Утвержден

РУСБ.10205-01-УД

ПК «ВИУ»

Руководство администратора

РУСБ.10205-01 95 01

Листов 54

2016

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

АННОТАЦИЯ

Настоящий документ является руководством администратора программного комплекса виртуализации и управления (ПК «ВИУ») РУСБ.10205-01 (далее по тексту — ПК).

В документе приведено назначение ПК, описание его установки, настройки и порядка применения с учетом особенностей операционной системы, под управлением которой функционирует ПК.

СОДЕРЖАНИЕ

1. Общие сведения 5
1.1. Контролируемые функции 5
1.2. Требования к техническим средствам 9
2. Состав и принципы функционирования 11
2.1. Принципы функционирования 11
2.2. Средства эмуляции аппаратного обеспечения на основе QEMU 15
2.3. Сервер виртуализации `libvirt` 16
2.4. Служба доступа к сетевой защищенной файловой системе `Samba` 16
2.5. Консольный интерфейс управления виртуальными машинами `virsh` 16
2.6. Графический интерфейс управления виртуальными машинами `virt-manager` . 17
2.7. Графический интерфейс для удаленного доступа (VDI) пользователей к виртуальным машинам по протоколам VNC и Spice `virt-viewer` 17
3. Установка и настройка 19
3.1. Установка серверной части ПК 21
3.2. Установка клиентской части ПК 21
3.3. Настройка 21
3.4. Конфигурационный файл службы сервера виртуализации `libvirtd.conf` 23
3.5. Конфигурационный файл средства эмуляции аппаратного обеспечения QEMU . . 23
3.6. Конфигурационные файлы SASL сервера виртуализации 24
4. Управление виртуальными машинами 26
4.1. Создание виртуальных машин 26
4.2. Управление конфигурацией виртуальных машин с помощью графической и консольных утилит 30
4.3. Управление файлами-образов дисков виртуальных машин 31
4.4. Управление виртуальными сетевыми интерфейсами и сетями 32
4.5. Запуск и управление функционированием виртуальных машин 35
4.6. Ограничения по конфигурированию и запуску виртуальных машин в условиях мандатного разграничения доступа 36
5. Отказоустойчивые решения 37
5.1. Миграция виртуальных машин между серверами виртуализации 37
6. Службы 40
6.1. Служба сервера виртуализауии `libvirtd` 40

6.2. Служба предоставления доступа к сетевой защищенной файловой системе Samba 40
7. Утилиты командной строки 41
8. Взаимодействие администратора с СЗИ 42
8.1. Идентификация и аутентификация при доступе к серверу виртуализации libvirt . 42
8.2. Идентификация и аутентификация при доступе к рабочему столу виртуальных машин 44
8.3. Дискреционное управление доступом к виртуальным машинам 45
8.4. Мандатное управление доступом к виртуальным машинам 47
8.5. Режимы запрета модификации файлов-образов виртуальных машин 48
9. Тестирование запуска виртуальной машины 49
Перечень сокращений 53

1. ОБЩИЕ СВЕДЕНИЯ

ПК предназначен для создания защищенной виртуальной среды, обеспечивающей функционирование виртуальных машин и управление ими в операционной системе специального назначения «Astra Linux Special Edition» РУСБ.10015-07 (далее по тексту — ОС СН) в условиях дискреционного и мандатного разграничения доступа.

ПК обеспечивает решение следующих функциональных задач:

– обеспечение создания тонких (терминальных) клиентов с использованием технологии VDI (Virtual Desktop Infrastructure);

– использование аппаратных возможностей архитектуры x86-64 по виртуализации процессоров на основе модуля KVM (Kernel-based Virtual Machine) из состава ОС СН и средств эмуляции аппаратного обеспечения QEMU;

– идентификация и аутентификация пользователя до предоставления доступа к функциям виртуализации и управления ПК, в том числе в режиме взаимодействия со средствами создания единого пространства пользователей (ALD) из состава ОС СН;

– создание виртуальных машин с помощью графической и консольных утилит;

– запуск виртуальной машины в виде отдельного процесса ОС СН, который функционирует от имени учетной записи пользователя с его мандатными атрибутами безопасности;

– предоставление пользователям удаленного доступа к виртуальным машинам в соответствии с дискреционными и мандатными правилами разграничения доступа;

– управление конфигурацией виртуальных машин с помощью графической и консольных утилит;

– взаимодействие между виртуальными машинами по протоколам стека IPv4 в условиях мандатного разграничения доступа;

– взаимодействие между процессами пользователей и виртуальными машинами по протоколам стека IPv4 в условиях мандатного разграничения доступа;

– маршрутизация сетевых пакетов виртуальных машин;

– возможность защиты файлов-образов виртуальных машин от модификации в процессе функционирования виртуальных машин.

1.1. Контролируемые функции

ПК функционирует под управлением ОС СН и совместно с ней обеспечивает реализацию следующих функций по защите информации от НСД:

1) Идентификация и аутентификация пользователей.

Первоначальная идентификация и аутентификация пользователя обеспечивается

средствами ОС СН при входе пользователя в систему. В дальнейшем все субъекты доступа (процессы) ПК при доступе к серверу виртуализации проходят обязательные процедуры идентификации и аутентификации. Следует различать доступ к серверу виртуализации libvirt для управления виртуальными машинами и доступ непосредственно к рабочему столу конкретной виртуальной машины по протоколам VNC и Spice, причем параметры аутентификации для этих взаимодействий настраиваются отдельно.

В качестве способа аутентификации при использовании ПК в условиях ЕПП применяется аутентификация Kerberos с помощью механизма SASL gssapi, в иных случаях возможно применение других механизмов SASL или SSL/TLS аутентификации, основанной на сертификатах.

Описание идентификации и аутентификации при доступе к серверу виртуализации libvirt и рабочим столам виртуальных машин приведены в разделах 8.1 и 8.2, соответственно.

2) Дискреционный принцип контроля доступа.

В ПК реализован дискреционный принцип контроля доступа к управлению виртуальными машинами. Основанием для принятия решения о предоставлении доступа является сравнение дискреционных атрибутов виртуальной машины и запрашиваемого субъектом типа доступа (операций) к ней.

Все операции с виртуальной машиной разделяются на непривилегированные и привилегированные, например: операции получения информации о списке машин или о конфигурации конкретной машины являются операциями непривилегированными, а операции создания, удаления или изменения конфигурации виртуальных машин — привилегированными.

В момент запуска виртуальной машины ее владельцем становится пользователь, от имени которого функционирует процесс, инициировавший ее запуск. Для процесса виртуальной машины устанавливается соответствующий UID (идентификатор пользователя), а для необходимых устройств и файл-образов, принадлежащих виртуальной машине, данный UID используется для указания владельца объекта. При последующих операциях эта информация используется для проверки дискреционного доступа к виртуальной машине.

Назначение UID пользователя процессу виртуальной машины обеспечивает применение дискреционного принципа контроля доступа при обращении виртуальной машины к ресурсам хостовой ОС.

Подробное описание управления дискреционным доступом приведено в 8.3.

3) Мандатный принцип контроля доступа.

В ПК реализован мандатный принцип контроля доступа к управлению виртуальными машинами и при взаимодействии между виртуальными машинами и хостовой ОС. Основанием для принятия решения о предоставлении доступа является сравнение мандатных атрибутов процесса виртуальной машины и мандатных атрибутов субъекта доступа (процесса) с учетом выполняемой операции.

Все операции с виртуальной машиной разделяются на операции чтения и записи, например: операции получения информации о списке машин или о конфигурации конкретной машины являются операциями чтения, а операции создания, удаления или изменения конфигурации виртуальных машин — записи. Мандатное разграничение доступа осуществляется следующим образом:

– остановленная виртуальная машина не обладает мандатными атрибутами (если для нее мандатные атрибуты не заданы статически);
– запущенная виртуальная машина наследует мандатные атрибуты субъекта доступа, инициировавшего ее запуск;
– доступ к функционирующей виртуальной машине предоставляется только при равенстве мандатных атрибутов субъекта доступа мандатным атрибутам виртуальной машины;
– доступ к получению информации о виртуальной машине и ее конфигурации предоставляется в соответствии с мандатными атрибутами субъекта доступа.

В момент запуска виртуальной машины ее процессу хостовой ОС устанавливаются мандатные атрибуты, полученные из сетевого соединения и унаследованные от субъекта доступа, инициировавшего запуск виртуальной машины. Полученные мандатные атрибуты устанавливаются как в качестве мандатных атрибутов процесса виртуальной машины в ОС СН, так и для необходимых устройств и файлов-образов, принадлежащих виртуальной машине. При последующих операциях эта информация используется средствами мандатного разграничения доступа к виртуальной машине.

В наличие у виртуальной машины мандатных атрибутов обеспечивает в процессе ее функционирования применение мандатного принципа контроля доступа при обращении виртуальной машины к ресурсам хостовой ОС.

Средства ПК выполняют автоматическую маркировку сетевого трафика меткой виртуальной машины, что обеспечивает взаимодействие между процессами пользователей и между виртуальными машинами по протоколам стека IPv4 в условиях мандатного разграничения доступа.

Подробное описание управления мандатным доступом приведено в 8.4.

4) Очистка памяти.

В ПК реализована поддержка функционирования виртуальной машины в режиме запрета модификации ее файлов-образов на основе специальных способов запуска виртуальной машины, при которых основной файл-образ защищается от записи. В зависимости от выбранного режима используется создание физической копии или различные варианты создания снимков файлов-образов с последующим их удалением после завершения работы виртуальной машины.

В ПК реализовано функционирование каждой виртуальной машины, запущенной в виде отдельного процесса хостовой ОС от имени определенного пользователя с установленными мандатными атрибутами. Это обеспечивает очистку областей оперативной и внешней памяти.

Подробно данный режим функционирования виртуальной машины описан в 8.5.

5) Изоляция модулей.

В ПК реализовано функционирование каждой виртуальной машины в виде отдельного процесса хостовой ОС, запущенной от имени определенного пользователя с установленными мандатными атрибутами. Это обеспечивает изоляцию модулей субъекта доступа друг от друга.

6) Маркировка документов.

Маркировка документов осуществляется при формирований из виртуальных машин заданий печати на защищенном сервере печати из состава ОС СН. Механизм маркировки основан на реализации в ПК функционирования каждой виртуальной машины в виде отдельного процесса хостовой ОС, запущенной от имени определенного пользователя с установленными мандатными атрибутами. Указанный подход обеспечивает автоматическую маркировку всех сетевых пакетов протокола IPv4 соответствующими мандатными атрибутами, унаследованными от процесса виртуальной машины. Взаимодействие с защищенными сервером печати при формировании задания печати из виртуальной машины осуществляется по протоколу IPv4. Таким образом, автоматическая маркировка заданий печати осуществляется на основе наследования метки процесса виртуальной машины.

7) Защита ввода-вывода на отчуждаемый физический носитель информации и сопоставление пользователя с устройством.

В ПК реализовано функционирование каждой виртуальной машины, запущенной в виде отдельного процесса хостовой ОС от имени определенного пользователя с установленными мандатными атрибутами. Для файлов-образов и устройств, соответствующих виртуальной машине (идентифицированных), в качестве владельца устанавливается UID пользователя и соответствующие мандатные атрибуты.

Таким образом, ПК обеспечивает ввод-вывод при соответствии мандатных атрибутов субъекта доступа (процесса виртуальной машины) и мандатных атрибутов устройств и надежное сопоставление пользователя с устройством.

8) Регистрация событий.

В ПК реализовано функционирование каждой виртуальной машины, запущенной в виде отдельного процесса хостовой ОС от имени определенного пользователя с установленными мандатными атрибутами. Это обеспечивает регистрацию событий по доступу виртуальных машин к объектам ОС СН с использованием механизмов файлового аудита ОС СН.

Кроме того, сервер виртуализации ПК содержит средства регистрации событий, позволяющие регистрировать все действия пользователей по управлению виртуальными машинами.

9) Восстановление изделия после сбоев и отказов оборудования.

ПК позволяет создавать различные варианты отказоустойчивых конфигураций обеспечивающих не только восстановление изделия после сбоев и отказов оборудования, но и функционирование виртуальных машин в режиме высокой доступности (см. 5).

10) Контроль целостности комплекса средств защиты ПК.

Все средства защиты информации ПК реализованы в исполняемых модулях формата ELF, которые подписаны для функционирования в режиме замкнутой программной среды ОС СН. Таким образом, целостность средств защиты ПК проверяется каждый раз при отображении исполняемых модулей в адресное пространство процесса.

Запуск сервера виртуализации реализован на основе сценария (скрипта). Указанный скрипт может быть при необходимости модифицирован для автоматического регламентного контроля целостности соответствующими средствами из состава ОС СН.

Описание контролируемых функций ОС СН приведено в документе РУСБ.10015-07 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1».

1.2. Требования к техническим средствам

Сервер и клиентское программное обеспечение ПК со встроенными СЗИ от НСД должны функционировать на оборудовании, отвечающих требованиям к аппаратному обеспечению под управлением ОС СН.

Для функционирования сервера виртуализации ПК необходима следующая мини-

мальная конфигурация оборудования:

– аппаратная платформа — процессор с архитектурой x86-64 с аппаратной поддержкой виртуализации (AMD, Intel);
– оперативная память — от 8 ГБ;
– объем свободного дискового пространства — от 30 ГБ;
– сетевая плата 100 Мбит/с;
– источник бесперебойного питания;
– устройство чтения/записи CD/DVD-дисков.

2. СОСТАВ И ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ

ПК состоит из компонентов, представляющих собой набор программ, сгруппированных по определенным признакам.

1) Серверная часть:
   - средства эмуляции аппаратного обеспечения (2.2);
   - сервер виртуализации (2.3);
   - служба доступа к сетевой защищенной файловой системе (2.4).
2) Клиентская часть:
   - консольный интерфейс управления виртуальными машинами (2.5);
   - графический интерфейс управления виртуальными машинами (2.6);
   - графический интерфейс для удаленного доступа (VDI) пользователей к виртуальным машинам по протоколам VNC и Spice (2.7).

Данные компоненты являются составными частями ПК, взаимодействуют между собой в процессе ее функционирования и располагаются в соответствующих deb-пакетах (см. 3).

2.1. Принципы функционирования

ПК функционирует под управлением ОС СН и использует архитектуру «клиент-сервер». Схема функционирования ПК в среде ОС СН представлена на рис. 1.

Серверная часть обеспечивает создание и функционирование защищенной виртуальной среды, содержащей набор виртуальных машин с определенной конфигурацией аппаратных средств и развернутыми в них гостевыми ОС.

Функционирование виртуальных машин обеспечивается модулем ядра KVM из состава ОС СН, который использует аппаратные возможности архитектуры x86-64 по виртуализации процессоров, средства эмуляции аппаратного обеспечения на основе QEMU и сервера виртуализации на основе libvirt.

Клиентские части обеспечивают удаленный доступ пользователей ПК к рабочим столам виртуальных машин с использованием технологии VDI и удаленное управление администраторами конфигурацией виртуальных машин.

С использованием консольных или графических средств осуществляется создание и настройка виртуальных машин, включая формирование конфигурации аппаратных средств виртуальных машин и установку гостевых ОС в виртуальные машины.

