УТВЕРЖДЕН

ТАСП.62.01.11.000.002 32 01-ЛУ

ПРОГРАММНЫЙ КОМПЛЕКС

«СЕРВЕРНАЯ ОПЕРАЦИОННАЯ СИСТЕМА»

ПК «СинтезМ-Сервер»

Руководство системного программиста

ТАСП.62.01.11.000.002 32 01

Листов 121

2017

Литера__

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

АННОТАЦИЯ

Настоящий документ является руководством системного программиста программного комплекса «Серверная операционная система» (далее по тексту – изделие или ПК «СинтезМ-Сервер») и содержит сведения о структуре изделия, правила установки и настройки изделия.

СОДЕРЖАНИЕ

ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1. Назначение программы

Изделие представляет собой серверную операционную систему, предназначенную для обеспечения функционирования серверов x86_64, виртуальных машин, менеджера ВМ, терминального сервера, терминальных клиентов, пользователей автоматизированной системы в защищенном исполнении, серверов приложений, системы управления базами данных, системы хранения данных и подсистемы защиты информации. Изделие создает среду виртуализации, поддерживающую в качестве гостевых защищенных операционных систем «СинтезМ», Windows, Linux, а также обеспечивает изоляцию виртуальных машин и управляет всем жизненным циклом виртуальных машин.

1.2. Функции программы

ПК «СинтезМ-Сервер» предназначен для выполнения следующих функций:

– создания среды виртуализации на базе дистрибутива Linux с открытым исходным кодом с использованием технологии KVM, поддерживающей в качестве гостевых операционных систем ПК «Синтез-Клиент» из состава ЗОС «Синтез» версия 6.4, ПК «СинтезМ-Клиент» из состава ЗОС «СинтезМ», ОС семейства Windows и ОС семейства Linux;

– создания доверенной среды виртуализации с изоляцией виртуальных машин на основе дистрибутива Linux с открытым исходным кодом с использованием технологии KVM;

– обеспечения функционирования средств конфигурирования сетевого взаимодействия между виртуальными машинами и физическими серверами за счет применения Linux Bridge и фильтров ebtables;

– предоставления консольного доступа к виртуальным машинам посредством протоколов SPICE и VNC;

– подключения USB – и audio-устройств посредством протокола SPICE в виртуальные машины с тонких клиентов и рабочих станций пользователя;

– подключения PCI–устройств в виртуальные машины;

– подключения COM–портов в виртуальные машины;

– подключения к системам хранения данных по следующим протоколам: NFS, iSCSI, FC;

– функционирования сервера управления ВМ под управлением ПК «СинтезМ-Сервер» в виртуальной машине в режиме высокой доступности посредством технологии Self-Hosted Engine;

– поддержки настраиваемых посредством ПС «СинтезМ-АБ» механизмов защиты информации (идентификации и аутентификации; разграничения доступа; управление конфигурируемым сетевым взаимодействием, регистрации и учета; контроля целостности; антивирусной защиты; самотестирование компонентов ПСЗИ; защиты от НСД).

– поддержки работы операционной системы с пакетом OpenSSL, функционирующим с применением алгоритмов криптографического преобразования ГОСТ 28147-89, ГОСТ 34.10-2012;

– поддержки функции ядра IPv4-маршрутизации;

– запуска, остановки и отслеживания загруженности серверов виртуализации;

– балансировки нагрузки между серверами виртуализации с обеспечением равномерного распределения виртуальных машин между серверами виртуализации и запуска новых ВМ на наименее нагруженных серверах виртуализации;

– создания и удаления виртуальных машин;

– запуска, остановки виртуальных машин;

– миграции виртуальных машин с одного физического сервера на другой без прекращения работы виртуальной машины и остановки сервисов;

– создания для виртуальных машин режима высокой доступности;

– загрузки бездисковых терминалов (тонких клиентов) посредством использования среды загрузки PXE с помощью сетевой карты тонкого клиента;

– функционирования бездисковых терминалов и возможности их подключения к серверам виртуализации по протоколу SPICE для обеспечения работы консольных сессий в виртуальных машинах;

– поддержки кластеризации;

– поддержки синхронизации с сервером единого времени;

– поддержки файловых систем (NTFS, Ext2, Ext3, Ext4, Xfs, Vfat (fat32)).

– обеспечения взаимодействия с программным обеспечением, функционирующим на рабочей станции под управлением ПК «Синтез-Клиент» () из состава ЗОС «Синтез» версия 6.4, ПК «СинтезМ-Клиент» (версия 7.2) из состава ЗОС «СинтезМ».

1.3. Функциональные ограничения на применение

Для установки серверов виртуализации и авторизации необходима физическая серверная платформа. Количество серверов, наличие в системе СХД, уточняется заказчиком исходя из нагрузки на серверы и необходимости горячего резервирования данных серверов, а также не противоречит концепции создания доверенной среды функционирования АСЗИ.

Для установки и настройки изделия понадобятся АРМ администратора с установленной Linux-подобной операционной системой, коммутатор для сетевого взаимодействия.

На серверах необходимо настроить RAID-контроллеры (во время установки использовался RAID5) и консоли удалённого управления серверами.

Организационно-техническими мерами, из состава технических средств, на которые осуществляется загрузка образа терминального клиента, должны быть исключены устройства постоянного хранения информации. Устройства чтения/записи оптических дисков не должны быть установлены на технических средствах непривилегированных пользователей.

Организационно-техническими и техническими мерами должен исключаться несанкционированный доступ пользователей в помещения, где размещены технические средства, функционирующие под управлением ЗОС «СинтезМ».

Организационно техническими мерами должно быть исключено функционирование более одного DHCP-сервера и более одного терминального сервера в составе каждой подсети, в рамках которой функционирует ЗОС «СинтезМ».

Организационно-техническими мерами должно быть исключено несанкционированное подключение к ЛВС, в составе которой работают технические средства, функционирующие под управлением ЗОС «СинтезМ».

Организационными мерами и программными средствами (ПС «СинтезМ-АБ») должен быть исключен доступ непривилегированных пользователей к средствам администрирования и управления, в том числе к текстовым консолям ОС и эмуляторам терминала. На этапе штатной эксплуатации АЗСИ не допускается изменение непривилегированными пользователями, а также администратором безопасности конфигурационных файлов. Права доступа непривилегированных пользователей должны быть установлены таким образом, чтобы пользователю не были доступны на запись следующие файлы:

- конфигурационные файлы загрузчиков;
- /selinux/enforce;
- /etc/SELinux/config;
- /etc/SELinux/targeted/Booleans;
- /etc/selinux/targeted/booleans.local;
- /etc/passwd;
- /etc/default/useradd;
- /etc/group;
- /etc/shadow;

– /etc/skel.

Доведение и применение политик на ВМ СУБД должно осуществляться после развертывания ВМ и до эксплуатации данной ВМ.

Запрещается начало эксплуатации БД, если после развертывания БД первое доведение и применение политик не завершилось успешно.

1.3.1. Ограничения для совместно функционирующего ПО

1. Запрещается реализация в совместно функционирующем ПО функций DHCP-сервера и PXE-сервера.

2. Запрещается реализация в совместно функционирующем ПО функционала по модификации данных, хранящихся в ИПА, за исключением создания служебных учетных записей пользователей (не соответствующих пользователям Системы) и их настройке, осуществляемых в процессе развертывания ПО.

3. Запрещается реализация функционала по созданию локальных учетных записей ОС, за исключением создания учетных записей технологических пользователей, осуществляемого на этапе развертывания ПО.

4. Для ПО, функционирующего под учетной записью привилегированного пользователя, запрещается изменять следующие файловые объекты:

– все файлы в директории /var/lib/aide/sintezm;
– файл /etc/init.d/aide_startup;
– файл counthash;
– файл /etc/aide.startup.conf;
– файл /etc/cron.d/counthash_timetable;
– файл /etc/aide.conf;
– файл /usr/bin/file_tree;
– файл /etc/sintezm/integrity_client.conf;
– файл /etc/passwd;

– файл /etc/default/useradd;
– файл /etc/group;
– файл /etc/shadow;
– файл /etc/skel;
– файл /etc/selinux/targeted/seusers;
– файлы в директории /etc/selinux/targeted/contexts;
– файл /var/www/Sintezm-Repo/web/.htaccess;
– файл /etc/httpd/conf.d/z-ovirt-engine-proxy.conf;
– файл /etc/aide.user-group.conf.

5. Привилегированному пользователю запрещается изменение параметра BIOS, содержащего описание источника загрузки образа ядра.

6. Привилегированному пользователю запрещается изменение образа ядра в существующем источнике.

7. Запрещается осуществлять запуск скрипта counthash.

8. Запрещается осуществлять изменение настроек запуска скрипта /etc/init.d/aide_startup посредством удаления и/или модификации ссылок на этот файл в подпапках папки /etc/rc.d/.

9. Запрещается реализация функционала по изменению таблицы системных вызовов ядра ОС.

10. Запрещается реализация функционала по передаче логина и пароля для получения TGT-билета Kerberos или подключения к Серверу управления виртуальными машинами.

11. Запрещается реализация функционала по редактированию файла /root/.ssh/authorized_keys на хостах системы.

12. Запрещается реализация функционала по редактированию содержимого следующих директорий:

а. /etc/sysconfig/rsyslog;

б. /etc/rsyslog.conf;

в. файлы из папки /etc/rsyslog.d/;

г. файлы из папки /etc/audit/.

13. Запрещается реализация остановки служб auditd и Rsyslog.

14. Запрещается предоставлять пользователю возможность выбора сервиса, который будет остановлен.

15. В случае реализации ПО модулей уровня ядра запрещается:

– создавать блочное или символьное устройство, позволяющее осуществлять запись в себя для непривилегированных пользователей;

– создавать блочное или символьное устройство, позволяющее осуществлять отображение содержимого файла устройства в контекст памяти пользовательского процесса для непривилегированных пользователей;

– создавать RAM-based интерфейс ядра в виде файла или директории с правами, позволяющими осуществлять запись в себя для непривилегированных пользователей;

– открывать UDP или NetLink сокет в режиме ядра;

– отправлять сигналы в user-space процессы;

– вызывать произвольный код из user-space.

16. ПО, устанавливаемому в ОС в виде драйверов, функционирующих в режиме ядра:

– драйвер не должен использовать механизм сетевых сокетов, а именно:

а. драйвер не должен использовать функционал Netlink сокетов,

б. драйвер не должен использовать функционал UDP сокетов;

– драйвер не должен добавлять новый системный вызов в таблицу системного вызова, скрипты сборки драйвера не должны модифицировать исходный код ОС для добавления системных вызовов;

– драйвер не должен использовать функцию upcall.

17. Программное обеспечение, функционирующее в рамках ЗОС «СинтезМ», должно пройти тематические исследования в части:

– анализа уязвимостей, присутствующих в нем или в используемых им компонентах сторонней разработки и описанных в общедоступных источниках;

– анализа уязвимостей и опасных возможностей, проводимого с использованием автоматизированного средства поиска уязвимостей, осуществляющего анализ исходных текстов ПО;

– В ЗОС «СинтезМ» не должно присутствовать уязвимостей и опасных возможностей, которые потенциально могут быть использованы непривилегированным пользователем для влияния на механизмы защиты ЗОС «СинтезМ».

18. В АСЗИ, в рамках которой устанавливается ЗОС «СинтезМ», не допускается наличие хостов с ОС, отличной от ЗОС «Синтез» (версия 6.4) и ЗОС «СинтезМ» (версия 7.2).

19. Единственным средством обращения прикладного ПО к SELinux является библиотека libselinux. Таблица с описаниями функций (названия и назначения) для библиотеки libselinux приведена в приложении Д.

1.4. Требования к программным средствам

Для функционирования изделия на базе физических серверов не требуется наличие предустановленного программного обеспечения.

При функционировании в составе АСЗИ изделие взаимодействует с:

1) Операционными системами:

– ПК «Синтез-Клиент» (ЗОС «Синтез» версия 6.4);

– ПК «СинтезМ-Клиент» (ЗОС «СинтезМ»).

2) Комплекс программных средств защиты информации, функционирующих в среде операционной системы «СинтезМ» (КП «ПСЗИ «СинтезМ»):

– ПК «СинтезМ-СБ»;

– ПС «СинтезМ-АБ»;

– ПК «СинтезМ-ИПА»;

– ПС «СинтезМ-УК»;

– ПС «СинтезМ-Репо».

3) Комплекс программных средств «Функциональные сервисы» (КПС «Функциональные сервисы») :

– ПС «СинтезМ-СУБД»;

– ПС «СинтезМ-СП»;

– КП «СинтезМ-О»;

– КП «СинтезМ-КУФ»;

– ПК «СинтезМ-КУФ.С»;

– ПК «СинтезМ-КУФ.ПС»;

– ПС «СинтезМ-КУФ.А»;

– ПС «СинтезМ-СХД-01»;

– ПС «СинтезМ-СХД-02»;

– ПС «СинтезМ-СРК»;

– КП «СинтезМ-ЗЭП»;

– ПС «СинтезМ-ЗЭП.С»;

– ПС «СинтезМ-ЗЭП.К».

4) Аппаратно-программное средство «Персональный идентификатор пользователя» (АПС «СинтезМ-ТК/ПИ»).

1.5. Требования к техническим средствам

Для функционирования изделия необходимы технические средства со следующими минимальными характеристиками:

– процессор архитектуры x86_x64 с поддержкой аппаратной виртуализации и тактовой частотой не менее 2 ГГц;

– оперативная память не менее 8 ГБ;

– объем жесткого диска не менее 50 ГБ;

– сетевой контроллер с минимальной пропускной способностью 1000 Мбит.

Требования к конфигурации технических средств для сервера виртуализации могут меняться в зависимости от сложности архитектуры АИС.

Расчет требований необходимо производить по следующим параметрам:

– количество ядер процессора;

– объем необходимого дискового пространства;

– объем оперативной памяти.

В общем случае расчет производится по формулам (1-3).

$$a_A = \frac{(0.5 + x)N_B + N_C + \left\lceil \frac{N_D}{20} \right\rceil + 0.5N_E + X}{2} \quad (1);$$

$$b_A = (10 + (8 + x)N_B + 110N_C + 20 + 2\left\lceil \frac{N_D}{20} \right\rceil + 5N_E + X \quad (2);$$

$$c_A = (2 + x)N_B + 2N_C + 2\left\lceil \frac{N_D}{20} \right\rceil + 0.5N_E + X \quad (3),$$

где:

a_A – количество ядер, необходимое для функционирования сервера виртуализации;

b_A – объем дискового пространства, необходимый для функционирования сервера виртуализации;

c_A – объем оперативной памяти, необходимый для функционирования сервера виртуализации;

x – необходимое количество ресурса для обеспечения функционирования клиентских частей устанавливаемого СПО;

X – необходимое количество ресурса для обеспечения функционирования серверных частей устанавливаемого СПО;

N_B – количество пользовательских ВМ;

N_C – количество серверов для файлового хранилища;

N_D – количество терминальных серверов (АРМ);

N_E – количество серверов управления доступом.

Примечания:

1. Символ «⌈ ⌉» указывает на операцию округления получившегося числа к большему целому значению;

2. Для большинства АИС параметры N_C, N_D и N_E будут принимать значения, равные единице.

2. СТРУКТУРА ПРОГРАММЫ

Таблица 2.1 Взаимодействие модулей

Наименование модуля		Назначение	Взаимодействующее программное обеспечение
Ядро			
Драйвера сетевого интерфейса		Обеспечение работы сетевой карты	Драйвер сетевого моста
Драйвера файловых систем		Обеспечение упорядоченного хранения данных на блочном устройстве, журналирования и поиска данных	Модуль работы с блочными устройствами
Модули управления сетью:	Драйвер сетевого моста	Обеспечение сетевого взаимодействия	Модули управления сетью:
	NetFilter/EbTables	Управление сетевым взаимодействием	
Служба аппаратной виртуализации KVM		Обеспечение работы аппаратной виртуализации по технологии Intel-VT	
Модуль работы с блочными устройствами		Обеспечение работы с жесткими дисками и подключаемыми системами хранения данных	
Модуль мандатного контроля SELinux		Обеспечение контроля над системными операциями	
Модули управления памятью		Управление трансляцией между физическими и виртуальными адресами оперативной памяти	
Загрузчик ОС			
Загрузчик GRUB		Обеспечение загрузки операционной системы	
Система взаимодействия с терминальными клиентами (ТС)			
Терминальный сервер		Обеспечение загрузки по сети образа терминального клиента	Сервер единого времени
Сервер управления динамическими IP-адресами (DHCP)		Выдача параметров сети тонким клиентам в момент их загрузки по PXE	
Служба сетевого доступа к файловой системе (NFS)		Обеспечение сетевого доступа тонкого клиента к директории, в которой хранится образ тонкого клиента	Драйвер файловой системы
Служба передачи файлов (TFTP)		Обеспечение загрузки ядра и образа первоначальной загрузки по протоколу TFTP	Драйвер файловой системы
Образ тонкого клиента		Предоставление корневой файловой системы для тонкого клиента	
Система виртуализации (Менеджер виртуализации)			

Модуль управления виртуализации (OVIRT)	Обеспечение управления средой виртуализации с помощью веб-интерфейса или через RestAPI	База данных с настройками виртуальных машин Служба взаимодействия с менеджером виртуализации (VDSM)
База данных с настройками виртуальных машин	Обеспечение хранения настроек виртуальных машин, серверов виртуализации и пр.	Драйвер файловой системы
Механизм аутентификации в среде виртуализации	Авторизация пользователей с использованием внешней системы управления доступом (IPA)	Сервер LDAP
Система виртуализации (Гипервизор)		
Библиотека управления виртуализацией (Libvirt)	Выполнение запуска виртуальных машин с требуемыми параметрами	Служба ВМ (QEMU)
Служба взаимодействия с менеджером виртуализации (VDSM)	Обеспечение управления сервером виртуализации со стороны менеджера управления ВМ	Библиотека управления виртуализацией (Libvirt)
Служба ВМ (QEMU) Сервер Spice-сессий	Обеспечение работы виртуальной машины и предоставление доступа к ее консоли через протокол Spice	Служба аппаратной виртуализации KVM
Служба отказоустойчивости менеджера виртуализации (Self-Hosted-Engine)	Обеспечение отказоустойчивой работы виртуальной машины менеджера управления ВМ	Модуль работы с блочными устройствами
Служба единого времени NTP		
Сервер единого времени	Выполнение функции источника времени для клиентов единого времени	
Клиент единого времени	Обеспечение синхронизации локального времени с источником времени	Сервер единого времени
Служба управления питанием		
Модуль взаимодействия с UPS	Опрос устройств бесперебойного питания APC	
Служба имен		
Модуль клиент службы имен (DNS)	Обеспечение получения IP адреса по имени хоста	Сервер службы имен (DNS)
SSH-сервер		
SSH-сервер	Предоставление удаленного доступа по защищенному соединению	

2.1. Ядро

Ядро изделия условно делится на три больших уровня. На верхнем уровне располагается интерфейс системных вызовов, который реализует базовые функции, например, чтение и запись. Ниже интерфейса системных вызовов располагается архитектурно-независимый код ядра. Этот код является общим для всех процессорных архитектур, поддерживаемых изделием. Еще ниже располагается архитектурно-зависимый код ядра.

Основные подсистемы ядра:

- интерфейс системных вызовов;
- управление процессами;
- управление памятью;
- виртуальная файловая система;
- сетевой стек;
- драйверы устройств;
- архитектурно-зависимый код.

2.1.1. Интерфейс системных вызовов

SCI – это тонкий уровень, предоставляющий средства для вызова функций ядра из пространства пользователя. Этот интерфейс может быть архитектурно зависимым, даже в пределах одного процессорного семейства. SCI фактически представляет собой службу мультиплексирования и демультиплексирования вызова функций. Реализация SCI находится в ./linux/kernel, а архитектурно-зависимая часть - в ./linux/arch.

2.1.2. Управление процессами

Управление процессами сконцентрировано на исполнении процессов. В ядре эти процессы называются потоками; они соответствуют отдельным виртуализованным объектам процессора (код потока, данные, стек, процессорные регистры). В пространстве пользователя обычно используется термин процесс, хотя в реализации Linux эти две концепции (процессы и потоки) не различают. Ядро предоставляет интерфейс программирования приложений (API) через SCI для создания нового процесса (порождения копии, запуска на исполнение, вызова

функций Portable Operating System Interface (POSIX), остановки процесса (kill, exit), взаимодействия и синхронизации между процессами (сигналы или механизмы POSIX).

Еще одна задача управления процессами – совместное использование процессора активными потоками. В ядре реализован алгоритм планирования сложности O (1), время работы которого не зависит от числа потоков, претендующих на ресурсы процессора. Алгоритм планирования сложности – O (1) – подчеркивает, что на диспетчеризацию одного потока затрачивается столько же времени, как и на множество потоков, O (1) поддерживает симметричные многопроцессорные конфигурации (SMP). Исходные коды системы управления процессами находятся в ./linux/kernel, а коды архитектурно-зависимой части – в ./linux/arch.

2.1.3. Управление памятью

Другой важный ресурс, которым управляет ядро - это память. Для повышения эффективности, учитывая механизм работы аппаратных средств с виртуальной памятью, память организуется в виде страниц (в основном размером 4 КБ). В Linux имеются средства для управления имеющейся памятью, а также аппаратными механизмами для установления соответствия между физической и виртуальной памятью. Linux предоставляет абстракции над этими 4 КБ буферами, например, механизм распределения slab allocator. Этот механизм управления базируется на 4КБ буферах, но затем размещает структуры внутри них, следя за тем, какие страницы полны, какие частично заполнены и какие пусты.

В условиях наличия большого числа пользователей памяти возможны ситуации, когда вся имеющаяся память будет исчерпана. В связи с этим страницы можно удалять из памяти и переносить на диск. Этот процесс обмена страниц между оперативной памятью и жестким диском называется подкачкой. Исходные коды управления памятью находятся в ./linux/mm.

2.1.4. Виртуальная файловая система

Еще один интересный аспект ядра Linux – виртуальная файловая система (VFS), которая предоставляет общую абстракцию интерфейса к файловым системам. VFS предоставляет уровень коммутации между SCI и файловыми системами, поддерживаемыми ядром.

На верхнем уровне VFS располагается единая API–абстракция таких функций, как открытие, закрытие, чтение и запись файлов. На нижнем уровне VFS находятся абстракции файловых систем, которые определяют, как реализуются функции верхнего уровня. Они представляют собой подключаемые модули для конкретных файловых систем (которых существует более 50). Исходные коды файловых систем находятся в ./linux/fs.

Ниже уровня файловой системы находится кэш буферов, предоставляющий общий набор функций к уровню файловой системы (независимый от конкретной файловой системы). Этот уровень кэширования оптимизирует доступ к физическим устройствам за счет краткосрочного хранения данных (или упреждающего чтения, обеспечивающего готовность данных к тому моменту, когда они понадобятся). Ниже кэша буферов находятся драйверы устройств, реализующие интерфейсы для конкретных физических устройств.

2.1.5. Сетевой стек

Сетевой стек по своей конструкции имеет многоуровневую архитектуру, повторяющую структуру самих протоколов. Протокол Internet Protocol (IP) - это базовый протокол сетевого уровня, располагающийся ниже транспортного протокола Transmission Control Protocol (TCP). Выше TCP находится уровень сокетов, вызываемый через SCI.

Уровень сокетов представляет собой стандартный API к сетевой подсистеме. Он предоставляет пользовательский интерфейс различным сетевым протоколам. Уровень сокетов реализует стандартизованный способ управления соединениями и передачи данных между конечными точками, от доступа к

"чистым" кадрам данных и блокам данных протокола IP (PDU) и до протоколов TCP и User Datagram Protocol (UDP). Исходные коды сетевой подсистемы ядра находятся в каталоге ./linux/net.

2.1.6. Драйверы устройств

Подавляющее большинство исходных кодов ядра Linux приходится на драйверы устройств, обеспечивающих возможность работы с конкретными аппаратными устройствами. В дереве исходных кодов Linux имеется подкаталог драйверов, в котором, в свою очередь, имеются подкаталоги для различных типов поддерживаемых устройств. Исходные коды драйверов устройств находятся в ./linux/drivers.

2.1.7. Архитектурно-зависимый код

Хотя основная часть Linux независима от архитектуры, на которой работает операционная система, в некоторых элементах для обеспечения нормальной работы и повышения эффективности необходимо учитывать архитектуру. В подкаталоге ./linux/arch находится архитектурно-зависимая часть исходного кода ядра, разделенная на ряд подкаталогов, соответствующих конкретным архитектурам. Все эти каталоги в совокупности образуют BSP. В случае обычного настольного ПК используется каталог i386. Подкаталог для каждой архитектуры содержит ряд вложенных подкаталогов, относящихся к конкретным аспектам ядра, таким как загрузка, ядро, управление памятью и так далее. Исходные коды архитектурно-зависимой части находятся в ./linux/arch.

2.2. Загрузчик ОС

Загрузчик операционной системы – необходим для обеспечения загрузки операционной системы непосредственно после включения компьютера.

Выполняемые задачи:

- загружает ядро операционной системы в оперативную память. Загрузка ядра операционной системы может происходит не только с жесткого диска, но и по сети;

- формирует входные параметры, передаваемые ядру операционной системы;

- передаёт управление ядру операционной системы.

Установщик GRUB прежде всего изменит код MBR на свой собственный сектор MBR, в котором содержится главная загрузочная запись. Здесь содержится код основного загрузчика (446 байт), таблица разделов с описанием как основных, так и вторичных разделов жесткого диска (64 байта). Поскольку сектор MBR обладает малым объемом, запуск GRUB укладывается в два этапа. В секторе MBR размещена ссылка на конфигурационный файл, который может находиться на любом жестком диске. По ней будет определяться загрузка, которая начинается на втором этапе. Все настройки и данные для работы GRUB считываются из конфигурационного файла. Если же конфигурационный файл не был найден на втором этапе, то процесс загрузки будет прекращен.

2.3. Терминальный сервер (система взаимодействия с терминальными клиентами)

Терминальный сервер ПК «СинтезМ-Сервер» включает пакеты для настройки DHCP-сервера. DHCP-сервер обеспечивает выдачу IP-адресов по установленным MAC-адресам на терминальные клиенты. Также, в состав терминального сервера входят пакеты для настройки TFTP и NFS-серверов, которые обеспечивают работу терминального клиента на АРМ.

2.4. Терминальный клиент

Терминальный клиент ПК «СинтезМ-Сервер» содержит графическое пользовательское окружение и компонент vir-viewer, с помощью которого выполняется подключение по протоколу spice к виртуальной машине пользователя.

2.5. Менеджер виртуализации (система виртуализации)

Менеджер виртуализации – это масштабируемая система управления виртуализацией. Административное управление возможно производить через веб-интерфейс, либо через вызов управляющих команд (API-интерфейс). Менеджер виртуализации является загружаемым модулем ядра ПК «СинтезМ-Сервер» и основан на технологии KVM (Kernel-based Virtual Machine).

Одна или несколько виртуальных машин запускаются на реальном сервере (гипервизоре) с установленной операционной системой ПК «СинтезМ-Клиент». Для управления виртуальными машинами используется Libvirt. Хост с виртуальными машинами и Libvirt в системе с менеджером виртуализации называется «управляемый узел». Узел состоит из Libvirt, гипервизора и гостевых операционных систем, которые запущены на этом гипервизоре. Таких узлов может быть несколько.

Административный узел включает в свой состав web management UI – интерфейс пользователя, построенный на базе Apache сервер и Java. Благодаря такому решению управление может осуществляться практически с любого компьютера с графическим браузером. Вычислительная часть построена на Java, который взаимодействует с Libvirt и другими компонентами. Например, для хранения текущей информации используется PostgreSQL. Администрирование, аутентификация и авторизация выполняются с помощью IPA-сервера, что защищает систему от несанкционированного доступа. Одной из основных возможностей административного узла является горячий перенос виртуальной машины с одного физического компьютера на другой без перебоев в работе и практически незаметно для пользователя. Административный узел может включать и хранилище iSCSI или NFS, которое также может быть развёрнуто на отдельном компьютере (хосте). Хранилище организует виртуальные диски для виртуальных машин.

2.6. Гипервизор (сервер виртуализации)

Сервер виртуализации представляет собой гипервизор на базе KVM, на котором исполняются образы виртуальных машин.

Гипервизор на базе KVM работает поверх базовой операционной системы, которая обеспечивает службы виртуализации, такие как поддержка устройства ввода/вывода и управление памятью.

Гипервизор на базе KVM позволяет создавать инфраструктуру виртуализации операционной системы, KVM поддерживает платформенно-зависимую виртуализацию на процессорах с аппаратными расширениями для виртуализации.

Модуль KVM является составной частью ядра операционной системы.

В состав каждого сервера виртуализации входит VDSM – это демон, который требуется для управления гостевыми виртуальными машинами. Демон VDSM управляет и контролирует состояние хоста, память и сеть. Также с помощью VDSM выполняется создание виртуальной машины, и другие задачи администрирования хоста, сбора статистики и журналов.

2.7. Служба единого времени NTPD

NTPD – это демон операционной системы, который устанавливает и поддерживает системное время, синхронизированное с серверами точного времени.

NTPD использует два варианта хранения файлов конфигурации:

Первый вариант - с единым конфигурационным файлом для запуска демона в режиме сервера или клиента. В этом случае файл конфигурации обычно называется ntp.conf и размещается в директории /etc.

Второй вариант размещения файла конфигурации NTPD: /etc/ntp/ntpd.conf для демона и /etc/ntp/ntp.conf для клиента.

NTPD выполняет обмен сообщениями с одним или более сконфигурированными серверами через определенные интервалы времени. При

старте, в первый, либо последующие разы, программе требуется несколько раз обменяться сообщениями с большинством этих серверов, чтобы алгоритмы обработки сигналов и оптимизации могли собрать и подготовить данные и установить время. По умолчанию интервал равен 64 секундам, поэтому для установки времени может потребоваться несколько минут. Начальная задержка в установке часов может быть уменьшена при использовании ключевого слова iburst с конфигурационной командой server, как описано на странице.

По умолчанию, NTPD работает в непрерывном режиме, в котором каждый из нескольких внешних серверов опрашивается через временные промежутки, определяемые сложным эвристическим алгоритмом, учитывающим джиттер задержки и погрешность частоты.

2.8. SSH-сервер

SSH (от англ. secure shell -безопасная оболочка) это набор программ, которые позволяют регистрироваться на компьютере по сети, удаленно выполнять на нем команды, а также копировать и перемещать файлы между компьютерами. SSH организует защищенное безопасное соединение поверх небезопасных каналов связи.

SSH предоставляет замены традиционным r-командам удаленного доступа с тем отличием, что они обладают повышенной безопасностью. Они выполняются поверх защищенных зашифрованных соединений, которые не позволяет прослушивать или подменять трафик. Кроме того, SSH может обеспечивать безопасное соединение для передачи любого другого трафика: например, почтовых сообщений или файлов.

Обеспечение безопасности SSH достигается с помощью:

- Шифрования соединения
- Авторизации на сервере (kerberos, pam)
- Аутентификации сервера (не позволяет выполнить подмену сервера)
- Аутентификации клиента

- Проверки целостности пакетов
- Временных параметров аутентификации (не позволяют воспользоваться соединением в том, случае, если спустя некоторое время после перехвата оно все-таки было расшифровано)

В SSH-соединении участвуют две стороны: клиент и сервер. Сервер SSH реализован в виде программы ssh. Программа ssh предназначена для регистрации на удаленном хосте с использование протокола ssh и удаленного выполнения команд. Кроме того, ssh позволяет выполнять туннелирование любых TCP-соединений внутри ssh-канала (port forwarding).
Стандартным портом, на котором ssh сервер ожидает соединение, является 22.

