**Дисциплина «Защита в операционных системах»**
**Лабораторная работа № 8**

**Тема:** Дискреционное (избирательное) разграничение доступа (Discretionary Access Control, DAC) в ОССН Astra Linux.

**Цель:** Изучить возможности ОССН Astra Linux при конфигурировании дискреционной модели разграничения доступа.

**Порядок выполнения лабораторной работы:** работа выполняется самостоятельно под руководством преподавателя.

**Время выполнения лабораторной работы (аудиторные часы) - 2 часа.**

**Оборудование и программное обеспечение:** работа выполняется на ПЭВМ типа IBM PC с использованием стандартных функций ОССН Astra Linux.

## 1. Теоретические сведения

### 1.1 Права доступа

Отправной точкой реализации управления доступом в ОССН являлось унаследованное от ОС проекта GNU/Linux дискреционное управление доступом (DAC — Discretionary Access Control). Поэтому, несмотря на то, что в настоящее время в ОССН используются мандатные управление доступом и контроль целостности, а в перспективе ролевое управление доступом, целесообразно рассмотреть основные элементы дискреционного управления доступом, не утратившие своей актуальности в современных релизах и версиях ОССН.

Дискреционное управление доступом в ОС проекта GNU/Linux базируется на понятии владения (использовании права доступа владения) файлом, каталогом, процессом (сущностями и субъект-сессиями). Так, в файловых системах семейства EXTFS, в частности, которая по умолчанию используется в ОССН, с каждым файлом или каталогом связана учётная запись пользователя — их владельца (owner). Процесс, функционирующий от имени такой учетной записи-владельца сущности, имеет право изменять дискреционные права доступа к ней, например назначать их учетным других пользователей ОССН на основе стандарта POSIX ACL.

Для оптимизации и облегчения администрирования дискреционного управления доступом в случаях, когда к одним и тем же файлам или каталогам требуется установить одинаковые права доступа более чем для одной учётной записи пользователя, в ОССН применяются группы учётных записей пользователей. В результате для файлов и каталогов владельцем (обладателем к ним правом доступа владения) может быть задана группа. При этом для них остаются владельцами и соответствующие учетные записи пользователей. В перспективе при реализации в ОССН ролевого управления доступом вместо учетных записей пользователей и групп владельцами будут задаваться роли или административные

роли.

Таким образом, при управлении доступом в ОССН, в том числе дискреционным, администрируют следующие его элементы:

- учётные записи пользователей (account);
- группы (логические объединения учётных записей пользователей с равными дискреционными правами доступа);
- права доступа к файлам, каталогам, другим объектам доступа (сущностям и субъект-сессиям);
- режимы доступа, обеспечивающие возможность учёта ряда особенностей управления доступом.

## Права доступа к файлам

Права доступа к файлу хранятся в специальном 16-битовом поле атрибутов файла (см. рис. 1).

| Тип объекта | | | | Особые признаки | | | Права доступа | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | SUID | SGID | sticky-bit | пользователь-владелец файла **(owner)** | | | группа владельца файла **(group)** | | | остальные пользователи **(others)** | | |
| # | # | # | # | s | s | t | r | w | x | r | w | x | r | w | x |

**- -** Отсутствие флага - обычный файл
**l** - Символическая ссылка
**d** - Директория
**b** - Блочное устройство
**c** - Символьное устройство
**p** - Канал, устройство fifo
**s** - Unix-сокет

| ОСТ | BIN | Наименование | Действие |
|---|---|---|---|
| 1 | 001 | sticky | удалить файл может только владелец или root |
| 2 | 010 | SGID | файл запускается на исполнение с правами группового владельца |
| 4 | 100 | SUID | файл запускается на исполнение с правами владельца |

| ОСТ | BIN | Символьное | Права на файл | Права на каталог |
|---|---|---|---|---|
| 0 | 000 | --- | отсутствие прав | отсутствие прав |
| 1 | 001 | --x | право на исполнение | право на доступ к файлам и атрибутам |
| 2 | 010 | -w- | право на запись | отсутствие прав |
| 3 | 011 | -wx | право на запись и исполнение | все права, кроме получения имен файлов |
| 4 | 100 | r-- | право на чтение | право на получение имен файлов |
| 5 | 101 | r-x | право на чтение и исполнение | право на получение имен файлов и доступ к ним |
| 6 | 110 | rw- | право на чтение и запись | право на получение имен файлов |
| 7 | 111 | rwx | все права | все права |

Рисунок 1 - 16-битовое поле атрибутов файла

Первые четыре бита устанавливают **флаг типа объекта**, они задаются при создании файла и не могут быть изменены.

Следующие три бита хранят **особые признаки**, влияющие на запуск исполняемых файлов и некоторые иные права:

- **sticky-бит** устанавливается для каталога, установка данного флага для файлов в современных системах игнорируется. Дословно sticky обозначает «липкий», что довольно хорошо соответствует его смыслу. После установки данного флага удалить файл из каталога может только его владелец или суперпользователь, даже если на файлы и папку стоят права 777.

