



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ДГТУ)**

КАФЕДРА «Кибербезопасность информационных систем»

**ОСНОВЫ АДМИНИСТРИРОВАНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ  
ASTRA LINUX SP**

**Практикум**

по выполнению лабораторных работ

по дисциплине

**«Операционные системы»**

Ростов-на-Дону

2025

УДК 004.451

Составители: к.т.н., доцент кафедры «Кибербезопасность информационных систем» Н.Н. Язвинская

Основы администрирования операционной системы Astra Linux SP: практикум по выполнению лабораторных работ по дисциплине «Операционные системы». / сост. Н.Н. Язвинская – Ростов-на-Дону: Донской гос. техн. ун-т, 2025. – 160с.

В практикуме кратко изложены теоретические вопросы, необходимые для успешного выполнения лабораторной работы, рабочее задание и контрольные вопросы для самопроверки.

Предназначен для обучающихся по направлению (шифр): 10.05.01 «Компьютерная безопасность»

УДК 004.451

Печатается по решению редакционно-издательского совета  
Донского государственного технического университета

Ответственный за выпуск:  
И.о. зав. кафедрой (руководитель структурного подразделения,  
ответственного за реализацию ОПОП) О.А. Сафарьян

---

В печать \_\_\_.2025 г.  
Формат 60×84/16. Объем \_\_ усл. п. л.  
Тираж \_\_ экз. Заказ № \_\_

---

Издательский центр ДГТУ  
Адрес университета и полиграфического предприятия:  
344003, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный  
технический университет, 2025

# **Содержание**

<b>Введение</b>	<b>4</b>
<b>Инструкция по технике безопасности при выполнении лабораторных работ</b>	<b>6</b>
<b>Материально-техническое обеспечение лабораторных работ</b>	<b>7</b>
<b>Общие положения</b>	<b>7</b>
<b>Требования по оформлению отчетов лабораторных работ</b>	<b>7</b>
<b>Лабораторная работа № 1 Установка и настройка операционной системы AstraLinuxSE</b>	<b>9</b>
<b>Лабораторная работа № 2 Конвейеры и перенаправление ввода-вывода, архивирование и сжатие</b>	<b>45</b>
<b>Лабораторная работа № 3 Управление процессами и планирование</b>	<b>48</b>
<b>Лабораторная работа № 4 Управление пользователями и правами</b>	<b>68</b>
<b>Лабораторная работа № 5 Файловая система ОС Astra Linux Special Edition. LVM. Swap</b>	<b>91</b>
<b>Лабораторная работа № 6 Шифрование дисков LUKS</b>	<b>110</b>
<b>Лабораторная работа № 7 РАМ - идентификация и аутентификация пользователей в AstraLinuxSE</b>	<b>117</b>
<b>Лабораторная работа № 8 Основы мандатного управления доступом. Настройка параметров мандатного управления доступом и мандатного контроля целостности</b>	<b>126</b>
<b>Перечень использованных информационных ресурсов</b>	<b>158</b>
<b>Приложение А Образец оформления лабораторной работы</b>	<b>159</b>
<b>Приложение Б Образец оформления титульного листа Журнала лабораторных работ</b>	<b>160</b>

## **Введение**

В данном пособии представлен лабораторный практикум по дисциплине Операционные системы (ОС). Цели лабораторных занятий – это формирование у будущих специалистов 10.05.01 – «Компьютерная безопасность» систематического и целостного представления о значении и месте операционных систем в системном программном обеспечении вычислительных систем, об основных способах инсталляции, настроек и поддержки системных программных продуктов. Задачи лабораторных занятий: практическое освоение пользовательского интерфейса современных операционных систем; изучение взаимодействия аппаратных и программных средств на различных уровнях; изучение различных функциональных компонент современных операционных систем; изучение принципов управления различными ресурсами вычислительной системы и структурами данных.

Для полного освоения курса ОС необходимо последовательно выполнить все задания каждой работы, предварительно ознакомившись с теоретическим материалом. Каждая лабораторная работа в данном пособии представляет собой решение отдельной проблемы для операционных систем семейства Unix, в частности, Astra Linux Special Edition – отечественной сертифицированной ОС со встроенными средствами защиты информации (СЗИ) для стабильной и безопасной работы ИТ-инфраструктур любого масштаба и обработки информации различной степени конфиденциальности.

В результате выполнения лабораторных работ по дисциплине ОС у студентов формируются следующие знания и навыки: версии ОС, их преимущества и недостатки; место ОС в составе информационной системы; основные функциональные компоненты ОС; средства мониторинга; способы выбора ОС; способы реализации информационных систем и устройств в ОС; навыки по инсталляции и отладки ОС и ее компонентов; эксплуатации современных ОС и решения поставленных задач в ОС; принципы работы

основных подсистем ОС и способы защиты от несанкционированного доступа; принципы построения и разработки ОС, а также методы расширения уже существующих систем; принципы взаимодействия аппаратных и программных средств на различных уровнях; пользовательский интерфейс современных ОС; принципы анализа и оценки эффективности функционирования ОС и ее компонентов; навыки инсталляции, настройки и администрирования параметров программного обеспечения ОС.

Лабораторные работы, представленные в пособии, можно выполнять не только на базе лабораторий университета, но и дома при наличии соответствующей учебной версии операционной системы на персональном компьютере. По результатам каждой лабораторной работы должен быть сформирован отчет, содержащий все команды и файлы, а также снимок экрана их выполнения. Каждая лабораторная работа содержит краткие теоретические сведения, которые являются дополнительным материалом к курсу лекций. В конце каждой работы есть вопросы для самоконтроля студента.

## **Инструкция по технике безопасности при выполнении лабораторных работ**

1. К самостоятельной работе с ПЭВМ допускаются лица, прошедшие инструктаж по технике безопасности, не имеющие противопоказаний по состоянию здоровья.

2. Пользователи ПЭВМ должны соблюдать правила внутреннего распорядка, установленные режимы труда и отдыха.

3. При работе с ПЭВМ возможно воздействие на работающих следующих опасных и вредных производственных факторов:

— электромагнитные излучения видеомониторов (видимые и невидимые, ионизирующие и неионизирующие);

— поражение электрическим током при работе на оборудовании без защитного заземления;

— зрительное утомление, а также неблагоприятное воздействие на зрение мерцаний символов и фона при неустойчивой работе видеомонитора, нечетком изображении на экране.

4. При работающем видеомониторе расстояние от глаз до экрана должно быть 0,6-0,7 м, уровень глаз должен находиться на центр экрана или 2/3 его высоты.

5. В случае возникновения у пользователей зрительного дискомфорта и других неблагоприятных субъективных ощущений следует ограничить время работы с ПЭВМ, сделать перерыв для отдыха или сменить деятельность на другую (без использования ПЭВМ).

6. В случае появления неисправности в работе ПЭВМ немедленно сообщить об этом преподавателю или ответственному за компьютерный класс лицу. Работу продолжать только после устранения возникшей неисправности.

7. Пользователи ПЭВМ обязаны соблюдать правила пожарной безопасности. При пожаре немедленно дождаться преподавателю и действовать по его указанию.

8. О каждом происшествии немедленно сообщать заведующему кафедрой. Работа в компьютерном классе должна быть прекращена.

## **Материально-техническое обеспечение лабораторных работ**

Для выполнения лабораторных работ необходим компьютерный класс, оснащённый ПЭВМ с установленным программным продуктом виртуализации для операционных систем. Свободные аппаратные ресурсы: от 8Гб RAM и от 40ГБ свободного места на диске.

### **Общие положения**

Трудоёмкость лабораторной работы 4 академических часа. За это время студенты должны выполнить работу, подготовить отчёт по ней и представить его преподавателю.

Каждая лабораторная работа содержит: номер, название, цель, теоретические сведения, методику выполнения, контрольные вопросы.

### **Требования по оформлению отчетов лабораторных работ**

Отчет по лабораторной работе должен быть выполнен в текстовом редакторе (например, МойОфис или др.) и оформлен согласно Приказу №242 от 16.12.2020г. «Правила оформления письменных работ обучающихся для технических направлений подготовки»

Требования по форматированию:

- шрифт TimesNewRoman
- интервал – полуторный
- поля левое – 3 см., правое – 1,5 см., верхнее и нижнее – 2 см

- абзацный отступ – 1,25
- текст должен быть выравнен по ширине.

Каждый рисунок должен располагаться по центру страницы, иметь подпись (Рисунок 1 – Окно интерфейса) и ссылку на него в тексте.

Отчет должен содержать: номер, тему лабораторной работы, кто выполнил (ФИО, группа) и проверил (ФИО, подпись, дата), цель работы и описанный процесс выполнения работы (Ход работы), далее – выводы по работе и ответы на контрольные вопросы, образец в Приложении А. В отчете **Ход работы** должен содержать скриншоты выполненной работы по пунктам задания с описанием к ним.

Если в **Методике выполнения** приводится таблица с заданиями по номерам, то номер задания соответствует списочному номеру студента в группе.

В конце отчета приводятся выводы о проделанной работе и ответ на контрольный вопрос письменно. Номер вопроса соответствует номеру студента в списке группы. Ответы на остальные контрольные вопросы готовятся устно.

Отчет защищается выполнением на ПЭВМ практических заданий согласно **Методике выполнения** и устным ответом на контрольные вопросы по выбору преподавателя.

Все отчеты по лабораторным работам скрепляются и формируются титульным листом «Журнал лабораторных работ», Приложение Б.

# **Лабораторная работа № 1**

## **Установка и настройка операционной системы AstraLinuxSE**

**Цель работы** – изучение требований к целевому компьютеру и подготовка к установке, установка ОС, настройка дополнительных параметров в Astra Linux SE.

### **Теоретические сведения**

#### **1 Введение в ОС AstraLinux**

Операционные системы Astra Linux предназначены для применения в составе информационных (автоматизированных) систем в целях обработки и защиты от несанкционированного доступа информации любой категории доступа (в соответствии с Федеральным законом от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации», статья 5, пункт 2), общедоступной информации, а также информации, доступ к которой ограничен федеральными законами (информации ограниченного доступа).

Под дистрибутивом понимается набор программного обеспечения на базе ядра Linux, поставляемый как единое целое.

Дистрибутивы Linux различаются как между собой, так и версиями дистрибутивов, хотя обладают многими схожими чертами.

Существует огромное количество дистрибутивов, большинство из них относится к одному из пяти семейств:

- Debian (представители: Debian, Ubuntu, Astra Linux);
- Red Hat (представители: Red Hat, Fedora, Centos);
- Slackware (представители: Slackware, Suse);
- Arch Linux (представители Arch Linux, Manjaro);

- Gentoo (представители Gentoo, Chromium).

Astra Linux относится к семейству UNIX-подобных ОС GNU/LINUX. Astra Linux базируется на Debian и является официально признанной веткой дистрибутива Debian. Встроенные средства защиты ОС разработаны совместно с Академией ФСБ России и Институтом системного программирования РАН.

Операционные системы Astra Linux Common Edition и Astra Linux Special Edition, рисунок 1, разработаны коллективом открытого акционерного общества «Научно-производственное объединение Русские базовые информационные технологии» и основаны на свободном программном обеспечении. С 17 декабря 2019 года правообладателем, разработчиком и производителем операционной системы специального назначения «Astra Linux Special Edition» является ООО «РусБИТех-Астра».

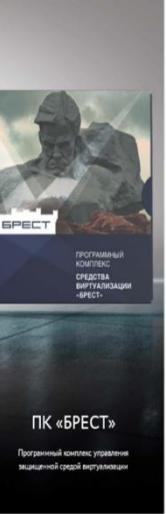
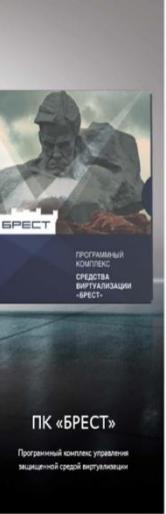
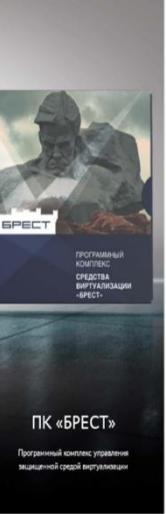
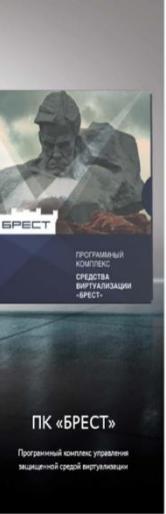
Старший номер версии	Архитектура аппаратной платформы			
1	x86-64 (Помимо Astra Linux SE использовалась для Astra Linux CE (ОС Общего Назначения (ОС ОН) Орёл версий 1.X))			
2	x86-64 (Используется для Astra Linux CE (ОС Общего Назначения (ОС ОН) Орёл версий начиная с 2.12))			
3	IBM System Z			
6	RISC MIPS little endian/big endian (Байкал Т1/Комдив 64)			
8	e2k-8с (Эльбрус-8С, Эльбрус-1С+)			

Рисунок 1 – Существующие варианты архитектуры Astra Linux

ОС Astra Linux Common Edition (релиз «Орёл») –версия ОС общего назначения без усиленных средств защиты данных. ОС в полной мере поддерживает стандарт аутентификации Kerberos. Предусмотрены три варианта централизованных систем управления аутентификацией и безопасностью:

- собственный домен Astra Linux Directory;
- Samba DC;
- универсальный open-source домен FreeIPA.

ОС специального назначения Astra Linux Special Edition предназначена для создания на ее основе автоматизированных систем в защищенном исполнении, обрабатывающих информацию со степенью секретности «совершенно секретно» включительно. ОС Astra Linux Special Edition создана и развивается на основе распространенных дистрибутивов Debian и Ubuntu.

Виды защищаемой информации:

- коммерческая тайна;
- конфиденциальная информация;
- персональные данные;
- государственная тайна, в том числе с грифом «особой важности»

(СВТ классов «3А», «2А» и «1А»)

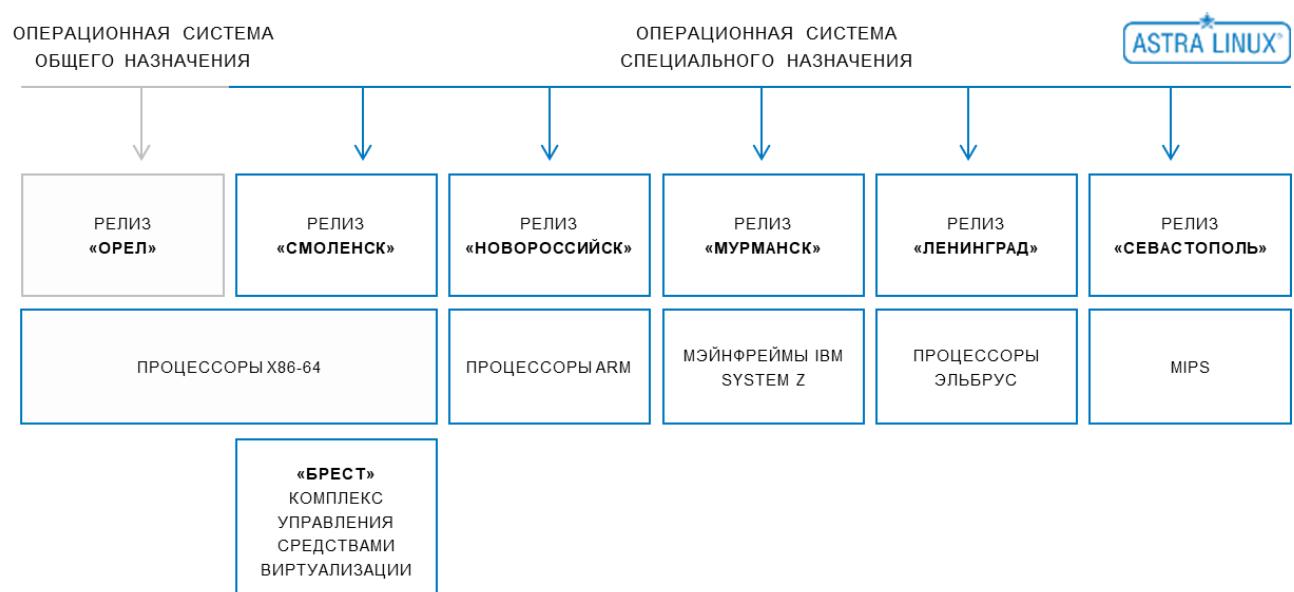


Рисунок 2 – Релизы ОС AstraLinux

Соответствие требованиям регуляторов обеспечивают:

- встроенная система безопасности PARSEC;
- мандатное разграничение доступа;

- изоляция модулей;
- очистка оперативной и внешней памяти и гарантированное удаление файлов;
- маркировка документов;
- регистрация событий;
- защита информации в графической подсистеме;
- ограничение действий пользователя (режим «киоск»);
- защита адресного пространства процессов;
- контроль замкнутости программной среды и целостности;
- средства организации единого пространства пользователей;
- защищенная среда виртуализации;
- защищенная реляционная СУБД;
- защищенные комплексы программ электронной почты и гипертекстовой обработки данных;
- средства для работы с мультимедиа и изображениями.

Продукты Astra Linux входят в реестр отечественного ПО при Минцифры. ОС Astra Linux принята в стандарт ФОИВов и госкорпораций, кроме того, она имеет полный набор сертификатов Минобороны России, ФСТЭК (Федеральная служба по техническому и экспортному контролю) и ФСБ.

## 2 Установка ОС Asrtalinux SE

Минимальные требования:

- аппаратная платформа — процессор с архитектурой x86-64 (AMD, Intel);
- оперативная память — от 4 ГБ;
- объем свободного дискового пространства — от 30 ГБ;
- устройство чтения DVD-дисков;
- стандартный монитор SVGA.

## **2.1 Начало процесса установки**

1. По умолчанию установка начнется графическая установка на русском языке, но при запуске установщика есть возможность выбрать язык. К выбору предлагаются варианты (на русском и английском языке): текстовая установка (Install), графическая установка (Graphical install), режим экспертовой установки (Expert Install) и Resque (Режим восстановления).
2. Лицензионное соглашение доступно только на русском языке.
3. Ввести имя машины (hostname)
4. Создание учетной записи пользователя. Будет создана учетная запись пользователя, входящего в группу astra-admin, то есть имеющего выполнять команды от имени суперпользователя с помощью sudo. Будет предложено выбрать имя пользователя и пароль. Если установка производится на сервер, доступный по сети, имя пользователя должно быть не словарное, чтобы уменьшить вероятность взлома сервера с помощью подбора имени пользователя и пароля. Не следует использовать простые, очевидные и легко подбираемые имена пользователей. Независимо от роли компьютера, на который устанавливается операционная система, с целью исключения несанкционированного доступа, не следует использовать короткие простые, легко подбираемые пароли. Пароль должен иметь размер не менее 8 символов (а лучше – больше), содержать символы латинского алфавита в верхнем и нижнем регистре, цифры, знаки препинания. Пароль не должен использоваться больше нигде, ни на других сервисах, сайтах и т.д.

5. Выберите ваш часовой пояс. Если вашего часового пояса нет, с помощью клавиши «Tab» выберите Go Back и выберите язык, страну и при необходимости регион, чтобы предоставить верный выбор часовых поясов.

6. Разметка диска. Доступны варианты разметки диска:

- автоматическая – используя весь диск;
- автоматическая – используя весь диск и LVM;

- автоматическая – используя весь диск и зашифрованный LVM;
- ручная.

**Пример разбиения диска емкостью 20Gb.** Он разбивается следующим образом:

- раздел ESP (EFI System Partition) – 512 Mb;
- основной раздел: файловая система Ext4, точка монтирования /, размер 10Gb;
- раздел swap – 5 Gb
- раздел /home: файловая система Ext4, точка монтирования /home, размер 5Gb

В примере разобран случай разбиения диска 20 Gb. Можно (и желательно) использовать и более емкий накопитель. Обязательно выделять раздел EFI в случае использования EFI, его размер чуть больше, чем 500 Мб. Под swap выделяется место, исходя из объема оперативной памяти. Если планируется гибернация – то объем swap должен быть не меньше, чем объем оперативной памяти (и обладать определенным запасом, +1-2Гб). В случае сервера достаточно под swap выделить 1-3 Гб, и в дальнейшем при необходимости увеличить. Под / и /home выделяется все оставшееся пространство, при этом большая часть дискового пространства выделяется под /. Отдельный раздел под /home имеет смысл выделять для сохранности документов пользователей в случае переустановки, и не имеет особого смысла для сервера.

## 2.2 Этапы разбиения диска

1. Выбирать диск для разметки – указывается его имя устройства, размер и тип. Чтобы вернуться к выбору типа разметки следует выбрать Guided partitioning.

2. Предупреждение об уничтожении предыдущей разметки. Если выбрано все устройство – это потенциально разрушительное действие. Если

на устройстве уже присутствуют разделы они будут стерты. Поэтому по умолчанию стоит отказ от продолжения – No, следует явным образом указать о согласии. Если устройство новое – выбираете Yes. Если на устройстве есть данные, и вы уверены, что они не нужны, также выбираете Yes. Если на устройстве есть важные данные, то следует нажать No и вернуться к предыдущему шагу.

3. Выделить свободное место, чтобы создать раздел.
4. Создать Efi- раздел по желанию.
5. Создать раздел для корня системы / - действия в интерфейсе аналогичны проделанным для предыдущего раздела: выбираем Create a new partition, размер 10 Gb.
5. Раздел для корневой системы Use as: Ext 4 journaling file system Mount point:/ Оставить по умолчанию и выбрать Done setting up partition, Enter.
6. Создать раздел swap по желанию. Для этого:
  - выбрать оставшийся свободный раздел;
  - Create a new partition, размер 5 Gb – по объему оперативной памяти + 1 Gb (для рабочей станции – резервируем место под гибернацию, для сервера достаточно 1-3Gb);
  - размещение Beginning;
  - Use as по умолчанию установлен как ext4, выбираем этот пункт, чтобы установить swap.

Создать swap – выбрать swap area и Enter, оставить Use as: swap area, нажать Done setting up the partition как это было сделано ранее.

7. Раздел для /home - выбрать оставшееся свободное пространство и нажать Enter. Выбрать Create a new partition, размер оставить предложенный. Далее оставить по умолчанию и нажимаем Done setting up the partition. В данном случае нужно оставить параметры по умолчанию:

- Use as: ext4 journaling file system
- Точка монтирования: /home. При необходимости, исправить (если предлагается точка монтирования /usr, следует исправить на /home).

8. Таблица разделов. Если есть ошибки, то выбрать нужный раздел и исправить. Если все верно, выбрать пункт Finish partitioning and write changes on disk.

Когда цель достигнута, то в наличии:

- раздел EFI для загрузчика (по желанию)
- раздел для операционной системы /
- раздел для swap (по желанию)
- раздел для /home – может быть полезен, если понадобится переустановить операционную систему, не уничтожая данные пользователя.

Далее необходимо еще раз все проверить. По умолчанию вопрос «Записать изменения на диск?» имеет ответ «No», выбрать вариант «Yes». После того, как подтвердили запись разделов на диск, начинается процесс установки.

9. Выбор ПО - оставить по умолчанию. При необходимости добавить SSH server (нажатием пробела), нажать TAB, выбрать Continue.

При необходимости удаленного доступа к компьютеру можно установить ssh. В этом случае следует учесть, что по ssh доступ к компьютеру может получить и злоумышленник, потому имя пользователя и пароль должны быть нетривиальными, не должно быть у злоумышленника возможности их быстро подобрать.

10. Установка ALD пока не нужна, пропустить, нажать Continue. Подождать пока система распаковывает и устанавливает ПО.

## **2.3 Дополнительные настройки**

Пробел устанавливает или снимает дополнительную опцию, Tab переключает на кнопку Continue. Выбрать Continue и Enter

Доступны для включения следующие опции:

- включить проверку сигнатуры исполняемых файлов – режим замкнутой программной среды, в котором могут быть запущены только те

ELF-приложения, исполняемые файлы которых подписаны цифровой подписью разработчика, чей открытый ключ добавлен в перечень ключей, которым доверяет операционная система. Исполняемые файлы и разделяемые библиотеки с неверной цифровой подписью или без нее не могут быть загружены и запущены;

- запретить установку бита исполнения – запрет на установку для файлов права x (chmod +x) для обычных пользователей;
- использовать Hardened-ядро – модифицированную версию ядра с дополнительными механизмами защиты, такими как очистка остаточной информации в стеке и куче ядра, содержит ряд изменений для защиты ядра от внедрения вредоносного кода;
- запретить вывод меню загрузчика – при запуске операционной системы меню загрузчика GRUB выводиться не будет;
- включить очистку swap – разделов страничного обмена (с учетом того, что очистка освобождаемых ресурсов как правило не работает на SSD-дисках);
- включить очистку освобождаемых областей для EXT-разделов. При включенной опции данные любых удаляемых/урезаемых файлов в пределах заданной ФС предварительно очищаются предопределенной или псевдослучайной маскирующей последовательностью. Следует учесть, что очистка освобождаемых ресурсов как правило не работает на SSD-дисках;
- включить блокировку консоли – запрещает обычным пользователям консольный вход в систему;
- включить межсетевой экран ufw – при выборе данного пункта будет включен межсетевой экран ufw и запущена фильтрация сетевых пакетов в соответствии с заданными настройками;
- включить системные ограничения ulimits – при выборе данного пункта будут включены системные ограничения, установленные в файле /etc/security/limits.conf
- отключить возможность трассировки ptrace – при выборе данного пункта будет отключена возможность трассировки и отладки выполнения

программного кода.

При включении блокировки интерпретаторов блокируется несанкционированное использование интерпретатора для выполнения кода напрямую из командной строки или из неименованного канала (pipe). При этом сценарии из каталога /usr/bin/, написанные для этих интерпретаторов, выполняются в штатном режиме.

- отключить автоматическую конфигурацию сети – при выборе данного пункта будет отключена автоматическая настройка сети в процессе установки Astra Linux;

-включить 32-битный загрузчик –при выборе данного пункта из системы будет удален 64-х битный загрузчик EFI и установлен 32-х битный загрузчик EFI. **Обратите внимание!** Выбор данной опции при установке на 64-х битную вычислительную машину с поддержкой EFI может привести к тому, что установленная система не загрузится.

11. Установка GRUB – введите и подтвердите пароль.
12. Система установлена, необходимо выполнить перезагрузку.
13. Войти в систему пользователем, созданным при установке.  
Терминальный вход пользователем root заблокирован (не установлен пароль). Пользователь, созданный во время установки, входит в группу astra-admin и может выполнять любые команды через sudo (/etc/sudoers).

14. Установить дополнения –Update.
15. Включить мандатный контроля целостности на системные каталоги.  
По умолчанию он выключен, его следует включить. Узнать состояние мандатного контроля целостности

```
sudo set-fs-ilev status
```

Включение мандатного контроля целостности на системные каталоги выполняется командой sudo set-fs-ilev

### **3 Вход в систему**

Сессия – сеанс взаимодействия пользователя с операционной системой, открывающийся после успешной авторизации в системе и завершающийся после выхода из сессии. Взаимодействие осуществляется благодаря оболочке – программе, предоставляющей интерфейс пользователя, графический или текстовый. Рабочий стол появляется на экране монитора после входа пользователя в графическую среду. Он содержит пространство рабочего стола с фоновым изображением, панель задач и графические элементы интерфейса пользователя. Само пространство рабочего стола и панель задач также являются элементами интерфейса пользователя.

Сессия – сеанс взаимодействия пользователя с операционной системой, осуществляемый благодаря текстовой (как правило *bash*) или графической (*Fly-wm*) оболочке. Вход в сессию осуществляется после успешной авторизации пользователя. Сессия включает набор процессов, запущенных пользователем оболочки или автоматически. Запущенные в сессии процесс получают привилегии и идентификатор пользователя, выполнившего вход в систему. Графическая сессия может быть запущена в одном из режимов (десктоп, планшет, мобильный), адаптированном под те или иные типы устройств, на которых запущена операционная система.

В состав Astra Linux входит рабочий стол Fly, который состоит из оконного менеджера и набора графических утилит и программ, как пользовательских, так и административных. После успешного ввода логина и пароля и, при необходимости, указания атрибутов безопасности учетной записи будет запущена сессия вашего пользователя и открыт рабочий стол, рисунок 3. Фон рабочего стола сигнализирует о режиме целостности – если он красный, это означает что сессия запущена в высоком режиме целостности и предназначена для администрирования. При низком уровне целостности фон рабочего стола – синий.

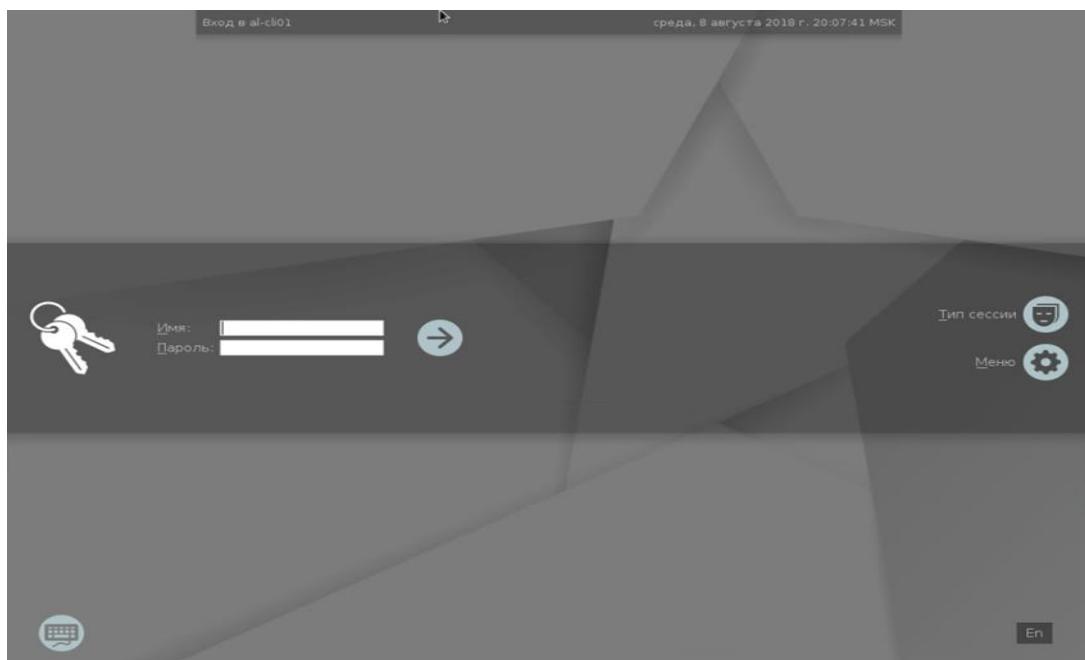


Рисунок 3 – Графический вход пользователя в систему

Графический вход пользователя в систему осуществляется при помощи утилит «Вход в систему: сервер» (fly-dm) и «Вход в систему: GUI» (fly-qdm), переход к которым происходит после окончания работы загрузчика. Интерфейс графического входа позволяет при входе выбрать тип сессий, в которой будет работать пользователь. Для этого перед входом нужно кликнуть на пиктограмму «Тип сессии», рисунок 4:

- десктоп – устанавливается по умолчанию;
- режим восстановления – в графической среде будет запущен только терминал, что позволяет администрировать в режиме командной строки, но при необходимости запускать и графические приложения;
- мобильный;
- планшетный.

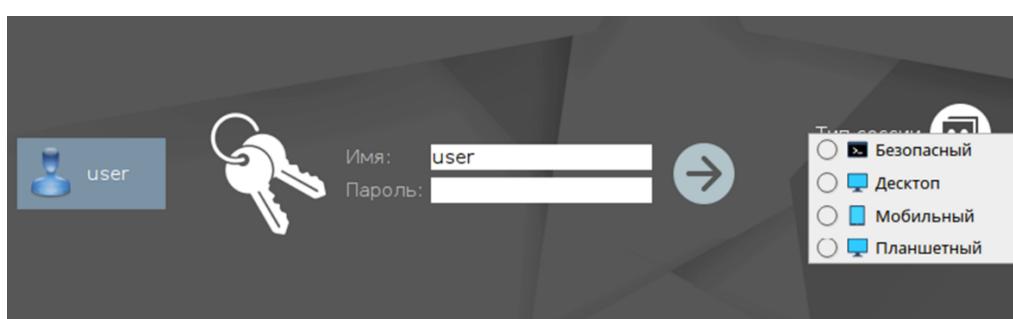


Рисунок 4 – Типы сессий

Для загрузки рабочего стола ОС необходимо при графическом входе в ОС установить тип сессии Desktop. Рабочий стол также запускается в режиме, оптимизированном для работы на устройствах с сенсорными экранами: в планшетном режиме (тип сессии Планшетный [Tablet]) и в режиме для мобильных устройств (тип сессии Мобильный [Mobile]).

Меню:

- сменить пользователям (Alt-I) – позволяет переключаться между активными сеансами пользователей (включая входы на виртуальные терминалы). При выборе этого пункта будут отображаться активные сессии и возможность выбора окна входа;
- перезапустить X Server (Alt-E) – перезапустить графическое окружение;
- виртуальная клавиатура (Alt-K) – откроет виртуальную клавиатуру (такой же эффект достигается щелчком по значку клавиатуры в левом нижнем углу экрана);
- консольный вход (Alt-N) –временно закроет графический режим и переведет в режим терминала, все сессии будут принудительно закрыты;
- завершение работы (Alt-S) – выведет окно завершения работы.



- щелчок по этому значку, в случае корректных имени пользователя и пароля откроет окно выбора уровня конфиденциальности и целостности. В случае неверного ввода имени пользователя и/или пароля на короткое время появится надпись «Login failed».

***Обратите внимание!*** на регистр – регистр имеет значение, буква в верхнем и нижнем регистре считаются разными символами.

Для перехода в консольный режим следует нажать на клавиатуре <Ctrl>+<Левый Alt>+<F1>, либо <Ctrl>+<Левый Alt>+<F2> и т.д. до <Ctrl>+<Левый Alt>+<F6>. Произойдет переход к одной из 6 текстовых виртуальных консолей. После окончания работы в текстовом режиме следует набрать команду: \$ exit

Обратно в графический рабочий стол можно вернуться, нажав сочетание клавиш <Ctrl>+<Левый Alt>+<F7>.

Защищенная графическая подсистема в составе ОС функционирует с использованием графического сервера Xorg. В нее также входит рабочий стол Fly, который состоит из оконного менеджера (fly-wm) и большого набора графических утилит и программ, как пользовательских, так и административных, рисунок 5.

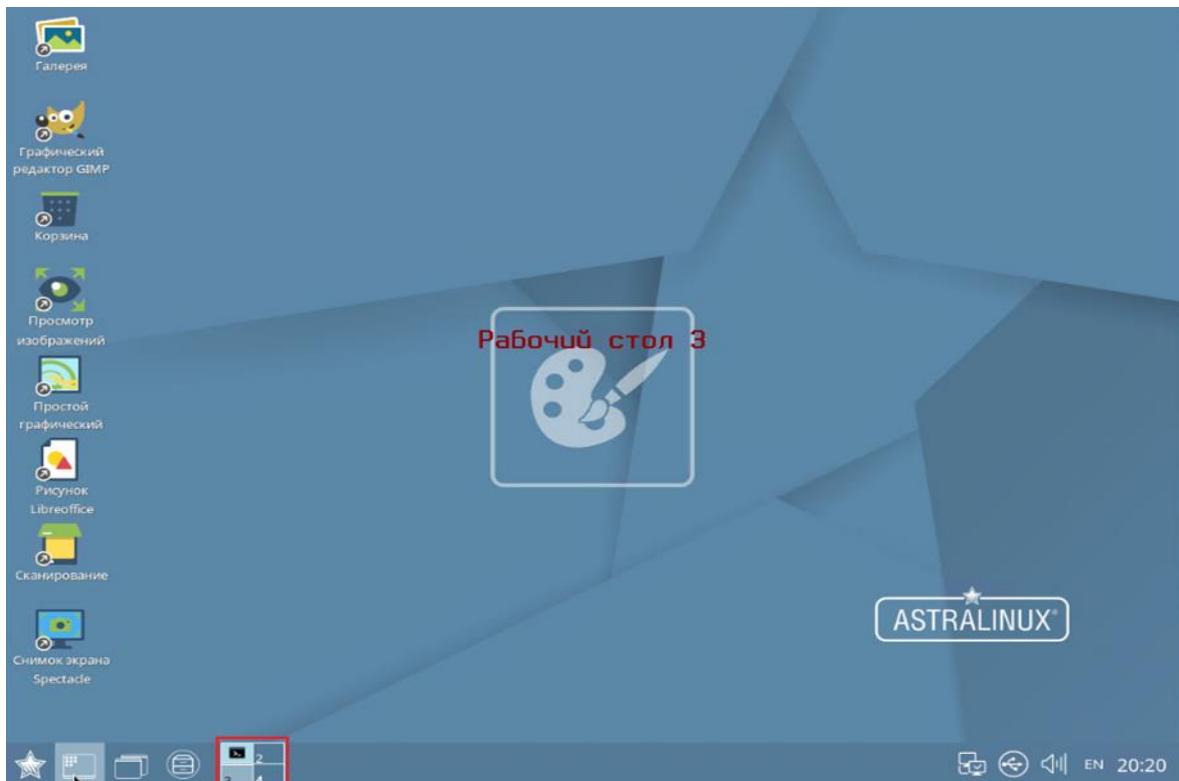


Рисунок 5 - Рабочий стол Fly

Интегрированный менеджер рабочих столов позволяет размещать окна приложений в пространстве, превышающем размер видимой области экрана.

