

Отчёт по лабораторной работе 4

Дискреционное разграничение прав в Linux. Расширенные атрибуты

Максимова Ксения НБИбд-02-18

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	11
5	Выводы	20
	Список литературы	21

List of Figures

4.1	Рис 1.Начальная проверка расширенных атрибутов	11
4.2	Рис 2.Установка прав на файл file1	12
4.3	Рис 3.Расширенный атрибут а	13
4.4	Рис 4.Расширенный атрибут а	14
4.5	Рис 5.Проверка	14
4.6	Рис 6.Запись	15
4.7	Рис 7.Проверка	15
4.8	Рис 8.Удаление файла	16
4.9	Рис 9.Изменение атрибутов	17
4.10	Рис 10.Изменение атрибутов	17
4.11	Рис 11.Проверка без атрибута	18
4.12	Рис 12.Проверка без атрибута	18
4.13	Рис 13.Проверка без атрибута	19

1 Цель работы

Получение практических навыков работы в консоли с расширенными атрибутами файлов.

2 Задание

От имени пользователя, не имеющего прав администратора, проверить, какие действия можно выполнять с файлом при установленном расширенном атрибуте а или і.

3 Теоретическое введение

В операционной системе Linux есть много отличных функций безопасности, но она из самых важных - это система прав доступа к файлам. Linux в отличие от Windows, изначально проектировался как многопользовательская система, поэтому права доступа к файлам в linux продуманы очень хорошо [1] .

В Linux так же существуют атрибуты файлов — это свойства метаданных, которые описывают поведение файла. Например, атрибут может указывать, сжат ли файл, или указывать, можно ли удалить файл [3].

Файловые атрибуты могут использовать администраторы и пользователи для защиты файлов от случайных удалений и изменений, а также их применяют злоумышленники, делая невозможным удаление вредоносного файла [2].

Различают следующие виды расширенных атрибутов [2]:

1. “а” - Файл с установленным атрибутом «а» можно открыть только в режиме добавления для записи. Только суперпользователь или процесс, обладающий возможностью CAP_LINUX_IMMUTABLE, может установить или очистить этот атрибут.
2. “А” - При доступе к файлу с установленным атрибутом «А» его запись atime не изменяется. Это позволяет избежать определённого количества дисковых операций ввода-вывода для портативных систем.
3. “с” - Файл с установленным атрибутом «с» автоматически сжимается на диске ядром. При чтении из этого файла возвращаются несжатые данные. Запись в этот файл сжимает данные перед их сохранением на диске.

4. “C” - Файл с установленным атрибутом «C» не подлежит обновлению «копирование при записи». Этот флаг поддерживается только в файловых системах, которые выполняют копирование при записи.
5. “d” - Файл с установленным атрибутом «d» не является кандидатом для резервного копирования при запуске программы dump.
6. “D” - При изменении каталога с установленным атрибутом «D» изменения синхронно записываются на диск; это эквивалентно опции монтирования `dirsync`, применяемой к подмножеству файлов.
7. “e” - Атрибут «e» указывает, что файл использует экстенды для отображения блоков на диске. Его нельзя удалить с помощью `chattr`.
8. “E” - Файл, каталог или символическая ссылка с установленным атрибутом «E» зашифрованы файловой системой. Этот атрибут нельзя установить или сбросить с помощью `chattr`, хотя он может быть отображён с помощью `lsattr`.
9. “F” - Директория с установленным атрибутом «F» указывает, что все поиски путей внутри этого каталога выполняются без учёта регистра. Этот атрибут можно изменить только в пустых каталогах в файловых системах с включённой функцией `casefold`.
10. “i” - Файл с атрибутом «i» не может быть изменён: его нельзя удалить или переименовать, нельзя создать ссылку на этот файл, большую часть метаданных файла нельзя изменить, и файл нельзя открыть в режиме записи. Только суперпользователь или процесс, обладающий возможностью `CAP_LINUX_IMMUTABLE`, может установить или очистить этот атрибут.
11. “I” - Атрибут «I» используется кодом `htree`, чтобы указать, что каталог индексируется с использованием хешированных деревьев. Его нельзя установить или очистить с помощью `chattr`, хотя его можно отобразить с помощью `lsattr`.