В качестве примера использования данного бита можно привести файлообменник, в котором организовать доступ для всех пользователей, но исключить возможность случайного или преднамеренного удаления чужих файлов. Установить sticky-бит может только суперпользователь, снять - суперпользователь или владелец каталога.

Флаги **SUID** и **SGID** позволяют любому пользователю запускать файл на

исполнение с правами его владельца или группы.

В обычных условиях файл запускается с правами текущего пользователя, что не всегда достаточно для его работы. Например, так работает утилита `passwd`, когда нужно чтобы пользователь имел возможность изменить свой пароль без повышения прав, но данная операция требует прав суперпользователя.

Если проверить права на утилиту `passwd`, то будет видна запись **rwsr-xr-x** или **4755** (см. рис. 2).

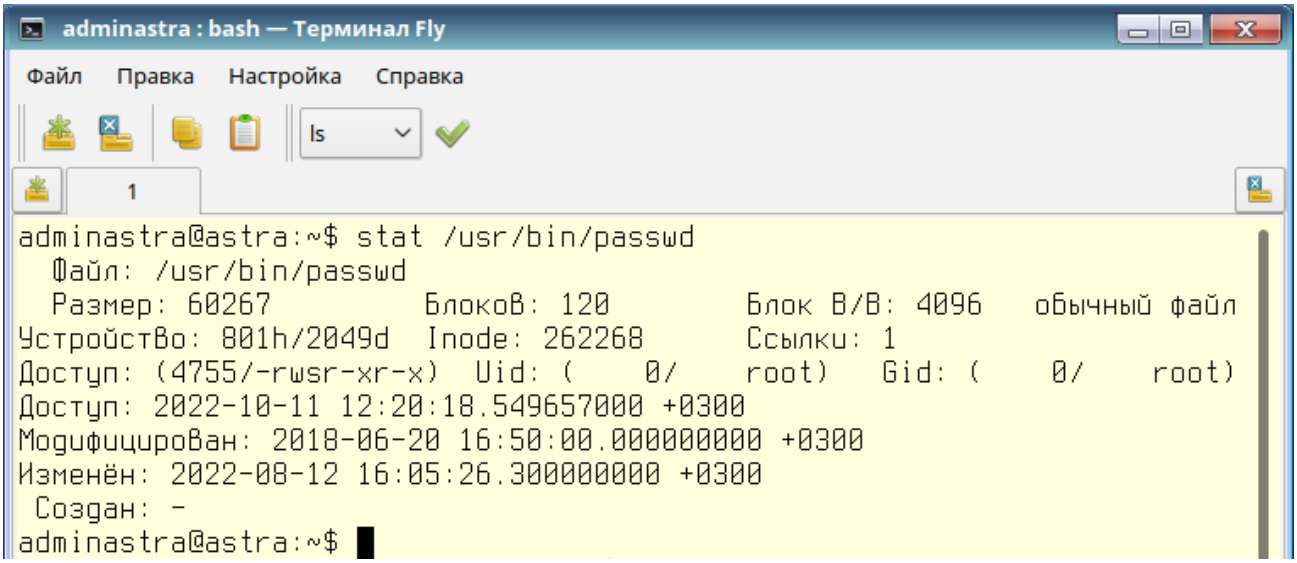


Рисунок 2 – Статистика и права доступа исполняемого файла утилиты passwd

При установке признаков SUID и SGID они заменяют символ **x** на **s** в соответствующей группе символьного представления (на рисунке 2 права доступа для субъекта-владельца) или записываются перед основными правами в восьмеричном виде.

Раздел «Права доступа» - это следующие девять бит, разделенные на блоки по три бита, которые содержат права доступа к файлу или директории.

Каждый файл в ОССН, в соответствии со стандартом POSIX ACL, должен иметь владельца (пользователь, user), группового владельца (группа, group) и остальных пользователей (остальные, other), каждый из этих пользователей может иметь права на чтение (**r**), запись (**w**) и исполнение (**x**).

Применительно к **каталогам** флаги имеют несколько иной смысл:

**r** - право получения имен файлов в каталоге (но не их атрибутов),

**x** - право получения доступа к файлам и их атрибутам,

**w** - право манипулировать именами файлов, т.е. создавать, удалять, переименовывать.

Минимальный разумный набор прав на каталог: **rx-**, так как только **r** позволит получить только имена файлов, но не их тип, размер и т.п.

Запись прав может производиться как в **символьной**, так и в **числовой форме**, для этого используют двоичное или восьмеричное (что удобнее) значение установленных битов.

Как правило, в системе используются преимущественно символьные обозначения, для целей администрирования обычно используются цифровые восьмеричные значения.

Стандартные права для вновь создаваемых файлов - **664**, папок - **775,** исключение - файлы и каталоги, созданные с правами суперпользователя, они получают **644** и **755** соответственно.

