Отчёта по лабораторной работе № 2

Информационная безопасность

Адоле Фейт Эне

Содержание

Список иллюстраций

Список таблиц

0.1 Цель работы

Получение практических навыков работы в консоли с атрибутами файлов, закрепление теоретических основ дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux.

0.2 Теоретическое введение

В операционной системе Linux есть много отличных функций безопасности, но одна из самых важных - это система прав доступа к файлам. Изначально каждый файл имел три параметра доступа. Вот они: • Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем • Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги • Выполнение - невозможно выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу Каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа: • Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение • Группа - любая группа пользователей, существующая в системе и

привязанная к файлу. Но это может быть только одна группа и обычно это группа владельца, хотя для файла можно назначить и другую группу

• Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла Команды, которые могут понадобиться при работе с правами доступа: • “ls -l” - для просмотра прав доступа к файлам и каталогам • “chmod категория действие флаг файл или каталог” - для изменения прав доступа к файлам и каталогам (категорию действие и флаг можно заменить на набор из трех цифр от 0 до 7) Значения флагов прав: • — - нет никаких прав • –x - разрешено только выполнение файла, как программы, но не изменение и не чтение • -w- - разрешена только запись и изменение файла • -wx - разрешено изменение и выполнение, но в случае с каталогом, невозможно посмотреть его содержимое • r– - права только на чтение • r-x - только чтение и выполнение, без права на запись • rw- - права на чтение и запись, но без выполнения • rwx - все права

0.3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе несколь ких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта,производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При

этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельтакомпрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

0.4 Выполнение лабораторной работы

В установленной при выполнении предыдущей лабораторной работы ОС создала учётную запись пользователя guest с помощью команды “sudo useradd guest” и задала пароль для этого пользователя командой “sudo passwd guest” (рис. 3.1).