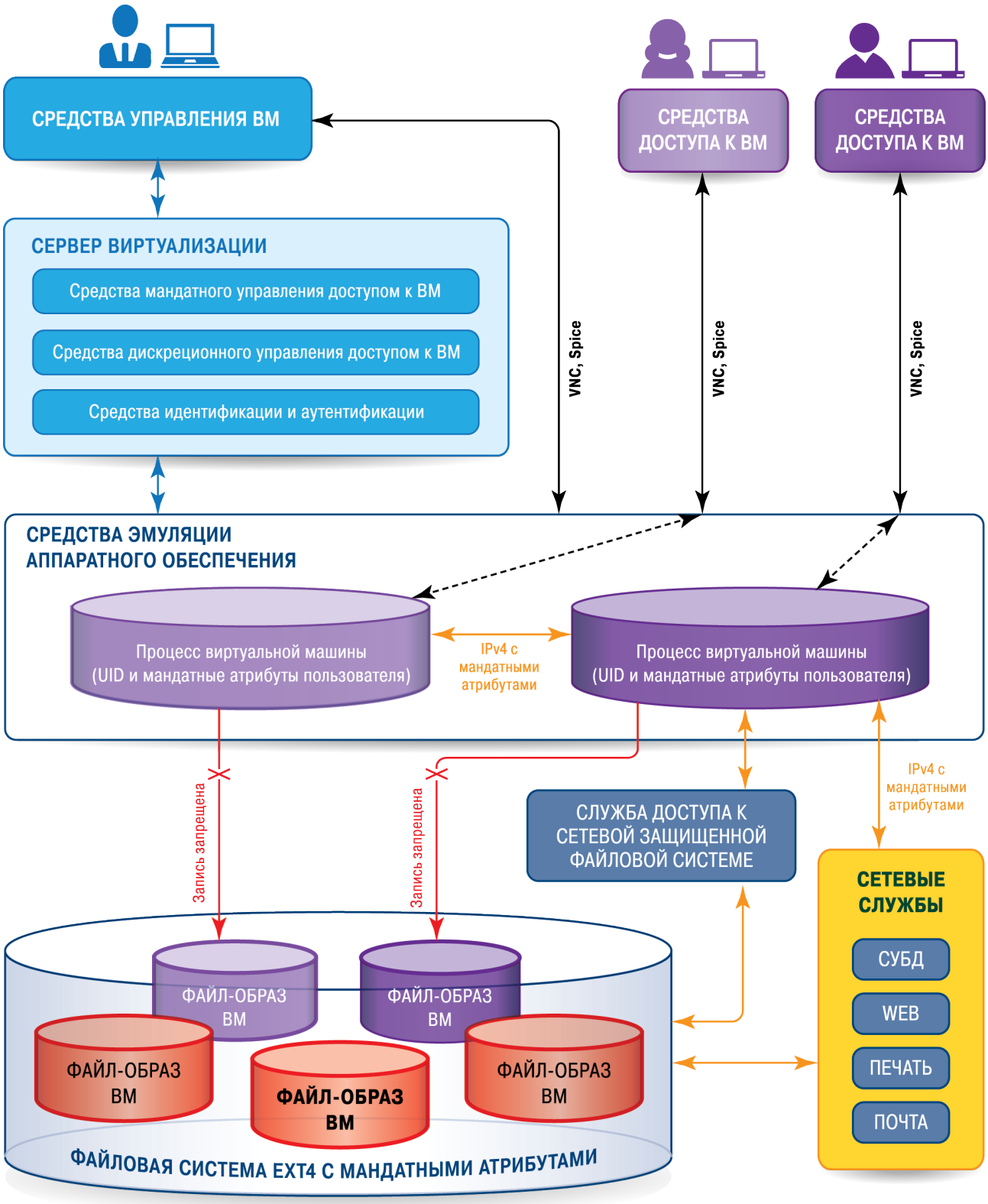


Рис. 1

Пользователи ПК после прохождения процедуры идентификации и аутентификации либо выполняют запуск виртуальной машины, либо с использованием технологии VDI по протоколам VNC и Spice получают доступ к ранее запущенным виртуальным машинам в соответствии с установленными правилами разграничения доступа. Запущенная виртуальная машина представляет собой процесс ОС СН, который функционирует от имени учетной

записи пользователя с его мандатными атрибутами безопасности.

В ПК реализовано дискреционное и мандатное управление доступом к виртуальным машинам с использованием драйверов доступа в сервере виртуализации. При этом дискреционное и мандатное управление доступом к файлам-образам виртуальных машин, а также управление сетевым и межпроцессным взаимодействием между виртуальными машинами и сетевыми службами осуществляется средствами эмуляции аппаратного обеспечения совместно со средствами защиты информации ОС СН.

Функционирование виртуальной машины может осуществляться в режиме запрета модификации ее файлов-образов.

Управление виртуальными машинами в ПК осуществляется с помощью сервера виртуализации на основе libvirt, который предоставляет средства создания и учета виртуальных машин, настройки их конфигурации и непосредственно запуска. В эти задачи входит управление файлами-образов дисковых носителей виртуальных машин, виртуальными сетевыми адаптерами и сетями и формирование контекста функционирования виртуальной машины в виде процесса ОС СН.

Для хранения конфигурации и параметров виртуальных машин используются XML-файлы описания конфигурации виртуальных машин. В файле конфигурации задается состав аппаратных средств, которые необходимо эмулировать для данной виртуальной машины, а также параметры использования ресурсов сервера виртуализации (процессора, оперативной памяти и других физических устройств).

При создании виртуальной машины задаются конфигурационные параметры и создается файл-образ загрузочного диска виртуальной машины. Формат файла-образа зависит от выбранного средства эмуляции аппаратного обеспечения. В ПК используются средства эмуляции аппаратного обеспечения на основе QEMU, которые поддерживают следующие форматы образов: `image` (raw-формат, является фактически представлением физического диска) и формат `qcow2` (родной формат QEMU, поддерживающий возможности сжатия, использования снимков и другие дополнительные возможности). Кроме того, существует возможность конвертирования форматов образов других средств эмуляции аппаратного обеспечения (например, VirtualBox).

При запуске виртуальной машины сервер виртуализации libvirt подготавливает необходимую для функционирования виртуальной машины инфраструктуру и формирует соответствующий набор параметров запуска средства эмуляции аппаратного обеспечения QEMU. После подготовительных действий производится порождения процесса ОС СН, в рамках которого будет функционировать виртуальная машина. Каждая запускаемая виртуальная машина функционирует от имени учетной записи запустившего ее пользователя и с его мандатными атрибутами безопасности.

Для обеспечения безопасности функционирования виртуальных машин сервер виртуализации libvirt использует концепцию драйверов безопасности sVirt . Данная концепция представляет собой специальный программный интерфейс для создания модулей безопасности, используемых для настройки окружения и инфраструктуры запуска и функционирования виртуальный машин в условиях их изоляции и мандатного разграничения доступа.

Выполнение требований по защите информации при функционировании виртуальных машин в ОС СН достигается совместным использованием sVirt модуля дискреционного разграничения доступа `dac` и специально разработанного модуля мандатного разграничения доступа `parsec`, взаимодействующего с подсистемой безопасности PARSEC ОС СН.

На рис. 2 приведен снимок вкладки «Безопасность» графической утилиты управления виртуальными машинами `virt-manager`.

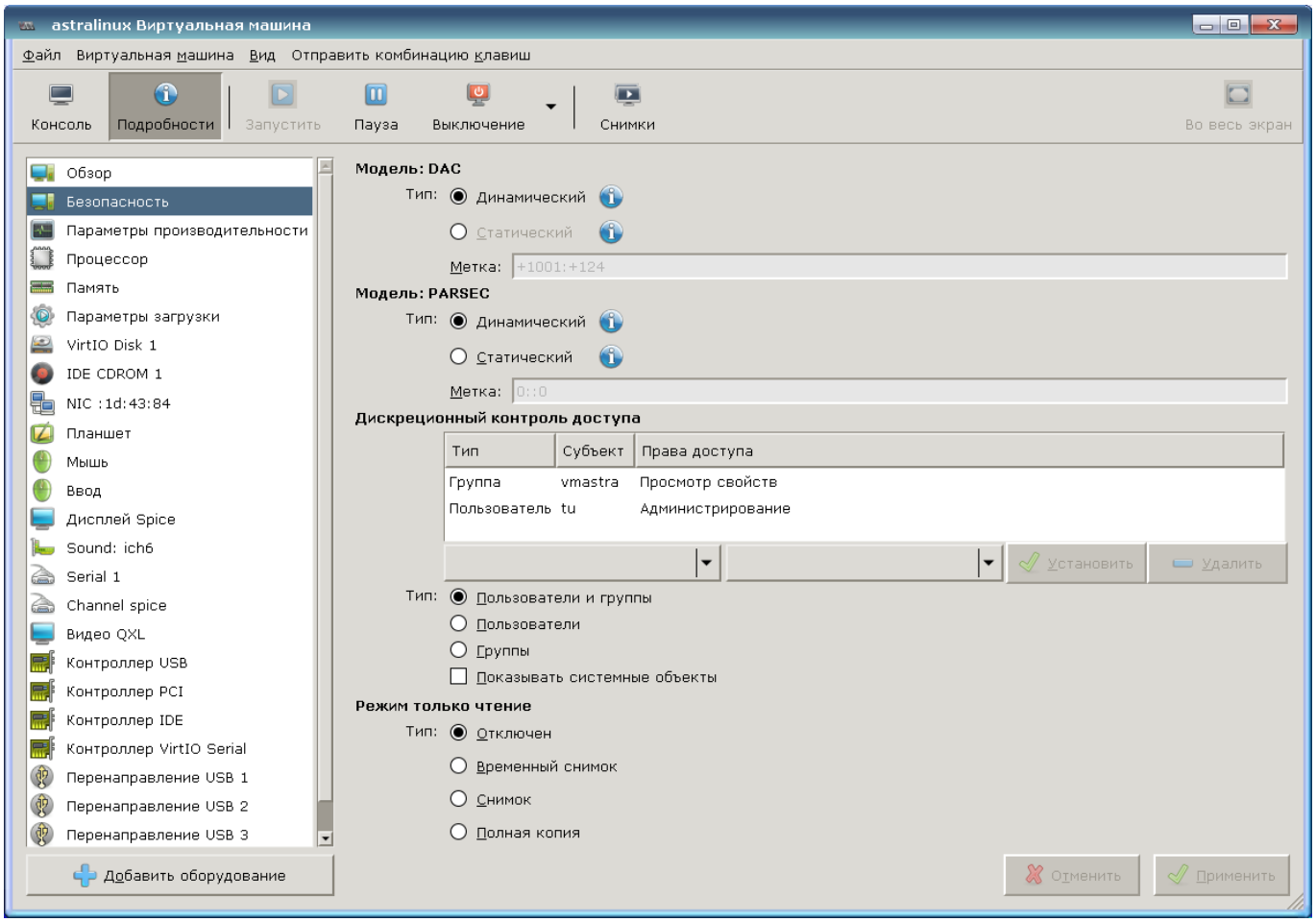


Рис. 2

Контекст безопасности виртуальной машины включает в себя метки безопасности используемых модулей sVirt, например:

– модуль поддержки дискреционного разграничения доступа `dac` использует метку безопасности, состоящую из уникального идентификатора владельца процесса виртуальной машины и уникального идентификатора группы;

– модуль поддержки мандатного разграничения доступа `parsec` использует мандатную метку подсистемы безопасности PARSEC ОС СН.

Метки безопасности могут быть динамическими и статическими:

– динамическая метка безопасности генерируется динамически в момент запуска виртуальной машины на основе атрибутов безопасности запускающего пользователя: его уникального идентификатора и мандатного уровня доступа;

– статическая метка безопасности задается администратором в конфигурации виртуальной машины определяет контекст безопасности ее запуска.

Поддержка дискреционного и мандатного управление доступом к виртуальным машинам в сервере виртуализации реализуется с помощью драйвера доступа `parsec`, специально разработанного с использованием прикладного программного интерфейса драйверов доступа libvirt. Подробное описание принципов управления доступом приведено в следующих подразделах (см. 8.3 и 8.4).

Поддержка функционирования виртуальной машины в режиме запрета модификации ее файлов-образов осуществляется специальными способами запуска виртуальной машины, при которых основной файл-образ защищается от записи. В зависимости от выбранного режима используется создание физической копии или различные варианты создания снимков файл-образов с последующим их удалением после завершения работы виртуальной машины. Более подробно данный режим функционирования виртуальной машины описан в 8.5.

Рассматривая вопросы идентификации и аутентификации при доступе к виртуальным машинам следует различать доступ к серверу виртуализации libvirt для управления виртуальными машинами и доступ пользователя непосредственно к рабочему столу виртуальной машины.

2.2. Средства эмуляции аппаратного обеспечения на основе QEMU

Средства эмуляции аппаратного обеспечения на основе QEMU реализуют программно-аппаратное окружение запускаемой виртуальной машины, включая заданную конфигурацию аппаратной платформы и набор эмулируемых устройств, доступных гостевой операционной системе. В случае совпадения гостевой аппаратной платформы аппаратной платформе хоста используются возможности аппаратной поддержки виртуализации средствами виртуализации KVM (for Kernel-based Virtual Machine) для хостовых операционных систем семейства Linux.

Компонент состоит из пакетов, предоставляющих средства эмуляции аппаратного обеспечения QEMU для различных аппаратных платформ (в частности аппаратных платформ x86 и x86-64) и необходимый набор утилит командной строки.

Запущенная средствами QEMU/KVM виртуальная машина представляет собой отдельный процесс хостовой операционной системы.

2.3. Сервер виртуализации libvirt

Управление виртуальными машинами осуществляется с помощью сервера виртуализации на основе libvirt, который предоставляет средства создания и учета виртуальных машин, настройки их конфигурации и непосредственно запуска. В эти задачи входит управление файлами-образов дисковых носителей виртуальных машин, виртуальными сетевыми адаптерами и сетями и формирование контекста функционирования виртуальной машины в виде процесса ОС СН.

Компонент состоит из службы сервера виртуализации libvirtd, предоставляющей возможность удаленного управления по сети с использованием различных протоколов и способов аутентификации, клиентской библиотеки libvirt0, командной оболочки virsh и ряда других утилит командной строки.

Для взаимодействия других программ с сервером виртуализации могут использоваться дополнительные программные интерфейсы из пакетов с префиксом libvirt-.

2.4. Служба доступа к сетевой защищенной файловой системе Samba

Для организации доступа к разделяемым каталогам сетевой защищенной файловой системы Samba/CIFS из гостевых операционных систем используется модифицированная служба доступа к сетевой защищенной файловой системе Samba, включающая сетевой сервер, клиентские интерфейсы и набор необходимых утилит.

При доступе к разделяемым каталогам из гостевых операционных систем применяется мандатное разграничение доступа с учетом мандатных меток, передаваемых по протоколам стека IPv4.

2.5. Консольный интерфейс управления виртуальными машинами virsh

В состав пакетов сервера виртуализации libvirt входит консольный интерфейс управления виртуальными машинами virsh, позволяющий с помощью командной оболочки производить действия по управлению конфигурацией виртуальных машин.

Командная оболочка содержит развитый набор команд по управлению виртуальными машинами, файлами-образов носителей, виртуальными интерфейсами и сетями и позволяет править конфигурационный файлы виртуальных машин.

Более подробно возможности консольного интерфейса управления виртуальными машинами virsh описаны в соответствующем руководстве man.

2.6. Графический интерфейс управления виртуальными машинами `virt-manager`

В состав ПК входит графический интерфейс управления виртуальными машинами `virt-manager`, предоставляющий доступ к возможностям сервера виртуализации libvirt из графического интерфейса пользователя (см. рис. 3).

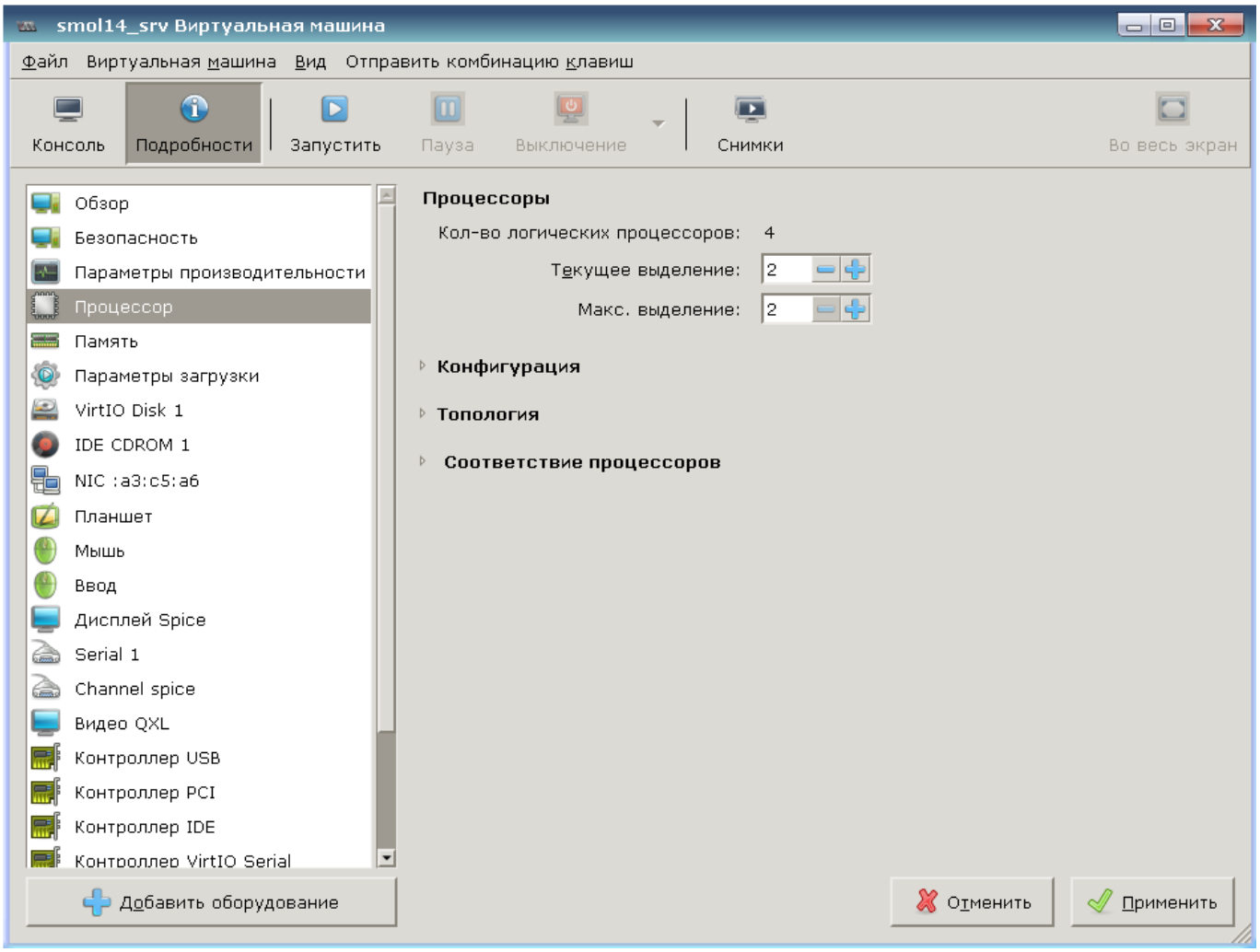


Рис. 3

Интерфейс позволяет наглядным образом выполнять действия по созданию виртуальных машин, управлению их конфигурацией и файлами-образов дисковых носителей. Утилита позволяет получить удаленный доступ к рабочему столу выбранной виртуальной машины по протоколам VNC и Spice.

