервизора;
- `ipc` — количество инструкций на такт;
- `cycl` — средняя тактовая частота;
- `curf` — текущая частота процессора.

Строка **CPL** отображает статистику загрузки процессора.

- `numcpu` — количество доступных CPU;

- `avg1/avg5/avg15` — средняя загрузка системы за последние 1, 5 и 15 минут;
- `csw` — количество переключений контекста;
- `intr` — количество аппаратных прерываний.

Мониторинг памяти

Строка **MEM** отображает потребление физической памяти.

- `tot` — общий объём памяти;
- `free` — свободная память;
- `avail` — доступная память;
- `cache` — используемая память под кэш;
- `dirty` — объём данных в памяти, которые должны быть записаны на диск;
- `slab` — объём памяти, используемой под SLAB-кэш;
- `shmem/shrss/shswp` — объём используемой общей памяти (shared memory);
- `pgtab` — используемая память под таблицы страниц.

Строка **SWP** отображает потребление SWAP-памяти (подкачка).

- `tot` — общий объём пространства подкачки (swap);
- `free` — свободный объём пространства подкачки (swap);
- `swcac` — объём памяти, кэшируемой в пространстве подкачки (swap);
- `zswap/zstor` — объём памяти в zswap/zstor;
- `vmcom` — используемая виртуальная память;
- `vmlim` — лимит виртуальной памяти.

Строка **PAG** отображает статистику виртуальной памяти.

- `scan/steal/stall/compact` — данные по работе системы виртуальной памяти (сканирование страниц, перехваты, остановки, компактирование);
- `pgin/pgout` — количество страниц, записанных в память/выгруженных из неё;
- `numamig/migrate` — данные по миграции страниц в NUMA;
- `oomkill` — количество завершений процессов из-за недостатка памяти.

Статистика уровня ресурсного давления на CPU

Метрика **PSI** (Pressure Stall Information) отражает снижение уровня ресурсного давления, что свидетельствует об уменьшении временных интервалов, в течение которых исполняемые процессы испытывали дефицит вычислительных ресурсов центрального процессора.

- `cpusome/memsome/memfull/iosome/iofull` — метрики давления на CPU, память и ввод/вывод.

Мониторинг устройств хранения данных

Строка **LVM** показывает статистику использования логических томов. Строка **DSK** отображает статистику физических дисков.

- `busy` — уровень загрузки тома;
- `read/write` — количество операций чтения/записи;
- `discrd` — количество операций сброса данных;
- `KiB/r/KiB/w` — средний размер операций чтения/записи в килобайтах;
- `MBr/s/MBw/s` — скорость чтения/записи;
- `avio` — среднее время ввода/вывода.

Сетевая статистика

Строка **NET** отображает статистику сетевой активности.

- `tcpi/tcpo/udpi/udpo` — статистика по TCP/UDP трафику;
- `pcki/pcko` — количество входящих/исходящих пакетов;
- `sp` — пропускная способность;
- `si/so` — скорость приёма/передачи данных;
- `erri` — количество ошибок при приёме пакетов;
- `erro` — количество ошибок при передаче пакетов;
- `drpi` — количество сброшенных входящих пакетов;
- `drpo` — количество сброшенных исходящих пакетов;

Анализ статистики atop

На что обратить внимание при анализе данных:

- **CPU**
  - высокий `sys` и `irq`, но низкий `idle` указывают на системные проблемы;
  - `wait` $> 0\%$ может свидетельствовать о задержках ввода/вывода.
- **MEM**
  - высокий `dirty` и низкий `free` указывают на потенциальные задержки при записи данных на диск;
  - проверить использование кэша и slab-кэша.
- **SWP**
  - высокое использование свопа (`free` близко к 0) может привести к замедлению системы.
- **PAG**
  - большое количество `scan` и `steal` указывает на перегрузку системы управления памятью.
- **DSK**
  - высокий `busy` ($> 90\%$) и большое время `avio` указывают на проблемы с производительностью дисков.
- **NET**
  - проверить `si` и `so` на наличие узких мест в сети. Высокие `erri` и `erro` указывают на ошибки в сети.
- **PSI**
  - значения `cpusome`, `memsome`, `iosome` выше 10% сигнализируют о перегрузке ресурсов.
- **CPL**
  - средняя нагрузка (`avg1`, `avg5`, `avg15`) выше числа ядер (`numcpu`) указывает на перегрузку.