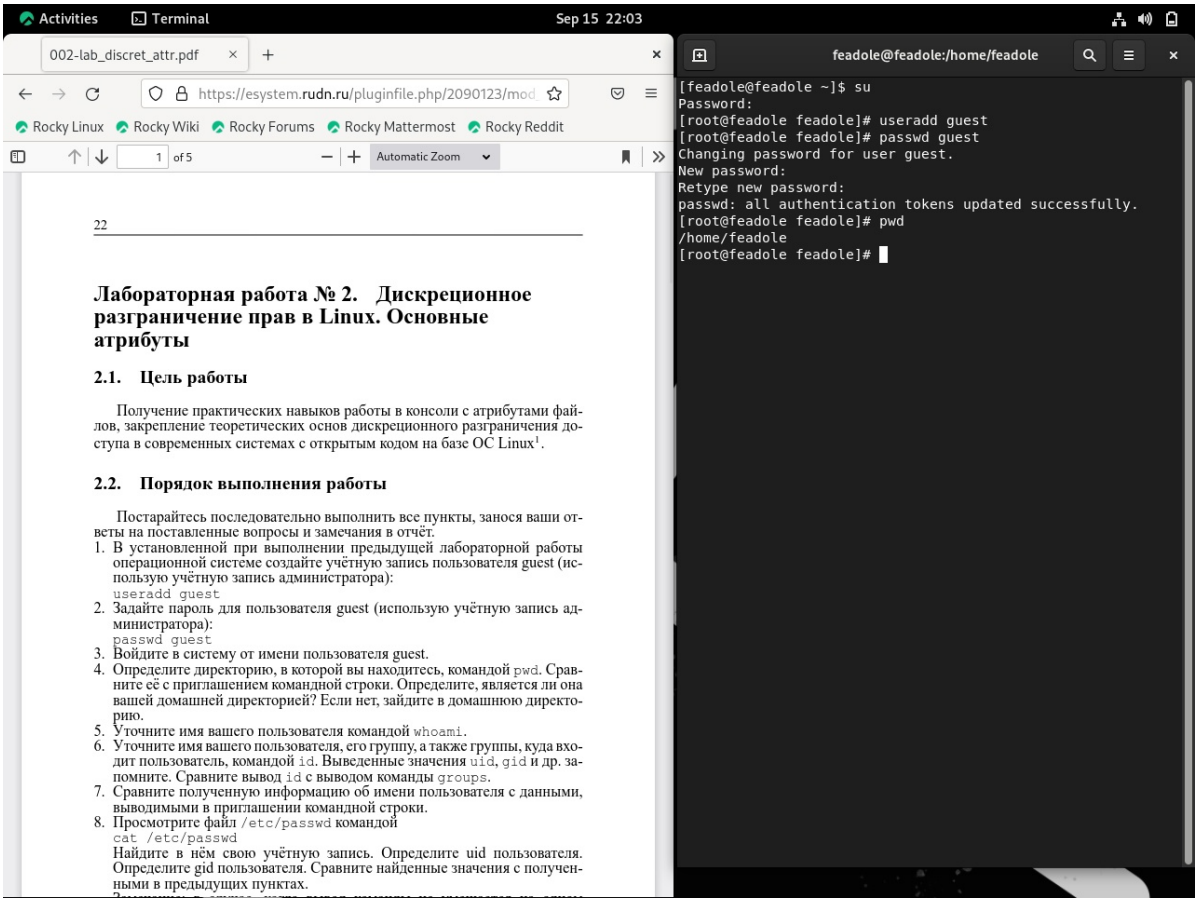

Рис. 1: Рис. 3.1: Создание пользователя

Вошла в систему от имени пользователя guest (рис. 3.2, 3.3)