12. “j” - Файл с атрибутом «j» имеет все данные, записанные в журнал ext3 или ext4 перед записью в сам файл, если файловая система смонтирована с параметрами «data=ordered» или «data=writeback» и файловая система имеет журнал. Если файловая система смонтирована с параметром «data=journal», все данные файла уже занесены в журнал, и этот атрибут не действует. Только суперпользователь или процесс, обладающий возможностью CAP_SYS_RESOURCE, может установить или очистить этот атрибут.
13. “m” - Файл с атрибутом «m» исключается из сжатия в файловых системах, которые поддерживают сжатие файлов.
14. “N” - Файл с установленным атрибутом «N» указывает, что файл содержит данные, хранящиеся внутри самого inode. Его нельзя установить или очистить с помощью chattr, хотя его можно отобразить с помощью lsattr.
15. “P” - Директория с установленным атрибутом «P» будет обеспечивать иерархическую структуру для идентификаторов проектов. Это означает, что файлы и каталоги, созданные в директории, будут наследовать идентификатор проекта каталога, операции переименования ограничены, поэтому, когда файл или каталог перемещается в другой каталог, идентификаторы проекта должны совпадать. Кроме того, жёсткая ссылка на файл может быть создана только в том случае, если идентификатор проекта для файла и целевой каталог совпадают.
16. “s” - Когда файл с установленным атрибутом «s» удаляется, его блоки обнуляются и записываются обратно на диск. Примечание: обязательно прочтите об ошибках и ограничениях в конце этого раздела.
17. “S” - При изменении файла с установленным атрибутом «S» изменения синхронно записываются на диск; это эквивалентно опции монтирования «sync», применяемой к подмножеству файлов.

18. “t” - Файл с атрибутом «t» не будет иметь фрагмент частичного блока в конце файла, объединённого с другими файлами (для тех файловых систем, которые поддерживают объединение хвостов).
19. “T” - Директория с атрибутом «T» будет считаться вершиной иерархии каталогов для целей распределителя блоков Орлова. Это подсказка распределителю блоков, используемому ext3 и ext4, что подкаталоги в этом каталоге не связаны и, следовательно, должны быть разделены для целей распределения.
20. “u” - Когда файл с установленным атрибутом «u» удаляется, его содержимое сохраняется. Это позволяет пользователю запрашивать его восстановление.
21. “x” - Атрибут «x» может быть установлен для каталога или файла. Если атрибут установлен в существующем каталоге, он будет унаследован всеми файлами и подкаталогами, которые впоследствии будут созданы в каталоге. Если существующий каталог содержал некоторые файлы и подкаталоги, изменение атрибута в родительском каталоге не изменяет атрибуты этих файлов и подкаталогов.
22. “V” - Для файла с установленным атрибутом «V» включена функция проверки подлинности. Он не может быть записан, и файловая система будет автоматически проверять все данные, считанные из неё, по криптографическому хешу, который покрывает всё содержимое файла, например через дерево Меркла. Это позволяет эффективно аутентифицировать файл.

Изменить атрибуты файла можно с помощью команды `chattr`.

Команда `chattr` имеет следующую общую формулу[3].:

`chattr OPTIONS OPERATOR ATTRIBUTES FILE...`

Значение части `OPERATOR` может быть одним из следующих символов:

- “+” — Оператор «плюс» сообщает `chattr` о необходимости добавления указанных атрибутов к существующим.

- “-” — Оператор минус указывает chatr удалить указанные атрибуты из существующих.
- “=” — Оператор равенства сообщает chatr о необходимости установить указанные атрибуты как единственные.

4 Выполнение лабораторной работы

От имени пользователя guest определите расширенные атрибуты файла file1 командой `lsattr file1`. На момент начала выполнения лабораторной работы на файле file1 не было установлено никаких расширенных атрибутов.