2.9. Служба управления питанием

IPMI (от англ. Intelligent Platform Management Interface) – интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ. Другими словами, IPMI – это средство управления, которое реализовано независимо от основного оборудования сервера и обеспечивает его включение, выключение и перезагрузку.

Ядром системы управления серверной платформой является специализированное устройство – Baseboard Management Controller (BMC).

Контроллеры BMC для серверных материнских плат подключаются к ним через системный интерфейс, названный IPMB (Intelligent Platform Management Bus/Bridge) или к другим контроллерам BMC через интерфейс IPMC (Intelligent Platform Management Chassis).

Управление серверной платформой происходит с помощью модуля Fence Agent, который используя данные из конфигурационного файла (IP-адрес, login, password, Password Script, Authentication type, Use Lanplus) производит подключение к BMC контроллеру для проверки состояния серверной платформы

и выполнения действий по восстановлению ее работоспособности: включение, перезагрузка.

2.10. Служба имен

Функция gethostbyname() возвращает структуру типа hostent машине с именем name. В данном случае name является либо именем машины, либо адресом IPv4 в стандартной точечной нотации, либо адресом IPv6 в нотации с двоеточием (и, возможно, точкой) в качестве разделителя (см. описание адресов IPv6 в RFC 1884). Если name является адресом IPv4 или IPv6, то поиск не производится и gethostbyname() просто копирует name в поле h_name, а его эквивалент для структуры struct in_addr копируется в поле h_addr_list[0] возвращаемой структуры hostent. Если name не оканчивается точкой и установлена переменная окружения HOSTALIASES, то в файле, на который указывает HOSTALIASES, будет произведен поиск name (формат файла приведен в hostname(7)). Если name не оканчивается точкой, то будет произведен поиск в текущем домене и его родителях.

Функция gethostbyaddr() возвращает структуру типа hostent указанному адресу машины addr, длина которого равна len, а тип адреса равен type. В данный момент типом адреса может быть только AF_INET.

Функция sethostent() (если значение stayopen истинно (1)) указывает на то, что соединенный сокет TCP должен быть использован для отправки запросов серверу имен, а также на то, что соединение должно оставаться открытым во время последующих запросов. В противном случае для отправки серверу имен запросов будут использоваться датаграммы UDP.

Функция endhostent() закрывает соединение TCP для запроса сервера имен.

Запросы имени домена, производимые функциями gethostbyname() и gethostbyaddr(), используют комбинацию данных любых или всех серверов имен named(8), записи из файла /etc/hosts и комбинацию данных сетевой информационной службы (NIS или YP) в зависимости от порядка строк файла

/etc/host.conf (см. resolv+(8). По умолчанию производится опрос named (8), затем /etc/hosts.

Структура hostent определена в <netdb.h> следующим образом:

```
struct hostent {
        char    *h_name;        /* официальное имя машины */
        char    **h_aliases;    /* список псевдонимов */
        int     h_addrtype;     /* тип адреса машины */
        int     h_length;       /* длина адреса */
        char    **h_addr_list;  /* список адресов */

}

#define  h_addr    h_addr_list[0]    /*  для  совместимости  с
предыдущими версиями */
```

Полями структуры hostent являются:

h_name (официальное имя машины);

h_aliases (оканчивающийся нулем массив альтернативных имен машины);

h_addrtype (тип адреса; в настоящее время всегда AF_INET);

h_length (длина адреса в байтах);

h_addr_list (оканчивающийся нулем массив сетевых адресов машины в сетевом порядке байтов);

h_addr (первый адрес в h_addr_list определен для совместимости с более ранними версиями).

3. НАСТРОЙКА ПРОГРАММЫ

Программный комплекс «СинтезМ-Сервер» включает в себя следующие дистрибутивы:

- СинтезМ-Сервер-М (Менеджер виртуализации);

- СинтезМ-Сервер-В (Сервер виртуализации);

- СинтезМ-Сервер-ТС (Терминальный сервер).

Менеджер виртуализации (сервер управления ВМ) – это система со специальным набором пакетов и программ, позволяющая управлять виртуальными машинами и системой визуализацией.

Сервер виртуализации (гипервизор) – это система виртуализации со специальным набором пакетов и программ, позволяющая предоставлять набор вычислительных ресурсов, абстрагированных от аппаратной реализации, и обеспечивающая логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

Терминальный сервер - это система со специальным набором пакетов и программ, обеспечивающая хранение программного обеспечения терминальных клиентов, не имеющих собственных средств хранения информации.

Установка и настройка компонент осуществляется в соответствие с приведенным ниже перечнем действий:

СинтезМ-Сервер-М	СинтезМ-Сервер-В	СинтезМ-Сервер-ТС	Примечание
п. 3.1	п. 3.1	п. 3.1	
п. 3.1.1	п. 3.1.1	п. 3.1.1	
п. 3.1.2	п. 3.1.2	п. 3.1.2	Создание раздела «/ovirt-engine» выполняется только для менеджера виртуализации (СинтезМ-Сервер-М)
п. 3.1.3	п. 3.1.3	п. 3.1.3	
п. 3.1.3	п. 3.1.3	п. 3.1.3	
п. 3.1.4	п. 3.1.4	п. 3.1.4	
п. 3.2	п. 3.2	п. 3.2	
п. 3.2.1	п. 3.2.1	п. 3.2.1	
п. 3.2.2	-	-	
п. 3.2.3	-	-	
п. 3.2.	-	-	
п. 3.3	-	-	
п. 3.4	-	-	
п. 3.5	-	-	
-	-	п. 3.7	
-	-	п. 3.8	

3.1. Установка сервера виртуализации

Для установки ПК «СинтезМ-Сервер» с помощью графического интерфейса необходимо вставить оптический диск с дистрибутивом изделия в устройства для чтения дисков сервера.

Примечания:

1. На сервере должна быть выставлена загрузка с диска или иного носителя.

2. На сервере должна быть выставлена загрузка в режиме Legacy.

3. На сервере должен быть настроен RAID-массив. В зависимости от архитектуры системы массив может быть разного уровня.

После загрузки оптического диска в режиме Legacy будет предоставлено окно выбора (рисунок 1), в котором необходимо выбрать пункт «Установить СинтезМ-Сервер» и нажать на клавишу [Enter]. После чего, запустится процесс загрузки данных с оптического диска и начнется установка сервера виртуализации.

Далее откроется окно выбора языка, который будет использоваться в процессе установки (рисунок 2). Необходимо выбрать «Русский>Русский (Россия)» и нажать на кнопку «Продолжить».

После выбора языка, откроется меню настройки изделия (рисунок 3), в котором необходимо последовательно настроить следующие параметры:

– дата и время;

– расположение установки (настройка разбиения диска);

– сеть и имя узла;

– выбор программ.

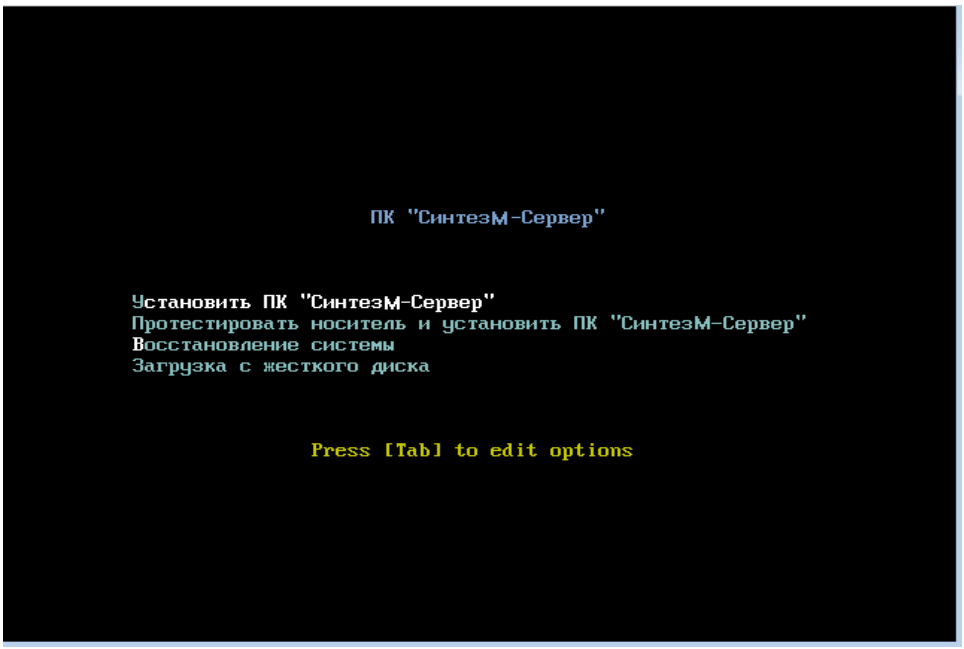

Рисунок 1 – Меню загрузчика

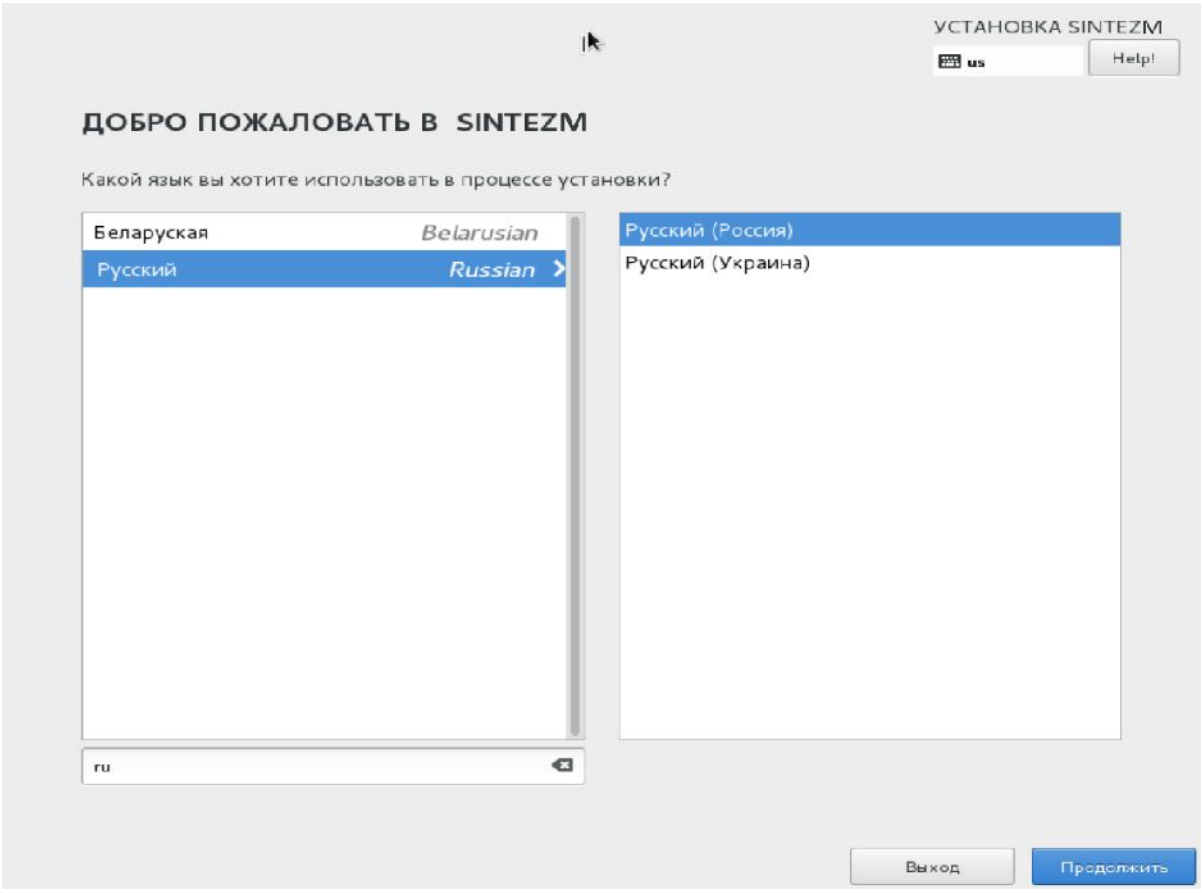

Рисунок 2 – Выбор языка

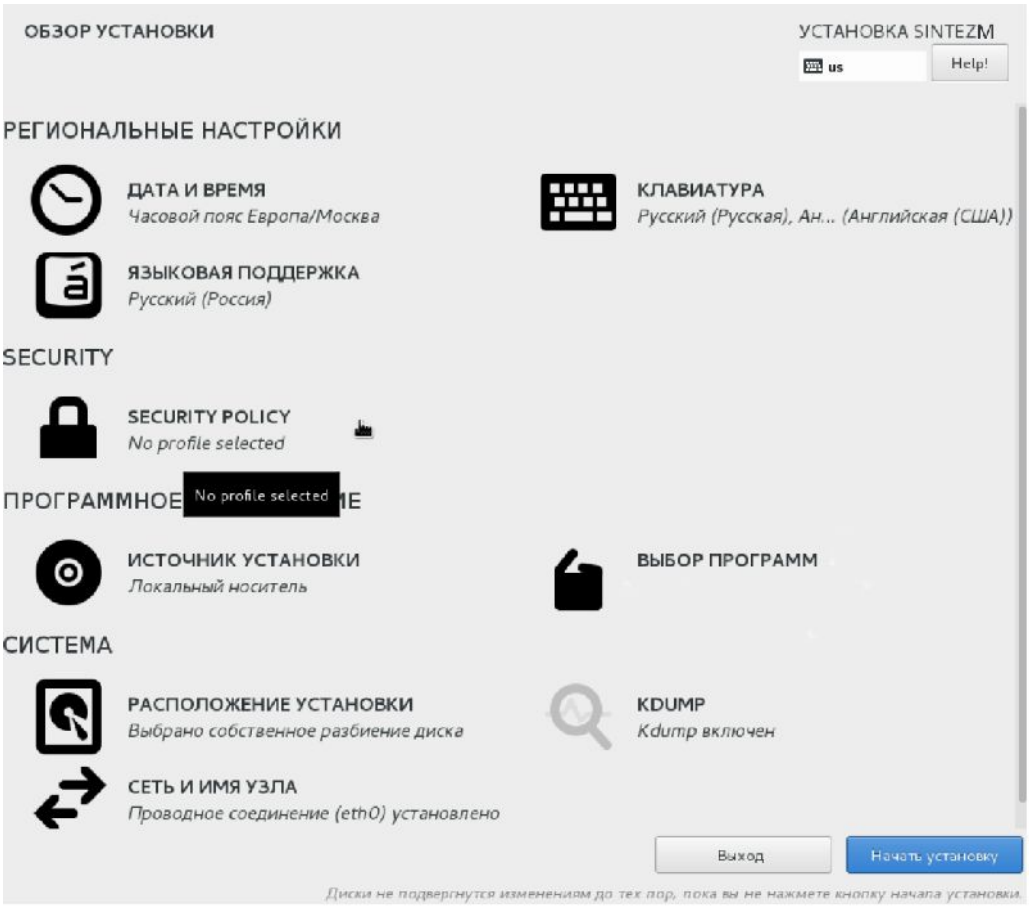

Рисунок 3 – Меню настройки изделия

3.1.1. Настройка даты и времени

Для настройки даты и времени необходимо выбрать пункт меню «Дата и время» (см. рисунок 3).

Примечание. Для выбора часового пояса необходимо указать на карте приблизительное географическое месторасположение сервера.

Для проведения полной настройки данного параметра, после выбора часового пояса, необходимо указать регион и город, выбрав соответствующие значения из ниспадающих списков, расположенных в левой верхней части окна.

Для указания даты и времени необходимо использовать поле, расположенное в нижней части окна. Для определения формата отображения времени необходимо выставить маркер напротив соответствующего формата.

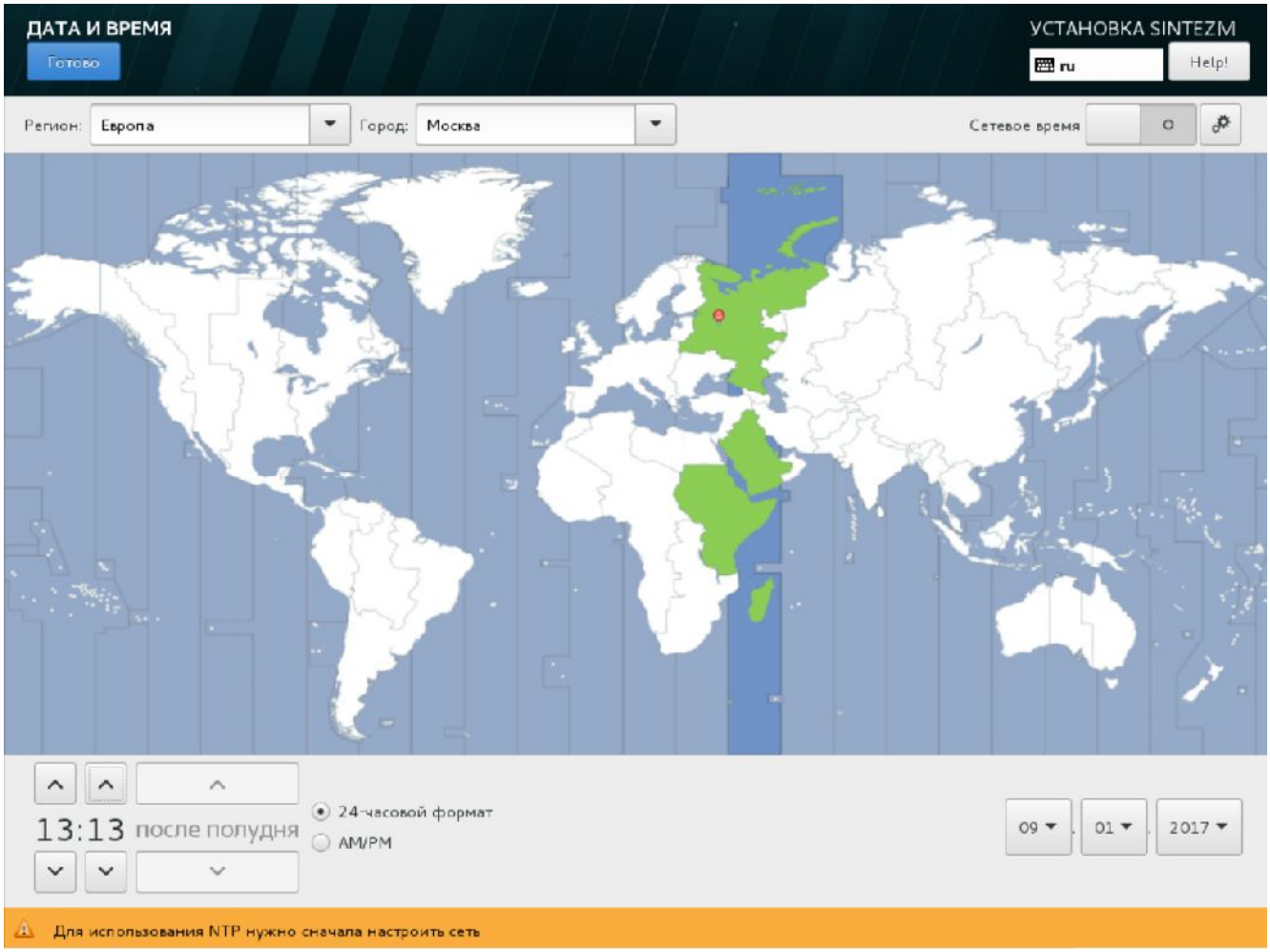

Рисунок 4 – Окно «Настройка даты и времени»

Если в системе используется удаленный сервер времени, необходимо ввести данные о нем. Для этого необходимо нажать на кнопку, содержащую изображение шестеренок, расположенную в верхней правой части окна. В появившемся окне (рисунок 5) необходимо убрать маркеры в поле «Выбрать» для стандартных серверов времени и добавить необходимые данные сервера: IP-адрес или доменное имя.

Для добавления сервера необходимо в поле ввода ввести IP-адрес соответствующего сервера и нажать на кнопку «+», расположенную в правой верхней части окна.

Также можно не добавлять новые сервера времени, а изменить стандартные. Возможность изменения, предоставляется путем двойного нажатия на имя соответствующего узла.

После выбора всех необходимых серверов времени необходимо нажать на кнопку «ОК».

После завершения настройки даты и времени необходимо нажать на кнопку «Готово», расположенную в верхней левой части окна «Дата и время» (см. рисунок 4).

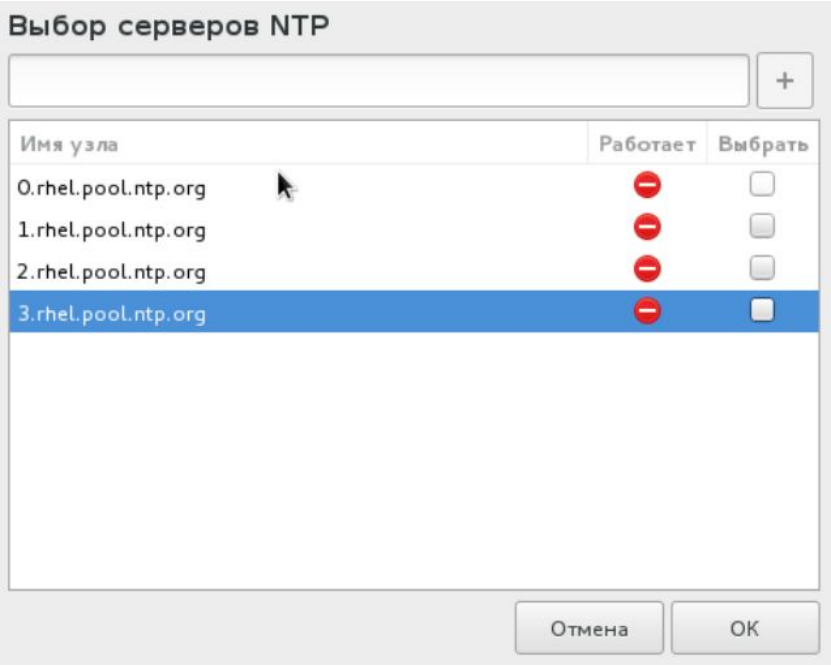

Рисунок 5 – Окно «Выбор серверов NTP»

3.1.2. Настройка расположения установки

Для настройки расположения установки необходимо выбрать соответствующий пункт в меню настройки изделия (см. рисунок 3). Откроется окно «Место установки» (рисунок 6), в котором необходимо выбрать диск, на который будет производиться установка изделия.

Установка изделия может производиться на локальный диск, в этом случае необходимо выбрать соответствующее устройство в поле «Локальные диски» или на сетевой диск.

Для установки изделия на сетевой диск необходимо нажать на кнопку «Добавить диск...», расположенную в поле «Специализированные и сетевые диски», указать тип подключения диска, выбрать необходимый диск из списка и нажать на кнопку «Готово».

Для установки параметров разбиения диска необходимо выставить маркер напротив соответствующего типа разбиения диска в поле «Другие параметры хранения данных». При выборе значения «Я настрою разделы», разбиение диска будет производиться вручную. Далее необходимо нажать на кнопку «Готово», расположенную в верхней левой части окна.

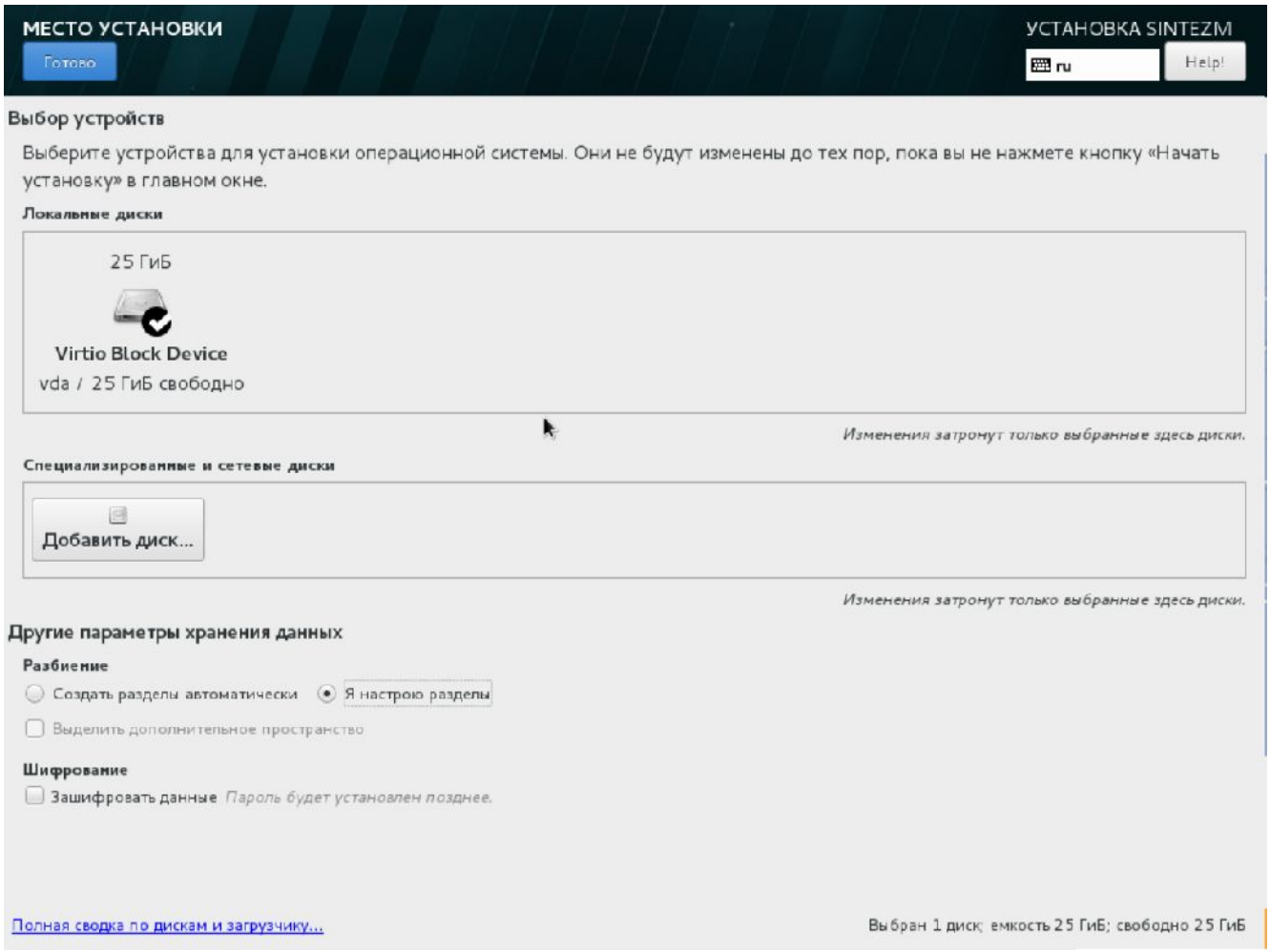

Рисунок 6 – Окно «Место установки»

Если было выбрано разбиение диска вручную, откроется окно «Разметка вручную» (рисунок 7), в котором необходимо нажать на ссылку «Создать их автоматически». После чего программа автоматически создаст разделы, в которых потребуется последующая ручная настройка.

При проведении ручной донастройки необходимо удалить созданный каталок /home, для этого необходимо выбрать его и нажать на кнопку «–» находящуюся в левом нижнем левом углу окна.

Примечание. Создание нового раздела «/ovirt-engine» необходимо формировать только для СинтезМ-Сервер-М (Менеджер виртуализации).

Далее необходимо создать новый раздел, нажав на кнопку «+», расположенную в нижнем левом углу окна. В открывшемся окне (рисунок 8) необходимо в поле «Точка монтирования» ввести значение «/ovirt-engine», в поле «Размер» указать «100 GB» и нажать на кнопку «Добавить».

Примечание. Переход на английскую раскладку клавиатуры осуществляется одновременным нажатием клавиш "Alt" и "Shift".

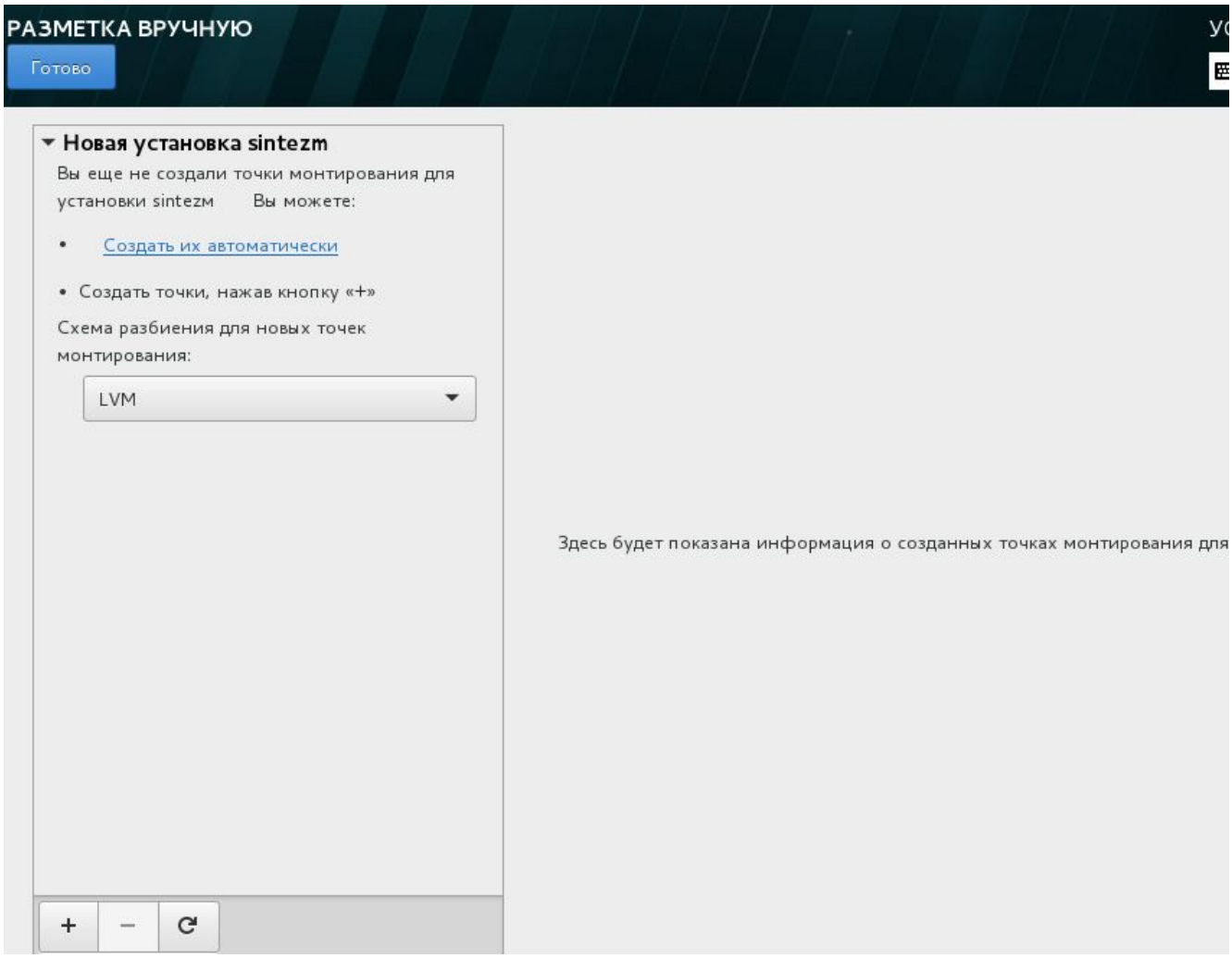

Рисунок 7 – Окно «Разметка вручную»

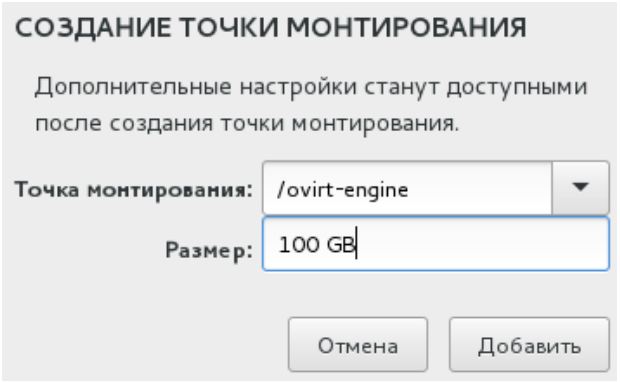

Рисунок 8 – Окно «Создание точки монтирования». Образ ВМ менеджера

Примечание. В директории /ovirt-engine будет находится образ виртуальной машины менеджера.