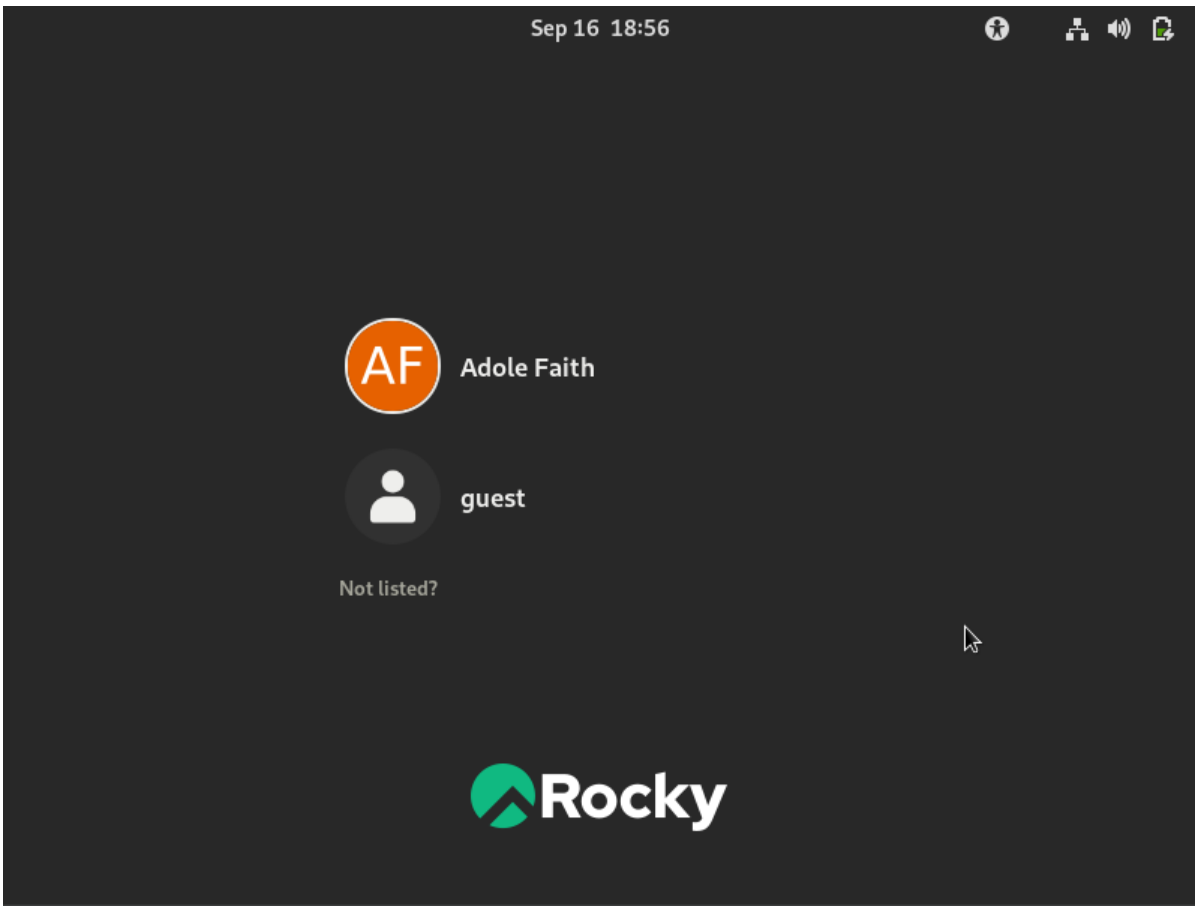

Рис. 2: Рис. 3.2: Вход в систему

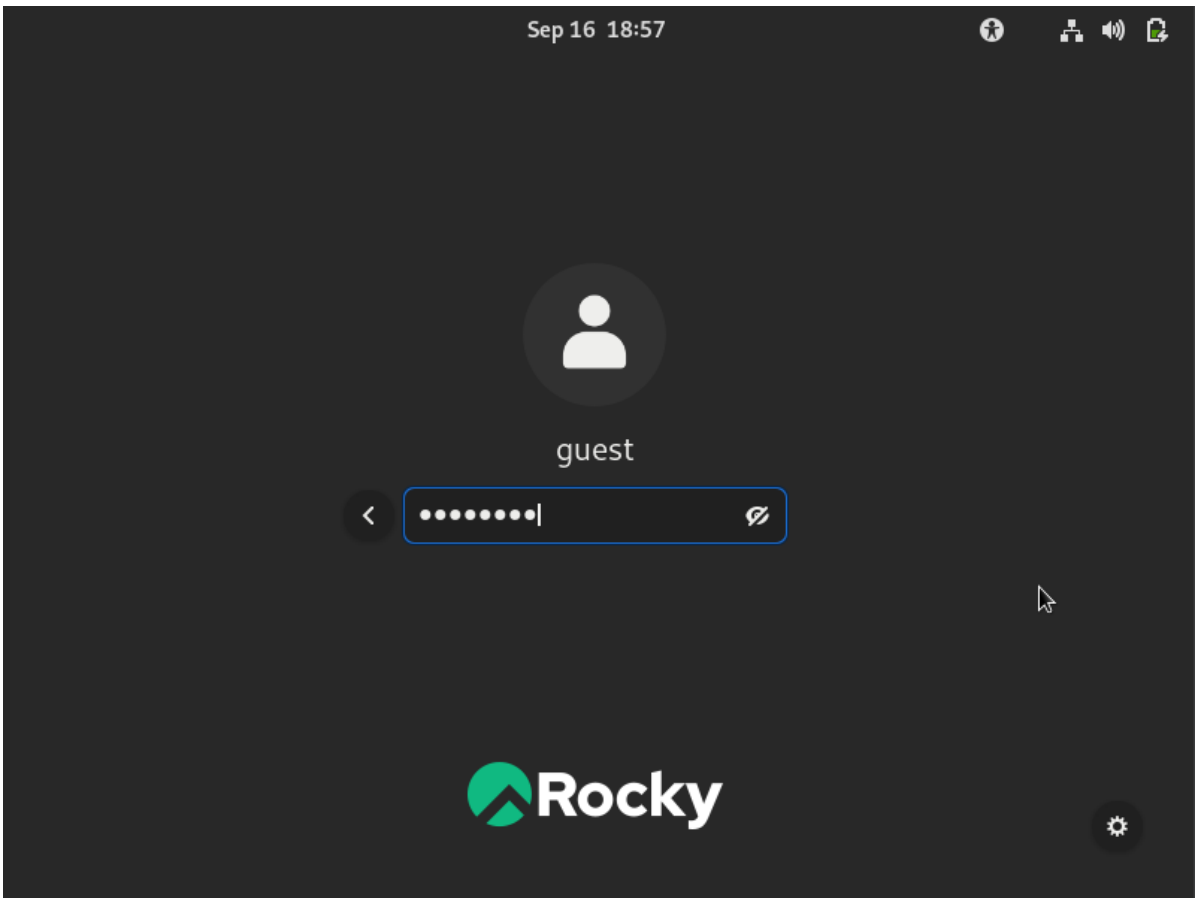

Рис. 3: Рис. 3.3: Вход в систему

Командой “pwd” определила, что нахожусь в директории /home/guest, которая и является моей домашней директорией (рис. 3.4). С приглашением командной строки совпадает. Уточнила имя моего пользователя командой “whoami” и получила вывод: guest (рис. 3.4). С помощью команды “id” определила имя своего пользователя - всё так же guest, uid = 1001 (guest), gid = 1001 (guest). Затем сравнила полученную информацию с выводом команды “groups”, которая вывела “guest”. Мой пользователь входит только в одну группу, состоящую из него самого, поэтому вывод обеих команд “id” и “groups” совпадает (рис. 3.4). Данные, выводимые в приглашении командной строки, совпадают с полученной информацией. Затем просмотрела файл /etc/passwd командой “cat /etc/passwd” (рис. 3.4).