Для получения списка файлов и папок в текущей директории используется команда `ls`. Пример использования команды `ls` приведен на рисунке 3.

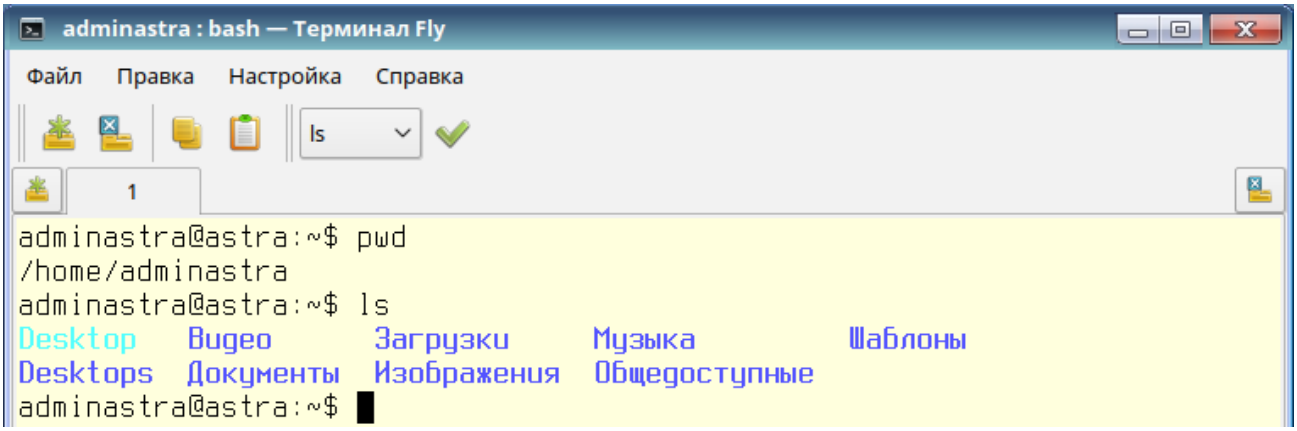


Рисунок 3 – Использование команды ls для поучения списка файлов в домашней директории пользователя

С командой `ls` наиболее часто применяются следующие ключи:

- **`l`** выдавать (в одноколоночном формате) тип файла, права доступа к файлу, количество ссылок на файл, имя владельца, имя группы, размер файла (в байтах), временной штамп и имя файла;
- **`a`** включать в список файлы с именем, начинающимся с точки (показывать скрытые файлы);
- **`C`** напечатать список файлов в колонках с вертикальной сортировкой;
- **`F`** для каждого имени каталога добавлять суффикс '/', для каждого имени FIFO — '|' и для каждого имени исполняемого файла '*'.
- **`R`** включить рекурсивную выдачу списка каталогов.
- **`c`** использовать при сортировке (при задании опции -t или -l) время изменения состояния файла вместо времени последней модификации файла.
- **`d`** выдавать имена каталогов, как будто они обычные файлы, вместо того, чтобы показывать их содержимое.
- **`i`** предварять вывод для каждого файла его серийным номером (номером inode).

При применении ключей их можно комбинировать, например, запись `ls -l -a -h` можно записать как `ls -lah` (см. рис. 4).

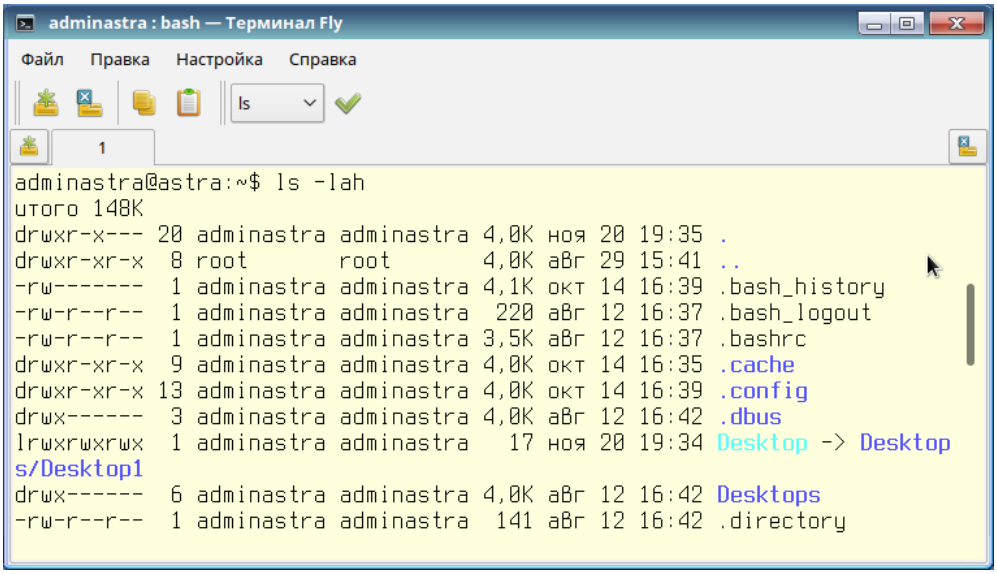


Рисунок 4 – Пример комбинирования ключей при использовании команды ls

Полный список ключей можно получить командой: `ls --help | less`

Передача вывода утилите `less` позволит выводить данные постранично, листание страниц производится клавишей **Пробел**, выход из режима кнопкой **Q**.