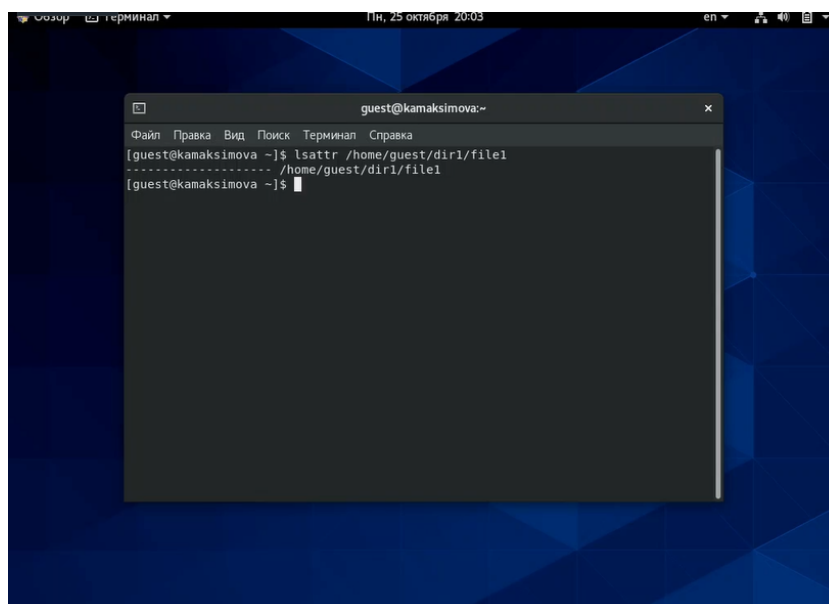


Figure 4.1: Рис 1. Начальная проверка расширенных атрибутов

Рисунок 1

Установите командой `chmod` на файл file1 права, разрешающие чтение и запись для владельца файла

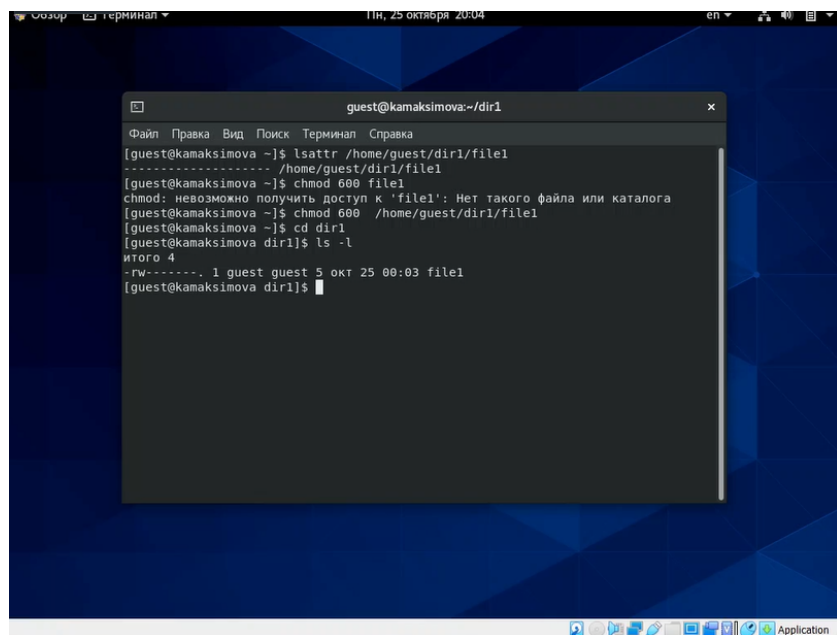


Figure 4.2: Рис 2.Установка прав на файл file1

Рисунок 2

Попробуйте установить на файл file1 расширенный атрибут а от имени пользователя guest В выполнении данной команды было отказано, так как на это у пользователя guest не достаточно прав