Далее необходимо создать раздел для сообщений системы. Для этого необходимо повторить действия по созданию нового раздела, указав в окне «Создание точки монтирования» параметры, указанные на рисунке 9.

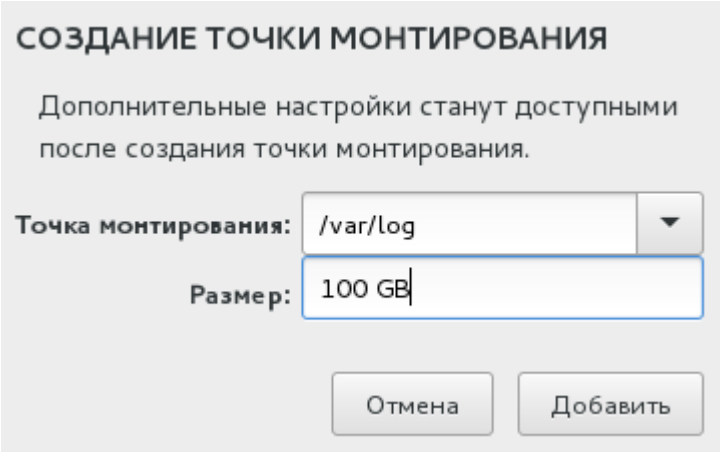

Рисунок 9 – Окно «Создание точки монтирования». Сообщения системы

Оставшееся место на диске необходимо распределить в директорию root. Для этого необходимо выбрать раздел «/» и задать максимально свободное место в поле «Требуемый размер» (максимально свободное пространство указывается в левом нижнем углу окна). После чего необходимо последовательно нажать на кнопку «Применить» и «Готово» (рисунок 10). В открывшемся окне «Обзор изменений» (рисунок 11) необходимо нажать на кнопку «Принять изменения».

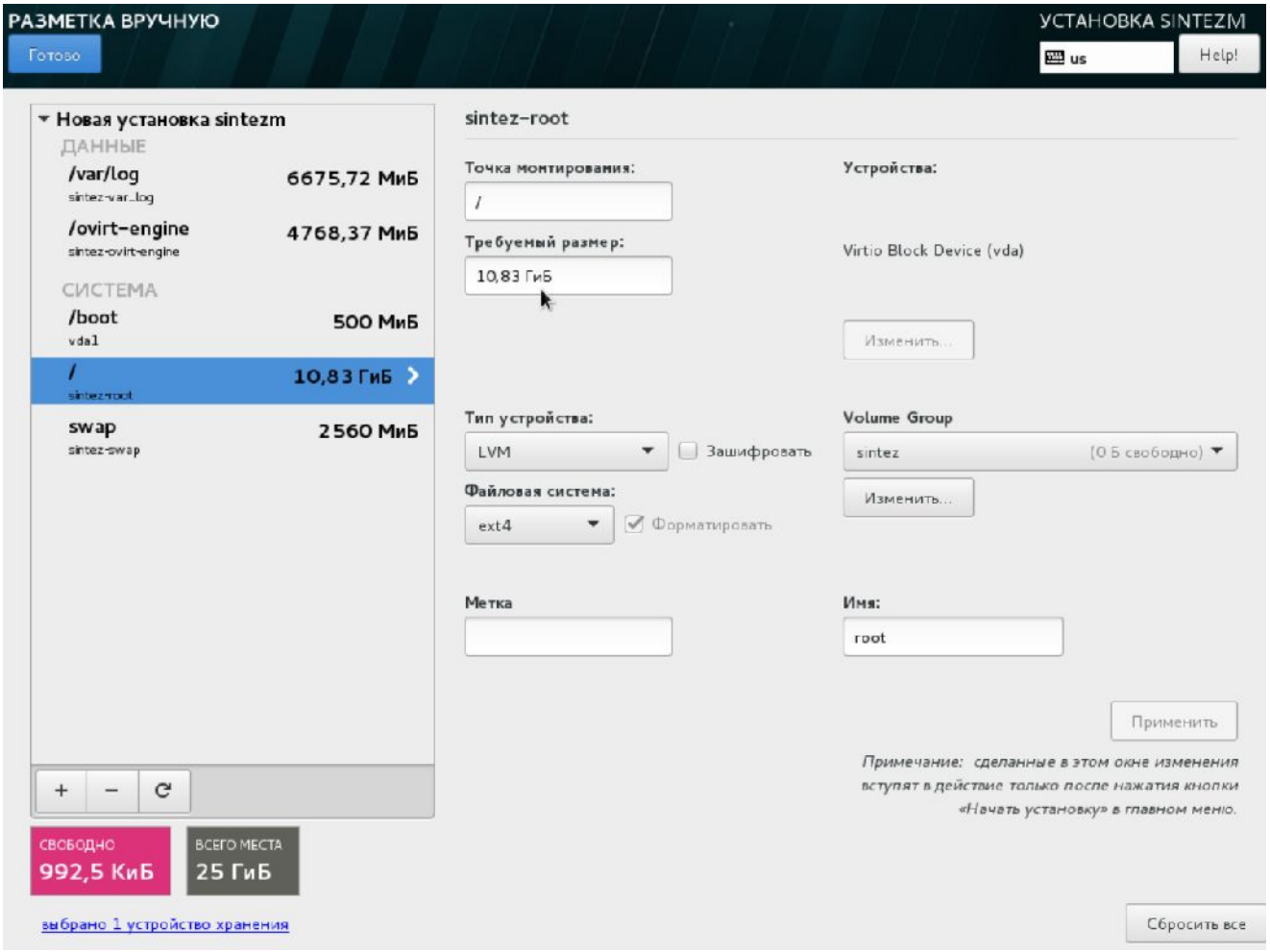

Рисунок 10 – Окно «Разметка вручную». Распределение данных в директорию root

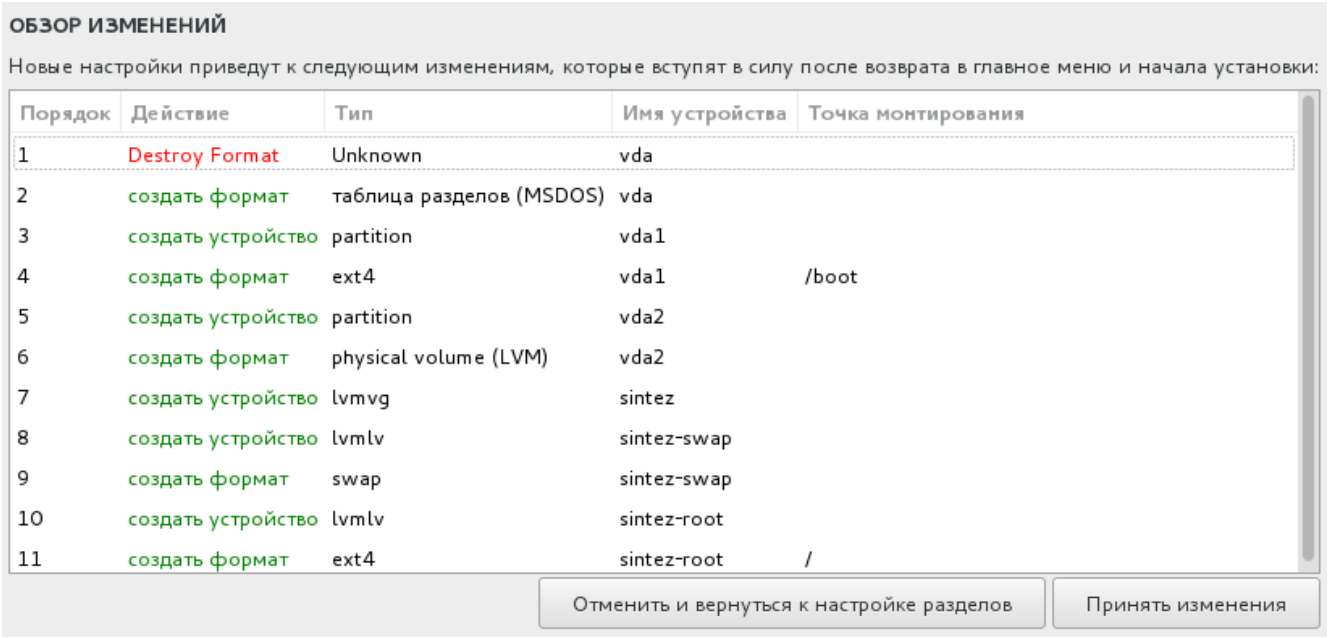

Рисунок 11 – Окно «Обзор изменений»

3.1.3. Настройка сети

Для настройки сети необходимо выбрать пункт «Сеть и имя узла» в меню настройки изделия (см. рисунок 3). В открывшемся окне (рисунок 12) выбрать активное сетевое соединение и нажать на кнопку «Настроить».

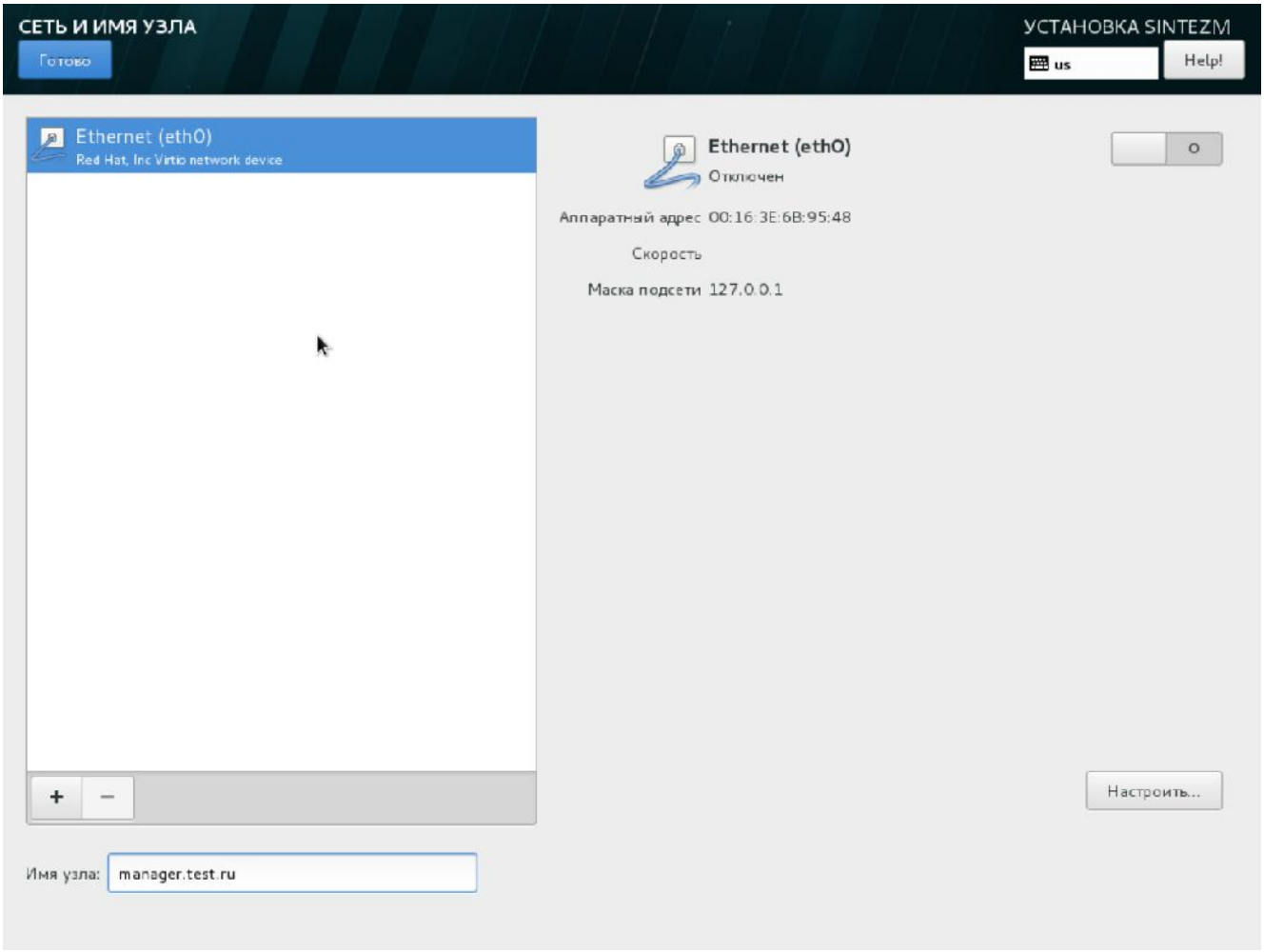

Рисунок 12 – Окно «Сети и имя узла»

Откроется окно «Изменение …» (рисунок 13), в котором необходимо, изменить название соединения (при необходимости). Далее необходимо перейти во вкладку «Параметры IPv4» и выбрать в поле «Способ настройки» из ниспадающего списка значение «Вручную». Для добавления адреса сервера необходимо нажать на кнопку «Добавить» и ввести IP-адрес сервера, маску сети и шлюз. В строке DNS необходимо прописать адрес сервера управление доступом.

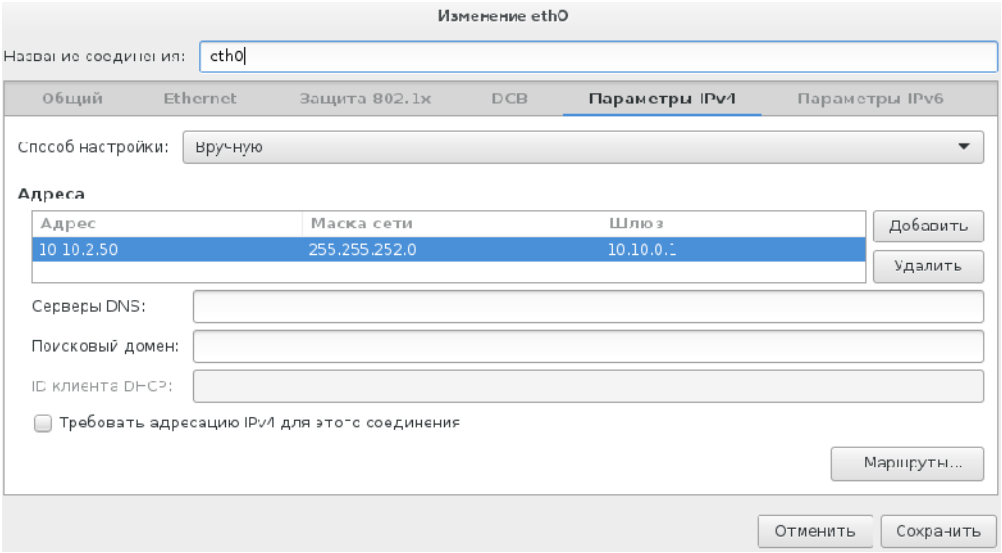

Рисунок 13 – Окно «Изменение …». Вкладка «Параметры IPv4»

Далее необходимо перейти во вкладку «Параметры IPv6» (рисунок 14) и выбрать в поле «Способ настройки» из ниспадающего списка значение «Игнорировать».

Далее необходимо перейти во вкладку «Общий» (рисунок 15) и выставить маркер «Автоматически подключаться к этой сети, когда она доступна» и нажать на кнопку «Сохранить».

В поле «Имя узла», расположенном в левом нижнем углу окна «Сети и имя узла» (см. рисунок 12) необходимо указать полное имя узла (hostname). После окончания всех сетевых настроек нажать на кнопку «Готово», расположенную в левой верхней части окна.

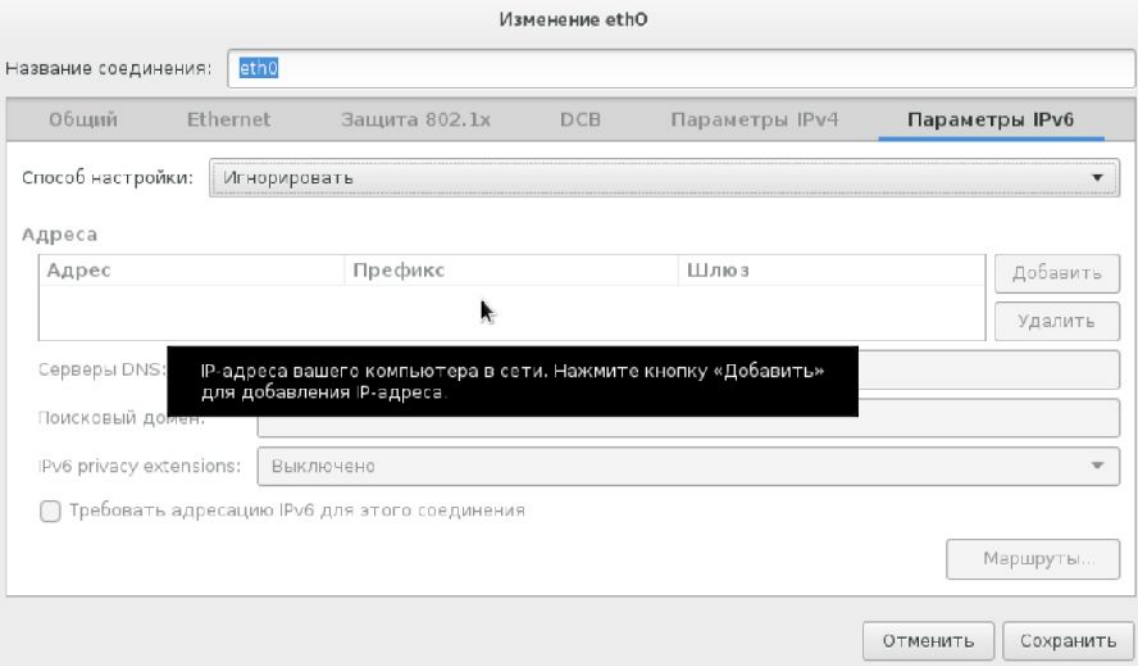

Рисунок 14 – Окно «Изменение …». Вкладка «Параметры IPv6»

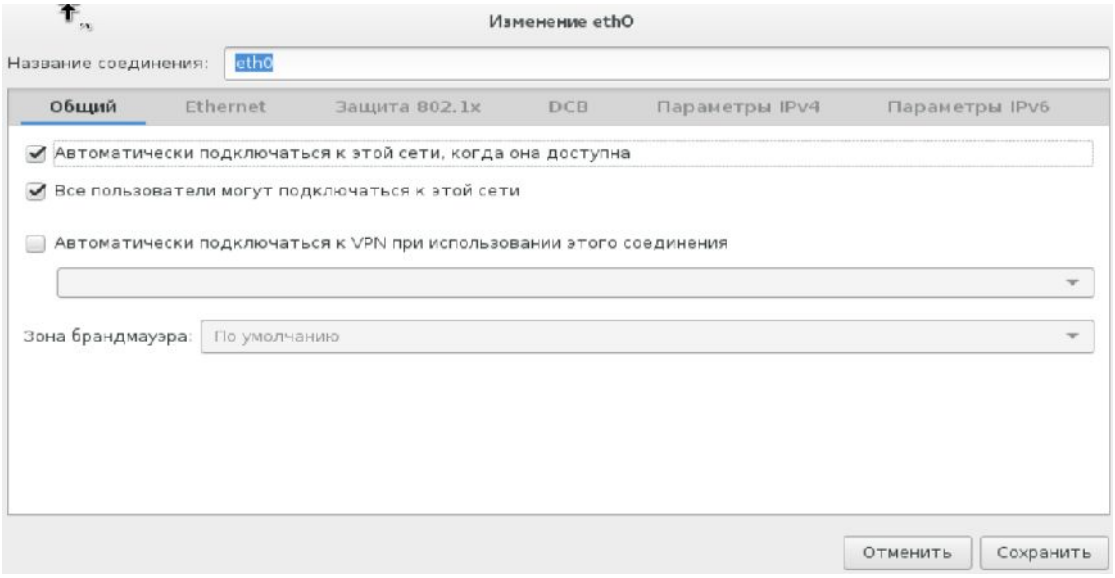

Рисунок 15 – Окно «Изменение …». Вкладка «Общий»

3.1.4. Запуск установки

После завершения настроек необходимо нажать на кнопку «Начать установку» (см. рисунок 3). Откроется окно, содержащее информацию о прогрессе установки (рисунок 16), необходимо дождаться ее окончания.

Примечание. Процесс установки может занимать от 5-30 минут.

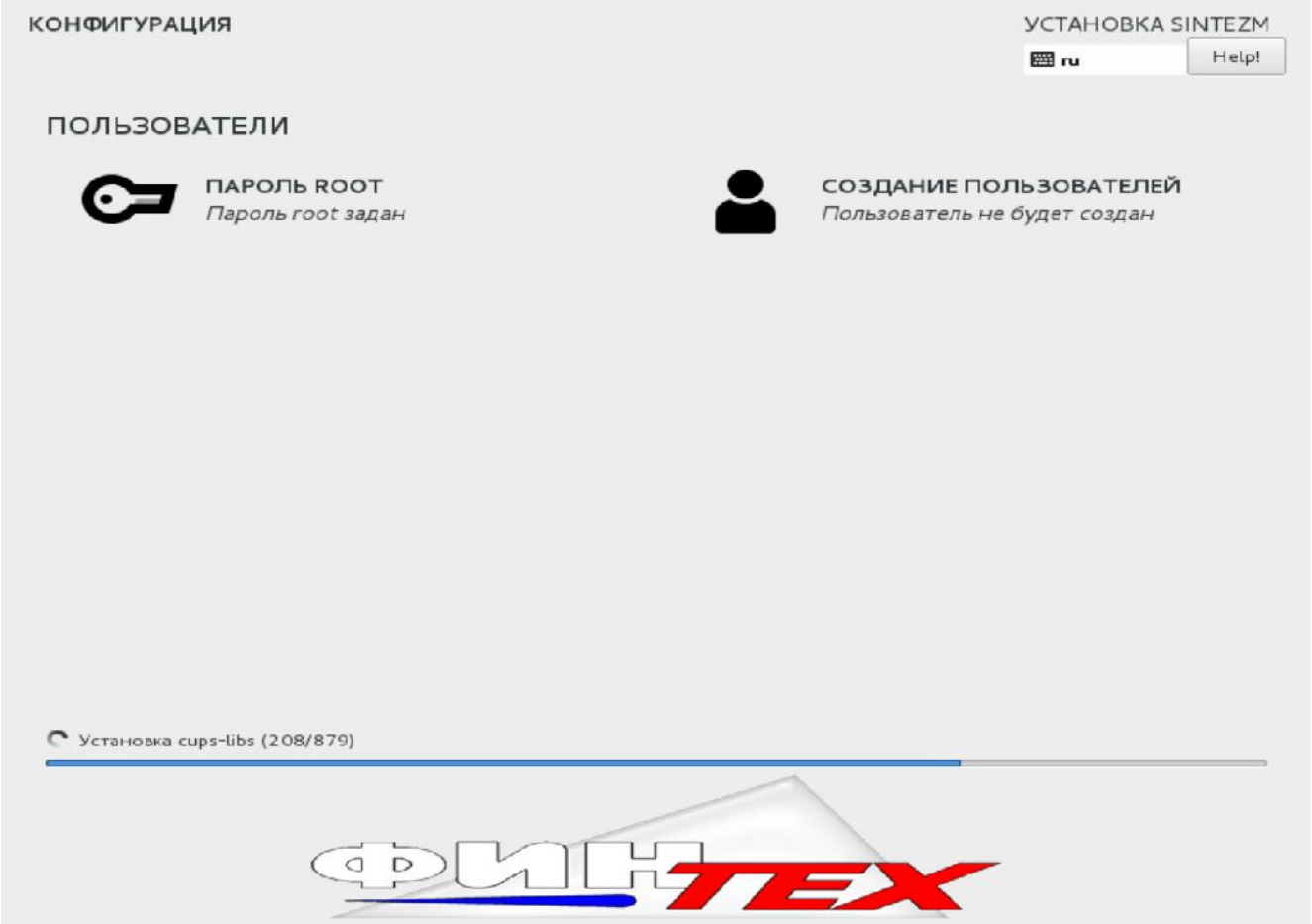

Рисунок 16 – Прогресс установки

Для установки пароля для корневого пользователя необходимо нажать на поле «Пароль root» (см. рисунок 16), в открывшемся окне задать пароль и нажать на кнопку «Готово».

Примечания:

1. Пароль системного пользователя root должен быть не короче 8 символов с применением символов верхнего и нижнего регистров и цифр, при этом вводимые символы не отображаются на экране. Необходимо ввести его дважды; если пароли не совпадают, программа установки запросит повторный ввод пароля.

2. Задание пароля корневого пользователя можно производить как в процессе установки сервера виртуализации, так и после его завершения.

После завершения установки необходимо нажать на кнопку «Перезагрузка» и извлечь оптический диск изделия из устройства для чтения дисков сервера. Далее необходимо дождаться загрузки изделия.

3.2. Настройка сервера виртуализации

После установки операционной системы необходимо провести настройку сервера виртуализации и ВМ авторизации.

Принцип работы ЗОС «СинтезМ» заключается в том, что для работы сервера авторизации физический сервер не требуется. При настройке сервера виртуализации, система предложит создать ВМ управления на самом сервере виртуализации.

3.2.1. Первоначальная настройка

Первоначально необходимо настроить на сервере виртуализации дату и время (если это не сделали при установке ОС). Для того, чтобы настроить дату и время необходимо открыть консоль сервера.

Примечания:

1. Консолью может служить монитор, подключенный к серверной стойке, или KVM-консоль. При наличии ноутбука возможно произвести настройку удаленно подключившись к серверу по протоколу ssh.

2. Все настройки серверов и ВМ необходимо производить в режиме командной строки (консоли) с использованием стандартного текстового редактора vim.

3. Сокращенная инструкция по использованию текстового редактора vim приведена в приложении Б.

Для проведения настройки необходимо в консоли ввести следующую команду для проверки времени:

```
# date
```

Пример:

В результате выполнения команды отобразятся сведения следующего формата:

Thu Dec 22 00:00:00 EST 2016

Значение EST показывает часовую зону расположения сервера. В приведенном примере значение EST обозначает европейское стандартное время.

Для указания часового пояса по месту расположения сервера необходимо ввести следующую команду:

```
ln -sf /usr/share/zoneinfo/Регион/Город/etc/localtime
```

Пример:

Указание московского часового пояса:

```
ln -sf /usr/share/zoneinfo/Europe/Moscow /etc/localtime
```

Указание часового пояса Владивостока:

```
ln -sf /usr/share/zoneinfo/Asia/Vladivostok /etc/localtime
```

Для указания даты и времени необходимо выполнить следующие команды:

```
date -s "mm/dd/yyyy"
date -s hh:mm
```

Пример:

```
date -s "12/22/2016"
date -s 15:15
```

Для синхронизации железного времени (время сервера, то есть BIOS) необходимо выполнить следующую команду:

```
hwclock -w
```

Визуальное представление описанных действий представлено на рисунке 17.

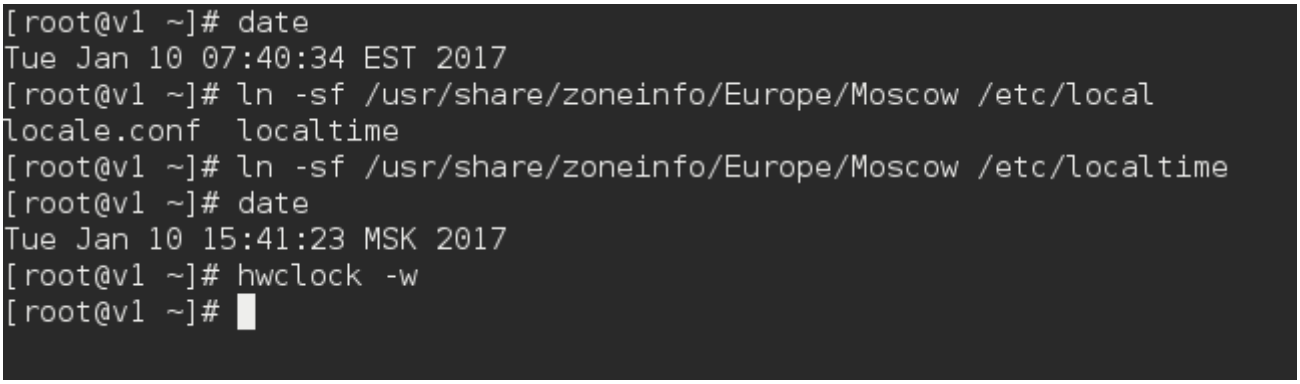

Рисунок 17 – Настройка даты и времени

Примечание. Дата, время и часовой пояс обязательно должны быть настроены на всех серверах и ВМ, так как в дальнейшем все объекты будут подключены к серверу управления доступом. Любая рассинхронизация объекта более чем на 5 минут, приводит к его неработоспособности, поэтому на каждом сервере и ВМ обязательно должна быть настроена синхронизация времени с сервером времени.

Сервер времени – это устройство, сервер или ВМ, на котором настроен сервис раздачи эталонного времени.

Обычно сервер времени настраивается на любом не виртуальном сервере, то есть сервер на котором будет настраиваться сервис времени, должен быть установлен непосредственно на сервер.