Каждый раздел требует анализа в контексте специфики нагрузки и ожидаемой производительности.

Дополнительная информация о процессах

Общий вывод команды `atop` содержит дополнительную информацию для каждого запущенного процесса:

- `PID` — идентификатор процесса;
- `RUID` — реальный идентификатор пользователя, от имени которого выполняется процесс;
- `EUID` — эффективный идентификатор пользователя, под которым выполняется процесс;
- `THR` — общее количество потоков для процесса;
- `SYSCPU` — потребление `CPU` за последний интервал в режиме `system` (обычно системные вызовы);
- `USRCPU` — потребление `CPU` за последний интервал в режиме `user`;
- `VGROW` — рост виртуальной памяти в течении последнего интервала;
- `RGROW` — резидентный (реальный, занятый процессом ОП) рост памяти в течении последнего интервала. Может быть отрицательным, если процесс выгружается из памяти в `swap`;
- `RDDSK` — размер переданных данных при чтении с диска;
- `WRDSK` — празмер переданных данных при записи на диск;
- `ST` — текущий статус процесса (`N` в первой позиции говорит, что процесс был запущен во время последнего интервала. Вторая позиция показывает закончился ли процесс за время последнего интервала: `E` — exit, `S` — прекращено, `C` — прекращено но остался дамп ядра (core dump));
- `EXC` — код завершения процесса;
- `S` — состояние процесса (`S=sleep` (ожидает событий), `R=running` (работает), `Z=zombie` (ожидает родительский процесс));
- `CPUNR` — идентифицирует `CPU`, который обрабатывает процесс;
- `CPU` — процент занятости выбранного ресурса;
- `CMD` — имя процесса.

Настройка записи логов atop

Конфигурационный файл для **atop** — `/etc/sysconfig/atop`.

Конфигурация для посуточной записи логов с шагом в 10 минут (параметр `INTERVAL` в секундах) выглядит следующим образом:

```
# sysconfig atop

# Current Day format
CURDAY=`date +%Y%m%d`
# Log files path
LOGPATH=/var/log/atop
# Binaries path
BINPATH=/usr/bin
# PID File
PIDFILE=/var/run/atop.pid
# interval (default 10 minutes)
INTERVAL=600
# Имя файла логов
OUTFILE="$LOGPATH/$CURDAY.log"
```

При необходимости можно изменить шаг записи, изменив параметр `INTERVAL`. Значение указывается в секундах.

После изменения конфигурационного файла необходимо перезапустить **atop**:

```
$ sudo systemctl restart atop
```

Для просмотра дневного лога необходимо выполнить следующую команду в «Терминале»:

```
$ atop -r /var/log/atop/atop_20250809
```

В данном случае, `/var/log/atop/atop_20250809` — путь до нужного файла с логами.

Для навигации по времени в дневном логе используются клавиши:

- `t` — перейти вперед по времени;
- `Shift + t` — перейти назад по времени.

Для остановки сбора логов необходимо выполнить следующую команду:

```
$ sudo systemctl stop atop
```

Более подробную информации об **atop** и его использовании вы можете найти в соответствующем руководстве, выполнив в «Терминале»:

```
$ man atop
```

Редактирование PDF-файлов в МСВСфера

В данной статье будут рассмотрены инструменты для редактирования PDF-файлов, доступные в МСВСфера.