По умолчанию менеджер поддерживает четыре рабочих стола в конфигурации два по вертикали и два по горизонтали. На панели быстрого доступа располагается переключатель рабочих столов. Для удобства навигации на каждом из рабочих столов, кроме первого, в центре располагается маркирующий этот стол рисунок.

В планшетном режиме на панели быстрого доступа появляются новые кнопки: кнопка для закрытия активного окна приложения и кнопка поворота рабочего стола на 90 градусов, а на панели задач в области уведомлений новые

значки: вызов экранной клавиатуры и индикатор заряда аккумулятора. Управление в планшетном режиме выполняется жестами мульти-касаний (пальцевыми командами).

Мобильный интерфейс. Программа fly-launcher («Лончер») загружает среду рабочего стола для мобильных устройств и организует мобильный интерфейс. Мобильный интерфейс предполагает работу с сенсорными экранами. Управление осуществляется жестами мульти-касаний.

Интегрированные настройки позволяют установить до трёх экранов на рабочем столе (по умолчанию три). Рабочий стол появляется на экране мобильного устройства после входа пользователя в графическую среду.

После установки ОС значения параметров настройки рабочего стола устанавливаются по умолчанию. Изменение настроек выполняется с помощью пунктов меню, открывающихся кнопкой «Настройки» на выдвигающейся панели управления. Утилиты, адаптированные для работы в режиме мобильного интерфейса, приведены в таблице 1.

Таблица 1 – Утилиты мобильного интерфейса

Утилита	Название	Описание
<b>fly-mail</b>	<b>Почта</b>	Работа с электронной почтой.
<b>fly-qml-dialer</b>	<b>Телефон</b>	Совершение звонков. Журнал вызовов, просмотр контактов, набор номера и т.д.
<b>fly-sms</b>	<b>СМС</b>	Просмотр, отправление и принятие текстовых сообщений (SMS).
<b>fly-phone-webbrowser</b>	<b>Интернет-браузер</b>	Веб-браузер, предназначенный для использования на мобильных устройствах и оптимизированный под маленький экран.

### **3.1 Уровни конфиденциальности, целостности, категории**

В случае успешного ввода имени пользователя и пароля, если учетная запись пользователя активна, появляется окно выбора уровня конфиденциальности, неиерархического уровня целостности и неиерархической категории целостности. Уровень конфиденциальности задает степень повышающейся секретности. По умолчанию (а также, если уровни конфиденциальности) не заданы – используется самый низкий уровень – Не секретно.

Возможные варианты уровней конфиденциальности – Не секретно, Для служебного пользования, Секретно, Совершенно секретно.

Уровень целостности (неиерархический уровень целостности) – атрибут, отвечающий за то, чтобы информацию не могли изменять те, кому ее не положено изменять. По умолчанию, доступны два уровня целостности – Низкий и Высокий.

Для обычного использования ЭВМ применяется низкий уровень целостности, для администрирования – высокий.

Категория (неиерархическая категория) – также, как и уровень конфиденциальности, служит для того, чтобы информация не попадала тому, кто не уполномочен ее получать. Если уровни конфиденциальности – «вертикальная градация» полномочий (с линейным повышением-понижением), то неиерархические категории – это «горизонтальная градация». Например, сотрудники разных отделов могут получить категории согласно их отделам, а руководитель вышестоящей организационной структуры может иметь доступ к категориям всех нижестоящих отделов.

В случае, если выбран высокий уровень целостности, цвет фона рабочего стола, рамок и заголовок окон, панели быстрого запуска и области уведомления, устанавливается красный, что сигнализирует о том, что практически все действия разрешены и могут быть потенциально разрушительны. Высокий уровень целостности следует применять только для

администрирования, он не подходит для постоянной работы. При выполнении административных задач требуется крайне внимательно подходить ко всем совершаемым действиям.

В панели уведомлений доступен индикатор мандатных уровней доступа. Он отображает уровень конфиденциальности (по умолчанию 0), также при наведении курсора мыши (во всплывающем информационном поле) либо при щелчке мыши по индикатору мандатных уровней (в отдельном окне) будет показана информация о мандатных уровнях. Уровень конфиденциальности, уровень целостности (63 – высокий уровень целостности), категориях и ролях. Элементы интерфейса (рабочий стол, ярлыки, кнопка и меню «Пуск», панель быстрого запуска, переключатель задач, область уведомлений и т.д.) при работе в низком уровне целостности идентичны (за исключением прав и расцветки).

## 4 Завершение работы

Если рабочий стол Fly запущен, то для завершения работы пользователю следует нажать кнопку меню [Пуск] на панели задач. На открывшейся панели меню нажать на кнопку [Завершение работы] (в случае классического меню «Пуск» – выбрать пункт «Завершение работы»), выбрать пункт «Диалог выхода» в подменю «Системные», либо набрать команду:

```
$ fly-shutdown-dialog
```

Команду можно набрать, открыв терминал Fly, используя меню [Пуск] → [Системные] → [Терминал Fly]

Блокировка – сессия пользователя будет заблокирована, появится окно блокировки. Для возобновления работы потребуется ввести пароль. Блокировку следует использовать, если требуется отойти от рабочего места.

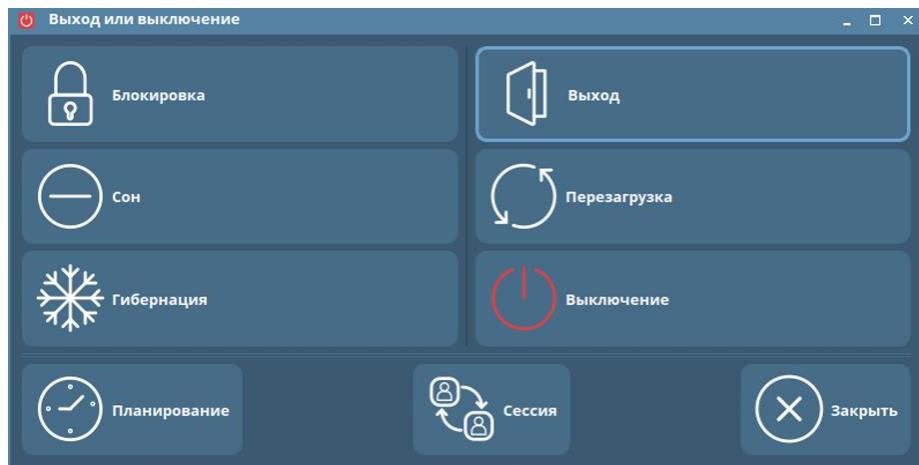


Рисунок 6 – Режимы выключения

Сон – «легкое выключение» компьютера. Компьютер приостанавливает работу, но в оперативной памяти сохраняются данные. Поэтому обратное «включение» – «пробуждение» компьютера происходит очень быстро. В случае отключения питания все несохраненные данные будут потеряны.

Гибернация – похоже на сон, но в этом случае данные из оперативной памяти сохраняются на диск. Возобновление работы возможно даже при отключении питания.

Выход из сессии – завершается сессия текущего пользователя и происходит переход в окно входа.

Перезагрузка – компьютер будет перезагружен.

Выключение – компьютер будет выключен, все программы будут завершены.

Планирование – позволяет выполнить любую из вышеперечисленных операций не мгновенно, а запланировать ее на некоторое время. Можно выбрать, как скоро операция будет выполнена, а также когда об этом следует напомнить – предупредить.

Новый вход – по нажатию на эту кнопку появляются/скрываются две дополнительные кнопки «Отдельный» (Console) и в «Окне». Данная операция может понадобится, когда потребуется временно войти в систему под другим пользователем, не закрывая программы и документы в открытой сессии.

Отдельный (Console) – в этом случае данная сессия будет

приостановлена (но не закрыта), и будет открыто окно входа. Чтобы вернуться обратно, нужно воспользоваться на экране входа «Меню»(Menu), выбрать пункт «Смена сессии»(Switch User) и выбрать открытую сессию для перехода в нее (будет открыт экран блокировки для ввода пароля пользователя).

В окне – данная сессия не будет приостановлена а новая сессия будет открыта во вложенном окне.

Закрыть – закрывает окно «Выход или выключение».

Чтобы переключаться между отдельными сессиями, можно воспользоваться Меню Пуск>Завершение работы (Exit)>Новая сессия(Switch)>Отдельная сессия (Console)>Экран входа>Меню> «Смена сессии» (Switch User) и выбрать нужную сессию. Значительно проще для переключения сессий использовать клавиатурные сокращения. Ctrl-Alt-F1, Ctrl-Alt-F2... Ctrl-Alt-F6 для текстовых сессий, Ctrl-Alt-F7, Ctrl-Alt-F8, Ctrl-Alt-F9 и т.д. для графических.

## 5 Пользовательские настройки

Утилита Менеджер файлов (**fly-fm**) предназначена для просмотра папок рабочего стола и элементов ФС и выполнения основных функций управления файлами, рисунок 7. Позволяет подключать и отключать ФС носителей доступных устройств хранения данных, таких как локальные жесткие диски и их разделы, компакт- и DVD-диски, USB-накопители.

В планшетном режиме программа по умолчанию запускается с установленными настройками, оптимизированными для работы на устройствах с сенсорными экранами. В частности, на панели просмотра слева от имени элемента отображаются значки для переключения флага выполнения групповых операций, и отображение графических элементов видоизменяется.

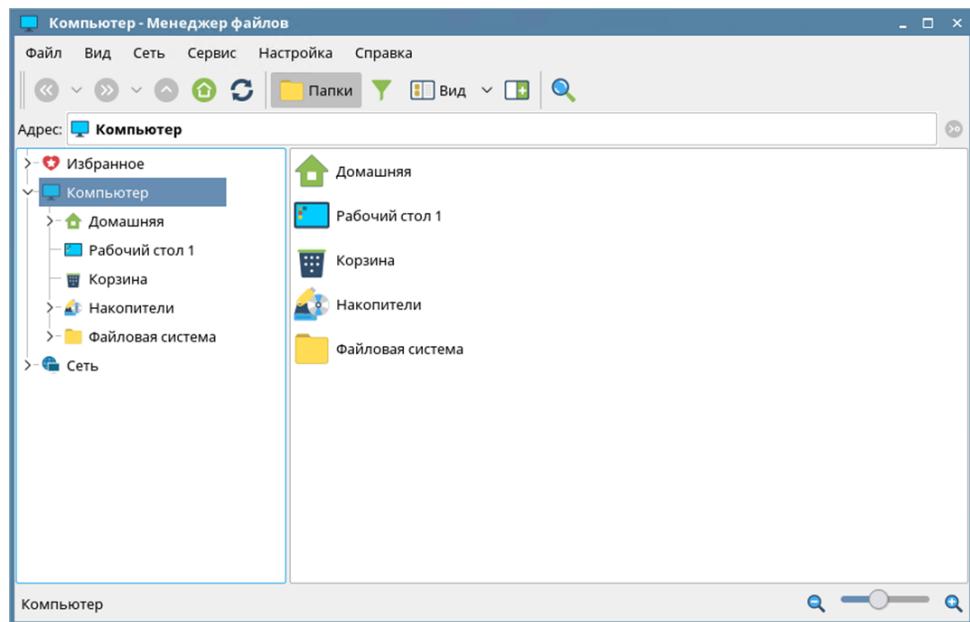


Рисунок 7 – менеджер файлов

Программа «Панель управления» (**fly-admin-center**) – предоставляет централизованный доступ к графическим утилитам настройки и администрирования системы.

Главное окно программы содержит боковую панель меню (Панель меню) с рабочей панелью (справа) и управляющие элементы, рисунок 8.

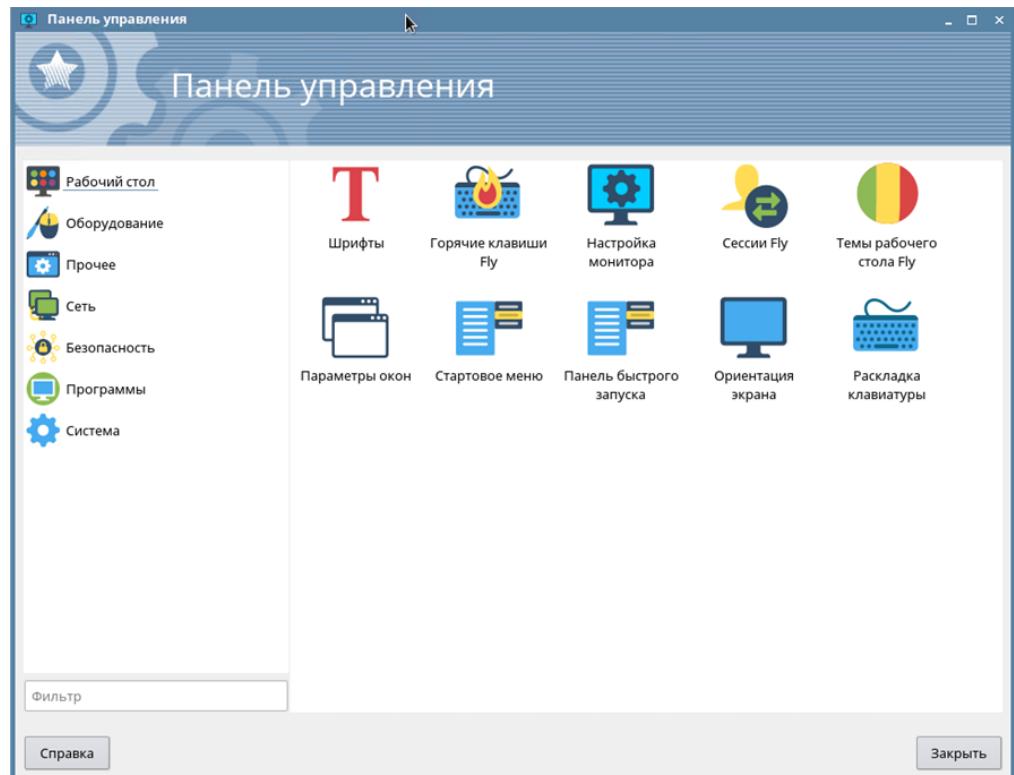


Рисунок 8 – Окно панели управления

Программа «Автозапуск» (**fly-admin-autostart**) отвечает за настройку автоматического запуска приложений при загрузке рабочего стола.

Для запуска программы «Автозапуск»: [Панель управления] → [Система] → [Автостарт], рисунок 9.

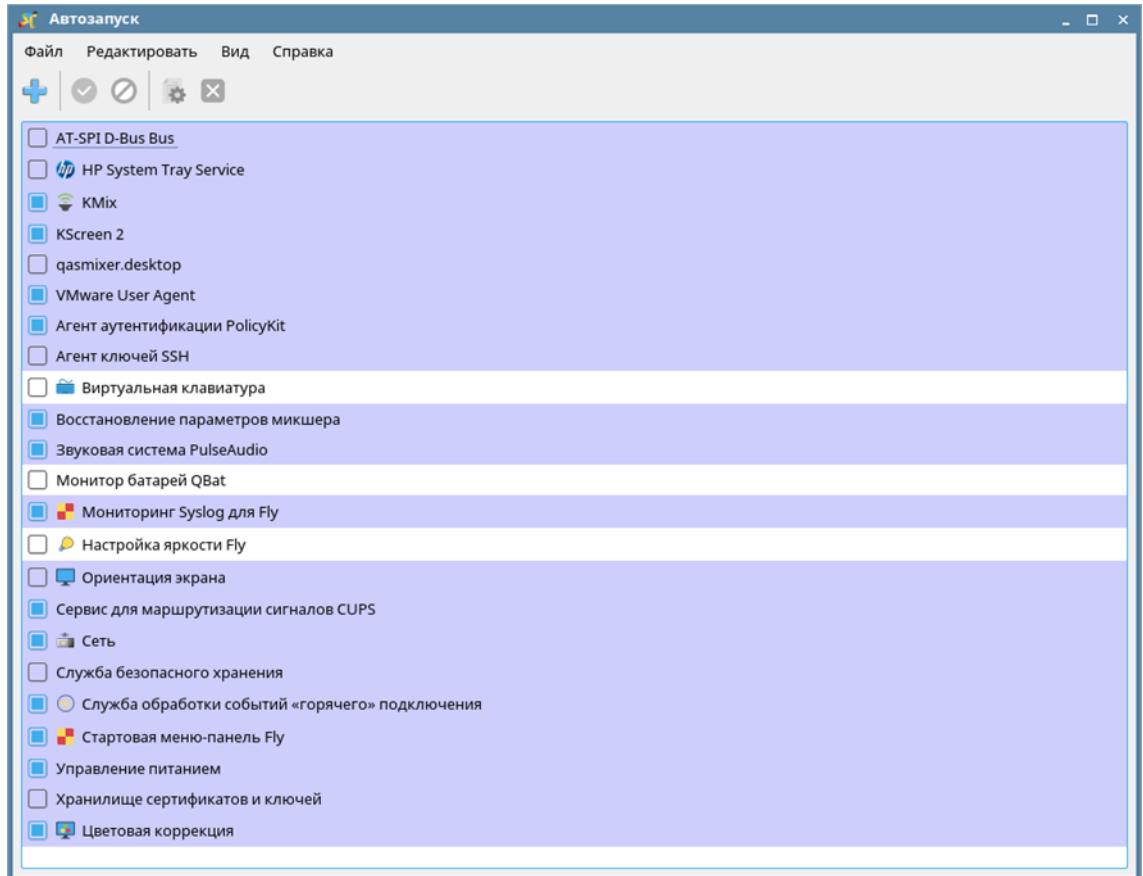


Рисунок 9 – Окно настройки автозапуска

Программа «Дата и время» (**fly-admin-date**) позволяет изменить формат отображения времени на системных часах и формат отображения даты и времени на всплывающем сообщении при наведении курсора мыши на системные часы в области уведомлений на панели задач.

Для запуска программы «Дата и время»: [Панель управления] → [Система] → [Дата и время].

Доступ к изменению установок регулируется программой «Санкции Policykit» (**fly-admin-policykit-1**), рисунок 11.

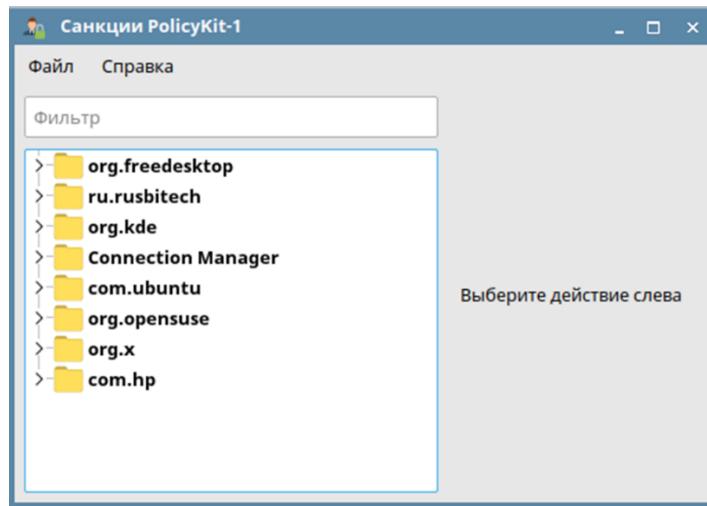


Рисунок 10 – Окно изменений установок Policykit

Программа «Настройка монитора» (**fly-admin-screen**) позволяет управлять параметрами всех подключенных мониторов/экранов при помощи раздела «Экраны» и настраивать параметры цветопередачи в разделе «Цветовая коррекция», рисунок 12.

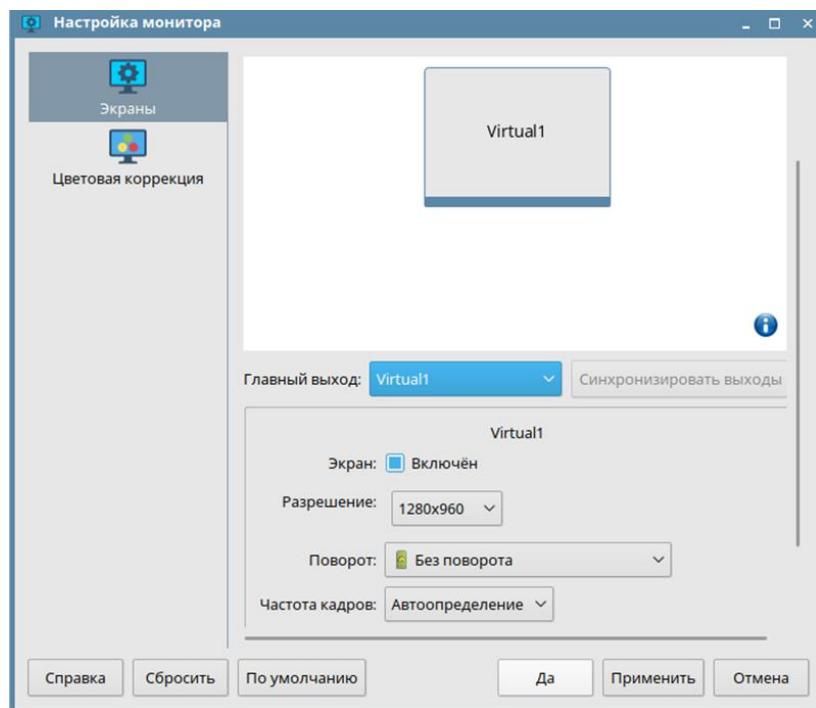


Рисунок 11- Окно настройки дисплея

Программа «Обработка подключения устройств» (**fly-admin-reflex**) отвечает за настройку службы обработки «горячего» подключения устройств, рисунок 12. Позволяет установить условия, в соответствии с которыми при подключении и отключении определенных устройств будут выполняться те или иные действия.

Доступные действия определяются в файлах конфигурации:

**/usr/share/fly/data/fly-reflex/actions/\*.desktop.**

**~/.fly/fly-reflex/actions/\*.desktop**

Для запуска программы

«Обработка подключения

устройств»:

[Панель управления] →

[Оборудование] →

[Обработка «горячего»

подключения].

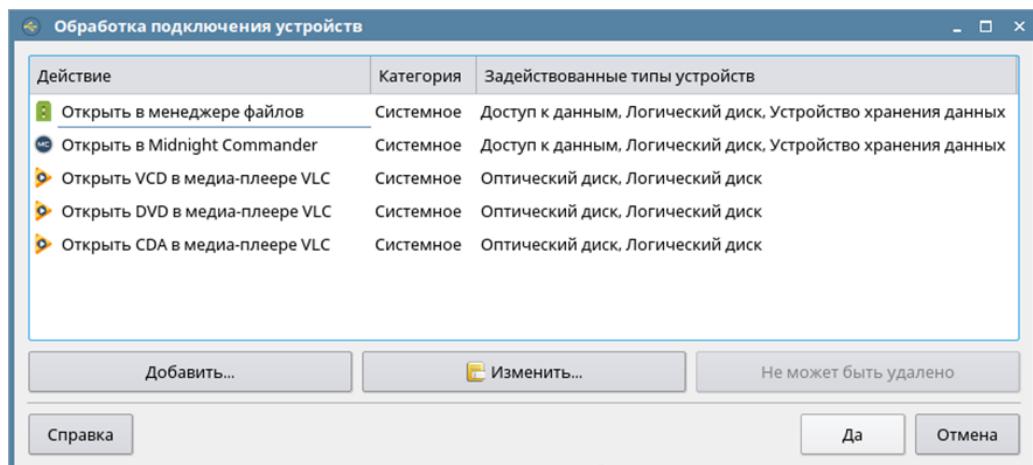


Рисунок 12 – Окно настройки службы обработки подключения устройств

Программа «Программа Меню и ярлыки» (fly-menedit) предназначена для редактирования панели быстрого запуска рабочего стола и стартового меню «Пуск», а также для создания и редактирования ярлыков и коллекций ярлыков с помощью этих меню. В заголовке окна рядом со значком отображается название меню и маршрутное имя соответствующего файла.

Для запуска программы «Программа Меню и ярлыки»: [Панель управления] → [Рабочий стол] → [Панель быстрого запуска].

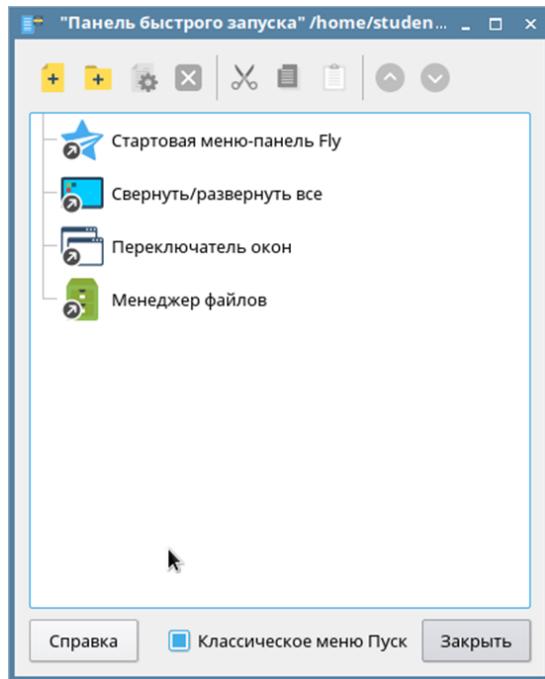


Рисунок 13 – Настройка панели быстрого запуска

Программа «Настройка параметров окон» (**fly-admin-winprops**) предназначена для настройки свойств окон.

Для запуска программы «Настройка параметров окон»: [Панель управления] → [Рабочий стол] → [Параметры окон].

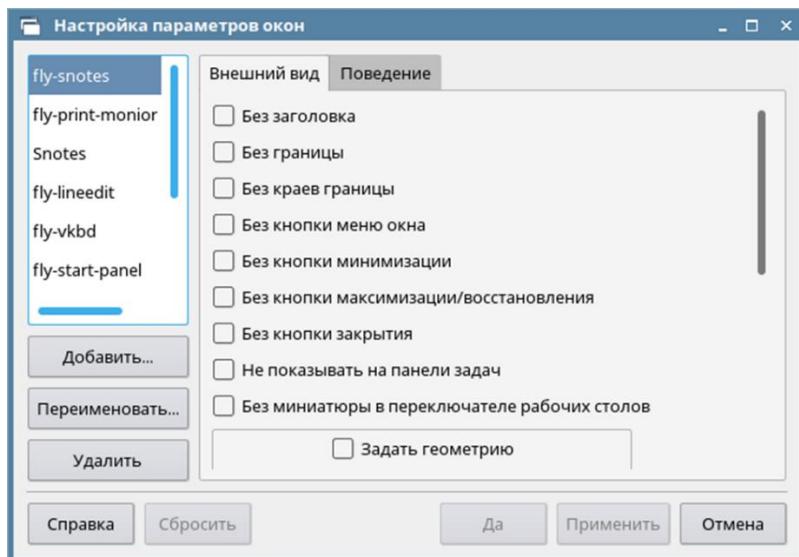


Рисунок 14 - Настройка параметров окон

Программа «Планировщик задач» (**fly-admin-cron**) позволяет в графическом режиме изменять настройки демона cron. С ее помощью можно установить расписание задач: назначить выполнение программы (сценария или приложения) в установленное время, установить время прекращения

выполнения назначеннной программы и настроить среду ее выполнения (переменные окружения), а также разрешить или запретить выполнение уже установленной задачи. При запуске с уровнем классификационной метки, отличным от нулевого, появляется сообщение о том, что какие-либо изменения не применяются.

Для запуска программы «Планировщик задач»: [Панель управления] → [Прочее] → [Планировщик задач].

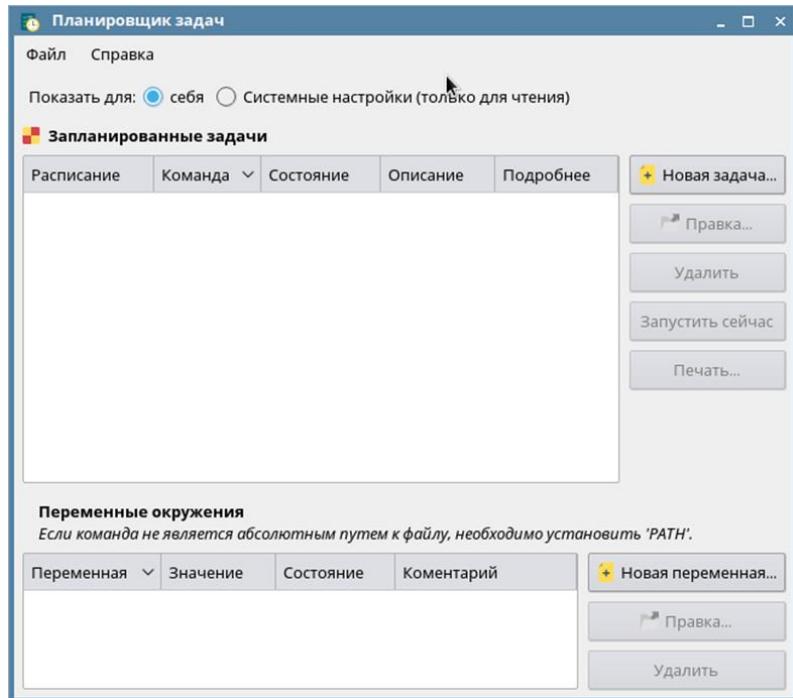


Рисунок 15 – Окно планировщика задач

Программа «Раскладка клавиатуры» (**fly-xkbmap**) предназначена для переключения раскладок клавиатуры. Основана на расширении XKB для X11. Позволяет использовать различные раскладки клавиатуры.

Для запуска программы «Раскладка клавиатуры»: [Панель управления] → [Рабочий стол] → [Раскладка клавиатуры].

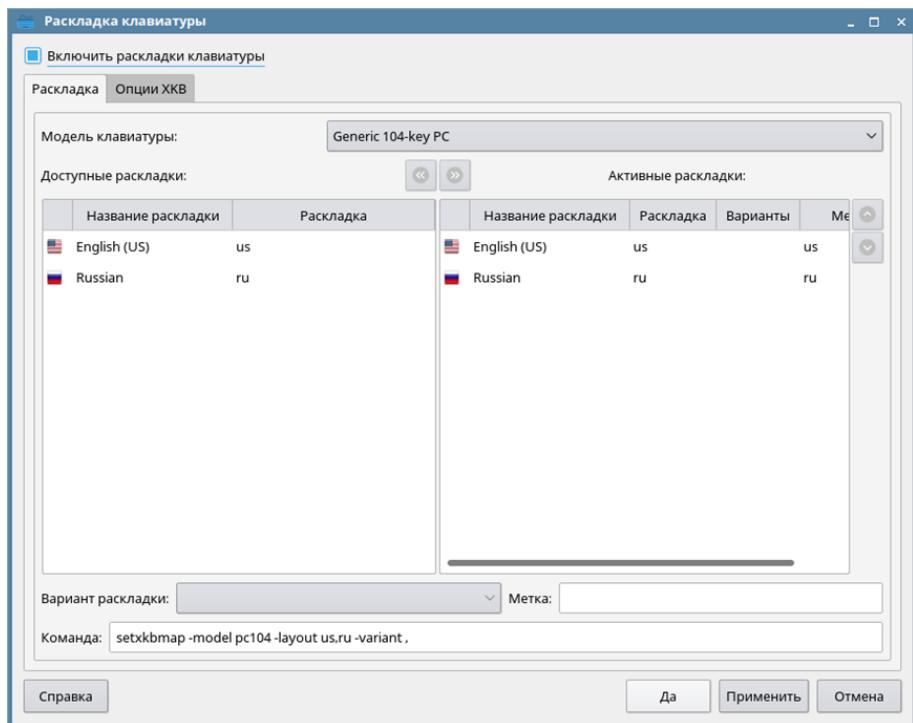


Рисунок 16 – Окно изменения раскладки клавиатуры

Программа «Программа Меню и ярлыки» (**fly-menedit**) предназначена для редактирования панели быстрого запуска рабочего стола и стартового меню «Пуск», а также для создания и редактирования ярлыков и коллекций ярлыков с помощью этих меню. В заголовке окна рядом со значком отображается название меню и маршрутное имя соответствующего файла.

Для запуска программы «Программа Меню и ярлыки»: [Панель управления] → [Рабочий стол] → [Стартовое меню].

Программа «Темы рабочего стола Fly» (**fly-admin-theme**) – это редактор тем и настроек рабочего стола.

Для запуска программы «Темы рабочего стола Fly»: [Панель управления] → [Рабочий стол] → [Темы рабочего стола]

Программа «Темы рабочего стола Fly» (**fly-admin-theme**) – это редактор тем и настроек рабочего стола.

Для запуска программы «Темы рабочего стола Fly»: [Панель управления] → [Рабочий стол] → [Темы рабочего стола]

Программа «Менеджер шрифтов» (**fly-admin-fonts**) позволяет получать информацию о доступных системных шрифтах и просматривать их

начертание, а также импортировать новые шрифты.

Для запуска программы «Менеджер шрифтов»: [Панель управления] → [Рабочий стол] → [Шрифты].

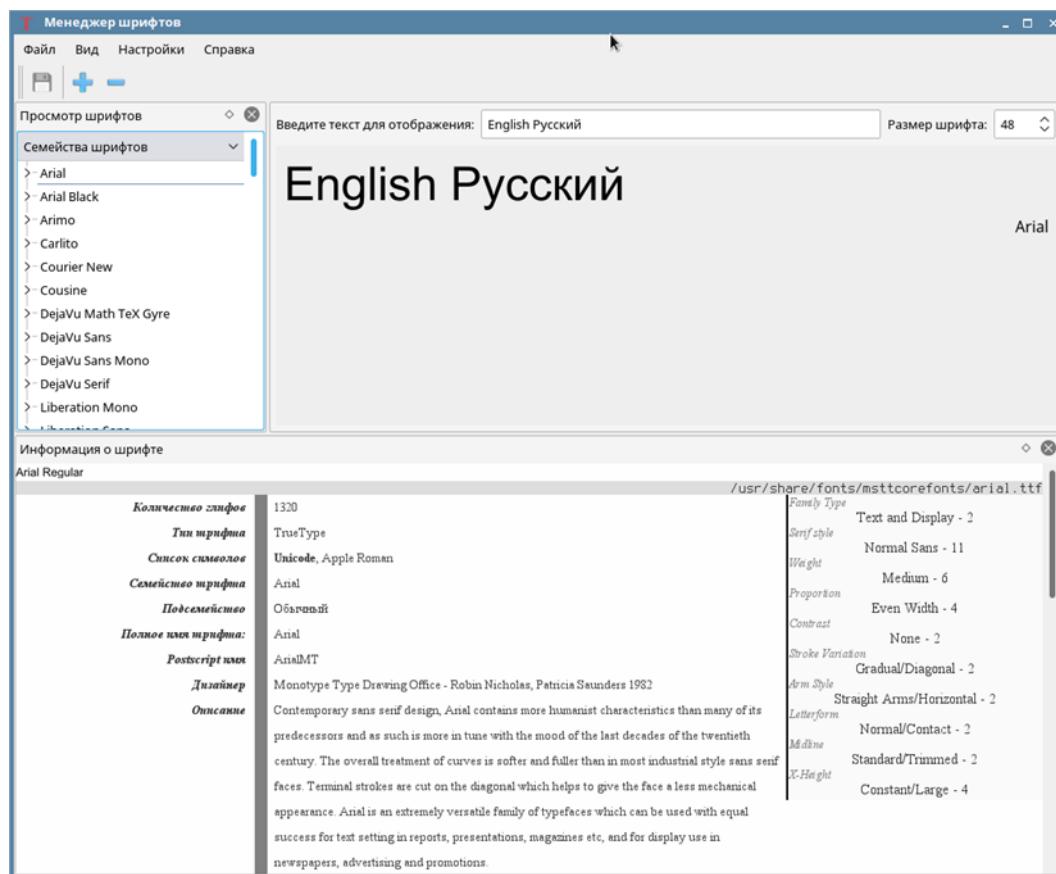


Рисунок 17 – Окно настройки шрифтов

Программа «Менеджер печати Fly» (**fly-admin-printer**) – это программа настройки печати в графическом режиме. Позволяет администрировать систему печати: устанавливать, настраивать и удалять принтеры и классы принтеров, а также настраивать сервер печати и управлять заданиями на печать. Действия по администрированию должны выполняться от имени пользователя из системной группы CUPS (по умолчанию lpadmin) или от администратора ОС, пользователя из группы astra-admins. Обычный пользователь может устанавливать настройки печати и опции принтера, а также управлять заданиями на печать (удалять, приостанавливать, возобновлять печать и устанавливать отложенную печать). Для вызова привилегированных действий запрашивается авторизация.