Для настройки данного сервиса необходимо открыть файл настроек с использованием стандартного текстового редактора vim, выполнив команду:

```
vim /etc/ntp.conf
```

В открывшемся файле необходимо найти следующую строку:

```
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

В указанной строке необходимо убрать комментарий, изменить данные подсети и маски на соответствующие адреса.

Пример:

```
restrict 10.10.2.0 mask 255.255.252.0 nomodify notrap
```

Далее необходимо добавить в файл следующие строки:

```
server 127.127.1.1
fudge 127.127.1.1 stratum 5
```

Далее необходимо найти и удалить следующую строку и выйти из файла с сохранением:

```
server 0.rhel.pool.ntp.org iburst
```

Визуальное представление описанных действий по настройке NTP представлено на рисунке 18.

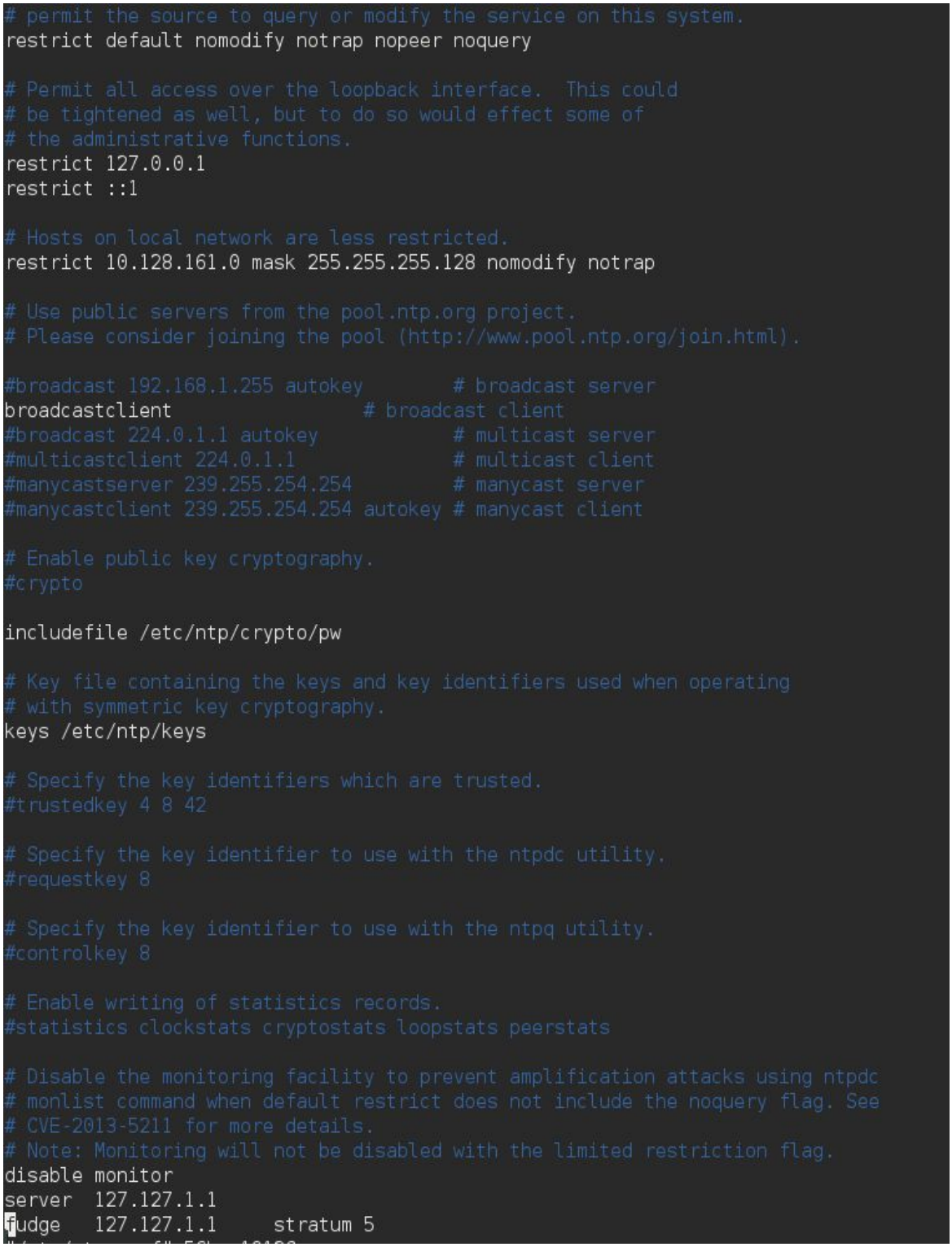

Рисунок 18 – Настройка NTP

Для настройки сервиса времени на клиенте необходимо открыть файл ntp.conf и изменить строку server 0.rhel.pool.ntp.org iburst на адрес сервера времени.

Примечание. Если запущены iptables или firewall необходимо добавить порт 123 tcp и udp.

Далее необходимо перезапустить службу времени и добавить ее в автозагрузку, для этого следует ввести следующие команды:

```
systemctl restart ntpd
systemctl enable ntpd
```

Сервис времени начнет раздачу времени приблизительно через 5 минут после перезагрузки. Для того, чтобы проверить раздачу времени необходимо ввести следующие команды:

```
ntpstat
synchronised to local net at stratum 6
time correct to within 11 ms
polling server every 64 s
```

Далее необходимо добавить узлы серверов виртуализации и менеджера в файл /etc/hosts, выполнив следующую команду:

```
vim /etc/host
```

В конце файла следует добавить следующие строки:

```
10.10.2.50 v1.test.ru
10.10.2.51 v2.test.ru
10.10.2.221 manager.test.ru
```

Примечание. В качестве примера настраиваются 2 сервера, если серверов виртуализации больше, необходимо аналогично добавить каждый из них.

Визуальное представление описанных действий по редактированию файла hosts представлено на рисунке 19.

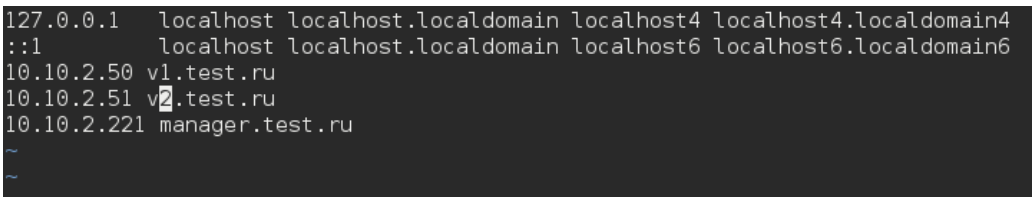

Рисунок 19 – Редактирование файла hosts

Далее необходимо отключить iptables на серверах виртуализации, для этого необходимо ввести следующие команды:

```
iptables -F
service iptables save
systemctl stop iptables
systemctl disable iptables
```

Для изменения имени узла необходимо выполнить следующие команды:

```
vim /etc/hostname
v1.test.ru
```

3.2.2. Загрузка образа системы

Загрузку образа системы можно провести двумя следующими способами:

– копирования ISO-образа системы на сервер виртуализации;

– создание ISO -образа с установочного диска системы.

Для загрузки образа операционной системы необходимо скопировать ISO-образ sintezm-server на сервер виртуализации в папку /tmp/. Для этого необходимо ввести команду копирования файла с АРМ-установщика, подключенного к сети:

```
scp sintezm-h.iso root@10.10.2.50:/tmp/
```

Для создания ISO-образа с установочного диска системы необходимо вставить диск с образом системы в привод дисковода сервера и ввести команду:

```
dd if=/dev/cdrom of=/tmp/image.iso
```

Примечание. Образ системы создается от 2-5 минут.

Для того, чтобы подключить репозиторий, необходимо выполнить команду: `mount -o loop образ-системы.iso /директория монтирования`. В файлах монтирования указать путь к данному образу:

```
vim /etc/yum.repos.d/sintezm-base.repo
```

```
name=base-sintezm-$releasever
baseurl=file:///mnt
enabled=1
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-sintezm-release
```

3.2.3. Настройка хранения данных

Для хранения данных используется локальный метод – это когда данные, хранятся непосредственно на сервере виртуализации в определенной директории

с определенными правами. При использовании данного типа хранения нет необходимости в отдельных серверах хранения данных.

Для настройки прав доступа на директорию /ovirt-engine, необходимо на сервере ввести следующие команды:

```
chmod 755 /ovirt-engine
chown 36:36 /ovirt-engine
```

Далее необходимо добавить в файл «exports» запись о хранилище – директории, где будут храниться данные, выполнив следующие команды (рисунок 20):

```
vim /etc/exports
/ovirt-engine *(rw)
```

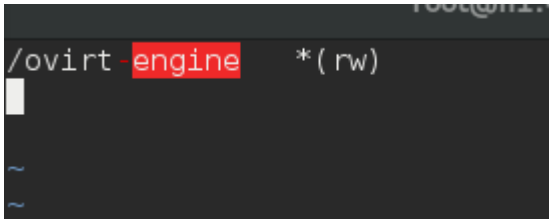

Рисунок 20 – Добавление записи о хранилище

Далее необходимо перезапустить службу nfs, выполнив команду:

```
service nfs-server restart
```

Потом следует включить службу в автозагрузку, выполнив команду:

```
chkconfig nfs-server on
```

Визуальное представление описанных действий по перезапуску службы nfs представлено на рисунке 21.

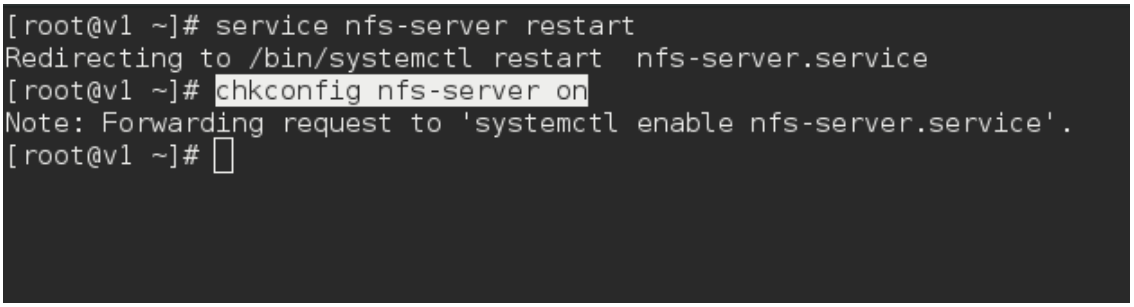

Рисунок 21 – Перезапуск службы nfs

3.3. Установка и настройка сервера управления ВМ

Сервер управления устанавливается на специальную виртуальную машину по технологии self-hosted engine, которая обеспечивает работу виртуальной машины в режиме высокой доступности.

Установка виртуальной машины выполняется после первоначальной настройки системы хранения данных и серверов виртуализации.

Для запуска установки необходимо выполнить следующую команду:

```
hosted-engine --deploy
```

Данная команда запустит процесс настройки свойств системы, параметров ВМ управления доступом, параметров служб, метода хранения данных и т.д.

1) Вариант использования системы хранения данных NFS:

```
     create a VM where you have to install the engine afterwards.
     Are you sure you want to continue? (Yes, No)[Yes]: Yes
     It has been detected that this program is executed through an SSH connection without using screen.
     Continuing with the installation may lead to broken installation if the network connection fails.
     It is highly recommended to abort the installation and run it inside a screen session using command "screen".
     Do you want to continue anyway? (Yes, No)[No]: Yes
     Please specify the storage you would like to use (glusterfs, iscsi, fc, nfs3, nfs4)[nfs3]: nfs3
     Please specify the full shared storage connection path to use (example: host:/path): v1.test.ru:/ovirt-engine
     iptables was detected on your computer, do you wish setup to configure it? (Yes, No)[Yes]: No
     Please indicate a pingable gateway IP address [10.10.0.1]: 10.10.0.1
     Please indicate a nic to set ovirtmgmt bridge on: (eno1) [eno1]: eno1
     Please specify the device to boot the VM from (choose disk for the oVirt engine appliance)
     (cdrom, disk, pxe) [disk]: cdrom
     Please specify the console type you would like to use to connect to the VM (vnc, spice) [vnc]: vnc
     Please specify the CPU type to be used by the VM [model_Haswell-noTSX]:
     Please specify the number of virtual CPUs for the VM [Defaults to minimum requirement: 2] 2
     Please specify the disk size of the VM in GB [Defaults to minimum requirement: 25]: 25
```

```
     You may specify a unicast MAC address for the VM or accept a
randomly generated default [00:16:3e:3c:c4:91]: 00:16:3e:3c:c4:91
     Please specify the memory size of the VM in MB [Defaults to
minimum requirement: 4096]: 4096
     Please specify path to installation media you would like to use
[None]: /tmp/server-sintezm.iso


     Enter engine admin password:
     Confirm engine admin password:

     Enter the name which will be used to identify this host inside
the Administrator Portal [hosted_engine_1]: hosted_engine_1
     Engine FQDN: []: manager.test.ru
     Please provide the name of the SMTP server through which we
will send notifications [localhost]: localhost
     Please provide the TCP port number of the SMTP server [25]: 25
     Please provide the email address from which notifications will
be sent [root@localhost]: root@localhost
     Please provide a comma-separated list of email addresses which
will get notifications [root@localhost]: root@localhost

     Stage: Setup validation

               --== CONFIGURATION PREVIEW ==--

               Bridge interface                   : eno2
               Engine FQDN                        : manager.test.ru
               Bridge name                        : ovirtmgmt
               Host address                       : r22.test.ru
               SSH daemon port                    : 22
               Gateway address                    : 10.128.160.126
               Host name for web application      : hosted_engine_1
               Storage Domain type                : nfs3
               Host ID                            : 1
               Image size GB                      : 25
               Storage  connection                                :
r22.test.ru:/ovirt-engine
               Console type                       : vnc
               Memory size MB                     : 4096
               MAC  address                                       :
00:16:3e:39:2b:40
               Boot type                          : cdrom
               Number of CPUs                     : 2
     ISO  image  (cdrom  boot/cloud-init)    :  /tmp/sintezm-7-server-
v1.2.10.iso
     CPU Type : model_Haswell-noTSX
     Please confirm installation settings (Yes, No)[Yes]: yes
```

Во время установки не должно возникать ошибок, все индикаторы INFO должны быть зеленого цвета. Далее необходимо дождаться окончания установки.

П р и м е ч а н и е . Если во время установки произошли ошибки, необходимо проверить доступность репозиториев, сети и т.д. Перед повторной попыткой сконфигурировать систему, необходимо очистить старые конфигурационные файлы. Далее необходимо заново запустить этап конфигурации системы, выполнив команду hosted-engine —deploy.

Для этого необходимо сделать следующее:

листинг из приложения В необходимо скопировать в текстовый файл с расширением .sh, например:

1) открываем клиент ssh (putty, терминал);

2) вписываем команду cat>clean.sh;

3) вставляем содержимое листинга на следующую строку

(см. приложение В);

4) Ctrl+D;

5) chmod+x clean.sh;

6) ./clean.sh.

После выполнения скрипта необходимо перезагрузить сервер командой reboot –f.

После перезагрузки необходимо установить перечень пакетов, представленных ниже:

- libgovirt;
- libvirt-client;
- libvirt-daemon;
- libvirt-daemon-config-nwfilter;
- libvirt-daemon-driver-interface;
- libvirt-daemon-driver-network;
- libvirt-daemon-driver-nodedev;
- libvirt-daemon-driver-nwfilter;
- libvirt-daemon-driver-qemu;

- libvirt-daemon-driver-secret;
- libvirt-daemon-driver-storage;
- libvirt-daemon-kvm;
- libvirt-lock-sanlock;
- libvirt-python;
- ovirt-engine-sdk-python;
- ovirt-host-deploy;
- ovirt-hosted-engine-ha;
- ovirt-hosted-engine-setup;
- ovirt-imageio-common;
- ovirt-imageio-daemon;
- ovirt-setup-lib;
- ovirt-vmconsole;
- ovirt-vmconsole-host;
- vdsm;
- vdsm-api;
- vdsm-cli;
- vdsm-hook-vmfex-dev;
- vdsm-infra;
- vdsm-jsonrpc;
- vdsm-python;
- vdsm-xmlrpc;
- vdsm-yajsonrpc;
- libguestfs;
- libguestfs-tools-c;
- libguestfs-winsupport;
- mom;
- python-libguestfs;

- safelease;
- virt-v2v;
- virt-viewer,

используя команду yum install <пакет> -y

Далее необходимо заново запустить этап конфигурации системы, выполнив команду `hosted-engine --deploy`.

2) Вариант использования системы хранения данных iSCSI:

```
hosted-engine --deploy

[ INFO  ] Stage: Initializing
[ INFO  ] Generating a temporary VNC password.
[ INFO  ] Stage: Environment setup
          During customization use CTRL-D to abort.
          Continuing will configure this host for serving as hypervisor and create a VM where you have to install the engine afterwards.
          Are you sure you want to continue? (Yes, No)[Yes]: Yes
          It has been detected that this program is executed through an SSH connection without using screen.
          Continuing with the installation may lead to broken installation if the network connection fails.
          It is highly recommended to abort the installation and run it inside a screen session using command "screen".
          Do you want to continue anyway? (Yes, No)[No]: Yes
[WARNING] Cannot detect if hardware supports virtualization
          Configuration files: []
          Log file: /var/log/ovirt-hosted-engine-setup/ovirt-hosted-engine-setup-20170313213725-mrj9mx.log
          Version: otopi-1.5.2 (otopi-1.5.2-1.el7ev)
[ INFO  ] Stage: Environment packages setup
[ INFO  ] Stage: Programs detection
[ INFO  ] Stage: Environment setup
[ INFO  ] Generating libvirt-spice certificates
[WARNING] Cannot locate gluster packages, Hyper Converged setup support will be disabled.
[ INFO  ] Please abort the setup and install vdsm-gluster, gluster-server >= 3.7.2 and restart vdsmd service in order to gain Hyper Conve
[ INFO  ] Stage: Environment customization

          --== STORAGE CONFIGURATION ==--

          Please specify the storage you would like to use (glusterfs, iscsi, fc, nfs3, nfs4)[nfs3]: iscsi
```

```
               Please specify the iSCSI portal IP address: указать
ip адрес iscsi таргета
               Please specify the iSCSI portal port [3260]: 3260
(указать порт iscsi)
               Please specify the iSCSI portal user: ввести имя
пользователя подключения к iscsi (по умолчанию авторизация в iscsi
отключена)
               Please   specify   the   target   name   (iqn.2003-
01.org.linux-iscsi.iscsi.x8664:sn.caef8e55d2c9)           [iqn.2003-
01.org.linux-iscsi             .iscsi.x8664:sn.caef8e55d2c9]: ввести
имя таргета
     [ INFO  ] Discovering iSCSI node
     [ INFO  ] Connecting to the storage server
               The following luns have been found on the requested
target:
                       [1]           36001405f751587b0d69439585e9d16d3
30GiB   LIO-ORG storage03
                                status: free, paths: 1 active
     «Минимальный размер лун 20 Гб»
               Please select the destination LUN (1, 2) [1]: 1
(выбор лун)
     [ INFO  ] Installing on first host

               --== SYSTEM CONFIGURATION ==--


               --== NETWORK CONFIGURATION ==--

               iptables was detected on your computer, do you wish
setup to configure it? (Yes, No)[Yes]: No
               Please indicate a pingable gateway IP address
[10.10.0.1]: указать ip адрес шлюза
               Please indicate a nic to set ovirtmgmt bridge on:
(eth0) [eth0]: указать сетевой интерфейс

               --== VM CONFIGURATION ==--

               Booting from cdrom on RHEL7 is ISO image based only,
as cdrom passthrough is disabled (BZ760885)
               Please specify the device to boot the VM from (choose
disk for the oVirt engine appliance)
               (cdrom, disk, pxe) [disk]: cdrom
               Please specify the console type you would like to use
to connect to the VM (vnc, spice) [vnc]: vnc
               The following CPU types are supported by this host:
                       - model_Westmere: Intel Westmere Family
                       - model_Nehalem: Intel Nehalem Family
                       - model_Penryn: Intel Penryn Family
                       - model_Conroe: Intel Conroe Family
               Please specify the CPU type to be used by the VM
[model_Westmere]: model_Westmere
```

```
              Please specify the number of virtual CPUs for the VM
(Defaults to minimum requirement): [2]: 2
              Please specify the disk size of the VM in GB
(Defaults to minimum requirement): [25]: указать размер диска ВМ
              You may specify a unicast MAC address for the VM or
accept a randomly generated default [00:16:3e:0d:f1:b7]: указать mac
адрес ВМ
     Please specify the memory size of the VM in MB [Defaults to
minimum requirement: 4096]: 4096 размер оперативной памяти ВМ
              Please specify path to installation media you would
like to use [None]: /tmp/server-sintezm.iso полный путь где
находится .iso образ синтезМ сервер
     [ INFO  ] Stage: Clean up
     [ INFO  ] Generating answer file '/var/lib/ovirt-hosted-engine-
setup/answers/answers-20170314190516.conf'
     [ INFO  ] Stage: Pre-termination
     [ INFO  ] Stage: Termination
              Log file is located at /var/log/ovirt-hosted-engine-
setup/ovirt-hosted-engine-setup-20170313213725-mrj9mx.log
```

Примечание. Если во время установки произошли ошибки, необходимо произвести следующие действия:

```
yum erase \*ovirt\* \*vdsm\* \*libvirt\* -y
dd if=/dev/zero of=/dev/sd* где * буква диска соответствующая
подключенному таргету
reboot -f
```

Затем провести установку заново, согласно данного руководства системного программиста.

3) Вариант использования оптической системы хранения данных Fibre Channel

```
hosted-engine --deploy

[ INFO  ] Stage: Initializing
[ INFO  ] Generating a temporary VNC password.
[ INFO  ] Stage: Environment setup
          During customization use CTRL-D to abort.
          Continuing will configure this host for serving as
hypervisor and create a VM where you have to install the engine
afterwards.
          Are you sure you want to continue? (Yes, No)[Yes]:
Yes
          It has been detected that this program is executed
through an SSH connection without using screen.
          Continuing with the installation may lead to broken
installation if the network connection fails.
```

```
                It is highly recommended to abort the installation
and run it inside a screen session using command "screen".
                Do you want to continue anyway? (Yes, No)[No]: Yes
     [WARNING] Cannot detect if hardware supports virtualization
                Configuration files: []
                Log  file:  /var/log/ovirt-hosted-engine-setup/ovirt-
hosted-engine-setup-20170313213725-mrj9mx.log
                Version: otopi-1.5.2 (otopi-1.5.2-1.el7ev)
     [ INFO  ] Stage: Environment packages setup
     [ INFO  ] Stage: Programs detection
     [ INFO  ] Stage: Environment setup
     [ INFO  ] Generating libvirt-spice certificates
     [WARNING] Cannot locate gluster packages, Hyper Converged setup
support will be disabled.
     [ INFO  ] Please abort the setup and install vdsm-gluster,
gluster-server >= 3.7.2 and restart vdsmd service in order to gain
Hyper Conve
     [ INFO  ] Stage: Environment customization

                --== STORAGE CONFIGURATION ==--

     Please specify the storage you would like to use (glusterfs,
iscsi, fc, nfs3, nfs4)[nfs3]: fc
```

После проведения вышеописанных команд, на экран будет выведен список доступных лун, подключенных оптическим кабелем, необходимо выбрать необходимую луну.

Если настройка данного этапа прошла успешно, то в консоли отобразится следующее сообщение:

```
     [ INFO ] Creating VM
     You can now connect to the VM with the following command:
     /bin/remote-viewer vnc://localhost:5900
     Use temporary password "2560PHfZ" to connect to vnc console.
     Please note that in order to use remote-viewer you need to be
able to run graphical applications.
     This means that if you are using ssh you have to supply the -Y
flag (enables trusted X11 forwarding).
     Otherwise you can run the command from a terminal in your
preferred desktop environment.
     If you cannot run graphical applications you can connect to the
graphic console from another host or connect to the serial console
using the following command:
     socat    UNIX-CONNECT:/var/run/ovirt-vmconsole-console/36a712b5-
8bd3-4e4d-819f-6b19bd700a62.sock,user=ovirt-vmconsole
STDIO,raw,echo=0,escape=1
```

```
     Please ensure that your Guest OS is properly configured to
support serial console according to your distro documentation.
     Follow
http://www.ovirt.org/Serial_Console_Setup#I_need_to_access_the_conso
le_the_old_way for more info.
     If you need to reboot the VM you will need to start it manually
using the command:
     hosted-engine --vm-start
     You can then set a temporary password using the command:
     hosted-engine --add-console-password
     The VM has been started.
     To continue please install OS and shutdown or reboot the VM.
     Make a selection from the options below:
     (1) Continue setup - OS installation is complete
     (2) Abort setup
     (3) Power off and restart the VM
     (4) Destroy VM and abort setup
     (1, 2, 3, 4)[1]:
```

Далее на вновь созданную ВМ необходимо установить операционную систему. Для этого, на АРМ-установщике необходимо открыть видео-сессию и ввести следующую команду:

```
/bin/remote-viewer vnc://localhost:5900,
```

где localhost – это адрес сервера виртуализации. Для ввода пароля сессии, необходимо просмотреть INFO о создании ВМ (сообщение о прохождении первого этапа настройки).

В открывшейся видео-сессии отобразится окно установки сервера (см. рисунок 1), необходимо выбрать пункт «Установить СинтезМ-Сервер».

Далее необходимо провести установку сервера, выполнив действия, описанные в п.п. 3.1 и 3.2 настоящего документа.

При проведении установки необходимо указать следующие параметры:

– доменное имя – manager.test.ru;

– собственный IP-адрес – 10.10.2.221;

– жесткий диск необходимо разбить на разделы, аналогичные указанным в п. 3.1.2 настоящего документа, но без добавления раздела «ovirt-engine»;

– при выборе программы необходимо выбрать установку сервера управления ВМ.

Далее необходимо дождаться установки ОС на ВМ и выполнить перезагрузку системы после ее окончания.

Затем необходимо вернуться в консоль сервера виртуализации, нажать 1 и клавишу ввода и ожидать вывода сообщения о продолжения установки.

Далее необходимо зайти по ssh на виртуальный сервер авторизации по адресу 10.10.2.221 и ввести пароль, заданный при установке:

```
ssh root@10.10.2.221
```

После авторизации необходимо произвести первоначальные настройки, аналогичные описанным в п.п. 3.2.1 и 3.2.2 настоящего документа.

Затем необходимо прописать адрес менеджера и сервера виртуализации в /etc/hosts на менеджере.

Потом необходимо скопировать образ системы на менеджер, смонтировать образ в папку mnt и создать репозиторий.

Пример:

```
scp /sintezm/sintezm-server.iso root@10.10.2.221:/tmp
```

На ВМ менеджера:

```
Примонтировать .iso образ
mount -o loop /tmp/sintezm-server.iso /mnt

Добавить репозиторий
vim /etc/yum.repos.d/sintezm.repo
name=base-sintezm-$releasever
baseurl=file:///mnt
enabled=1
gpgcheck=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-sintezm-release
```

После первичной настройки ВМ менеджера, необходимо выполнить команду конфигурирования ВМ авторизации и ответить на вопросы системы:

```
engine-setup
[ INFO  ] Stage: Initializing
[ INFO  ] Stage: Environment setup
Configuration    files:    ['/etc/ovirt-engine-setup.conf.d/10-
packaging-wsp.conf',             '/etc/ovirt-engine-setup.conf.d/10-
packaging.conf']
Log    file:    /var/log/ovirt-engine/setup/ovirt-engine-setup-
20170316172643-sdgxek.log
```

```
     Version: otopi-1.5.2 (otopi-1.5.2-1.el7ev)
     [ INFO  ] Stage: Environment packages setup
     [ INFO  ] Stage: Programs detection
     [ INFO  ] Stage: Environment setup
     [ INFO  ] Stage: Environment customization

               --== PRODUCT OPTIONS ==--

     Configure Engine on this host (Yes, No) [Yes]: Yes
     Configure Image I/O Proxy on this host? (Yes, No) [Yes]: Yes
     Configure WebSocket Proxy on this host (Yes, No) [Yes]: Yes
     Please note: Data Warehouse is required for the engine. If you
choose to not configure it on this host, you have to configure it on
a remote host, and then configure the engine on this host so that it
can access the database of the remote Data Warehouse host.
     Configure Data Warehouse on this host (Yes, No) [Yes]: Yes
     Configure VM Console Proxy on this host (Yes, No) [Yes]: Yes

               --== PACKAGES ==--

     [ INFO  ] Checking for product updates...
     [ INFO  ] No product updates found

               --== NETWORK CONFIGURATION ==--

     Host fully qualified DNS name of this server [manager.test.ru]:
     [WARNING] Failed to resolve manager.test.ru using DNS, it can
be resolved only locally
     Setup can automatically configure the firewall on this system.
     Note: automatic configuration of the firewall may overwrite
current settings.
     Do you want Setup to configure the firewall? (Yes, No) [Yes]:
No

               --== DATABASE CONFIGURATION ==--

     Where is the DWH database located? (Local, Remote) [Local]:
     Setup can configure the local postgresql server automatically
for the DWH to run. This may conflict with existing applications.
     Would you like Setup to automatically configure postgresql and
create DWH database, or prefer to perform that manually? (Automatic,
Manual) [Automatic]:
     Where is the Engine database located? (Local, Remote) [Local]:
     Setup can configure the local postgresql server automatically
for the engine to run. This may conflict with existing applications.
     Would you like Setup to automatically configure postgresql and
create Engine database, or prefer to perform that manually?
(Automatic, Manual) [Automatic]:

               --== OVIRT ENGINE CONFIGURATION ==--
```

```
          Engine admin password:
          Confirm engine admin password:
[WARNING] Password is weak: слишком простой
          Use weak password? (Yes, No) [No]: Yes
          Application mode (Virt, Gluster, Both) [Both]:

          --== STORAGE CONFIGURATION ==--

          Default SAN wipe after delete (Yes, No) [No]: No

          --== PKI CONFIGURATION ==--

          Organization name for certificate [test.ru]:

          --== APACHE CONFIGURATION ==--
```

Setup can configure the default page of the web server to present the application home page. This may conflict with existing applications.

Do you wish to set the application as the default page of the web server? (Yes, No) [Yes]: Yes

Setup can configure apache to use SSL using a certificate issued from the internal CA.

Do you wish Setup to configure that, or prefer to perform that manually? (Automatic, Manual) [Automatic]:

```
          --== SYSTEM CONFIGURATION ==--
```

Configure an NFS share on this server to be used as an ISO Domain? (Yes, No) [No]: Yes

Local ISO domain path [/var/lib/exports/iso]:

Please provide the ACL for the Local ISO domain.

See the exports(5) manpage for the format.

Examples:

- To allow access for host1, host2 and host3, input: host1(rw) host2(rw) host3(rw)
- To allow access to the entire Internet, input: *(rw)

For more information, see: http://www.ovirt.org/Troubleshooting_NFS_Storage_Issues

```
          Local ISO domain ACL: *(rw)
          Local ISO domain name [ISO_DOMAIN]:
          --== MISC CONFIGURATION ==--

          Please choose Data Warehouse sampling scale:
          (1) Basic
          (2) Full
          (1, 2)[1]: 1
```

```
          --== END OF CONFIGURATION ==--

[ INFO  ] Stage: Setup validation
[WARNING] Less than 16384MB of memory is available

          --== CONFIGURATION PREVIEW ==--

Application mode                        : both
Default SAN wipe after delete           : False
Update Firewall                         : False
Host FQDN                               : manager.test.ru
Engine database secured connection      : False
Engine database host                    : localhost
Engine database user name               : engine
Engine database name                    : engine
Engine database port                    : 5432
Engine database host name validation    : False
DWH database secured connection         : False
DWH database host                       : localhost
DWH database user name                  : ovirt_engine_history
DWH database name                       : ovirt_engine_history
DWH database port                       : 5432
DWH database host name validation       : False
Engine installation                     : True
NFS setup                               : True
PKI organization                        : test.ru
NFS export ACL                          : *(rw)
NFS mount point                         : /var/lib/exports/iso
Configure local Engine database         : True
Set application as default page         : True
Configure Apache SSL                    : True
DWH installation                        : True
Configure local DWH database            : True
Engine Host FQDN                        : manager.test.ru
Configure Image I/O Proxy               : True
Configure VMConsole Proxy               : True
Configure WebSocket Proxy               : True

Please confirm installation settings (OK, Cancel) [OK]: OK
```

Далее необходимо дождаться окончания конфигурирования ВМ авторизации, во время конфигурации ошибок быть не должно. После ее завершения необходимо перейти обратно в консоль сервера виртуализации, выбрать 1 пункт, нажать [1] и клавишу ввода.

На экране появится сообщение:

```
Checking for oVirt-Engine status at manager.test.ru...
[ INFO  ] Engine replied: DB Up!Welcome to Health Status!
```

```
     [ INFO  ] Acquiring internal CA cert from the engine
     [ INFO  ] The following CA certificate is going to be used,
please immediately interrupt if not correct:
     [ INFO  ] Issuer: C=US, O=test.ru, CN=manager.test.ru.30298,
Subject:C=US,O=test.ru,CN=manager.test.ru.30
298, Fingerprint (SHA-1): A40AB186F674EA610CA6D1C1A80EFEB0F10DADD5
     [ INFO  ] Connecting to the Engine

