

Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Автоматизированные системы обработки информации и управления»

Сёмкин П.С., Сёмкин А.П.

Методические материалы к лабораторным работам
по дисциплине
«Операционные системы»

Лабораторная работа № 6
**«ОС Alt Linux. Файловые системы. Расширенные права доступа
к каталогам и файлам. Биты расширенных прав.
Списки Posix ACL».**

Москва

2024 г.

ОГЛАВЛЕНИЕ

1	ЦЕЛЬ РАБОТЫ.....	3
2	ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	3
2.1	Использования битов расширенных разрешений.....	3
2.1.1	Разрешение для файла на установку идентификатора пользователя (setuid).....	3
2.1.2	Разрешение на установку идентификатора группы(setgid).....	3
2.1.3	Разрешение на защиту файла от случайного удаления (sticky bit).....	4
2.1.4	Установка битов расширенных разрешений.....	4
2.2	Использование списков контроля доступа ACL.....	6
2.2.1	Понятие ACL.....	6
2.2.2	Изменение и просмотр настроек ACL.....	6
2.2.3	Работа с ACL по умолчанию.....	7
3	ВЫПОЛНЕНИЕ РАБОТЫ	8
3.1	Задание.....	8
3.2	Порядок выполнения работы.....	8
3.2.1	Создание учётных записей пользователей и каталогов.....	9
3.2.2	Установка прав доступа с использованием битов расширенных разрешений.....	9
3.2.3	Управление расширенными правами доступа к файлам и каталогам с использованием ACL.....	10
3.3	Ответить на контрольные вопросы.....	11
4	КОНТРОЛЬНЫЕ ВОПРОСЫ	11
5	ЛИТЕРАТУРА.....	11
6	ПРИЛОЖЕНИЕ.....	12
6.1	Команды Alt Linux для работы с пользователями.....	12
6.1.1	Создание нового пользователя.....	12
6.2	Команды Alt Linux для работы с группами пользователей.....	12
6.2.1	Создание группы пользователей.....	12
6.2.2	Добавление учётной записи пользователя в группу.....	12
6.3	Монтирование файловой системы логического тома LVM.....	12
6.4	Команды Alt Linux для работы с файлами и каталогами.....	12
6.4.1	Создание каталога.....	12
6.4.2	Смена владельца каталога.....	13
6.4.3	Смена группы владельцев каталога со всеми вложенными каталогами и файлами.....	13
6.4.4	Установка возможности создавать файлы в каталоге для членов группы.....	13
6.4.5	Установка принадлежности файлов группе пользователей.....	13
6.4.6	Установка возможности для пользователей удалять из каталога только принадлежащие им файлы.....	13
6.4.7	Создание файлов.....	13
6.5	Управление расширенными правами доступа к файлам и каталогам с использованием ACL.....	14
6.5.1	Проверка текущих прав доступа к каталогу.....	14
6.5.2	Установка прав доступа к каталогу.....	14

1 Цель работы

Целью работы является приобретение практических навыков установки расширенных прав доступа к файлам и каталогам ОС Alt Linux.

2 Теоретическая часть

2.1 Использование битов расширенных разрешений

Помимо основных разрешений доступа пользователей к каталогам и файлам, в файловых системах ОС Alt Linux есть набор расширенных разрешений. Биты таких разрешения устанавливаются по умолчанию и могут быть при необходимости настроены пользователем.

2.1.1 Разрешение для файла на установку идентификатора пользователя (setuid)

Использование атрибута **setuid (set userID upon execution)** - «установка ID пользователя во время выполнения»

Когда для файла установлен атрибут **setuid**, то пользователь, запускающий этот файл на исполнение, получает повышение прав до пользователя-владельца файла (обычно **root**) в рамках запущенного процесса. После получения повышенных прав процесс может выполнять задачи, выполнение которых обычно пользователю недоступно.

Использование **setuid** может быть потенциально опасно, т.к. может предоставлять права доступа **root**.

В некоторых файлах операционная система устанавливает данный бит по умолчанию.