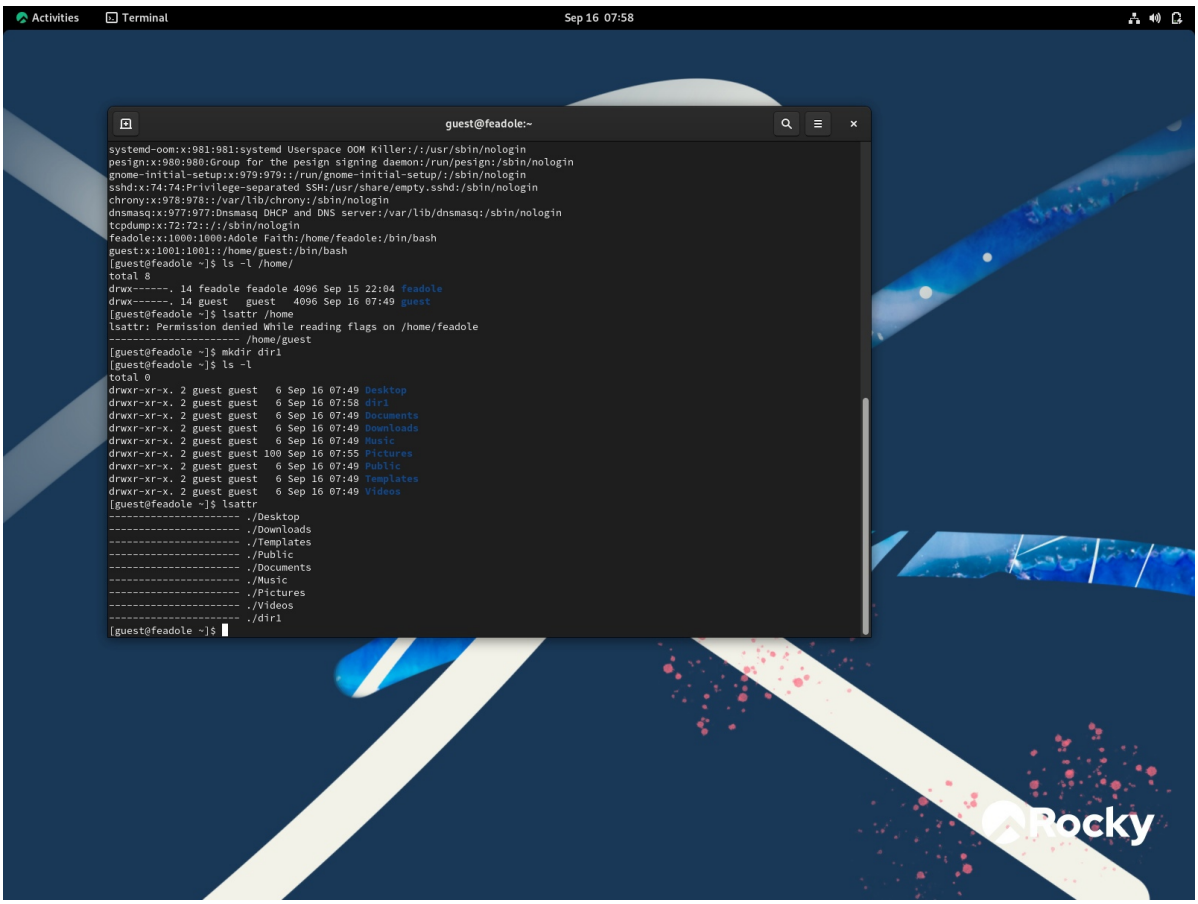

Рис. 4: Рис. 3.4: Команды pwd, whoami, id, groups, cat

Нашла в нём свою учётную запись в самом конце (рис. 3.5). Uid = 1001, gid = 1001, то есть они совпадают с тем, что мы получили ранее.