     Checking for oVirt-Engine status at manager.test.ru...
     [ INFO  ] Engine replied: DB Up!Welcome to Health Status!
     [ INFO  ] Waiting for the host to become operational in the
engine. This may take several minutes...
     [ INFO  ] Still waiting for VDSM host to become operational...
     [ INFO  ] The VDSM Host is now operational
     [ INFO  ] Saving hosted-engine configuration on the shared
storage domain
     Please shutdown the VM allowing the system to launch it as a
monitored service.
     The system will wait until the VM is down.
     [ INFO  ] Enabling and starting HA services
     [ INFO  ] Stage: Clean up
     [ INFO  ] Generating answer file '/var/lib/ovirt-hosted-engine-
setup/answers/answers-20170316180713.conf'
     [ INFO  ] Generating answer file '/etc/ovirt-hosted-
engine/answers.conf'
     [ INFO  ] Stage: Pre-termination
     [ INFO  ] Stage: Termination
     [ INFO  ] Hosted Engine successfully deployed
```

После вывода следующего сообщения:

```
     Please shutdown the VM allowing the system to launch it as a
monitored service.
     The system will wait until the VM is down
```

необходимо перейти обратно на ВМ авторизации и ввести команду на выключение poweroff. Сервер виртуализации автоматически до настроит менеджер и выведет сообщение об успешной настройке:

```
     [ INFO  ] Hosted Engine successfully deployed
```

Менеджер установлен и настроен. Для того, чтобы зайти в менеджер управления виртуальными машинами, необходимо открыть браузер, ввести доменное имя авторизации или ip адрес ВМ, подтвердить в браузере конфиденциальность страницы, выбрать админ портал, авторизоваться пользователем admin и ранее заданным паролем для этого пользователя, домен «internel». На данной странице находится интерфейс управления ВМ.

Для продолжения работы необходимо добавить хранилище в систему виртуализации. Подключение может быть реализовано как к локальному хранилищу (на гипервизоре), так и к внешней СХД.

1) Для создания локального хранилища (на гипервизоре) необходимо перейти на сервер виртуализации в консольном режиме, создать папку для хранения данных, задать ей права доступа и перезапустить службу экспортирования.

Пример:

```
mkdir /storage
chmod 777 /storage/
chown vdsm.kvm /storage/
vim /etc/exports
/ovirt-engine *(rw)
/storage *(rw)
:wq

service nfs-server restart
```

Далее необходимо перейти в веб-интерфейс ВМ управления, авторизоваться, перейти во вкладку «storage», нажать в заголовке «New Domain». В появившемся окне (Рисунок 22) выбрать соответствующий data-center (default), функцию домена хранение (data), тип подключаемого домена (NFS), узел, на котором будет задействован домен и путь к каталогу хранения (Export Path).

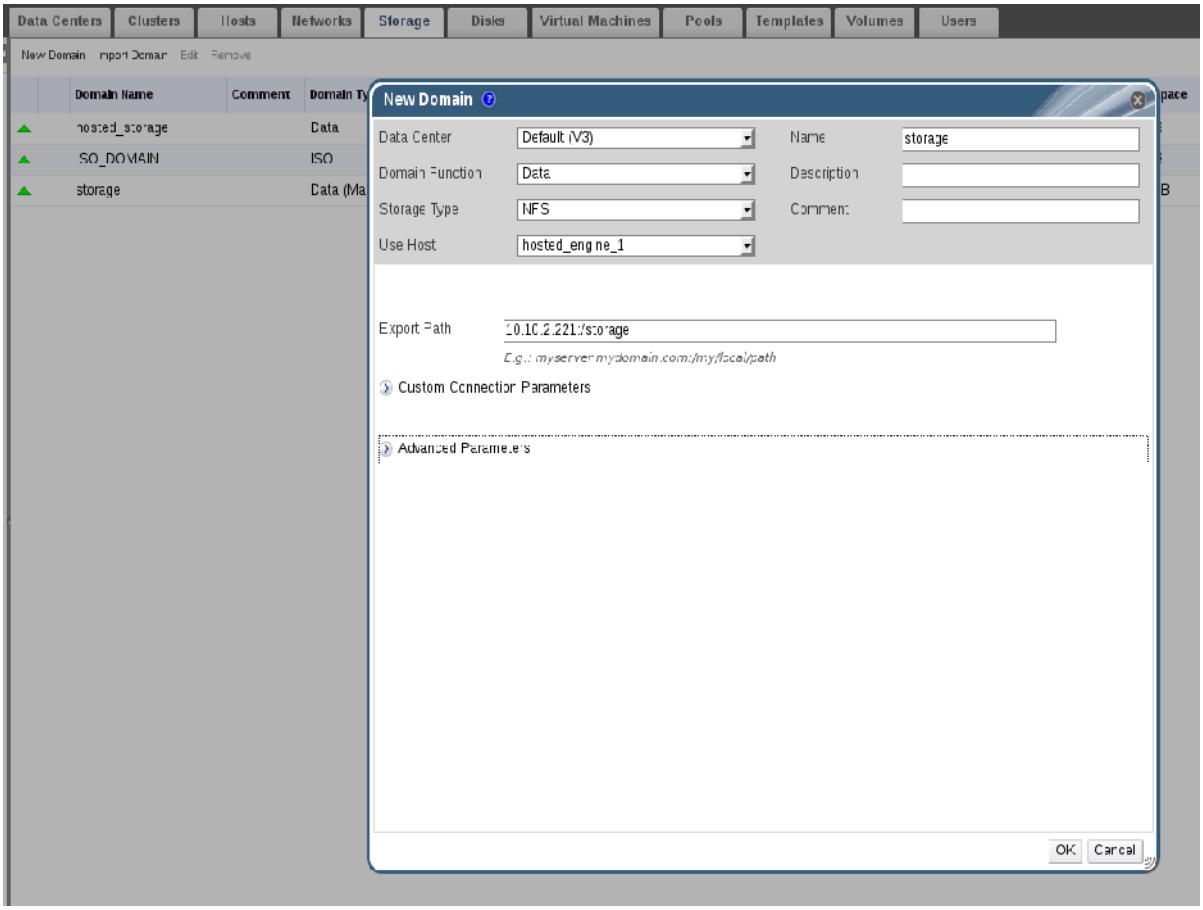

Рисунок 22 – Окно «New Domain» - подключение системы хранения данных NFS

Далее необходимо нажать на кнопку «OK» и дождаться инициализации хранилища, затем следует в ручном режиме активировать ISO_DOMAIN (нажав кнопку attach при выборе ISO_DOMAIN во вкладке Storage).

Примечание. При добавлении нового хранилища автоматически добавится хранилище виртуальной машины управления.

2) Для подключения системы хранения данных iSCSI, необходимо перейти в веб-интерфейс ВМ управления, авторизоваться, перейти во вкладку «storage», нажать в заголовке «New Domain». В появившемся окне (Рисунок 23) выбрать соответствующий data-center (default), функцию домена хранения (data), тип подключаемого домена (iSCSI).

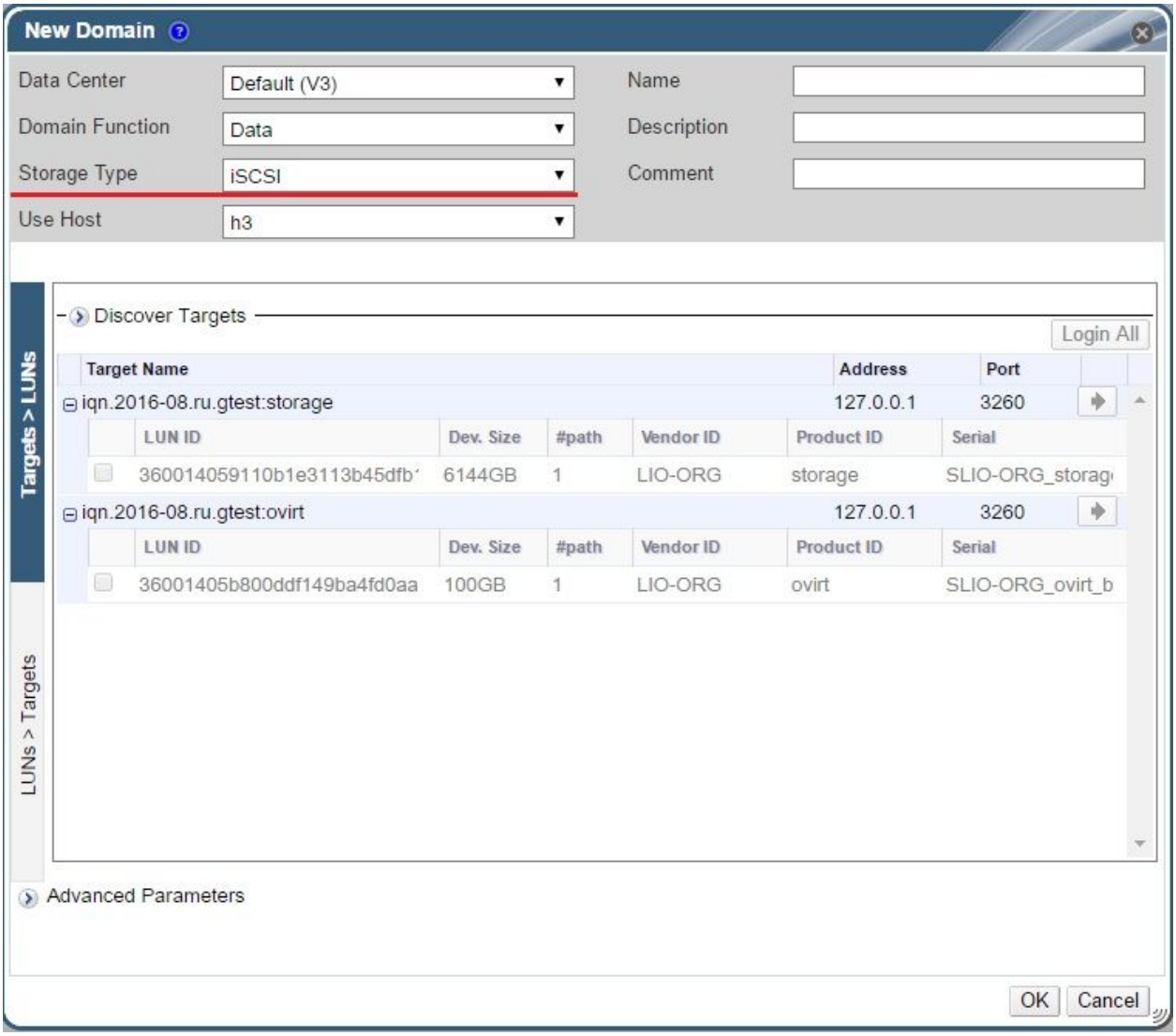

Рисунок 23 – Окно «New Domain» - подключение системы хранения данных iSCSI

Далее отобразится список подключенных лун, необходимо выбрать нужную и нажать на «OK».

3) Для подключения оптической системы хранения данных Fibre Channel, необходимо перейти в веб-интерфейс ВМ управления, авторизоваться, перейти во вкладку «storage» и нажать в заголовке «New Domain». В появившемся окне (Рисунок 24) выбрать соответствующий data-center (default), функцию домена хранения (data), тип подключаемого домена (Fibre Channel).

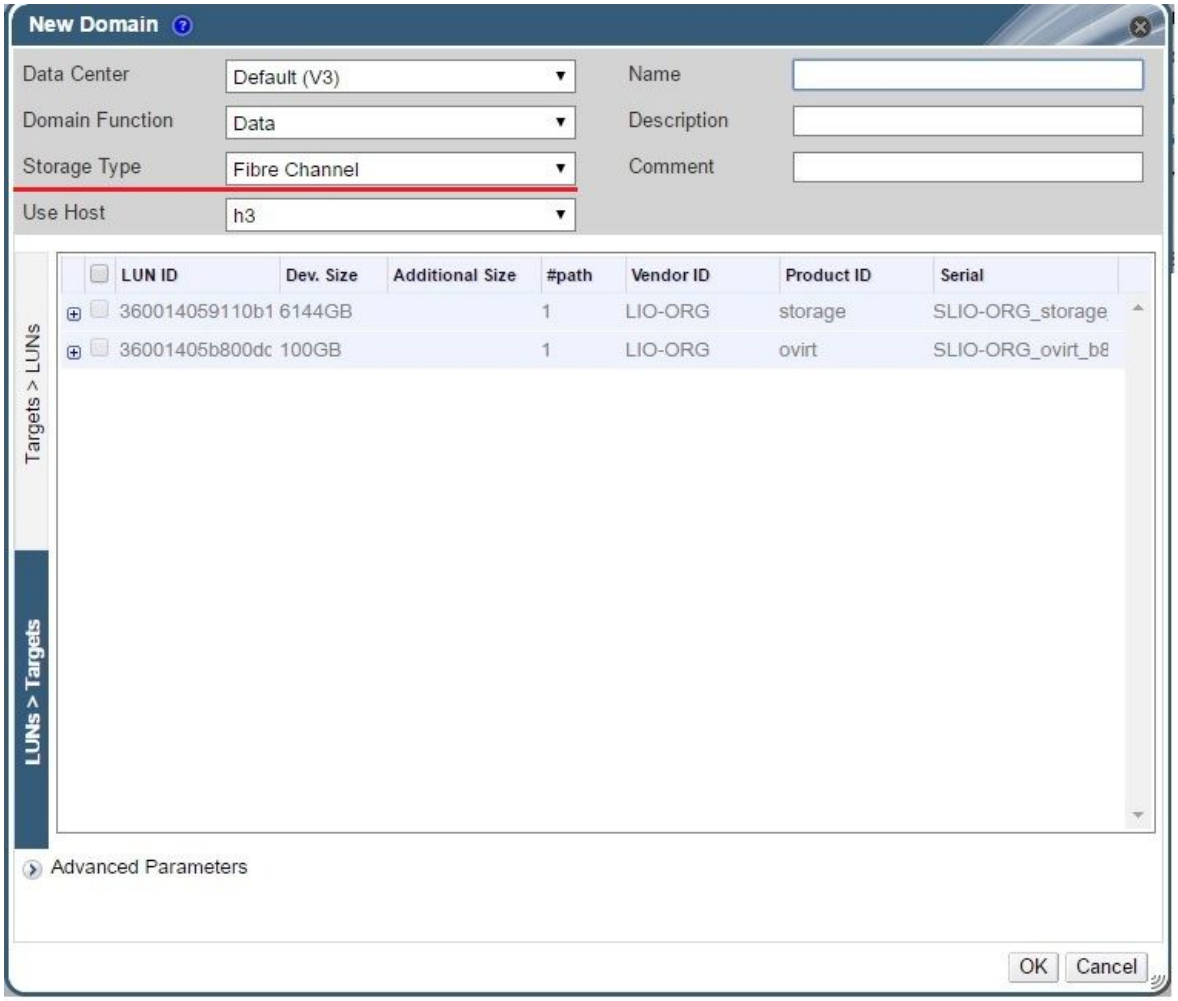

Рисунок 24 – Окно «New Domain» - подключение системы хранения данных Fibre Channel

Далее отобразится список подключенных лун, необходимо выбрать нужную и нажать на «OK».

Установка и настройка системы виртуализации изделия закончена. Система настроена, можно приступать к дальнейшему ее эксплуатированию, созданию ВМ, добавлению новых узлов и т.д.

Сервера имеют по 2 физических диска. Разделы на обоих узлах размечены одинаково. Операционная система устанавливалась на каждый сервер отдельного диска. Диски предварительно создавались в raid-0 каждый.

3.4. Добавление сервера виртуализации

Для добавления гипервизора (сервера виртуализации) в систему виртуализации, необходимо зайти в него по ssh, выполнив команду:

```
ssh root@10.10.2.51
```

После авторизации необходимо произвести первоначальные настройки, как и с первым сервером виртуализации: настроить время, занести данные о всех серверах виртуализации и о виртуальной машине авторизации в файл hosts. Далее необходимо отключить «iptables» и «firewalld», настроить службу «ntpd» на прием времени и включить ее в автозагрузку.

Далее необходимо ввести команду engine-setup и ответить на вопросы, заданные системой.

3.5. Создание виртуальной машины

Для создания виртуальной машины необходимо зайти на административный портал сервера авторизации. Для этого на АРМ-установщике необходимо запустить браузер, в адресной строке ввести адрес сервера авторизации, пройти аутентификацию (рисунок 25), перейти на вкладку «Virtual Machines» и нажать на кнопку «New VM» (рисунок 26).

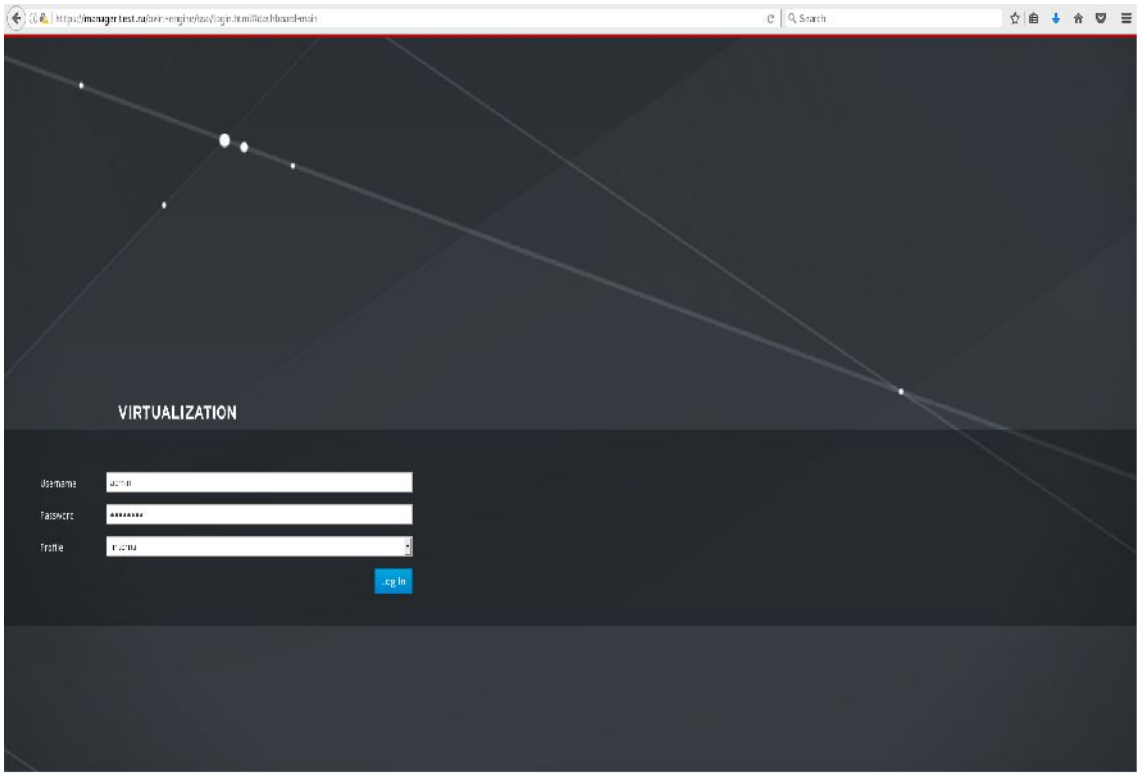

Рисунок 25 – Окно авторизации менеджера

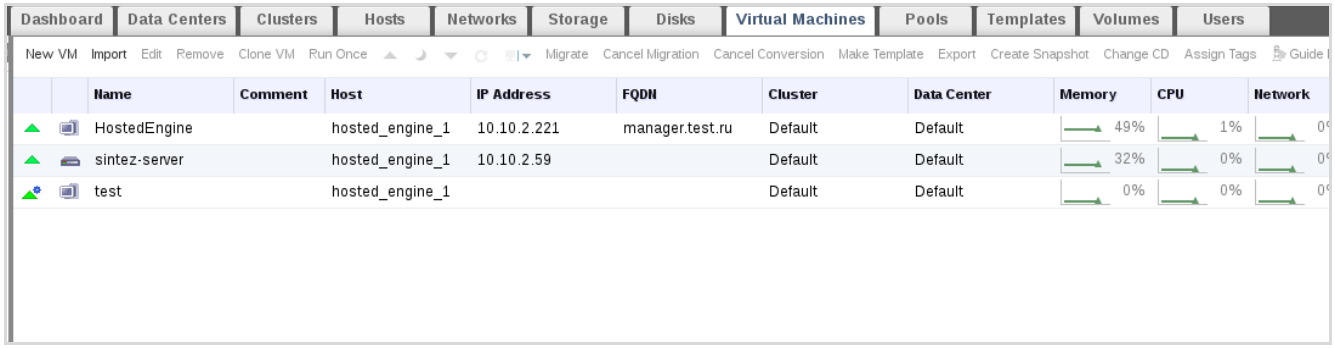

Рисунок 26 – Окно управления ВМ

В открывшемся окне настройки параметров виртуальной машины необходимо перейти во вкладку «General» и нажать на кнопку «Show Advanced Options» (рисунок 27).

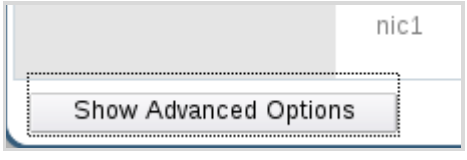

Рисунок 27 – Показать дополнительные параметры настройки ВМ

Далее необходимо настроить следующие параметры ВМ (Рисунок 28):

– «cluster» – выбрать Data Center, на котором будет функционировать ВМ («Default»);

– «template» – указывается в случае, если установка ВМ идет с готового шаблона (оставить без изменений);

– «operating system» – выбрать из ниспадающего списка устанавливаемую ОС и нажать на кнопку «ОК» («РК Sintezm-Sever»);

– «instance type» – автоматически задает параметры ВМ в зависимости от выбранного типа («Custom»);

– «optimized for» – указывает оптимизацию под серверное решение или рабочую станцию. Соответственно если выполняется установка виртуального сервера выбрать оптимизацию «server», если рабочей станции – то «desktop»;

– «name» – указывается имя виртуальной машины;

– «VM ID» – указывается индикационный номер виртуальной машины в базе данных. По умолчанию данный параметр можно не задавать и оставить его пустым;

– «description» – в данном поле указывается примечание для данной ВМ, можно оставить поле пустым или для удобства пометить, к примеру «Начальник отдела»;

– «comment» – аналогичен параметру «Description»;

– «stateless», «start in pause mode» и «delete protection» – необходимо оставить неактивными.

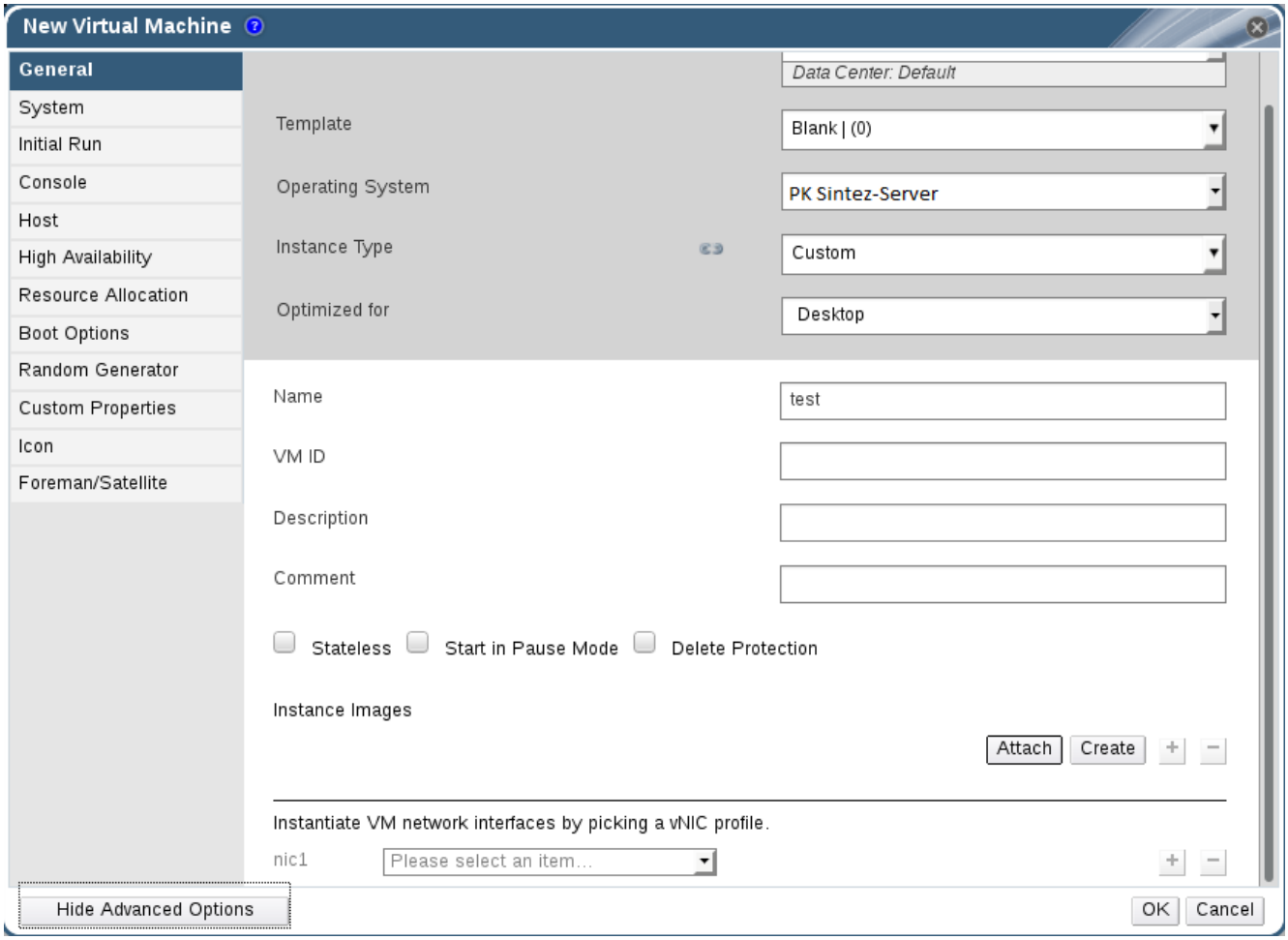

Рисунок 28 – Настройка параметров виртуальной машины

Параметр «instance image» позволяет добавлять или создавать виртуальный жесткий диск. Для этого необходимо нажать на кнопку «Create». В открывшемся окне (рисунок 29) необходимо указать параметры жесткого диска – размер, место хранения и другие параметры:

– Size GB – размер жесткого диска;

– Alias – название диска (оставить по умолчанию);

– Description – примечание (оставить по умолчанию);

– Interface – интерфейс жесткого диска (оставить по умолчанию);

– Storage Domain – домен хранения виртуальных образов (указать, где будут храниться ВМ);

– Allocation Policy – политика разбиения жесткого диска (оставить по умолчанию);

– Disk Profile – профиль диска (оставить по умолчанию);

– Wipe after Delete – очистить место после удаления (оставить по умолчанию);

– Bootable – диск будет загрузочным (оставить по умолчанию);

– Shareable – сделать общим доступным (оставить по умолчанию);

– Read Only – сделать диск только для чтения (оставить по умолчанию).

После настройки параметров жесткого диска необходимо нажать на кнопку «OK».

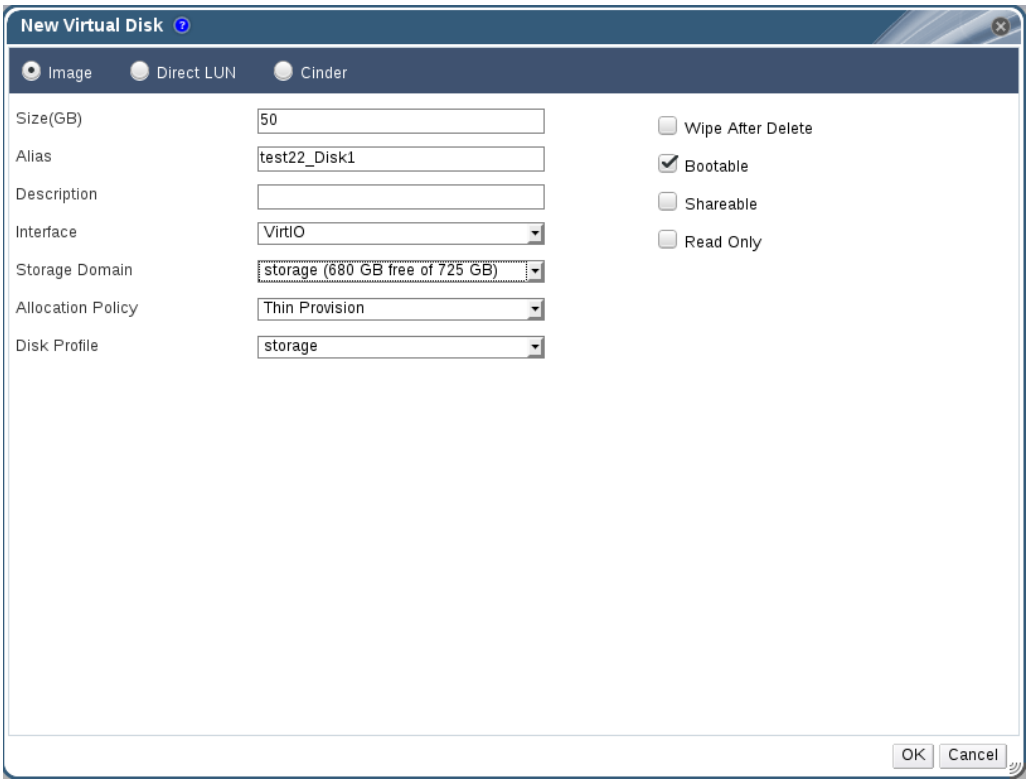

Рисунок 29 – Создания нового виртуального диска

Параметр «Instantiate VM network» позволяет выбрать сетевой интерфейс для данной ВМ – из ниспадающего списка необходимо выбрать подключение по стандарту «ovirtmgmt».

Примечание. Для того, чтобы изменить настройки сети, необходимо перейти во вкладку «Virtual Machines», выбрать ВМ, далее во вкладках данной ВМ выбрать «Network Interfaces», выбрать подключение и нажать на кнопку «Edit» (рисунок 30).

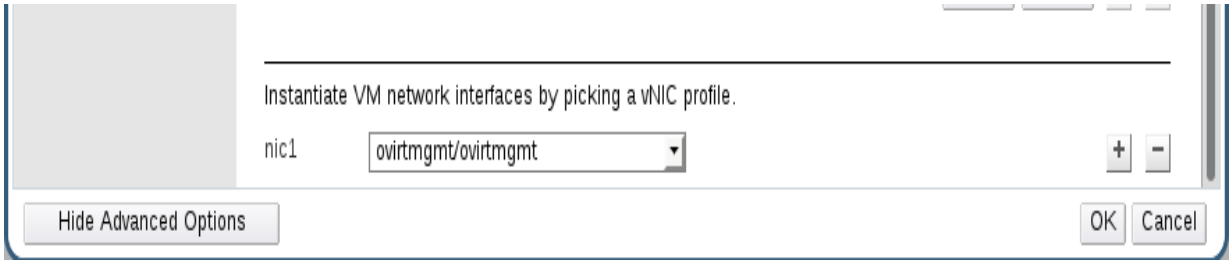

Рисунок 30 – Выбор сетевого подключения

Далее необходимо перейти во вкладку «system» и указать следующие параметры (рисунок 31):

– «memory size» – объем выделяемой памяти для ВМ;

– «Total Virtual CPUs» – количество выделяемых процессоров.

Примечание. В дополнительных параметрах CPU, указывается число виртуальных сокетов, число ядер на узле, тип процессора;

– «hardware clock time offset» – указывает смещение времени (то есть часовой пояс), указать пояс места нахождения сервера.

Остальные параметры необходимо оставить без изменений.

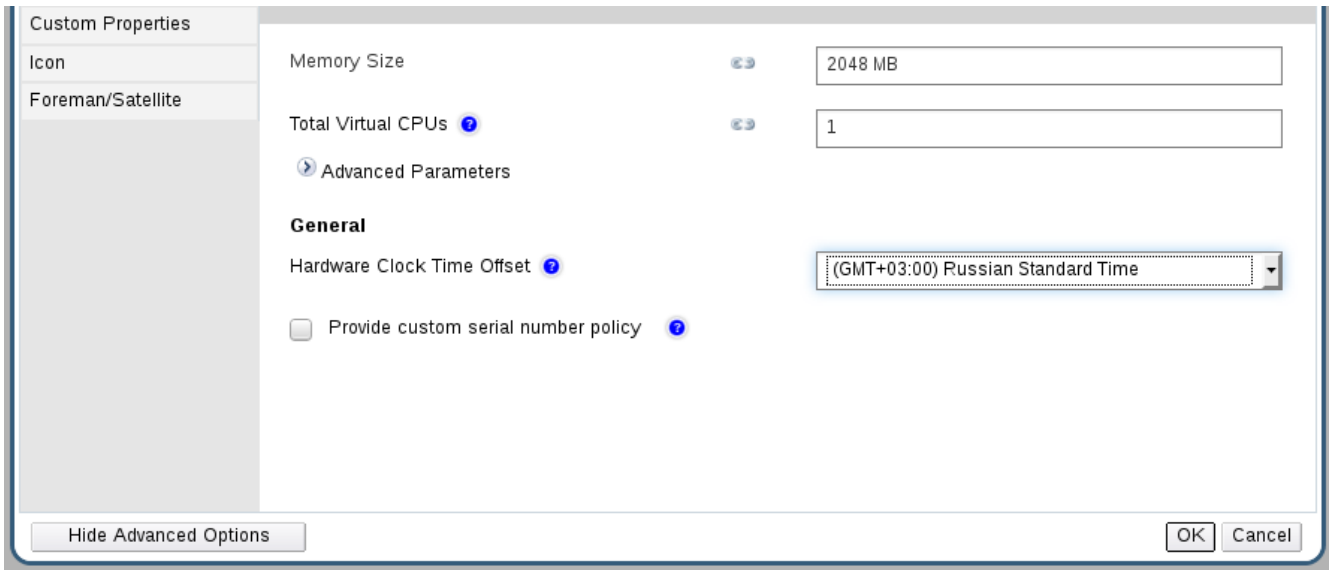

Рисунок 31 – Вкладка «System»

Далее необходимо перейти во вкладку «console» и указать следующие параметры (рисунок 32):

– «Video Type» – выбрать режим «QXL»;

– «Graphics protocol» – выбрать «Spice». Данный параметр выбирает графический протокол передачи данных;

– «USB Support» – оставить в выключенном положении. Данный параметр включает или выключает поддержку usb накопителей для ВМ;

– «Console Disconnect Action» – указать параметр lock screen. Данный параметр указывает действия ВМ при отключении пользователя;

– «Monitors» – указать количество мониторов у данной ВМ;

– «SmartCard Enabled» – оставить не активным, а параметр Single Sign On Method выбрать «Use Guest Agent»;

– в «Advanced Parameters» – ничего не менять;

– «Disable strict user checking» – отключение блокировки spice сессии при работе администратора (при необходимости).

– «Soundcard enable» – включение звуковой карты для данной ВМ (при необходимости).

Остальные параметры необходимо оставить без изменений.

Далее необходимо перейти во вкладку «Host», все настройки оставить без изменений (рисунок 33):

– «Start Runnig On» – настраивает запуск ВМ на определенном узле или на любом доступном;

– «Migration Options» – настраивает дополнительные параметры при миграции виртуальной машины на другие узлы;

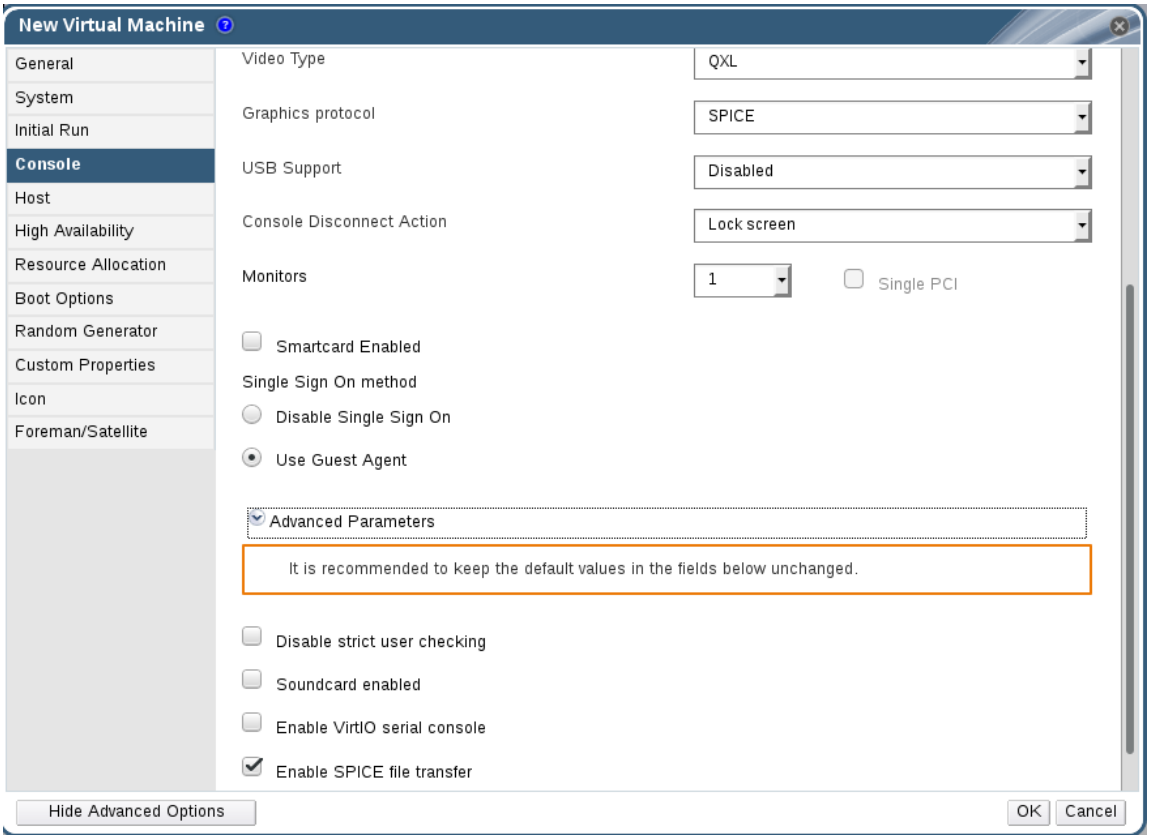

Рисунок 32 – Вкладка «Console»

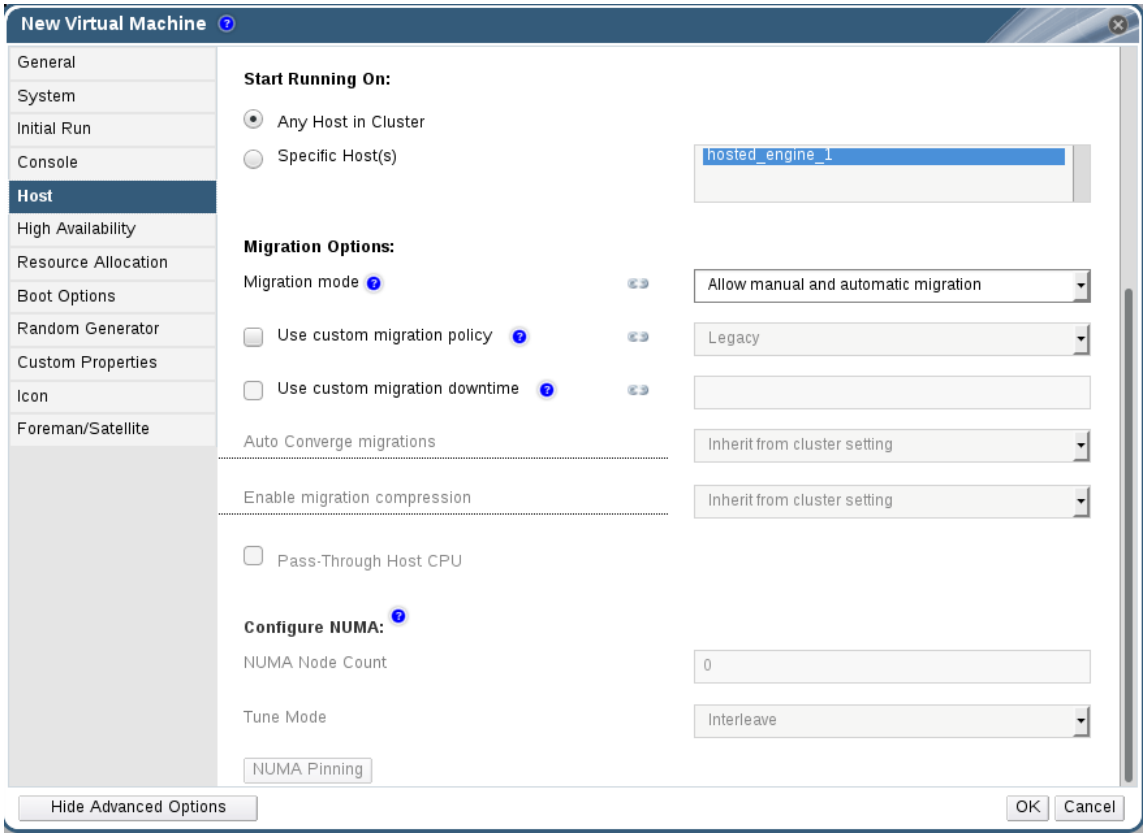

Рисунок 33 – Вкладка «Host»

Далее необходимо перейти во вкладку «High Availability» и указать следующие параметры (рисунок 34):

– «High Availability» – выставить нужный приоритет миграции для ВМ;

– «Watchdog» – указать модель устройства из выпадающего списка и указать действие из выпадающего списка перезагрузки.

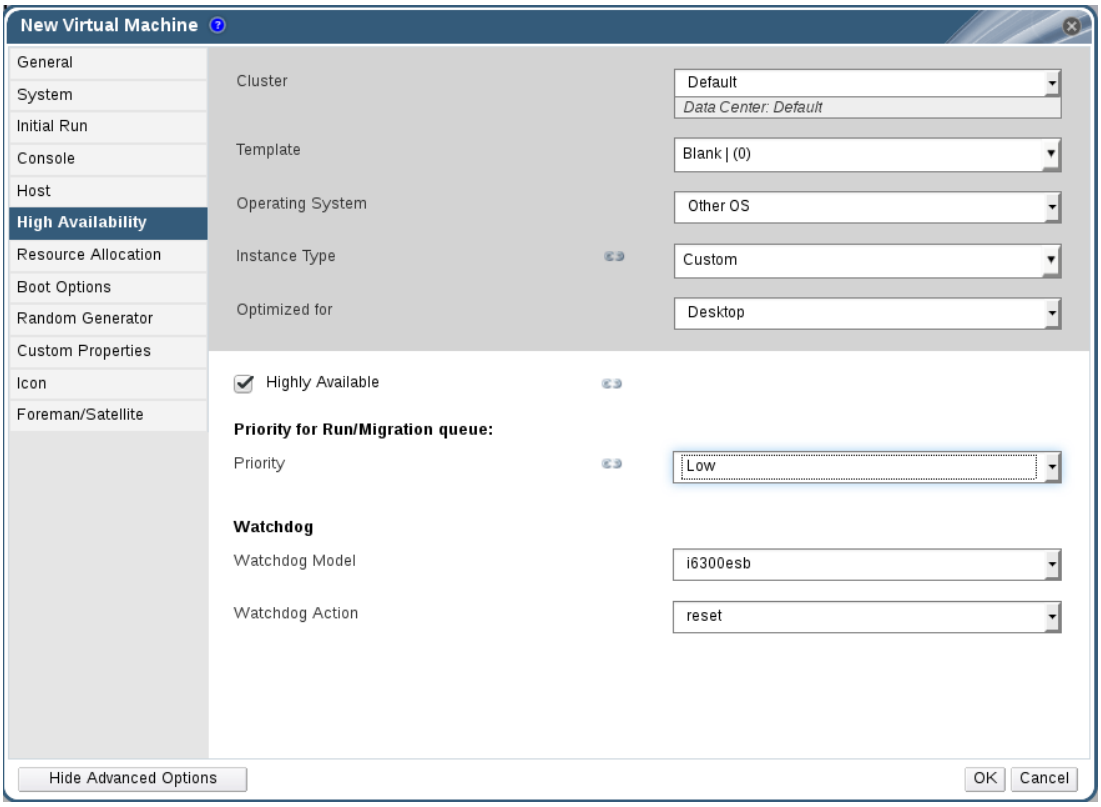

Рисунок 34 – Вкладка «High Availability»

Далее необходимо перейти во вкладку «Resource Allocation», все настройки оставив без изменений (рисунок 35):

– «CPU Allocation» – данный параметр устанавливает приоритет использования всех ресурсов для ВМ на хосте. Разбиение по группам приоритетов следующее: Normal имеет в два раза больший приоритет по сравнению с Low, а High в два раза больший приоритет по сравнению с Normal. Этот параметр вступает в силу только в том случае, когда выделенных ресурсов начинает не хватать;

– «Memory Allocation» – выдает гарантированный объем оперативной памяти ВМ.

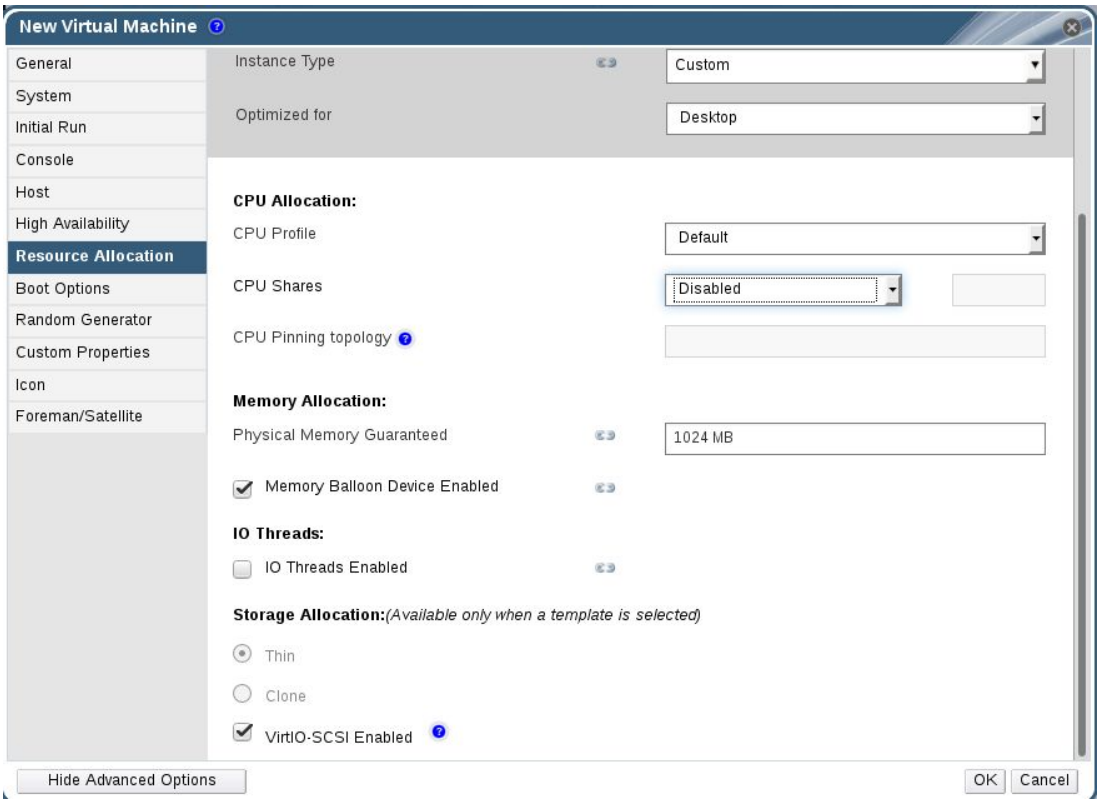

Рисунок 35 – Вкладка «Resource Allocation»

После завершения всех настроек необходимо нажать на кнопку «OK». ВМ будет создана и отобразится в списке (рисунок 36).

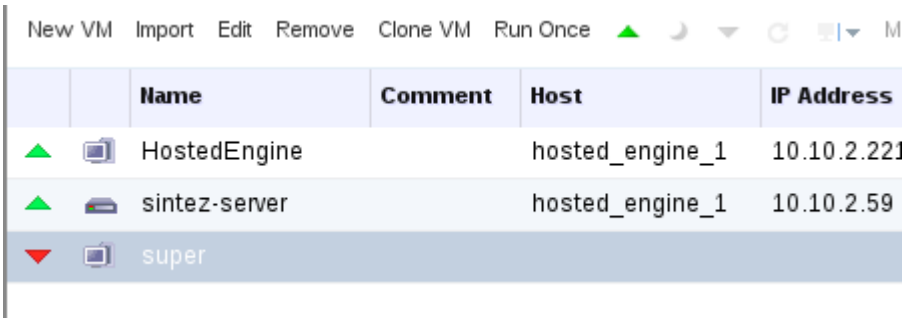

Рисунок 36 – Отображение созданной ВМ

Примечание. Перед тем как продолжить установку ВМ, необходимо скопировать образ системы на менеджере управления машин и выполнить команду загрузки образа в систему виртуализации. Для этого необходимо использовать команду `scp` на АРМ-установщике, а далее на менеджере выполнить команду загрузки образа в систему.

Пример:

На АРМ-установщике ввести команду:

```
scp /tmp/sintezm-client.iso root@10.10.2.221:/tmp
```

На сервере авторизации ввести команду:

```
engine-iso-uploader -i ISO_DOMAIN upload /tmp/sintezm-client.iso
```

После введения указанной команды, в консоли отобразится следующее сообщение:

```
Please provide the REST API password for the admin@internal oVirt Engine user (CTRL+D to abort): введите пароль от администратора портала виртуализации (internal)
```

Начнется загрузка образа, с отображением прогресса в консоли. Далее следует ожидать окончания загрузки.

Для запуска созданных ВМ необходимо:

– выбрать соответствующую ВМ из списка;

– нажать на кнопку «Run Once» (см. рисунок 36).

Откроется окно с параметрами разовой загрузки ВМ. В нем необходимо выбрать подменю «Boot Option», в данном меню можно выбрать загрузку с жесткого диска, загрузку с диска или по сети. Также существуют другие подменю параметров настройки данной ВМ: часовой пояс, выбор протокола по которому будут передаваться видео данные и т.д.

Значения дополнительных параметров необходимо оставить по умолчанию. Далее необходимо настроить загрузку с образа диска. Выставить маркер в поле «Attach CD», в выпадающем меню необходимо выбрать образ системы и нажать на кнопку «OK» (рисунок 37).

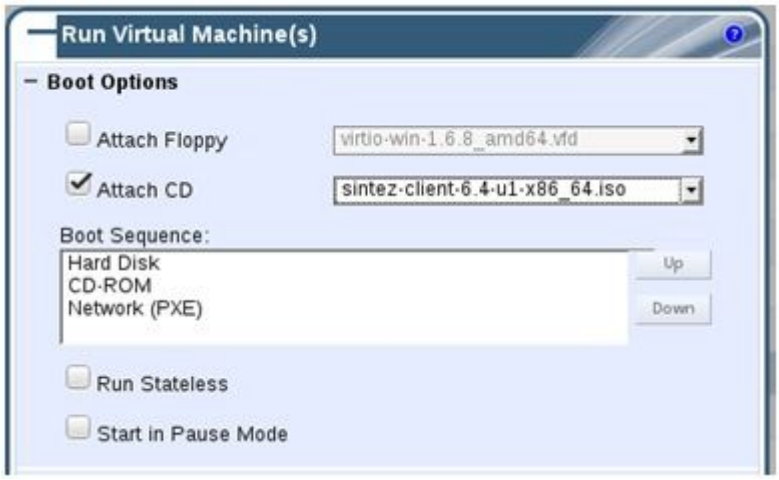

Рисунок 37 – Загрузка виртуальной машины с образа диска

3.5.1. Установка ОС на ВМ

Для запуска консоли виртуальной машины пользователя необходимо выбрать ВМ и нажать на кнопку «Console» (рисунок 38).

Рисунок 38 – Запуск консоли

Примечание. На машине установщика под управлением Linux должен быть установлен пакет remote-viewer для возможности отображения видеоконсоли. Данный пакет можно найти на серверном образе данной системы.

Примечание. На машине установщика должен быть настроен файл хост, в Linux OS это /etc/hosts в Windows OS это /windows/system32/driver/etc/host. В нем должна присутствовать запись о сервере авторизации (менеджере).

Пример:

```
10.10.2.221 manager.test.ru
```

В открывшемся окне необходимо выбрать параметр установки ОС и нажать клавишу [Enter]. После чего произойдет стандартная установка ОС, если это серверная часть – необходимо выполнить действия, аналогичные описанным в п.п. 3.1 и 3.2 настоящего документа. Порядок установки и настройки клиентской операционной системы ПК «СинтезМ-Клиент» приведен в эксплуатационном

документе ТАСП.62.01.11.000.003 32 01. Руководство системного программиста ПК «СинтезМ-Клиент».

3.6. Настройка АРМ установщика

Для дальнейшей настройки системы виртуализации необходимо настроить АРМ установщика. Для настройки АРМ установщика необходимо подключить АРМ к сети Ethernet (ЛВС), к которой подключен сервер, и настроить сетевой адрес АРМ. Это возможно сделать несколькими способами: в графическом интерфейсе или в командной строке:

1) в графическом интерфейсе в правом верхнем углу навести мышкой на иконку двух рядом стоящих мониторов. Нажать на нее правой кнопкой мыши и выбрать строку «Изменить соединения». В открывшемся окне выбрать строку System eth0 и нажать на кнопку «Изменить». В следующем окне выбрать вкладку Параметры IPv4. Установить значение поля «Профиль» – «Вручную». Нажать на кнопку «Добавить» и задать адрес. Маска подсети, адрес подсети и шлюз должны быть точно такими же, как у сервера виртуализации. В качестве сетевого адреса данного АРМ необходимо выбрать любой незадействованный.

После указания сетевого адреса необходимо нажать на кнопку «Применить» (рисунок 39).

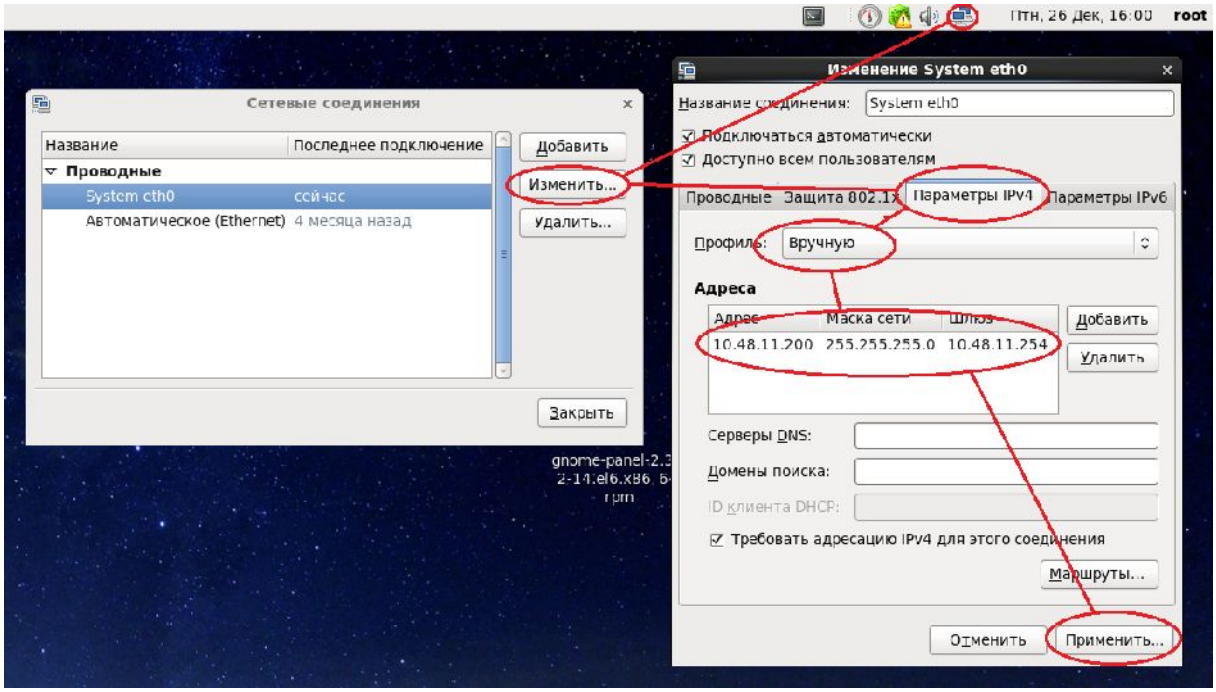

Рисунок 39 – Настройка АРМ-установщика

Далее в верхнем правом углу необходимо навести мышкой на иконку двух рядом стоящих мониторов, нажать на нее левой кнопкой мыши и выбрать строчку «Отключиться». Повторить данное действие, только в данном случае выбрать строчку System eth0 «Обновится данное подключение». Проверить командой `ping`.

В командной строке АРМ установщика (терминал) ввести команду:

```
ifconfig eth0 XX.XX.XX.XX/YY up
Где, XX.XX.XX.XX назначаемый ip адрес
YY маска подсети.
```

Пример:

```
# ifconfig eth0 10.48.11.200/24 up
```

После применение сетевых настроек перейти к редактированию файл host:

```
# vim /etc/hosts
```

В самом файле необходимо добавить IP-адрес менеджера и его доменное имя (рисунок 40). После внесения изменений необходимо сохранить файл и выйти из редактора.

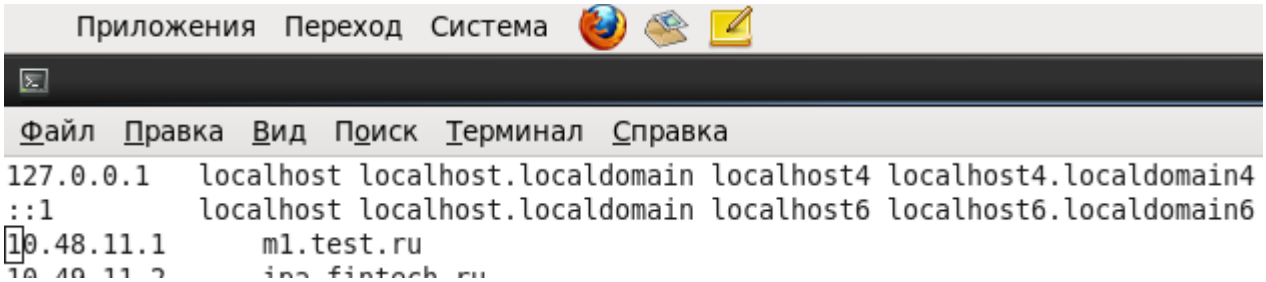

Рисунок 40 – Настройка hosts на АРМ-установщике

Далее необходимо открыть файл resolv.conf введя команду

```
# vim /etc/resolv.conf
```

В файле после строки:

```
# Generated by NetworkManager
```

Добавить строку:

```
search test.ru (Домен системы виртуализации)
nameserver .XX.XX.XX.XX
```

Где, XX.XX.XX.XX – ip адрес DNS сервера, (IPA) (Рисунок 41).

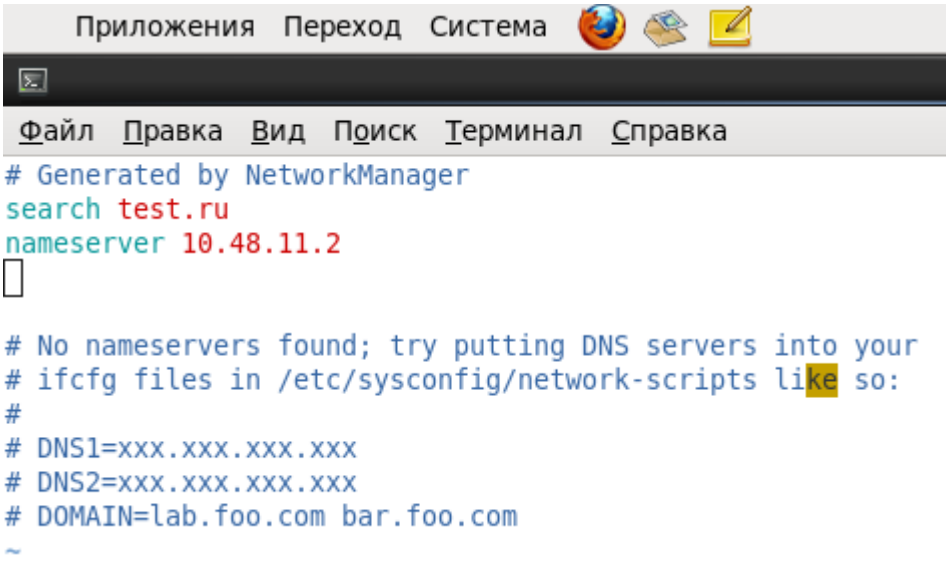

Рисунок 41 – Настройка АРМ-установщика файл Resolv.conf

Далее необходимо установить на АРМ установщика пакет spice_xpi командой:

```
# yum install remote-viewer
```

Данный пакет можно получить с репозитория изделия.

После выполнения всех описанных действий АРМ установщика будет настроен для установки и настройки сервера виртуализации и его компонентов.

3.7. Настройка терминального сервера

Для дальнейшей настройки терминального сервера необходимо на терминальном сервере установить сервисы в автозагрузку, выполнив команду:

```
# systemctl enable xinetd dhcpd nfs-server
```

Затем необходимо выполнить настройку iptables, выполнив команды:

```
# iptables -F
# service iptables save
```

Далее требуется осуществить настройку сервера tftp изменив конфигурационный файл, выполнив команду:

```
# vim /etc/xinetd.d/tftp
```

Содержимое конфигурационного файла должно выглядеть следующим образом:

```
service tftp
{
       socket_type             = dgram
       protocol                = udp
       wait                    = yes
       user                    = root
       server                  = /usr/bin/tftp
       server_args             = -s /tftpboot
       disable                 = no
       per_source              = 11
       cps                     = 100 2
}
```

После изменения конфигурационного файла необходимо перезагрузить сервис xinetd, выполнив команду:

```
# systemctl restart xinetd
```

Для настройки сервера dhcpd необходимо изменить конфигурационный файл, выполнив команду:

```
# vim /etc/dhcp/dhcpd.conf
```

Содержимое конфигурационного файла должно выглядеть следующим образом:

Примечание: в конфигурационном файле необходимо указать реальные значения IP-адресов и MAC-адресов.