2.1.2 Разрешение на установку идентификатора группы (setgid)

Использование атрибута **setgid (set group ID upon execution)** - «установка ID группы во время выполнения».

При применении к исполняемому файлу, данный бит дает пользователю, который исполняет файл, разрешения владельца группы этого файла. Таким образом, **setgid** в этом случае выполняет то же самое, что и **setuid**.

Как и в случае с разрешением **setuid**, **setgid** применяется к некоторым системным файлам в качестве настройки по умолчанию.

Когда **setgid** применяется к каталогу, его можно использовать для установки владельца группы по умолчанию для файлов и подкаталогов, созданных в этом каталоге. По умолчанию, когда пользователь создает файл, его эффективная первичная группа устанавливается как владелец группы для этого файла.

Разрешение **setgid** является очень полезным разрешением для установки в каталогах общих групп.

2.1.3 Разрешение на защиту файла от случайного удаления (*sticky bit*)

Это разрешение полезно для защиты файлов от случайного удаления, если несколько пользователей имеют права на запись в один и тот же каталог.

При использовании **sticky bit**, пользователь может удалять файлы, только если выполняется одно из следующих условий:

- пользователь является владельцем файла;
- пользователь является владельцем каталога, в котором находится файл.

Поэтому он применяется в качестве разрешения по умолчанию для каталога **/tmp** и может быть полезен также для каталогов общих групп.

2.1.4 Установка битов расширенных разрешений

Чтобы установить биты разрешений **setuid**, **setgid** и **sticky bit**, можно использовать команду **chmod**.

При этом

setuid имеет числовое значение **4**,

setgid имеет числовое значение **2**,

sticky bit имеет числовое значение **1**.

Для задания расширенных разрешений могут быть использованы **абсолютный и относительный режимы**.

В первом случае команде **chmod** необходимо задание всех устанавливаемых прав, причём первая цифра относится к расширенным разрешениям.

Пример. Следующая команда, добавит разрешение **setgid** на каталог и установит **rwx** для пользователя и **rx** для группы и других:

```
# chmod 2755 /userdir
```

В **относительном режиме** можно применить только одно какое-либо из расширенных разрешений:

1. Для **setuid** **chmod u+s**
2. Для **setgid** **chmod g+s**
3. Для **sticky bit** **chmod +t**

Права доступа	Восьмеричное представление	Символьное представление	Действие
setuid	4	u+s	Пользователь выполняет файл с разрешениями владельца файла
setgid	2	g+s	Пользователь выполняет файл с разрешениями владельца группы
sticky bit	1	+t	

Таблица 1. Способы представления расширенных прав доступа

Найти все файлы, у которых установлены биты **setuid** и **setgid** можно с помощью команд:

```
# find / -perm -u+s
```

```
# find / -perm -g+s
```

2.2 Использование списков контроля доступа ACL

2.2.1 Понятие ACL

Традиционный для Linux подход в назначении прав доступа к каталогам и файлам в виде **User Group Others** и использовании дополнительных битов **setuid**, **setgid** и **sticky bit** не всегда обеспечивает гибкость назначения прав в различных конкретных задачах.

Для повышения гибкости распределения прав доступа к объектам файловых систем используется технология назначения прав в соответствии со стандартом Posix. В этом случае используются так называемые списки контроля доступа к файловой системе (**ACL, Access Control List**) предоставляющие расширенный и более гибкий механизм.

Списки ACL предназначены для расширения прав доступа к файлам и позволяют устанавливать разрешения любым пользователям или группам пользователей для различных файлов и каталогов

Кроме того, списки ACL позволяют администраторам устанавливать разрешения по умолчанию сложным способом, при котором установленные разрешения могут различаться в разных каталогах.

2.2.2 Изменение и просмотр настроек ACL

Если нужно, чтобы создаваемые в некотором каталоге файлы принадлежали к какой-то группе и имели определенные права доступа, то для этого следует определить параметры ACL для каталога.