С помощью команды `ls` можно просмотреть содержимое любой директории, находясь в текущей директории, например:

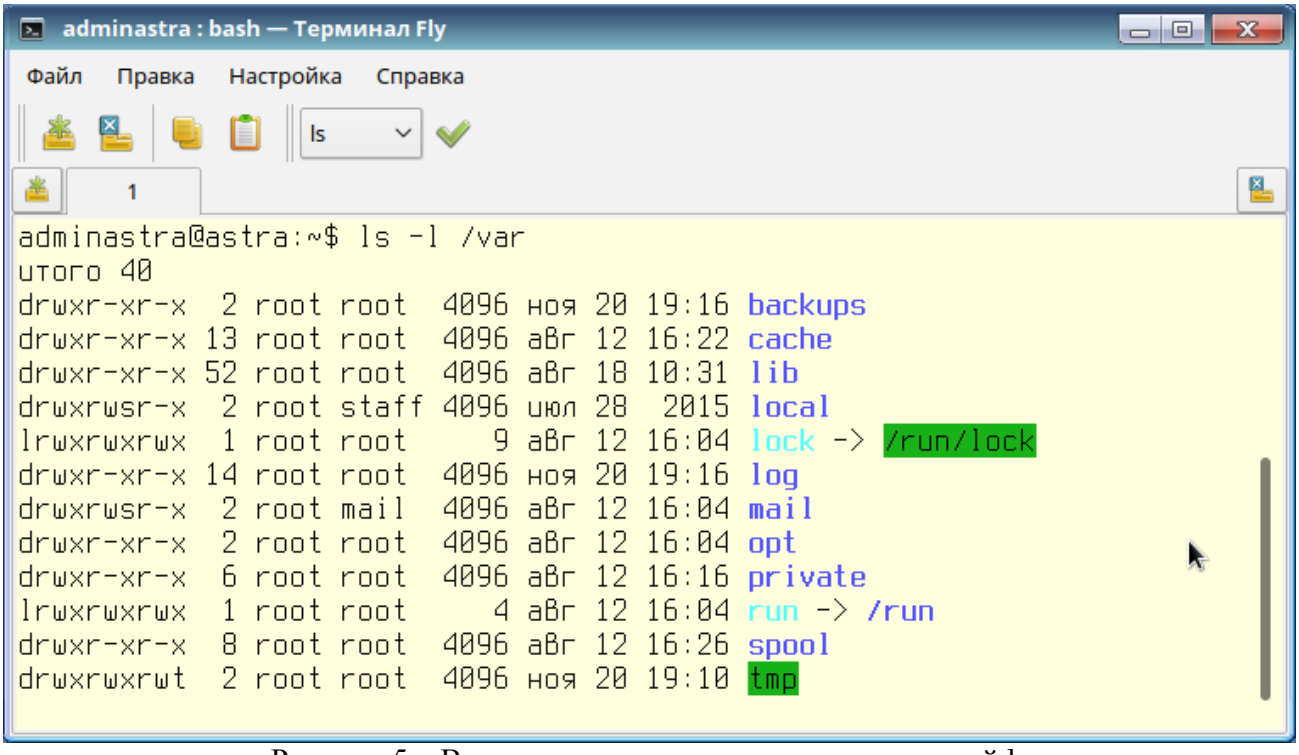


Рисунок 5 – Вывод содержимого папки var командой ls

## Изменение прав доступа

Права доступа могут быть изменены только владельцем файла или пользователем с правами администратора системы. Для изменения прав используется команда:

**chmod[u|g|o|a] [+|-|=] [r|w|x] name1 [name2 ...]**

В данном примере указан вариант задания прав доступа в символьном виде. Использование символьного вида основано на однобуквенных обозначениях, которые определяют класс доступа и права доступа для членов данного класса. Права доступа к файлу зависят от идентификатора пользователя и идентификатора группы, в которую он входит. В качестве аргументов команда принимает указание классов доступа («u» - владелец-пользователь, «g» - владелец-группа, «o» - остальные пользователи, «a» - все вышеперечисленные группы вместе), права доступа **(**«r» - чтение, «w» - запись, «x» - выполнение) и операцию, которую необходимо произвести («+» - добавить, «-» -убрать, «=» - присвоить).