2.7. Графический интерфейс для удаленного доступа (VDI) пользователей к виртуальным машинам по протоколам VNC и Spice `virt-viewer`

В состав ПК входит графический интерфейс для удаленного доступа (VDI) пользователей к виртуальным машинам по протоколам VNC и Spice `virt-viewer`, предоставляющий доступ пользователя к рабочему столу выбранной виртуальной машины по про-

токолам VNC или Spice (см. рис. 4). Виртуальная машина должна быть уже запущена с помощью консольного или графического интерфейсов управления.

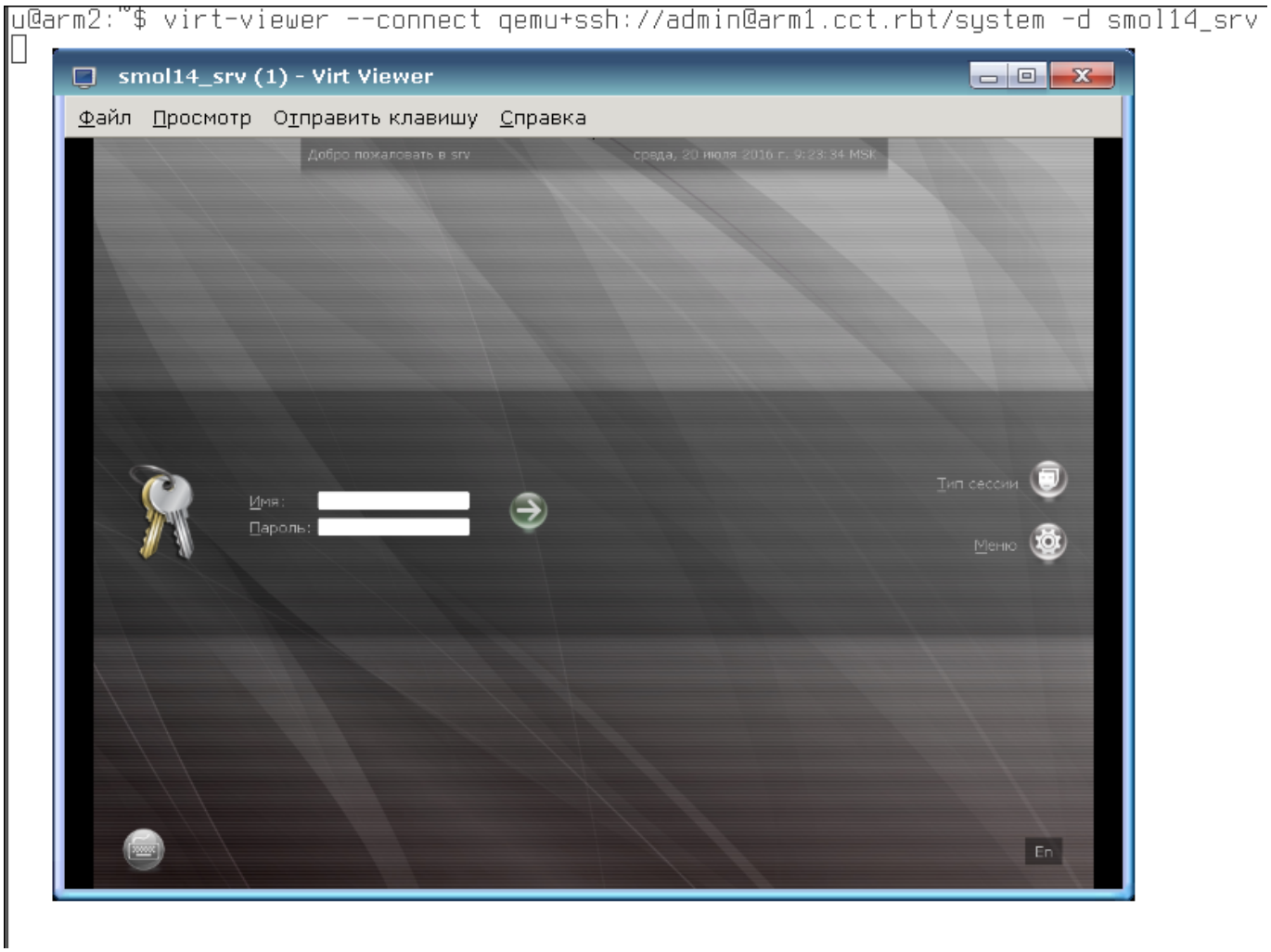


Рис. 4

3. УСТАНОВКА И НАСТРОЙКА

Загрузочный модуль ПК поставляется в виде репозитория на компакт-диске, содержащем основные пакеты, реализующие функциональные возможности ПК, и вспомогательные пакеты, необходимые для его функционирования.

Основными пакетами серверной части ПК являются пакеты средства эмуляции аппаратного обеспечения на основе QEMU и сервера виртуализации libvirt.

Компоненты средства эмуляции аппаратного обеспечения на основе QEMU содержаться в следующих пакетах:

– `qemu-kvm` — средства виртуализации KVM (for Kernel-based Virtual Machine) для хостовых операционных систем семейства Linux на аппаратной платформе x86-64, позволяющая выполнять гостевые виртуальные машины с архитектурой x86 и x86-64, каждая из которых использует собственное виртуальное аппаратное обеспечение: сетевые и графические адаптеры, жесткие диски и т.д. KVM функционирует на процессорах с аппаратной поддержкой виртуализации, обеспечиваемой технологиями Intel VT (Intel Virtualization Technology) или AMD virtualization (AMD-V);

– `qemu-system-common` — предоставляет общие для всех аппаратных платформ файлы средства эмуляции аппаратного обеспечения QEMU;

– `qemu-system-x86` — предоставляет файлы средства эмуляции аппаратного обеспечения QEMU для аппаратных платформ x86 и x86-64;

– `qemu-system` — является метапакетом QEMU, который устанавливает соответствующий аппаратных платформе пакет эмуляции;

– `qemu-user-binfmt` — предоставляет поддержку регистрации средств эмуляции в режиме пользователя из пакета `qemu-user`. Пакет содержит только сценарии, которые выполняются при установке и удалении пакета;

– `qemu-user-static` — предоставляет файлы средства эмуляции в режиме пользователя, скомпонованных статически. Это позволяет запускать файлы, собранные для другой аппаратной платформы. Если пакет `binfmt-support` установлен, то пакетом `qemu-user-static` будут зарегистрированы форматы, которые могут обрабатываться эмулятором таким образом, что появится возможность непосредственно выполнять файлы для других аппаратных платформ;

– `qemu-user` — предоставляет файлы средства эмуляции в режиме пользователя, что позволяет запускать файлы собранные для другой аппаратной платформы. Если пакет `qemu-user-binfmt` установлен, то будут зарегистрированы обработчики из пакета `qemu-user`, что позволит непосредственно выполнять файлы для других аппаратных платформ. Однако указанная возможность не может быть ис-

пользована внутри измененного корневого каталога (chroot). В подобном случае используется пакет qemu-user-static;

– qemu-utils — предоставляет следующие утилиты средства эмуляции аппаратного обеспечения QEMU: qemu-img, qemu-io, qemu-nbd.

Компоненты сервера виртуализации libvirt содержатся в следующих пакетах:

– libvirt-clients — оболочка virsh и клиентские программы;

– libvirt-daemon-system — конфигурационные файлы для запуска службы сервера виртуализации libvirtd как системной службы;

– libvirt-daemon — служба сервера виртуализации libvirtd;

– libvirt-doc — документация на библиотеку libvirt;

– libvirt-sanlock — плагин sanlock;

– libvirt0 — библиотека libvirt для взаимодействия с механизмами виртуализации в среде Linux.

Для взаимодействия других программ могут использоваться дополнительные пакеты с префиксом libvirt-*.

Для использования сетевой защищенной файловой системы Samba/CIFS из гостевых операционных систем на хостовой системе должна быть обновлена служба организации доступа к сетевой защищенной файловой системе Samba, компоненты которой содержатся в следующих пакетах:

– libnss-winbind — NSS модуль интеграции Samba;

– libpam-smbpass — PAM модуль Samba;

– libpam-winbind — PAM модуль интеграции Samba;

– libsmbclient — клиентская библиотека Samba;

– libwbclient0 — клиентская библиотека сервера интеграции Samba

– samba-common-bin — общие для клиента и сервера СЗФС бинарные файлы;

– samba-common — общие для клиента и сервера СЗФС файлы;

– samba-doc-pdf — документация Samba в формате PDF;

– samba-doc — документация Samba;

– samba-tools — утилиты тестирования Samba;

– samba — службы доступа к сетевой защищенной файловой системе;

– smbclient — клиентские утилиты командной строки для доступа к сетевой защищенной файловой системе;

– swat — средства WEB-администрирования Samba.

Основными пакетами клиентской части ПК являются пакеты консольных и графических интерфейсов управления виртуальными машинами:

– libvirt-clients — оболочка virsh и клиентские программы;

– `virt-manager` — графический интерфейс управления виртуальными машинами `virt-manager`;

– `virt-viewer` — графический интерфейс для удаленного доступа (VDI) пользователей к виртуальным машинам по протоколам VNC и Spice `virt-viewer`;

3.1. Установка серверной части ПК

Установка и настройка ПК выполняется от имени учетной записи администратора через механизм `sudo`.

Для установки серверной части ПК необходимо:

1) вставить компакт-диск с дистрибутивом ПК в устройство чтения CD/DVD-дисков;
2) добавить содержимое компакт-диска с дистрибутивом ПК в репозиторий, выполнив команду:

```
apt-cdrom add
```

3) обновить репозиторий, выполнив команду:

```
apt-get update
```

4) установить серверную часть ПК, выполнив команду:

```
apt-get install virt-manager libvirt-bin
```

3.2. Установка клиентской части ПК

Для установки клиентской части ПК необходимо:

1) вставить компакт-диск с дистрибутивом ПК в устройство чтения CD/DVD-дисков;
2) добавить содержимое компакт-диска с дистрибутивом ПК в репозиторий, выполнив команду:

```
apt-cdrom add
```

3) обновить репозиторий, выполнив команду:

```
apt-get update
```

4) установить клиентскую часть ПК, выполнив команду:

```
apt-get install virt-manager
```

3.3. Настройка

ПК представляет собой набор программных средств, требующих настройки. В первую очередь это относится к серверной части ПК.

Сервер виртуализации использует следующие каталоги хостовой ФС:

1) `/etc/libvirt/` — каталог конфигурации сервера виртуализации libvirt:
   – `qemu/` — каталог конфигурационных XML-файлов виртуальных машин QEMU:
     – `network/` — каталог конфигурационных XML-файлов виртуальных сетей;

– `*.xml` — конфигурационные XML-файлы виртуальных машин QEMU.

– `storage/` — каталог конфигурационных файлов пулов файлов-образов;

– `libvirt.conf` — клиентский конфигурационный файл libvirt;

– `libvirtd.conf` — конфигурационный файл службы сервера виртуализации `libvirtd` (см. 3.4);

– `qemu.conf` — конфигурационный файл средства эмуляции аппаратного обеспечения QEMU (см. 3.5).

2) `/var/lib/libvirt/` — рабочий каталог сервера виртуализации libvirt:

– `images/` — каталог файлов-образов по умолчанию:

– `network/` — рабочий каталог виртуальных сетей;

– `qemu/` — рабочий каталог запущенных виртуальных машин QEMU:

– `save/` — каталог сохраненных состояний виртуальных машин;

– `snapshot` — каталог снимков виртуальных машин.

– `runimages/` — каталог расположения копий файлов-образов виртуальных машин, запущенных в режиме запрета модификации файлов-образов.

3) `/var/run/libvirt/` — каталог текущего рабочего состояния сервера виртуализации libvirt:

– `network/` — рабочий рабочий запущенных виртуальных сетей;

– `qemu/` — каталог текущих конфигурационных XML-файлов запущенных виртуальных машин QEMU;

– `libvirt-sock` — Unix-сокет для локальных соединений со службой сервера виртуализации `libvirtd`;

– `libvirt-sock-ro` — Unix-сокет, доступный только для чтения, для локальных соединений со службой сервера виртуализации `libvirtd`.

ВНИМАНИЕ! Никакие конфигурационные файлы или файлы, содержащие ключевую информацию Kerberos или PKI, сервера виртуализации libvirt не должны быть доступны пользователям.

Примечание. Для работы с ПК пользователи может требоваться вхождение пользователей в локальные группы `kvm` и `libvirt-qemu`.

ВНИМАНИЕ! Операции по изменению состава или конфигурации виртуальных машин требуют вхождения пользователя в специальную локальную административную группу `libvirt-admin`.

3.4. Конфигурационный файл службы сервера виртуализации `libvirtd.conf`

Конфигурационный файл службы сервера виртуализации `libvirtd` содержит необходимые для работы службы настройки и параметры и расположен в `/etc/libvirt/libvirtd.conf`. Конфигурационный файл разбит на подразделы, отвечающие за различные аспекты функционирования службы сервера виртуализации: интерфейсы взаимодействия и права доступа к ним, способы и параметры аутентификации, политику разграничения доступа, состав выводимой в журнал информации и т.п. Наиболее важные параметры приведены в таблице 1.

Таблица 1 – Параметры конфигурационного файла `/etc/libvirt/libvirtd.conf`

Параметр	Описание
`listen_tls`	Принимать TLS соединения с использованием сертификатов
`listen_tcp`	Принимать TCP соединения **ВНИМАНИЕ!** Одной установки данного параметра недостаточно: необходимо указать опцию `-l` для параметра `libvirtd_opts` в конфигурационном файле `/etc/default/libvirtd`
`listen_addr`	Адрес сетевого интерфейса для приема соединений
`tls_port` и `tcp_port`	Значения портов для сетевых соединений TLS и TCP
`auth_tcp`	Используемая для TCP-соединений аутентификация. Параметр должен содержать значение `sasl` (см. 3.6).
`access_drivers`	Применяемый драйвер доступа к серверу виртуализации. Параметр должен содержать значение `[ "parsec" ]`.
`admin_group`	Группа администраторов сервера виртуализации (по умолчанию `"libvirt-admin"`)

Примечание. Конфигурационные параметры TLS для доступа к серверу виртуализации libvirt рассматриваются в 8.1.

3.5. Конфигурационный файл средства эмуляции аппаратного обеспечения QEMU

Конфигурационный файл средства эмуляции аппаратного обеспечения QEMU содержит необходимые для запуска виртуальных машин параметры и расположен в `/etc/libvirt/qemu.conf`. Конфигурационные параметры отвечают за различные аспекты функционирования виртуальных машин: интерфейсы взаимодействия с рабочим столом виртуальных машин, способы и параметры аутентификации, политику управления безопасностью и изоляцией виртуальных машин и значения по умолчанию некоторых параметров конфигурации виртуальных машин. Наиболее важные параметры приведены в таблице 2.

Таблица 2 – Параметры конфигурационного файла `/etc/libvirt/qemu.conf`

Параметр	Описание
`vnc_listen`	Адрес сетевого интерфейса для приема соединений по протоколу VNC
`vnc_tls`	Использовать TLS для приема соединений по протоколу VNC
`vnc_tls_x509_cert_dir`	Путь к сертификатам TLS при работе по протоколу VNC
`vnc_password`	Пароль для соединений по протоколу VNC
`vnc_sasl`	Использовать SASL для приема соединений протоколу по VNC
`vnc_sasl_dir`	Каталог конфигурационного файла `qemu.conf`, содержащий SASL-параметры для приема соединений по протоколу VNC (см. 3.6)
`spice_listen`	Адрес сетевого интерфейса для приема соединений по протоколу Spice
`spice_tls`	Использовать TLS для приема соединений по протоколу Spice
`spice_tls_x509_cert_dir`	Путь к сертификатам TLS при работе по протоколу Spice
`spice_password`	Пароль для соединений по протоколу Spice
`spice_sasl`	Использовать SASL для приема соединений по протоколу Spice
`spice_sasl_dir`	Каталог конфигурационного файла `qemu.conf`, содержащий SASL-параметры для приема соединений по протоколу Spice (см. 3.6)
`security_driver`	Применяемый драйвер безопасности. Параметр должен содержать значение `"parsec"`.
`run_images_dir`	Каталог расположения копий файлов-образов виртуальных машин, запущенных в режиме запрета модификации файлов-образов

Примечание. Конфигурационные параметры TLS для доступа к рабочим столам виртуальных машин по протоколу VNC рассматриваются в 8.2.