Чтобы увидеть текущие настройки ACL, необходимо выполнить команду **getfacl**.

Команда **ls -l** не показывает существующих ACL; она просто показывает + после списка разрешений, который указывает, что списки ACL применяются к данному файлу.

Перед настройкой списков ACL всегда полезно посмотреть текущие настройки ACL с помощью команда **getfacl**.

В результате выполнения команды **getfacl** разрешения будут показаны для трех разных объектов: **пользователя, группы и других.**

Для установки ACL используется команда **setfacl**.

Примеры использования

setfacl -m g:test:rx /dir.

-m -указывает, что текущие настройки ACL необходимо изменить.

g:test:rx -установить ACL для каталога **/dir** для чтения и выполнения (**rx**) для группы (**g**) **test**

setfacl -m u:student:rwx /dir

m -указывает, что текущие настройки ACL необходимо изменить.

u:student:rwx -дает разрешения пользователю **student** в каталоге **/dir**, не делая его владельцем и не изменяя назначение текущего владельца.

Команда **setfacl** имеет много возможностей и опций.

Если используется параметр **-R**, то происходит настройка ACL для всех файлов и подкаталогов, которые в настоящее время существуют в каталоге, где устанавливается ACL. **Рекомендуется всегда использовать эту опцию при изменении списков ACL для существующих каталогов.**

2.2.3 Работа с ACL по умолчанию

Одним из преимуществ использования списков ACL является то, что можно давать разрешения нескольким пользователям или группам в каталоге. Еще одним преимуществом является то, что можно включить наследование, работая с ACL по умолчанию.

Установив ACL по умолчанию, определяют разрешения, которые будут установлены для всех новых элементов, создаваемых в каталоге. ACL по умолчанию не меняет разрешения для существующих файлов и подкаталогов. Чтобы изменить их, нужно добавить и обычный ACL.

Чтобы установить ACL по умолчанию, просто нужно добавить опцию **d** после опции **-m** (порядок имеет значение!). Надо использовать **setfacl -m**

d:g:test:rx /data, чтобы группа **test** имела доступ на чтение и выполнение всего, что когда-либо будет создано в каталоге **/data**.

Чтобы использовать ACL для настройки доступа нескольких пользователей или групп к одному и тому же каталогу, необходимо установить ACL дважды. Сначала необходимо выполнить **setfacl -R -m**, чтобы изменить ACL для текущих файлов. Затем выполнить **setfacl -m d:**, чтобы позаботиться обо всех новых элементах, которые также будут созданы.

ACL и обычные разрешения не всегда хорошо интегрированы. Проблемы могут возникнуть, если применили ACL по умолчанию к каталогу, после чего элементы были добавлены в этот каталог, и затем попытались изменить обычные разрешения.

Изменения, которые применяются к обычным разрешениям, не будут хорошо отражены в обзоре ACL. Чтобы избежать проблем, сначала надо установить обычные разрешения, после чего установить ACL по умолчанию.

3 Выполнение работы

3.1 Задание

1. Создать учётные записи пользователей, каталог **ДЕКАНАТ** и обеспечить доступ к данному каталогу группы пользователей, используя расширенные разрешения (биты **setgid** и **sticky**).

2. Обеспечить разграничение доступа пользователей группы **group_dek** к подкаталогам каталога **ДЕКАНАТ**, используя **ACL**.

3.2 Порядок выполнения работы

1. Войти в операционную систему компьютера под учётной записью **stud_XX** (XX –индекс группы).
2. Запустить программу виртуализации **Oracle VM VirtualBox**.
3. Запустить виртуальную машину **Alt-10**.

4. Войти в гостевую операционную систему **Alt Linux** под учётной записью **root/adminroot**

3.2.1 Создание учётных записей пользователей и каталогов.