Таким образом, чтобы разрешить выполнение файла ip.py, который находиться в директории /home/adminastra/Загрузки всем пользователем необходимо выполнить команду (рис. 6):

```
adminastra@astra:~$ chmod a+x Загрузки/ip.py
```

Рисунок 6 – Команда, выдающая права на исполнение файла

Далее, чтобы оставить права записи только для владельца файла необходимо выполнить (рис.7):

```
adminastra@astra:~$ chmod go-w Загрузки/ip.py
```

Рисунок 7 – Команда, позволяющая оставить права записи только для владельца файла

Рассмотрим еще несколько примеров:

- `chmod go=w ip.py` - установить право на запись для всех пользователей кроме владельца;
- `chmod a+x ip.py` - предоставить право на запись для всех пользователей;
- `chmod g+x-w ip.py` - добавить для группы право на выполнения файла, но снять право на запись.
- `chmod a+tw ip.py` или `chmod 1777 ip.py` – установка sticky-бита для файла ip.py.

Как указано ранее, права доступа могут быть представлены в абсолютной форме, в виде числовой последовательности,

Таким образом, для команды chmod 755 ip имеем (рис. 8):

```
adminastra@astra:~$ chmod 755 Загрузки/ip.py
```

Рисунок 8 - Пример использования команды chmod

## Управление файлами

В ОС Linux следует различать физическую файловую систему, которая отвечает за управление дисковым пространством и размещение файлов в физических адресах диска и логическую файловую систему, которая обеспечивает логическую структуру хранения файлов - пространство имен файлов. ОС Unix и Linux могут работать с различными физическими файловыми системами (Ext2, ext3,

ufs), логическое же представление файловой системы в Unix/Linux структурировано. Все файлы в логической файловой системе располагаются в виде дерева, промежуточные вершины которого соответствуют каталогам, и листья - файлам и пустым каталогам. Реально на каждом логическом диске (разделе физического дискового пакета) располагается отдельная иерархия каталогов и файлов. Для получения общего дерева в динамике используется «монтирование» отдельных иерархий к фиксированной корневой файловой системе в качестве ветвей общего дерева. Самым верхом иерархии является корень, который имеет предопределенное имя «/». Этот же символ используется как разделитель имен в пути. Далее в корне находятся папки с определенными для каждого дистрибутива именами (etc, home, bin, mnt, proc и т.д.).

Полное имя файла, например, /bin/sh означает, что в корневом каталоге должно содержаться имя каталога bin, а в каталоге bin должно содержаться имя файла sh. Коротким или относительным именем файла называется имя, задающее путь к файлу от текущего рабочего каталога. В каждом каталоге содержатся два специальных имени, имя «.» - ссылка на текущий каталог, и имя «...» - ссылка «родительский» каталог данного текущего каталога, т.е. каталог, непосредственно предшествующий данному в иерархии каталогов. Так, например, для структуры, показанной на рис. 9 доступ к файлу file2 из текущего каталога (laba) возможен по полному имени: /home/myvar/file2 или по относительному имени: .../.../.../myvar/file2.

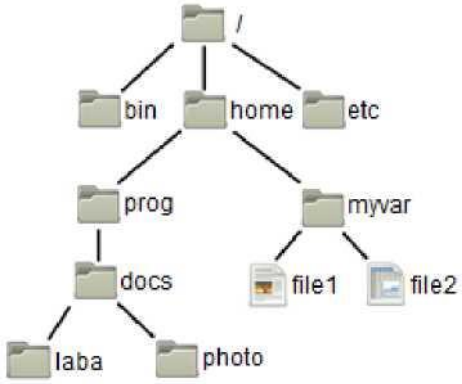


Рисунок 9 – Пример дерева каталогов

## Ссылки

Существуют жесткие и символические (мягкие) ссылки.

***Жесткая ссылка*** является просто еще одним именем для исходного файла и не является типом файла. Она прописывается в индексном дескрипторе исходного файла (в структуре, хранящей метаданные файла). После создания жесткой ссылки невозможно различить, где исходное имя файла, а где ссылка. Если вы удаляете один из этих файлов (точнее одно из этих имен), то файл еще сохраняется на диске (пока у него есть хоть одно имя - жесткая ссылка). Очень трудно различить первоначальное имя файла и позже созданные жесткие ссылки на него.

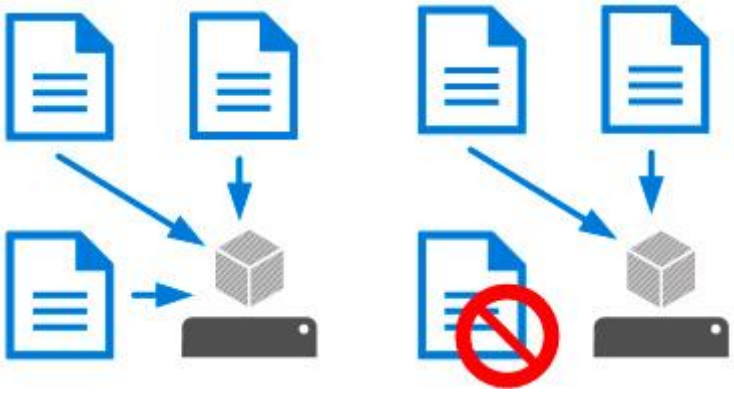

Рисунок 10 – Схематичное представление использования жестких ссылок в ОССН

Жесткие ссылки применяются там, где отслеживать различия и не требуется. Одно из применений жестких ссылок состоит в том, чтобы предотвратить возможность случайного удаления файла. Особенностью жестких ссылок является то, что они прямо указывают на номер индексного дескриптора, а, следовательно, такие имена могут указывать только на файлы внутри той же самой файловой системы (т. е., на том же самом носителе, на котором находится каталог, содержащий это имя).

