

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



**Сёмкин П.С., Сёмкин А.П.**

Методические указания по выполнению лабораторных работ  
по дисциплине  
«Операционные системы»

Лабораторная работа № 12  
«Каталоги и файлы ОС Ubuntu. Контроль доступа»

**Москва**

**2017 г.**

## ОГЛАВЛЕНИЕ

<b>1</b>	<b>ЦЕЛЬ РАБОТЫ.....</b>	<b>2</b>
<b>2</b>	<b>ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....</b>	<b>2</b>
2.1	Файлы .....	2
2.2	Каталоги .....	3
2.3	Безопасность файлов.....	3
2.4	Поиск данных с помощью пути .....	4
2.5	Работа с файлами и каталогами .....	5
2.5.1	Права доступа к файлам и папкам.....	5
2.5.2	Задание прав доступа при создании файла или каталога .....	6
2.5.3	Изменение прав доступа .....	6
2.5.4	Изменение владельца файла или каталога.....	7
<b>3</b>	<b>ЗАДАНИЕ НА ВЫПОЛНЕНИЕ РАБОТЫ.....</b>	<b>7</b>
<b>4</b>	<b>КОНТРОЛЬНЫЕ ВОПРОСЫ .....</b>	<b>7</b>
<b>5</b>	<b>ЛИТЕРАТУРА.....</b>	<b>8</b>
<b>6</b>	<b>ПРИЛОЖЕНИЕ. ОСНОВНЫЕ КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ И КАТАЛОГАМИ.....</b>	<b>8</b>
6.1	Создание файлов и каталогов .....	8
6.2	Создание жёстких и символических ссылок .....	9
6.3	Изменение прав доступа .....	9

## 1 Цель работы

Целью работы является работа с файлами и каталогами ОС Ubuntu

Продолжительность работы – 2 часа.

## 2 Теоретическая часть

### 2.1 Файлы

ОС Ubuntu поддерживает следующие типы файлов:

- **Обычный файл (regular file)** - содержит данные в некотором формате. Интерпретация содержимого производится прикладной программой. Для ОС это просто последовательность байтов.
- **Каталог (directory)** - содержит имена файлов и указатели на метаданные (номера inode)

- **Специальный файл устройства (special device file)** - обеспечивает доступ к физическому устройству путем открытия, чтения и записи в специальный файл устройства:
  - символьные файлы(character) – для небуфферизированного обмена данными
  - блочные файлы (block) – обмен данными в виде пакетов фиксированной длины - блоков
- **Именованный канал (named pipe)** - используется для связи между процессами
- **Связь(link)** -Позволяет косвенно адресовать файл (символическая связь)
- **Сокет(socket)** - предназначен для организации взаимодействия между процессами

## 2.2 Каталоги

**Каталоги** представляют собой объекты, связывающие имена файлов с номерами индексных дескрипторов файлов при помощи элементов каталогов (directory entry, dentry).

Каждый элемент каталога включает:

- номер индексного узла,
- длину элемента каталога,
- длину имени файла,
- тип файла
- имя файла

## 2.3 Безопасность файлов

Индексный узел каждого файла содержит информацию, используемую ядром для поддержки политик контроля доступа.

В файловой системе ext, индексные узлы включают два поля, связанные с безопасностью:

- **файловые разрешения (file permission)**
- **файловые атрибуты (file attribute);**

Файловые разрешения определяют права на чтение, запись и исполнение для трех категорий пользователей:

- **владелец файла** (обычно пользователь, создавший файл),
- **группы пользователей**, которые могут иметь доступ к файлу (обычно, группы, к которой принадлежит пользователь, создавший файл),
- **остальные пользователи**.

Файловые атрибуты контролируют возможность модификации данных.

Например, файловый атрибут «только добавление» (append-only) означает, что пользователи могут добавлять данные к файлу, но не могут модифицировать данные, которые уже в нем присутствуют.

Файловая система ext позволяет расширять перечень файловых атрибутов для поддержки других функций безопасности. Например, в ext в дополнительных файловых атрибутах хранятся метаданные контроля доступа, предназначенные для реализации списков контроля доступа POSIX.