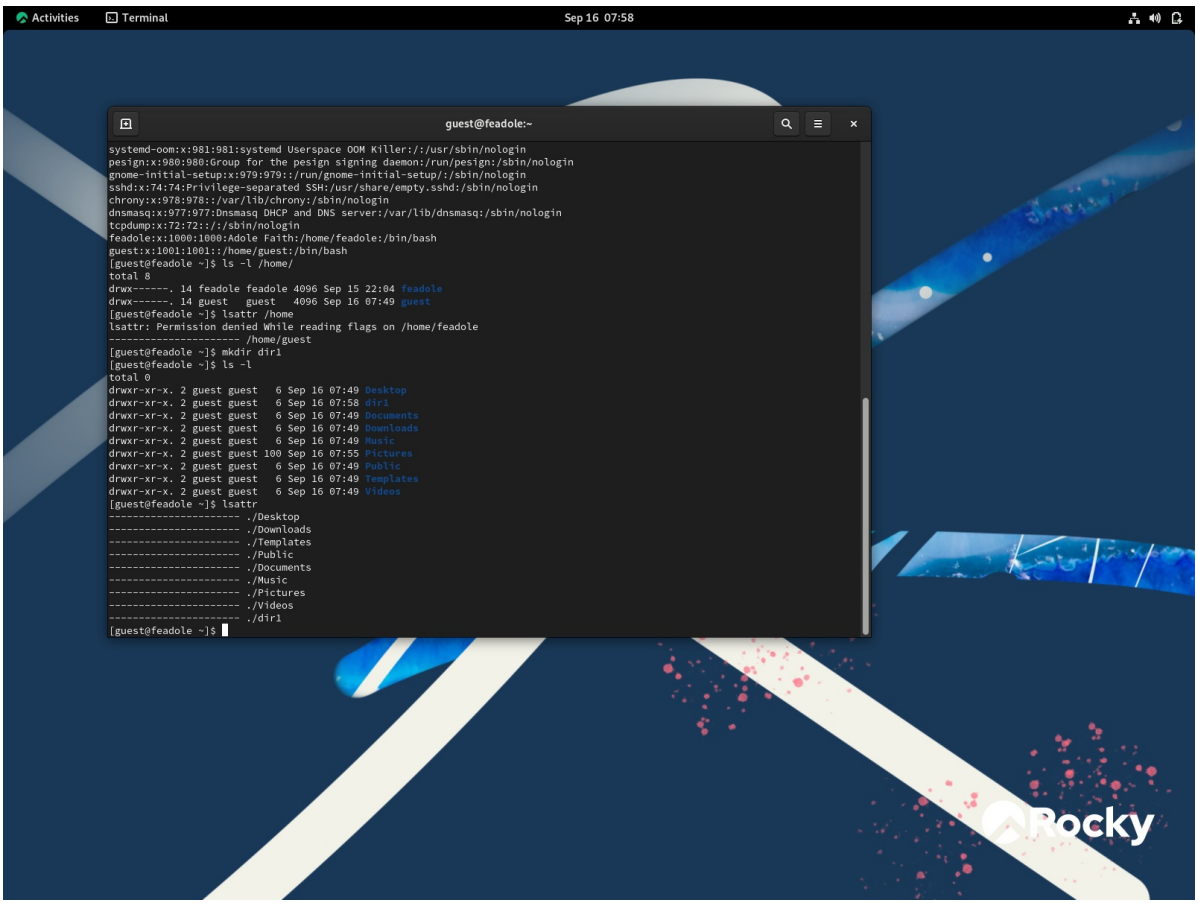

Рис. 5: Рис. 3.5: Содержание файла /etc/passwd

Посмотрела, какие директории существуют в системе командой “ls -l /home/” (рис. 3.6). Список поддиректорий директории /home получить удалось. На директориях установлены права чтения, записи и выполнения для самого пользователя (для группы и остальных пользователей никаких прав доступа не установлено). Проверила, какие расширенные атрибуты установлены на поддиректориях, находящихся в директории /home, командой “lsattr /home” (рис. 3.6). Удалось увидеть расширенные атрибуты только директории того пользователя, от имени которого я нахожусь в системе. Создала в домашней директории поддиректорию dir1 командой “mkdir dir1” и определила, какие права доступа и расширенные атрибуты были на неё выставлены: чтение, запись и выполнение доступны для самого пользователя и для группы, для остальных - только чтение и выполнение,

расширенных атрибутов не установлено (рис. 3.6).