***Символические (мягкие) ссылки*** тоже могут рассматриваться как дополнительные имена файлов, но в то же время они представляются отдельными файлами - файлами типа мягких ссылок и являются самостоятельным типом файла со своим inode. Однако блоки данных файла в системе представляются в одном экземпляре, у файла-ссылки адреса блоков данных те же, что и у исходного файла. В отличие от жестких ссылок мягкие ссылки могут указывать на файлы, расположенные в другой файловой системе, например, на монтируемом носителе, или даже на другом компьютере. Если исходный файл удален, мягкая ссылка не удаляется, но становится бесполезной (см. рис. 10).

Рисунок 10 – Схематичное представление использования символических ссылок в ОССН

Символические ссылки широко применяются в случаях, когда один и тот-же файл должен быть доступен под разными именами или в разных местах, но при этом его содержимое не должно меняться. Например, некое приложение требует библиотеку **lib-1.0.1.so**, в то время как есть совместимая библиотека **lib-1.0.5.so**, в этом случае необходимо создать символическую ссылку на **lib-1.0.5.so** с именем **lib-1.0.1.so** Другой случай, приложение требует **lib-1.0.5.so**, которая есть в системе, но ищет ее в другом месте, в этом случае необходимо сделать символическую ссылку с таким же именем, но в нужное расположение.

Создание любой ссылки внешне подобно копированию файла, но фактически как исходное имя файла, так и ссылка указывают на один и тот же реальный файл на диске. Поэтому, например, если вы внесли изменения в файл, обратившись к нему под одним именем, вы обнаружите эти изменения и тогда, когда обратитесь к файлу по имени-ссылке.

Для создания ссылки, используется команда `ln` (рис. 11):

```
ln [-f] файл1 [файл2 ...] целевой_файл
```

Команда `ln` делает целевой_файл ссылкой на файл1. Файл1 не должен совпадать с целевым_файлом. Если целевой_файл является каталогом, то в нем создаются ссылки на файл1, файл2,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов. Если целевой_файл существует и не является каталогом, его старое содержимое теряется.

Аргументы:

`-f` - удаление существующего целевого файла;

`-s` - создание мягкой ссылки (по умолчанию создается жесткая ссылка).

```
adminastra@astra:~$ ln -f Загрузки/ip.py Документы/ip_2.py
```

Рисунок 10 – Пример создания ссылки в ОССН

Конфигурировать права доступа можно используя GUI ОССН. Для этого необходимо на файле необходимо нажать ПКМ и перейти на вкладку «Дискреционные атрибуты» (см. рис. 12).