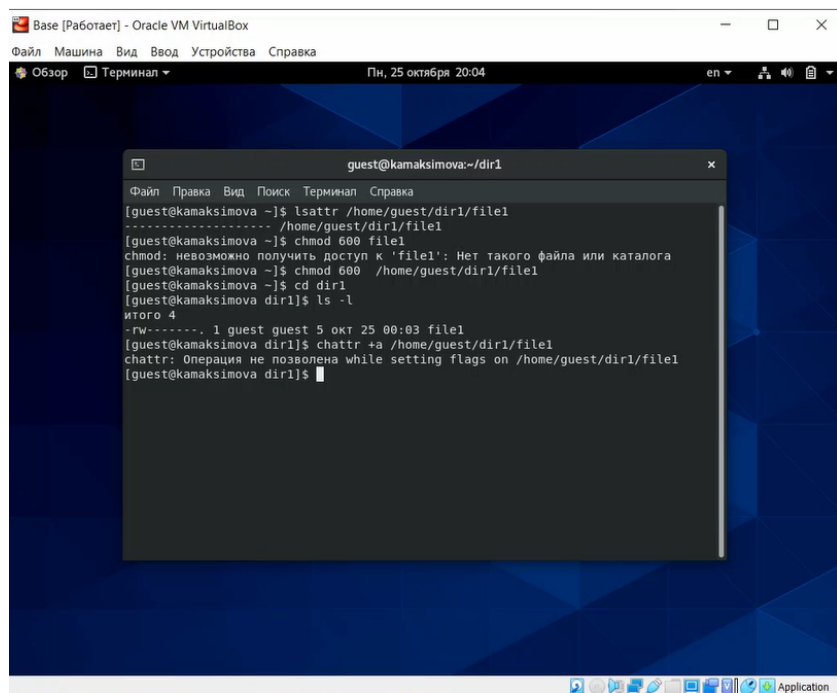


Figure 4.3: Рис 3.Расширенный атрибут а

Рисунок 3

Попробуйте установить расширенный атрибут а на файл file1 от имени супер-пользователя. Я устанавливала от имени пользователя kamaksimova

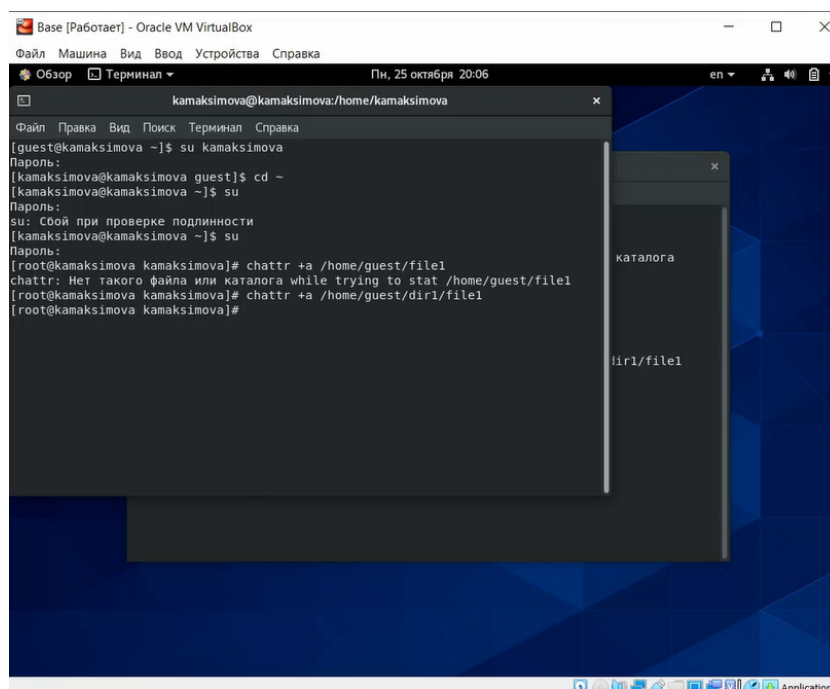


Figure 4.4: Рис 4.Расширенный атрибут а

Рисунок 4

От имени пользователя guest проверьте правильность установления атрибута
Атрибут установился верно