```
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#
ddns-update-style interim;
ignore client-updates;
allow booting;
allow bootp;

subnet 10.128.150.0 netmask 255.255.255.128 {
        option routers 10.128.150.126;
        option subnet-mask 255.255.255.128;
        option time-offset 18000;
        default-lease-time 21600;
        max-lease-time 43200;
}

host client_1 {
        option domain-name-servers 10.128.150.2;
        option domain-name "stand.ru";
        next-server 10.128.150.33;
        fixed-address  10.128.150.34;
        hardware ethernet  00:00:00:00:00:03;
        option host-name "tk.stand.ru";
        filename "pxelinux.0";
}
```

После изменения конфигурационного файла необходимо перезагрузить сервис dhcpd, выполнив команду:

```
# systemctl restart dhcpd
```

3.8 Установка и настройка тонкого клиента

Затем необходимо выполнить на терминальном сервере настройку службы nfs, выполнив команду:

```
# systemctl enable nfs-server
# systemctl restart nfs-server
```

Далее необходимо изменить конфигурационный файл exports, выполнив команду:

```
# vim /etc/exports
```

Содержимое конфигурационного файла должно выглядеть следующим образом:

```
/exported/root/directory *(rw,async,no_root_squash)
```

Для установки и настройки тонкого клиента необходимо на терминальный сервер установить пакет tka, выполнив команду:

```
# yum install tka-0.1-25.el6.sz.noarch.rpm
```

Далее осуществления настройки образа ТК необходимо последовательно выполнить следующие команды:

```
# cd /opt/ts/
# lzop -dc directory.tar.lzo | tar xvf -
# mkdir /exported/root -p
# mv directory /exported/root/
# export PATH=$PATH:/sbin
# export PATH=$PATH:/bin
# chroot /exported/root/directory
```

Для настройки репозитория необходимо изменить конфигурационный файл на терминальном сервере в образе терминального клиента, выполнив команду:

```
# vim etc/yum.repos.d/tk.repo
```

Содержимое конфигурационного файла должно выглядеть следующим образом:

```
[tk]
name=tk
baseurl=http://10.10.0.30/stor/out/64/PSZI_new/ABTK_6/isolinux/
enabled=1
gpgcheck=0
```

Далее необходимо установить системные утилиты из дистрибутива pszi-abts-1.0-x86_64.iso, выполнив следующие команды:

```
# rpm -e rhevm-sdk rhevm-guest-agent
# yum install @tk-utils -y
```

Затем для обеспечения возможности подключения по ssh необходимо добавить строку «auth sufficient pam_unix.so» в конфигурационном файле sshd, выполнив команду:

```
# vim /exported/root/directory/etc/pam.d/sshd
```

Далее для настройки службы времени тонкого клиента необходимо указать IP-адрес сервера времени в конфигурационном файле ntp и поставить службу в автозагрузку, выполнив команды:

```
# vim /etc/ntp.conf
# chkconfig ntpd on
# exit
```

Затем необходимо выполнить на терминальном сервере копирование загрузчика и ядра тонкого клиента на сервер tftp, последовательно выполнив команды:

```
# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
# cd /var/lib/tftpboot/
# tar -xvzf /exported/root/directory/root/root/kernel.tar.gz
# mkdir /var/lib/tftpboot/pxelinux.cfg
# chmod -R 755 /var/lib/tftpboot/
# restorecon -rv /var/lib/tftpboot/
```

4. ПРОВЕРКА ПРОГРАММЫ

Для проверки программы необходимо включить тонкий клиент, войти под доменным пользователем – должен открыться пользовательский портал, зайти в пользовательский портал – должна открыться консоль виртуальной машины в полноэкранном режиме.

При сворачивании консоли и окна браузера, никакие другие функции не доступны. Переход в виртуальные консоли невозможен.

Проверка считается успешной, а программа работоспособной, если при выполнении указанных действий не произошло ошибок и реакция изделия на выполнения проверки соответствует представленному описанию.

5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

В процессе установки и настройки изделия системный программист может получать системные сообщения. Сообщения могут быть информационные, предупреждающие или сообщения об ошибках. Возможные сообщения и необходимые действия системного программиста при их появлении описаны в разделе 3 настоящего документа.

6. ОГРАНИЧЕНИЯ ПРИ ЭКСПЛУАТАЦИИ

При эксплуатации изделия необходимо соблюдать следующие условия:

1. Запрещается устанавливать ЗОС «СинтезМ» на АРМ, запрещается использование для функционирования терминальных клиентов и установка на виртуальные машины, к которым пользователь имеет доступ с помощью средств удаленного управления (SPICE).
2. Непривилегированному пользователю запрещено предоставлять права доступа на редактирование файла /etc/mailcap.
3. Непривилегированному пользователю запрещено предоставлять права доступа на редактирование файлов в директории /dev.
4. Запрещается устанавливать программное обеспечение, позволяющее пользователю АСЗИ выполнять произвольный код.
5. Запрещается подключение неучтенных устройств (в том числе манипуляторов типа «мышь» и так далее).
6. Непривилегированному пользователю не должны предоставляться привилегии CAP_NET_ADMIN, CAP_SYS_RAWIO.
7. Запрещено устанавливать параметр max_message_unix_fds пакета dbus-1.6.12 равным нечетному числу.
8. Должна быть обеспечена невозможность доступа непривилегированного пользователя к серверным компонентам ЗОС «СинтезМ».
9. Запрещается предоставлять права на исполнение утилит из пакета hivex-1.3.10-5.7. Выполнение данных утилит запрещено.
10. Запрещается запускать программное обеспечение под учётной записью root.
11. Запрещается запускать программное обеспечение (Firebird, JRE, JBoss) под учётными записями, имеющими:

    а) доступ к следующим системным файлам:

    – `system.map;`

– kallsyms;

– sys_call_table;

– postgresql.conf и pg_hba.conf в директории /var/lib/pgsql/data;

– файлы в директориях var/log/sintezm и var/lib/psql/data/pg_log и var/lib/psql/data/pg_log;

– ld.so.conf;

– ld.so.cache;

– файлы в директории ld.so.conf.d.

б) доступ к файловым системам:

– selinuxfs;

– procfs;

– sysfs;

12. Запрещается назначать файлам программного обеспечения (Firebird, JRE, JBoss) следующие привилегии:

1) CAP_SYS_PTRACE;

2) CAP_KILL;

3) CAP_FOWNER.

ПРИЛОЖЕНИЕ А. ТАБЛИЦА НАВИГАЦИИ
(обязательное)

Интерфейс изделия работает в текстовом и графическом режимах.

Работа в текстовом режиме должна проводиться с учетом следующих основных принципов:

1) Переход между диалогами выполняется с помощью клавиш. Для перемещения курсора необходимо использовать клавиши со стрелками «**Влево»**, «**Вправо»**, «**Вверх»** и «**Вниз»**. Также используются клавиша «**Tab»** и комбинация клавиш «**Alt+Tab»** для перехода от одного элемента экрана к другому. Внизу экрана обычно показаны сочетания клавиш для перемещения курсора.

2) Для того, чтобы «нажать» кнопку, необходимо поместить курсор в ее пределах (например, с помощью клавиши **Tab**) и нажать клавишу «**Пробел»** или клавишу «**Enter»**. Чтобы выбрать элемент из списка, необходимо установить курсор на этот элемент и нажать клавишу «**Enter»**. Чтобы отметить элемент с помощью флажка, необходимо поместить курсор на флажок и нажать клавишу «**Пробел»**. Для отмены выбора нажмите клавишу «**Пробел»** ещё раз.

В графическом режиме навигация осуществляется с помощью компьютерной мыши. Также переход между элементами интерфейса возможно осуществлять и с помощью клавиатуры. Так, клавиша «**Tab»** позволяет перемещаться между полями ввода, клавиши «**Вверх»** и «**Вниз»** осуществляют прокрутку списков, «**+»** и «–» разворачивают и сворачивают списки, клавиши «**Пробел»** и «**Enter»** выбирают и отменяют выбор элемента.

ПРИЛОЖЕНИЕ Б. КРАТКАЯ ИНСТРУКЦИЯ ПО ИСПОЛЬЗОВАНИЮ РЕДАКТОРА VIM

(справочное)

Vim – это свободный режимный текстовый редактор, созданный на основе более старого редактора «vi». В настоящее время это один из мощнейших текстовых редакторов с полной свободой настройкой и автоматизацией в Linux-подобных операционных системах.

При работе с редактором используются следующие основные команды:

– открыть файл текстовым редактором

```
#vim /путь/имя файла;
```

– войти в режим редактирования – нажатие клавиши [i] (в данном режиме возможен ввод текста);

– выйти из режима редактирования – нажатие клавиши [esc] (в данном режиме выполняются команды копирования и удаления строк, сохранения файла или выход из него без сохранения);

– выйти из текстового редактора vim с сохранением – команда `:wq!` (выполняются данные команды не в режиме редактирования файла);

– выйти из текстового редактора vim без сохранения – команда `:q!`;

– удалить строку – двойное нажатие клавиши [d];

– скопировать строку – двойное нажатие клавиши [y];

– вставить строку – нажатие клавиши [p];

– поиск по файлу – нажатие клавиши [/] «запрос поиска».

ПРИЛОЖЕНИЕ В. ЛИСТИНГ СКРИПТА CLEAN.SH

(обязательное)

```
#!/bin/bash

echo "stopping services"
service vdsmd stop 2>/dev/null
service supervdsmd stop 2>/dev/null
initctl stop libvirtd 2>/dev/null

echo "removing packages"
yum remove \*ovirt\* \*vdsm\* \*libvirt\* -y

rm -fR /etc/*ovirt* /etc/*vdsm* /etc/*libvirt* /etc/pki/vdsm

FILES=" /etc/init/libvirtd.conf"
FILES+=" /etc/libvirt/nwfilter/vdsm-no-mac-spoofing.xml"
FILES+=" /etc/ovirt-hosted-engine/answers.conf"
FILES+=" etc/vdsm/vdsm.conf"
FILES+=" etc/pki/vdsm/*/*.pem"
FILES+=" etc/pki/CA/cacert.pem"
FILES+=" etc/pki/libvirt/*.pem"
FILES+=" etc/pki/libvirt/private/*.pem"
for f in $FILES
do
   [ ! -e $f ] && echo "? $f already missing" && continue
   echo "- removing $f"
   rm -f $f && continue
   echo "! error removing $f"
   exit 1
done