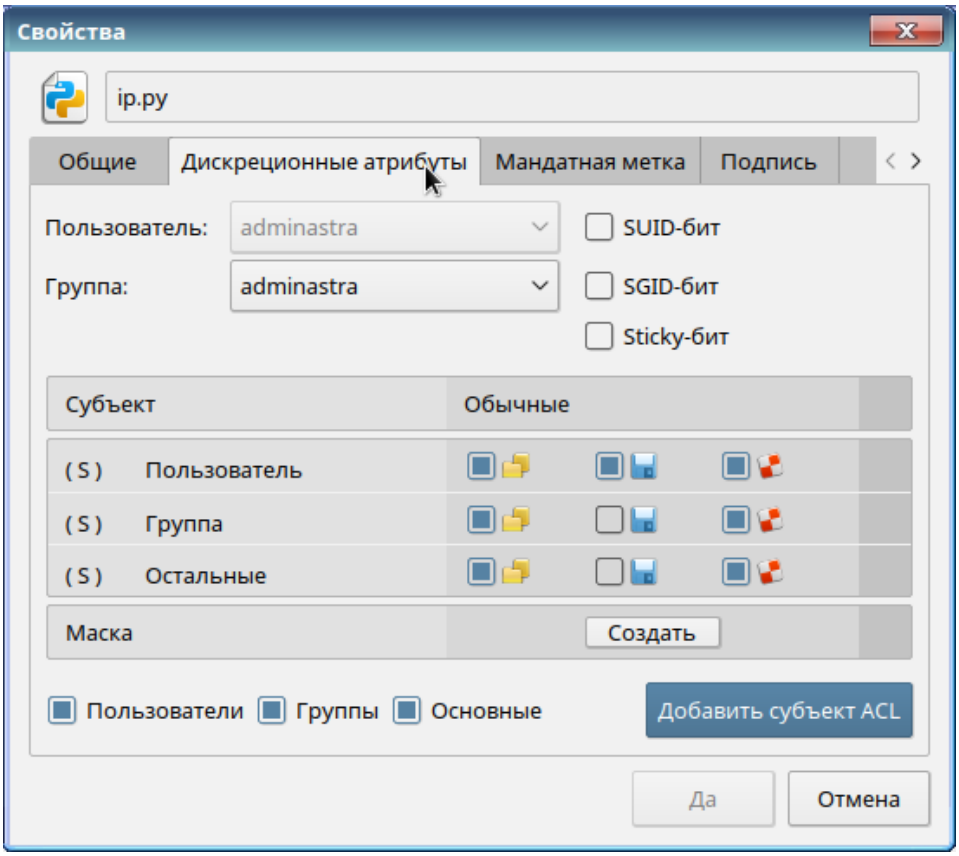


Рисунок 12 – Конфигурирование прав доступа в ОССН с использованием GUI

## 1.2 Файловый менеджер

Управление файлами также можно выполнять с помощью Midnight Commander (mc) - один из файловых менеджеров с текстовым интерфейсом типа Norton Commander для UNIX-подобных операционных систем. Запуск mc из консоли выполняется с помощью команды mc.

Установить файловый менеджер mc так, как показано на рис. 13.

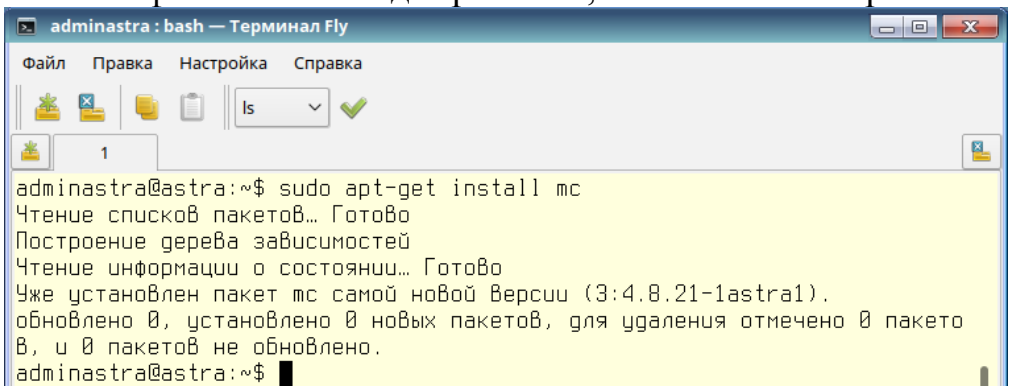


Рисунок 13 – Установка mc

Достоинство mc том, что есть встроенные средства редактирования и просмотра текстовых файлов (какими являются конфигурационные файлы). При работе с mc необходимы права суперпользователя. Общая последовательность действий по редактированию конфигурационного файла:

- запустить программу (ввод команды mc);
- используя клавиши управления курсором и клавишу «Enter», добраться до нужного файла и выбрать его;
- нажатием клавиши «F4» открыть файл для редактирования;
- внести необходимые изменения;
- сохранить их (клавиша «F2»);
- выйти из режима редактирования (клавиша «F10»).

Просмотр файла выполняется аналогично, только клавишей «F3».

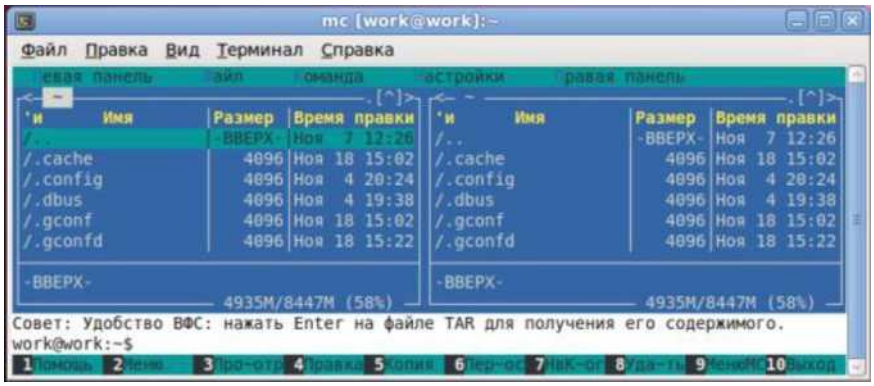


Рисунок 14 – Интерфейс Midnight Commander

Выйти из mc можно тоже клавишей «F10». Общий вид mc приведен на рис. 14.

## 2. Задание

1. Авторизоваться в ОССН в графическом режиме с учётной записью пользователя `user1` (уровень доступа — 0, неиерархические категории — нет, уровень целостности — «Высокий»).