1. Создать с использованием утилиты **useradd** учётные записи пользователей **dekan, sotrudnik, admin_dek** (пароль **dekanat**)

2. Создать группу пользователей **group_dek** и включить в неё пользователей **dekan, sotrudnik, admin_dek** (включить пользователей также в группу **wheel**)

3. Войти в систему под учётной записью **admin_dek**

4. Создать в домашнем каталоге пользователя **admin_dek** точку монтирования **ДЕКАНАТ** и смонтировать в неё файловую систему логического тома **vtom_lin** (том был создан в процессе выполнения лабораторной работы № 4)

5. Установить владельцем каталога **ДЕКАНАТ** пользователя **admin_dek**

6. Установить группу владельцев каталога **ДЕКАНАТ** группу **group_dek** (со всеми вложенными каталогами и файлами)

3.2.2 Установка прав доступа с использованием битов расширенных разрешений

1. Установить, чтобы члены группы пользователей **group_dek** имели возможность создавать файлы в каталоге **ДЕКАНАТ**.

2. Установить, что все файлы, созданные в каталоге **ДЕКАНАТ**, должны принадлежать группе пользователей **group_dek**.

3. Установить, что пользователи должны иметь возможность удалять из каталога **ДЕКАНАТ** только принадлежащие им файлы.

4. Создать в каталоге **ДЕКАНАТ** каталоги

ПРИКАЗЫ и ИНФОРМАЦИЯ

5. Проверить корректность созданных каталогов.

6. Войти в систему с использованием учетных записей пользователей, состоящих в группе **group_dek**, создать файлы и получить информацию о правах доступа к ним:

- Для пользователя **dekan**:

создать файл / **ДЕКАНАТ** / **ПРИКАЗЫ** / **Приказ061121.txt**

- для пользователя **sotrudnik**

создать файл / **ДЕКАНАТ** / **ИНФОРМАЦИЯ** / **Новости.txt**

3.2.3 Управление расширенными правами доступа к файлам и каталогам с использованием ACL

Необходимо обеспечить разграничение доступа пользователей группы **group_dek** к подкаталогам каталога **ДЕКАНАТ**.

1. Войти в систему под учётной записью **admin_dek**
2. Проверить, используя команду **getfacl**, текущие права доступа к каталогу

ПРИКАЗЫ

3. Установить, используя команду **setfacl**, права доступа к каталогу **ПРИКАЗЫ** для пользователей **dekan** (полный доступ) , и **sotrudnik** (только чтение)

4. Проверить текущие права доступа к каталогу **ПРИКАЗЫ**

5. Проверить текущие права доступа к каталогу **ИНФОРМАЦИЯ**

6. Установить права доступа к каталогу **ИНФОРМАЦИЯ** для пользователей **sotrudnik** (полный доступ) и **dekan** (только чтение)

7. Проверить текущие права доступа к каталогу **ИНФОРМАЦИЯ**

3.3 Ответить на контрольные вопросы

4 Контрольные вопросы

1. Каково назначение расширенных разрешений setuid, setgid и sticky bit?
2. В чём отличие задания расширенных разрешений в абсолютном и относительном режимах?
3. Как назначаются права при использовании ACL?

5 Литература

1. Сёмкин П.С., Аксёнов А.Н. Файловые системы. Логическая организация и физическая реализация. Сборник учебно-методических работ кафедры «Системы обработки информации и управления» (бакалавры). Учебное пособие. Вып. 1./Под ред: В.М. Черненко. –М: «АртКом», 2013. – стр. 95-120
2. Сёмкин П.С., Семкин А.П. Файловые системы операционных систем Windows и Unix. Сборник учебно-методических работ кафедры «Системы обработки информации и управления» (бакалавры). Учебное пособие. Вып. 2./Под ред. В.М. Чёрненко. –М: «АртКом», 2014. – стр. 160-189
3. Семкин П.С., Семкин А.П., Горячкин Б.С. Лабораторный практикум по дисциплине «Операционные системы». Часть 1. ОС Alt Linux. Управление пользователями. Администрирование дисковой подсистемы: Учебно-методическое пособие. –М.: Издательство «Спутник+», 2023. -78 с.
4. Семкин П.С., Семкин А.П., Горячкин Б.С. Лабораторный практикум по дисциплине «Операционные системы». Часть 2. ОС Alt Linux. Файловые системы. Права доступа к каталогам и файлам. Совместное использование каталогов и файлов: Учебно-методическое пособие. –М.: Издательство «Спутник+», 2024. -38 с.
5. Документация для ОС «Альт Рабочая станция». Режим доступа: <https://www.basealt.ru/alt-workstation/docs>