DIRS="/etc/ovirt-hosted-engine  /var/lib/libvirt/  /var/lib/vdsm/
/var/lib/ovirt-hosted-engine-*    /var/log/ovirt-hosted-engine-setup/
/var/cache/libvirt/"
for d in $DIRS
do
   [ ! -d $f ] && echo "? $d already missing" && continue
   echo "- removing $d"
   rm -fR $d && continue
   echo "! error removing $d"
   exit 1
done
```

ПРИЛОЖЕНИЕ Г. ПЕРЕЧЕНЬ ТЕРМИНОВ

(справочное)

Автоматизированная система – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций

Автоматизированная система в защищенном исполнении – автоматизированная система, предназначенная для обработки информации ограниченного доступа и реализующая информационную технологию выполнения установленных функций в соответствии с требованиями нормативных документов по защите информации ФСБ России *(далее АСЗИ или Система)*

Виртуальный сетевой мост – построенная на базе технологии Linux Bridge программная реализация коммутатора, предназначенного для соединения нескольких узлов (ВМ, физических серверов, АРМ и технических средств) сети и обеспечивающего «жесткое» подключение физических и виртуальных интерфейсов сетевых адаптеров в вертикальную сетевую инфраструктуру.

Защищаемая информация – информация, содержащая охраняемые сведения, а также информация, для которой ее владельцем установлены требуемые характеристики безопасности.

Дистрибутив ПО – программное обеспечение, представленное в унифицированной форме, которая дает возможность выполнить установку ПО в автоматизированной системе в защищенном исполнении.

Исполняемый файл – файл, содержимое которого является готовой к исполнению компьютерной программой (в том числе двоичное представление машинных инструкций, скрипты bash, файлы сценариев python и т.д.).

Канал репозитория – совокупность ресурсов репозитория (ПО, iso-образы), доступных агенту для установки на ВМ и физические сервера.

Комплекс средств автоматизации – совокупность всех компонентов автоматизированной системы за исключением персонала.

Метаданные дистрибутива ПО – формализованное описание общих сведений и специальных характеристик программного обеспечения, представленное в форматах XML и JSON.

Метка безопасности – набор атрибутов, однозначно сопоставляемых с субъектами доступа, объектами доступа и назначающих их характеристики.

Монтирование – процедура подключения файловой системы, находящейся на СФХ в иерархию файлов и каталогов файловой системы ВМ.

Пользователь – конкретное должностное лицо, обладающее определенными правами доступа к объектам доступа, для выполнения функциональных обязанностей которого в автоматизированном режиме в АСЗИ создаётся виртуальная машина или выделяется рабочая станция.

Пользовательские виртуальные машины – виртуальные машины, предназначенные для исполнения программных средств Системы, с которыми непосредственно взаимодействуют ее пользователи.

Сетевое файловое хранилище (СФХ) – специализированное хранилище файлов и каталогов, подключенное к локальной сети.

Терминальный клиент – бездисковый АРМ обеспечивающий, взаимодействие пользователя с ресурсами АСЗИ посредством протокола SPICE.

Терминальный сервер – инфраструктурная ВМ обеспечивающая, выдачу терминальным клиентам ip-адресов по протоколу DHCP, а также загрузку образа ОС на терминальные клиенты, с использованием PXE, по локальной сети.

Узел – физические серверы, пользовательские ВМ, ВМ СПО, инфраструктурные ВМ, ТК или рабочие станции, зарегистрированные на сервере управления доступом.

Конфигурируемое сетевое взаимодействие – сетевое взаимодействие, предполагающее реализацию сетевого обмена между приложениями, функционирующими в ВМ и/или на физических серверах КСА по назначаемым администратором безопасности сетевым адресам, портам и протоколам транспортного уровня при гарантированном отсутствии взаимодействия вне установленных администратором безопасности адресов, портов и протоколов.

Шаблон виртуальной машины – формальное описание требуемых виртуальной машине ресурсов гипервизора, ее программной среды, состава программного обеспечения, задающее возможность создания в автоматизированном режиме на базе данного шаблона одной или нескольких единообразных по содержимому программного обеспечения и специального программного обеспечения виртуальных машин (пользовательских, инфраструктурных, специального программного обеспечения, баз данных, сетевых файловых хранилищ).

Шаблон программной среды – формальное описание перечня программного обеспечения, задающее возможность развертывания в автоматизированном режиме на базе данного шаблона одного или нескольких единообразных по содержимому программного обеспечения физических серверов и/или рабочих станций из состава АСЗИ.

Web-клиент **–** пользовательское приложение, посредством которого осуществляется взаимодействие пользователя с web-сервисами.

ПРИЛОЖЕНИЕ Д. ОПИСАНИЕ ФУНКЦИЙ ДЛЯ БИБЛИОТЕКИ LIBSELINUX

Название функции	Описание назначения
avc_add_callback	Функция регистрирует обработчик событий безопасности
avc_audit	Аудит предоставления или отказа разрешений в соответствии с политикой. Эта функция обычно вызывается avc_has_perm (3) после проверки разрешений, но также может быть вызвана непосредственно вызывающими методами, которые используют avc_has_perm_noaudit (3), для отделения проверки разрешений от аудита. Например, это разделение полезно, когда проверка разрешений должна выполняться под блокировкой, чтобы позволить блокировке высвободиться прежде вызова аудита кода.
avc_av_stats	Логирует статистику таблицы журналов. Регистрирует сообщение с информацией о размере и распределении таблицы векторов доступа. Для печати сообщения используется обратный вызов аудита.
avc_cache_stats	Получает кеш статистики доступа. Заполняет прилагаемую структуру информацией о деятельности AVC с момента последнего вызова avc_init (3) или avc_reset (3).
avc_cleanup	Удаляет неиспользуемые записи в кэше AVC. Ищет в таблице SID структуры SID с нулевым количеством ссылок и удаляет их вместе со всеми элементами AVC, которые ссылаются на них. Это можно использовать для высвобождения памяти в системе.
avc_compute_create	Вычисляет SID для маркировки нового объекта. Вызывает сервер безопасности, чтобы получить контекст для маркировки нового объекта.
avc_compute_member	Вычисляет SID для полиинстантации. Вызывает сервер безопасности, чтобы получить контекст для маркировки экземпляра объекта.
avc_context_to_sid	Получает SID для контекста. Просмотрите контекст безопасности ctx в таблице SID, создав новую запись, если ctx не найден. Хранит указатель на структуру SID в памяти, на которую ссылается sid, возвращая 0 при успехе или -1 при ошибке с установкой флага errno.
avc_context_to_sid_raw	Получает SID для контекста.
avc_destroy	Освобождает все структуры AVC. Уничтожает все структуры AVC и освобождает всю выделенную память. Удерживаемые пользователем блокировки, память и обратные вызовы аудита будут сохранены, но обратные вызовы событий безопасности не будут. Все SID будут аннулированы. Пользователь должен вызвать avc_init (3), если требуется дальнейшее использование AVC.
avc_get_initial_sid	Получает SID для исходного идентификатора безопасности ядра. Получает контекст для исходного идентификатора безопасности ядра, указанного по имени, с помощью

Название функции	Описание назначения
	security_get_initial_context (3), а затем вызывает avc_context_to_sid (3), чтобы получить соответствующий идентификатор безопасности.
avc_has_perm	Проверяет разрешения и выполняет любой соответствующий аудит. Осуществляет запрос AVC, чтобы определить, предоставлены ли запрашиваемые разрешения для SID (ssid, tsid), интерпретирует разрешения на основе tclass и вызове сервера безопасности в пропуске кеша, чтобы получить новое решение и добавить его в кеш. Обновляет aeref, чтобы ссылаться на запись AVC с принятыми решениями. Аудит предоставления или отказа в разрешениях в соответствии с политикой. Возвращает 0, если все запрошенные разрешения предоставлены, -1 с errno, установленным в EACCES, если какие-либо разрешения отклоняются или к другому значению при других ошибках.
avc_has_perm_noaudit	Проверяет разрешения, но не выполняет аудит. Проверяет AVC, чтобы определить, предоставлены ли запрашиваемые разрешения для пары SID (ssid, tsid), интерпретации разрешений на основе tclass и вызове сервера безопасности в пропуске кеша, для получения нового решения и добавления его в кеш. Обновляет aeref, чтобы ссылаться на запись AVC с принятыми решениями и возвращает копию решений в avd. Возвращает 0, если все запрашиваемые разрешения предоставляются, -1 с errno, установленным в EACCES, если разрешены какие-либо разрешения или другое значение при возникновении других ошибок. Эта функция обычно вызывается avc_has_perm (3), но также может быть вызвана непосредственно (самостоятельно), чтобы отделить проверку разрешений от аудита, например. в случаях, когда блокировка должна удерживаться для проверки, но должна быть осуществлена для аудита.
avc_init	Использует avc_open. Инициализирует AVC. Инициализация кеша вектора доступа. Возврат 0 при успехе или -1 с флагом errno, установленным при ошибке.
avc_netlink_acquire_fd	Создает сокет netlink и подключается к ядру.
avc_netlink_check_nb	Ожидает сообщения netlink из ядра.
avc_netlink_close	Закрывает сокет netlink.
avc_netlink_loop	Устанавливает netlink сокет fd. Позволяет приложению управлять сообщениями из сокета netlink в своем собственном основном цикле.
avc_netlink_open	Освобождает netlink сокет fd. Возвращает право собственности на сокет netlink к библиотеке.
avc_netlink_release_fd	Проверяет netlink сокет для новых сообщений.
avc_open	Инициализирует AVC. Эта функция идентична avc_init (3), за исключением того, что префикс сообщения установлен на «avc», и любые требуемые обратные вызовы должны быть указаны через selinux_set_callback (3).

Название функции	Описание назначения
avc_reset	Очищает кеш и сбрасывает статистику. Удаляет все записи из кэша и сбрасывает все статистические данные доступа (возвращаемые avc_cache_stats (3)) на ноль. Не влияет на отображение SID . Возвращает 0 при успехе, -1 с флагом errno, установленным при ошибке.
avc_sid_stats	Статистика таблицы SID журнала. Записывает сообщение с информацией о размере и распределении таблицы SID. Для вывода сообщения используется обратный вызов аудита.
avc_sid_to_context	Получает копию контекста, соответствующего SID. Возвращает копию контекста безопасности, соответствующего входному sid в памяти, на которое ссылается ctx.
avc_sid_to_context_raw	Получает копию контекста, соответствующего SID. Возвращает копию контекста безопасности, соответствующего входному sid в памяти, на которое ссылается ctx.
checkPasswdAccess	Используйте selinux_check_passwd_access (3) или предпочтительно selinux_check_access (3). Проверьте разрешение в классе passwd. Возвращает 0, если предоставлено или -1 в противном случае. Данная функция является "заглушкой".
context_free	Освобождает хранилище, используемое контекстом.
context_new	Возвращает новый контекст, инициализированный строкой контекста.
context_range_get	Получает указатель на диапазон.
context_range_set	Задает компонент диапазона. Возвращает ненулевое значение, если оно не выполнено.
context_role_get	Получает указатель на роль.
context_role_set	Установает компонент роли. Возвращает ненулевое значение, если оно не выполнено.
context_str	Возвращает указатель на строковое значение context_t. Действителен до следующего вызова context_str или context_free для одного и того же context_t*.
context_type_get	Получает указатель на тип.
context_type_set	Устанавливает тип компонента. Возвращает ненулевое значение, если оно не выполнено.
context_user_get	Получает указатель на пользователя.
context_user_set	Устанавливает пользовательский компонент. Возвращает ненулевое значение, если оно не выполнено.
fgetfilecon	Wrapper для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освобождаться через freecon.
fgetfilecon_raw	Обертка для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освобождаться через freecon.
fini_selinuxmnt	Очищает переменную selinuxmnt и освобождает выделенную память.

Название функции	Описание назначения
freecon	Освобождает память, выделенную для контекста любым вызовом get *.
freeconary	Освобождает память, выделенную для массива контекста, с помощью security_compute_user (3).
fsetfilecon	Обертка для API xattr - Устанавливает контекст файла.
fsetfilecon_raw	Обертка для API xattr - Устанавливает контекст файла.
getcon	Получает текущий контекст и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освободить через freecon (3).
getcon_raw	Получает текущий контекст и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освободить через freecon (3).
get_default_context	Получает контекст безопасности по умолчанию для сеанса пользователя для «пользователя», порожденного «fromcon», и устанавливает для него *newcon, чтобы ссылаться на него. Контекст будет одним из полномочий, разрешенных политикой, но выбор значения по умолчанию зависит от пользовательских настроек. Если ’fromcon’ равно NULL, по умолчанию используется текущий контекст. Возвращает 0 при успехе или -1 в противном случае. Вызывающий метод должен освобождаться через freecon.
get_default_context_with_level	То же, что и get_default_context (3), но используется предоставленный уровень MLS, а не уровень по умолчанию для пользователя.
get_default_context_with_role	То же, что и get_default_context (3), но возвращается только контекст с указанной ролью.
get_default_context_with_rolelevel	То же, что и get_default_context (3), но возвращается только контекст с указанной ролью и уровнем.
get_default_type	Получает тип по умолчанию (домен) для «роли» и устанавливает для него «тип». Вызывающий метод должен освобождаться через free(3). Возвращает 0 при успехе или -1 в противном случае.
getexeccon	Получает контекст exec и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст exec, т. е. используется значение по умолчанию.
getexeccon_raw	Получает контекст exec и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст exec, т. е. используется значение по умолчанию.
getfilecon	Wrapper для xattr API. Получает контекст файла и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getfilecon_raw	Wrapper для xattr API. Получает контекст файла и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getfscreatecon	Получает контекст fscreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если fs не создавал контекст, т. е. используется по умолчанию.

Название функции	Описание назначения
getfscreatecon_raw	Получает контекст fscreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если fs не создавал контекст, т. е. используется по умолчанию.
getkeycreatecon	Получает контекст keycreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст создания ключа, т. е. используется значение по умолчанию.
getkeycreatecon_raw	Получает контекст keycreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст создания ключа, т. е. используется значение по умолчанию.
get_ordered_context_list	Получает упорядоченный список разрешенных контекстов безопасности для сеанса пользователя и устанавливает *conary для ссылки на набор контекстов с завершающим действием NULL. Каждая запись в списке будет разрешена политикой, но запись зависит от пользовательских настроек. Возвращает количество записей в *conary. Если ’fromcon’ равно NULL, по умолчанию используется текущий контекст. Вызывающий метод должен освобождаться через freeconary(3).
get_ordered_context_list_with_level	То же, что и get_ordered_context_list (3), но используется предоставленный уровень MLS, а не уровень по умолчанию для пользователя.
getpeercon	Wrapper для сокета API. Получает контекст одноранговой сети и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освобождаться через freecon (3).
getpeercon_raw	Wrapper для сокета API. Получает контекст одноранговой сети и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освобождаться через freecon (3).
getpidcon	Получает контекст процесса, идентифицированный pid, и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getpidcon_raw	Получает контекст процесса, идентифицированный pid, и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getprevcon	Получает предыдущий контекст (до последнего exec) и устанавливает *con для ссылки на него. Звонящий должен освободить через freecon (3).
getprevcon_raw	Получает предыдущий контекст (до последнего exec) и устанавливает *con для ссылки на него. Звонящий должен освободить через freecon (3).
getseuser	Получает имя пользователя и уровень SELinux для использования Linux данного имени и службы. Эти значения затем могут быть переданы в функции get_ordered_context_list* и get_default_context* для получения контекста пользователя. Возвращает 0 при

Название функции	Описание назначения
	успехе или -1 в противном случае. Вызывающий метод должен освободить возвращенные строки через free(3).
getseuserbyname	Получает имя пользователя и уровень SELinux для использования Linux данного имени пользователя . Эти значения затем могут быть переданы в функции get_ordered_context_list* и get_default_context* для получения контекста пользователя. Возвращает 0 при успехе или -1 в противном случае. Вызывающий метод должен освободить возвращенные строки через free(3).
getsockcreatecon	Получает контекст sockcreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст сокета, т. е. используется по умолчанию. Если не NULL, Вызывающий метод должен освободиться через freecon(3).
getsockcreatecon_raw	Получает контекст sockcreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст сокета, т. е. используется по умолчанию. Если не NULL, Вызывающий метод должен освободиться через freecon(3).
_init (если _initselinuxmnt - то вот)	Для этого есть справочная страница, однако она не является доступной пользователю функцией (только для внутреннего использования - хотя доступ к fini_selinuxmnt доступен).
is_context_customizable	Возвращает, настраивается ли контекст файла, и его нельзя перемаркировать.
is_selinux_enabled	Возвращает 1, если выполняется в ядре SELinux, или 0, если нет, или -1 в случае ошибки.
is_selinux_mls_enabled	Возвращает 1, если мы работаем на ядре SELinux MLS, или 0 в противном случае.
lgetfilecon	Wrapper для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освободиться через freecon (3).
lgetfilecon_raw	Wrapper для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освободиться через freecon (3).
lsetfilecon	Wrapper для xattr API. Устанавливает контекст файла для символической ссылки.
lsetfilecon_raw	Wrapper для xattr API. Устанавливает контекст файла для символической ссылки.
manual_user_enter_context	Разрешает пользователю вручную вводить контекст в качестве резерва, если список разрешенных контекстов не может быть получен. Вызывающий метод должен освободиться через freecon (3). Возвращает 0 при успехе или -1 в противном случае.
matchmediacon	Совмещает указанный носитель и конфигурацию контекста мультимедиа и устанавливает *con для ссылки на итоговый контекст. Вызывающий метод должен освобождаться через freecon.

Название функции	Описание назначения
matchpathcon	Сопоставляет указанный путь и режим с настройкой контекста файла и устанавливает *con для ссылки на результирующий контекст. «mode» может быть 0, чтобы отключить соответствие режима. Вызывающий метод должен освобождаться через freecon. Если matchpathcon_init (3) еще не был вызван, эта функция вызовет его при первом вызове с NULL-путем.
matchpathcon_checkmatches	Проверяет, не соответствуют ли какие-либо спецификации и сообщениям о них. «Str» используется как префикс для любых предупреждающих сообщений.
matchpathcon_filespec_add	Поддерживает связь между индексом и индексом спецификации и проверяет, связана ли конфликтующая спецификация с тем же inode (например, из-за нескольких жестких ссылок). Если это так, используйте последнюю из двух спецификаций на основе их порядка в конфигурации контекста файла. Возвращает использованный индекс спецификации.
matchpathcon_filespec_destroy	Уничтожает все ассоциации inode, которые были добавлены, например, для перезагрузки новой файловой системы.
matchpathcon_filespec_eval	Отображает статистику использования хэш-таблицы для ассоциаций.
matchpathcon_fini	Освобождает память, выделенную matchpathcon_init.
matchpathcon_index	То же, что и matchpathcon(3), но возвращает индекс спецификации для последующего использования в вызове matchpathcon_filespec_add(3).
matchpathcon_init	Загружает конфигурацию контекста файла, указанную в «path» в памяти, для использования последующими вызовами matchpathcon. Если «path» равен NULL, загружает конфигурацию контекста активного файла, то есть путь, возвращаемый selinux_file_context_path (3). Если флаг MATCHPATHCON_BASEONLY не установлен, эта функция также проверяет файл «path».homedirs и файл «path».local и загружает дополнительные спецификации из них, если они есть.
matchpathcon_init_prefix	То же, что и matchpathcon init (3), но загружает только записи с регулярным выражением, которые имеют ветки которые являются префиксами «prefix».
mode_to_security_class	Переводит mode_t в имя строки класса безопасности (например, S_ISREG = "file").
print_access_vector	Отображает вектор доступа в строчном представлении.
query_user_context	Учитывая список разрешенных контекстов безопасности для пользователя, запрашивает выбор пользователя и устанавливает для него *newcon. Вызывающий метод должен освободиться через freecon (3). Возвращает 0 в случае sucess или -1 в обратном случае.
realpath_not_final	Резрешает все символические ссылки и относительные части имени пути, но НЕ последний компонент (тот же самый realpath (3), если конечный компонент не является

Название функции	Описание назначения
	символической ссылкой. Итоговым путем должен быть путь размером PATH_MAX + 1.
rpm_execcon	Выполняет вспомогательную операцию для rpm в соответствующем контексте безопасности.
security_av_perm_to_string	Преобразовывает разрешения на доступ к именам строк.
security_av_string	Возвращает вектор доступа в строчном представлении. Пользователь должен освободить возвращаемую строку через free(3).
security_canonicalize_context	Канонизирует контекст безопасности. Возвращает указатель на каноническую (основную) форму контекста безопасности в canoncon, который использует ядро, а не то, что предоставляется приложением приложения userpace в con.
security_canonicalize_context_raw	Канонизирует (сохраняет) контекст безопасности. Возвращает указатель на каноническую (основную) форму контекста безопасности в canoncon, который использует ядро, а не то, что предоставляется пользовательским приложением в con.
security_check_context	Проверяет валидность контекста безопасности.
security_check_context_raw	Проверяет валидность контекста безопасности.
security_class_to_string	Преобразует значения класса безопасности в имена строк.
security_commit_booleans	Применяет значения логических параметров
security_compute_av	Вычисляет решение о доступе. Запрашивает, разрешает ли политика доступов исходный контекстный scon для доступа к целевому контексту tcon через класс tclass с запрошенным вектором доступа. Решение возвращается в avd.
security_compute_av_flags	Вычисляет решение доступа и возвращает флаги. Запрашивает, разрешает ли политика доступы исходного контекста scon к целевому контексту tcon через класс tclass с вектором запрошенного доступа. Решение возвращается в avd. который имеет дополнительный флаг. В настоящее время единственным определяемым флагом является SELINUX_AVD_FLAGS_PERMISSIVE, который указывает, что решение было вычислено в разрешающем домене (то есть, язык разрешающей политики был использован в политике или семанаге(8), который использовался для установки домена в разрешающем режиме). Обратите внимание, что это не означает, что SELinux работает в разрешающем режиме, а только в домене scon.
security_compute_av_flags_raw	Вычисляет решение доступа и возвращает флаги. Запрашивает, разрешает ли политика доступы исходного контекста scon к целевому контексту tcon через класс tclass с вектором запрошенного доступа. Решение возвращается в avd. который имеет дополнительный флаг. В настоящее время единственным определяемым флагом является SELINUX_AVD_FLAGS_PERMISSIVE, который указывает, что решение было вычислено в разрешающем домене (то есть, язык разрешающей политики был использован в

Название функции	Описание назначения
	политике или семанаге(8), который использовался для установки домена в разрешающем режиме). Обратите внимание, что это не означает, что SELinux работает в разрешающем режиме, только в домене scon.
security_compute_av_raw	Вычисляет решение о доступе. Запрашивает, разрешает ли политика доступов исходный контекстный scon для доступа к целевому контексту tcon через класс tclass с запрошенным вектором доступа. Решение возвращается в avd.
security_compute_create	Вычисляет решение для маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon(3).
security_compute_create_name	Функция идентична security_compute_create(3), но также принимает имя нового объекта при создании в качестве аргумента. Когда правило [#5.7.6.type_transition |outline type_transition] </tt> для данного класса, а пара scon/tcon имеет расширение имени объекта, newcon будет возвращаться в соответствии с политикой. Обратите внимание, что этот интерфейс поддерживается только на ядрах 2.6.40 или новее. Для старых ядер имя объекта игнорируется.
security_compute_create_name_raw	Функция идентична security_compute_create(3), но также принимает имя нового объекта при создании в качестве аргумента. Когда правило [#5.7.6.type_transition |outline type_transition] </tt> для данного класса, а пара scon/tcon имеет расширение имени объекта, newcon будет возвращаться в соответствии с политикой. Обратите внимание, что этот интерфейс поддерживается только на ядрах 2.6.40 или новее. Для старых ядер имя объекта игнорируется.
security_compute_create_raw	Вычисляет решение для маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon(3).
security_compute_member	Вычисляет решение для члена polyinstantiation и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
security_compute_member_raw	Вычисляет решение для члена polyinstantiation и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
security_compute_relabel	Вычисляет решение для повторной маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
security_compute_relabel_raw	Вычисляет решение для повторной маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
security_compute_user	Вычисляет набор доступных контекстов пользователя и устанавливает *con для ссылки на массив контекстов с завершающим действием NULL. Вызывающий метод

Название функции	Описание назначения
	должен освобождаться через freeconary(3).
security_compute_user_raw	Вычисляет набор доступных контекстов пользователя и устанавливает *con для ссылки на массив контекстов с завершающим действием NULL. Вызывающий метод должен освобождаться через freeconary(3).
security_deny_unknown	Получает поведение для неопределенных классов / разрешений.
security_disable	Отключает SELinux во время выполнения (необходимо выполнить до начальной загрузки политики).
security_get_boolean_active	Получает активное значение для логического.
security_get_boolean_names	Получает логические имена
security_get_boolean_pending	Получает ожидающее значение для логического.
security_getenforce	Получает значение флага принудительного исполнения.
security_get_initial_context	Получает контекст исходного идентификатора безопасности ядра по имени. Вызывающий метод должен освободить через freecon (3).
security_get_initial_context_raw	Получает контекст исходного идентификатора безопасности ядра по имени. Вызывающий метод должен освободить через freecon (3).
security_load_booleans	Загрузите логические настройки политики. Путь может быть NULL, и в этом случае логические значения загружаются из активного файла конфигурации логической политики.
security_load_policy	Загружает конфигурацию политики.
security_policyvers	Получает номер версии политики.
security_set_boolean	Устанавливает ожидающее значение для булева.
security_set_boolean_list	Сохраняет список логических элементов в одной транзакции.
security_setenforce	Устанавливает значение флага принудительного исполнения.
selabel_close	Уничтожает указанный дескриптор, закрывает файлы, освобождает выделенную память и т. д. Хендл после закрытия больше не может использоваться.
selabel_cmp	Сравнивает два дескриптора
selabel_digest	Получает хэш дескриптора
selabel_lookup	Выполняет операцию поиска меток. Возвращает 0 при успехе, -1 с errno, устанавливающемся при сбое. Аргументы key и type являются входными (данными) для операции поиска; соответствующие значения продиктованы используемым бэкэндом. Результат возвращается в точку памяти, на которую указывает con, и он должен быть освобожден с помощью freecon.
selabel_lookup_best_match	Выбирает одну наиболее подходящую условию поиска метку

Название функции	Описание назначения
selabel_lookup_best_match_raw	Выбирает одну наиболее подходящую условию поиска метку
selabel_lookup_raw	Выполняет операцию поиска меток. Возвращает 0 при успехе, -1 с errno, устанавливающемся при сбое. Аргументы key и type являются входными (данными) для операции поиска; соответствующие значения продиктованы используемым бэкэндом. Результат возвращается в точку памяти, на которую указывает con, и который должен быть освобожден с помощью freecon.
selabel_open	Создает дескриптор маркировки. Откройте бэкэнд для использования маркировки. Доступными бэкэнд идентификаторами являются: SELABEL_CTX_FILE - file_contexts. SELABEL_CTX_MEDIA - media contexts. SELABEL_CTX_X - x_contexts. SELABEL_CTX_DB - contexts SE-PostgreSQL. SELABEL_CTX_ANDROID_PROP - property_contexts. Параметры могут предоставляться с помощью параметра opts; доступны следующие варианты: SELABEL_OPT_UNUSED - опция no-op, полезная для неиспользуемых слотов в массиве опций. SELABEL_OPT_VALIDATE - проверять контексты перед их возвратом (логическое значение). SELABEL_OPT_BASEONLY - не используйте локальные настройки для создания бэкэнд-данных (логическое значение). SELABEL_OPT_PATH - указать альтернативный путь для использования при загрузке данных на бэкэнд. SELABEL_OPT_SUBSET - выберите подмножество поискового пространства в качестве оптимизации (файловый сервер). Не все опции могут поддерживаться каждым бэкэнд. Возвращаемое значение - это успешно созданный дескриптор или NULL с errno, установленным при ошибке. "
selabel_partial_match	Выбирает одну наиболее подходящую условию поиска метку
selabel_stats	Логирует сообщение с информацией о количестве выполненных запросов, количестве неиспользуемых совпадающих записей или другой рабочей статистике. Сообщение является специфичным для бэкэнда, некоторые серверы могут не выводить сообщение.
selabel_subs_init	Инициализрует работу с метками
selinux_binary_policy_path	Возвращает путь к двоичному файлу политики из корневого каталога политики.
selinux_booleans_path	Возвращает путь к логическому файлу в корневом каталоге политики.
selinux_booleans_subs_path	Возвращает путь к файлу конфигурации booleans.subs_dist.

Название функции	Описание назначения
selinux_boolean_sub	Считывает файл /etc/selinux/TYPE/booleans.subs_dist, ища запись с именем boolean_name. Если существует запись, selinux_boolean_sub (3) возвращает переведенное имя, иначе возвращается исходное имя. Возвращаемое значение должно быть освобождено. При ошибке будет возвращен NULL.
selinux_check_access	Используется для проверки того, имеет ли исходный контекст разрешение доступа для указанного класса в целевом контексте. Обратите внимание, что разрешение и класс являются ссылочными строками. Параметр aux может ссылаться на дополнительную информацию аудита. Аудит обрабатывается, как описано в avc_audit (3). См. Security_deny_unknown (3) о том, как флаг deny_unknown может влиять на решения политики.
selinux_check_passwd_access	Проверяет разрешение в классе passwd. Возвращает 0, если предоставлено или -1 в противном случае. Заменено selinux_check_access (3)
selinux_check_securetty_context	Проверяет, что tty_context определяется как securetty. Возвращает 0, если безопасно, <0 в противном случае.
selinux_colors_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_contexts_path	Возвращает путь к каталогу контекстов в корневом каталоге политики.
selinux_current_policy_path	Возвращает путь к текущей политике.
selinux_customizable_types_path	Возвращает путь к файлу customizable_types в корневом каталоге политики.
selinux_default_context_path	Возвращает к файлу default_context в корневом каталоге политики.
selinux_default_type_path	Возвращает путь к файлу default_type.
selinux_failsafe_context_path	Возвращает путь к файлу failafe_context в корневом каталоге политики.
selinux_file_context_cmp	Сравнивает два контекста файлов, возвращайте 0, если они эквивалентны.
selinux_file_context_homedir_path	Возвращает путь к файлу file_context.homedir в корневом каталоге политики.
selinux_file_context_local_path	Возвращает путь к файлу file_context.local в корневом каталоге политики.
selinux_file_context_path	Возвращает путь к файлу file_context в корневом каталоге политики.
selinux_file_context_subs_dist_path	Возвращает путь к файлу file_context.subs_dist в корневом каталоге политики.
selinux_file_context_subs_path	Возвращает путь к файлу file_context.subs в корневом каталоге политики.
selinux_file_context_verify	Проверяет контекст файла "path" к политике. Возвращает 0, если это правильно.
selinuxfs_exists	Проверка существоввания пседофайловой системы SELinux

Название функции	Описание назначения
selinux_get_callback	Используется для получения указателя на функцию обратного вызова данного типа. Функции обратного вызова устанавливаются с помощью selinux_set_callback (3).
selinux_getenforcemode	Читает файл /etc/selinux/config и определяет, следует ли запускать машину в режиме принудительного (1), разрешающего (0) или отключенного (-1) режима.
selinux_getpolicytype	Читает файл / etc / selinux / config и определяет, что такое политика по умолчанию для машины. Вызов приложения должен освобождаться policytype.
selinux_homedir_context_path	Возвращает путь к файлу в корневом каталоге политики. Обратите внимание, что этот файл будет отображаться только в более старых версиях политики в этом месте. В системах, управляемых с помощью semanage(8), это теперь находится в хранилище политик.
selinux_init_load_policy	Выполняет начальную загрузку политики. Эта функция определяет желаемый режим принудительного исполнения, устанавливает аргумент *enforce (принудительного) использования для Вызывающий метод, устанавливает соответствие статуса принудительного использования ядра SELinux и загружает политику. Он также внутренне обрабатывает начальное монтирование selinuxfs, необходимое для выполнения этих действий. Функция возвращает 0, если все, включая загрузку политики, завершается успешно. В этом случае init, как ожидается, будет повторно выполнять сам, чтобы перейти к надлежащему контексту безопасности. В противном случае функция возвращает -1, и init должен проверить *enforce, чтобы определить, как действовать. Если принудительное выполнение (*inforce> 0), то init должен остановить систему. В противном случае init может нормально работать без повторного запуска.
selinux_lsetfilecon_default	Эта функция устанавливает контекст файла в системные значения по умолчанию. Возвращает 0 при успехе.
selinux_lxc_contexts_path	Возвращает путь к файлу конфигурации lxc_contexts.
selinux_media_context_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_mkload_policy	Создает слепок политики и загружает его. Эта функция обеспечивает интерфейс более высокого уровня для загрузки политики, чем security_load_policy (3), внутренне определяет правильную версию политики, устанавливает и открывает файл политики, сопоставляет ее в памяти, манипулируя ею (памятью) по мере необходимости для текущих логических настроек и/или локальных определений и затем вызовет security_load_policy (3), чтобы загрузить ее. «preservebools» - это логический флаг, указывающий, следует ли сохранять текущие логические значения политики в новой политике (если 1) или сбросить сохраненные параметры политики (если 0). Первый случай является значением по умолчанию для перезагрузки

Название функции	Описание назначения
	политики, в то время как последний случай является вариантом для перезагрузки политики, но в первую очередь для начальной загрузки политики.
selinux_netfilter_context_path	Возвращает путь к файлу netfilter_context в корневом каталоге политики.
selinux_openssh_contexts_path	Возвращает путь к файлу openssh_context в корневом каталоге политики.
selinux_path	Возвращает путь к корневому каталогу политики.
selinux_policy_root	Считывает файл /etc/selinux/config и возвращает каталог верхнего уровня.
selinux_raw_context_to_color	Выполняет контекстный перевод между контекстами безопасности и цветами отображения. Возвращает список разделенных пробелами десяти шестнадцатеричных триггеров RGB с префиксом хэш-меток, например. "#ff0000". Вызывающий метод должен освободить полученную строку через free(3). Возвращает -1 при ошибке или 0 в противном случае.
selinux_raw_to_trans_context	Выполняет контекстный перевод между человеко-понятным форматом («translated») и внутренним системным форматом («raw»). Вызывающий метод должен освободить полученный контекст через freecon(3). Возвращает -1 при ошибке или 0 в противном случае. Если передано NULL, задается возвращаемый контекст в NULL и возвращает 0.
selinux_removable_context_path	Возвращает путь к файлу removeable_context в корневом каталоге политики.
selinux_reset_config	Принудительный сброс загруженной конфигурации.
selinux_restorecon	Восстанавливает соединение с псевдофайловой системой
selinux_restorecon_default_handle	Восстанавливает соединение с псевдофайловой системой
selinux_restorecon_set_exclude_list	Восстанавливает соединение с псевдофайловой системой
selinux_restorecon_set_sehandle	Восстанавливает соединение с псевдофайловой системой
selinux_securetty_types_path	Возвращает путь к файлу securetty_types в корневом каталоге политики.
selinux_sepgsql_context_path	Возвращает путь к файлу sepgsql_context в корневом каталоге политики.
selinux_set_callback	Устанавливает обратный вызов в соответствии с типом: SELINUX_CB_LOG, SELINUX_CB_AUDIT, SELINUX_CB_VALIDATE, SELINUX_CB_SETENFORCE, SELINUX_CB_POLICYLOAD
selinux_set_mapping	Поддержка сопоставления класса пространства пользователей, которая устанавливает сопоставление от

Название функции	Описание назначения
	пользовательского заказа классов объектов и разрешений на номера, фактически используемые загруженной системной политикой.
selinux_set_policy_root	Устанавливает альтернативный путь корневого каталога политики, в соответствии с которым существуют файлы скомпилированного файла политики и файлы конфигурации контекста.
selinux_snapperd_contexts_path	Устанавливает путь до базовых контекстов
selinux_status_close	Отменяет и закрывает страницу состояния ядра.
selinux_status_deny_unknown	Получает поведение для неопределенных классов/разрешений.
selinux_status_getenforce	Получает значение флага принудительного исполнения.
selinux_status_open	Открывает и отображает страницу состояния ядра SELinux.
selinux_status_policyload	Получает количество загрузок политики.
selinux_status_updated	Сообщает, обновлен ли статус ядра.
selinux_systemd_contexts_path	Возвращает путь к файлу конфигурации systemd_contexts.
selinux_translations_path	Возвращает путь к файлу setrans.conf в корневом каталоге политики.
selinux_trans_to_raw_context	Выполняет контекстный перевод между человеко-читаемым форматом («translated») и внутренним системным форматом («raw»). Вызывающий метод должен освободить полученный контекст через freecon(3). Возвращает -1 при ошибке или 0 в противном случае. Если передано NULL, задается возвращаемый контекст в NULL и возвращает 0.
selinux_user_contexts_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_usersconf_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_users_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_virtual_domain_context_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_virtual_image_context_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_x_context_path	Возвращает путь к файлу x_context в корневом каталоге политики.
setcon	Устанавливает для текущего контекста безопасности значение con. Обратите внимание, что для использования этой функции требуется, чтобы всему приложению было доверено поддерживать любое разделение между старыми и новыми контекстами безопасности, в отличие от переходов на основе exec, выполняемых через setexeccon (3). Когда это возможно, разложите приложение и используйте setexeccon (3) + execve (3). Обратите внимание, что приложение может потерять доступ к своим открытым дескрипторам в результате setcon (3), если только политика не позволяет использовать дескрипторы, открытые старым контекстом.

Название функции	Описание назначения
setcon_raw	Устанавливает для текущего контекста безопасности значение con. Обратите внимание, что для использования этой функции требуется, чтобы всему приложению было доверено поддерживать любое разделение между старыми и новыми контекстами безопасности, в отличие от переходов на основе exec, выполняемых через setexeccon (3). Когда это возможно, разложите приложение и используйте setexeccon (3) + execve (3). Обратите внимание, что приложение может потерять доступ к своим открытым дескрипторам в результате setcon (3), если только политика не позволяет использовать дескрипторы, открытые старым контекстом.
setexeccon	Устанавливает контекст безопасности exec для следующего execve (3). Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setexeccon_raw	Устанавливает контекст безопасности exec для следующего execve (3). Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setexecfilecon	Устанавливает соответствующий контекст безопасности, основанный на имени файла вспомогательной программы, возвращаясь к новому контексту с указанным типом.
setfilecon	Обертка для xattr API - устанавливает контекст файла.
setfilecon_raw	Обертка для xattr API - устанавливает контекст файла.
setfscreatecon	Устанавливает контекст безопасности fscreate для последующего создания файлов. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setfscreatecon_raw	Устанавливает контекст безопасности fscreate для последующего создания файлов. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setkeycreatecon	Устанавливает контекст безопасности keycreate для последующих создания ключей. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setkeycreatecon_raw	Устанавливает контекст безопасности keycreate для последующих создания ключей. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
set_matchpathcon_canoncon	То же, что и set_matchpathcon_invalidcon (3), но также позволяет канонизацию контекста, изменяя * context на ссылку на каноническую форму. Если он не установлен, а invalidcon также не установлен, то по умолчанию используется вызов security_canonicalize_context (3).
set_matchpathcon_flags	Устанавливает флаги, управляющие работой matchpathcon_init (3) или matchpathcon (3): MATCHPATHCON_BASEONLY - обрабатывает только файл base_contexts. MATCHPATHCON_NOTRANS - не выполняет никакого контекстного перевода. MATCHPATHCON_VALIDATE - проверяет/канонизирует контексты во время init.
set_matchpathcon_invalidcon	Устанавливает функцию, используемую matchpathcon_init (3) при проверке действительности контекста в

Название функции	Описание назначения
	конфигурации file_contexts. Если это не задано, то по умолчанию используется тест на основе security_check_context (3). Эта функция также отвечает за сообщение о любой такой ошибке и может включать в себя такие сообщения, как «path» и «lineno».
set_matchpathcon_printf	Установливает функцию, используемую matchpathcon_init (3) при отображении ошибок в конфигурации file_contexts. Если не задано, то по умолчанию используется fprintf (stderr, fmt, ...).
set_selinuxmnt	Явно устанавливает путь к точке монтирования selinuxfs. Обычно это определяется автоматически во время инициализации libselinux, но это не всегда возможно, например, для/sbin/init, который выполняет начальную установку selinuxfs.
setsockcreatecon	Устанавливает контекст безопасности sockcreate для последующих созданий сокетов. Вызовите с NULL, если вы хотите сбросить к значению по умолчанию.
setsockcreatecon_raw	Устанавливает контекст безопасности sockcreate для последующих созданий сокетов. Вызовите с NULL, если вы хотите сбросить к значению по умолчанию.
sidget	Из 2.0.86 - не поддерживается
sidput	Из 2.0.86 - не поддерживается
string_to_av_perm	Преобразование имен строк для доступа к векторным разрешениям.
string_to_security_class	Преобразование имен строк в значения классов безопасности.

Приложение Е
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ
(обязательное)

АИС	–	автоматизированная информационная система
АПС «СинтезМ-ТК/ПИ»	–	аппаратно-программное средство «Персональный идентификатор пользователя»
АРМ	–	автоматизированное рабочее место
АСЗИ	–	автоматизированная система в защищенном исполнении
ВМ	–	виртуальная машина
КП ПСЗИ «СинтезМ»	–	комплекс программ «Программные средства защиты информации»
КСА	–	комплекс средств автоматизации
НСД	–	несанкционированный доступ
ОС	–	операционная система
ПК «СинтезМ-Сервер»	–	программный комплекс «Серверная операционная система»
ПК «СинтезМ-Клиент»	–	программный комплекс «Клиентская операционная система»
ПС «СинтезМ-АБ»	–	программное средство «Агент безопасности»
ПС «СинтезМ-ИПА»	–	программное средство «Сервер управления доступом»
ПС «СинтезМ-ЗЭП»	–	программное средство «Защищенная электрованная почта»
ПС «СинтезМ-КУФ»	–	программное средство «Контроль и управление функционированием»
ПС «СинтезМ-СБ»	–	программное средство «Сервер безопасности»
ПС «СинтезМ-СП»	–	программное средство «Сервер приложений»
ПС «СинтезМ- СУБД»	–	программное средство «Система управления базами данных»
ПС «СинтезМ-СХД»	–	программное средство «Системы хранения данных»
ПС «СинтезМ-О»	–	программное средство «Офис»
ПС «СинтезМ-УК»	–	программное средство «Управление конфигурацией»
ПС «СинтезМ-Репо»	–	программное средство «Репозиторий программного обеспечения»
ПС	–	программное средство
ПСЗИ	–	программное средство защиты информации
СХД	–	система хранения данных
ЦОД	–	центр обработки данных

API–интерфейс	–	*(англ.Application Programming Interface*) – это интерфейс программирования приложений или интерфейс прикладного программирования. API – это набор готовых процедур, классов, структур, функций, констант, которые предоставляются библиотекой (то есть сервером) для использования во внешних программных продуктах
BMC	–	(Baseboard management controller) – это микросхема-контроллер. Современные контроллеры BMC обеспечивают управление серверными платформами через веб-интерфейс, а также обеспечивают удаленное подключение устройств CD/DVD и работу клавиатуры-видео-мыши по сети (IP KVM), что позволяет легко изменять настройки BIOS или выполнять установку операционной системы, не имея физического доступа к оборудованию сервера
BSP	–	*Board Support Package*– пакет поддержки платформы– интегрированный пакет драйверов и/или модулей операционной системы, в операционных системах: код для поддержки конкретной операционной системы
DHCP	–	(англ. *Dynamic Host Configuration Protocol* протокол динамической настройки узла) сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому *серверу DHCP* и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок
Ebtables	–	средство для фильтрации пакетов для программных мостов Linux, работает преимущественно на втором (канальном) уровне сетевого стека
iSCSI	–	(*англ. Internet Small Computer System Interface*) протокол, который базируется на TCP/IP и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами
Intel VT	–	(Intel Virtualization Technology) виртуализация режима реальной адресации (режим совместимости с 8086)
IP-протокол	–	*Internet Protocol* (IP, досл. «межсетевой протокол») маршрутизируемый протокол сетевого уровня стека TCP/IP. Именно IP стал тем протоколом, который объединил отдельные компьютерные сети во всемирную сеть Интернет. Неотъемлемой частью протокола является адресация сети
IOMMU	–	(англ. *input/output memory management unit*) блок управления памятью (MMU) для операций ввода-вывода, блок занимается трансляцией виртуальных адресов, видимых аппаратным устройством, в физические адреса. Некоторые IOMMU также позволяют задавать различные ограничения операций ввода-вывода для защиты от неправильно работающих устройств или для изоляции, например, при использовании виртуализации

IPMB	–	(от англ. Intelligent Platform Management Interface) – интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ
IPMI	–	(от англ. *Intelligent Platform Management Interface*) – интеллектуальный интерфейс управления платформой, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ. Ключевые характеристики IPMI – мониторинг, восстановление функций управления, журналирование и инвентаризация, которые доступны независимо от процессора, BIOS'a и операционной системы. Функции управления платформой могут быть доступны, даже если система находится в выключенном состоянии
FC		(*англ. fibre channel –волоконный канал*) – семейство протоколов для высокоскоростной передачи данных
GRUB		(англ. *GRand Unified Bootloader*) загрузчик операционной системы от проекта GNU. GRUB позволяет пользователю иметь несколько установленных операционных систем и при включении компьютера выбирать одну из них для загрузки
KVM	–	(*англ. Kernel-based Virtual Machine*) – программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine)
libvirt	–	это отлаженная и со всех сторон протестированная библиотека, с помощью которой любое приложение может быть легко обучено управлению виртуальными серверами
Linux	–	общее название Unix-подобных операционных систем, основанных на одноименном ядре
Linux Bridge	–	Бридж (англ. *bridge*, мост) – это способ соединения двух сегментов Ethernet на канальном уровне, то есть без использования протоколов более высокого уровня, таких как IP. Пакеты передаются на основе Ethernet-адресов, а не IP-адресов (как в маршрутизаторе). Поскольку передача выполняется на канальном уровне (уровень 2 модели OSI), все протоколы более высокого уровня прозрачно проходят через мост
LDAP	–	(англ. *Lightweight Directory Access Protocol* «облегчённый протокол доступа к каталогам») протокол прикладного уровня для доступа к службе каталогов X.500, разработанный IETF как облегчённый вариант разработанного ITU-T протокола DAP. LDAP – относительно простой протокол, использующий TCP/IP и позволяющий производить операции аутентификации, поиска и сравнения, а также операции добавления, изменения или удаления записей
MBR	–	главная загрузочная запись (*англ. master boot record*) код и данные, необходимые для последующей загрузки операционной системы и расположенные в первых физических секторах (чаще всего в самом первом) на жёстком диске или другом устройстве хранения информации

MMU	–	(*англ. memory management unit*) – компонент аппаратного обеспечения компьютера, отвечающий за управление доступом к памяти, запрашиваемым центральным процессором. Его функции заключаются в трансляции адресов виртуальной памяти в адреса физической памяти (то есть управление виртуальной памятью), защите памяти, управлении кеш-памятью, арбитражем шины и, в более простых компьютерных архитектурах (особенно 8-битных), переключением блоков памяти Self-Hosted Engine
SELinux	–	(*англ. Security-Enhanced Linux – Linux с улучшенной безопасностью*) –реализация системы принудительного контроля доступа, которая может работать параллельно с классической избирательной системой контроля доступа
SSH	–	(*от англ. secure shell - безопасная оболочка*) это набор программ, которые позволяют регистрироваться на компьютере по сети, удаленно выполнять на нем команды, а также копировать и перемещать файлы между компьютерами
SMP	–	симметричные многопроцессорные конфигурации
SPICE	–	(*сокр. от англ. «Simple Protocol for Independent Computing Environments», то есть «Простой протокол для независимой вычислительной среды»*) –протокол, используемый в рамках проекта с аналогичным названием (но пишется строчными буквами: Spice). Проект представляет собой систему отображения удаленного дисплея, построенную для виртуальной среды, которая позволяет просматривать виртуальный «рабочий стол» вычислительной среды не только на машине, на которой он запущен, но и откуда угодно через Интернет, причем для просмотра можно использовать широкий спектр машинных архитектур
SCI	–	тонкий уровень, предоставляющий средства для вызова функций ядра из пространства пользователя
TFTP	–	(англ. Trivial File Transfer Protocol – простой протокол передачи файлов) используется главным образом для первоначальной загрузки бездисковых рабочих станций. TFTP не содержит возможностей аутентификации (хотя возможна фильтрация по IP-адресу) и основан на транспортном протоколе UDP
TLB	–	Буфер ассоциативной трансляции (англ. Translation lookaside buffer) – это специализированный кэш центрального процессора, используемый для ускорения трансляции адреса виртуальной памяти в адрес физической памяти. TLB используется всеми современными процессорами с поддержкой страничной организации памяти. TLB содержит фиксированный набор записей (от 8 до 4096) и является ассоциативной памятью. Каждая запись содержит соответствие адреса страницы виртуальной памяти адресу физической памяти
Netfilter	–	межсетевой экран (брандмауэр), встроен в ядро Linux с версии 2.4
NFS	–	*(Network File System*) протокол сетевого доступа к файловым системам. За основу взят протокол вызова удалённых процедур (ONC RPC). Позволяет подключать (монтировать) удалённые файловые системы через сеть

NTPD		(Network Time Protocol daemon) - программа-демон, которая устанавливает и поддерживает системное время, используется для синхронизации серверами точного времени
NUMA	–	(*англ. Non-Uniform Memory Access*) – это архитектура совместного доступа к памяти в многопроцессорных системах, в которой время доступа к участку памяти определяется его расположением относительно процессора
oVirt	–	свободная, кроссплатформенная система управления виртуализацией
RAID	–	*(англ. Reduntant array of independent disks)* –избыточный массив независимых дисков. Основная задача RAID-массива увеличивать скорость обмена данных и увеличивать надежность хранения данных.
REST	–	(*сокр. от англ. Representational State Transfer «передача состояния представления»*) архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы
UDP	–	(англ. *User Datagram Protocol* –протокол пользовательских датаграмм) –один из ключевых элементов TCP/IP, набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных
USB	–	*(ю-эс-би, англ. UniversalSerial Bus*) последовательный интерфейс для подключения периферийных устройств к вычислительной технике. Получил широчайшее распространение и фактически стал основным интерфейсом подключения периферии к бытовой цифровой технике.
COM-порт	–	*после́довательный порт (англ. serial port, англ. communications port*) сленговое название интерфейса стандарта RS-232 Порт называется «последовательным», так как информация через него передаётся по одному биту, последовательно бит за битом (в отличие от параллельного порта)
PDU	–	*Protocol Data Unit* – протокольная единица обмена, модуль данных протокола (в OSI представляет собой объект данных, которыми обмениваются "машины протокола" (сущности уровня) в пределах данного уровня; содержит как управляющую информацию (PCI), так и пользовательские данные)
PCI	–	(*англ. Peripheral component interconnect, дословно – взаимосвязь периферийных компонентов*) – шина ввода-вывода для подключения периферийных устройств к материнской плате компьютера
POSIX	–	(англ. *portable operating system interface*)– переносимый интерфейс операционных систем) – набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов. Стандарт создан для обеспечения совместимости различных UNIX-подобных операционных систем и переносимости прикладных программ на уровне исходного кода, но может быть использован и для не-Unix систем

PostgreSQL	–	(произносится *«Пост-Грэс-Кью-Эл»*) – свободная объектно-реляционная система управления базами данных (СУБД)
PXE	–	*(англ. Preboot eXecution Environment, произносится пикси*) среда для загрузки компьютера с помощью сетевой карты без использования локальных носителей данных (жёсткого диска, USB-накопителя и т.п.)
QEMU	–	свободная программа с открытым исходным кодом для эмуляции аппаратного обеспечения различных платформ
NFS-сервер	–	*Network File System (NFS)* – протокол сетевого доступа к файловым системам, первоначально разработан Sun Microsystems в 1984 году. За основу взят протокол вызова удалённых процедур, он позволяет подключать (монтировать) удалённые файловые системы через сеть
VFS	–	виртуальная файловая система, предоставляющая общую абстракцию интерфейса к файловым системам
VMCS	–	(Virtual Machine Control Structure) является небольшим участком физической оперативной памяти, хранящим минимально необходимые данные для запуска гостевой операционной системы, а также данные, необходимые для безопасного выхода из режима работы гостевой ОС, и некоторые настройки, относящиеся к управлению этой виртуальной машиной
VNC	–	(Virtual Network Computing) –система удалённого доступа к рабочему столу компьютера, использующая протокол RFB (англ. *Remote FrameBuffer, удалённый кадровый буфер*). Управление осуществляется путём передачи нажатий клавиш на клавиатуре и движений мыши с одного компьютера на другой и ретрансляции содержимого экрана через компьютерную сеть
VT-x (VMX)	–	Ранее известная под кодовым названием «Vanderpool», VT-x представляет собой технологию виртуализации Intel на платформе x86. Часто обозначается аббревиатурой *VMX*

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	Номер документа	Входящий номер сопрово-дительного докум. и дата	Подп.	Дата
	изме-ненных	заме-ненных	новых	аннули-рованных					