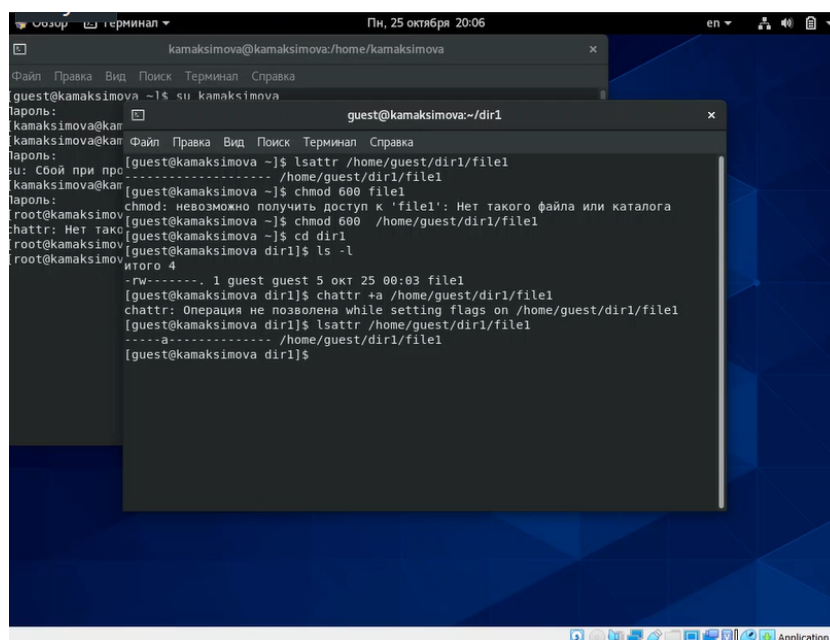


Figure 4.5: Рис 5.Проверка

Рисунок 5

Выполните дозапись в файл file1 слова «test» командой echo. После этого выполните чтение файла file1 командой cat. Слово успешно записалось