6 Приложение

6.1 Команды Alt Linux для работы с пользователями

6.1.1 Создание нового пользователя

```
# useradd sotrudnik
```

```
# useradd admin_dek
```

6.2 Команды Alt Linux для работы с группами пользователей

6.2.1 Создание группы пользователей

```
# groupadd dekanat
```

6.2.2 Добавление учётной записи пользователя в группу

```
# usermod опции имя_группы имя_пользователя
```

-a – добавить пользователя в новую группу (используется с опцией **G**)

-g – назначить главной группой

G – назначить вторичной группой

```
# usermod -a -G dekanat,wheel dekan
```

```
# usermod -a -G dekanat,wheel sotrudnik
```

```
# usermod -a -G dekanat,wheel admin_dek
```

6.3 Монтирование файловой системы логического тома LVM

```
# mkdir / home / admin_dek / ДЕКАНАТ – создание точки монтирования
```

```
# mount -t ext4 / dev / mapper / vdisk1-lv1 / home / admin_dek /
```

ДЕКАНАТ – монтирование файловой системы логического тома

```
# df -m / home / admin_dek / ДЕКАНАТ – проверка дискового пространства
```

```
# ls /home/admin_dek/ДЕКАНАТ
```

6.4 Команды Alt Linux для работы с файлами и каталогами

6.4.1 Создание каталога

```
# mkdir <имя каталога> - создание каталога
```

```
# mkdir -p / ДЕКАНАТ
```

6.4.2 Смена владельца каталога

```
# chown admin_dek:group_dek / ДЕКАНАТ
```

6.4.3 Смена группы владельцев каталога со всеми вложенными каталогами и файлами

```
# chgrp -R group_dek / ДЕКАНАТ
```

6.4.4 Установка возможности создавать файлы в каталоге для членов группы

Установить, что члены группы пользователей **group_dek** должны иметь возможность создавать файлы в каталоге **ДЕКАНАТ**.

```
# chmod -R 770 / ДЕКАНАТ
```

6.4.5 Установка принадлежности файлов группе пользователей

Установить, что все файлы, созданные в каталоге **ДЕКАНАТ**, должны принадлежать группе пользователей **group_dek**.

```
# chmod -R 2770 /ДЕКАНАТ
```

2 –означает, что включён бит *setgid* и создаваемые файлы наследуют ту же группу, что и каталог, а вновь создаваемые вложенные каталоги будут наследовать *setgid* родительского

В листинге `ls -la` в разрешениях групп будет **rws**

6.4.6 Установка возможности для пользователей удалять из каталога только принадлежащие им файлы

Установить, что пользователи должны иметь возможность удалять из каталога **ДЕКАНАТ** только принадлежащие им файлы.

```
# chmod +t / ДЕКАНАТ
```

В листинге `ls -la` в разрешениях прочих будет **--T**

6.4.7 Создание файлов

```
# touch / ДЕКАНАТ / ПРИКАЗЫ /Приказ061121.txt
```

6.5 Управление расширенными правами доступа к файлам и катало- гам с использованием ACL

6.5.1 Проверка текущих прав доступа к каталогу

```
# getfacl / ДЕКАНАТ / ПРИКАЗЫ
```

6.5.2 Установка прав доступа к каталогу

```
# setfacl -m u:dekan:rwX, u:sotrudnik:r-- / ДЕКАНАТ / ПРИКАЗЫ
```

```
# setfacl -m u:sotrudnik:rwX, u:dekan:r-- / ДЕКАНАТ / ИНФОРМАЦИЯ
```