Окружение

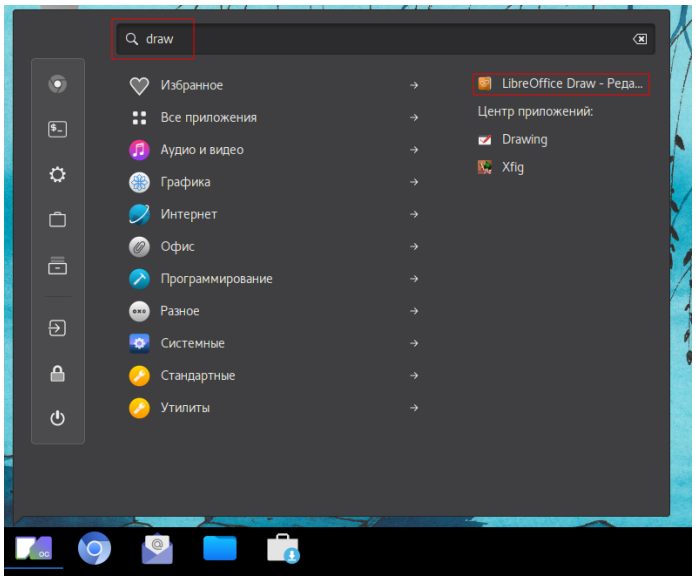
- МСВСфера АРМ/Сервер версии 8.x
- МСВСфера АРМ/Сервер версии 9.x
- МСВСфера АРМ/Сервер версии 10.x

LibreOffice Draw

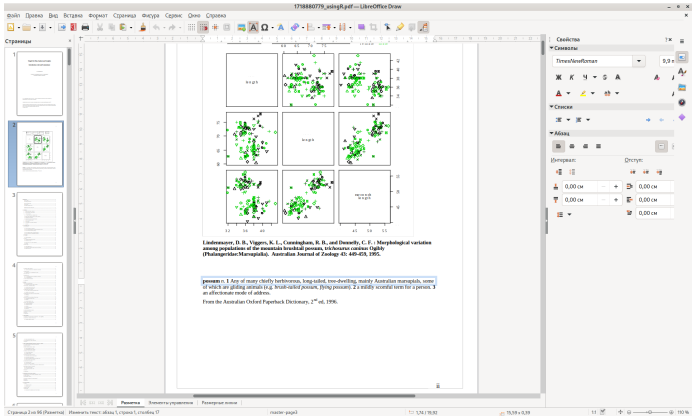
По умолчанию в МСВСфера установлен офисный пакет **LibreOffice**, который включает в себя инструменты для работы с документами. Один из инструментов — **LibreOffice Draw**.

LibreOffice Draw позволяет внести в файл PDF больше изменений, чем просто базовое редактирование страниц.

Для запуска откройте **Главное Меню**, в строке поиска наберите **LibreOffice Draw** и запустите приложение.



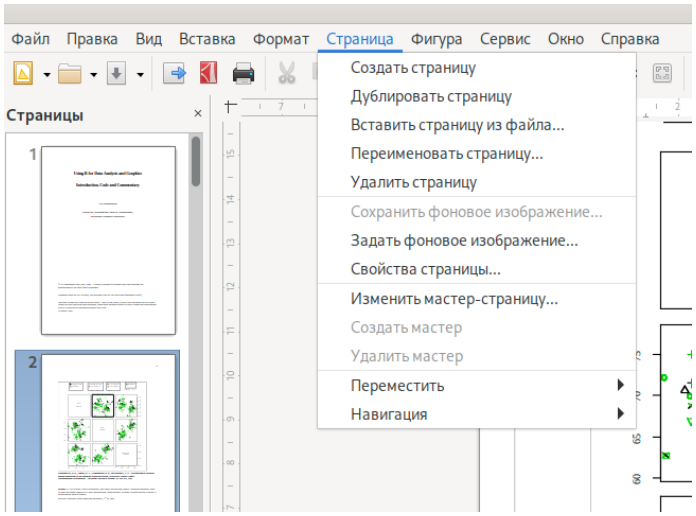
Интерфейс выглядит следующим образом:



В левой части экрана доступна панель рисования и панель для просмотра страниц. В правой части экрана находится панель **Свойства**. В этой панели отображены настройки страниц.

При редактировании текста вы можете изменить его стиль. Для этого существует панель свойств, находящаяся в левой части экрана.