Для запуска программы «Менеджер печати»: [Панель управления] →

[Оборудование] → [Принтеры].

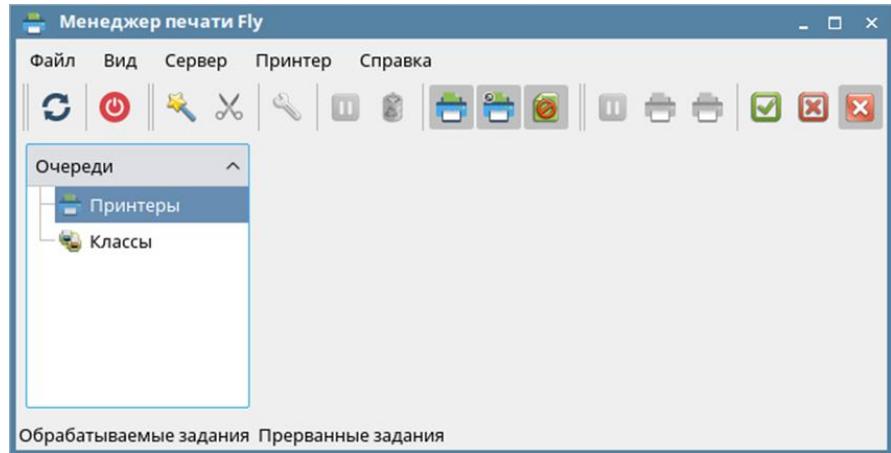


Рисунок 18 – Настройка менеджера печати

**Установка.** Программа «Менеджер пакетов Synaptic» (Synaptic Package Manager) – это графический интерфейс для системы управления пакетами apt.

Основные функции:

- Установка пакетов.
- Удаление пакетов.
- Обновление отдельных пакетов.
- Обновление всей системы (dist-upgrade).
- Поиск пакетов.
- Ведение списка используемых репозиториев (sources.list).

Для запуска программы «Менеджер пакетов Synaptic»: [Панель управления] → [Программы] → [Менеджер пакетов Synaptic].

Программа «Менеджер устройств» (**fly-admin-device-manager**) предназначена для получения информации об устройствах, доступных в системе, а также для настройки некоторых из них.

Для запуска программы «Менеджер устройств»: [Системные] → [Менеджер устройств].

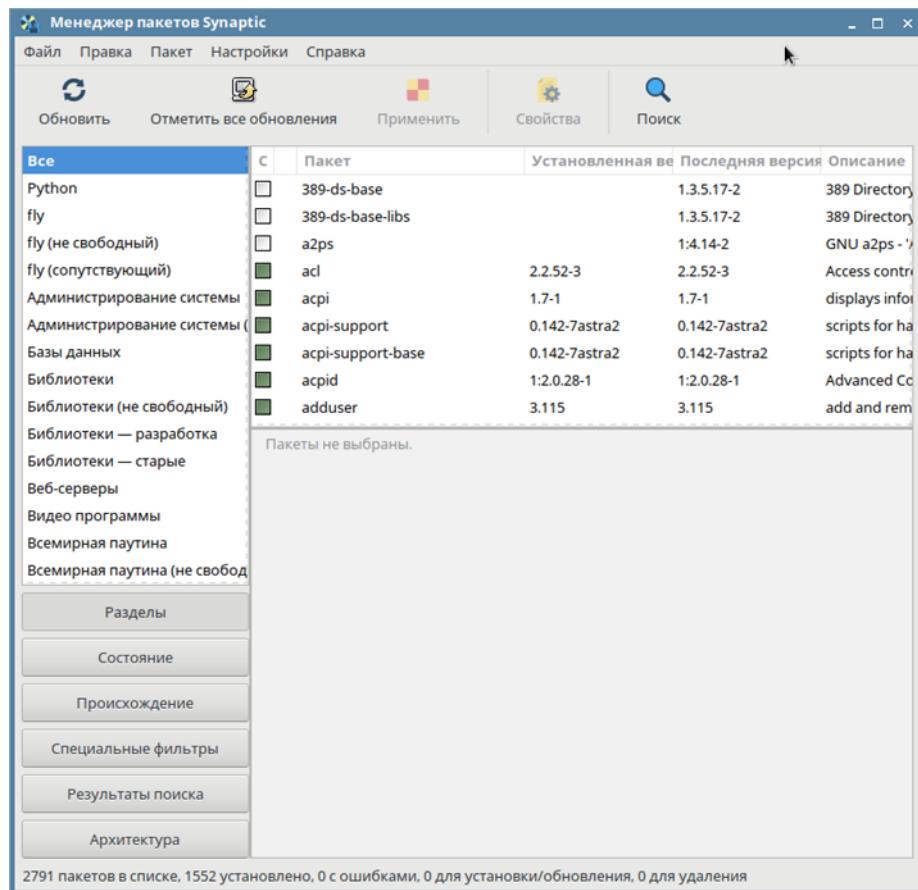


Рисунок 19 – Окно менеджера пакетов

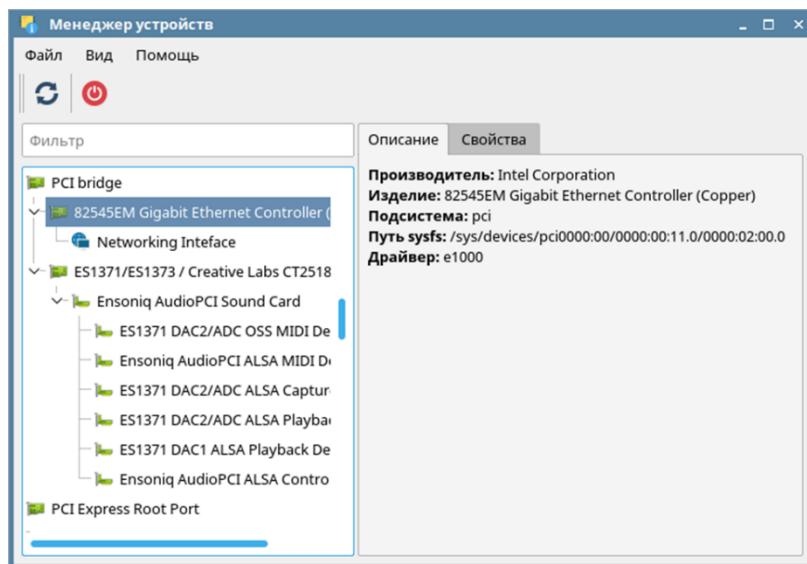


Рисунок 20 – Менеджер устройств

Программа Gparted (GNOME Partition Editor) предназначенный для создания, организации и удаления разделов диска.

Для запуска программы «Gparted»: [Системные] → [Редактор разделов Gparted].

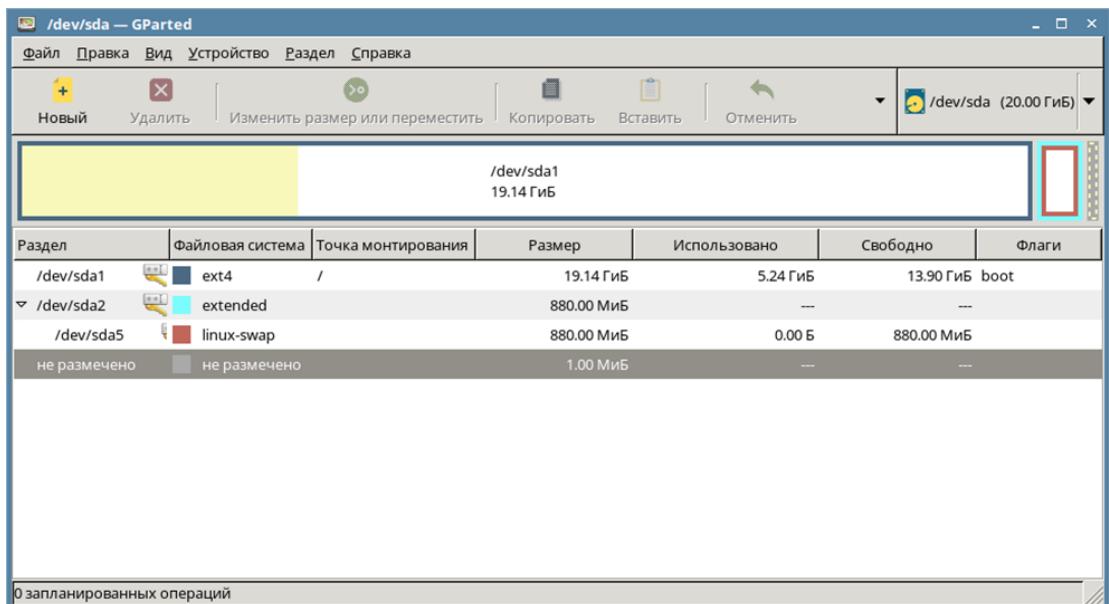


Рисунок 21 – Утилита Gparted

Программа «Системный монитор» (**ksysguard**) позволяет просматривать общую информацию о системе, процессах, точках мониторинга, а также сетевую статистику

Для запуска программы «Системный монитор»: [Системные] → [Системный монитор].

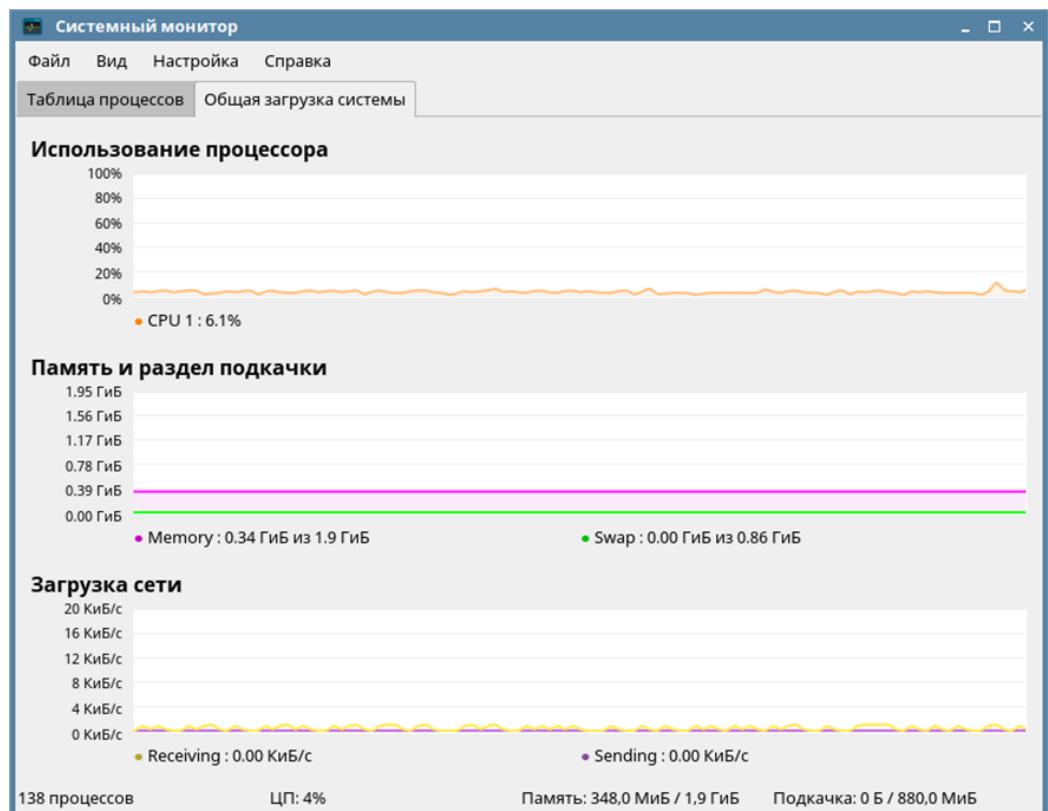


Рисунок 22 – Системный монитор

Программа «Терминал Fly» (**fly-term**) представляет собой эмулятор терминала с расширенными возможностями: многооконным режимом, возможностью прокрутки экрана, редактируемым

Для запуска программы «Терминал Fly»: [Системные] → [Терминал Fly].

## 6 Работа с носителями информации

В качестве файловых систем на носителях информации компьютеров с ОС (в том числе съемных машинных носителях информации) должны использоваться только файловые системы, поддерживающие расширенные (в т.ч. мандатные) атрибуты пользователей и обеспечивающие гарантированное уничтожение (стирание) информации. Рекомендуемые файловые системы: Ext2/Ext3/Ext4.

Перед использованием, любое устройство с файловой системой должно быть смонтировано (Mount) в системе.

Подключить (смонтировать) накопитель можно:

- щелкнув по всплывающему окошку в правой части панели задач
- с помощью менеджера файлов
- с помощью командной строки.

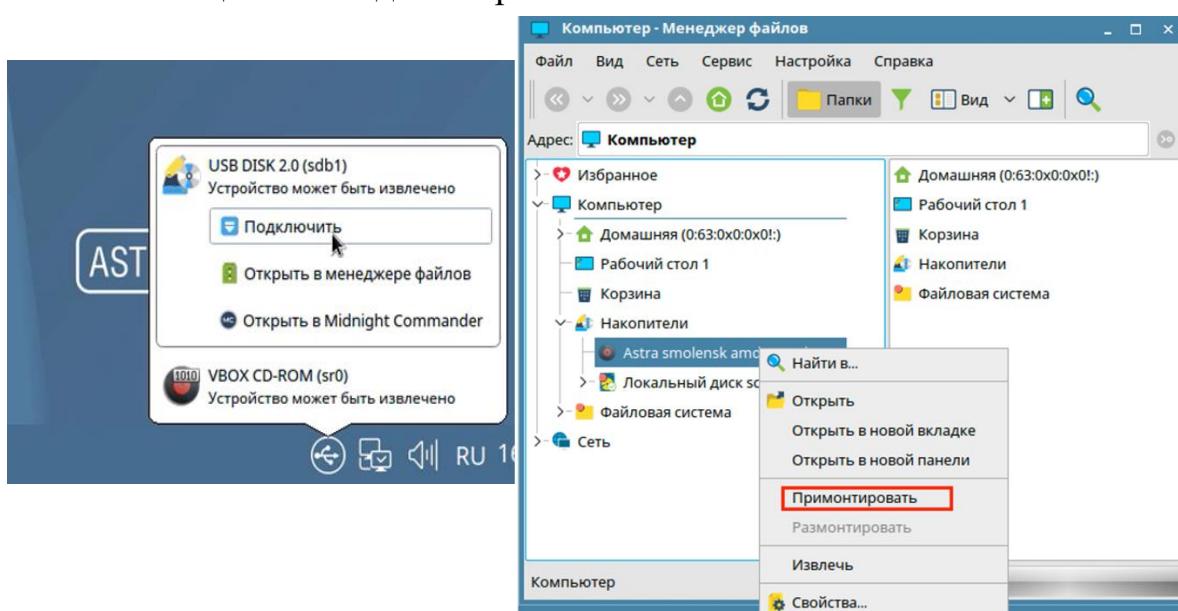


Рисунок 23 - Монтируемые съемные накопители

Для монтирования CD/DVD в командной строке воспользуйтесь следующей командой:

```
$ mount /media/cdrom /mnt/
```

Для безопасного завершения работы рекомендуется перед завершением работы съемный носитель отключить, или, что тоже самое, размонтировать. Сделать это можно несколькими способами.

Подключить (смонтировать) накопитель можно:

- щелкнув в системном трее по значку USB-накопителя;
- с помощью менеджера файлов;
- с помощью командной строки.

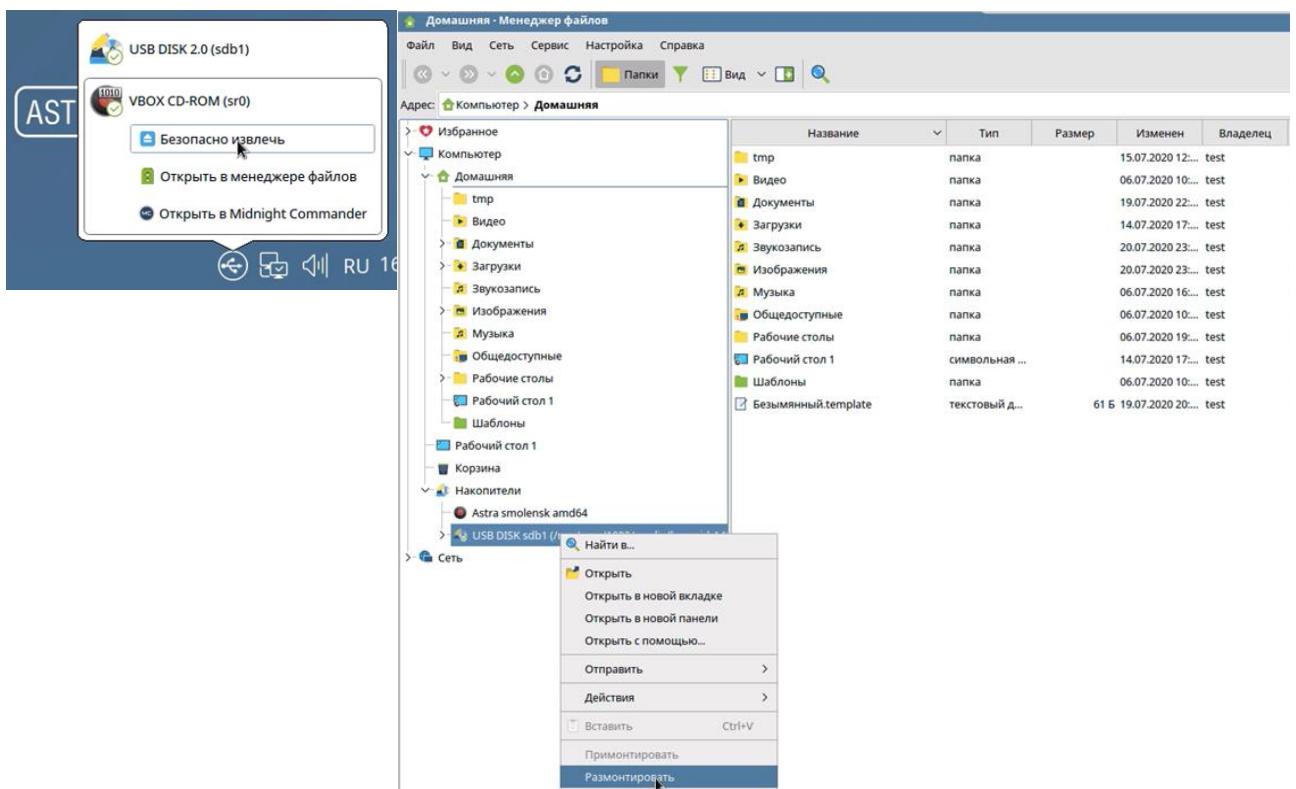


Рисунок 24 - Размонтиране съемных накопителей

## 7 Работа с офисными приложениями

LibreOffice – бесплатно доступный полностью функциональный офисный пакет, который использует нативный формат файла Open Document

Format (ODF). LibreOffice также может открывать и сохранять документы во множестве прочих форматов, в том числе в форматах, используемых различными версиями Microsoft Office. LibreOffice состоит из следующих компонентов:

- текстовый редактор Writer,
- табличный редактор Calc,
- средство создания и демонстрации презентаций Impress,
- редактор векторной графики Draw,
- редактор формул Math,
- система управления базами данных Base.

**Текстовый редактор Writer.** Writer – это инструмент, снабженный богатым набором функций для создания писем, книг, отчетов, новостей, брошюр и прочих документов. Текстовый редактор позволяет вставлять графические и прочие объекты из других компонентов LibreOffice внутрь документа. Writer позволяет экспортить файлы в формате HTML XHTML XML Adobe Portable Document Format (ODF) и множества версий Microsoft Word. Также присутствует возможность подключения к почтовому клиенту.

**Табличный редактор Calc.** Calc обладает всеми необходимыми возможностями для ведения расширенной аналитики, создания диаграмм, поддержки принятия решений и множеством других функций, характерных для высокоуровневых электронных таблиц. В него включено более 300 функций для выполнения финансовых, статистических, математических и многих других операций. Также он содержит диспетчер сценариев для анализа «Что, если». Calc позволяет создавать 2D и 3D диаграммы, которые могут быть интегрированы в другие документы LibreOffice. Также он позволяет открывать и работать с книгами Microsoft Excel и сохранять книги в формате Excel, а также также позволяет экспортить рабочие листы в различных форматах, в том числе Comma Separated Value (CSV), PDF и HTML.

**Средство создания и демонстрации презентаций Impress.** Impress предоставляет все основные инструменты для предоставления мультимедиа

презентаций, такие как специальные эффекты, анимация и инструменты рисования. Он интегрирован с расширенными графическими возможностями компонентов LibreOffice: Draw и Math . Слайд-шоу может быть улучшено при помощи специальных эффектов для текста, а также аудио и видеоклипов. Impress совместим с форматом файлов Microsoft PowerPoint, а также позволяет сохранять и работать в различных графических форматах.

**Редактор векторной графики Draw.** Draw – это инструмент векторной графики который предоставляет всё необходимое от простых диаграмм или графиков до художественных 3D работ. Его возможность умных подключений (Smart Connectors) позволяет определять собственные точки подключения. Draw можно использовать для создания рисунков, а также использовать его совместно с другими компонентами LibreOffice и создавать собственные иллюстрации, а затем добавлять их в галерею. Draw может импортировать графику из множества популярных форматов и сохранять более чем в 20, в том числе PNG, HTML, PDF и Flash.

**Редактор формул Math.** Math – это редактор формул или выражений. Его можно использовать для создания комплексных выражений, которые включают символы и буквы недоступные в стандартном наборе шрифтов. В большинстве случаев Math используется для создания формул в прочих документах LibreOffice, таких как документы Writer и Impress. Math также может работать как независимый инструмент, можно сохранять формулы в стандартном формате Mathematical Markup Language (MathML) для включения на веб-страницы и прочие документы, не созданные при помощи LibreOffice.

**Система управления базами данных Base.** Base – это инструмент для ежедневных работ с базой данных с простым интерфейсом. Он может создавать и редактировать формы, отчеты, запросы, представления и связи, а также управлять базой данных также, как и другие популярные приложения для работы с базами данных. Base предоставляет множество новых возможностей, таких как анализ и редактирование связей в представлении диаграммы. В Base встроена реляционная база данных HSQLDB и PostgreSQL.

Он также может использовать DBase, Microsoft Access, MySQL, Oracle или любую другую совместимую с ODBC или JDBC базу данных. Base также предоставляет поддержку для стандарта ANSI-92 SQL.

### **Методика выполнения**

1. Выполнить поэтапную установку Astra Linux SE с **ручной разметкой накопителя!** (уметь рассказать процесс) согласно п. 2 Теоретических сведений.
2. Рассмотреть и произвести дополнительные настройки (рассказать о них преподавателю).
3. Установить дополнения.
4. Включить мандатный контроль целостности на системные каталоги.
5. Установить и запустить утилиту GParted (сделать скриншот).
6. Осуществить вход в систему (рассказать типы сессий, меню, все уровни конфиденциальности и целостности, категории)
7. Выполнить пользовательские настройки интерфейса работы.
8. Показать навыки монтирования/размонтирования съемных носителей.
9. Ознакомиться с работой LibreOffice, создать и оформить документы приложений LibreOffice.

### **Контрольные вопросы**

1. Назначение ОС AstraLinux SE.
2. Сравнительная характеристика AstraLinux SE и AstraLinux CE.
3. Виды защищаемой информации AstraLinux SE.
4. Какой раздел необходимо создать обязательно при EFI-установке?
5. В какую группу будет включен созданный при установке

пользователь?

6. Какое действие следует сделать после установки для включения мандатного контроля целостности на системные каталоги?

7. Что означает цвет Рабочего стола AstraLinux SE?

8. Какие в AstraLinux SE уровни конфиденциальности и целостности, категории?

9. Как открыть новую сессию не используя режим окна?

10. Как вернуться в одну из предыдущих запущенных сессий с помощью элементов интерфейса?

11. Как вернуться в одну из предыдущих запущенных сессий, используя клавиатурные сокращения?

# Лабораторная работа № 2

## Конвейеры и перенаправление ввода-вывода, архивирование и сжатие

**Цель работы** – научиться работать с командами поиска и архивами. Рассмотреть практические примеры применения конвейеров и использования перенаправления ввода-вывода. Научиться работать с архивами. Изучить утилиты для сжатия и архивирования файлов. Научиться применять данные утилиты совместно с другими командами.

### Теоретические сведения

#### 1 Поиск файлов



Рисунок 1 - Параметры поиска файлов

Процессы работают с файлами с использованием дескрипторов файлов. При этом три специальных файла открыты для любого процесса. Это потоки ввода-вывода (I/O stream):

- stdin (0) – стандартный поток ввода;
- stdout (1) – стандартный поток вывода;
- stderr (2) – стандартный поток ошибок;

где в скобках указан номер дескриптора стандартного потока:

0 (stdin) – входные данные;

1 (stdout) - выходные данные;

2 (stderr) – ошибки.

> - перезапись

>> - добавление в конец

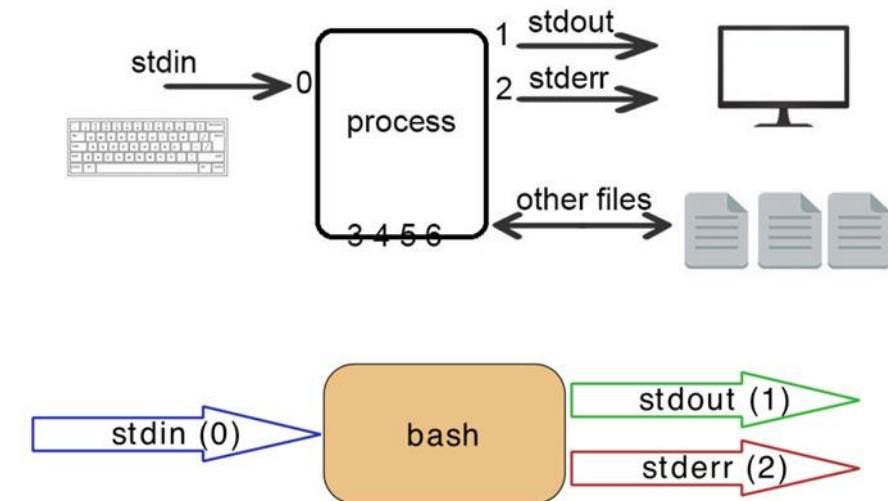


Рисунок 2 – Схема перенаправления стандартных потоков в файл или из файла

Перенаправления:

- подменить ввод клавиатуры файлом

< или 0< имя\_файла

- подменить вывод на экран выводом в файл (файл будет перезаписан)

> или 1> имя\_файла

- Подменить вывод ошибок на экран выводом в файл

2> имя\_файла

- Перенаправить вывод в конец файла

>> имя\_файла

- объединить вывод stdout и stderr и направить в файл

>& имя\_файла или > имя\_файла 2>&1

## **2 Использование состыкованных команд (конвейер)**

Конвейер (pipe) передает вывод предыдущей команды на ввод следующей или на вход командного интерпретатора. Метод часто используется для связывания последовательности команд в единую цепочку. Конвейер обозначается следующим символом: |.

```
$ cat filename | grep something
```

```
$ ls /etc | more
```

## **3 Архивирование и сжатие файлов**

Команды архива : GZIP, BZIP2, TAR

Сжатие: LZMA, XZ- новые и высокоэффективные алгоритмы сжатия, XZ обратно совместим с Lzma.

### **Методика выполнения**

Задания выполняются в терминальном режиме

1. Найдите строку, содержащую слово student в файле /etc/group
2. Отобразите все строки, содержащие слово root из файлов /etc/passwd и /etc/group
3. Подсчитайте количество строк, в которых содержится пользователь student(любой, кроме root) в файле /etc/group
4. Подсчитайте количество строк, в которых НЕ содержится пользователь student(любой, кроме root) в файле /etc/group
5. Используя команду grep, отобразите все строки, содержащие слово List или слово System в файле /etc/passwd
6. Найдите в домашней директории пользователя student все файлы и директории, имя которых заканчивается на 1
7. Найдите в директории /usr все файлы и директории, имя которых

заканчивается на ln

8. Найдите в домашней директории пользователя student(любого, кроме root) только файлы

9. Найдите в домашней директории пользователя student(любого, кроме root) только директории

10. У кого из пользователей в качестве оболочки используется bash

11. Найдите все файлы, принадлежащие пользователю student

12. Найдите все файлы, принадлежащие любой другой группе, например ftp

13. Найдите все файлы несуществующих пользователей или групп.

14. Найдите все файлы, являющиеся символическими ссылками в директории /etc

15. Найдите все файлы в системе размером более 200Мб

16. Найдите все файлы с расширением .html в директории /usr/share/doc

17. Найдите в директории /tmp файлы нулевого размера, а затем удалите не нужные из них.

18. Используйте команду find, чтобы определить местонахождение всех файлов, принадлежащих пользователю root в домашней директории пользователя student, а затем создайте архив с помощью команды tar. Архив должен находиться в каталоге /tmp

19. Добавьте содержимое каталога /etc в архив в директорию /tmp. Посмотрите размер полученных файлов (сделать скриншот).

20. Используйте gzip для сжатия ваших архивов и снова посмотрите получившийся размер файлов.

21. Отмените сжатие существующих файлов и сожмите их снова с помощью утилиты bzip2, и снова посмотрите размер (сделать скриншот).

## **Контрольные вопросы**

Сделать выводы о механизмах архивирования и сжатия.

# **Лабораторная работа № 3**

## **Управление процессами и планирование**

**Цель работы** – научиться управлять процессами ОС, передавать сигналы процессам. Разобраться в текущем состоянии ОС, насколько система загружена, какие процессы имеют приоритеты. Научиться управлять приоритетами процессов. Рассмотреть разные возможности планирования задач. Научится планировать задания, требующие периодического выполнения (например, создание бэкапа системы), а также задачи, которые запускаются только один раз в заранее установленное время.

### **Теоретические сведения**

Абсолютный и относительный приоритеты. В системах с относительными приоритетами активный процесс выполняется до тех пор, пока он сам не покинет процессор, перейдя в состояние Ожидание (или же произойдет ошибка, или процесс завершится). В системах с абсолютными приоритетами выполнение активного процесса прерывается еще при одном условии: если в очереди готовых процессов появился процесс, приоритет которого выше приоритета активного процесса. В этом случае прерванный процесс переходит в состояние Готовности, см. Лекции Блок2.

### **1 Основные характеристики процессов и управление процессами**

Загрузка Linux завершается тем, что на всех виртуальных консолях (на самом деле - на всех терминалах системы), предназначенных для работы пользователей, запускается программа getty. Программа выводит приглашение и ожидает активности пользователя, который может захотеть работать именно на этом терминале. Введённое входное имя getty передаёт программе login, которая вводит пароль и определяет, разрешено ли работать

в системе с этим входным именем и этим паролем. Если `login` приходит к выводу, что работать можно, он запускает стартовый командный интерпретатор, посредством которого пользователь и командует системой.

Выполняющаяся программа называется в Linux процессом. Все процессы система регистрирует в таблице процессов, присваивая каждому уникальный номер — идентификатор процесса (process identifier, PID). Манипулируя процессами, система имеет дело именно с их идентификаторами, другого способа отличить один процесс от другого, по большому счёту, нет. Для просмотра своих процессов можно воспользоваться утилитой `ps` («process status»):

```
root@astra:~# ps -f
UID      PID  PPID  C STIME TTY          TIME CMD
root     1138  1132  0 мар24 pts/0  00:00:00 sudo su -
root     1139  1138  0 мар24 pts/0  00:00:00 su -
root     1143  1139  0 мар24 pts/0  00:00:00 -su
root    2351  1143  0 03:51 pts/0    00:00:00 ps -f
root@astra:~#
```

Ключ «`-f`» («full»), для того чтобы добыть побольше информации. В поле `PPID`

(«parent process identifier») указан идентификатор родительского процесса, т. е. процесса, породившего данный. В выдаче не оказалось строки для этого `login`, равно как и для большинства других процессов системы, так как они не принадлежат текущему пользователю.

Процесс - выполняющаяся программа в Linux. Каждый процесс имеет уникальный идентификатор процесса, PID. Процессы получают доступ к ресурсам системы (оперативной памяти, файлам, внешним устройствам и т. п.) и могут изменять их содержимое. Доступ регулируется с помощью идентификатора пользователя и идентификатора группы, которые система присваивает каждому процессу.

Запуск одного процесса вместо другого устроен в Linux с помощью системного вызова `exec()`. Старый процесс из памяти удаляется навсегда, вместо него загружается новый, при этом настройка окружения не меняется, даже PID остаётся прежним. Вернуться к выполнению старого процесса невозможно, разве что запустить его по новой с помощью того же `exec()` (от

«execute» — «исполнить»).

Для работы командного интерпретатора недостаточно одного exec(). В самом деле, shell не просто запускает утилиту, а дожидается её завершения, обрабатывает результаты её работы и продолжает диалог с пользователем. Для этого в Linux служит системный вызов fork() («вилка, раздилка»), применение которого приводит к возникновению ещё одного, дочернего, процесса — точной копии породившего его родительского. Дочерний процесс ничем не отличается от родительского: имеет такое же окружение, те же стандартный ввод и стандартный вывод, одинаковое содержимое памяти и продолжает работу с той же самой точки (возврат из fork()). Отличия два: во-первых, эти процессы имеют разные PID, под которыми они зарегистрированы в таблице процессов, а во-вторых, различается возвращаемое значение fork(): родительский процесс получает в качестве результата fork() идентификатор процесса-потомка, а процесс-потомок получает «0».