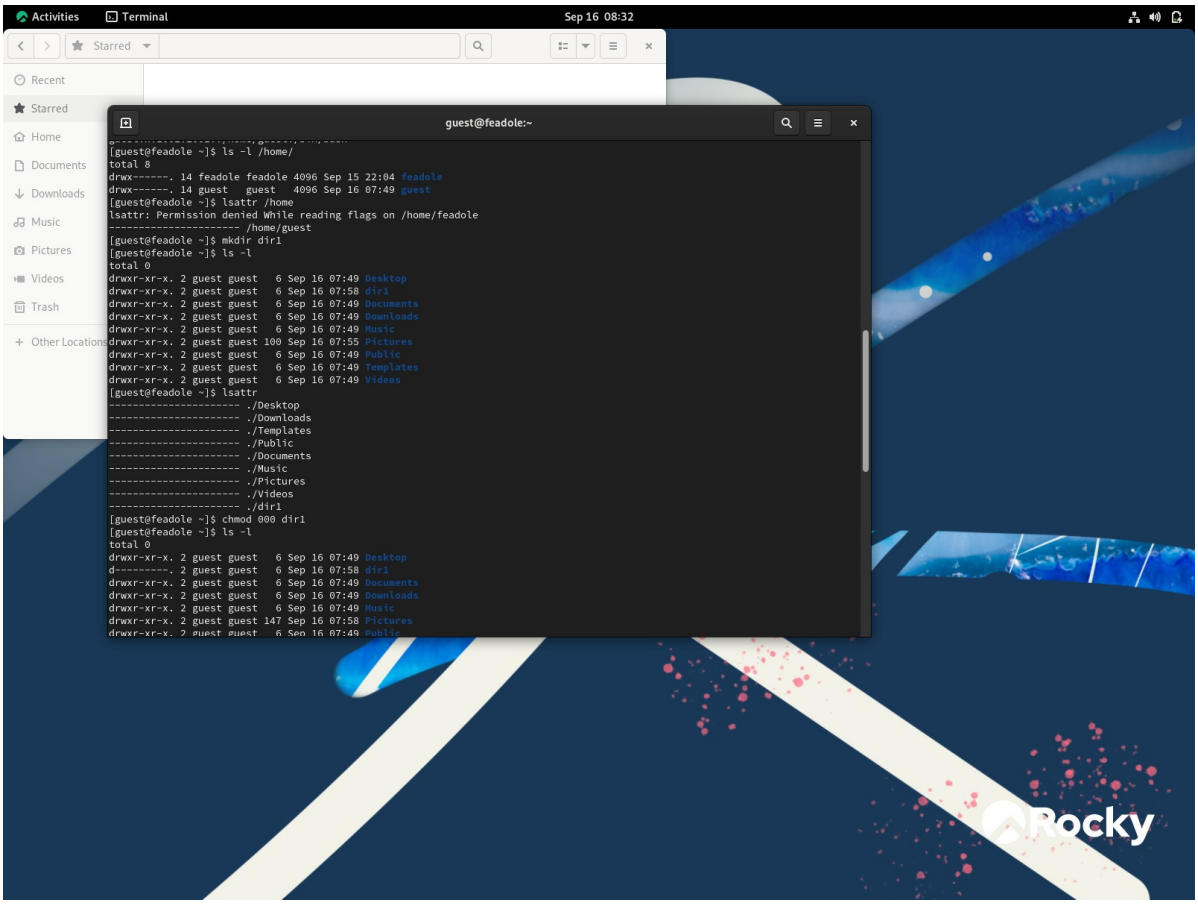

Рис. 6: Рис. 3.6: Права доступа и расширенные атрибуты

Сняла с директории dir1 все атрибуты командой “chmod 000 dir1” и проверила с её помощью правильность выполнения команды “ls -l”. Действительно, все атрибуты были сняты (рис. 3.7). Попыталась создать в директории dir1 файл file1 командой echo “test” > /home/guest/dir1/file1 (рис. 3.7). Этого сделать не получилось, т.к. предыдущим действием мы убрали право доступа на запись в директории. В итоге файл не был создан (открыть директорию с помощью команды “ls -l /home/guest/dir1” изначально тоже не удалось по той же причине, поэтому я поменяла права доступа и снова воспользовалась этой командой, и тогда смогла просмотреть содержимое директории, убедившись, что файл не был создан).