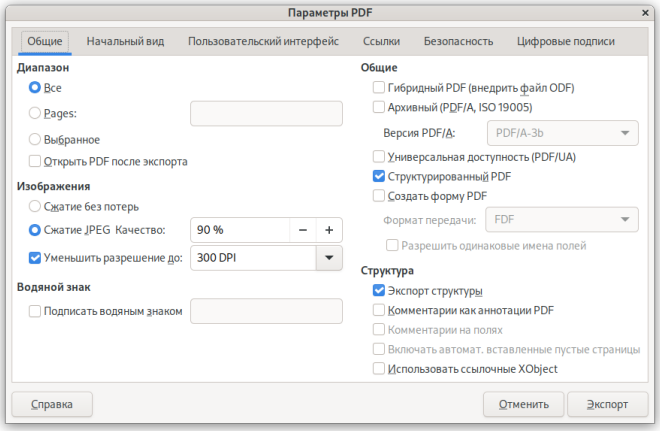
Draw также позволяет добавлять новые страницы или удалять старые. Чтобы добавить или удалить страницу, перейдите в раздел **Страница** в верхней панели.



После завершения редактирования, чтобы сохранить файл в формате `pdf`, перейдите в раздел **Файл → Экспорт в ... → Экспорт в PDF**. После этого откроется окно с параметрами сохранения. В нем вы можете настроить следующие параметры:

- диапазон сохраняемых страниц;
- выполнить сжатие и уменьшение разрешения;
- подпись водяными знаками;
- изменить структуру файла;

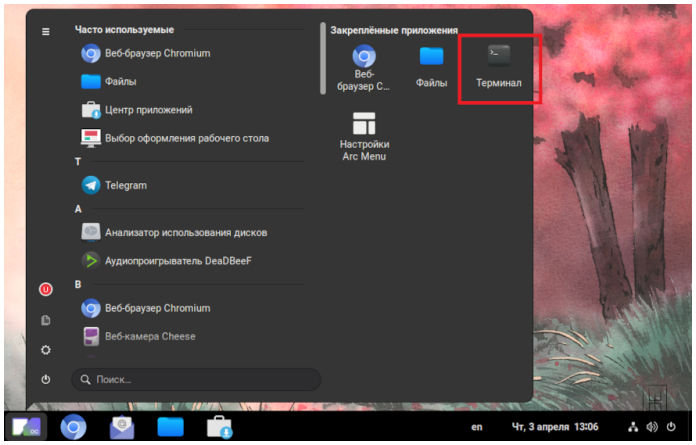
- добавить ссылки;
- добавить шифрование;
- добавить цифровую подпись.



Также **LibreOffice Draw** можно использовать для конвертации PDF в некоторые другие форматы. Для этого откройте ваш документ, перейдите в раздел **Файл** в верхней панели и выберите пункт **Экспорт...**. После этого выберите нужный вам формат. Доступные форматы для конвертации: **HTML, XHTML, SWF, BMP, EMP, EPS, GIF, JPEG, PNG, SVG, TIFF, WMF**. После того как вы выбрали, в какой формат конвертировать ваш документ, появится окно для настройки параметров сохранения.

Конвертация PDF-файлов с помощью Poppler

По умолчанию в МСВСфера установлена утилита **Poppler**, позволяющая конвертировать PDF-файлы в различные форматы. Для использования утилиты **Poppler** откройте «Терминал».



Ниже приведены несколько примеров команд для конвертации.

Преобразование PDF в PS

```
$ pdftocairo -ps example.pdf example.ps
```

Преобразование PDF в PNG

```
$ pdftocairo -png example.pdf example.png
```

Преобразование PDF в HTML

```
$ pdftohtml example.pdf example.html
```

Преобразование PDF в JPEG

```
$ pdftocairo -jpeg -f 2 -l 4 example.pdf  example.jpg
```

Преобразование PDF в PPM

```
$ pdftoppm example.pdf example
```

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм	измененных	замененных	новых	аннулированных					