3.6. Конфигурационные файлы SASL сервера виртуализации

При использовании механизмов SALS для доступа к серверу виртуализации libvirt или к рабочим столам виртуальных машин по протоколам VNC и Spice необходимо наличие соответствующих конфигурационных файлов с параметрами SASL в каталоге `/etc/sasl2`. Для сервера виртуализации файл должен иметь имя `libvirt.conf`, для QEMU (протоколы VNC и Spice) — `qemu.conf`. Для протоколов VNC и Spice могут быть заданы другие пути расположения конфигурационных файлов SASL (см. 3.5).

Конфигурационный файл SASL содержит следующие важные параметры:

Таблица 3 – Параметры конфигурационного файла `/etc/sasl2/libvirt.conf`

Параметр	Описание
`mech_list`	Список механизмов SASL. При использовании ПК в ЕПП (ALD) должен содержать только значение `gssapi`.
`keytab`	Путь к файлу ключевой информации Kerberos. Параметр необходим при использовании в ЕПП (ALD). Должен содержать корректные значения для файлов, содержащих ключевую информацию для VNC и Spice
`sasldb_path`	Путь к базе данных SASL. При использовании ПК в ЕПП (ALD) не применяется и должен быть закомментирован.

ВНИМАНИЕ! Другие параметры при использовании ПК в ЕПП (ALD) не применяются и должны быть закомментированы.

ВНИМАНИЕ! Файлы ключевой информации Kerberos для VNC и Spice должны быть доступны на чтение пользователям, запускающим виртуальные машины, и группе `libvirt-qemu`.

4. УПРАВЛЕНИЕ ВИРТУАЛЬНЫМИ МАШИНАМИ

Управление виртуальными машинами в ПК осуществляется с помощью консольного или графического интерфейсов управления виртуальными машинами путем обращения к соответствующему серверу виртуализации libvirt.

Функциональные обязанности администратора включают в себя выполнения следующих задач администрирования:

– настройка доступа к серверу виртуализации libvirt (см. 8);
– создание виртуальных машин (см. 4.1);
– управление конфигурацией виртуальных машин (см. 4.2);
– управление файлами-образов дисков виртуальных машин (см. 4.3);
– управление виртуальными сетевыми интерфейсами и сетями (см. 4.4);
– запуск и управление функционированием виртуальных машин (см. 4.5);
– управление разграничением доступа к виртуальным машинам (см. 8);

ВНИМАНИЕ! Операции по изменению состава или конфигурации виртуальных машин требуют вхождения пользователя в специальную локальную административную группу `libvirt-admin`.

ВНИМАНИЕ! Существуют ограничения по конфигурированию и запуску виртуальных машин в условиях мандатного разграничения доступа (см. 4.6).

4.1. Создание виртуальных машин

Для создания виртуальных машин с помощью графической утилиты необходимо войти в ОС СН под пользователем, включенным в группу `astra-admin` и запустить менеджер виртуальных машин `virt-manager` (рис. 5);

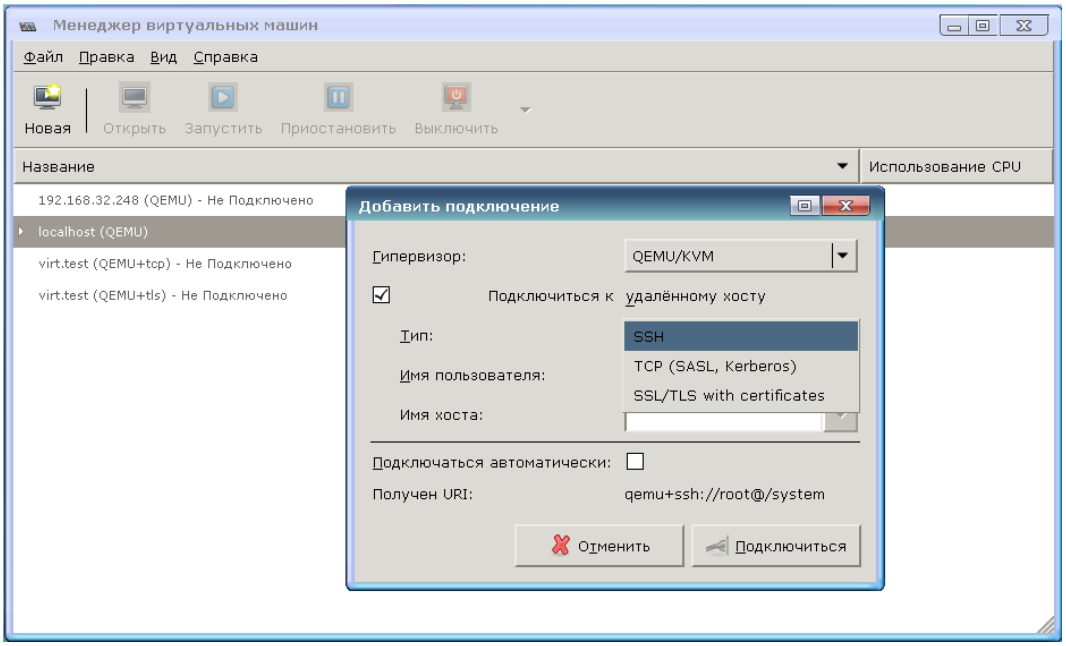


Рис. 5

После установки соединений с нужным сервером виртуализации необходимо нажать на кнопку «Новая» или в панели меню выбрать «Файл — Новая виртуальная машина», выбрать «Установка с локального носителя», архитектура «x86-64» (рис. 6).

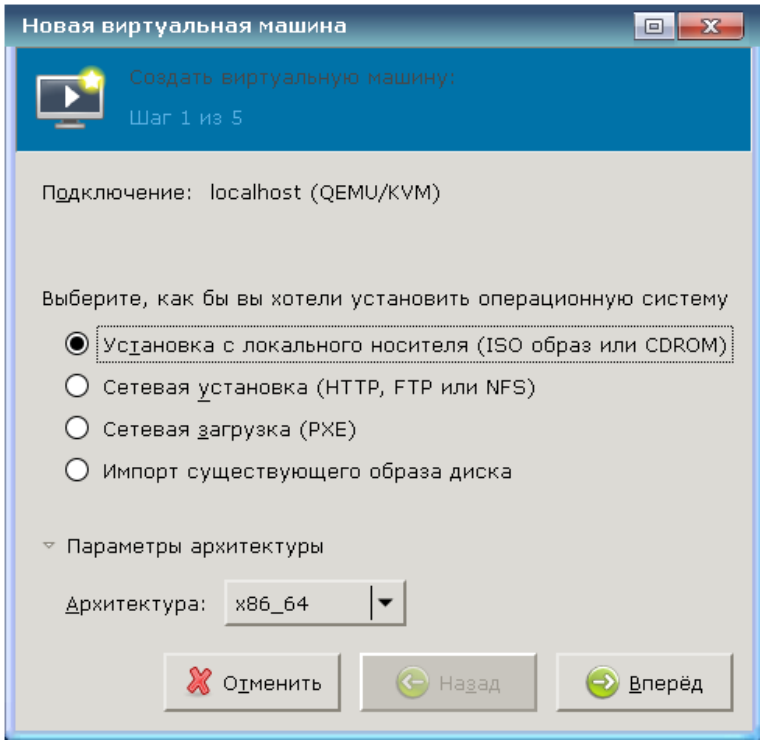


Рис. 6

Для установки гостевой ОС требуется указать ISO-образ установочного диска ОС СН или DVD-диск с дистрибутивом ОС СН, предварительно вставив DVD-диск устройство чтения CD/DVD-дисков (рис. 7).

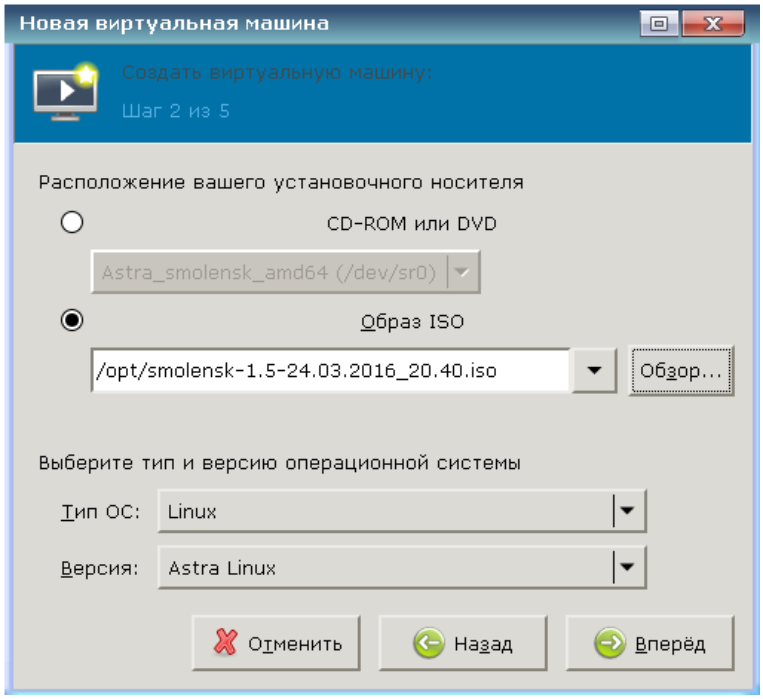


Рис. 7

Для виртуальной машины следует указать размер памяти и количество процессоров. После чего создать файл-образ основного дискового носителя виртуальной машины и задать его размер (минимум 6 ГБ) (рис. 8).

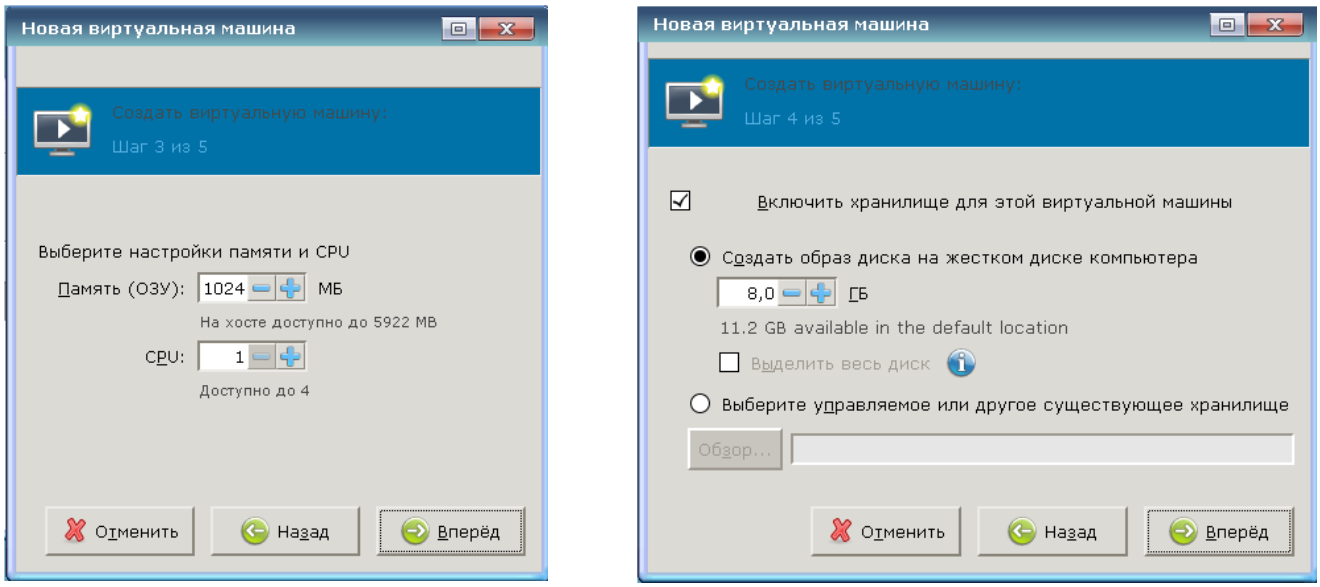


Рис. 8

По завершению конфигурирования виртуальной машины требуется подтвердить все настройки, нажав на кнопку «Завершить» (рис. 9).

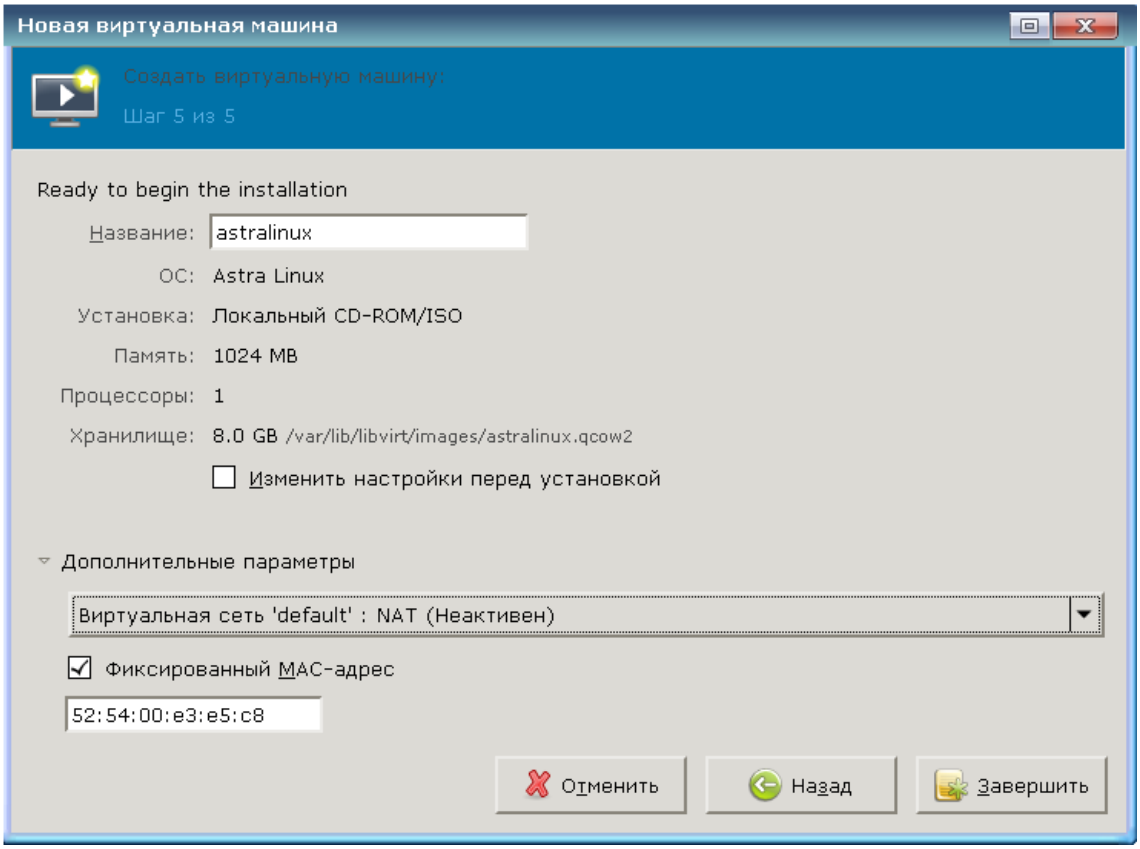


Рис. 9

В результате запустится созданная виртуальная машина с окном установки ОС СН

(рис. 10).