Дальнейшие действия shell при запуске какой-либо программы очевидны. Shell-потомок немедленно вызывает эту программу с помощью exec(), а shell-родитель дожидается завершения работы процесса-потомка (PID которого ему известен) с помощью ещё одного системного вызова, wait(). Дождавшись и проанализировав результат команды, shell продолжает работу:

```
student@astra:~$ cat > loop
```

```
while true; do true; done
```

```
^D
```

```
student@astra:~$ sh loop
```

```
^C
```

```
student@astra:~$
```

Был создан сценарий для sh (или bash, на таком уровне их команды совпадают), который ничего не делает. Точнее этот сценарий делает ничего, бесконечно повторяя в цикле команду, вся работа которой состоит в том, что она завершается без ошибок («> файл» в командной строке просто перенаправляет стандартный вывод команды в файл). Запустив этот сценарий

с помощью команды вида `sh имя_сценария`, мы ничего не увидели, но услышали, как загудел вентилятор охлаждения центрального процессора. Управляющий символ «`^C`», как обычно, привёл к завершению активного процесса, и командный интерпретатор продолжил работу. Если бы в описанной выше ситуации родительский процесс не ждал, пока дочерний завершится, а сразу продолжал работать, получилось бы, что оба процесса выполняются «параллельно»: пока запущенный процесс что-то делает, пользователь продолжает командовать оболочкой. Для того, чтобы запустить процесс параллельно, в shell достаточно добавить «`&`» в конец командной строки:

```
root@astra:~# ps -f
UID      PID  PPID  C STIME TTY          TIME CMD
root     1138  1132  0 мар24 pts/0  00:00:00 sudo su -
root     1139  1138  0 мар24 pts/0  00:00:00 su -
root     1143  1139  0 мар24 pts/0  00:00:00 -su
root    2933  1143  98 08:30 pts/0      00:00:11 sh loop
root    2934  1143  0 08:31 pts/0      00:00:00 ps -f
root@astra:~# ■
```

Процесс, запускаемый параллельно, называется фоновым (background). Фоновые процессы не имеют возможности вводить данные с того же терминала, что и породивший их shell (только из файла), зато выводить на это терминал могут (правда, когда на одном и том же терминале вперемежку появляются сообщения от нескольких фоновых процессов, начинается сущая неразбериха). При каждом терминале в каждый момент времени может быть не больше одного активного (foreground) процесса, которому разрешено с этого терминала вводить. На время, пока команда (например, `cat`) работает в активном режиме, породивший её командный интерпретатор «ходит в фон», и там, в фоне, выполняет свой `wait()`.

**Активный процесс** - процесс, имеющий возможность вводить данные с терминала.

В каждый момент у каждого терминала может быть не более одного активного процеса.

**Фоновый процесс** - процесс, не имеющий возможность вводить данные с терминала.

Пользователь может запустить любое, не превосходящее заранее

заданного в системе, число фоновых процессов.

Далеко не всем процессам, зарегистрированным в системе необходимо давать поработать наравне с другими. Большинству процессов работать прямо сейчас не нужно: они ожидают какого-нибудь события, которое им нужно обработать. Чаще всего процессы ждут завершения операции ввода-вывода. Чтобы посмотреть, как потребляются ресурсы системы, можно использовать утилиту top.

Утилита top позволяет отобразить наиболее активные процессы (столько, сколько их помещается на экран) с достаточно полной информацией о них (для пользователя утилиты представляет ограниченный набор выводимых параметров).

**Пример.** Запустить бесконечный сценарий: посмотрим, как два процесса конкурируют за ресурсы между собой:

```
student@astra:~$ bash loop&
```

```
student@astra:~$ top
```

```
top - 08:34:22 up 12:18, 2 users, load average: 1,67, 0,74, 0,29
Tasks: 104 total, 3 running, 71 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99,6 us, 0,4 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 2041532 total, 751156 free, 292804 used, 997572 buff/cache
KiB Swap: 2095100 total, 2095100 free, 0 used, 1581732 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2933	root	20	0	11180	2840	2616	R	49,5	0,1	2:56.11	sh
2935	root	20	0	11180	2996	2780	R	49,2	0,1	0:34.85	bash
616	fly-dm	20	0	487540	141700	50516	S	1,0	6,9	0:08.51	Xorg
1017	student	20	0	592308	50212	43680	S	0,3	2,5	0:00.35	fly-start-panel
1128	student	20	0	575428	61220	44256	S	0,3	3,0	0:08.24	fly-term
2920	root	20	0	0	0	0	I	0,3	0,0	0:00.04	kworker/u2:1
2936	root	20	0	45096	3804	3196	R	0,3	0,2	0:00.08	top

Оказалось, что конкурируют даже не два процесса, а три: sh (первый из запущенных интерпретаторов loop), bash (второй) и сам top. Правда, по сведениям из поля %CPU, львиную долю процессорного времени отобрали sh и bash, а top довольствуется десятой долей процента (а то и меньшей: ошибки округления). Без знания архитектуры Linux большая часть информации не имеет смысла.

Впрочем, некоторая часть всё же понятна: объём оперативной памяти (всей, используемой и свободной), время работы машины, объём памяти, занимаемой процессами и т. п.

Последний процесс, запущенный из оболочки в фоне, можно из этой оболочки сделать активным при помощи команды `fg` («foreground» — «передний план»).

```
[student@astra:~$ fg  
bash loop  
^C
```

**Сигнал** — это способность процессов обмениваться стандартными короткими сообщениями непосредственно с помощью системы. Сообщение-сигнал не содержит никакой информации, кроме номера сигнала (для удобства вместо номера можно использовать предопределённое системой имя). Для того, чтобы передать сигнал, процессу достаточно задействовать системный вызов `kill()`, а для того, чтобы принять сигнал, не нужно ничего. Если процессу нужно как-то по-особенному реагировать на сигнал, он может зарегистрировать обработчик, а если обработчика нет, за него отреагирует система. Как правило, это приводит к немедленному завершению процесса, получившего сигнал.

Обработчик сигнала запускается асинхронно, немедленно после получения сигнала, что бы процесс в это время ни делал.

**Сигнал** - короткое сообщение, посыпаемое системой или процессом другому процессу. Обрабатывается асинхронно специальной подпрограммой-обработчиком. Если процесс не обрабатывает сигнал самостоятельно, это делает система. Два сигнала — 9 (KILL) и 19 (STOP) — всегда обрабатывает система. Первый из них нужен для того, чтобы убить процесс наверняка (отсюда и название). Сигнал STOP приостанавливает процесс: в таком состоянии процесс не удаляется из таблицы процессов, но и не выполняется до тех пор, пока не получит сигнал 18 (CONT) — после чего продолжит работу. В Linux сигнал STOP можно передать активному процессу с помощью управляющего символа «`^Z`»:

```
student@astra:~$ sh loop  
^Z
```

```
[1]+ Stopped sh loop
```

```
student@astra:~$ bg
```

```
[1]+ sh loop &
```

```
student@astra:~$ fg
```

```
sh loop
```

```
^C
```

```
student@astra:~$
```

Сначала запустили вечный цикл в качестве активного процесса, затем передали ему сигнал STOP с помощью «^Z», после чего дал команду bg (background), запускающую в фоне последний остановленный процесс. Затем student снова перевёл этот процесс в активный режим, и, наконец, убил его.

Передавать сигналы из командной строки можно любым процессам с помощью команды kill -сигнал PID или просто kill PID, которая передаёт сигнал 15 (TERM).

Можно попробовать запустить несколько процессов, а потом выборочно поубивать их. Для этого он, вдобавок к уже висящему в фоне sh loop, запустил в качестве активного процесса новый командный интерпретатор, sh (при этом изменилась приглашение командной строки). Из этого sh он запустил в фоне ещё один sh loop и новый bash loop.

Сделал он это одной командной строкой (при этом команды разделяются символом «&», т. е. «И»; выходит так, что запускается и та, и другая команда). В ps он использовал новый ключ — «-H» («Hierarchy», «иерархия»), который добавляет в выдачу ps отступы, показывающие отношения «родитель–потомок» между процессами

```
root@astra:~# kill 2943
root@astra:~# ps -fH
UID      PID  PPID  C STIME TTY          TIME CMD
root    1138  1132  0 mar24 pts/0  00:00:00 sudo su -
root    1139  1138  0 mar24 pts/0  00:00:00   su -
root    1143  1139  0 mar24 pts/0  00:00:00   -su
root    2933  1143  56 08:30 pts/0   00:08:41     sh loop
root    2935  1143  48 08:33 pts/0   00:06:20     bash loop
root    2942  1143  26 08:45 pts/0   00:00:11     bash loop
root    2946  1143  0 08:46 pts/0   00:00:00   ps -fH
[4]+  Завершено                          bash loop
root@astra:~#
```

Например, остановим работу давно запущенного sh, выполнявшего сценарий с вечным циклом (PID 2943).

## 2 Планирование задач с помощью cron

Cron это программа, выполняющая задания по расписанию. Позволяет неоднократный запуск заданий. Т.е. задание можно запустить в определенное время или через определенный промежуток времени. В системах Linux многие задачи планируются для регулярного запуска:

- ротация журналов;
- обновление базы данных для программы locate;
- резервное копирование;
- сценарии обслуживания (такие как удаление временных файлов).

Cron — это демон, отвечающий за запуск запланированных и повторяющихся команд (каждый день, каждую неделю и т. д.); atd — демон, работающий с командами, которые должны запускаться однократно, но в конкретный момент времени в будущем.

Служба Cron является базовой и запускается через systemd  
systemctl status cron

При загрузке системы, запускается демон cron и проверяет очередь заданий at и заданий пользователей в файлах crontab. При запуске, демон cron сначала проверяет каталог /var/spool/cron на наличие файлов crontab, файлы crontab имеют имена пользователей, соответствующие именам пользователей из /etc/passwd Каждый пользователь может иметь только один файл crontab, записей в файле может быть несколько.

Другими словами - файлы crontab содержат инструкции для демона cron, который запустит задание(я) описаное в файле crontab. Все файлы crontab из каталога /var/spool/cron загружаются в память, одновременно с ними загружаются файлы из /etc/cron.d После этого демон cron загружает содержимое файла /etc/crontab При стандартных настройках, содержимое

/etc/crontab имеет следующий вид:

```
SHELL=/bin/bash  
PATH=/sbin:/bin:/usr/sbin:/usr/bin  
MAILTO=root  
HOME=/  
# run-parts  
01 * * * * root run-parts /etc/cron.hourly  
02 4 * * * root run-parts /etc/cron.daily  
22 4 * * 0 root run-parts /etc/cron.weekly  
42 4 1 * * root run-parts /etc/cron.monthly
```

Информация файла указывает, что:

- содержимое каталога /etc/cron.hourly будет запускаться каждый час на первой минуте часа.
- содержимое каталога /etc/cron.daily будет запускаться каждый день на второй минуте четвертого часа.
- содержимое каталога /etc/cron.weekly будет запускаться каждое воскресенье на 22'ой минуте 4'го часа.
- содержимое каталога /etc/cron.monthly будет запускаться каждый первый день месяца на 42'ой минуте 4'го часа.

SHELL=/bin/bash означает использовать для запуска команд /bin/bash , если переменная не указана, то значение будет взято из /etc/passwd для пользователя являющимся владельцем файла.

HOME=/ корневой каталог для пользователя (параметр не обязательный) При необходимости доступа к специальным свойствам интерпретатора, значения переменных SHELL и HOME можно изменить, не зависимо от того, что прописано в /etc/passwd

MAILTO=root означает кому отсылать сообщение о результате работы команд.

Все содержимое из этих каталогов будет запускаться с правами доступа

пользователя root и файлы должны иметь права доступа на "выполнение", поэтому перед размещением файлов в одном из этих каталогов необходимо убедиться, что сценарии не нанесут вред системе.

После того, как демон cron запущен и прочел содержимое всех файлов crontab, он бездействует, просыпаясь каждую минуту и проверяя не требуется ли запуск какой-либо команды в данную минуту, или не появился ли новый файл crontab который необходимо обработать. Демон cron определяет изменения по времени модификации файлов или каталогов, такое его свойство избавляет от необходимости перезапуска демона.

Как отмечалось выше, размещение файлов для cron в каталогах

/etc/cron.hourly

/etc/cron.daily

/etc/cron.weekly

/etc/cron.monthly

доступно только пользователю root, для использования файлов crontab пользователями, нужно использовать команду crontab. Команда служит для создания, изменения и добавления файла для демона cron

Доступ в каталог /var/spool/cron непривилегированному пользователю закрыт, что бы посмотреть под пользователем есть ли у него файл crontab, достаточно набрать команду crontab -l , если файл существует-будет показано его содержимое.

Для удаления файла используется команда crontab -r. Для редактирования

crontab -e

Для управления файлами crontab пользователем "root" используется синтаксис:

crontab -u user\_name file - создание файла crontab из файла "file" для пользователя "user\_name"

-u означает чей crontab будет обработан, Если опция не задана, то будет обработан crontab того пользователя, который запустил команду crontab.

crontab -u user\_name -l - просмотр файла crontab пользователя "user\_name"

crontab -u user\_name -r - удаление файла crontab пользователя "user\_name"

crontab -u user\_name -e - редактирование файла crontab пользователя "user\_name" используя редактор, заданный переменной окружения VISUAL или EDITOR

Каждая команда в пользовательском файле crontab занимает одну строку и состоит из шести полей. Пользовательские файлы crontab находятся в каталоге /var/spool/cron

Общий формат команды: минута час день\_месяца месяц день\_недели команда

Допустимые значения:

- минута: от 0 до 59
- час: от 0 до 23
- день\_месяца: от 1 до 31
- месяц: от 1 до 12 (можно три буквы из названия месяца, регистр не имеет

значения от jan до dec)

- день\_недели: от 0 до 6 (0 это воскресенье, можно писать от sun до sat)

Каждое из полей даты и времени может быть обозначено символом\*, будет соответствовать любому возможному значению. Для этих полей можно указывать диапазоны значений, разделенных дефисом, например:

\* 5 4-10 0-3 \* echo "HELLO" -печатать HELLO в 5:00 на 4,5,6,7,8,9,10 дни января,

февраля, марта и апреля, а пошаговая запись

\* \*/2 \* \* sat echo "HELLO" -печатать HELLO каждый четный час, каждую субботу

Равнозначная предыдущему примеру запись (списком)

\* 0,2,4,6,8,10,12,14,16,18,20,22 \* \* sat echo "HELLO" -печатать HELLO

каждый четный час, каждую субботу. То же самое с указанием диапазона

\* 0-23/2 \* \* sat echo "HELLO" -печать HELLO каждый четный час, каждую

субботу

59 23 31 dec \* echo "Happy new year" - поздравит с новым годом

Для отладки задания cron, можно перенаправить результат в файл

**Пример:**

0-59 \* \* \* \* /home/user/mail 2>/tmp/tmp.cron

Если при запуске команды /home/user/mail возникнут ошибки, то они будут записаны в файл /tmp/tmp.cron и вы всегда сможете узнать причину. В случае перенаправления вывода в файл, письмо, юзеру указанному в переменной MAILTO отправлено не будет.

Посмотреть информацию о всех командах запускаемых демоном cron можно в каталоге /var/log называются cron, cron1 и т.д.

В файле /var/log/cron записано время запуска всех заданий cron за предыдущий

день

Mar 29 04:03:00 rst CROND[4434]: (user) CMD (/home/user/mail)

Mar 29 04:03:59 rst CROND[4493]: (user) CMD (/home/user/mail)

Mar 29 04:05:00 rst CROND[4507]: (user) CMD (/home/user/mail)

Mar 29 04:06:00 rst CROND[4549]: (user) CMD (/home/user/mail)

В остальных файлах cron1,cron2 находится подобная информация, но более старая чем в cron.

### **3 Планирование задач с помощью утилиты at**

Использование команды at. at запускает команду в заданный момент времени в будущем. Она принимает желаемые время и дату как параметры командной строки, а команду, которую необходимо запустить, через

стандартный ввод. Команда будет запущена, как если бы она была введена в текущей оболочке. at беспокоится даже о том, чтобы сохранить текущее окружение, чтобы воспроизвести те же условия при вызове команды.

Время указывается в соответствии со следующими соглашениями: 16:12 или 4:12pm означают 4:12 после полудня. Дата может быть задана в разных европейских и западных форматах, в том числе ДД.ММ.ГГ (27.07.21 означает 27 июля 2021 года), ГГГГ-ММ-ДД (та же дата выражается как 2021-07-27), ММ/ДД/[ГГ]ГГ (то есть 12/25/21 или 12/25/2021 будут 25 декабря 2021 года), или просто ММДД[ГГ]ГГ (так что 122512 или 12252012 будет, точно так же, соответствовать 25 декабря 2021 года). Без неё команда будет запущена, как только часы подойдут к указанному времени (в тот же день или на следующий день, если сегодня это время уже прошло). Можно также просто написать «today» или «tomorrow» — сегодня или завтра, соответственно.

```
$ at 09:00 27.07.21 <<END
> echo "Не забудь поздравить Иванова с днём рождения!" \
> | mail user@example.com
> END
warning: commands will be executed using /bin/sh
job 31 at Fri Jul 27 09:00:00 2021
```

Альтернативный синтаксис откладывает запуск на заданный промежуток времени: at now + число период. Значение период может быть в минутах, часах, днях или неделях. Значение число указывает число указанных единиц, которое должно пройти перед запуском программы.

Для отмены задачи, запланированной cron, нужно просто запустить crontab -e и удалить соответствующую строку в файле crontab. Для задач at это почти так же легко: надо запустить atrm номер-задачи. Номер задачи указывается командой at при её планировании, а также её можно найти с помощью команды atq, выводящей текущий список запланированных задач.

**Планирование асинхронных задач:** anacron. Anacron — это демон, дополняющий cron на компьютерах, которые не включены всё время.

Поскольку регулярные задачи обычно планируются на середину ночи, они никогда не будут запускаться, если компьютер в это время выключен. Назначение anacron — запустить их, принимая во внимание периоды, в которые компьютер не работает.

## Методика выполнения

1. Из консоли пользователя командой `ps -efl | more` выведите расширенный поэкранный список исполняемых процессов (перечень параметров для расширенного вывода информации можно уточнить с помощью электронного справочника `man`).

Разберитесь с выводимой информацией и объясните преподавателю. Определите и осуществите вывод процессов согласно выбору преподавателя:

- по типу: системные, демоны, пользовательские (тип процесса определяется по косвенным признакам, в частности, по имени);
- по состоянию S: (исполняющиеся, ожидающие записи на диск, ожидающие событий, приостановленные, зомби);
- по текущему динамическому приоритету PRI (наименьшее значение у высокоприоритетных процессов);
- по относительному приоритету N1.

2. С помощью `Ctrl+Alt+F2` (`Alt+F2`) откройте второй текстовый терминал и зарегистрируйтесь в нем как суперпользователь, запустите утилиту `top` для текущего контроля процессов.

3. Нажатием `Ctrl+Alt-F1` (`Alt-F1`) вернитесь в первую консоль. Из первой консоли создайте процесс `od /dev/zero > /dev/null`. В соответствии с введенной командой утилита `od` читает и выводит непрерывный поток нулевых байт из «рога изобилия» в нулевое устройство. Переключившись в другую консоль, с помощью команды `top` просмотрите список наиболее активных процессов. Найдите и идентифицируйте запущенный процесс, найдите по идентификатору PPID его «родителя», определите его приоритет (возможно

это - величина переменная), долю загрузки центрального процессора %CPU и оперативной памяти %MEM.

4. Поочередно из первой и второй консолей с правами администратора и пользователя с помощью команды od /dev/zero > /dev/null & создайте по 2-3 одинаковых фоновых процесса.

По мере создания новых процессов отслеживайте в третьей консоли их текущий приоритет, загрузку процессора и памяти. Имеются ли различия в приоритете процессов, выполняемых от имени администратора и пользователя?

5. В консоли пользователя измените приоритет одного из принадлежащих ему процессов. Для этого воспользуйтесь командой renice -10 PID. Изменился ли относительный приоритет процесса?

6. Повторите предыдущий пункт с правами администратора и ответьте на вопрос.

7. Переключитесь в консоль пользователя и измените приоритет одного из принадлежащих ему процессов командой renice 5 PID. Произошло ли изменение приоритета?

8. Проконтролируйте из третьей консоли изменение приоритетов запущенных процессов. Что произошло?

9. Удалите созданные процессы командой kill.

10. С правами пользователя создайте в своей директории сценарий с именем abcd. Сценарий можно создать с помощью редактора nano:

```
# nano abcd
#!/bin/bash
while : rem обратите внимание на пробел перед двоеточием!
do
echo HELLO!
done
```

11. Используя команду chmod, присвойте пользователю полные права на чтение, запись и исполнение данного сценария. Запустите сценарий на

исполнение (на экран должны непрерывно выводиться приветствия HELLO!)

12. Перейдите в третью консоль, с помощью команды top просмотрите список процессов и найдите в нем «зависший» процесс, запущенный пользователем (на самом деле это только имитация зависания, которое пользователь легко может прекратить сам). Прочтайте идентификатор процесса PID.

13. Нажатием Ctrl+C из второй консоли остановите процесс. Как изменилось при этом состояние процесса?

14. Повторно запустите из второй консоли процесс, перейдите в первую консоль и отправьте "зависшему" процессу сигнал на останов (команда kill -15 PID\_process).

15. Перейдите в другую консоль и отправьте процессу сигнал kill -20 PID. Как реагирует процесс на данный сигнал? Что означает данный сигнал?

16. С помощью команды kill -9 PID отправьте этому процессу сигнал принудительного завершения. С другой консоли проконтролируйте выполнение команды. Остановился ли процесс? Остался ли он в списке процессов? Какая программа на самом деле перехватывает и исполняет команду kill -9 PID?

17. С помощью команды echo \$PATH поочередно из консоли администратора и пользователя user1 выведите список директорий, в которых производится поиск исполняемых файлов, заданных только по имени. В чем заключается различие выведенных списков? Почему в списке PATH администратора отсутствует текущий каталог (.)? Почему в списке PATH пользователя отсутствует каталог /sbin? Имеет ли пользователь возможность изменить порядок проверки каталогов для администратора?

18. Попробуйте запустить несколько утилит из второй консоли с правами пользователя (например, renice -10 PID, date -s 0). Как реагирует система на ваши попытки? Опишите.

19. Перейдите в административную консоль и повторите запуск утилит с правами суперпользователя. Что произошло?

20. Убедитесь в том, что пользователю разрешен запуск указанных утилит. Объясните, почему пользователь не может запустить утилиты с некоторыми «критичными» параметрами? Где, по вашему мнению, расположен механизм контроля за ходом запуска (в ядре операционной системы, в командной оболочке, в самой утилите?). Ответ обоснуйте.

21. С правами пользователя скопируйте в свой рабочий каталог один из исполняемых файлов с параметром SUID каталога /bin (исполняемые файлы выделены цветом и символом \*, а параметр SUID отмечен символом «s» в правах владельца на исполнение). Как изменились права доступа к файлу после его копирования?

22. Из второй консоли с правами пользователя скопируйте в свой домашний каталог утилиту, которую разрешено запускать только администратору (например, chattr). Копирование производите с параметрами, гарантирующими переход копии во владение пользователю. Убедитесь и покажите, что пользователь имеет на скопированный файл все необходимые права.

Используйте свою копию утилиты по ее назначению (в случае копирования утилиты chattr установите дополнительный атрибут +i одному из своих файлов). Сделайте выводы.

23. Настроить на запуск однократно некоторую задачу в определенное время. Вы также должны наметить другую задачу, которая стартовала бы каждые десять минут (между 8:00 и 17:00). Задачи выбрать самостоятельно.

24. Создать напоминание, например, о встрече с X в 12:00, сегодня.

Войдите в систему как root и введите следующие команды:

at noon; <Нажмите Enter >

echo "Time for meet with X." <Нажмите Enter >

<После завершения нажмите ctrl-D >

Проверьте очередь заданий, используя команду atq.

25. Настройте получение информации о статусе системы каждые десять минут.

Войдя в систему как root, используйте команду crontab -e, чтобы редактировать ваш cron файл.

Введите следующую строку в ваш crontab файл:

```
*/10 8-17 * * * /usr/bin/free >> /root/free.txt
```

Проверьте содержимое файла /root/free.txt, после того как задание отработает

26. Задача:

- убедиться, что каталог, где сохраняются задания at, пуст;
- проверить время и дату в терминале;
- запланировать создание файла в корневом каталоге через две минуты;
- в каталоге, где собираются задания в очереди на выполнение должен появиться новый файл, проверить
  - по команде at -l получим информацию о поставленном в очередь задании (указывается имя пользователя, инициировавшего задание, идентификатор задания, по которому его можно удалить, и время планируемого запуска).

## **Контрольные вопросы**

1. Имеются ли различия в приоритете процессов, выполняемых от имени администратора и пользователя?
2. Как изменяется относительный приоритет процесса?
3. Почему в списке PATH администратора отсутствует текущий каталог (.)?
4. Почему в списке PATH пользователя отсутствует каталог /sbin?
5. Имеет ли пользователь возможность изменить порядок проверки каталогов для администратора?
6. Где расположен механизм контроля за ходом запуска утилит (в ядре операционной системы, в командной оболочке, в самой утилите?). Ответ обоснуйте.

7. Покажите какие основные три процесса наиболее потребляют ресурсы на вашем компьютере?
8. Покажите присутствуют ли на вашей машине процессы-зомби?
9. Назначение Cron и Anacron.
10. Crontab и at применение.
11. С помощью какой команды пользователь может составить расписание для запуска периодических заданий?
12. Какой командой можно увидеть активные таймеры?

## **Лабораторная работа № 4**

### **Управление пользователями и правами**

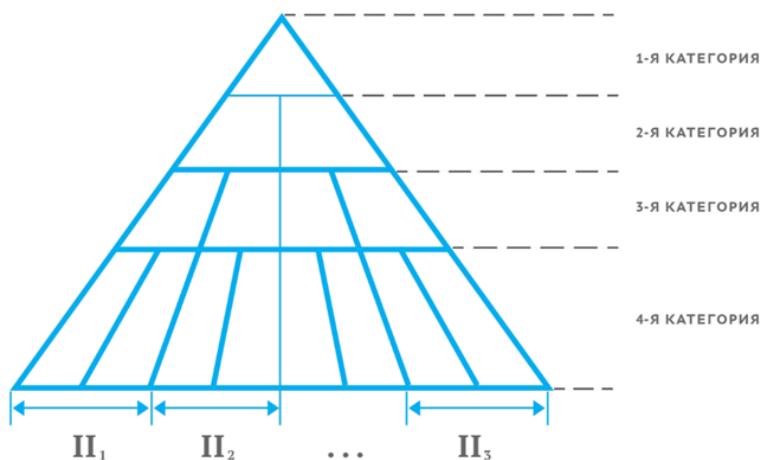
**Цель работы** – научится администрировать учетные записи пользователей: создавать, удалять менять политики доступа. Изучить на практике дискреционные полномочия. Научится работать с правами доступа и расширенными списками ACL на файлы и директории.

#### **Теоретические сведения**

##### **1 Модели безопасности системы**

Уровень секретности (конфиденциальности) – это административная или законодательная мера, соответствующая мере ответственности лица за утечку или потерю конкретной секретной информации, регламентируемой специальным документом, с учетом государственных, военно-стратегических, коммерческих, служебных или частных интересов. Пример:

- Не секретно
- ДСП
- Секретно
- Совершенно секретно



**Рисунок 1 – Уровни секретности**

ОС Astra Linux Special Edition подтвердила сертификат ФСТЭК России на соответствие первому, высочайшему, уровню доверия к средствам технической защиты информации и обеспечения безопасности информационных технологий. Это подтверждает правомерность применения операционной системы для обработки любой информации ограниченного доступа, в том числе государственной тайны до степени секретности «особой важности» включительно.

Метод формальной разработки системы опирается на модель безопасности (модель управления доступом, модель политики безопасности).

Целью модели безопасности - выражение сути требований по безопасности к данной системе. Модель позволяет провести анализ свойств системы, но не накладывает ограничений на реализацию тех или иных механизмов защиты.

Основные модели управления доступом:

- дискреционная
- мандатная
- ролевая.

## **2.1 Модель дискреционного доступа (DAC)**

Избирательный подход, описываемый дискреционной моделью разграничения доступа, является наиболее простым одноуровневым подходом к обеспечению безопасности. Основными его понятиями являются:

- субъект - системный идентификатор, от имени которого ОС выполняет определенные действия над определенными объектами. Понятие субъекта отличается от понятия пользователь компьютерной системы, поскольку инициировать изменение информации могут также и системные процессы;
- объект защиты - часть, на которую распространяется действие конкретного правила безопасности.

Таким образом, в дискреционной модели контролируется доступ

субъектов (пользователей или приложений) к объектам, представляющим собой различные информационные ресурсы: файлы, приложения, устройства вывода и т.д.

Примеры субъектов: пользователь, программа выполняющаяся под именем пользователя.

Примеры объектов: файлы, каталоги, внешние накопители (CD,DVD,USB и т.д.), принтер, сетевой адаптер.

Дискреционная модель разграничения доступа основывается на следующих основных положениях:

1. Все субъекты и объекты должны быть однозначно идентифицированы;
2. Для любого объекта должен быть определен пользователь - владелец;
3. Владелец объекта обладает правом определения прав доступа к объекту со стороны любых субъектов;
4. В системе существует привилегированный пользователь, обладающий правом полного доступа к любому объекту (или правом становиться владельцем любого объекта).

Последнее свойство определяет невозможность существования в системе потенциально недоступных объектов, владелец которых отсутствует. Но реализация этого положения не означает, что привилегированный пользователь может использовать свои полномочия незаметно для реального владельца объекта.

Дискреционное разграничение доступа реализуется обычно в виде матрицы доступа, строки которой соответствуют субъектам компьютерной системы, а столбцы — ее объектам. Элементы матрицы доступа определяют права доступа субъектов к объектам. В целях сокращения затрат памяти матрица доступа может задаваться в виде списков прав субъектов (для каждого из них создается список всех объектов, к которым разрешен доступ со стороны данного субъекта) или в виде списков контроля доступа (для каждого объекта информационной системы создается список всех субъектов,

которым разрешен доступ к данному объекту).

К достоинствам дискреционного разграничения доступа относятся относительно простая реализация (проверка прав доступа субъекта к объекту производится в момент открытия этого объекта в процессе субъекта) и хорошая изученность (в наиболее распространенных операционных системах универсального назначения типа Microsoft Windows и Unix применяется именно эта модель разграничения доступа).

Недостатки. Прежде всего, к ним относится статичность разграничения доступа — права доступа к уже открытому субъектом объекту в дальнейшем не изменяются независимо от изменения состояния компьютерной системы.

При использовании дискреционного разграничения доступа не существует возможности проверки, не приведет ли разрешение доступа к объекту для некоторого субъекта к нарушению безопасности информации в компьютерной системе (например, владелец базы данных с конфиденциальной информацией, дав разрешение на ее чтение другому пользователю, делает этого пользователя фактически владельцем защищаемой информации). Иначе говоря, дискреционное разграничение доступа не обеспечивает защиты от утечки конфиденциальной информации. А так же автоматическое назначение прав доступа субъектам (из-за большого количества объектов в информационной системе в качестве субъектов доступа остаются только ее пользователи, а значение элемента матрицы доступа вычисляется с помощью функции, определяющей права доступа порожденного пользователем субъекта к данному объекту компьютерной системы).

Пример. Когда вы расписываете доступ к файлу, вы указываете:

- имя владельца файла (субъект);
- права на чтение;
- права на запись;
- права на запуск на выполнение.

Пример дискреционного управления доступом к файлам в Linux

представлен на рисунке 2.

	O <sub>1</sub> 	O <sub>2</sub> 	O <sub>3</sub> 
S <sub>1</sub> 	R W X	R X	R W X
S <sub>2</sub> 	R X	R W	W X
S <sub>3</sub> 	R	X	W

```
# ls -l
drwxr-xr-x. 2 root root 4096 янв. 30 18:37 anaconda
drwxr-x---. 2 root root 4096 апр. 1 21:27 audit
-rw-r--r--. 1 root root 12094 апр. 2 03:33 boot.log
-rw-----. 1 root utmp 384 апр. 2 15:25 btmp
-rw-----. 1 root utmp 1536 марта 15 07:41 btmp-20120401
-rw-w----. 1 root 997 0 янв. 30 18:35 clamav-milter.log
drwxr-xr-x. 2 root root 4096 янв. 30 14:44 ConsoleKit
-rw-r--r--. 1 root root 267059 апр. 2 15:25 cron
-rw-r--r--. 1 root root 1241791 апр. 1 03:22 cron-20120401
-r-----. 1 root root 94710 марта 13 12:51 dracut.log-20120314
drwx-----. 2 root root 12288 апр. 1 03:22 httpd
drwxr-xr-x. 2 root root 4096 февр. 6 12:26 iptraf-ng
-rw-r--r--. 1 root root 292000 апр. 2 15:25 lastlog
drwxr-xr-x. 2 root root 4096 янв. 30 18:34 mail
-rw-r-----. 1 mysql mysql 3277 апр. 2 12:04 mysqld.log
```

Рисунок 2 – Дискреционное разграничение доступа

Дискреционное управление доступом в ОС проекта GNU/Linux основано на понятии владения (использовании права доступа владения) файлом, процессом, каталогом (сущностями и субъект-сессиями). Так, с каждым файлом или каталогом связана учётная запись пользователя — их владельца (owner). Процесс, который функционирует от имени такой учётной записи-владельца сущности, имеет право изменять дискреционные права доступа к ней, например, назначать их учётным записям других пользователей ОС на основе стандарта POSIX ACL. Однако зачастую настраивать права доступа для каждой учетной записи бывает затруднительно и поэтому пользователей объединяют в группы, которым тоже можно назначить права доступа.

## **2 ACL-списки (Access control list)**

Списки управления доступом - это защищенные информационные ресурсы, в которых указано, кому и к каким ресурсам разрешен доступ. Операционная система обеспечивает раздельную защиту для разных режимов доступа. Владелец информации может предоставить другим пользователям права на запись или чтение для своего ресурса. Пользователь, которому предоставлены права доступа к ресурсу, может передавать эти права другим пользователям. Такая схема позволяет управлять распространением информации в системе; права доступа к объекту задаются его владельцем.

Права доступа пользовательского типа разрешают пользователю доступ только к его собственным объектам. Обычно пользователи получают также права доступа группы или права доступа по умолчанию для ресурса. Основная задача управления правами доступа состоит в определении групп пользователей, поскольку именно принадлежность к группе дает пользователю права доступа к чужим объектам.

Списки управления доступом (ACL) повышают эффективность управления защитой файлов, добавляя расширенные права к Базовые права доступа, предоставленным конкретным пользователям и группам. С помощью Расширенные права доступа можно разрешить или запретить доступ к ресурсу конкретным пользователям или группам, не меняя базовые права доступа.