2. Создать пакетный файл в домашней директории пользователя `user1` (если пользователя с таким именем нет – создайте его, в качестве содержимого пакетного файла можно использовать задание из лабораторной работы № 8).

3. Запустить терминал Fly.

4. Проверить текущие права доступа к созданному файлу.

5. Авторизоваться в ОССН в графическом режиме с учётной записью пользователя `user3`.

6. Попытайтесь запустить указанный файл от имени пользователя `user3`.

7. От имени пользователя `user1` назначьте полный доступ к пакетному файлу.

8. Попытайтесь запустить указанный файл от имени пользователя `user3`. Объясните полученный результат.

9. Измените пользователя `user1`, переместив его в ту же группу, к которой принадлежит `user3` (по умолчанию эта группа также называется `user3`).

10. Завершите сеанс пользователя `user3` (команда logout) и снова войдите в систему от его имени.

11. Попытайтесь снова запустить пакетный файл, созданный в пункте 3. Объясните полученный результат.

12. Удалите пользователя `user1`.

13. Для одного и тоже файла создайте мягкую и жесткую ссылку в домашнем каталоге. Попробуйте создать ссылки одновременно для нескольких файлов.

14. Проверьте работу файлового менеджера mc (в случае отсутствия файлового менеджера установите его).

15. Запишите абсолютный и символьный варианты команды, устанавливающей права доступа к файлу в соответствии с вариантом задания (номер варианта задания соответствует номеру студента в журнале группы).

| № варианта | Задание | № варианта | Задание |
|---|---|---|---|
| 1. | Доступ всем пользователям только на чтение. | 7. | Полный доступ для пользователя и группы, доступ на чтение и выполнение для остальных. |
| 2. | Доступ всем пользователям на чтение и запись. | 8. | Доступ пользователя на чтение и запись, доступ группы на чтение и выполнение, доступ только на чтение для остальных. |
| 3. | Полный доступ текущему пользователю, доступ остальных | 9. | Доступ пользователя на чтение и запись, доступ |

| | | | |
|---|---|---|---|
| | только на чтение. | | группы и остальных пользователей только на чтение. |
| 4. | Доступ для пользователя и группы на чтение и запуск, доступ остальным только на чтение. | 10. | Доступ пользователя и группы на чтение и запись, доступ остальных только на чтение. |
| 5. | Доступ всем пользователям на чтение и выполнение. | 11. | Полный доступ для пользователя, доступ на чтение и выполнение для группы, доступ на чтение и запись для остальных. |
| 6. | Полный доступ для пользователя и группы, отсутствие доступа для остальных пользователей. | 12. | Доступ на чтение и выполнение для пользователя, доступ на чтение для группы, доступ запрещен для остальных. |

| | | | |
|---|---|---|---|
| | пользователю, доступ остальных только на чтение. | | чтение и запись, доступ группы и остальных пользователей только на чтение. |
| 4. | Доступ для пользователя и группы на чтение и запуск, доступ остальным только на чтение. | 10. | Доступ пользователя и группы на чтение и запись, доступ остальных только на чтение. |
| 5. | Доступ всем пользователям на чтение и выполнение. | 11. | Полный доступ для пользователя, доступ на чтение и выполнение для группы, доступ на чтение и запись для остальных. |
| 6. | Полный доступ для пользователя и группы, отсутствие доступа для остальных пользователей. | 12. | Доступ на чтение и выполнение для пользователя, доступ на чтение для группы, доступ запрещен для остальных. |

## 3. Контрольные вопросы

1. Как организуется разграничение доступа к файлам в Linux?
2. Как изменить права доступа к файлу? Как сделать это с помощью битовой строки?

3. Что такое stisky-bit, SUID и SGID, для чего предназначены эти особые признаки?
4. Какие могут быть файловые системы в Linux?
5. Какие типы файлов существуют в Linux?
6. Что такое символические и жесткие ссылки в Linux, приведите примеры их использования.
7. Какие права могут быть назначены файлу? Назовите способы конфигурирования прав доступа в ОССН.
8. По отношению к каким категориям применяются права доступа к файлу?

## 4. Требования к отчёту

Отчёт выполняется каждым студентом индивидуально. Работа должна быть оформлена в электронном виде в формате .doc и распечатана на листах формата А4. На титульном листе указываются: наименование учебного учреждения, наименование дисциплины, название и номер работы, вариант, выполнил: фамилия, имя, отчество, группа, проверил: преподаватель ФИО (образец титульного листа представлен в приложении 1).

Отчет должен содержать:

- титульный лист;
- цель работы;
- краткие теоретические сведения, ответы на контрольные вопросы;
- описание хода выполнения работы со скриншотами;
- выводы.

**Приложение 1**

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

**Лабораторная работа № ____**
на тему «_____»

Выполнил обучающийся гр._____

_____
(Фамилия, Имя, Отчество)

Проверил:

_____
(должность, Фамилия, Имя, Отчество)

Ростов-на-Дону
20__