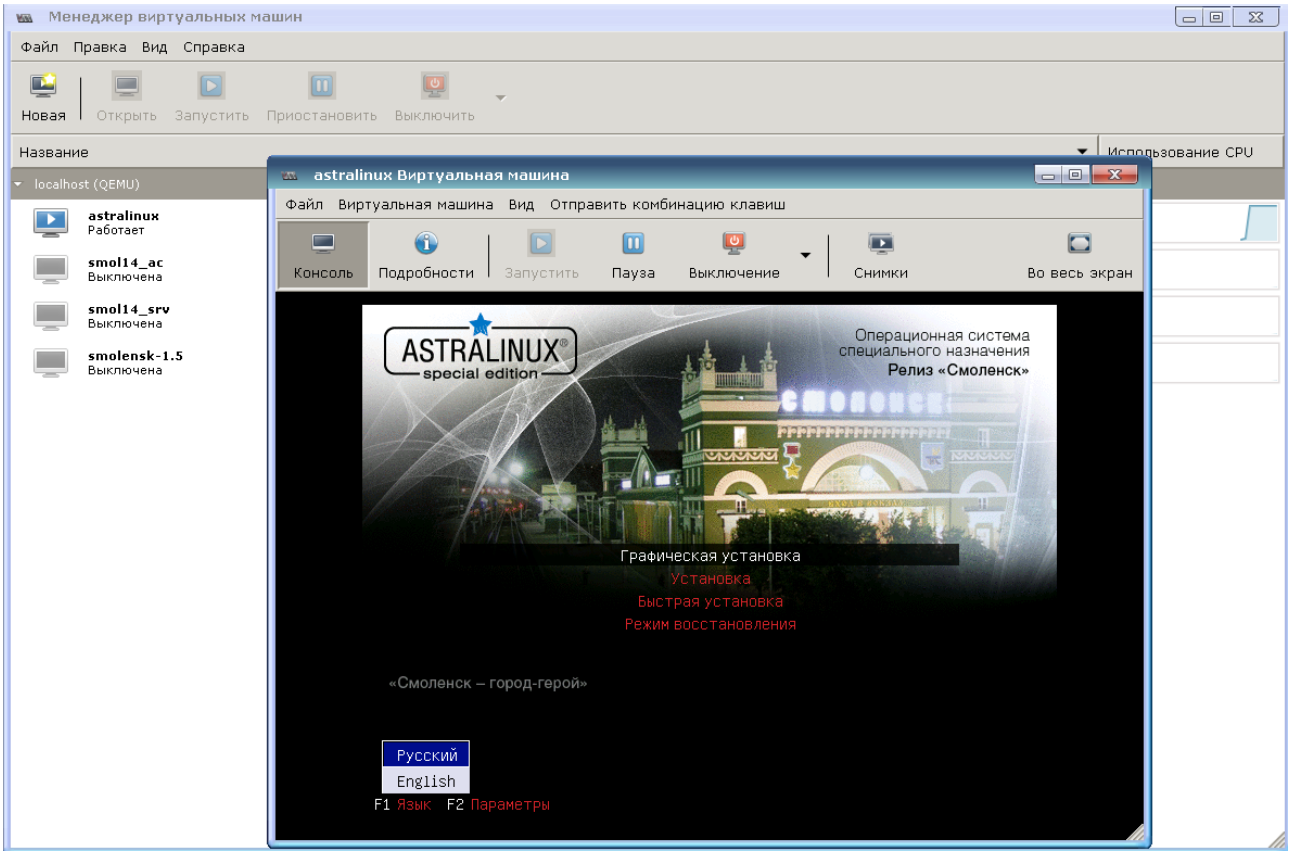


Рис. 10

Для создания виртуальных машин с помощью консольной утилиты необходимо войти в ОС СН под пользователем, включенным в группу `astra-admin`, запустить эмулятор терминала (с помощью сочетания клавиш **<Alt+F2>** запустить утилиту «Выполнить команду» и в строке ввести `fly-term`) и выполнить команду:

```
virt-install --connect qemu:///system -n smol14_acn
             -r 2048 --vcpus=2 -c  /path/to/iso/smolensk.iso
             --disk pool=default,size=6,bus=virtio,format=qcow2
             --os-type=linux  --os-variant=astralinux
```

где:

`-n smol14_acn` — название виртуальной машины;

`--vcpus=2` — количество процессоров;

`-r 2048` — объем оперативной памяти;

`/path/to/iso/smolensk.iso` — путь к ISO-образу установочного диска ОС СН;

`pool=default` — хранилище по умолчанию;

`size=6` — размер хранилища в ГБ.

В результате запустится созданная виртуальная машина с окном установки ОС СН (рис. 10).

4.2. Управление конфигурацией виртуальных машин с помощью графической и консольных утилит

Для управления конфигурацией виртуальных машин с помощью графической утилиты необходимо выбрать виртуальную машину из списка, нажать на кнопку «Открыть» (рис. 11) и нажать на кнопку «Подробности» (рис. 12).

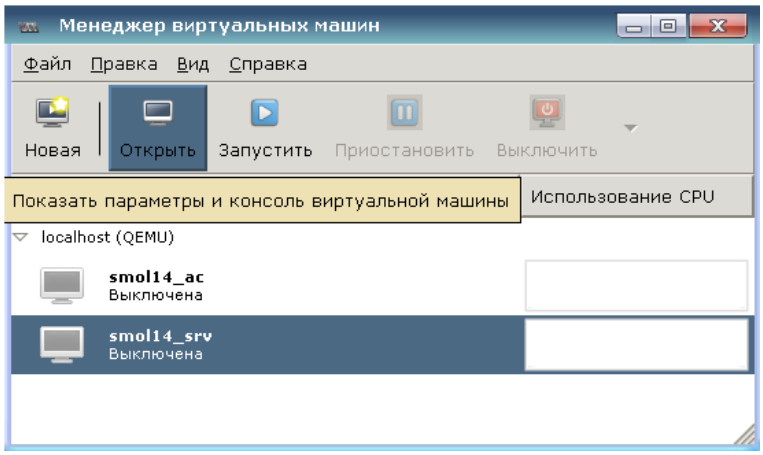


Рис. 11

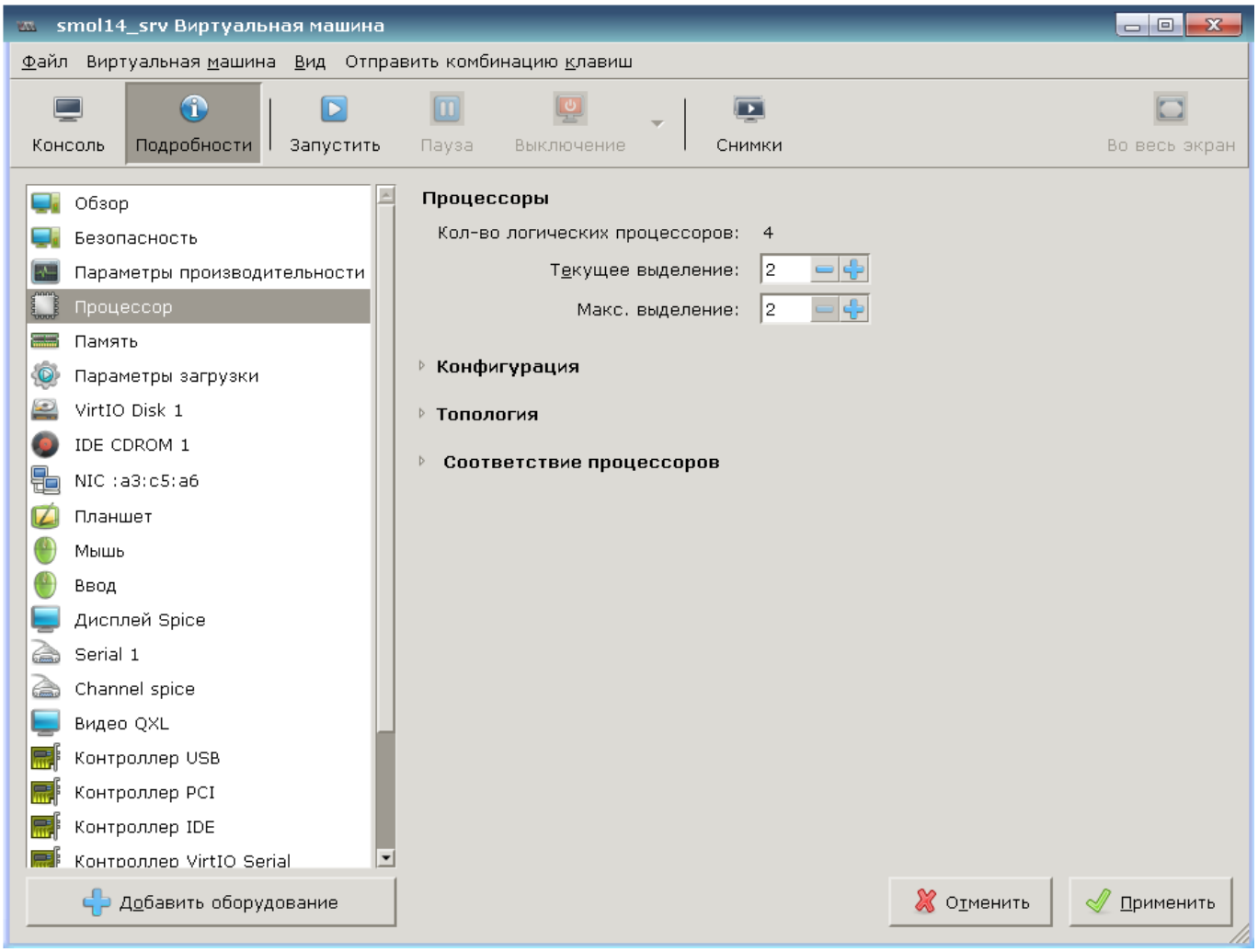


Рис. 12

Для изменение требуемого параметра необходимо перейти на нужную вкладку, внести изменения и подтвердить операцию.

Для управления конфигурацией виртуальных машин с помощью консольной утилиты необходимо выполнить следующие команды (изменения размера оперативной памяти и количества виртуальных процессоров соответственно):

```
virsh --connect qemu:///system setmaxmem --size 4194304 smol14_srv
virsh --connect qemu:///system setvcpus --config smol14_srv 3 -maximum
```

Консольная утилита `virsh` содержит множество команд, подробное описание которых приведено в руководстве `man`.

4.3. Управление файлами-образов дисков виртуальных машин

В ПК работа с файлами-образов носителей производится с помощью пулов носителей. Пул носителей определяет физическое расположение файлов-образов и может быть различных типов.

Для настройки пулов носителей с помощью графической утилиты необходимо войти в ОС СН под пользователем, включенным в группу `astra-admin` и запустить менеджер виртуальных машин `virt-manager`, после чего выбрать подключение к нужному серверу виртуализации и перейти на вкладку «Хранилище» (рис. 13).

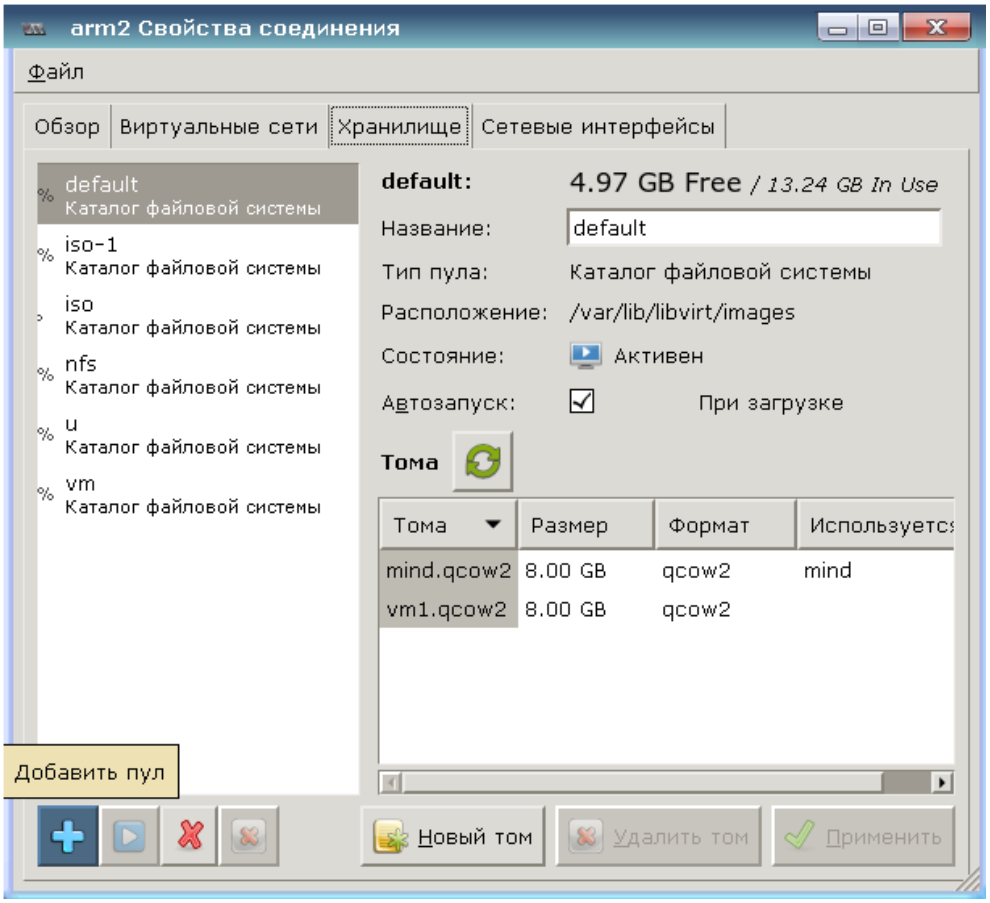


Рис. 13

На рис. 14 приведен внешний вид диалоговых окон при создании пула и файла-образа нового диска. В выпадающем списке выбирается тип пула или необходимый тип файла-образа носителя.

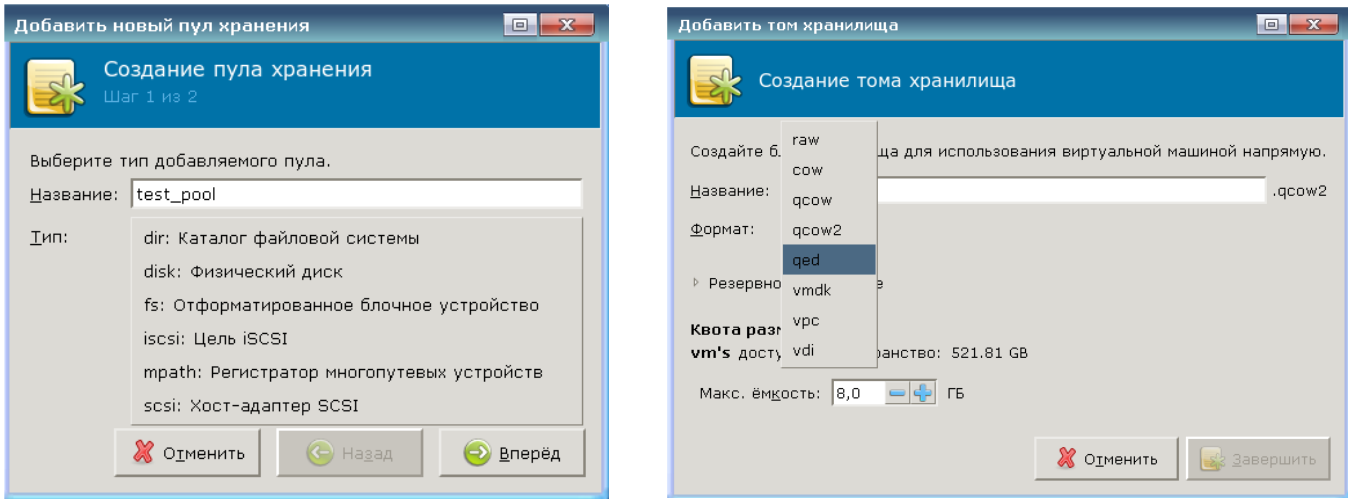


Рис. 14

ВНИМАНИЕ! В ПК используются средства эмуляции аппаратного обеспечения на основе QEMU, которые поддерживают следующие форматы образов: `raw` (raw-формат, является фактически представлением физического диска) и формат `qcow2` (родной формат QEMU, поддерживающий возможности сжатия, использования снимков и другие дополнительные возможности).

Примечание. Существует возможность использования форматов образов других средств эмуляции аппаратного обеспечения (например, VirtualBox). Но для запуска виртуальной машины требуется конвертация в родной формат QEMU.

После создания файл-образ может быть добавлен в конфигурацию виртуальной машины, например с помощью окна свойств виртуальной машины в менеджере виртуальных машин `virt-manager`.

4.4. Управление виртуальными сетевыми интерфейсами и сетями

В ПК существует возможность настройки виртуальных сетевых интерфейсов и сетей для обеспечения сетевого взаимодействия виртуальных машин как между собой, так и с хостовой операционной системой.

Для настройки сетевых интерфейсов с помощью графической утилиты необходимо войти в ОС СН под пользователем, включенным в группу `astra-admin` и запустить менеджер виртуальных машин `virt-manager`, после чего выбрать подключение к нужному серверу виртуализации и перейти на вкладку «Сетевые интерфейсы» (рис. 15).