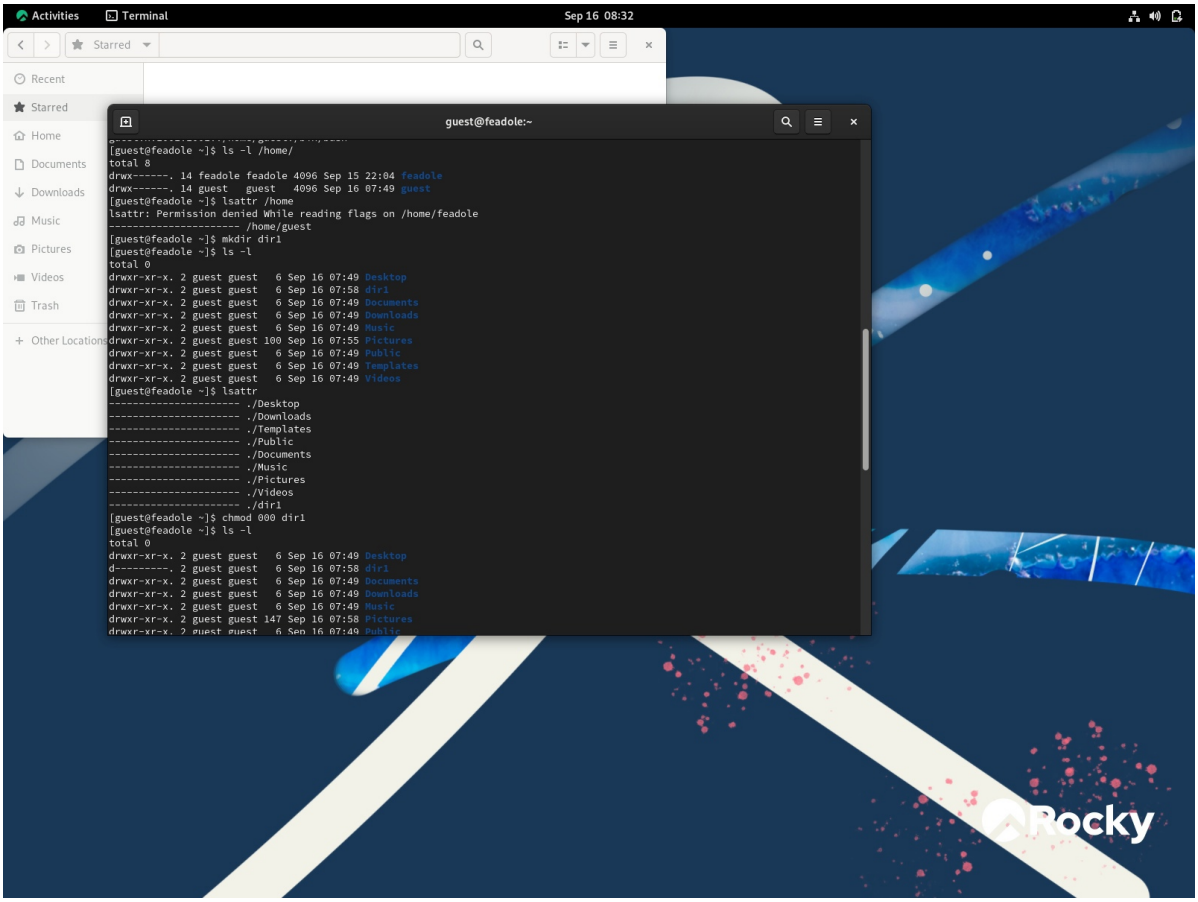

Рис. 7: Рис. 3.7: Попытка создать файл в директории

Заполним таблицу «Установленные права и разрешённые действия» 3.1. Создание файла: “echo”text” > /home/guest/dir1/file2” Удаление файла: “rm -r /home/guest/dir1/file1” Запись в файл: “echo”textnew” > /home/guest/dir1/file1” Чтение файла: “cat /home/guest/dir1/file1” Смена директории: “cd dir1” Просмотр файлов в директории: “ls dir1” Переименование файла: “mv /home/guest/dir1/file1 filenew” Смена атрибутов файла: “chattr -a /home/guest/dir1/file1”

0.5 Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы в консоли с атрибутами файлов, закрепила теоретические основы

Таблица 1: Установление права и разрешённых действий

Права директории	000	100	200	300	400	500	600	700
Права файла	000	100	200	300	400	500	600	700
Создание файла	-	-	-	+	-	-	-	+
Удаление файла	-	-	-	+	-	-		+
Запись в файл	-	+	-	+	-	+	-	+
Чтение файла	-	+	-	+	-	+	-	+
Смена директории	-	-	-	+	-	+	-	+
Просмотр файлов в директории	-	-	-	-	+	+	+	+
Переименование файла	-	-	-	+	-	-	-	+
Смена атрибутов файла	-	-	-	+	-	-	-	+

дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux.

0.6 Список литературы

Права доступа к файлам в Linux [Электронный ресурс]. 2019. URL: https://losst.ru/prava-dostupa-k-fajlam-v-linux.