```

1: # file: somedir/
2: # owner: lisa
3: # group: staff
4: # flags: -s-
5: user::rwx
6: user:joe:rwx
7: group::rwx
8: group:cool:r-x
9: mask::r-x
10: other::r-x
11: default:user::rwx
12: default:user:joe:rwx
13: default:group::r-x
14: default:mask::r-x
15: default:other::---

```

Информация о файле, его владельце, владеющей группе и режимах доступа

The diagram illustrates the breakdown of an ACL entry into three categories:

- Базовые разрешения** (Basic Permissions) are shown in green, corresponding to lines 5-10.
- Расширенные разрешения** (Extended Permissions) are shown in red, corresponding to lines 11-14.
- Разрешения по умолчанию** (Default Permissions) are shown in orange, corresponding to line 15.

Each category is associated with its respective effective permission: #effective:r-x for basic, #effective:r-x for extended, and #effective:r-x for default.

Рисунок 3 – ACL-списки

Типы файлов		Назначение
Обычные файлы	—	Хранение символьных и двоичных данных
Каталоги	d	Организация доступа к файлам
Символьные ссылки	l	Предоставление доступа к файлам, расположенных на любых носителях
Блочные устройства	b	Предоставление интерфейса для взаимодействия с аппаратным обеспечением компьютера
Символьные устройства	c	
Каналы	p	Организация взаимодействия процессов в операционной системе
Сокеты	s	

Рисунок 4 - Типы файлов в Linux и права доступа

POSIX ACL – POSIX-совместимый стандарт определения разрешений на доступ к объектам, основанный на списках контроля доступа (ACL – Access Control Lists). Он реализует несколько вариантов дискреционной модели, отличающихся уровнем функциональности и методами хранения списков контроля доступа.

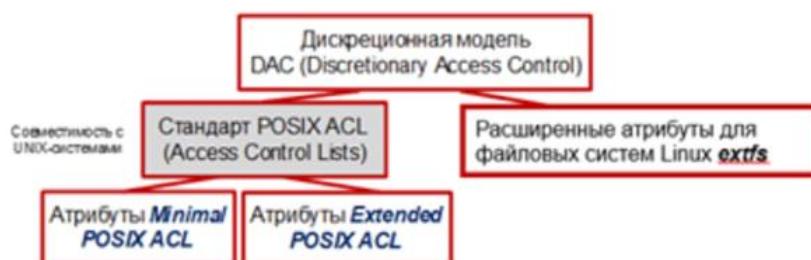


Рисунок 5 - Стандарты ACL

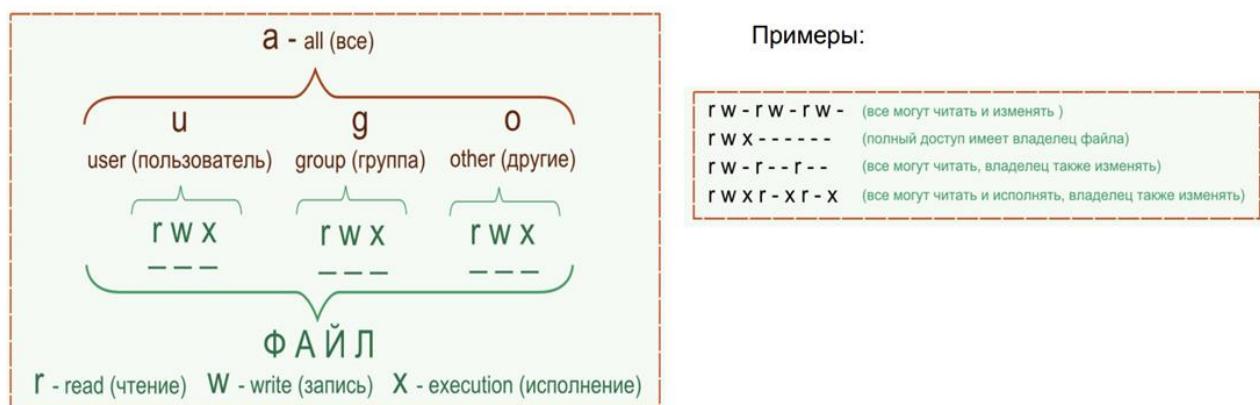
Как видно из схемы, существует 2 вида атрибутов для вышеупомянутого стандарта.

## 2.1 Minimal POSIX ACL. Список контроля на уровне inode файла

В метаданных каждого объекта (в inode файла/директории) содержится список разрешений на доступ к нему для разных категорий субъектов.

Атрибуты Minimal ACL поддерживают три базовых класса субъектов доступа к файлу (класс All объединяет все три класса):

- User access (u) – доступ для владельца файла;
- Group access (g) – доступ для группы, владеющей файлом;
- Other access (o) – доступ для остальных пользователей (кроме пользователя root).
- All access (a) – доступ для всех субъектов доступа (u, g, o).



Для каждого из этих классов определены три типа разрешений:

- на чтение содержимого файла (read) – символ «г».
- на запись внутри файла или изменения его содержимого (write) – символ «w».
- на исполнение файла (если это бинарный исполняемый файл или файл сценария интерпретатора (execute)) – символ «x».

Список разрешений Minimal ACL представлен в inode файла (директории) девятью байтами (символами). Каждый байт определяет одно из разрешений (символы r, w, x) или их отсутствие (символ «-»). Байты

разрешений сгруппированы в следующие классы:

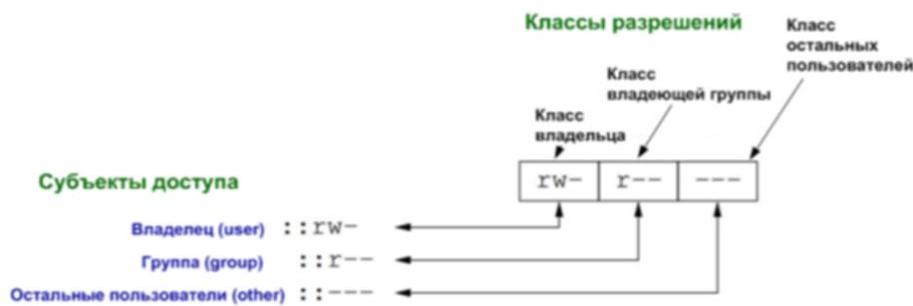


Рисунок 6 - Список разрешений Minimal ACL

Для того, чтобы посмотреть атрибуты отдельного файла или директории в символьном виде можно воспользоваться командой:

```
ls -l <имя_файла>
```

Таким образом понятно, что это файл обычного типа (-). Для владельца разрешены чтение, запись и исполнение (rwx), для владеющей группы – чтение и запись (rw-), для остальных пользователей – только чтение (r--).

Для директорий трактовка типов разрешений иная:

r – разрешение на «открытие» директории, то есть на чтение списка файлов, которые содержит эта директория.

w – разрешение на «модификацию» этого списка файлов (создание/удаление/переименование/перемещение) файлов этой директории.

x – разрешение на «исполнение» директории, то есть на возможность перейти в нее.

Чтобы изменить расширения для определенного файла используется команда chmod с соответствующими аргументами:

Субъекты доступа	Действия	Классы разрешений
<b>u</b>	+ добавление разрешения	<b>r</b>
<b>g</b>	- удаление разрешения	<b>w</b>
<b>o</b>		<b>x</b>
<b>a</b>	= установка именно этого разрешения	

Изменение прав доступа, команда chmod:

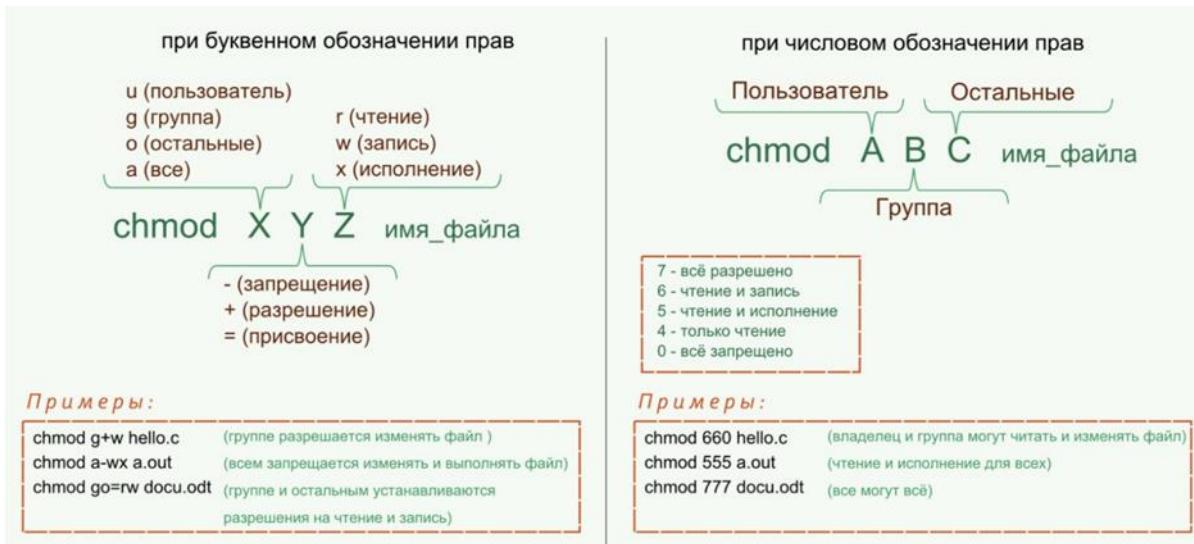


Рисунок 7 - Принцип работы chmod

Пример.

`chmod o=r test.file` – установка разрешения «чтение» для остальных пользователей (не владельцев объекта), вне зависимости какие разрешения были установлены ранее

`chmod g+rw test.file` – добавление разрешений «чтение» и «запись» для группы, владеющей файлом.

Также Minimal ACL имеют числовое представление (Numeric Notation) в виде трех восьмеричных чисел. Эти числа определяют разрешение на доступ к файлу или директории трех субъектов доступа (u,g,o). Каждое из них формируется путем суммирования восьмеричных значений необходимых разрешений:

- чтение = 4;
- запись = 2;
- выполнение = 1.

Например, полный доступ (rwx) – это:  $4+2+1=7$

Таким образом, можно задать разрешения для файла следующим образом:

`chmod 0740 test.file`, что будет соответствовать атрибутам `-rwxr----`

Примеры записи прав доступа в **двоичной** форме:

110 110 110	(все могут читать и изменять)
111 000 000	(полный доступ имеет владелец файла)
110 100 100	(все могут читать, владелец также изменять)
111 101 101	(все могут читать и исполнять, владелец также изменять)

Примеры записи прав доступа в **восьмеричной** форме:

6 6 6	(все могут читать и изменять)
7 0 0	(полный доступ имеет владелец файла)
6 4 4	(все могут читать, владелец также изменять)
7 5 5	(все могут читать и исполнять, владелец также изменять)

Перевод представления прав доступа к **восьмеричной** форме:

гнх-представление	Двоичное число	Восьмеричное число	Значение
- - -	0 0 0	0	Все запрещено
- - x	0 0 1	1	
- w -	0 1 0	2	
- w x	0 1 1	3	
r - -	1 0 0	4	Только чтение
r - x	1 0 1	5	Чтение и исполнение
r w -	1 1 0	6	Чтение и запись
r w x	1 1 1	7	Все разрешено

Рисунок 8 - Права доступа к файлам, как числовая нотация

В ряде случаев в ходе пользовательского сеанса возникает необходимость смены разрешений при доступе к файлам и директориям.

Например, для файла /etc/shadow любой пользователь может записать в него хеш своего пароля

```
root@astra-client:~# ls -l /etc/shadow
-rw-r----- 1 root shadow 1115 Окт 22 18:17 /etc/shadow
root@astra-client:~# █
```

Такая смена разрешений называется изменение режима доступа.

Она реализуется установкой специальных флагов в зарезервированном десятом байте списка Minimal POSIX ACL или заменой ими байтов разрешений.

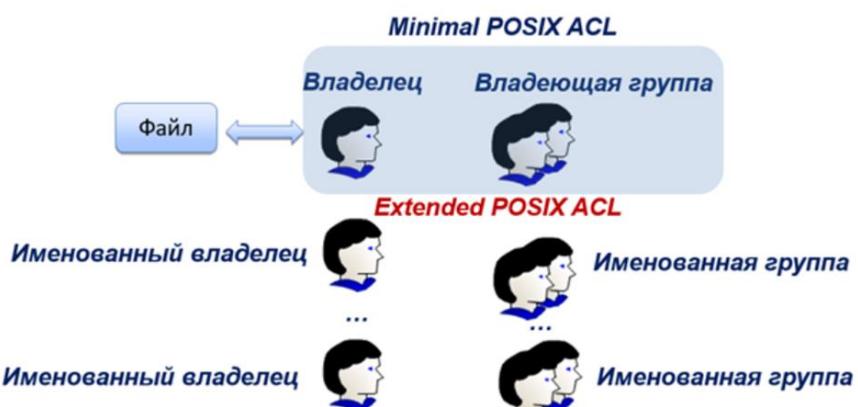
Флаг -t (sticky bit). Устанавливается в разрешениях директорий и разрешает манипулирование файлами внутри этих директорий только их владельцем

Флаг -s (флаг режимов SUID и SGID). Устанавливается вместо разрешения «x» у владельца (режим SUID) и/или владеющей группы (режим SGID) и разрешает исполнение этого файла с разрешениями его владельца и/или владеющей группы, а не с разрешениями пользователя, запустившего файл.

## 2.2 Extended POSIX ACL (EA). Расширенные списки контроля доступа

Вполне естественно, что такая простая схема, как в Minimal POSIX ACL имеет ряд недостатков. Самый явный из них, это отсутствие гибкости при совместном доступе субъектов к объектам. К тому же, списки доступа Extended POSIX ACL создаются и хранятся в системных таблицах ядра ОС.

Основой Extended POSIX ACL является понятие «именованный пользователь (группа)» (named user (group), что позволяет нам выдавать отдельный набор разрешений для конкретных пользователей или групп.



Для работы с Extended POSIX ACL используются следующие команды:

Просмотр: системный вызов `getfacl <имя файла>`

Установка: системный вызов `setfacl`

Маска `umask` - это пользовательская маска (user mask), которая используется для определения конечных прав доступа

```
$ umask 022
```

### Методика выполнения и контрольные вопросы

У вас должно быть право на чтение файла `/etc/passwd`.

Для ответа на контрольные вопросы воспользуйтесь для вывода на экран `/etc/passwd` утилитой `cat` или `less`.

1. Просмотрите в файле `/etc/passwd` поля с информацией о пользователях

вашей системы.

- a) Какой символ используется для разделения полей в /etc/passwd?
- b) Сколько полей используется для описания каждого пользователя?
- c) Сколько пользователей в вашей системе?

2. Сколько различных входных оболочек используется в вашей системе?

(Совет: посмотрите на последнее поле.)

3. Второе поле в /etc/passwd хранит пароли пользователей в закодированной форме. Если поле пароля содержит букву x, то ваша система использует теневые пароли и хранит закодированные пароли в другом месте. Применяются ли в вашей системе теневые пароли?

4. Добавьте трех новых пользователей с соответствующими домашними директориями: student7, student8, student9. Задайте пароли для каждого из них.

5. Создайте группу course и добавьте в нее всех трех пользователей.

6. Для пользователя student7 выставите ограничение: срок действия пароля 5 месяцев и предупреждение об окончании срока действия пароля 7 дней.

7. Заблокируйте пользователя student8. Проверьте, что блокировка подействовала.

8. Войдите в систему под пользователем student9. Создайте два новых файла file1 и file2 и две новых директории dir1 и dir2.

9. Войдите в систему под пользователем root. Сделайте file1 исполняемым. Запускаться файл должен от имени владельца и группы файла. Установите в качестве владельца файла file1 пользователя student7 и группу course.

10. Войдите в систему под пользователем student7. Попробуйте перейти в директорию пользователя student9 и сделать запись в файле file1.

11. Войдите в систему под пользователем root. Разблокируйте пользователя student8. Проверьте, что блокировка снята.

12. Управление учетными записями пользователей. Создайте в ОС Linux двух пользователей (user1 и user2) и задайте их пароли. Зарегистрируйтесь в

первой консоли как user1.

13. С помощью Ctrl+Alt+F2 (Alt+F2) откройте второй текстовый терминал и зарегистрируйтесь как user2.

14. Аналогично откройте третий текстовый терминал и зарегистрируйтесь в нем с правами суперпользователя.

15. Нажатием Ctrl+Alt-F1 (Alt-F1) вернитесь в первую консоль. Теперь, переключая консоль, вы можете работать с объектами операционной системы от имени двух разных пользователей и администратора системы. Основная часть задания выполняется с правами обычного пользователя. Переходите в третью консоль и используйте права root только при выполнении соответствующих пунктов задания.

16. С правами user1 попробуйте войти в каталог /root. Объясните результат. С помощью команды ls -la / просмотрите список основных каталогов и укажите, каких прав доступа вам недостает для входа в каждый из каталогов.

17. Переключитесь в консоль администратора и создайте два новых временных каталога mkdir -m 777 /home/temp1 и mkdir -m 1777 /home/temp2. Проверьте права доступа к каталогам /home/user1 и /home/user2: они должны быть установлены в 755. Вернитесь в консоль user1.

18. Пользуясь командой mkdir, создайте в домашнем каталоге пользователя /home/user1 четыре каталога с именами: qu1, qu2, qu3, qu4. При создании каталогов объявице следующие права доступа к ним: (qu1 - 777, qu2 - 404, qu3 - 1333, qu4 - 505. Пример: cd; mkdir -m 777 qu1). С помощью команды ls /home/user1 убедитесь в том, что каталоги созданы. Какие из предоставленных прав кажутся Вам лишенными смысла? Почему?

19. Задайте права доступа к файлам "по умолчанию". Для этого установите umask 022.

Поясните, какие права к вновь создаваемым файлам и каталогам будут предоставляться пользователю, членам его группы и остальным.

20. В каждом из каталогов создайте по три текстовых файла с именами

(jan, feb, mar), (apr, may, jun), (jul, aug, sep), (oct, nov, dec). В каждый файл запишите календарь на определенный месяц текущего года. Например, команда cal 1 2021 > jan создает в текущем каталоге файл jan и записывает в него календарь на январь 2021 года. Не забывайте, что использование относительного (короткого) имени файла требует, чтобы вы находились в нужном каталоге. В противном случае следует указывать полный путь к создаваемому файлу. Для навигации по каталогам используйте команды cd и pwd. В каком случае создание файлов не удалось? Почему?

21. С помощью команды chmod измените нужные права доступа в "недоступные" каталоги qu2, qu4 и создайте там указанные файлы. После этого верните каталогам прежние права доступа.

22. С помощью команд cd и ls войдите в каждый из созданных каталогов и просмотрите список созданных файлов. Для просмотра каталога необходимо последовательно ввести две команды: cd и ls. При просмотре используйте два режима: ls без аргументов и ls -l. В каких случаях не удалось войти в каталог? В каких случаях не удалось посмотреть список файлов? Почему?

23. Прочитайте содержимое одного из файлов в "темном" каталоге (например, cd /home/user1/qu3; cat aug). Сделайте выводы.

24. Перейдите во 2-ю консоль и с правами пользователя user2 войдите в каталог /home/user1/qu1. Создайте в каталоге /home/user2 новый файл quart1 путем конкатенации нескольких имеющихся (cat jan feb mar >/home/user2/quart1). С помощью команды file определите тип созданного файла. Попробуйте вывести его на экран командой cat. Что представляет собой данный файл?

25. С помощью команды chmod установите права доступа 077 на созданный файл quart1.

Вновь попробуйте прочесть его. Ответьте, почему владельцу файла запрещается доступ, если файл доступен для всех? Что необходимо сделать, чтобы вернуть владельцу права на доступ?

26. Установите для файла quart1 права на доступ 4700. Кому и какие

права вы при этом предоставили? Как воспользоваться этими правами? Какие из предоставленных прав не имеют смысла?

27. Перейдите в консоль администратора и передайте право владения на файлы may и aug пользователю user2 (команда chown). Поочередно из консолей user1 и user2 проверьте, как изменились права владения файлами после его передачи. Может ли пользователь user2 воспользоваться предоставленными правами?

28. С правами пользователя user1 из каталогов /home/temp1 и /home/temp2 с помощью команды ln создайте две "жесткие" ссылки на файл dec с именами dec\_h1 и dec\_h2 (пример: ln /home/user1/qu4/dec /home/temp1/dec\_h1). Чем созданные ссылки отличаются от исходного файла? На сколько байт уменьшилось дисковое пространство после создания этих ссылок?

29. С помощью команды ln -s создайте из каталогов /home/temp1 и /home/temp2 две символические ссылки на файл dec с именами dec\_s1 и dec\_s2.

Чем отличаются созданные ссылки от исходного файла? Попытайтесь прочитать содержимое файлов символических ссылок. Что они собой представляют?

30. С правами пользователя user2 с помощью команды cp создайте в каталогах

/home/temp1 и /home/temp2 по одной копии файла dec с другим именем (dec\_copy1). Чем отличаются исходный файл и его копия (обратите внимание на то, кто является владельцем исходного файла и его копии)? Чем отличаются права доступа на эти файлы?

31. Вернитесь в консоль user1. С помощью команды rm удалите файл dec. Что произошло с "жесткими" и символическими ссылками на данный файл? Что произошло с его копиями? Что нужно сделать для того, чтобы файл перестал существовать (на логическом уровне)?

32. С правами user1 удалите файлы из каталогов /home/temp1 и /home/temp2. Какие файлы не удалось удалить? Почему? Попробуйте удалить

оставшиеся файлы правами пользователя user2. Объясните результат.

33. Попробуйте удалить любой из каталогов qu1, qu2, qu3, qu4 с помощью команды rmdir (не удаляя предварительно из них файлов). Объясните результат.

34. Войдите в консоль администратора и с правами root, пользуясь командой chattr, заблокируйте файл feb от любых изменений. Установите параметр запрета любых операций, кроме добавления данных для файла mar. Вернитесь в консоль user1. С помощью команды lsattr -l проверьте наличие дополнительных атрибутов у файлов.

35. С правами пользователя user1 добавьте одну строку finish в конец файлов feb и mar

(воспользуйтесь для этого командой echo fi nish >> fi le\_name).

Убедитесь в

успешном завершении операции, объясните результат.

36. С правами пользователя user1 с помощью команды rm -rf последовательно удалите ранее созданные каталоги qu2, qu3, qu4 вместе с файлами. Объясните результат.

37. С помощью команды md5sum вычислите и запишите контрольную сумму для одного из файлов в каталоге /home/user1/qu1. Добавьте один символ в этот файл с помощью команды echo (например, echo a >> /home/user1/qu1/jan).

Вновь вычислите контрольную сумму файла и сравните два результата.

38. С помощью команды find с правами администратора найдите в корневом каталоге файлы:

- имеющие атрибуты SUID (find / -type f -perm -4000);
- файлы, которые разрешено модифицировать всем (find / -type f - perm -2);
- файлы, не имеющие владельца (find / -nouser);
- объясните, какой интерес могут представлять для администратора указанные категории файлов?

39. Зарегистрируйтесь в системе в консольном режиме с правами root.

Используя команду cat с правами root, просмотрите содержимое файлов /etc/passwd и /etc/shadow.

Задание. Вам необходимо создать учетные записи и определить права доступа для десяти (10) сотрудников: w\_gromov, n\_kalinina, e\_ivanova, r\_klinova, b\_rebrov, k\_beglov, i\_frolov, d\_lavrov, m\_kruglov, t\_uporov, работающих в одном подразделении и занятых созданием и редактированием текстовых документов различного уровня конфиденциальности.

Разграничение доступа к информации должно быть произведено на основании следующих требований:

- допуск к секретным сведениям имеют четыре пользователя: w\_gromov, n\_kalinina, b\_rebrov, k\_beglov;
- три пользователя: n\_kalinina, b\_rebrov, k\_beglov работают над созданием секретных документов, каждый по своему профилю. Их домашние каталоги и файлы должны быть полностью недоступными как друг для друга, так и для всех остальных, исключая w\_gromov;
- три пользователя: i\_frolov, d\_lavrov, e\_ivanova имеют допуск к конфиденциальной информации и работают над документами с соответствующим грифом. Они имеют право читать файлы с конфиденциальной информацией, созданные своими коллегами, без права их модификации;
- все секретоносители имеют право знакомиться с конфиденциальными файлами;
- три пользователя: r\_klinova, m\_kruglov, t\_uporov могут работать только с открытой информацией. Их файлы должны быть доступны для чтения каждому сотруднику подразделения (без права модификации);
- w\_gromov является редактором подразделения и имеет право читать и модифицировать файлы всех сотрудников и всех уровней конфиденциальности. Завершенные документы копируются пользователем w\_gromov в его домашний каталог, который должен быть недоступен для всех остальных сотрудников подразделения.

Укажите в отчете, какие коллизии вы усматриваете в сформулированных требованиях?

Как реализовать указанные требования таким образом, чтобы пользователи не могли по своему усмотрению изменять установленный порядок?

40. С помощью команды groupadd создайте четыре пользовательских группы: alfa, beta, nabla, sigma. Формат команды groupadd -g GID group\_name.

Идентификатор группы GID можно назначать произвольно, начиная с номера 100 (например, groupadd -g 101 alfa).

41. Создайте учетные записи для вышеуказанных десяти новых пользователей.

Регистрационные данные (кроме паролей и групп) сведены в таблицу 1. Пароли назначайте произвольно, длиной не менее 8 символов, не забывая фиксировать их в отчете. Для пользователей e\_ivanova, r\_klinova задайте одинаковые пароли. Распределите сотрудников по группам таким образом, чтобы удовлетворить вышеперечисленным требованиям. Изобразите в отчете схему, поясняющую разграничение доступа сотрудников подразделения к компьютерной информации.

42. Пять первых пользователей (w\_gromov, n\_kalinina, e\_ivanova, r\_klinova, b\_rebrov) зарегистрируйте с помощью команды useradd. Синтаксис команды:

```
useradd - u UID - g group_name - d dir_home - m - p password -e  
date_del_user user_name.
```

Например, useradd -u 501 -g sigma -d /home/n\_kalinina -p v5g7K2S4 -e 2011-01-07 n\_kalinina. Параметр -m обеспечивает создание домашнего каталога пользователя, если он еще не существует.

Прочие параметры команды можно не указывать. Помните, имя пользователя не должно начинаться с цифры и содержать заглавных и русских букв, символов типа

\*#%0Л ....

Идентификаторы пользователей UID назначаются, начиная с 500. Дата удаления учетной записи пользователя вводится в формате ГГГГ-ММ-ДД.

43. Пять последних пользователей зарегистрируйте с помощью командного файла `adduser`, которая запрашивает значения в интерактивном режиме. При вводе данных ориентируйтесь на подсказки системы [в квадратных скобках]. Все параметры, кроме имени пользователя, его идентификатора, имени группы, пароля и домашнего каталога можно игнорировать. Для ввода параметра по умолчанию вводить `Enter`.

Переключаясь во вторую консоль, отслеживайте изменения, происходящие в файле `/etc/passwd` по мере ввода новых учетных записей.

44. Посмотрите с правами пользователя файл `/etc/shadow`. Повторите попытку просмотра с консоли суперпользователя. Почему поля, отведенные для хэшированных паролей у пользователей `e_ivanova` и `r_klinova` различаются?

45. Из первой консоли с помощью команды `su` измените права администратора на права пользователя `w_gromov`. Почему система не запрашивает пароль?

С помощью команды `exit` верните себе права администратора. Был ли запрошен пароль? (В различных дистрибутивах Linux возврат полномочий администратора организован различным образом).

46. Запустите оболочку `Midnight Commander` в режиме редактирования (F4) файла паролей `/etc/passwd` и удалите в учетной записи пользователя `n_kalinina` символ признака пароля (между первым и вторым двоеточием), включая пробел. Сохраните изменения в файле, завершите сеанс в `Midnight Commander`, с помощью `Ctrl+Alt+F2`

(Alt+F2) откройте второй текстовый терминал и зарегистрируйтесь пользователем `n_kalinina`, но теперь с «пустым» паролем. Сделайте вывод относительно опасности предоставления прав на запись в этот файл. Завершите сеанс для пользователя `n_kalinina` с помощью команды `exit`.

47. Пользователь `d_lavrov` уволен за дисциплинарный проступок. С помощью команды `userdel -r user_name` удалите его учетную запись вместе с домашним каталогом. В реальных условиях необходимо вначале скопировать в другую директорию файлы пользователя, представляющие ценность для организации.

48. Зарегистрируйте вместо уволенного пользователя нового сотрудника `f_mironov` с предоставлением ему аналогичных прав (пароль должен быть новым!).

49. Пользователь `r_klinova` убыла в командировку сроком на две недели. Заблокируйте ее учетную запись, для чего с правами администратора войдите в режим редактирования файла паролей и вставьте во второе поле (между первым и вторым двоеточием) любой символ, который не разрешено использовать для пароля.

Попытайтесь зарегистрироваться во второй консоли с правами `r_klinova` и убедитесь в том, что для этого пользователя система не доступна.

50. Зарегистрируйтесь во второй консоли с правами пользователя `k_beglov`, вызовите команду `passwd` и измените свой пароль. В качестве нового пароля введите `qwerty`.

51. Перейдите в консоль администратора и назначьте пользователю `k_beglov` новый пароль `zxcvbnm`. Затем с помощью команды `chage` (`change aging` – изменить информацию об устаревании) установите для этого пользователя минимальное время действия паролей, равное 5 дням. С какой целью устанавливается минимальный срок действия пароля?

52. Просмотрите электронную справку по файлу `/etc/sudoers`. Отредактируйте его таким образом, чтобы предоставить следующим пользователям дополнительные права за счет использования команды `sudo`:

- пользователю `e_ivanova` - право монтировать файловые системы,
- пользователю `b_rebrov` - право изменения владельца файлов.

Ответьте, чем отличается предоставление прав пользователям с помощью `sudo` от использования эффективных идентификаторов SUID?

53. Из второй консоли с правами пользователя f\_mironov создайте файл cal 2021 >/home/f\_mironov/cal2021. С помощью команды su переключите консоль на пользователя b\_rebrov и с помощью временно предоставленных ему привилегий передайте права на созданный f\_mironov файл другому владельцу n\_kalinina. Каким еще путем можно предоставить подобные права пользователям, не передавая им "опасных" полномочий администратора?

54. Просмотрите с правами администратора системные журналы в каталоге /var/log и убедитесь, что система зафиксировала факты присвоения полномочий администратора.

55. Отключитесь от всех терминалов и переключитесь на 1-ый терминал: <CTRL-ALT-F1>. Войдите в систему как обычный пользователь (не root). Получите информацию о данном логине, представьте преподавателю.

Получите информацию обо всех параллельных логинах на локальной рабочей станции (должен только быть один пользователь, зарегистрированный в системе).

Переключитесь в виртуальный терминал 2: <CTRL-ALT-F2>

Войдите с данного терминала как пользователь student

Получите информацию о данном логине, представьте преподавателю. Получите информацию обо всех параллельных логинах на локальной рабочей станции.

Находясь в системе, под пользователем student посмотрите маску.

Создайте пару файлов и директорий, не изменяя прав доступа к ним.

Измените вашу маску на более безопасную, а затем создайте новый файл и новую

Директорию. Как изменились права доступа у новых файлов?

56. Для выполнения этого задания, Вы должны находиться в системе под пользователем root. Создайте общую директорию для всех пользователей /var/ftp/pub. Все пользователи должны иметь возможность записи в директорию, но удалять файлы из директории пользователь может только в том случае, если файл принадлежит ему.

57. Создайте новый файл file2 в вашей рабочей директории. Посмотрите ACL для файла file2. Совпадают ли права доступа с установленной маской?

Измените права доступа на чтение, запись и выполнение для группы файла file2.

Посмотрите права на file2 с помощью команды ls –l а также посмотрите ACL для этого файла.

58. Установите маску для файла file2 - только чтение. Посмотрите права на file2 с помощью команды ls –l а также посмотрите ACL для этого файла.

59. Если группа group1 не существует в вашей системе, создайте эту группу с ID равным 101. Добавьте в ACL группу group1 для файла file2. Установите только права на чтение и выполнение для данной группы.

60. Добавьте в ACL пользователя user10 для файла file2. Добавьте для этого пользователя доступ по исполнению файла.

# **Лабораторная работа № 5**

## **Файловая система ОС Astra Linux Special Edition. LVM. Swap**

**Цель работы** – Научится работать с символьическими и жесткими ссылками, добавлять новые диски в систему и обслуживать их (разбивать диски на разделы, создавать нужную файловую систему (ФС), изменять параметры файловой системы). Научится создавать и обслуживать разделы LVM. Настройка swap.

### **Теоретические сведения**

#### **1 Управление файловыми системами**

Поддерживаемые типы файлов системы:

- модули ядра (драйверы) поддерживаемых файловых систем в установленной операционной системы `ls/lib/modules/$(uname -r)/kernel/fs`
- список драйверов файловых систем, загруженных в данный момент `cat/proc/filesystems` (nodev-псевдофайловые или временные файловые системы)
- все ФС должны предоставлять VFS информацию о суперблоке, inode, dentry, блоках данных
  - «родные» для Linux ФС – ext2/ext3, ext4
  - дисковые ФС, поддерживаемые ядром: XFS, BtrFS, iso9660, udf.

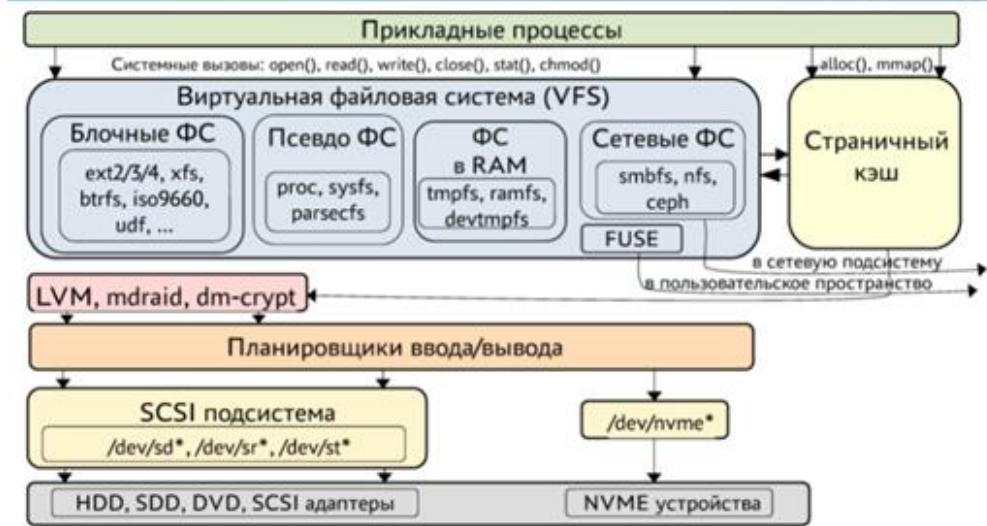


Рисунок 1 – Архитектура подсистемы хранения данных

Именование файлов дисковых устройств:

- `/dev/sd*` - дисковые устройства с интерфейсом последовательной передачи данных (SCSI, SATA, см. man 4 sd)
- `/dev/hd*` дисковые устройства с интерфейсом параллельной передачи данных (PATA, см. man 4 hd)
- Буква (a,b,c,...) после sd или hd – номер диска
- Число после буквы – номер раздела (`/dev/sdb4`)
- Если числа нет, то весь диск (`/dev/sdc`)
- В случае BIOS/MBR: поддерживается 4 первичных раздела (1-4), один из которых может объявлен, как расширенный. В расширенном разделе могут быть созданы логические разделы (первый логический раздел -5)
- В UEFI/GPT все разделы первичные (1-128)

Файловая система ext2:

- Макс. размер файла:
  - 16 GiB (размер блока 1 KiB)
  - 1 TiB (размер блока 4 KiB)
- Макс. размер ФС:
  - 4 TiB (размер блока 1 KiB)
  - 16 TiB (размер блока 4 KiB)
- Пространство ФС разбито на группы блоков: резервный суперблок, таблица описания группы, битовые карты inode и блоков данных, области inode и блоков данных (dump2fs)

Суперблок ФС содержит информацию о самой файловой системе (тип, размер, состояние, UUID и т.д.). Содержимое суперблока можно посмотреть командой: `tune2fs -l /dev/имя_раздела`

Суперблок в ФС дублируется:

`dump2fs/dev/имя_раздела | grep superblock` позволяет определить номера резервных суперблоков.

При монтировании устройства можно указать резервный суперблок (параметр `-o sb=n` команды `mount`, `n`- номер резервного суперблока).

Размер inode хранится в сперблоке (128.256 байт).

Начиная с ядра 2.6.10 – 256 байт (для хранения точного времени и расширенных атрибутов, включая метку безопасности (мандатную метку)). Команда для просмотра inode: `debugfs -R «stat имя_файла» имя_блочного_устройства`

Для адресов блоков данных отведено: 12 прямых указателей, 1 косвенный указатель, 1 – указатель с двойной косвенной адресацией, 1 – с тройной косвенной адресацией.

Файловая система **ext3**. Главные отличия от ext2:

- наличие журнала,
- онлайн увеличение размера ФС,
- использование сбалансированного дерева для индексирования больших каталогов обеспечило более быстрый поиск файлов,
- ext3 совместима с ext2 (ext2 может быть преобразована в ext3 (`tune2fs -j`) и наоборот (`tune2fs -O^has_journal`),
  - режимы работы журнала: journal, ordered, writeback,
  - журнал обычно размещается в конце раздела ФС,
  - максимальный размер ФС 1 ЕiВ, файла -16 Тб (при размере блока 4 Кб).

Отличительные черты:

- размещение данных экстентами,
- отложенное распределение (выделение) блоков – распределение блоков откладывается до тех пор, пока не пойдет запись на диск,
- предварительное выделение места для файла на диске,
- контрольные суммы журнала – для определения возможных проблем в журнале.

Основные характеристики ФС **XFS**:

-64-х разрядная ФС (максимальный размер файла -8ЕiВ, максимальный размер ФС 8ЕiВ)

- увеличение производительности путем использования линейных областей (allocation groups)

- использование экстентов для выделения места в области данных

- журналируемая ФС (только метаданные)

- индексные дескрипторы выделяются динамически

- поддержка дефрагментации «на лету»

- возможность увеличения размера ФС «на лету»

- поддержка отложенного выделения места (delayed allocation).

Основные характеристики ФС **Btrfs**:

- 64х- разрядная ФС (макс. Размер файла 16 ЕiВ, раздела – 16 ЕiВ)

- поддержка механизма «капирования при записи» (Copy On Write, COW)

- поддержка подтомов (subvolumes)

- поддержка снимков состояния ФС (использует механизм COW и подтома)

- дефрагментация и сжатие данных «на лету»

- динамическое размещение inode

- целостность данных (вычисление контрольных сумм для данных и метаданных)

- встроенная поддержка многодисковых ФС (RAID и LVM)

## 2 Создание дисковых разделов

Утилиты для разметки диска:

fdisk имя\_диска

parted имя\_диска

GUID Partition Table, GPT — стандарт формата размещения таблиц разделов на физическом жестком диске используется в настоящее время вместо MBR.

Утилиты создания разделов:

- `sfdisk` - предназначена для использования в сценариях (скриптах)
- `cfdisk` - псевдографическая утилита
- `gparted` - графическая утилита

Список дисков и разделов `fdisk -l`

Список разделов на одном диске `fdisk -l имя_диска`

Запуск `fdisk` в интерактивном режиме `fdisk имя_диска`

Например, `sudo fdisk /dev/sdb`

Основные команды **fdisk**:

**m**- помощь

**a** - включение или выключение флага `boot` для раздела;

**d** - удалить раздел;

**F** - показать свободное место;

**l** - вывести список известных типов разделов;

**n** - создать новый раздел;

**p** - вывести таблицу разделов;

**t** - изменение типа раздела;

**i** - вывести информацию о разделе;

**I** и **O** - записать или загрузить разметку в файл сценария `sfdisk`;

**w** - записать новую таблицу разделов на диск;

**q** - выйти без сохранения;

**g** - создать пустую таблицу разделов GPT;

**o** - создать пустую таблицу разделов MBR.

**Примечание.** Если после записи таблицы разделов `fdisk` команда `lsblk` не показывает созданные разделы, то надо вызвать команду `partprobe`, чтобы уведомить ядро об изменениях таблицы разделов.

Основные команды **parted**:

**help** команда - помощь по выбранной команде;

**mkpart** тип раздела файловая система начало конец - создание раздела linux с файловой системой начиная с позиции начало заканчивая конец, два последних параметра задаются в мегабайтах по умолчанию;

**mkttable** тип - создать таблицу разделов;

**print** - отобразить таблицу разделов;

**quit** - выйти;

**resizerpart** раздел конец - изменить размер раздела;

**rm** раздел - удалить раздел;

**select** раздел - установить раздел как текущий;

**set** раздел флаг состояние - установить флаг для раздела. Состояние может быть on (включен) или off(выключен);

**unit** единицы – установка единиц измерения (s, MiB, GiB, MB, GB)

Утилиту parted можно использовать в командной строке:

parted имя\_диска команды

### 3 Создание файловой системы

ФС создается командой mkfs.тип\_ФС параметры файл\_устройства

Количество индексных дескрипторов для файловых систем семейства ext задается при создании ФС и фиксированно.

Параметр –i задает плотность индексных дескрипторов

Установки, которые применяются при создании ФС по умолчанию, находятся в etc/mke2fs.conf

Чтобы отмонтированный раздел диска был доступен, нужно подключить его к каталогу в дереве ФС (точке монтирования).

Варианты монтирования ФС:

- временное монтирование с помощью mount

- постоянное монтирование с помощью etc/fstab

- монтирование с помощью systemd

Временное монтирование. Команда sudo mount/dev/устройство точка\_монтирования

Точка монтирования должна быть предварительно создана – обычно это пустой каталог (можно использовать каталог /mnt)

Для размонтирования используется один из вариантов:

sudo umount/dev/устройство

или

sudo umount точка\_монтирования

**Примечание.** Чтобы размонтирование прошло успешно, на устройстве не должно быть занятых файловых ресурсов.

Настройка автоматического монтирования ФС может быть выполнена путем соответствующих настроек в /etc/fstab

Файл /etc/fstab содержит следующие поля:

- файл устройства/метка/UUID
- точка монтирования
- тип файловой системы
- параметры (обычно defaults)
- признак для команды dump (обычно 0)
- признак для команды fsck (1 – для корневой файловой системы, 2 – для остальных)

Для монтирования ФС через **systemd** требуется создать юнит типа mount, в котором следует описать какое устройство и как должно быть смонтировано. Название юнита должно совпадать с именем точки монтирования, но вместо символа «/» должен использоваться символ «-».

В юните должна быть секция [mount] со следующими параметрами:

- What - имя устройства (имя файла устройства, метка, UUID)
- Where - точка монтирования
- Type - тип ФС
- Options - параметры монтирования

Утилиты для работы с файловой системой:

- tune2fs - настройка параметров ФС
- dumpe2fs - вывод информации о структуре ФС
- e2fsck – проверка целостности структуры ФС
- resize2fs – изменение размеров ФС
- e4defrag – дефрагментация ФС ext4
- debugfs - отладчик ФС
- e2image - сохранения метаданных ФС в файл
- df (-h, -i) – информация о свободном месте в областях данных и inode
- du -sh каталог – общий размер файлов в каталоге
- lsblk – список блочных устройств

## 4 Управление логическими томами LVM

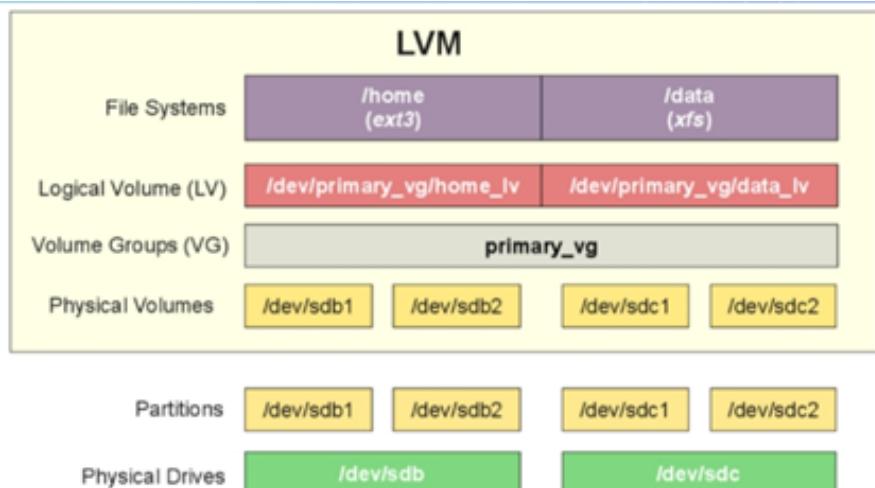


Рисунок 2 – LVM

Работа с томами с помощью LVM происходит на 3-х уровнях абстракции:

1. Физический уровень (**PV**). Сначала диск инициализируется командой **pvcreate** — в начале диска создается дескриптор группы томов. При этом важно заметить, что диск не обязательно должен быть физическим — мы можем отметить на использование обычный раздел диска.

2. Группа томов (**VG**). С помощью команды **vgcreate** создается группа томов из инициализированных на предыдущем этапе дисков.

3. Логический том (**LV**). Группы томов нарезаются на логические тома командой **lvcreate**.

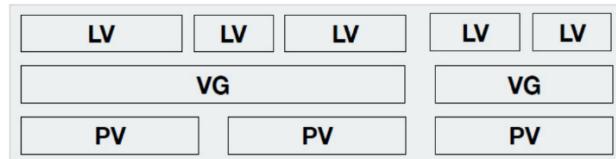


Рисунок 3 – Уровни абстракции

Вывод списка блочных устройств:

```
administrator@astra:~$ sudo lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda     8:0    0   8G  0 disk 
└─sda1  8:1    0   8G  0 part /
sdb     8:16   0   8G  0 disk 
sdc     8:32   0   8G  0 disk 
sdd     8:48   0   8G  0 disk 
sde     8:64   0   8G  0 disk 
sr0    11:0    1  8,6G 0 rom
administrator@astra:~$
```

Установка пакета LVM – sudo apt install lvm2 –y

**Создание физических томов** с помощью pvcreate /dev/sdb

Посмотреть, что диск может использоваться LVM можно командой:

pvdisplay – вывод атрибутов PV, pvscan – сканирование дисков на PV, pvs – вывод информации о PV.

```
administrator@astra:~$ sudo pvdisplay
"/dev/sdc" is a new physical volume of "8,00 GiB"
--- NEW Physical volume ---
PV Name           /dev/sdc
VG Name
PV Size          8,00 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          8snd8A-Hdf8-YLdS-Hwz0-DenI-Hxtq-AdLCqd

"/dev/sdb" is a new physical volume of "8,00 GiB"
--- NEW Physical volume ---
PV Name           /dev/sdb
```

PV Name — имя диска.

VG Name — группа томов, в которую входит данный диск (в нашем случае пусто, так как мы еще не добавили его в группу).

PV Size — размер диска.

Allocatable — распределение по группам. Если NO, то диск еще не задействован и его необходимо для использования включить в группу.

PE Size — размер физического фрагмента (экстента). Пока диск не добавлен в группу, значение будет 0.

Total PE — количество физических экстентов.

Free PE — количество свободных физических экстентов.

Allocated PE — распределенные экстенты.

PV UUID — идентификатор физического раздела.

**Создание групп томов.** Инициализированные на первом этапе диски должны быть объединены в группы. Группа может быть создана:

```
vgcreate vg01 /dev/sdb /dev/sdc
```

**vg01** — произвольное имя создаваемой группы; `/dev/sdb`, `/dev/sdc` — наши диски

Просмотреть информацию о созданных группах можно командой:

```
vgdisplay
```

VG Name — имя группы.

Format — версия подсистемы, используемая для создания группы.

Metadata Areas — область размещения метаданных. Увеличивается на единицу с созданием каждой группы.

VG Access — уровень доступа к группе томов.

VG Size — суммарный объем всех дисков, которые входят в группу.

PE Size — размер физического фрагмента (экстента).

Total PE — количество физических экстентов.

Alloc PE / Size — распределенное пространство: количество экстентов / объем.

Free PE / Size — свободное пространство: количество экстентов / объем.

VG UUID — идентификатор группы.

**Создание логических томов.** Последний этап — создание логического раздела их группы томов командой `lvcreate`. Ее синтаксис: `lvcreate [опции] <имя группы томов>`

Примеры создания логических томов:

```
administrator@astra:~$ sudo lvcreate -L 9G -n lv01 vg01
Logical volume "lv01" created.
```

Информация о логических томах lvdisplay, lvscan, lvs:

LV Path — путь к устройству логического тома.

LV Name — имя логического тома.

VG Name — имя группы томов.

LV UUID — идентификатор.

LV Write Access — уровень доступа.

LV Creation host, time — имя компьютера и дата, когда был создан том.

LV Size — объем дискового пространства, доступный для использования.

Current LE — количество логических экстентов.

**Создание файловой системы и монтирование тома.** Чтобы начать использовать созданный том, необходимо его отформатировать, создав файловую систему и примонтировать раздел в каталог.

Процесс создания файловой системы на томах LVM ничем не отличается от работы с любыми другими разделами.

Например, для создания файловой системы ext4:

```
mkfs.ext4 /dev/vg01/lv01
```

**vg01** — наша группа томов; **lv01** — логический том.

Далее надо создать точку монтирования и смонтировать том

```
administrator@astra:~$ sudo mkdir /data
administrator@astra:~$ sudo mount /dev/vg01/lv01 /data
```

где /dev/vg01/lv01 — созданный нами логический том, /data — раздел, в который мы хотим примонтировать раздел.

Создаем юнит system для монтирования:

```
administrator@astra:~$ sudo vi /etc/systemd/system/data.mount

[Unit]
Description=Mount data

[Mount]
What=/dev/vg01/lv01
Where=/data
Type=ext4
Option=defaults,noexec

[Install]
WantedBy=multi-user.target
```

Проверяем файловую систему на наличие ошибок с помощью fsck.ext4:

```
administrator@astra:~$ sudo fsck.ext4 /dev/vg01/lv01
e2fsck 1.43.4 (31-Jan-2017)
/dev/vg01/lv01: clean, 11/589824 files, 62641/2359296 blocks
```

или e2fsck:

```
administrator@astra:~$ sudo e2fsck -f /dev/vg01/lv01
e2fsck 1.43.4 (31-Jan-2017)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg01/lv01: 11/589824 files (0.0% non-contiguous), 62641/2359296 blocks
```

Чтобы уменьшить размер логического тома, сначала необходимо уменьшить размер файловой системы и затем том с помощью команды resize2fs:

```
administrator@astra:~$ sudo resize2fs -p /dev/vg01/lv01 7G
resize2fs 1.43.4 (31-Jan-2017)
Resizing the filesystem on /dev/vg01/lv01 to 1835008 (4k) blocks.
Begin pass 3 (max = 72)
Scanning inode table XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/vg01/lv01 is now 1835008 (4k) blocks long.
```

Затем уменьшить размер логического тома с 9 Гб до 7Гб

```
lvreduce -L 7G /dev/vg01/lv01
```

**Удаление физического тома.** Перед удалением физического тома надо переместить экстенты с него:

```
administrator@astra:~$ sudo pmove /dev/sdb
/dev/sdb: Moved: 0,56%
/dev/sdb: Moved: 100,00%
```

Командой vgreduce физический том удалить из группы томов:

```
administrator@astra:~$ sudo vgreduce vg01 /dev/sdb
Removed "/dev/sdb" from volume group "vg01"
```

Удалить сам физический том:

```
administrator@astra:~$ sudo pvremove /dev/sdb
Labels on physical volume "/dev/sdb" successfully wiped.
```

**Добавление физического тома.** Для этого необходимо создать физический том sudo pvcreate /dev/sdb.

Добавить /dev/sdb в группу физических томов sudo vgextend vg01 /dev/sdb

Увеличим размер логического тома на 3Гб

```
sudo lvextend -l +3G dev/vg01/lv01
```

Увеличиваем файловую систему

```
sudo resize2fs dev/vg01/lv01
```

**Работа со снапшотами.** Снимки диска позволяют нам откатить состояние на определенный момент. Это может послужить быстрым вариантом резервного копирования. Однако нужно понимать, что данные хранятся на одном и том же физическом носителе, а значит, данный способ не является полноценным резервным копированием.

Необходимо смонтировать файловую систему и создать произвольный файл, по которому можно изучить снимки состояний.

Создать снимок можно : sudo systemctl start data.mount:

```
administrator@astra:~$ sudo systemctl start data.mount
1 /dev/vg01/lv01
Using default stripesize 64,00 KiB.
Logical volume "snap090721" created.
```

Затем надо создать каталог и смонтировать снимок состояния:

```
administrator@astra:~$ sudo mkdir /snap01
administrator@astra:~$ sudo mount /dev/vg01/snap090721 /snap01
```

## Методика выполнения

### 1 Символические и жесткие ссылки

1. Скопируйте файл passwd из каталога /etc в домашнюю директорию пользователя student

2. Создайте символьическую ссылку symlink на скопированный файл passwd

3. Создайте две жестких ссылки hardlink и hardlink1 на скопированный файл passwd. Проверьте что ссылки работают.

4. Удалите оригинал /home/student/passwd. Что изменилось, опишите.

5. Посмотрите номер inode оставшихся жестких ссылок. Как найти все жесткие ссылки, если они находятся в разных директориях, и вы не знаете в каких. Покажите это на примере ваших жестких ссылок.

## 2 Работа с жесткими дисками

1. Используйте fdisk -l, чтобы посмотреть информации о дисках и разделах на вашем компьютере. Какие из ваших дисков в системе имеют разметку? Описать. Посмотрите свободное место на этих разделах с помощью команды df

2. Добавьте 2 новых жестких диска в виртуальную машину (SCSI и IDE). Для добавления новых дисков виртуальную машину нужно выключить. После добавления новых дисков загрузите Astra Linux. Далее проверьте, что новые диски появились в системе.

Например, один из наших новых дисков определился в системе как /dev/sdc.

Используя команду fdisk, создайте два новых раздела (/dev/sdc1 и /dev/sdc2) на данном диске, размером 512 М. Перезагрузитесь, чтобы удостовериться, что вы корректно изменили разделы на жестком диске.

3. Используя команду mke2fs или команду mkfs.ext2, создайте новую файловую систему ext2 на новом логическом разделе /dev/sdc1.

4. На разделе /dev/sdc2 создайте файловую систему ext4

5. Создайте директорию /data, в которую вы будете монтировать новый логический том /dev/sdc2.

6. Используйте команду `mount`, чтобы смонтировать новый логический том в директорию `/data`. Скопируйте `/etc/passwd` в директорию `/data` и проверьте, что копирование было успешно.

7. Затем размонтируйте директорию. Еще раз проверьте содержимое каталога `/data`.

8. Добавьте метку `/data` к новому разделу с помощью команды `e2label`.

9. Отредактируйте файл `/etc/fstab` так, чтобы новый раздел монтировался при

загрузке системы. Проверьте, что вы правильно прописали данную строку в файл `fstab`

10. Перезагрузите систему и убедитесь, что новый раздел монтируется автоматически.

11. Посмотрите свойства файловой системы для раздела `/dev/sdc1`. Включено ли журналирование данной файловой системы? Если нет, то включите журналирование для раздела `/dev/sdc1`. Создайте для данного раздела точку монтирования `/data1` и смонтируйте туда этот слайс.

12. Перейдите в раздел `/data1` и попробуйте размонтировать его. Почему вы не можете размонтировать данный раздел?

13. Посмотрите, кто из пользователей, и какими процессами занял раздел `/data1`. Завершите все процессы на разделе `/data1`

14. Попробуйте размонтировать раздел `/data1` снова. Что произошло?

15. Увеличьте резервируемое место (`minfree`) файловой системой до 10% на разделе `/dev/sdc1`. Осуществите проверку файловой системы для раздела `/dev/sdc1`

### 3 LVM

Для данного раздела нужно установить пакет `lvm2` (если он у вас не установлен).

1. Создайте три новых раздела на диске по 512 М. При необходимости добавьте в систему новый диск (если вы работаете на виртуальной машине). Предположим, что в нашем случае мы добавили новый диск и он был распознан системой как /dev/sda (соответственно, новые разделы будут /dev/sda1, /dev/sda2 и /dev/sda3). Покажите и опишите свой диск и разделы. Разделы на диске должны иметь тип LVM

2. LVM строится на основе разделов жёсткого диска и/или целых жёстких дисков. Поэтому на первых двух из созданных нами разделов (/dev/sda1, /dev/sda2) создайте физический том (physical volume). Проверьте, что физические тома созданы корректно.

3. На первых двух физических томах (/dev/sda1 и /dev/sda2) создаём группу томов, которая будет называться, например, vg1. Проверьте, что группа vg1 создана корректно. Групп можно создать несколько, каждая со своим набором томов. Но обычно это не требуется.

4. В группе томов создайте логический том lv1 размером 200 Мб и lv2 размером 300 Мб. Проверьте, что том создан корректно.

5. Теперь у вас есть блочные устройства /dev/vg1/lv1 и /dev/vg1/lv2 . Создайте на них файловую систему: на lv1 – ext4 и на lv2 – ext3.

**Примечание.** Удаление LVM (или отдельных его частей, например, логических томов или групп томов) происходит в обратном порядке - сначала нужно отмонтировать разделы, затем удалить логические тома (lvremove), после этого можно удалить группы томов (vgremove) и ненужные физические тома (pvremove). **На данном шаге удалять ничего не нужно!**

6. Смонтируйте созданные логические тома lv1 и lv2 в директории /lmdir1 и /lmdir2 (предварительно их создав). Посмотрите информацию по смонтированным томам, опишите. Выполните настройки для автоматического мониторинга данных томов во время запуска системы. Для этого внесите изменения в файл /etc/fstab. Проверьте корректность правок в файле /etc/fstab без перезагрузки.

7. Чтобы добавить раздел /dev/sda3 в группу томов, создайте физический том для этого раздела. Далее добавьте его в группу.

8. Увеличьте размер существующего тома lv2 за счет физического тома, добавленного в группу. Для этого размонтируйте том lv2 и проверьте его на наличие ошибок. Увеличьте lv2 на 512М. После увеличения выполните проверку файловой системы на наличие ошибок. Подмонтируйте измененный том и проверьте, изменился ли его размер.

9. Чтобы убрать из работающей группы томов раздел /dev/sda2 сначала перенесите все данные с него на другие диски, затем удалите его из группы томов, после этого удалите физический том.

## 4 Swap

1. Посмотрите текущие параметры swap-а в системе swapon –s
2. На одном из дисков (например, /dev/sdc) выделите отдельный раздел, например, /dev/sdc3 (использовать под swap). Размер раздела можно задать произвольно. Не забудьте изменить id раздела (Linux swap). В подменю команды fdisk: n – новый раздел; w-сохранение разбиения на диск; t – id раздела (для swap ставим 82).

3. Создайте swap из нового раздела командой mkswap, подключите созданный swap-раздел, проверьте, что swap-раздел был добавлен к основному swap.

**Примечание.** Если на диске нет места под создание новых разделов, можно создать файл для использования его в качестве swap.

4. Создайте файл определенного размера, например, 500М  
`dd if=/dev/zero of=/swapfile bs=1M count=500`
5. Создайте swap из нового файла командой mkswap. Подключите созданный swap-файл.
6. В выводе команды top или команды free должно появиться упоминание, что swapинга в системе прибавилось. Проверьте это.

7. Отключите файл подкачки swapoff /swapfile

8. Чтобы не подключать swap-файл или swap-раздел каждый раз, добавьте соответствующую запись в /etc/fstab:

```
# vi /etc/fstab
```

```
/dev/sdc3 none swap sw 0 0
```

```
/swapfile none swap sw 0 0
```

9. Установите, наивысший приоритет (p1) для файла подкачки командой swapon

10. Произведите очистку swap-пространства на вашем компьютере

11. Измените параметр swappiness временно (до перезагрузки системы):

```
echo 50 > /proc/sys/vm/swappiness
```

Также измените значение по умолчанию, для этого необходимо изменить параметр

```
vm.swappiness в файле /etc/sysctl.conf
```

```
vm.swappiness=50
```

Следует отметить, что при больших значениях система потеряет в отзывчивости (будет вытеснять память, с которой работают приложения, в свою очередь оперативной памяти ещё много). При малых значениях система работает отзывчивей, но когда оперативная память заканчивается, система начинает активно свопиться и притормаживать.

12. Также можно попробовать увеличить\уменьшить объём потребляемой системой памяти за счёт изменения размеров дискового кеша. Уровень выделяемой под кеш памяти хранится в /proc/sys/vm/vfs\_cache\_pressure. Значение по умолчанию: 100.

Чтобы использовать меньше памяти под дисковые кеши (не желательно), поставьте значение 50. Если, наоборот, хочется больше отзывчивости системы, увеличьте размер кеша:

```
echo 1000 > /proc/sys/vm/vfs_cache_pressure
```

Измените параметры до полного удовлетворения. Для того, чтобы настройки стали постоянными, занесите нужный параметр в файл /etc/sysctl.conf

```
# vi /etc/sysctl.conf  
vm.vfs_cache_pressure = 1000
```

Привести скриншот и пояснить работу данного пункта.

### **Контрольные вопросы**

1. Какой файл устройства соответствует 2-ому разделу 3-его SATA диска?
2. Какая команда помогает найти причину, почему не получается размонтировать ФС?
3. Какую утилиту следует использовать, чтобы создать на разделе ФС типа ext3?
4. В каком порядке следует уменьшать размер логического тома?
5. В каком порядке следует увеличивать размер логического тома?
6. Что следует сделать перед удалением физического тома?
7. Основные характеристики ФС iso9660 и udf.

## **Лабораторная работа № 6**

### **Шифрование дисков LUKS**

**Цель работы** – научится выполнять шифрование разделов/дисков, проводить мониторинг жестких дисков.

#### **Теоретические сведения**

##### **1 Проверка работоспособности жесткого диска**

Утилита **smartmontools** - предназначена для проверки состояния жестких дисков при помощи SMART (Self-Monitoring Analisys and Reporting Technology - в современных жестких дисках встроенный модуль самоконтроля S. M. A. R. T., который анализирует данные накопителя и помогает определить неисправность на первоначальной стадии). Так же может осуществлять проверку в постоянном режиме и оправлять уведомления по почте.

Smartmontools состоит из двух утилит — smartctl и smartd.

Подробнее о Smartmontools можно узнать на сайте разработчиков, там же можно и скачать последние версии данного программного обеспечения: <http://www.smartmontools.org>.

Для работы необходимо установить пакет:

```
sudo aptitude install smartmontools
```

Утилита **hdparm** предназначена для установки/ получения различных параметров SATA/IDE устройств, к которым относятся жесткие диски. Утилита может установить объём кеш-памяти накопителя, перевести жёсткий диск в спящий режим, управлять питанием и акустикой и изменять настройки DMA. Обычно Hdparm применяется для оптимизации жёсткого диска, для повышения его производительности, активации многорежимности IDE.

Для работы необходимо установить пакет:

```
sudo aptitude install hdparm
```

## **2 Шифрование диска**

Методы шифрования:

- шифрование на уровне файловой системы: eCryptfs, ENCfs
- блочное шифрование на уровне устройства: Loop-AES, TrueCrypt, dm-crypt+LUKS (Linux Unified Key Setup)

eCryptfs - это криптографическая файловая система Linux. Она хранит криптографические метаданные для каждого файла в отдельном файле, таким образом, что файлы можно копировать между компьютерами. Файл будет успешно расшифрован, если у вас есть ключ.

2. EncFS - обеспечивает шифрованную файловую систему в пространстве пользователя. Она работает без каких-либо дополнительных привилегий и использует библиотеку fuse и модуль ядра для обеспечения интерфейса файловой системы. EncFS - это свободное программное обеспечение и она распространяется под лицензией GPL.

Loop-AES - быстрая и прозрачная файловая система, а также пакет для шифрования раздела подкачки в Linux.

TrueCrypt - это бесплатное решение с открытым исходным кодом для шифрования диска

dm-crypt+LUKS - dm-crypt - это прозрачная подсистема для шифрования диска, поддерживается шифрование целых дисков, съемных носителей, разделов, томов RAID, программного обеспечения, логических томов и файлов.

LUKS (Linux Unified Key Setup - протокол шифрования блочного устройства. Чтобы выполнить шифрование диска linux используется модуль ядра dm-crypt. Этот модуль позволяет создавать в каталоге /dev/mapper виртуальное блочное устройство с прозрачным для файловой системы и пользователя шифрованием. Фактически все данные лежат на зашифрованном физическом разделе. Если пользователь пытается записать данные на виртуальное устройство, они на лету шифруются и записываются на диск, при

чтении с виртуального устройства, выполняется обратная операция - данные расшифровываются с физического диска и передаются в открытом виде через виртуальный диск пользователю. Обычно для шифрования используется метод AES, потому что под него оптимизированы большинство современных процессоров. Важно заметить, что вы можете шифровать не только разделы и диски, но и обычные файлы, создав в них файловую систему и подключив как loop устройство.

Алгоритм LUKS определяют какие действия и в каком порядке будут выполняться во время работы с шифрованными носителями. Для работы с LUKS и модулем dm-crypt используется утилита Cryptsetup.



Рисунок1 - Формат раздела LUKS

Утилита Cryptsetup предназначена для управления шифрованием дисков, с помощью которой можно:

- создавать шифрованные разделы LUKS;
- открывать/закрывать разделы LUKS;
- управлять слотами ключей;
- делать дамп заголовка LUKS и мастер-ключа.

Установка: `sudo apt install cryptsetup`

Синтаксис команды:

`cryptsetup [опции] [операции] <параметры>`

Операции, которые можно сделать с помощью этой утилиты:

`luksFormat` - создать зашифрованный раздел `luks linux`;

`luksOpen` - подключить виртуальное устройство (нужен ключ);

`luksClose` - закрыть виртуальное устройство `luks linux`;

`luksAddKey` - добавить ключ шифрования;

`luksRemoveKey` - удалить ключ шифрования;

`luksUUID` - показать UUID раздела;

luksDump - создать резервную копию заголовков LUKS.

В начале выполнения шифрования жесткого диска надо выполнить инициализацию раздела и установку пароля. При этом будет предупреждение об уничтожении данных:

```
administrator@astra:~$ sudo cryptsetup luksFormat /dev/sdd
WARNING!
=====
This will overwrite data on /dev/sdd irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
```

Далее необходимо открыть LUKS-том:

```
administrator@astra:~$ sudo cryptsetup luksOpen /dev/sdd disk1
Enter passphrase for /dev/sdd:
```

И теперь на разделе можно создать файловую систему и смонтировать ее.

**Примечание.** Один LUKS-раздел может открываться одним из 8 возможных ключей. А также можно использовать единственный ключ в одном слоте. Чтобы узнать состояние всех слотов, применяется команда: cryptsetup luksDump

Для добавления **нового ключа** LUKS на зашифрованный раздел используется команда: cryptsetup luksAddKey

При запросе «Enter any passphrase» требуется ввести один из уже имеющихся паролей для LUKS. Далее нужно ввести новый пароль, который займет новый слот соответственно (слот 1).

Чтобы **удалить** какой-то определенный **ключ** LUKS, надо знать парольную фразу одного из слотов, с помощью: cryptsetup luksKillSlot и проверить с помощью : cryptsetup luksDump.

Можно добавить бинарный ключ или записать пароль в текстовый документ и добавить к ключам (**добавление ключа из файла**). Для этого надо создать 256-битный ключ:

```
administrator@astra:~$ sudo dd if=/dev/random of=/my.key bs=1 count=256
256+0 записей получено
256+0 записей отправлено
256 байт скопировано, 240,125 s, 0,0 kB/s
```

и записать его в один из слотов.

После того, как закончена работа с секретными файлами на зашифрованном устройстве, надо размонтировать файловую систему и полностью закрыть диск.

```
administrator@astra:~$ sudo umount /disk1
administrator@astra:~$ ls /dev/mapper/disk1
/dev/mapper/disk1
administrator@astra:~$ sudo cryptsetup luksClose disk1
```

Обратите внимание, что после размонтирования директории, виртуальное устройство /dev/mapper/disk1 еще присутствует в системе.

Далее, в следующем сеансе работы с зашифрованным разделом, его нужно **открыть** с помощью ключа-пароля или с помощью ключа, записанного в файл.

Чтобы выполнить автоматическое монтирование раздела LUKS пользователь должен ввести пароль во время загрузки. В этом случае в файлы etc/fstab и etc/crypttab добавляется следующая информация:

```
administrator@astra:~$ cat /etc/crypttab
# <target name> <source device>          <key file>      <options>
disk1      /dev/sdd                      none           luks

administrator@astra:~$ cat /etc/fstab
UUID=9a332120-6162-49f8-b884-67c78439fda7      /      ext4      errors=remount-ro,
usrquota,grpquota,secdelrnd=6      0      1
UUID=3af25722-abb9-4a4b-a045-51b5ad0f5e14    none      swap      sw      0      0
/dev/sr0   /media/cdrom0  udf,iso9660  user,noauto  0      0
/dev/mapper/disk1   /disk1   ext4      defaults      0      0
```

Так же можно смонтировать раздел с помощью ключа, который надо **хранить на отдельном носителе!** В этом случае в файле etc/crypttab нужно указать путь к ключу.

## Методика выполнения

### 1 Проверка работоспособности жесткого диска

1. Проверить общее состояние диска с помощью smartctl.
2. Посмотреть дополнительную информацию по диску с помощью smartctl.

3. Выполнить расширенный тест диска с помощью smartctl.
4. Распечатать журналы ошибок диска с помощью smartctl.
5. Посмотреть информацию о диске с помощью утилиты hdparm.
6. Посмотреть текущие настройки для различных флагов диска с помощью утилиты hdparm.

## 2 Шифрование диска

1. Установите cryptsetup-luks
2. Создайте новый раздел на диске (можно использовать весь диск)
3. Отформатируйте раздел (диск) LUKS, например /dev/sdc1.
4. Подключите зашифрованный диск
5. Создайте файловую систему на подключенном диске
6. Создайте директорию для монтирования зашифрованного раздела и смонтируйте зашифрованный раздел в нее
7. Просмотрите список используемых ключей. Сколько свободных слотов для ключей присутствует?
8. Добавьте ключевую фразу к слоту
9. Добавьте ключевой файл
10. Разблокируйте зашифрованный раздел диска при помощи ключевого файла.

11. Удалите один из ключей
12. Чтобы операционная система сама научилась подключать и монтировать нужные криптованные устройства во время загрузки, а затем корректно отключать их во время останова системы, добавьте по одной строке в файлы /etc/crypttab и /etc/fstab:

```
# vi /etc/crypttab
lkfs /dev/sda5 none luks,cipher=aes-cbc-essiv:sha256"
# vi /etc/fstab
/dev/mapper/lkfs /mnt/lkfs ext4 defaults 0 0
```

Теперь во время каждой загрузки ОС будет спрашивать пароль для доступа к криптованному разделу, если он будет указан неправильно – загрузка остановится.

#### 13. Выполните шифрование домашнего каталога.

Шифрование домашнего каталога производится по точно такой же схеме с тем лишь исключением, что перед добавлением новой записи в /etc/fstab следует удалить старую запись, ссылающуюся на /home.

#### 14. Выполните шифрование флешки.

При создании шифрованной флешки специальные записи в /etc/crypttab и /etc/fstab не требуются. Подсистема HAL сама определит наличие на устройстве хранения LUKS-раздела и передаст эту информацию среде рабочего стола (Gnome, KDE, XFCE), которая, в свою очередь, выведет на экран окно с просьбой ввести пароль. Единственное, что необходимо сделать – при первом монтировании изменить права доступа на ее корневой каталог:

```
$ sudo chown -R student:student /media/usb_name
```

```
$ sudo chmod g+s /media/usb_nam
```

### **Контрольные вопросы**

1. В каком файле необходимо указывать путь к ключу для автоматического монтирования зашифрованного раздела/диска?
2. Сколько слотов для хранения ключей содержит LUKS?
3. С помощью какой команды можно получить доступ к зашифрованному разделу/диску (открыть раздел/диск)?
4. Если вы планируете осуществлять мониторинг жестких дисков, какой пакет нужно установить?
5. Какая утилита ориентирована на работу со SCSI устройствами (включая SATA, IEEE1394 и USB)?
6. С помощью какой утилиты можно уменьшить шум от диска?

# Лабораторная работа № 7

## ПАМ - идентификация и аутентификация пользователей в AstraLinuxSE

**Цель работы** – понимать, как работает система ПАМ и как составлять конфигурационные файлы.

### Теоретические сведения

ПАМ (Pluggable Authentication Modules, подключаемые модули аутентификации) – это набор библиотек для Linux и Unix-подобных операционных систем, предназначенный для настройки аутентификации между различными приложениями, рисунок 1.



Рисунок 1 – Модуль ПАМ

До появления ПАМ такие приложения, как login (а также rlogin, telnet, rsh), искали имя пользователя в файле /etc/passwd, сравнивали оба значения и выполняли аутентификацию пользователя по введенному им имени. Все приложения использовали эти общие службы, хотя детали реализации и полномочия для их настройки различались.

Затем разработчики приложений попытались создавать собственные процессы. Для этого понадобилось разделить модули приложения и безопасности (общий модуль безопасности мог использоваться всеми приложениями и настраиваться как необходимо).

Существует три реализации ПАМ:

- Linux-PAM: охватывает все типы PAM, обсуждаемые в этой статье.

Основная архитектура PAM на любой платформе Linux подобна версии Linux-PAM.

- OpenPAM - другая реализация PAM, разработанная Дагом-Эрлингом Сморгравом (Dag-Erling Smorgrav) в NAI Labs, в рамках программы исследований DARPA-CHATS. Поскольку это реализация с открытым кодом, она в основном используется во FreeBSD, NetBSD и приложениях (а также в Mac OS X).

- Java™ PAM или JPat: PAM – это фактически стандартные модули аутентификации с поддержкой Linux и UNIX. JPat выступает в качестве моста между кодом Java и обычными PAM. При помощи JPat приложения на Java могут использовать одули PAM и связанные с ними средства (такие как auth, account, passwd, session, т. п.).

Несмотря на существование различных реализаций PAM их основная функциональность остается одинаковой.

Механизм PAM (Pluggable Authentication Modules — подключаемые модули аутентификации) позволяет интегрировать различные низкоуровневые методы аутентификации и предоставить единые механизмы для использования прикладных программ в процессе аутентификации. Механизм состоит из набора разделяемых библиотек и конфигурационных файлов — сценариев процедур аутентификации.

В каталоге /etc/pam.d расположены конфигурационные файлы PAM для соответствующих сервисов, в т. ч. и для login (авторизованный вход в систему). В конфигурационном файле сервиса дана информация по проведению аутентификации.

Рассмотрим систему PAM подробнее. Основные функции, или действия, или задачи, которые выполняет система PAM - разбиты на четыре группы которые имеют определенные названия:

- группа auth - это действия, связанные непосредственно с аутентификацией. То есть действия или функции, которые позволяют

определить, что вы это вы. Это может быть аутентификация по паролю, по смарт-карте, биометрическая аутентификация (отпечаток пальца и т.д.) и другие.

- группа account - это действия, связанные с управлением учетными записями.

Например, даже если вы аутентифицировались в системе, то вашей учетной записи можно поставить запрет на работу в определенное время суток. Или разрешить заходить в консольном режиме, но запретить заходить в графическом режиме и т.д.

- группа session - действия этой группы осуществляют выделение пользователю необходимых для работы ресурсов. Самый простой пример - это разрешение на монтирование каталогов.

-группа password - действия, которые реализуют изменение аутентификационных данных пользователя. Чаще всего это действия по управлению паролями пользователя.

Для выполнения каждой функции может быть перечислено несколько модулей PAM, которые будут вызываться последовательно, образуя стек PAM для данной задачи.

Каждый вызываемый модуль возвращает в стек результат своей работы: успешный (PAM\_SUCCESS), неуспешный (PAM\_AUTH\_ERR), игнорирующий (PAM\_IGNORE) или иной.

Для каждого вызова может быть указан набор управляющих флагов в виде соответствия кода возврата и того, как результат работы модуля скажется на обработке всей сервисной задачи, например, ignore, ok, die.

Основной конфигурационный файл системы PAM - это файл /etc/pam.conf. Кроме файла /etc/pam.conf, настройки PAM хранятся в файлах каталога /etc/pam.d/. Внутри каталога находятся текстовые файлы которые содержат в себе последовательность действий (алгоритм) для программ которые используют PAM. Например, файл /etc/pam.d/login содержит алгоритм работы системы PAM для программы login, а файл /etc/pam.d/passwd

для программы passwd.

Рассмотрим сначала формат файла /etc/pam.conf. Файл состоит из строк. Файл может состоять из одной строки, а может из нескольких строк складываясь в цепочку последовательных действий. Каждая строка описывает одно правило или один шаг такой цепочки (алгоритма). Стока состоит из четырех полей. Первое поле — это имя программы к которой относится данный шаг. Второе поле, это тип действия (auth, account, session, password). Третье поле — это поле в котором задается поведение системы PAM после завершения этого шага на этом шаге алгоритма (чуть ниже остановимся подробнее на этом вопросе). Четвертое поле - это имя файла модуля. Также в строке могут присутствовать некоторые параметры, передаваемые модулю.

Структура файлов находящихся в каталоге /etc/pam.d/, такая же. Отличие только в тсутствии первого поля - имени. Так как имя программы берется из имени самого файла.

Посмотрим на пример такого файла. Назовем его testpam.

1	auth	sufficient	pam_root
2	auth	required	pam_unix.so
3	account	required	pam_unix.so

Рассмотрим первую строку. Поле auth говорит, что первым шагом будет аутентификация. Третье поле - это модуль, который будет выполнять аутентификацию и возвращать результат выполнения. В данном примере модуль pam\_rootok.so проверяет соответствует ли учетная запись пользователю root. Если, да то будет возвращен успех (true), если нет, то будет возвращена ошибка или отказ (false). Второе поле - это реакция или влияние полученного результата на цепочку в целом.

Реакция может быть четырех типов: required, requisite, optional, sufficient.

На примере строки auth sufficient pam\_rootok.so рассмотрим, что означают эти значения. Если во втором поле установлено значение requisite, то это означает, что если модуль pam\_rootok.so завершился с ошибкой, то

дальнейшее выполнение файла testpam прерывается и система PAM возвращает приложению ошибку. Если модуль вернул положительные результат, то выполнение цепочки продолжается.

`required` похож на `requisite`. Если модуль `pam_rootok.so` завершился с ошибкой, то PAM также вернет ошибку, но после того как будут выполнены остальные модули, то есть цепочка не прерывается. Если модуль вернул положительные результат, то выполнение цепочки продолжается.

`sufficient` - если модуль `pam_rootok.so` вернул успех, то система PAM возвращает приложению успех, и дальнейшее выполнение цепочки прерывается. Если неудача, то продолжается выполнение цепочки.

`optional` - этот параметр никак не влияет на ход цепочки. Указывается для тех модулей, которые не выполняют никаких проверочных действий. Если в файле будут только строки с параметром `optional`, то PAM вернет приложению успех.

PAM обеспечивает различные функциональные возможности, такие как: аутентификация с однократной регистрацией, управление доступом и другие. Их реализация обеспечивается различными модулями:

- `pam_access` обеспечивает управление входом в систему в виде протоколируемой службы при помощи имени пользователя и домена в зависимости от правил, указанных заранее в файле `/etc/security/access.conf`.

- `pam_cracklib` проверяет пароли на соответствие правилам для паролей.
- `pam_env` `sets/unsets` устанавливает и сбрасывает переменные среды из файла `/etc/security/pam_env.conf`.

- `pam_debug` выполняет отладку PAM.
- `pam_deny` блокирует модули PAM.
- `pam_echo` выводит сообщения.
- `pam_exec` выполняет внешнюю команду.
- `pam_ftp` модуль для анонимного доступа.
- `pam_localuser` проверяет наличие имени пользователя в файле `/etc/passwd`.

- pam\_unix выполняет обычную аутентификацию на основе пароля из файла /etc/passwd.

Существует множество других модулей (pam\_userdb, pam\_warn, pam\_xauth),), перехватывающих набор возвращаемых значений.

## **Методика выполнения и контрольные вопросы**

1. Создать собственное РАМ-приложение, которое поможет понять, как работает сеанс РАМ. Для этого:

-включите заголовочные файлы, необходимые для использования РАМ (например, pam\_appl.h, pam\_misc.h);

-в функции main инициализируйте библиотеку РАМ libpam.so (которая загружает модули, указанные в файле конфигурации для приложения) при помощи уникального идентификатора;

- попытайтесь выполнить аутентификацию для всех модулей и рассмотрите сценарии отказов;

- проверьте учетные данные пользователей и параметры учетных записей;

- откройте новый сеанс РАМ;

- создайте среду для пользователя при помощи учетных данных;

- по завершении работы пользователя закройте пользовательскую среду;

- закройте сеанс РАМ;

- выйдите из библиотеки libpam.so с соответствующим идентификатором;

- выход.

2. Вы должны находиться под пользователем root/ Перейдите в каталог /etc/pam.d/. Перенесите файл su в домашнюю директорию (чтобы можно было восстановить его). Выполните команду su из под пользователя student в терминале, чтобы перейти в режим суперпользователя. После ввода пароля система выдаст ошибку аутентификации, так как отсутствует

конфигурационный файл для программы su, сделать скриншот.

3. Создать файл /etc/pam.d/su и написать в нем такую строку:

auth sufficient pam\_permit.so Сохранить.

Снова выполнить команду su из под пользователя student, что произошло?

Скриншот. Это произошло потому, что модуль pam\_permit.so всегда возвращает положительный результат, sufficient тут же прерывает выполнение цепочки и система РАМ возвращает положительный результат.

Отредактируйте файл к следующему виду:

auth required pam\_permit.so

auth requisite pam\_deny.so

auth sufficient pam\_permit.so

Модуль pam\_deny.so всегда возвращает ошибку. Какой будет результат?

Проверьте. А если заменить requisite на required? Скриншоты.

4. Напишите в файле следующее правило:

auth required pam\_unix.so

После выполнения команды su будет запрошен пароль пользователя root. Если пароль ввести правильно, то вы станете root-ом, если пароль будет неверный, то останетесь обычным пользователем. Скриншоты.

5. Добавьте в файл еще одну строку так, чтобы получились следующие правила:

auth requisite pam\_wheel.so

auth required pam\_unix.so

Модуль pam\_wheel.so возвращает успех если учетная запись пользователя принадлежит группе wheel (в некоторых версиях группе root).

Выполнить команду su, что произошло? Скриншот. Команду su смогут выполнить только пользователи, который входят в группу wheel и знают пароль учетной записи root.

Создайте группу wheel и добавьте туда свою учетную запись, выполните команду su. Скриншот.

После выполнения задания удалите из группы учетную запись и удалите

группу wheel.

6. Измените файл, укажите две строки:

```
auth requisite pam_wheel.so
```

```
auth required pam_permit.so
```

Ответить кто сможет успешно выполнить команду su и, что для этого нужно будет сделать?

После выполнения задания верните на место оригинальный файл su.

7. Конфигурационные файлы в каталоге /etc/pam.d/ можно создавать только для файлов которые используют систему PAM. Создайте файл /etc/pam.d/ls со строкой

```
auth requisite pam_deny.so
```

```
# touch /etc/pam.d/ls
```

```
# vi /etc/pam.d/ls
```

```
auth required pam_permit.so
```

Проверьте, будет ли выполняться команда ls? Какой результат?

Скриншоты.

8. Если команда не использует систему PAM, то создание конфигурационного файла в директории pam.d ни на что не повлияет.

Проверить использует ли команда систему PAM позволяет команда ldd, которой в качестве параметра передается полный путь к файлу команды.

Команда ldd покажет какие библиотеки использует программа и если в перечне есть libpam.so.0, libpam\_misc.so.0 значит программа использует систему PAM.

Проверьте использует ли команда su PAM библиотеки? Скриншоты

Протестируйте на выбор другие две команды, доступные пользователю root, скриншоты.

9. Проверьте, используется ли в вашей системе аутентификация через PAM. Для этого посмотрите содержимое файла /etc/nsswitch.conf. Первые три строки этого файла как раз и задают какая система аутентификации будет работать в системе. Ключевое слово compat показывает, что в качестве

системы аутентификации будет использована система PAM. Скриншот.

10. Создайте стандартного пользователя student10. Задайте ему пароль. Измените минимальную длину пароля, вместо 8 символов установите 9. Проверьте, работают ли новые настройки. Для этого войдите в систему под обычным пользователем student10 в текстовом терминале (Ctrl+Alt+F1), и поменяйте пароль используя команду # passwd. Выходите из под пользователя student10.

Усложните пароль (под пользователем root). Обязательными символами в пароле должны быть: минимум 1 прописная буква, минимум одна строчная буква и хотя бы одна цифра. Проверьте что изменения сработали, войдя под пользователем student10.

11. Создайте файл /etc/nologin. Добавьте текстовое сообщение в файл. Попробуйте зайти в систему под обычным пользователем. Почему обычному пользователю удалось или не удалось войти в систему? Удалите файл /etc/nologin.

# **Лабораторная работа № 8**

## **Основы мандатного управления доступом. Настройка параметров мандатного управления доступом и мандатного контроля целостности**

**Цель работы** – изучить особенности администрирования локальных учётных записей пользователей и групп в ОССН (операционная система специального назначения) и основы мандатного управления доступом. Освоить администрирование основных параметров мандатного управления доступом и мандатного контроля целостности в ОССН с применением графических утилит и консольных команд.

### **Теоретические сведения**

#### **1 Общие сведения**

Для улучшения безопасности ОС используются два подхода. Первый, *архитектурный*, заключается в том, что разрабатываются и внедряются различные средства защиты информации еще на этапе проектирования. Эти средства образуют *комплекс средств защиты (КСЗ)*, который реализует функции безопасности для минимизации риска возможных потенциальных угроз.

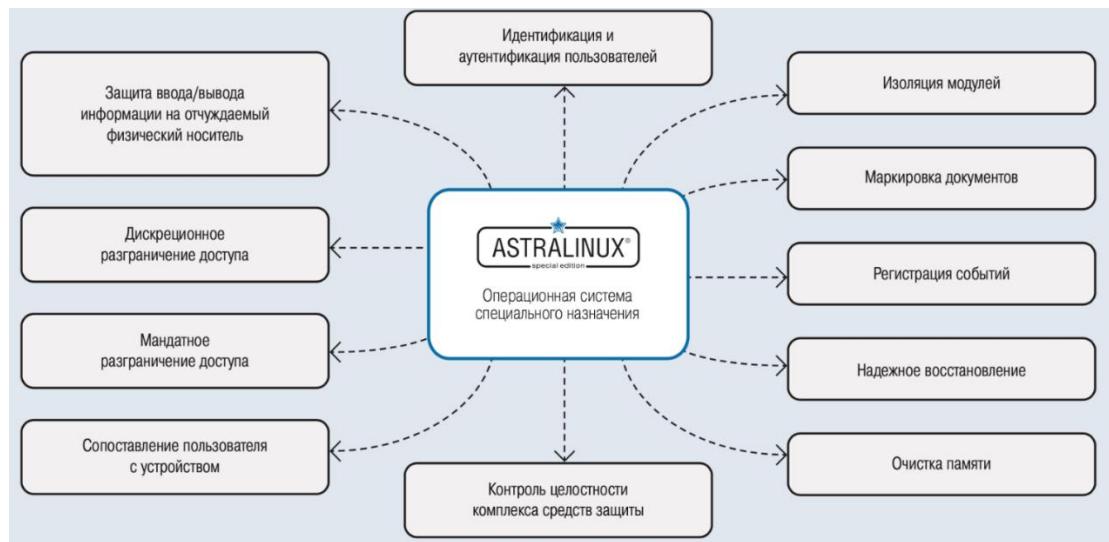


Рисунок 1- Реализованные функции системы защиты информации (CЗИ) от несанкционированного доступа (НСД)

КСЗ: модуль безопасности PARSEC, библиотеки РАМ, утилиты безопасности, система протоколирования, модули аутентификации, графическая подсистема, консольный вход, средства контроля целостности, средства восстановления, средства разграничения доступа.

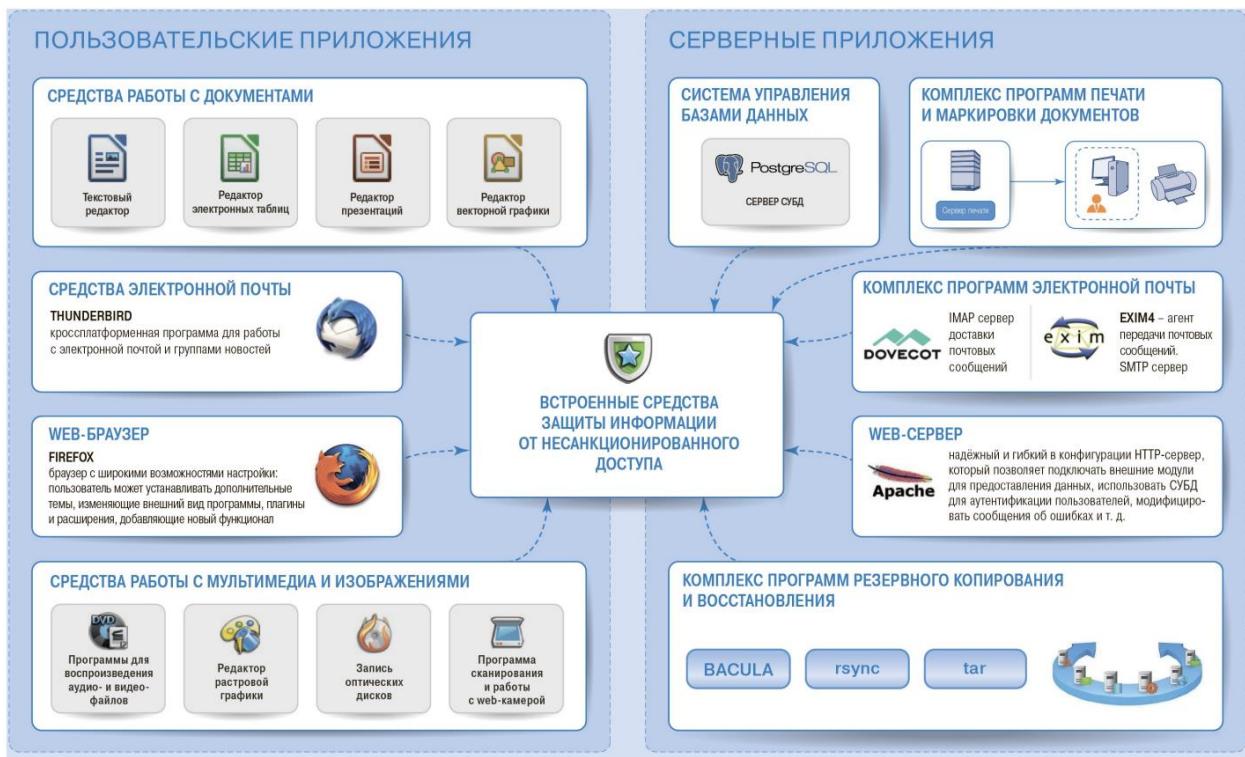


Рисунок 2 – Состав ОС и СЗИ

Второй - *процессный* подход: постоянно выявляются и собираются сведения об уязвимостях, прорабатывается эта информация и передаются результаты в банк данных уязвимостей ФСТЭК России. В результате, готовятся и выпускаются плановые и оперативные обновления ОС.

Мандатная модель управления доступом (MAC) и контроля целостности (MIC) реализуют механизм, когда все компоненты системы иерархически разделяются по степени важности для ее безопасности от самых недоверенных, пользовательских (уровень целостности 0), до системных, административных (уровень целостности по умолчанию 63).

Контроль целостности компьютерной системы — это механизм, необходимый для отслеживания неизменности файлов, документов, реестра, конфигурации оборудования и других сущностей, которые находятся на компьютере или сервере.

Ключевым компонентом ОС является ядро. Соответственно, для него обеспечивается максимально защищенная среда выполнения в самой операционной системе, чтобы уменьшить количество возможных способов атаки на ядро. Для этого реализуется в операционной системе мандатный многоуровневый контроль целостности, за счет чего ОС сегментируется по различным подсистемам — так, чтобы взлом одной подсистемы не повлиял на работоспособность других.

Если произойдет взлом непrivилегированного пользователя ОС (уровень целостности 0) или сетевой подсистемы (уровень целостности 1), системы виртуализации (уровень целостности 2), графического интерфейса (уровень целостности 8) или другого компонента, это не повлечет за собой дискредитацию всего КСЗ (уровень целостности 63).

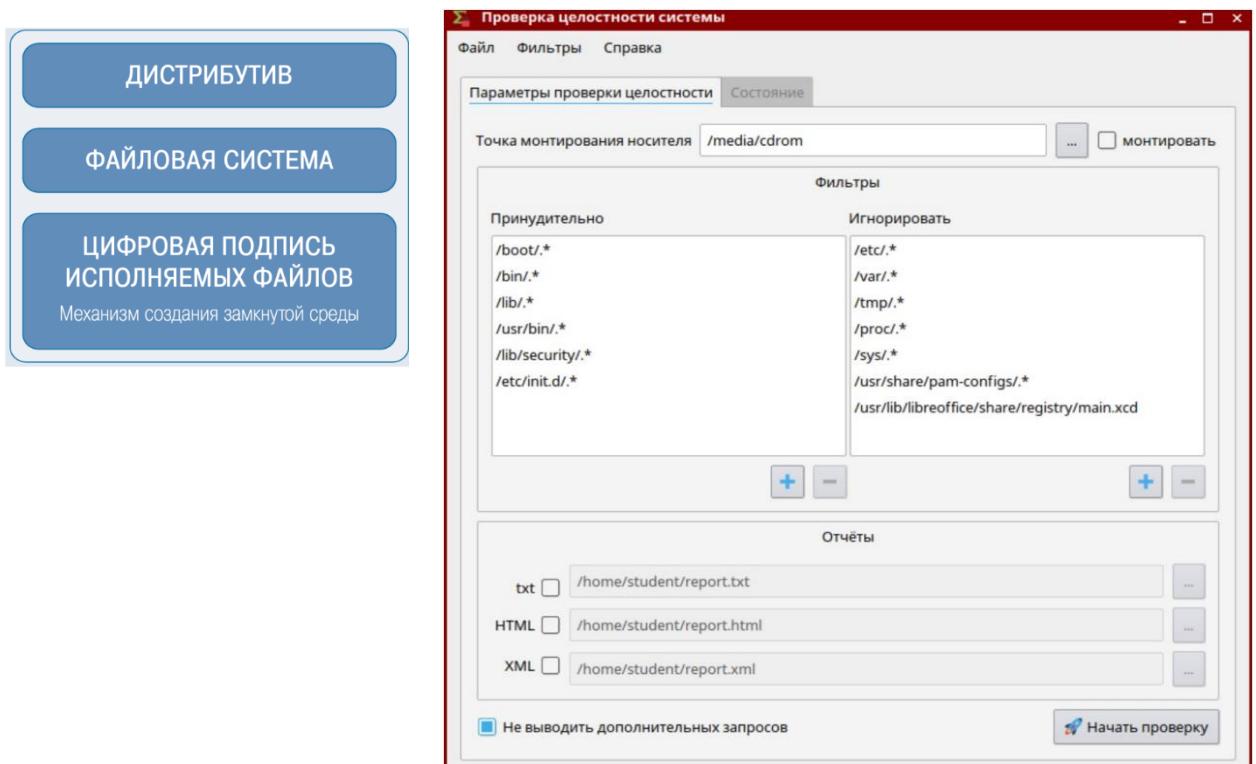


Рисунок 3 – Проверка целостности системы

При этом надо отметить, что указанные уровни не являются иерархическими, то есть не располагаются друг над другом и полностью изолированы друг от друга с точки зрения возможности прав записи. Принадлежность объекта к тому или иному уровню целостности монитор обращений определяет по битовой маске:

	000	0000 0000	Нулевой уровень. "Низкий", или "Low"
1	001	0000 0001	Уровень задействован как "Сетевые сервисы"
2	002	0000 0010	Уровень задействован как "Виртуализация"
3	004	0000 0100	Уровень задействован как "Специальное ПО"
4	008	0000 1000	Уровень задействован как "Графический сервер"
5	016	0001 0000	Свободен, может быть использован для СУБД
6	032	0010 0000	Свободен, может быть использован для сетевых сервисов
7	064	0100 0000	Зарезервирован, и может быть использован при поднятии max_ilev
8	128	1000 0000	Зарезервирован, и может быть использован при поднятии max_ilev

Чтобы уровни целостности не воспринимались как иерархические — то есть, например, «уровень 8 имеет больше прав, чем уровень 2», что неверно — каждый из уровней получает свое наименование. Так, например, восьмой

уровень целостности называется «Графический сервер», максимально возможный уровень целостности администратора в системе — «Высокий», а нулевой уровень целостности (пользовательский) — «Низкий».

Процесс, выполняющийся на низком уровне целостности, не имеет возможности:

- получать доступ к процессам, выполняющимся на более высоких уровнях целостности, в том числе не может направлять управляющие сообщения их окнам;
- порождать процессы, выполняющиеся от имени другой учётной записи пользователя, с использованием механизмов su, sudo, suid/sgid;
- порождать процессы, выполняющиеся на высоком уровне целостности.

Монитор обращений контролирует и исключает возможность влияния друг на друга процессов с метками разных уровней целостности.

Таким образом операционная система получает набор правил, как изолировать друг от друга системные процессы, и теперь понимает, какие именно процессы, даже запущенные пользователем с высокими привилегиями, не имеют права на запись в другие процессы или файлы. Поэтому если в результате эксплуатации уязвимости (в том числе, нулевого дня) злоумышленник получит контроль над каким-либо процессом в системе и повысит свои полномочия до привилегированного пользователя (например, root), его метка целостности останется прежней, и, соответственно, он не получит возможности влиять на системные процессы, менять настройки или скрыть свое присутствие в системе.

Таким образом, значимой мишенью для злоумышленника становится уже не вся операционная система, а только hardened ядро и максимально компактный монитор обращений, что уже существенно сокращает поверхность атаки.

Помимо мандатного, есть еще динамический и регламентный контроль целостности. Их применяют для исключения запуска и использования

недоверенного или стороннего ПО, а также периодических проверок целостности системы.

*Динамический контроль* вычисляет и проверяет электронную цифровую подпись (ЭЦП) исполняемых файлов в момент их запуска. Если ЭЦП нет или она неправильная, в запуске программ будет отказано.

*Регламентный контроль* проверяет целостность и неизменность ключевых для системы файлов, сравнивая их контрольные суммы с эталонными значениями. Это могут быть как конфигурационные файлы, так и любые другие.

Таким образом, в ОС применяется эшелонированная защита от уязвимостей в приложениях и их подмены, чем минимизируется вред от угроз безопасности, в том числе и тех, которые используют уязвимости «нулевого» дня.

## 2 Параметрами мандатного управления доступом

В ОС реализована комбинация:

- дискреционного (избирательного) разграничения доступа (Discretionary Access Control, DAC),
- ролевого разграничения доступа (Role Based Access Control, RBAC),
- мандатного (принудительного, обычно многоуровневого) разграничения доступа (Mandatory Access Control, MAC).



Рисунок 4 – Виды разграничения прав доступа

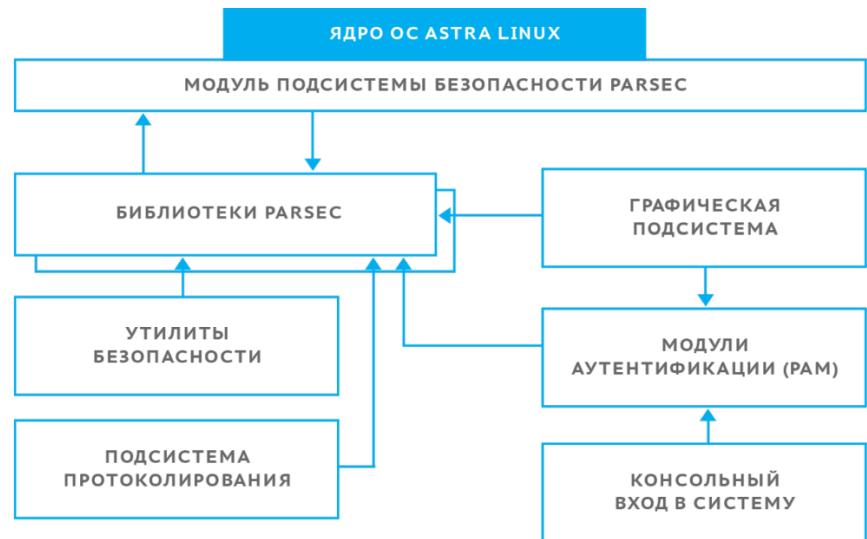


Рисунок 5 – Взаимодействие модуля подсистемы PARSEC

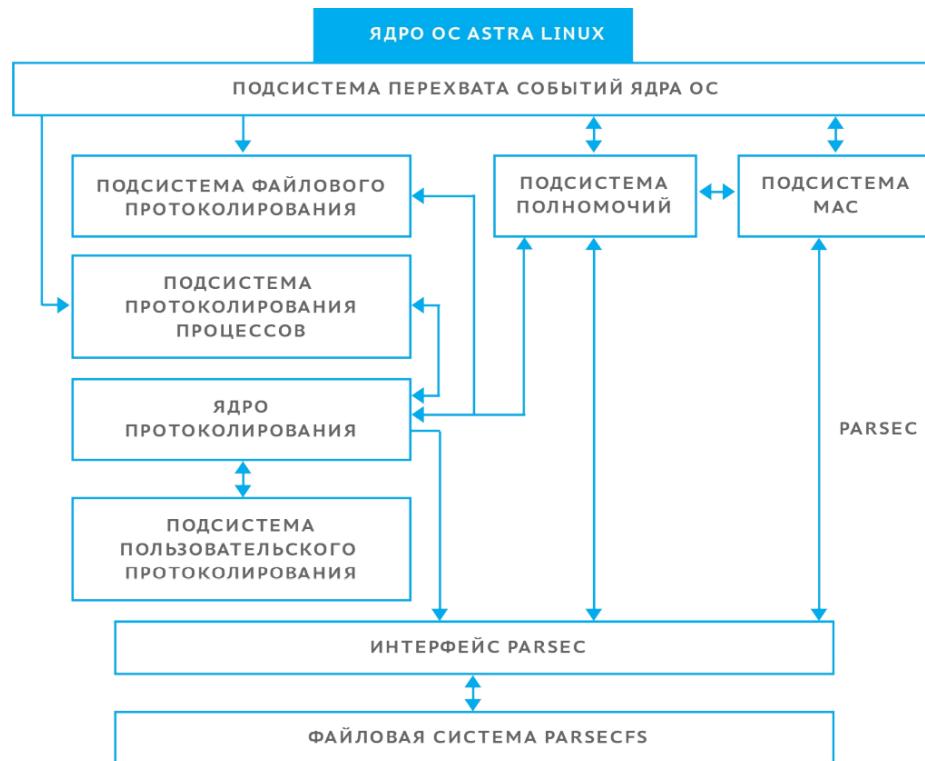


Рисунок 6 – Мандатное управление доступом

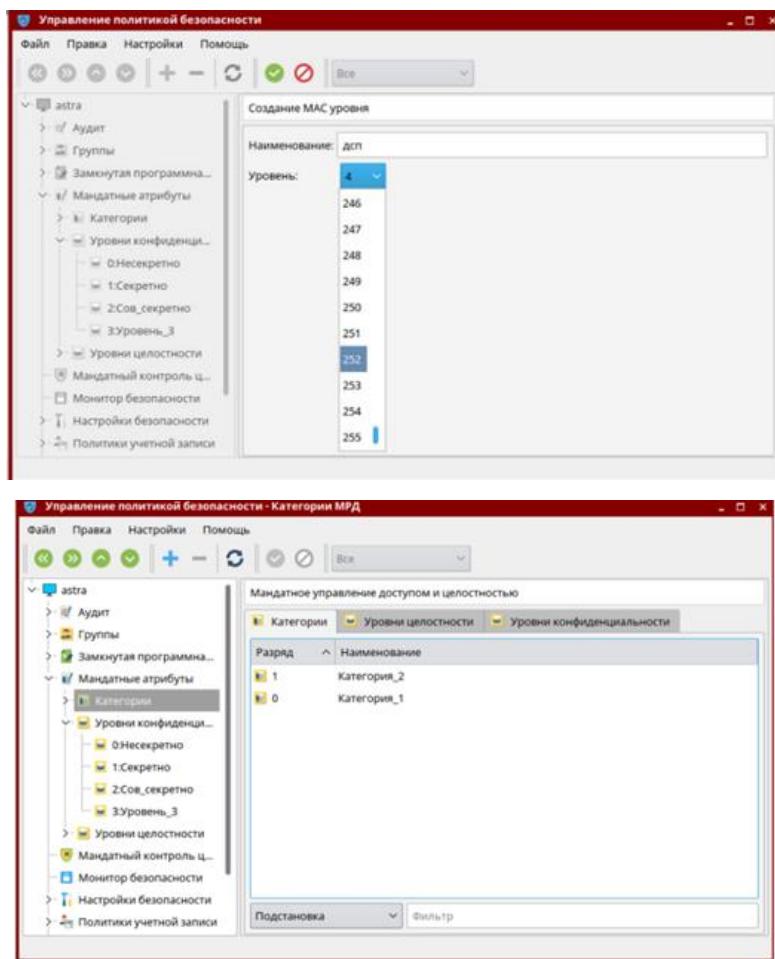
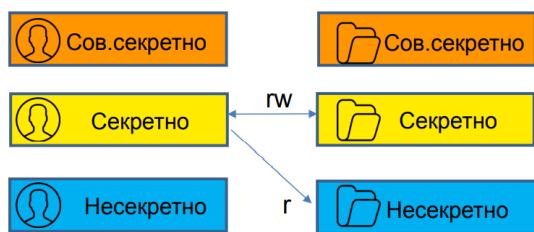


Рисунок 7 - Мандатные уровни доступа и категории

Индикатор мандатного уровня (цвет):

- «Уровень 0» — голубой;
- «Уровень 1» — желтый;
- «Уровень 2» — оранжевый;
- «Уровень 3» — темно-розовый;
- «Уровень 4» — красный;
- «Уровень 5» — коричневый;
- «Уровень 6» — пурпурный;
- «Уровень 7» — темно-фиолетовый.



Параметрами мандатного управления доступом (мандатной меткой) в AstraLinux SE являются следующие элементы:

- 1) уровень доступа или конфиденциальности (соответствует уровню конфиденциальности сущности или доступа субъект-сессии);
- 2) набор неиерархических категорий сущности и субъект-сессии;
- 3) уровень целостности сущности и субъект-сессии;
- 4) специальные атрибуты сущности (CCNR, CCNRI, E\_Hole, W\_Hole).

При установке ОССН (по умолчанию) задаются следующие параметры мандатного управления доступом и мандатного контроля целостности:

**1.2** непосредственно используемых уровня целостности («Низкий» значение 0,

«Высокий» – 63);

**2.4** уровня доступа/конфиденциальности («Уровень\_0» значение 0, «Уровень\_1» – 1, «Уровень\_2» – 2, «Уровень\_3» – 3);

**3** неиерархические категории – «Категория\_1», «Категория\_2».

Мандатное управление доступом процессов (субъект-сессий) к ресурсам (сущностям) основано на реализации соответствующего механизма в ядре ОССН. При этом принятие решения о запрете или разрешении доступа субъект-сессии к сущности осуществляется в соответствии с правилами, описанными в рамках МРОСЛ ДП-модели, и зависит от запрашиваемого вида доступа (чтение, запись, применение права доступа на выполнение) и мандатного контекста (используемых в запросе уровней конфиденциальности, доступа и целостности).

Для администрирования параметров мандатных управления доступом и контроля целостности применяются следующие команды и графические утилиты:

`pdpl-user` — команда просмотра и изменения допустимых мандатных уровней и

неиерархические категории учётных записей пользователей;

pdpl-file — команда установки параметров мандатного управления доступом на

сущность файловой системы;

pdp-id — команда вывода параметров мандатных управления доступом и контроля целостности для текущей сессии;

userlev — команда просмотра и редактирования уровней доступа, заданных в ОССН;

usercat — команда просмотра и редактирования неиерархических категорий учётных записей пользователей в ОССН;

usercaps — команда просмотра и редактирования привилегий учётных записей

пользователей;

pdp-init-fs — скрипт инициализации мандатных атрибутов ФС;

pdp-ls — вывод информации о файлах с отображением мандатных атрибутов;

pdpl-ps — управление мандатными атрибутами процессов;

sumac — запуск процесса с заданными мандатными уровнем и категорией;

fly-fm - менеджер файлов;

fly-admin-smc — графическая утилита, позволяющая решать весь комплекс задач по администрированию учётных записей пользователей и групп, в том числе, администрировать параметры мандатных управления доступом и контроля целостности.

chmac — управление мандатными атрибутами файлов;

lsm — информации о файлах с отображением мандатных атрибутов;

macid — отображение мандатных атрибутов сессии пользователя ОС;

psmac — управление мандатными атрибутами процессов;

usermac — управление допустимыми мандатными уровнями и категориями пользователей ОС;

getfmac — получение мандатных меток файловых объектов;

`setfmac` — изменение мандатных меток файловых объектов.

Мандатная метка уровня (по умолчанию 4: от 0 до 3)

Целостность (по умолчанию 0 и 63)

Категории (по умолчанию 2)

Специальные атрибуты (ccnr, ccnri, CCNRA=ccnr+ccnri, whole, ehole)