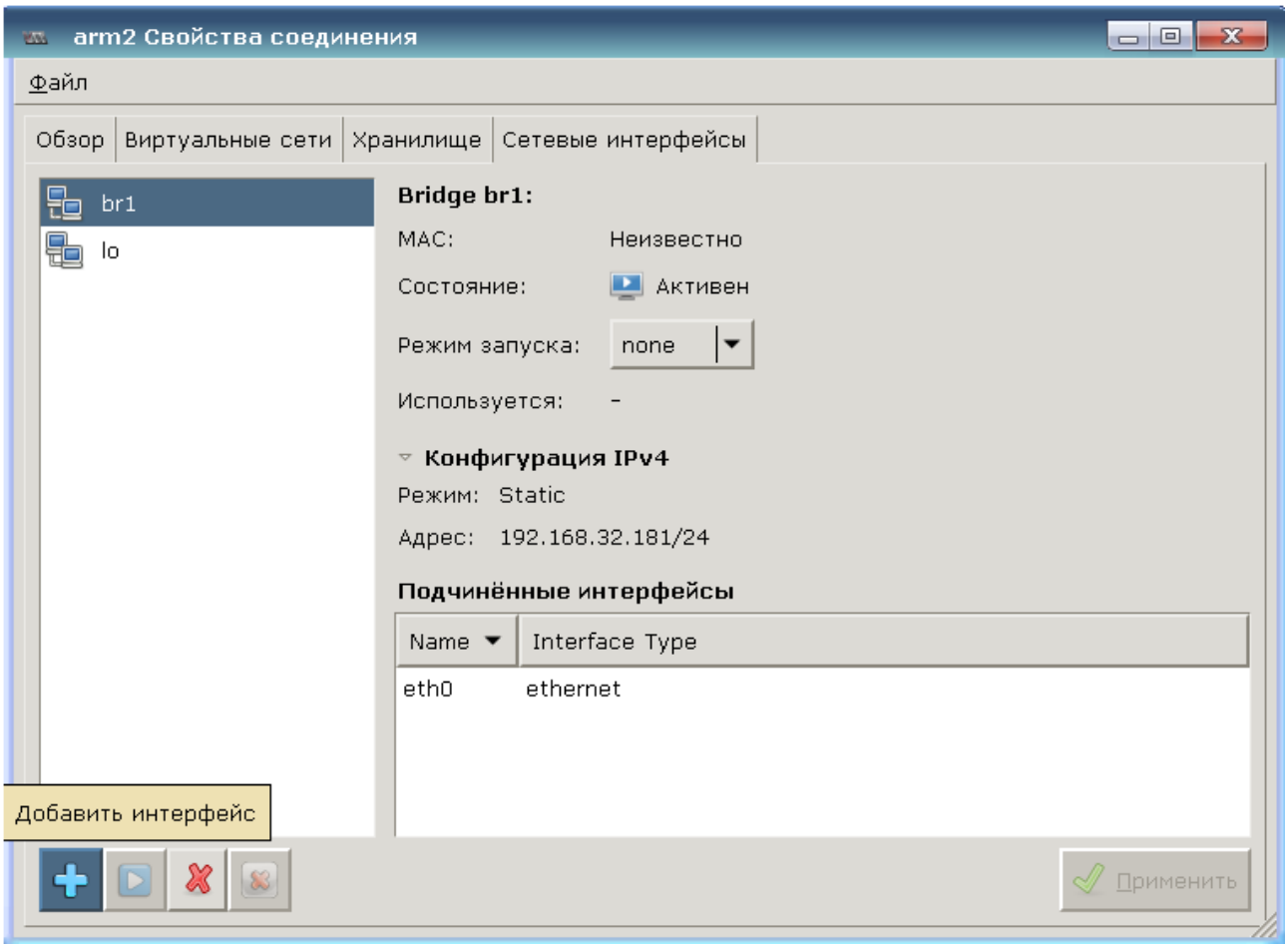
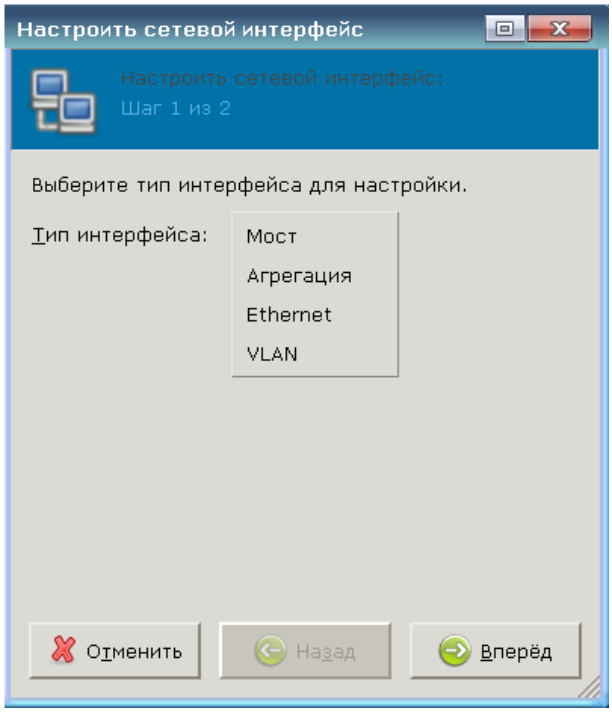


Рис. 15

Для добавления нового сетевого интерфейса следует нажать «+», выбрать тип сетевого интерфейса, ввести название и выбрать режим запуска (рис. 16).



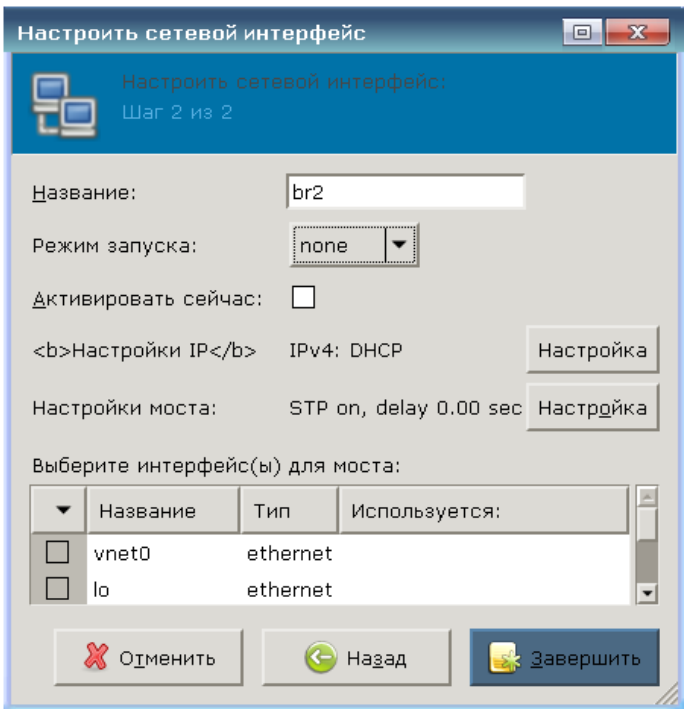


Рис. 16

В дальнейшем выбор виртуального сетевого интерфейса может быть выполнен в свойствах виртуальной машины (рис. 17).

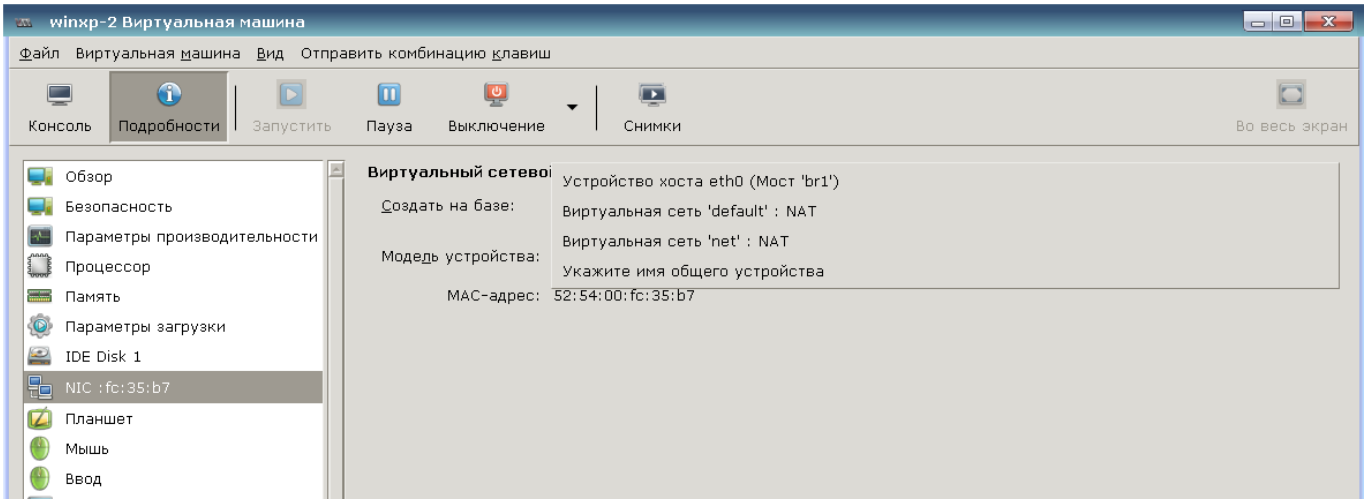


Рис. 17

Для настройки виртуальных сетей с помощью графической утилиты необходимо войти в ОС СН под пользователем, включенным в группу `astra-admin` и запустить менеджер виртуальных машин `virt-manager`, после чего выбрать подключение к нужному серверу виртуализации и перейти на вкладку «Виртуальные сети» (рис. 18).

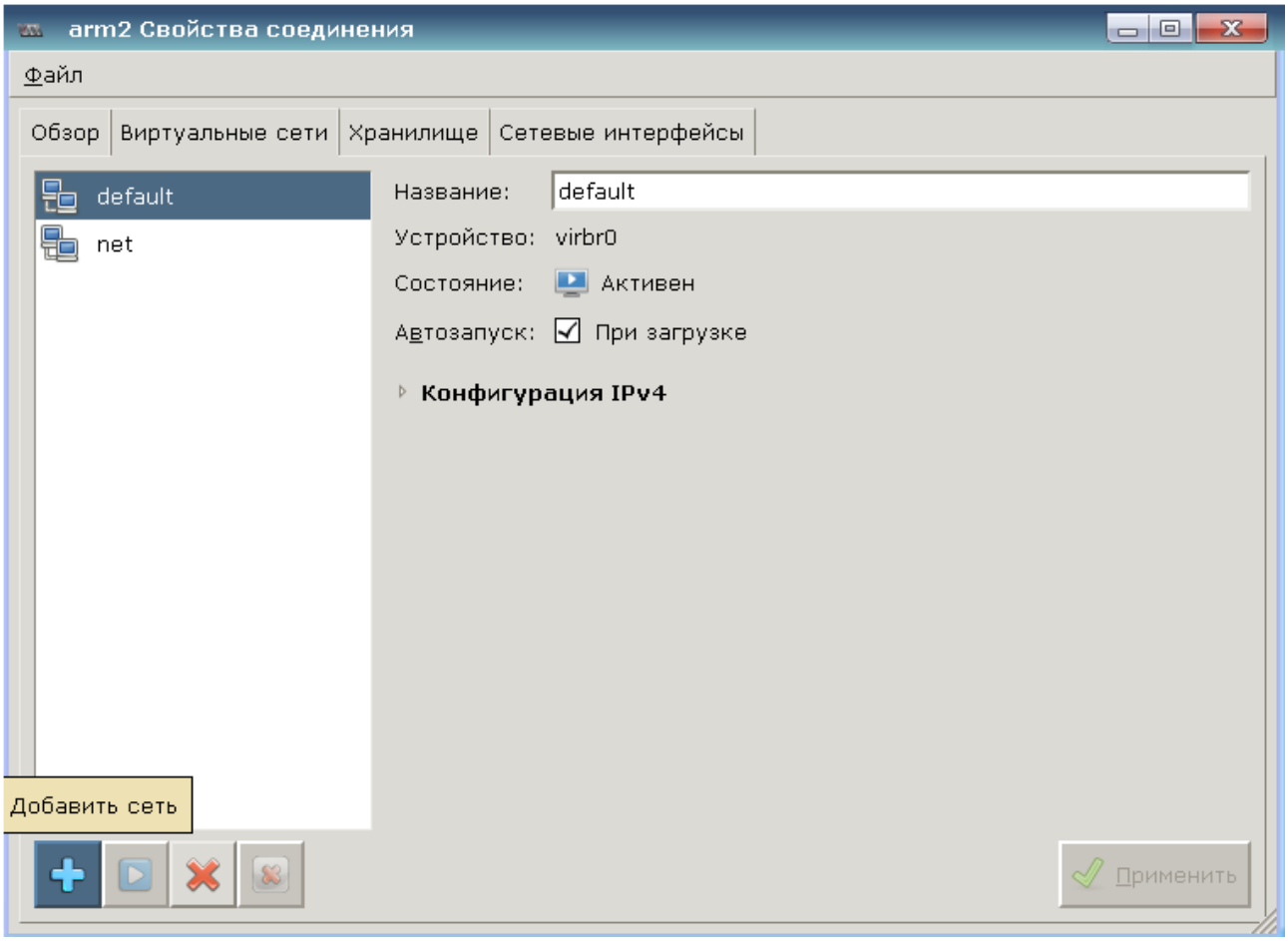


Рис. 18

Для добавления виртуальной сети следует нажать «+», ввести название и задать необходимые настройки: подсеть и подключение к физической сети (рис. 19).

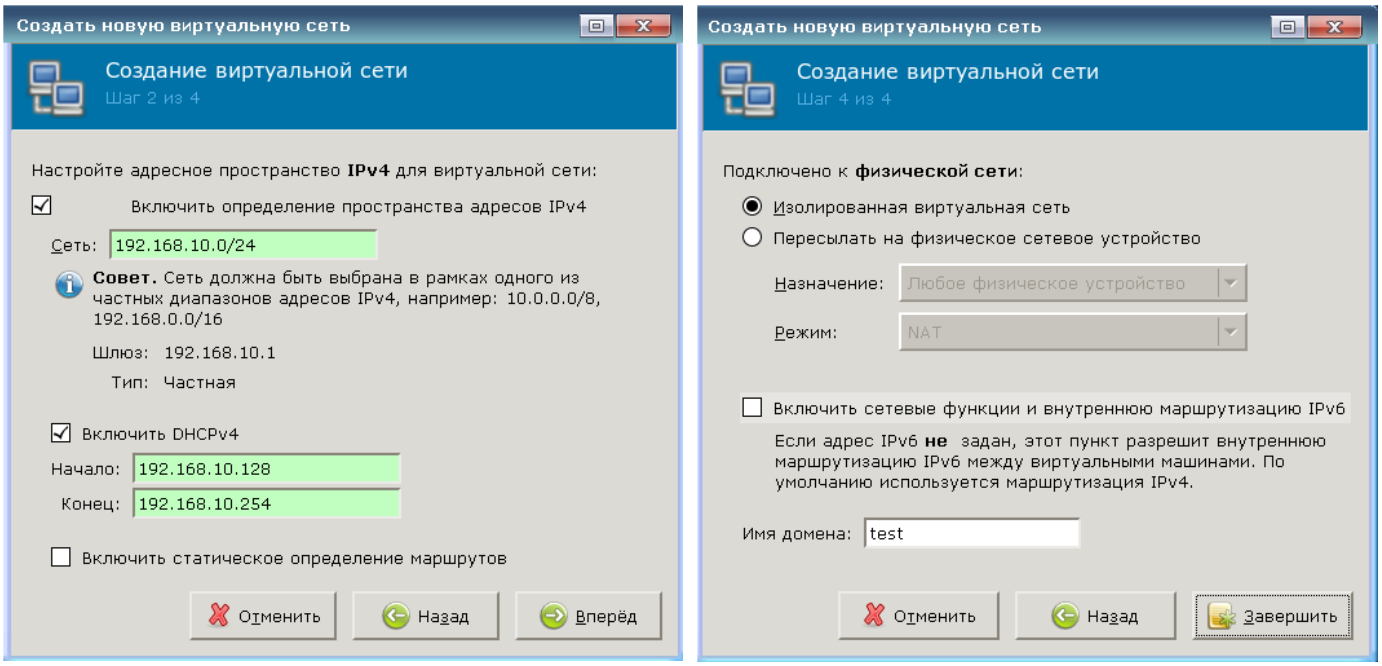


Рис. 19

4.5. Запуск и управление функционированием виртуальных машин

Для запуска виртуальной машины от имени учетной записи пользователя с его мандатными атрибутами безопасности необходимо войти в ОС СН под пользователем с заданным мандатным уровнем. После чего запустить менеджер виртуальных машин `virt-manager`, выбрать виртуальную машину из списка и нажать на кнопку «Запустить» (рис. 20).

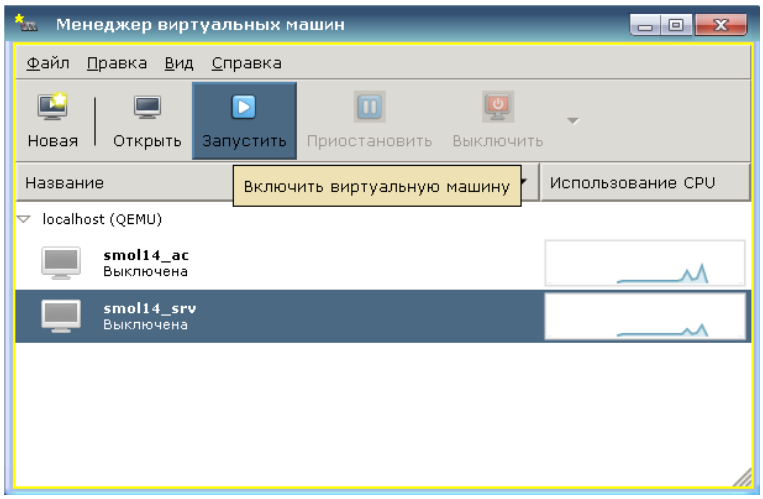


Рис. 20

Примечание. При запуске виртуальная машина приобретает динамические метки безопасности, включающие идентификатор запустившего пользователя и его ман-

датные атрибуты. После этого, при доступе к данной виртуальной машине применяются дискреционное и мандатное разграничение доступа.

ВНИМАНИЕ! Только запустивший виртуальную машину пользователь или пользователь, входящий в группу администраторов `libvirt-admin`, могут управлять функционированием виртуальной машины (приостанавливать, останавливать и т.п.).