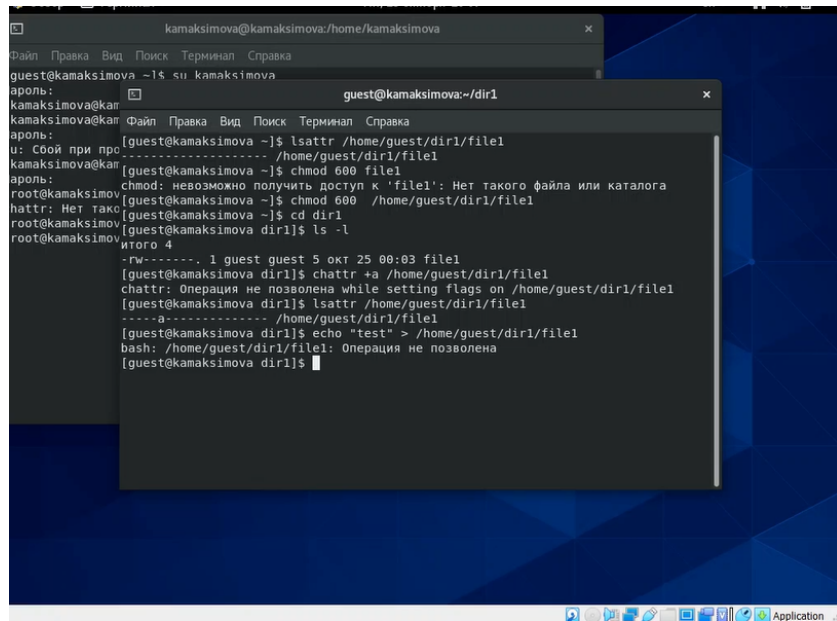


Figure 4.6: Рис 6.Запись

Рисунок 6

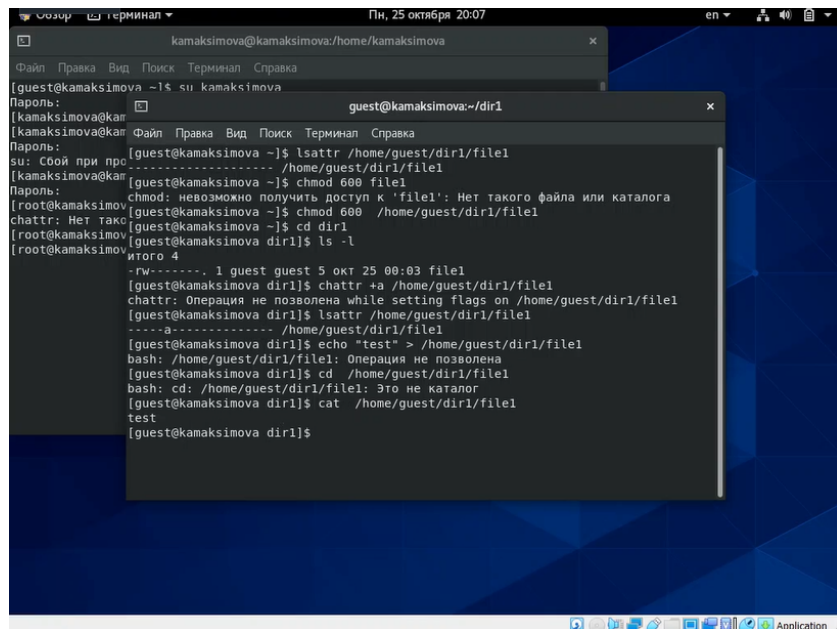
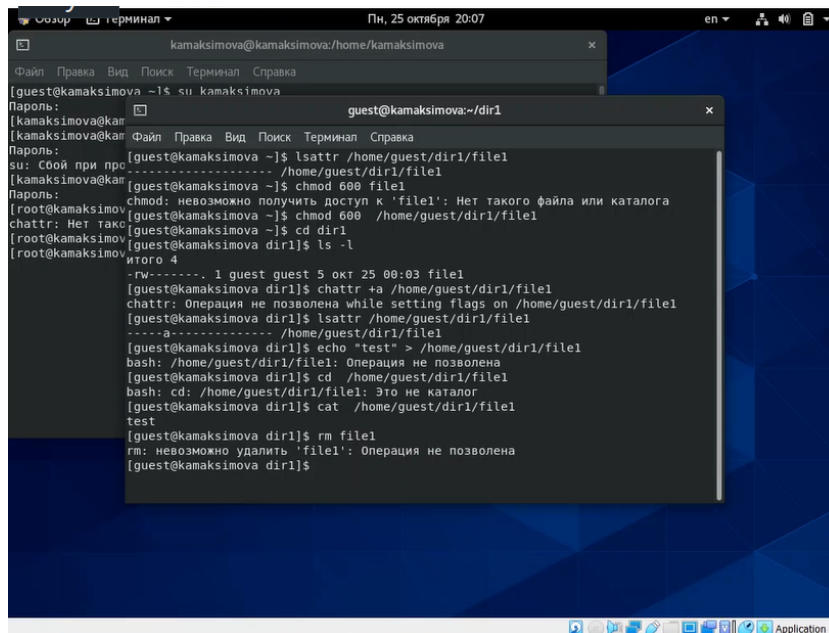


Figure 4.7: Рис 7.Проверка

Рисунок 7

Попробуйте удалить файл file1 либо стереть имеющуюся в нём информацию. В доступе отказано



```
kamaksimova@kamaksimova:~/kamaksimova
[guest@kamaksimova ~]$ su kamaksimova
Пароль:
[kamaksimova@kamaksimova ~]$ cd /home/guest/dir1/file1
[kamaksimova@kamaksimova ~]$ chmod 600 file1
chmod: невозможно получить доступ к 'file1': Нет такого файла или каталога
[kamaksimova@kamaksimova ~]$ chmod 600 /home/guest/dir1/file1
chattr: Нет такого файла или каталога
[kamaksimova@kamaksimova ~]$ cd dir1
[kamaksimova@kamaksimova dir1]$ ls -l
итого 4
-rw-----. 1 guest guest 5 окт 25 00:03 file1
[kamaksimova@kamaksimova dir1]$ chattr +a /home/guest/dir1/file1
chattr: Операция не позволена while setting flags on /home/guest/dir1/file1
[kamaksimova@kamaksimova dir1]$ lsattr /home/guest/dir1/file1
----a----- /home/guest/dir1/file1
[kamaksimova@kamaksimova dir1]$ echo "test" > /home/guest/dir1/file1
bash: /home/guest/dir1/file1: Операция не позволена
[kamaksimova@kamaksimova dir1]$ cd /home/guest/dir1/file1
bash: cd: /home/guest/dir1/file1: Это не каталог
[kamaksimova@kamaksimova dir1]$ cat /home/guest/dir1/file1
test
[kamaksimova@kamaksimova dir1]$ rm file1
rm: невозможно удалить 'file1': Операция не позволена
[kamaksimova@kamaksimova dir1]$
```