```
# pdp-ls -M /var/log/auth.log
```

```
-rw-r--r---- 1 root adm Уровень_0:Низкий:Нет:0x0 /var/log/auth.log
```

```
# pdpl-user student
```

минимальная метка: Уровень\_0:Низкий:Нет:0x0

0:0:0x0:0x0

максимальная метка: Уровень\_0:Высокий:Нет:0x0

0:63:0x0:0x0

Как правило, файлы, владельцами которых являются учётные записи пользователей, хранятся в соответствующих им домашних каталогах, находящихся в каталоге `/home`.

При этом, во время первого входа в ОССН с заданными уровнем доступа (Num1), уровнем целостности (Num2) и набором неиерархических категорий (Num3) (например, 0x2 — вторая категория) создаётся уникальный каталог с именем вида: `/home/.pdp/имя_пользователя/lNum1iNum2cNum3t0x0`, что позволяет распределить по каталогам файлы (в том числе, документы) в зависимости от их уровней конфиденциальности и целостности.

Доступ субъект-сессий (процессов), функционирующих от имени других учётных записей пользователей, к домашнему каталогу в ОССН версии 1.6 может быть ограничен с использованием как параметров (меток конфиденциальности) мандатного управления доступом, так и дискреционных прав доступа. При работе от имени учётной записи администратора в ОССН версии 1.6 желательно использовать только высокий уровень целостности (по

умолчанию он равен 63) и минимальный уровень конфиденциальности в связи с особенностями монтирования его домашнего каталога.

При администрировании ОССН необходимо руководствоваться следующими рекомендациями. Если в ней включён мандатный контроль целостности на файловой системе, то для администрирования ОССН требуется его временно отключить, для чего:

- 1) снять мандатный контроль целостности с файловой системы ОССН с помощью графической утилиты fly-admin-smc или командой unset-fs-illev;
- 2) выполнить необходимые действия по администрированию ОССН (настроить ОССН, установить пакеты, и т. д.);
- 3) включить мандатный контроль целостности на файловой системе ОССН с помощью графической утилиты fly-admin-smc или командой set-fs-illev;
- 4) настроить метки целостности установленных системных объектов.

Для полного выключения режима мандатного контроля целостности:

- 5) при использовании графического интерфейса с помощью графической утилиты fly-admin-smc выбрать «Мандатный контроль целостности» и снять отметку «подсистема МКЦ»;
- 6) при использовании терминала Fly выполнить команду astra-mic-control disable.

Независимо от способа выключения, чтобы изменения вступили в силу необходимо перезагрузить ОССН. Полностью выключать мандатный контроль целостности крайне не рекомендуется, т.к. многие механизмы защиты связаны с его включённым режимом, а именно: блокировка интерпретаторов, режим блокировки установки бита исполнения – nochmodx, блокировка доступа к конфиденциальной информации и т.д.

**Управление квотами.** Квота — это ограничение на объем дискового пространства, который может использовать пользователь или группа пользователей:

```
root@server:~# cat /etc/fstab
UUID=9a332120-6162-49f8-b884-67c78439fda7 / ext4 errors=remount-ro,usrquota,grpquota 0 1
UUID=3af25722-abb9-4a4b-a045-51b5ad0f5e14 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
```

Система квот позволяет задать лимиты: мягкий и жёсткий:

- «мягкий» лимит может быть временно превышен; пользователь просто получит предупреждение о превышении.
- «жёсткий» лимит никогда не может быть превышен: система отклонит любую операцию, приводящую к превышению жёсткой квоты.

Команды: edquota, setquota

### **Linux привилегии:**

- «cap\_chown» - игнорировать ограничения по изменению прав на владение флагом со стороны пользователя или группы;
- «cap\_dac\_override» - игнорировать права доступа к файлу;
- «cap\_dac\_read\_search» - игнорировать права на чтение или поиск файла или каталога;
- «cap\_fowner» - игнорировать проверки прав на владение файлом;
- «cap\_fsetid» - игнорировать ограничения по установке флагов setuid и setgid для файлов;
- «cap\_kill» - обходить проверки ограничений при генерировании сигналов;
- «cap\_setgid» - игнорировать ограничения на манипуляции с правами процесса, относящимися к группе пользователей;
- «cap\_setuid» - игнорировать ограничения на манипуляции с правами процесса, относящимися к пользователю;
- «cap\_setpcap» - разрешить манипуляции с привилегиями других процессов;
- «cap\_cap\_linux\_immutable» - разрешить модификацию постоянных файлов и файлов «только для записи» в файловых системах Ext2 и Ext3;

«cap\_net\_bind\_service» - разрешить привязку сокетов TCP/IP к портам ниже 1024;

«cap\_net\_broadcast» - разрешить широковещательную и многоадресную рассылку;

«cap\_net\_admin» - разрешить общее сетевое администрирование;

«cap\_net\_raw» - разрешить использование сокетов RAW и PACKET;

«cap\_ipc\_lock» - разрешить блокировку страницы и совместно используемых блоков памяти;

«cap\_ipc\_owner» - пропускать проверку владельца при межпроцессном взаимодействии;

«cap\_sys\_module» - разрешить загрузку и выгрузку модулей ядра;

«cap\_sys\_rawio» - разрешить доступ к портам ввода и вывода с помощью системных вызовов ioperm() и iopl();

«cap\_sys\_chroot» - разрешить вызов chroot;

«cap\_sys\_ptrace» - разрешить вызов ptrace() для любого процесса;

«cap\_sys\_pacct» - разрешить настройку процессорного учета;

«cap\_sys\_admin» - разрешить общее системное администрирование;

«cap\_sys\_boot» - разрешить вызов reboot();

«cap\_sys\_nice» - пропустить проверку прав доступа для системных вызовов nice(), setpriority() и позволить создавать

процессы реального времени;

«cap\_sys\_resource» - разрешить поднятие лимитов на ресурсы;

«cap\_sys\_time» - разрешить манипуляцию с часами и часами реального времени;

«cap\_sys\_tty\_config» - разрешить настройку терминала и применение системного вызова

vhangup();

«cap\_mknod» - разрешить привилегированные операции mknod();

«cap\_lease» - разрешить блокировку «аренда файла»;

«cap\_audit\_write» - разрешить запись в журнал аудита ядра;

«cap\_audit\_control» - разрешить контроль аудита ядра;

«cap\_setfcap» - разрешить установку привилегий файлов

### **Parsec привилегии:**

«parsec\_cap\_file\_cap» - право устанавливать привилегии на файлы;

«parsec\_cap\_audit» - право управления политикой аудита;

«parsec\_cap\_setmac» - разрешает изменить мандатную метку и установить другие привилегии;

«parsec\_cap\_chmac» - дает право менять мандатные метки файлов;

«parsec\_cap\_ignmaclvl» - право игнорировать мандатную политику по уровням;

«parsec\_cap\_ignmaccat» - право игнорировать мандатную политику по категориям;

«parsec\_cap\_sig» - позволяет посыпать сигналы процессам, игнорируя дискреционные и мандатные права;

«parsec\_cap\_update\_atime» - право изменять время доступа к файлу (в настоящее время не используется);

«parsec\_cap\_priv\_sock» - позволяет создавать привилегированный сокет и менять его мандатную метку.

Привилегированный сокет позволяет осуществлять сетевое взаимодействие, игнорируя мандатную политику;

«parsec\_cap\_readsearch» - позволяет игнорировать мандатную политику при чтении и поиске файлов (но не при записи);

«parsec\_cap\_cap» - право устанавливать привилегии на файлы;

«parsec\_cap\_mac\_sock» - возможность смены мандатной точки соединения.

### **Дополнительные мандатные атрибуты управления доступом:**

ccnr - может присваиваться сущностям, являющимся контейнерами;

`ccnri` может присваиваться контейнерам и определяет, что контейнер может содержать сущности с различными уровнями целостности, но не большими, чем его собственный уровень целостности;

`ehole` может присваиваться сущностям (файлам), имеющим минимальную классификационную метку, и приводить к игнорированию мандатных правил управления доступом к ним;

`whole` присваивается сущностям (файлам) с ненулевой классификационной меткой и разрешает запись в них субъектам, имеющим более низкую классификационную метку.

#### **Дополнительные мандатные атрибуты. Файловая система:**

/ - имеет наивысший мандатный уровень (по умолчанию 3), в битовой маске неиерархических категорий устанавливаются все биты, устанавливаются атрибуты CCNR и CCNRI.

`/bin, /boot, /etc, /lib, /lib32, /lib64, /lost+found, /media, /mnt, /opt, /proc, /root, /sbin, /selinux, /srv, /sys, /usr` - нулевой мандатный уровень и пустая (нулевая) маска неиерархических категорий, атрибутов CCNR и CCNRI нет.

`/media` - большинство сменных носителей монтируется в каталог `/home/%user%/media`, несанкционированный доступ одного пользователя к сменным носителям другого пользователя - невозможен.

`/dev, /run и /var` – наивысший мандатный уровень и все неиерархические мандатные категории, устанавливается атрибут CCNR. Объекты, расположенные в каталогах `/dev, /run и /var`, имеют нулевые мандатные метки.

`/parsecfs` - нулевой мандатный уровень, нулевая битовая маска неиерархических категорий и атрибут EHole.

`/tmp` - наивысший мандатный уровень и все неиерархические мандатные категории, устанавливаются атрибуты CCNR и EHole.

`/home` - наивысший мандатный уровень, ему присвоены все неиерархические мандатные категории и атрибут CCNR.

Такие же мандатные атрибуты присваиваются служебному подкаталогу `/home/.pdp`. Мандатные атрибуты домашних каталогов пользователей соответствуют мандатным атрибутам учётных записей пользователей.

## Методика выполнения

### 1 Основы мандатного управления доступом

1. В ОССН версии 1.6 создать учётную запись пользователя `user`, с параметрами: максимальный и минимальный уровни доступа — 0, неиерархические категории — нет, уровень целостности — «Высокий», входит в группу администраторов — `astra-admin` (вторичная группа), разрешено выполнение привилегированных команд (`sudo`). Войти в ОССН с учётной записью пользователя `user` (Уровень\_0, «Высокий»).

2. Запустить графическую утилиту редактирования учётных записей пользователей «Политика безопасности» через меню «Панель управления» главного пользовательского меню. Открыть раздел настройки локальных пользователей, и для созданных учётных записей пользователей `user1`, `user2`, `user3`, `user4` произвольно задать их параметры:

- максимальный и минимальный уровни доступа;
- минимальные и максимальные наборы неиерархических категорий;
- максимальный уровень целостности.

3. Настроить параметры учётной записи пользователя `user2`:

- установить минимальное количество дней между сменой пароля — 180 дней, число дней до выдачи предупреждения о смене пароля — 5 дней;
- выбрать максимальный уровень — «Уровень\_3»;
- проверить возможность задать минимальный или максимальный набор неиерархических категорий.

Войти в ОССН с учётной записью пользователя `user2`, выбрав уровень доступа «Уровень\_1». Проверить возможность выбора набора

неиерархических категорий и уровня целостности. Создать в каталоге «Документы» файл 1.txt. Выйти из ОССН.

Войти в ОССН с учётной записью пользователя user2, выбрав уровень доступа «Уровень\_2». Создать в каталоге «Документы» файл 2.txt.

Проверить возможность чтения объектов файловой системы ОССН, владельцем которых является учётная запись пользователя user2 (на текущем уровне доступа «Уровень\_2»):

- открыть каталог «Документы» уровня доступа «Уровень\_1» (Компьютер/Домашний/mac/11i0c0x0t0x0/Документы);
- открыть файл 1.txt, проверив возможность его чтения или записи;
- выйти из ОССН.

4. Проверить наличие и возможность чтения объектов файловой системы ОССН, владельцем которых является учётная запись пользователя user2 на текущем уровне доступа («Уровень\_1»):

- войти в ОССН с учётной записью пользователя user2, выбрав уровень доступа
  - «Уровень\_1»;
  - проверить возможность открытия каталога «Документы» для уровня доступа «Уровень\_2» (Компьютер/Домашний/mac/12i0c0x0t0x0/Документы);
  - выйти из ОССН.

5. Войти в ОССН с учётной записью пользователя user. Запустить графическую утилиту «Политика безопасности». Сравнить списки вторичных групп для учётных записей пользователей user, user1, user2, user3, user4, при этом определив:

- учётные записи пользователей, являющиеся администраторами (ходящими в группу astra-admin);
- учётные записи пользователей, входящие в группу users.

6. Создать новую учётную запись пользователя user10, при этом:

- установить минимальное количество дней между сменой пароля — 180 дней и число дней выдачи предупреждения до смены пароля — 5 дней;

- выбрать максимальный уровень доступа — «Уровень\_3», минимальный уровень доступа — «Уровень\_0», уровень целостности — «Высокий»;
- добавить в список вторичных групп группы astra-admin и lpadmin;
  - проверить возможность входа в ОССН с учётной записью пользователя user10 с уровнями доступа «Уровень\_2» или «Уровень\_3» (уровень целостности «Низкий»).

Войти в ОССН с учётной записью пользователя user10 (уровень доступа – «Уровень\_0», уровень целостности «Высокий»). Проверить возможность создания новой учётной записи пользователя user11 с использованием графической утилиты fly-admin-smc без использования и с использованием команды sudo. Выйти из ОССН.

7. Войти в ОССН с учётной записью пользователя user1 с уровнем доступа – «Уровень\_0». Осуществить попытки запуска графической утилиты «Политика безопасности» через главное пользовательское меню и запуска её с использованием терминала Fly командой fly-admin-smc. Проанализировать результаты и предупреждения ОССН.

8. Осуществить переключение между сессиями различных учётных записей пользователей без выхода из ОССН:

- через меню «Завершение работы» главного пользовательского меню перейти в подменю «Сессия» и далее «Отдельная» и войти в ОССН с учётной записью пользователя user (уровень целостности «Высокий»);
- аналогично вернуться и далее закрыть сеанс от имени учётной записи пользователя user1.

9. С использованием графической утилиты «Политика безопасности» заблокировать пароль учётной записи пользователя user1. Проверить изменения файлов /etc/passwd и /etc/shadow, осуществив следующие действия:

- в терминале Fly выполнить команды sudo cat /etc/passwd и sudo cat /etc/shadow;

- проверить наличие блокировки учётной записи пользователя по файлу /etc/shadow (должен быть установлен знак «!» в начале свёртки пароля);
- проверить функционирование блокировки путём осуществления попытки входа в ОССН в отдельном сеансе от имени учётной записи пользователя user1;
- снять блокировку (выполнить удаление пароля и блокировки входа, задать повторно пароль) и проверить возможность входа в ОССН с учётной записью пользователя user1.

10. Выполнить удаление учётных записей пользователей:

- удалить учётную запись пользователя user10 с использованием графической утилиты «Политика безопасности»;
- удалить учётную запись пользователя user2 командой sudo deluser user2;
- удалить учётную запись пользователя user1 командой sudo userdel user1;
- проверить наличие домашних каталогов учётных записей пользователей user1 и user2, после чего с использованием справочной информации по команде userdel определить её параметры, позволяющие удалять содержимое домашнего каталога учётной записи пользователя;
- удалить домашние каталоги учётных записей пользователей user1 и user2 непосредственно командами rm -r /home/userone и rm -r /home/usertwo, осуществив попытки удаления без использования и с использованием команды sudo;
- проверить наличие домашних каталогов учётных записей пользователей user1, user2 и user10 в каталоге /home/.pdf.

11. Создать новые группы group3 (с использованием графической утилиты «Политика безопасности») и группу group4 (командой sudo addgroup group4, выполненной в терминале Fly).

12. Добавить учётную запись пользователя user3 во вторичную группу group3 командой usermod -a -G group3 user3 и во вторичную группу group4 с

помощью графической утилиты «Политика безопасности». Проверить включение учётной записи пользователя user3 в группы group3 и group4 путем просмотра содержимого файла /etc/group командами cat /etc/group | grep "^group3" и cat /etc/group | grep "^group4"

13. Выполнить удаление учётной записи пользователя user3 из группы group3 с использованием графической утилиты «Политика безопасности» и из группы group4 командой gpasswd -d user3 group4.

14. Удалить группу group3 командой sudo delgroup group3 в терминале Fly и группу group4 с помощью графической утилиты «Политика безопасности».

15. Изучить порядок хранения параметров мандатного управления доступом и мандатного контроля целостности для учётных записей пользователей. Для этого выполнить следующие действия:

- определить уровни доступа, заданные в ОССН, для этого вывести в терминал Fly содержимое файла /etc(parsec/mac\_levels;

- определить неиерархические категории, заданные в ОССН, для этого вывести в

терминал Fly содержимое файла /etc(parsec/mac\_categories;

- определить идентификатор учётной записи пользователя user1 по файлу /etc/passwd командой cat /etc/passwd | grep "^user1:" | cut -d : -f 3;

- считать параметры мандатного управления доступом и мандатного контроля целостности для учётной записи пользователя user1 командой cat /etc(parsec/macdb/\$(cat /etc/passwd | grep "^user1:" | cut -d : -f 3) и проверить их соответствие данным, отображаемым в графической утилите «Политика безопасности».

## **2 Настройка параметров мандатного управления доступом и мандатного контроля целостности**

1. Войти в ОССН в графическом режиме с учётной записью пользователя user (уровень доступа — 0, неиерархические категории — нет, уровень целостности — «Высокий»).

Запустить графическую утилиту редактирования учётных записей пользователей «Политика безопасности» через меню «Панель управления» главного пользовательского меню.

Модифицировать параметры мандатного управления доступом, для этого осуществить следующие действия:

-открыть раздел «Мандатные атрибуты», «Уровни конфиденциальности» и выбрать «0»:Уровень\_0» и переименовать данный уровень доступа: «Уровень0»;

- выполнить создание уровня доступа с именем «Уровень\_4», задав значение равное 4, после чего проверить наличие записи «Уровень\_4» в списке «Уровни конфиденциальности»;

-выполнить обратное переименование: «Уровень0» в «Уровень\_0».

2. Создать учётную запись пользователя user1, установив максимальный уровень

доступа: «Уровень\_4».

3. Выполнить удаление уровня доступа 4 из раздела «Уровни конфиденциальности»

путём выбора в контекстном меню пункта «Удалить».

4. Открыть учётную запись пользователя user1 и вкладке «МРД» в элементе «Максимальный уровень» проверить отсутствие записи имени уровня, при этом, в списке выбора уровня «Уровень\_4» также будет отсутствовать.

5. Вывести в терминал Fly параметры мандатного управления доступом для учётной записи пользователя user1. Для этого выполнить следующие действия:

-запустить терминал Fly и перейти в каталог /etc/parssec/macdb;

-прочитать параметры учётной записи user1 командой sudo grep “^user1:” \*;

-определить максимальный уровень доступа учётной записи user1 командой sudo

```
grep “user1:” * | cut -d : -f 5;
```

-определить минимальный уровень доступа учётной записи user1 командой sudo

grep “user1” \* | cut -d : -f 3 и проверить его соответствие данным, отображаемым в графической утилите «Политика безопасности».

6. Создать неиерархические категории с использованием графической утилиты «Политика безопасности». Для этого выполнить следующие действия:

- в разделе «Категории» удалить исходные неиерархические категории;
- создать новую неиерархическую категорию с именем «Otdel1», «Разряд» –0;
- в разделе «Категории» создать новые неиерархические категории: «Otdel2» («Разряд» – 1), «Upravlenie» («Разряд» – 2).

7. Изменить набор неиерархических категорий с использованием графической утилиты «Политика безопасности», для этого выполнить следующие действия в разделе «Категории»:

- выбрать неиерархическую категорию «Otdel1» и ввести наименование «Отдел\_1»;
- аналогично переименовать неиерархические категории «Otdel2» и «Upravlenie» в «Отдел\_2» и «Управление», соответственно;
- проанализировать возможность одновременного изменения элемента «Разряд».

8. Изменить мандатный уровень доступа с использованием графической утилиты «Политика безопасности», для этого выполнить следующие действия:

-создать новую группу с именем «office1» и задать первичную группу учётной записи пользователя user1 — «office1»;

-создать новую учётную запись пользователя user2 и установить её первичную группу — «office1»;

-во вкладке «Дополнительные» осуществить попытку выбора минимального набора неиерархических категорий — «Отдел\_2», и проанализировать результат;

- во вкладке «Дополнительные» выбрать максимальный уровень доступа — «Уровень\_3», максимальный набор неиерархических категорий — «Отдел\_2», после чего задать минимальный набор неиерархических категорий — «Отдел\_2»;

-открыть параметры учётной записи пользователя user1 и выбрать максимальный

уровень доступа — «Уровень\_3», максимальный набор неиерархических категорий — «Отдел\_1», минимальный набор неиерархических категорий — «Отдел\_1»;

- создать учётную запись пользователя rukoffice1 и задать первичную группу: «office1»;

-во вкладке «Дополнительные» выбрать максимальный уровень: «Уровень\_3», максимальный набор категорий: «Отдел\_1», «Отдел\_2», «Управление».

9. Создать общий каталог для работы от имени учётных записей пользователей user1, user2, rukoffice1 в каталоге /home/work. При этом, для работы от имени учётных записей пользователей с наборами неиерархическими категорий равными «Отдел\_1», «Отдел\_2» и «Управление» выделить отдельные каталоги «otdel1», «otdel2» и «upr», соответственно. При этом обеспечить хранение файлов с различными уровнями конфиденциальности в каталогах с использованием специального атрибута CCNR, для чего осуществить следующие действия:

-запустить терминал Fly в «привилегированном» режиме командой sudo fly-term;

-создать каталог work и задать параметры мандатного и дискреционного управления доступом командами:

```
mkdir /home/work
```

```
chown user:office1 /home/work
```

```
chmod 750 /home/work
```

```
pdpl-file 3:0:Отдел_1,Отдел_2,Управление:ccnr /home/work
```

-создать каталог для работы от имени учётных записей пользователей с набором неиерархических категорий равным «Отдел\_1» и задать параметры мандатного и дискреционного управления доступом командами:

```
user1:office1 otdel1
```

```
chmod 770 otdel1
```

```
pdpl-file 3:0:Отдел_1:ccnr otdel1
```

-создать каталог для работы от имени учётных записей пользователей с набором неиерархических категорий равным «Отдел\_2» и задать параметры мандатного и

дискреционного управления доступом командами:

```
user2:office1 otdel2
```

Дать права: владелец – чтение, запись, выполнение; группа – чтение, запись, выполнение; остальные - ничего не делать.

```
pdpl-file 3:0:Отдел_2:ccnr otdel2
```

-создать каталог upr для работы от имени учётных записей пользователей с набором неиерархических категорий равным «Управление» командами:

```
rukoffice1:office1 upr
```

Дать права: владелец – чтение, запись, выполнение; группа – чтение, запись, выполнение; остальные - ничего не делать.pdpl-file 3:0:Управление:ccnr upr

-создать вложенные каталоги У1, У2, У3 в каталогах otdel1, otdel2, upr  
-установить для каталогов otdel1, otdel2, upr необходимые уровни (см. команды для каталога upr):

```
pdpl-file 1:0:Управление:0 /home/work/upr/У1
pdpl-file 2:0:Управление:0 /home/work/upr/У2
pdpl-file 3:0:Управление:0 /home/work/upr/У3
chown rukoffice1:office1 upr/У{1,2,3}
chmod 770 upr/У{1,2,3}
```

10. Установить дискреционные права доступа с разрешением на запись и чтение для группы office1 в графическом файловом менеджере Fly (fly-fm). Выполнить последовательные входы в ОССН с учётной записью пользователя user1 (неиерархическая категория — «Отдел\_1», уровни доступа 1, 2, 3). При работе на уровнях доступа 1, 2 и 3 создать в каталоге /home/work/otdel1/уровеньX файлы с именами 11.txt, 12.txt, 13.txt, соответственно.

11. Выполнить последовательные входы в ОССН с учётной записью пользователя user2 (неиерархическая категория — «Отдел\_2», уровни доступа 1, 2, 3). При работе на мандатных уровнях доступа 1, 2 и 3 создать в каталоге /home/work/otdel2/уровеньX файлы с именами 21.txt, 22.txt, 23.txt, соответственно, и установить дискреционные права доступа с разрешением на запись и чтение для группы office1 в файловом менеджере Fly.

12. Войти в ОССН с учётной записью пользователя rukoffice1 (уровень доступа — 3, неиерархическая категория — «Отдел2») и проверить возможность получения следующих доступов к файлам: доступ на чтение к файлам 21.txt, 22.txt, 23.txt, доступ на запись к файлу 23.txt.

13. Войти в ОССН с учётной записью пользователя rukoffice1 (уровень доступа — 2, неиерархическая категория — «Отдел\_1») и проверить возможность получения следующих доступов к файлам: доступ на чтение к файлам 11.txt, 12.txt, доступ на запись к файлу 12.txt.

14. Войти в ОССН с учётной записью пользователя rukoffice1 (уровень доступа — 3, набор неиерархических категорий — «Отдел\_1», «Отдел\_2», «Управление») и проверить возможность получения доступа на чтение к файлам 11.txt, 12.txt, 13.txt, 21.txt, 22.txt, 23.txt.

15. Войти в ОССН с учётной записью пользователя rukoffice1 (уровень доступа — 3, неиерархическая категория — «Управление»). Создать файл u3.txt в каталоге /home/work/upr/U3.

16. Войти в ОССН с учётной записью пользователя rukoffice1 (уровень доступа — 3, набор неиерархических категорий: «Отдел\_1», «Отдел\_2», «Управление») и проверить возможность получения следующих доступов к файлам: доступ на запись к файлу u3.txt, доступ на чтение к файлам u3.txt, 11.txt, 12.txt, 13.txt, 21.txt, 22.txt, 23.txt.

17. Для доступа к терминалу Fly настроить включение учётных записей пользователей user1, user2, rukoffice1 во вторичную группу astra-console. Это позволит данным учётным записям пользователей запускать терминал Fly с использованием комбинации Win+R.

18. Вывести в терминал Fly параметры мандатного управления доступом и мандатного контроля целостности для учётных записей пользователей. Для этого выполнить следующие действия:

-войти в ОССН с учётной записью пользователя rukoffice1 (уровень доступа – 2, набор неиерархических категорий: «Отдел\_1», «Управление»);

-в терминале Fly выполнить команду pdp-id -a, проанализировать результат;

-выполнить избирательный вывод параметров мандатного управления доступом (с числовыми значениями) командами pdp-id -l и pdp-id -c;

-выполнить избирательный вывод параметров мандатного управления доступом (с именами) командами pdp-id -ln и pdp-id -cn.

19. Изменить параметры мандатного управления доступом и мандатного контроля целостности учётной записи пользователя rukoffice1. Для этого выполнить следующие действия:

-войти в ОССН с учётной записью пользователя user (уровень доступа — 0, неиерархические категории — нет, уровень целостности — «Высокий») и запустить терминал Fly в «привилегированном» режиме;

- изменить минимальный и максимальный уровни доступа учётной записи пользователя rukoffice1 командой pdpl-user -l 0:2 rukoffice1, а также минимальный и максимальный наборы неиерархических категорий пользователя rukoffice1 командой pdpl-user -c 0:2 rukoffice1;

- обнулить значения уровней доступа и наборов неиерархических категорий в параметрах учётной записи пользователя rukoffice1 командой pdpl-user -z rukoffice1;

- установить значения уровней доступа и наборов неиерархических категорий в параметрах учётной записи пользователя rukoffice1 1:3 -c 0:7 rukoffice1.

20. Считать параметры мандатного управления доступом и мандатного контроля целостности учётной записи пользователя rukoffice1 из файлов настроек. Для этого выполнить следующие действия:

-перейти в каталог /etc/parsec/macdb и считать минимальный и максимальный уровни доступа командами grep “rukoffice1” \* | cut -d : -f 3 и grep “rukoffice1” \* | cut -d : -f 5, соответственно;

-считать минимальный и максимальный наборы неиерархических категорий 4 и 6, соответственно.

21. Создать и модифицировать мандатные уровни доступа, осуществив следующие действия:

-вывести в терминал созданные уровни доступа командой userlev и сравнить полученные данные с настройками в утилите «Политика безопасности»;

-добавить новый уровень доступа с именем «Уровень\_4» (значение 4) командой userlev Уровень\_4 --add 4 и вывести в терминал уровни доступа командой userlev;

-выполнить переименование уровня доступа «Уровень\_4» в «НовыйУровень» командой userlev;

-добавить возможность работы от имени учётной записи пользователя rukoffice1 на уровне доступа 4;

-выполнить попытку изменения значения уровня доступа «НовыйУровень» на 3 командой userlev, проанализировать результат;

-изменить значение уровня доступа «НовыйУровень» на 5 командой userlev и вывести в терминал максимальный уровень доступа учётной записи пользователя rukoffice1, проанализировать результат;

-установить максимальный уровень доступа учётной записи пользователя rukoffice1 равным 5;

-удалить уровень доступа с именем «НовыйУровень» командой в терминале и определить максимальный уровень доступа учётной записи пользователя rukoffice1 командой в терминале, проанализировать результат;

-восстановить набор неиерархических категорий и уровней доступа учётной записи пользователя rukoffice1 :3 -c 0:7.

## 22. Создать и модифицировать неиерархические категории:

-в терминале Fly, запущенном в «привилегированном» режиме, вывести неиерархические категории командой usercat;

-добавить новую неиерархическую категорию командой usercat;

-переименовать неиерархическую категорию «otdel3» в «Отдел\_3» командой usercat;

-осуществить попытку модификации наборов неиерархических категорий учётной записи пользователя rukoffice1 командой pdpl-user, проанализировать результат;

-добавить неиерархическую категорию «Отдел\_3» в наборы неиерархических категорий учётной записи пользователя rukoffice1 командой pdpl-user, обратить внимание на то, что неиерархическая категория задаётся в шестнадцатеричном формате;

-осуществить попытку изменения значения неиерархической категории «Отдел\_3» на значение 2 командой usercat, проанализировать результат;

-изменить значение неиерархической категории «Отдел\_3» на 0x10 командой usercat;

-изменить значение неиерархической категории «Отдел\_3» на 0x20 командой usercat, обратить внимание на то, что независимо от указания типа числа по префиксу «0x» (десятичное или шестнадцатеричное) значение неиерархической категории задаётся в шестнадцатеричном формате;

-удалить неиерархическую категорию «Отдел\_3» командой usercat Отдел\_3 --delete;

-изменить значение неиерархической категории «Управление» на 0x10 командой usercat, проанализировать результат по данным, выводимым командой pdpl-user rukoffice1;

-изменить значение неиерархической категории «Управление» на 4 командой usercat.

23. Для настройки привилегий учётных записей пользователей осуществить следующие действия:

-вывести в терминал заданные в ОССН привилегии учётных записей пользователей командой usercaps, при работе в терминале Fly в «привилегированном» режиме;

-запустить графическую утилиту «Политика безопасности» и открыть настройки учётной записи пользователя user1, вкладке «Привилегии» установить Linux-привилегии cap\_kill, cap\_fowner и PARSEC-привилегии parsec\_cap\_chmac, parsec\_cap\_sig, после чего закончить работу с утилитой;

-вывести привилегии учётной записи пользователя user1 командой usercaps user1;

-в графической утилите «Политика безопасности» открыть параметры учётной записи пользователя user, вкладке «Привилегии» выбрать Linux-привилегии cap\_kill,

cap\_fowner и PARSEC-привилегии parsec\_cap\_chmac, parsec\_cap\_sig;

-запустить терминал Fly в «непrivилегированном» режиме командой fly-term и осуществить попытку запуска команды usercaps;

-определить расположение файла usercaps командой which usercaps, выполненной из «привилегированного» режима, а затем выполнить в «непривилегированном» режиме команду /usr/sbin/usercaps, проанализировать результат;

-запустить терминал Fly в «привилегированном» режиме командой sudo fly-term и выполнить модификацию Linux-привилегий и PARSEC-привилегий командами:

```
usercaps -l 9 user1
```

```
usercaps -m 2 user1
```

```
usercaps -m 11 user1
```

-с использованием графической утилиты «Политика безопасности» определить установленные привилегии и формат параметра модификации привилегий учётных записей пользователей (десятичная, восьмеричная или шестнадцатеричная система счисления при этом используется?);

-установить для учётной записи пользователя user1 полный список привилегий командой usercaps -f user1, затем удалить все привилегии учётной записи пользователя user1 командой usercaps -z user1;

-вывести списки Linux-привилегий и PARSEC-привилегий командами usercaps -L и usercaps -M, соответственно.

## **Контрольные вопросы**

1. Какие имеются особенности создания учётных записей пользователей с использованием команд adduser, useradd и графической утилиты «Политика безопасности» (fly-admin-smc), в том числе:

- какой группе должна принадлежать учётная запись пользователя, чтобы была возможность выполнения команды adduser?
- какими командами создаётся учётная запись пользователя, и какие дополнительные параметры при этом вводятся?

- какие ограничения накладываются на пароль учётной записи пользователя при его создании?

- в какие группы автоматически добавляется учётная запись пользователя?

2. Как выполнять привилегированные команды?

3. Создаются ли домашние каталоги учётных записей пользователей при добавлении их с использованием графической утилиты «Политика безопасности»?

4. Создаются ли домашние каталоги учётных записей пользователей при их добавлении с использованием команд adduser и useradd?

5. Какие минимальный и максимальный уровни доступа задаются по умолчанию для учётных записей пользователей, создаваемых командами adduser и useradd?

6. Какими способами можно добавить или удалить учётную запись пользователя из группы?

7. Каким образом организовано хранение сущностей файловой системы ОССН, созданных процессами, обладающими различными уровнями доступа?

8. Где и в каком формате хранятся параметры мандатного управления доступом и мандатного контроля целостности, заданные в ОССН?

9. Где и в каком формате хранятся параметры мандатного управления доступом и мандатного контроля целостности для учётных записей пользователей?

10. Какой командой задаётся максимальный набор неиерархических категорий для текущей учётной записи пользователя?

11. Каким образом осуществляется переход от текущего сеанса к сеансу, функционирующему от имени другой учётной записи пользователя?

12. Позволяют ли команды useradd и adduser задавать параметры мандатного управления доступом и мандатного контроля целостности для создаваемых учётных записей пользователей?

## **Перечень использованных информационных ресурсов**

1. РУСБ.10015-01 93 01. Операционная система специального назначения «Astra Linux Special Edition». Руководство пользователя
2. РУСБ.10015-01 31 01. Операционная система специального назначения «Astra Linux Special Edition». Описание применения
3. РУСБ.10015-01 95 01-1. Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора.  
**Часть1**
4. РУСБ.10015-01 95 01-2. Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора.  
**Часть2**
5. Справочный центр Astra Linux <https://wiki.astralinux.ru/>
6. Вовк Е. ASTRA LINUX. Руководство по национальной операционной системе и совместимым офисным программам / Е. Вовк – Москва: ООО «Манн, Иванов и Фербер», 2022. – 398с. - ISBN 9785001951360
7. Матвеев М. Д. Astra linux. Установка, настройка, администрирование / М.Д. Матвеев - СПб.: Изд. Наука и техника, 2023. - 416 с. - ISBN 978-5-907592-07-0
8. Девягин П.Н. Моделирование и верификация политик безопасности управления доступом в операционных системах / П. Н. Девягин, Д. В. Ефремов, В. В. Кулямин и др. – М.: Горячая линия – Телеком, 2019. – 214 с.- ISBN 978-5-9912-0787-4.
9. Шоттс У. Командная строка Linux. Полное руководство. Серия «Для профессионалов». 2-е межд. изд. — СПб.: Питер, 2020. — 544 с. - ISBN 978-5-4461-1430-6
11. Андреев Е. Системное администрирование Astra Linux – СПб.: БХВ. – 2024. – 400с. - ISBN 978-5-9775-1993-9

**Приложение А**  
**Образец оформления лабораторной работы**

**Лабораторная работа №1**  
**Установка и настройка операционной системы**  
**AstraLinuxSE**

Выполнил: Иванов И.И., гр. ВКБ-31

Проверил: \_\_\_\_\_

(*ф.и.о, подпись, дата*)

**Цель работы** – изучение требований к целевому компьютеру и подготовка к установке, установка ОС, настройка дополнительных параметров в Astra Linux SE.

**Ход работы**

1. ...
2. ... и т.д.

**Вывод:**...

**Ответы на контрольные вопросы**

1. ...
2. ... и т.д.

**Приложение Б**  
**Образец оформления титульного листа Журнала**  
**лабораторных работ**



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»  
(ДГТУ)**

Факультет Информатика и вычислительная техника

Кафедра «Кибербезопасность информационных систем»

**ЖУРНАЛ ЛАБОРАТОРНЫХ РАБОТ**

по дисциплине «Операционные системы»

Обучающийся \_\_\_\_\_  
подпись, дата

Группа \_\_\_\_\_

Специальность \_\_\_\_\_

Проверил: \_\_\_\_\_ кт.н., доцент Н.Н. Язвинская  
отметка, подпись, дата

Ростов-на-Дону

202\_\_\_\_