4.6. Ограничения по конфигурированию и запуску виртуальных машин в условиях мандатного разграничения доступа

ВНИМАНИЕ! В случае использования виртуальной машины с ненулевым мандатным контекстом в качестве сетевого адаптера не может быть выбрано устройство `virtio`.

Примечание. В случае использования в качестве гостевой системы ОС СН, виртуальная машина не должна запускаться в мандатном контексте. Вместо этого необходимо выполнять удаленный вход с требуемым мандатным уровнем доступа средствами ОС СН.

Примечание. Настоятельно рекомендуется использовать режим «Только чтения» при запуске виртуальных машин в ненулевом мандатном контексте.

Более подробно о функционировании виртуальных мандатное управлении доступом к виртуальным машинам приведено в 8.4.

5. ОТКАЗОУСТОЙЧИВЫЕ РЕШЕНИЯ

При использовании ПК существуют различные возможности по повышению отказоустойчивости функционирования виртуальных машин:

– миграция виртуальных машин между серверами виртуализации (см. 5.1);

5.1. Миграция виртуальных машин между серверами виртуализации

ПК поддерживает возможность миграции виртуальных машин между серверами виртуализации. Миграция позволяет перенести работу виртуальный машины с одного физического хоста на другой без остановки ее работы.

В примере используется два физически разнесенных сервера виртуализации (в дальнейшем - сервер 1 и сервер 2) с установленной на них ОС СН и отдельный NFS-сервер.

Первоначально необходимо выполнить подготовительные действия:

– на каждом сервере виртуализации должен быть установлен комплект приложений ПК и пакет `netcat-openbsd`;

– для настройки автоматического подключения к удаленному серверу виртуализации на серверах виртуализации необходимо сгенерировать и пробросить SSH-ключ. Для этого на каждом сервере из-под пользователя, от имени которого запускается утилита `virt-manager`, требуется выполнить в консоли следующие команды:

```
ssh-keygen -t rsa
ssh-copy-id root@server
```

соответственно заменив `server` на IP-адрес сервера виртуализации (куда кладется ключ).

– На NFS-сервере создать каталог `/srv/nfs4` и дать максимальные права (777). В этом каталоге нужно создавать новые виртуальные машины или клонировать туда существующие. В конфигурационном файле `/etc/exports` необходимо настроить разрешения монтирования (пример для NFSv4):

```
/srv/nfs4 192.168.32.0/24(rw,async,fsid=0,crossmnt,no_subtree_check)
```

и перезапустить nfs-сервер командой:

```
/etc/init.d/nfs-kernel-server restart
```

Для подключения NFS-хранилища необходимо на каждом сервере виртуализации настроить монтирование NFS-хранилища. В первую очередь следует создать точку монтирования (например: `/opt/vm`). Для автоматического монтирования в конфигурационный файл `/etc/fstab` добавляется следующая строка:

```
server:/ /opt/vm nfs rw  0   0
```

где `server` является IP-адресом другого сервера виртуализации, т. е. в конфигурационный

`/etc/fstab` на сервере 1 вписать:

```
сервер2:/ /opt/vm  nfs rw 0 0
```

Для подключения второго сервера виртуализации в утилите `virt-manager` необходимо выбрать пункт меню «Файл->Добавить соединение» и заполнить поля в появившемся окне:

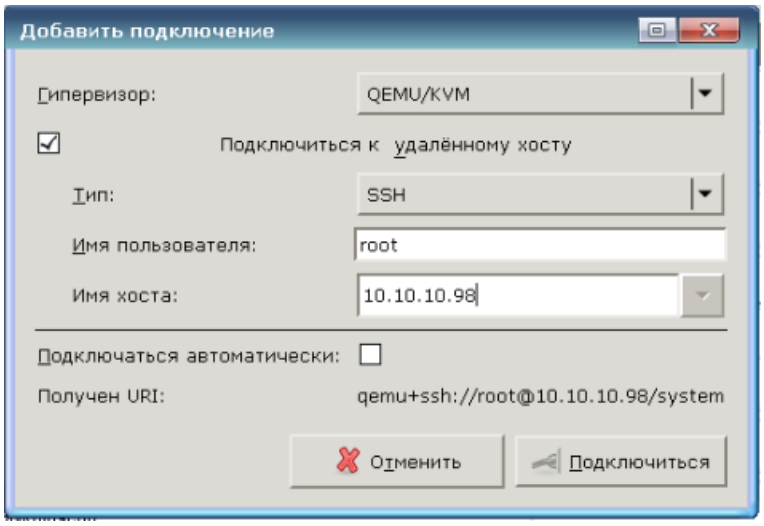


Рис. 21

После этого, при проброшенном ключе SSH, произойдет успешное подключение к серверу виртуализации второго узла и будут доступны его виртуальные машины.

Живая миграция позволяет перенести работу виртуальный машины с одного физического хоста на другой без остановки ее работы. При этом образ диска виртуальной машины должен быть доступен на обоих серверах виртуализации. Для этого используется NFS-хранилище. На обоих серверах виртуализации NFS-хранилище должно монтироваться в одно и тоже место, образ виртуальной машины должен быть расположен в этом хранилище.

ВНИМАНИЕ! Необходимым условием надежной миграции является отсутствие для заданной виртуальной машины функции использования кеширования ОС при работе с дисковой подсистемой. Для этого в свойствах виртуальной машины в параметрах кеширования используемых файлов-образов носителей следует указать значение `none`.

Для миграции машины с помощью утилиты `virt-manager` необходимо кликнуть правой кнопкой мыши на нужной виртуальной машине (она должна быть запущена) и выбрать в меню пункт «Миграция». В появившемся окне выбрать конечный узел и нажать на кнопку «Миграция». При этом конфигурационный файл перемещаемой машины не переходит на новый узел, поэтому при ее выключении она вновь появится на старом хосте. В связи с этим, для совершения полной живой миграции, при котором конфигурация виртуальной машины будет перемещена на новый узел, необходимо воспользоваться утилитой командной строки `virsh`, заменив имя виртуальной машины `vm_name` и адрес конечного

хоста (10.0.0.2):

```
virsh qemu:///system migrate --live --persistent --undefinesource \
 vm_name qemu+ssh://10.0.0.2/system
```

6. СЛУЖБЫ

В состав ПК входят следующие основные службы:

– служба сервера виртуализауии `libvirtd`;

– служба предоставления доступа к сетевой защищенной файловой системе Samba.

6.1. Служба сервера виртуализауии `libvirtd`

Служба сервера виртуализации `libvirtd` предоставляет возможность удаленного управления сервером виртуализации по сети с использованием различных протоколов и способов аутентификации. При этом поддерживается возможность решения всех задач по созданию и учету виртуальных машин, настройке их конфигурации и непосредственно запуска.

Для взаимодействия других программ с сервером виртуализации могут использоваться дополнительные программные интерфейсы из пакетов с префиксом `libvirt-`.

Доступ к службе сервера виртуализации возможен как с помощью локальных Unix-сокетов, так и по сети с помощью консольных или графических инструментов управления виртуальными машинами.

Настройка сервера виртуализации приведена в 3.3.

6.2. Служба предоставления доступа к сетевой защищенной файловой системе Samba

Служба предоставления доступа к сетевой защищенной файловой системе Samba является модифицированной версией службы из состава ОС СН.

Описание службы, ее настройки и особенности функционирования приведены в документе РУСБ.10015-07 95 01-1 «Операционная система специального назначения «Astra Linux Special Editionю Руководство администратора. Часть 1».

Для организации доступа к разделяемым каталогам сетевой защищенной файловой системы Samba/CIFS из гостевых операционных систем предусмотрен дополнительный конфигурационный параметр в конфигурационном файле `smb.conf`:

```
use socket MAC label = yes
```

Данный параметр обеспечивает определение мандатного контекста по мандатной метке входящего соединения.

7. УТИЛИТЫ КОМАНДНОЙ СТРОКИ

В состав ПК входят следующие утилиты командной строки:

– `qemu-img` — утилита работы с образами дисков виртуальной машины QEMU: позволяет выполнять операции по созданию образов различных форматов, конвертировать файлы-образы между этими форматами, получать информацию об образах и объединять снимки виртуальных машин для тех форматов, которые это поддерживают;

– `qemu-io` — утилита диагностирования файлов-образов виртуальных носителей QEMU;

– `qemu-nbd` — утилита экспортирования файлов-образов виртуальных носителей QEMU с использованием протокола NBD (Network Block Device — сетевого блочного устройства);

– `qemu-system-i386` — средство эмуляции 32-х разрядной платформы x86;

– `qemu-system-x86_64` — средство эмуляции 64-х разрядной платформы x86;

– `virsh` — консольный интерфейс управления виртуальными машинами;

– `virt-clone` — утилита клонирования виртуальных машин;

– `virt-convert` — утилита конвертирования виртуальных машин между различными форматами и программно-аппаратными платформами;

– `virt-host-validate` — утилита проверки конфигурации хоста сервера виртуализации;

– `virt-image` — утилита создания виртуальных машин по их XML описанию;

– `virt-install` — утилита создания виртуальных машин с помощью опций командной строки;

– `virt-pki-validate` — утилита проверки корректности настройки системы PKI (сертификатов) сервера виртуализации libvirt;

– `virt-sanlock-cleanup` — утилита удаления устаревших файлов блокировки виртуальных машин;

– `virt-xml` — утилита редактирования XML описаний виртуальных машин с помощью опций командной строки;

– `virt-xml-validate` — утилита проверки корректности XML описаний виртуальных машин.

Возможности и способы использования перечисленных консольных утилит описаны в соответствующих разделах руководства `man`.

8. ВЗАИМОДЕЙСТВИЕ АДМИНИСТРАТОРА С СЗИ

Взаимодействие администратора с СЗИ ПК состоит из обязательного прохождения процедуры аутентификации на сервере виртуализации libvirt и работы в условиях применения дискреционных и мандатных правил разграничения доступа.

Основными задачами администрирования являются:

– настройка удаленного доступа к серверу виртуализации libvirt (см. 8.1);

– настройка удаленного доступа к рабочим столам виртуальных машин по протоколам VNC и Spice (см. 8.2);

– дискреционное управление доступом к виртуальным машинам (см. 8.3);

– мандатное управление доступом к виртуальным машинам (см. 8.4);

– управление режимом запрета модификации файлов-образов виртуальных машин (см. 8.5).

ВНИМАНИЕ! Операции по изменению состава или конфигурации виртуальных машин требуют вхождения пользователя в специальную локальную административную группу `libvirt-admin`.

8.1. Идентификация и аутентификация при доступе к серверу виртуализации libvirt

На рис. 22 приведен снимок диалогового окна задания параметров подключения к серверу виртуализации libvirt.

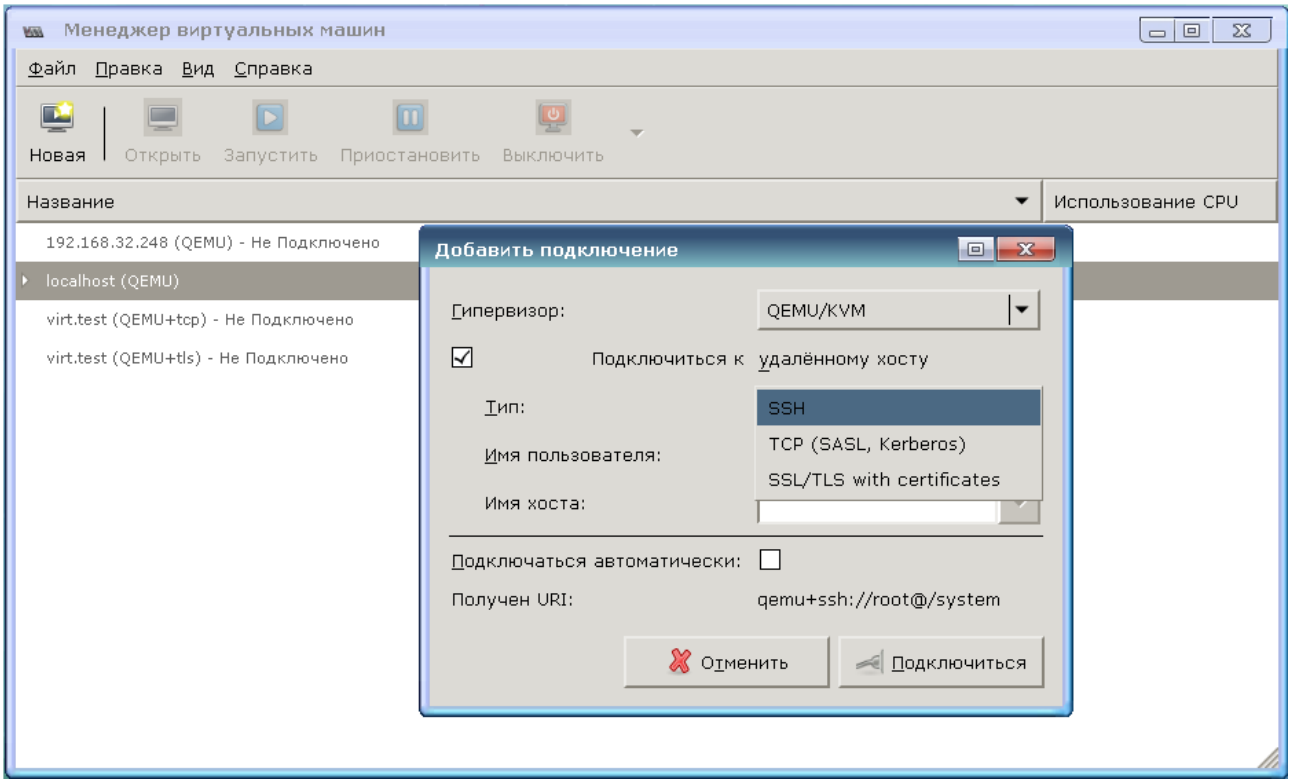


Рис. 22

Сервер виртуализации может использовать для идентификации и аутентификации клиентов следующие механизмы:

– локальная peer-cred аутентификация;

– удаленная SSH аутентификация (строка соединения `qemu+ssh://...`);

– удаленная SASL аутентификация в том числе с поддержкой Kerberos (строка соединения `qemu+tcp://...`);

– удаленная TLS аутентификация с использованием сертификатов (строка соединения `qemu+tls://...`).

Параметры аутентификации задаются в конфигурационном файле `/etc/libvirt/libvirtd.conf`. В конфигурационном файле отдельно задаются параметры для различных способов аутентификации: параметры локальных UNIX сокетов (секция «UNIX socket access control»), разрешение приема сетевых соединений (параметры `listen_tls` и `listen_tcp`) и порты для tcp и tls (параметры `tls_port` и `tcp_port`), расположение необходимых файлов при использовании сертификатов x509 (секция «TSL x509 certificate configuration») и варианты авторизации (параметры `auth_unix_ro`, `auth_unix_rw`, `auth_tcp`, `auth_tls`).

По умолчанию используется следующее расположение файлов сертификатов на сервере для аутентификации к серверу виртуализации libvirt:

– `/etc/pki/CA/cacert.pem` — корневой сертификат;

– `/etc/pki/CA/crl.pem` — файл отозванных сертификатов;

– `/etc/pki/libvirt/servercert.pem` — сертификат открытого ключа сервера виртуализации libvirt;

– `/etc/pki/libvirt/private/serverkey.pem` — закрытый ключ сервера виртуализации libvirt.

Примечание. Файлы ключей сервера виртуализации libvirt должны быть доступны на чтение группе `libvirt-qemu`.

По умолчанию используется следующее расположение файлов сертификатов на клиенте для аутентификации к серверу виртуализации libvirt (в домашнем каталоге пользователя ~):

– `/etc/pki/CA/cacert.pem` — корневой сертификат;

– `~/.pki/libvirt/clientcert.pem` — сертификат открытого ключа клиента;

– `~/.pki/libvirt/clientkey.pem` — закрытый ключ клиента.

В случае SASL аутентификации используется конфигурационный файл `/etc/sasl2/libvirt.conf`, в котором задаются параметры аутентификации SASL (например, применяемые механизмы). Имя сервиса сервера виртуализации libvirt при использовании SASL аутентификации регистрируется как `libvirt/<имя сервера>@<домен>`.

ВНИМАНИЕ! При указании механизма SASL `gssapi` следует в конфигурационном файле `/etc/default/libvirtd` указать с помощью соответствующей переменной окружения расположение файла ключей Kerberos сервера виртуализации, например:

`export KRB5_KTNAME=/etc/libvirt/libvirt.keytab`.

Примечание. Настройка сервера виртуализации для работы в ЕПП ОС СН производится в соответствием с документом РУСБ.10015-07 95 01-1 «Операционная система специального назначения «Astra Linux Special Edition Руководство администратора. Часть 1».

8.2. Идентификация и аутентификация при доступе к рабочему столу виртуальных машин

Параметры аутентификации при доступе к рабочему столу виртуальной машины задаются в конфигурационном файле `/etc/libvirt/qemu.conf` раздельно для протоколов VNC и Spice. В данном конфигурационном файле указываются необходимые параметры аутентификации и пути к конфигурационным файлам SASL, например `/etc/sasl2/qemu.conf`. Имя сервисов VNC и Spice при использовании SASL аутентификации регистрируется как `vnc/<имя сервера>@<домен>` и `spice/<имя сервера>@<домен>`, соответственно.