Figure 4.8: Рис 8.Удаление файла

Рисунок 8

Попробуйте с помощью команды chmod установить на файл file1 права, например, запрещающие чтение и запись для владельца файла В доступе отказано

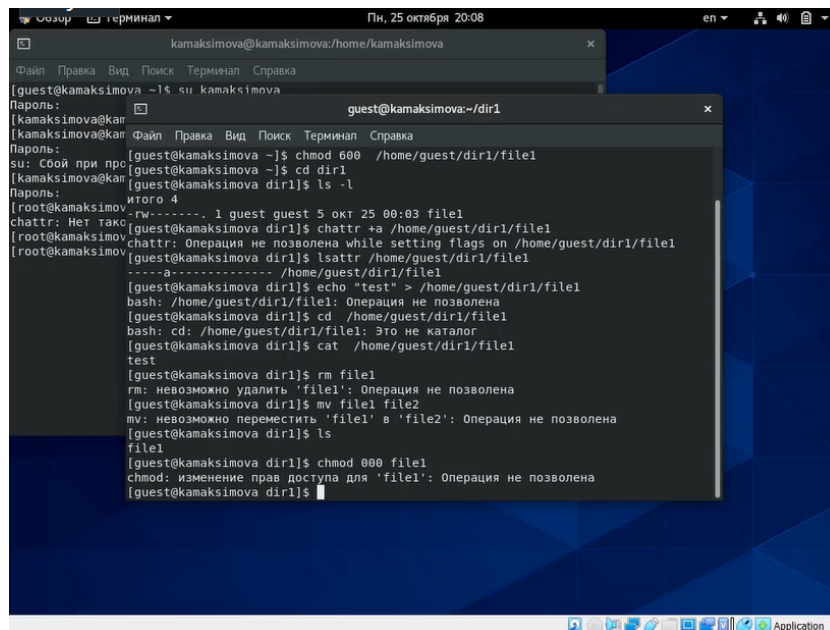


Figure 4.9: Рис 9.Изменение атрибутов

Рисунок 9

Снимите расширенный атрибут а с файла file1 от имени суперпользователя

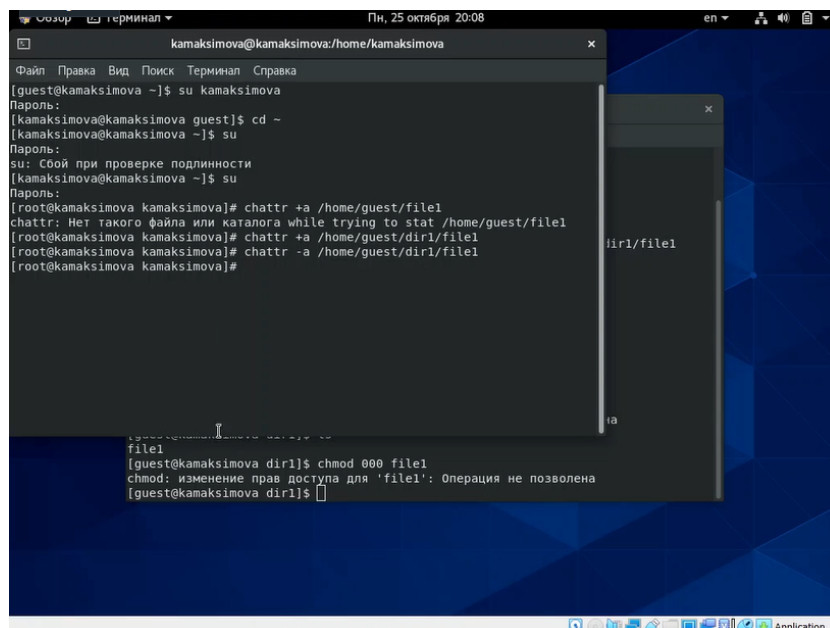


Figure 4.10: Рис 10.Изменение атрибутов

Рисунок 10

Повторите операции, которые вам ранее не удавалось выполнить

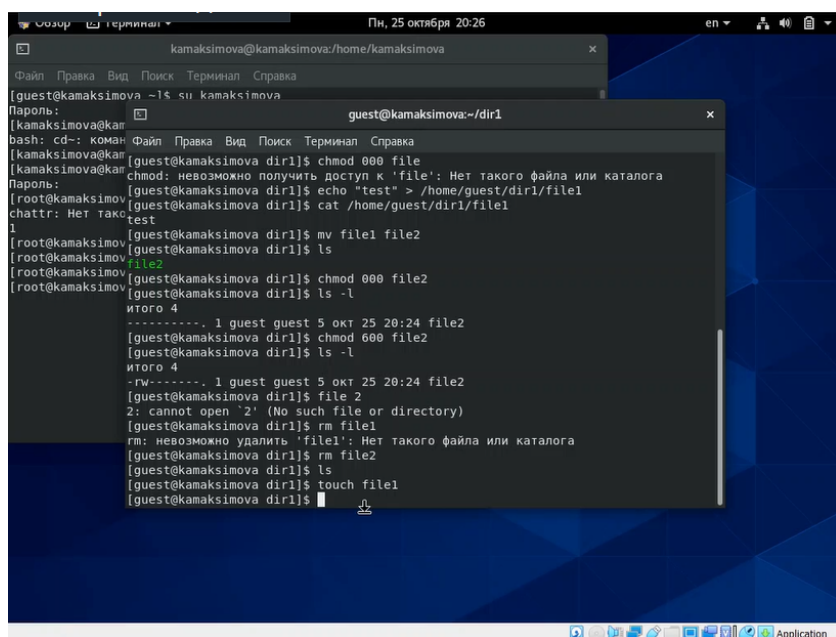


Figure 4.11: Рис 11.Проверка без атрибута

Рисунок 11

Устанавливаем на файл file1 расширенный атрибут i

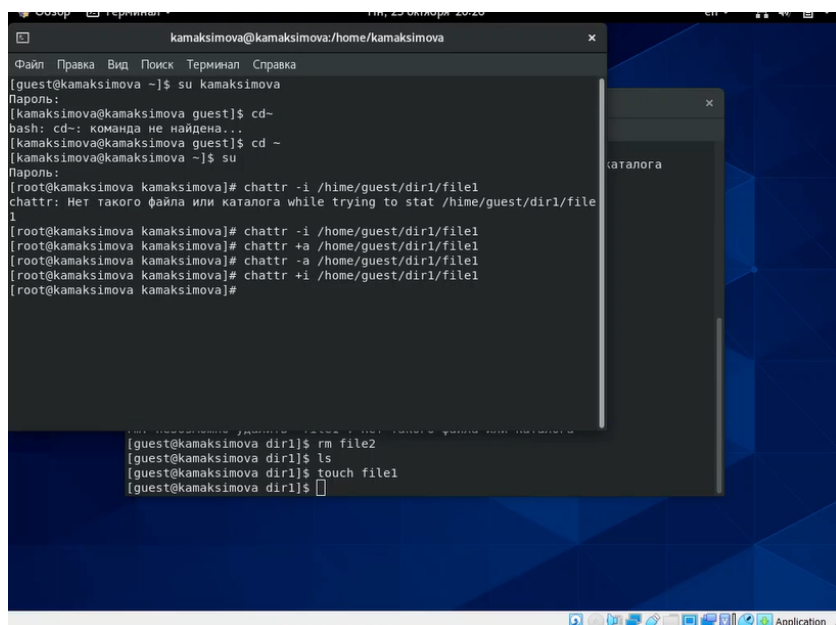


Figure 4.12: Рис 12.Проверка без атрибута

Рисунок 12

Попробуем дозаписать информацию в файл, прочитать файл, удалить его и поменять атрибуты

Figure 4.13: Рис 13.Проверка без атрибута

Рисунок 13

5 Выводы

В результате выполнения работы мы повысили свои навыки использования интерфейса командой строки (CLI), познакомились на примерах с тем, как используются основные и расширенные атрибуты при разграничении доступа. Имели возможность связать теорию дискреционного разделения доступа (дискреционная политика безопасности) с её реализацией на практике в ОС Linux. Опробовали действие на практике расширенных атрибутов «а» и «і».

Список литературы

1. Права доступа к файлам в Linux
2. Файловые атрибуты
3. Команда Chattr в Linux