Примечание. Настройка сервисов VNC и Spice для работы в ЕПП ОС СН производится в соответствии с документом РУСБ.10015-07 95 01-1.

По умолчанию используется следующее расположение файлов сертификатов на сервере для аутентификации к виртуальной машине для по протоколу VNC:

– `/etc/pki/libvirt-vnc/ca-cert.pem` — корневой сертификат;

– `/etc/pki/libvirt-vnc/server-cert.pem` — сертификат открытого ключа сервера VNC QEMU;

– `/etc/pki/libvirt-vnc/server-key.pem` — закрытый ключ сервера VNC QEMU.

Примечание. Файлы ключей сервера VNC QEMU должны быть доступны на чтение группе `libvirt-qemu`.

По умолчанию используется следующее расположение файлов сертификатов на клиенте для аутентификации к серверу VNC QEMU (в домашнем каталоге пользователя ~):

– `/etc/pki/CA/cacert.pem` — корневой сертификат;

– `~/.pki/libvirt-vnc/clientcert.pem` — сертификат открытого ключа клиента;

– `~/.pki/libvirt-vnc/private/clientkey.pem` — закрытый ключ клиента.

8.3. Дискреционное управление доступом к виртуальным машинам

При взаимодействии с сервером виртуализации libvirt осуществляется дискреционное разграничение доступа к управлению виртуальными машинами.

Разграничение выполняется драйвером доступа `parsec`, специально разработанного с использованием прикладного программного интерфейса драйверов доступа libvirt.

Основанием для принятия решения о предоставлении доступа является сравнение дискреционных атрибутов виртуальной машины и дискреционных атрибутов пользователя с учетом выполняемой операции.

Все операции с виртуальной машиной разделяются на непривилегированные и привилегированные, например: операции получения информации о списке машин или о конфигурации конкретной машины являются операциями непривилегированными, а операции создания, удаления или изменения конфигурации виртуальных машин – привилегированными.

Операции по изменению состава или конфигурации виртуальных машин требуют вхождения пользователя в специальную локальную административную группу. Имя административной группы задается в конфигурационном файле `/etc/libvirt/libvirtd.conf` параметром:

```
admin_group = "libvirt-admin"
```

Для каждой виртуальной машины задается список контроля доступа, в котором указываются субъекты доступа (пользователи и группы), обладающие доступом к виртуальной машине. Различаются три типа доступа к виртуальной машине (см. рис. 23):

– «Просмотр свойств» — видимость виртуальной машины в списке и просмотр ее свойств;

– «Использование» — просмотр свойств, запуск и работа с виртуальной машиной;

– «Администрирование» — полный доступ к виртуальной машине, включая запуск, правку ее свойств и управление правами доступа к ней.

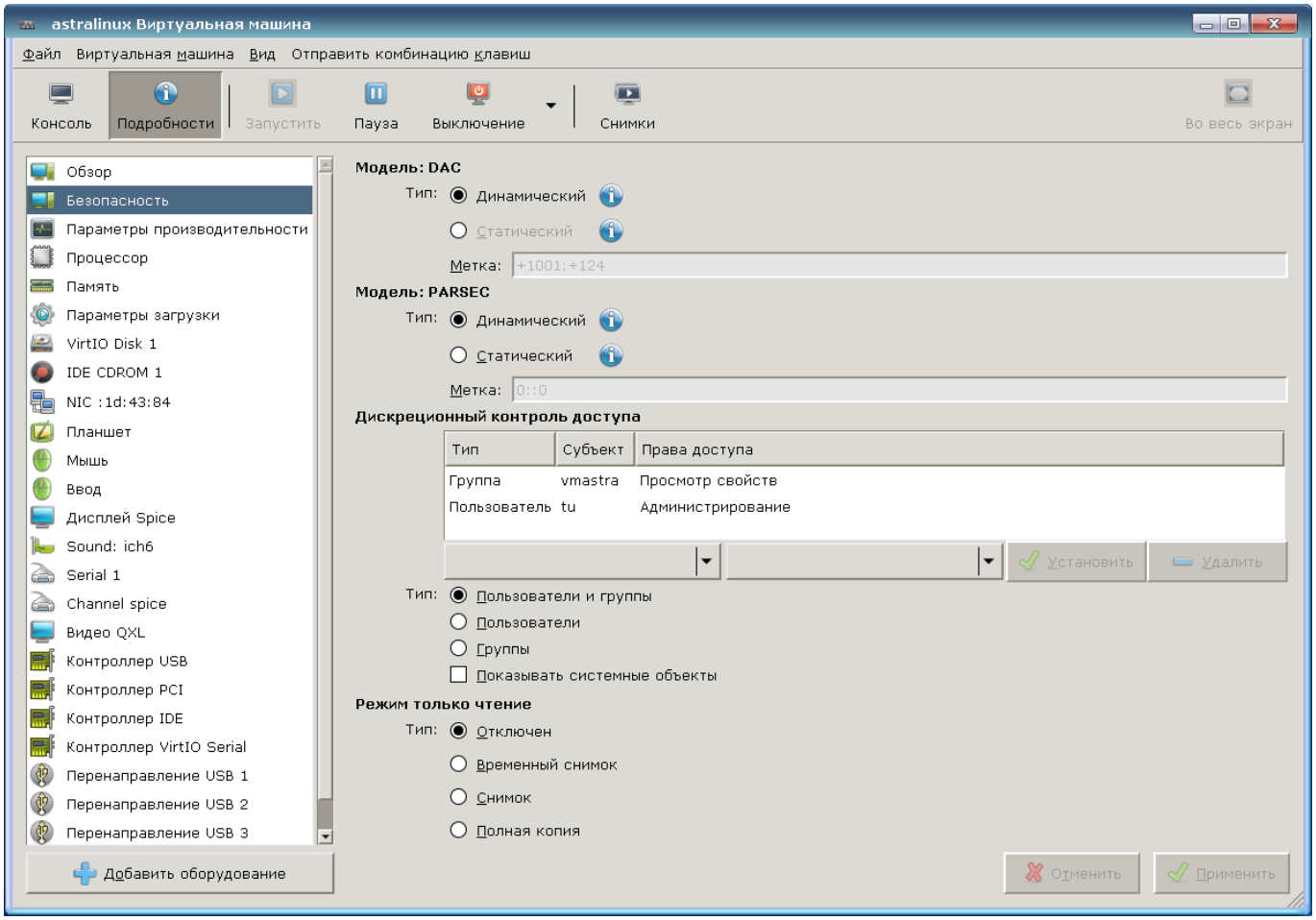


Рис. 23

Пользователи видят в списке виртуальных машин только те виртуальные машины, к которым им явно предоставлен доступ.

После создания виртуальной машины список контроля доступа пуст и доступ к виртуальной машине разрешен только членам административной группы. При наличии права использования пользователям позволяется запускать виртуальную машину и работать с ней. При этом, завершить работу или произвести иные разрешенные действия с виртуальной машиной может только пользователь, который ее запустил. Члены административной группы могут завершать работу виртуальных машин других пользователей.

В момент запуска виртуальной машины модуль поддержки дискреционного разграничения доступа `dac` устанавливает виртуальной машине владельцем запускающего пользователя, а группой – группу `libvirt-qemu`. Это отражается в виде наличия динамической метки безопасности модели `DAC` и идентификатора владельца как процесса виртуальной машины в ОС СН, так и необходимых устройств и файл-образов, принадлежащих виртуальной машине. При последующих операциях эта информация используется для проверки дискреционного доступа к виртуальной машине.

8.4. Мандатное управление доступом к виртуальным машинам

При взаимодействии с сервером виртуализации libvirt осуществляется мандатное разграничение доступа к управлению виртуальными машинами.

Разграничение выполняется драйвером доступа `parsec`, специально разработанного с использование прикладного программного интерфейса драйверов доступа libvirt.

Основанием для принятия решения о предоставлении доступа является сравнение метки безопасности виртуальной машины и мандатного уровня доступа пользователя с учетом выполняемой операции.

Все операции с виртуальной машиной разделяются на операции чтения и записи, например: операции получения информации о списке машин или о конфигурации конкретной машины являются операциями чтения, а операции создания, удаления или изменения конфигурации виртуальных машин – записи. Мандатное разграничение доступа осуществляется следующим образом:

– остановленная виртуальная машина не обладает мандатными атрибутами (если для нее не задана статическая метка безопасности модели `PARSEC`);

– запущенная виртуальная машина наследует мандатную метку запускающего пользователя;

– доступ к функционирующей виртуальной машине предоставляется только при равенстве мандатного уровня доступа пользователя мандатной метке виртуальной машины;

– доступ к получению информации о виртуальной машине и ее конфигурации предоставляется в соответствии с мандатным уровнем пользователя.

В момент запуска виртуальной машины модуль поддержки мандатного разграничения доступа `parsec` устанавливает виртуальной машине мандатный уровень, определенный по соединению запускающего пользователя. Это отражается в виде наличия динамической метки безопасности модели `PARSEC` и мандатной метки как процесса виртуальной машины в ОС СН, так и необходимых устройств и файл-образов, принадлежащих виртуальной машине. При последующих операциях эта информация используется для проверки мандатного доступа к виртуальной машине.

Существует возможность задания статической метки безопасности модели `PARSEC`. В этом случае виртуальная машины может быть запущена только под заданным мандатным уровнем доступа.

ВНИМАНИЕ! Существуют ограничения по конфигурированию виртуальной машины: в качестве сетевого адаптера не может быть выбрано устройство `virtio`.

Примечание. В случае использования в качестве гостевой системы ОС СН, виртуальная машина не должна запускаться в мандатном контексте. Вместо этого необходимо

выполнять удаленный вход с требуемым мандатным уровнем доступа средствами ОС СН.

Примечание. Настоятельно рекомендуется использовать режим «Только чтение» при запуске виртуальных машин в ненулевом мандатном контексте.

8.5. Режимы запрета модификации файлов-образов виртуальных машин

В некоторых случаях требуется обеспечение неизменности файлов-образов виртуальной машины в процессе ее функционирования, что позволяет выполнять повторный запуск заранее подготовленных виртуальных машин из фиксированного состояния. В этом случае все результаты работы пользователя после завершения функционирования виртуальной машины удаляются.

В ПК такой режим работы называется «Режим только чтение» и достигается, как представлено на рис. 23, тремя разными способами:

– временный снимок — режим функционирования средства эмуляции аппаратного обеспечения QEMU, при котором основной файл-образ защищается от записи, а все результаты работы пользователя фиксируются во временном снимке, существующем только в процессе функционирования виртуальной машины;

– снимок — создание во время запуска виртуальной машины в заданном каталоге снимка, удаляемого после завершения функционирования виртуальной машины;

– полная копия — создание во время запуска виртуальной машины в заданном каталоге полной копии файла-образа, удаляемого после завершения функционирования виртуальной машины;

Вариант создания полной копии требует значительного времени запуска виртуальной машины за счет копирования файла-образа, который может обладать большим размером. Но данный вариант позволяет использовать возможности по сохранению состояния виртуальной машины для последующего его восстановления. При использовании снимков любое выключение виртуальной машины вследствие выключения или аппаратного сбоя сервера виртуализации, все результаты работы виртуальной машины будут потеряны.

Каталог размещения временных файлов задается в конфигурационном файле `/etc/libvirt/qemu.conf` следующим конфигурационным параметром:

```
run_images_dir = "/var/lib/libvirt/runimages"
```

9. ТЕСТИРОВАНИЕ ЗАПУСКА ВИРТУАЛЬНОЙ МАШИНЫ

Для проведения автоматического тестирования корректности функционирования ПК может быть использован следующий сценарий:

```
 #!/bin/bash

red=$(tput setaf 1)
green=$(tput setaf 2)
coff=$(tput sgr0)

cause1="${green}Мандатный уровень 0 верен${coff}\n"
cause2="${green}UID верен${coff}\n"
cause3="${green}Мандатный уровень 1 верен${coff}\n"
cause4="${green}UID верен${coff}\n"

test_result="${green}TEST PASS${coff}\n"

user=$(whoami) # user name
uid=$(id -u) # user uid
macid=$(pdp-id -l) # user mac_level

# check libvirt server installed
dpkg -l | grep libvirt-bin &> /dev/null
[ $? != 0 ] && { echo -e "\nlibvirt server not installed\n"; exit 10; }

# check users group membership
group_file="/etc/group"
for group in libvirt: libvirt-admin: astra-admin:
    do
        grep $group $group_file | grep $user &> /dev/null
            if [ $? != 0 ]; then
                echo -e "\nВы должны входить в группу $group\n"; exit 11;
            fi
    done

# check available mac level
mac_path="/etc/parsec/macdb"
ls $mac_path/$uid &> /dev/null
[ $? != 0 ] && { echo -e "Задайте пользователю мандатный уровень > 0"; exit 11; }
```

```
max_lvl=$(awk -F : '{print$4}' $mac_path/$uid)
[ $max_lvl == 0 ] && \
  { echo -e "Задайте пользователю мандатный уровень > 0"; exit 11; }

# set default network up
virsh --connect qemu:///system net-autostart default > /dev/null

# help message
help_me=$(cat <<EOF
  Порядок использования: `basename $0` <name>
EOF
)

# args check
[ $# -eq 0 ] && { echo -e "${help_me}"; exit 3; }  # script with no args?

# create and start test vm
vm_name=$1
mem="16"
vcpus="1"
os_type="linux"

virt-install --connect qemu:///system -n $vm_name -r $mem --vcpus=$vcpus \
--boot hd -q --disk pool=default,size=1,bus=virtio,format=qcow2 \
--os-type=$os_type &> /dev/null &

sleep 5

# PID mac level definition
pid="$(ps uanx | grep $vm_name | grep qemu-system | awk '{print $2}')"
puid="$(ps uanx | grep $vm_name | grep qemu-system | awk '{print $1}')"
pid_mac_lvl=$(sudo pdp-pslbl -n $pid | awk '{print$2}'| cut -d ':' -f1)

if [ $pid_mac_lvl != $macid ];
    then test_result="${red}TEST FAIL${coff}\n"
    cause1="${red}Уровень процесса не равен уровню пользователя (0)${coff}\n"
elif [ -z $pid_mac_lvl ];
```

```
    then test_result="${red}TEST FAIL${coff}\n"
    cause1="${red}Что-то пошло не так...${coff}\n"
fi

if [ $uid != $puid ];
    then test_result="${red}TEST FAIL${coff}\n"
    cause2="${red}UID не соответствует пользователю${coff}\n"
fi

kill $pid

sleep 5

# start with mac level 1
sumac -x -l 1 '/usr/bin/virsh -c qemu:///system start '$vm_name''

sleep 5

# PID mac level definition
pid_lvl="$(sudo ps uanx | grep $vm_name | grep qemu-system | awk '{print $2}')"
puid_lvl="$(sudo ps uanx | grep $vm_name | grep qemu-system | awk '{print $1}')"
pid_mac_lvl_1=$(sudo pdp-pslbl -n $pid_lvl | awk '{print$2}'| cut -d ':' -f1)

if [ $pid_mac_lvl_1 != 1 ];
    then test_result="${red}TEST FAIL${coff}\n"
    cause3="${red}Уровень процесса не равен уровню пользователя (1)${coff}\n"
elif [ -z $pid_mac_lvl_1 ];
    then test_result="${red}TEST FAIL${coff}\n"
    cause3="${red}Что-то пошло не так...${coff}\n"
fi

if [ $uid != $puid_lvl ];
    then test_result="${red}TEST FAIL${coff}\n"
    cause4="${red}UID не соответствует пользователю${coff}\n"
elif [ -z $uid ];
    then test_result="${red}TEST FAIL${coff}\n"
    cause4="${red}Что-то пошло не так...${coff}\n"
fi
```

```
sudo kill $pid_lvl

sleep 5

# cleaning
virsh  --connect qemu:///system undefine $vm_name \
--remove-all-storage --wipe-storage > /dev/null

# result
echo
printf "Проверка под нулевым мандатным уровнем:\n"
printf "$cause1"
printf "$cause2"
printf "Проверка под мандатным уровнем 1:\n"
printf "$cause3"
printf "$cause4"
printf "Результат проверки:\n"
printf "$test_result"

exit 0
```

Сценарий позволяет выполнить проверку корректности запуска виртуальной машины — процесс виртуальной машины должен наследовать `UID` пользователя и мандатные атрибуты его сессии. При неуспешном завершении тестирования для блокировки доступа к виртуальным машинам сценарий может быть расширен командами остановки служб и процессов ПК.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ЕПП — единое пространство пользователей

ОС СН — операционная система специального назначения

ПК — программный комплекс

СЗИ — средства защиты информации

ФС — файловая система

ALD — Astra Linux Directory (единое пространство пользователей)

KVM — Kernel-based Virtual Machine (программное решение, обеспечивающее виртуализацию в среде Linux на платформе, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine))

QEMU — Quick Emulator (средства эмуляции аппаратного обеспечения)

VDI — Virtual Desktop Infrastructure (инфраструктура виртуальных рабочих столов)

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	заменённых	новых	аннулированных					