## **2.4 Поиск данных с помощью пути**

Чтобы найти файл в файловой системе, необходимо выполнить преобразование «путь - индексный узел».

Пример поиск файла, расположенного по пути: **/home/admin/mydoc**.

Преобразование начинается с поиска индексного узла, представляющего корневой каталог файловой системы. Номер индексного узла корневого каталога (/) хранится в суперблоке файловой системы (и всегда равен 2). Этот номер inode задает inode корневого каталога соответствующей блочной группы. Блоки с данными, на которые ссылается inode, содержат элементы каталога для корневого каталога.

Затем файловая система осуществляет поиск элементов каталога для номера индексного узла директории **home**.

Этот процесс повторяется до тех пор, пока не будет найден индексный узел, соответствующий файлу **mydoc**.

По мере движения по иерархии индексных узлов, файловая система **проверяет информацию о разрешениях, чтобы убедиться, что доступ к данному индексному узлу разрешен.**

Доступ к файловым данным может быть получен непосредственно после того, как будет найден корректный индексный узел и файловые данные (вместе с метаданными) будут кэшированы системой.

## **2.5 Работа с файлами и каталогами**

### **2.5.1 Права доступа к файлам и папкам**

Возможности доступа к файлам, выполнения команд и перехода в тот или иной каталог можно ограничить путем настройки прав доступа для **пользователя, группы и остальных пользователей.**

При выводе на экран длинного списка (**ls -l**) файлов и каталогов в Linux первые десять отображаемых символов являются индикаторами, что представляет собой соответствующий элемент (файл, каталог и т. д.), наряду с тем, возможны ли чтение, запись и/или выполнение этого элемента.

После создания каталога и файла **первый символ** в соответствующем длинном листинге выступает в роли индикатора:

каталог(**d**); файл(**-**); символьное устройство(**c**); блочное устройство(**b**); символьная ссылка(**l**); именованный канал(**p**); сокет(**s**).

Следующие девять символов представляют права доступа, настроенные для файла и каталога.

Права доступа можно представить посредством восьмеричного числа (где чтение - это 4, запись - 2, а выполнение – 1), либо букв (rwx).

Права доступа с возможностью чтения позволяют просматривать содержимое каталога, с возможностью записи - изменять (добавлять или модифицировать) содержимое каталога, а с возможностью выполнения - переходить (иными словами, получать доступ) в каталог.

### 2.5.2 Задание прав доступа при создании файла или каталога

Права доступа, определенные для файла или каталога при их создании, настраиваются исходя из текущего значения **umask** пользователя. Используя команду **umask**, можно настроить права доступа, задаваемые в случае с файлом или каталогом при их создании.

### 2.5.3 Изменение прав доступа

Права доступа в отношении файлов или каталогов пользователя-владельца можно изменить с помощью команды **chmod**.

Первый 0 в строке режима обычно не указывается (поэтому можно ввести 777 вместо 0777). Этот заполнитель имеет особое значение. Он представляет собой восьмеричное число, которое можно использовать в случае с командами (выполняемыми файлами) в качестве индикатора того, что команда может выполняться как программа-установщик идентификатора пользователя (UID) (4), программа-установщик идентификатора группы (GID) (2) или стать битом закрепления в памяти (1). При использовании **setuid** и **setgid** команда выполняется с правами доступа, присвоенными пользователю или группе (вместо того чтобы выполняться с правами доступа пользователя или группы, к которой относится запустивший ее выполнение).

Задание бита закрепления в памяти для каталога не позволит пользователям удалять или переименовывать файлы в этом каталоге, которыми они не владеют. Однако если указать соответствующие настройки прав доступа, то пользователи смогут изменять содержимое файлов, которыми не владеют, расположенных в каталоге с битом закрепления в памяти.

Последним символом, определяющим права доступа, является **t** вместо **x** в случае с каталогом с битом закрепления в памяти. Раньше команды с битом закрепления обычно оставались в памяти, даже если при этом не использовались. Это старый UNIX-параметр, который не поддерживается в Linux.

Параметр **-R** — это полезная особенность команды **chmod**. Используя **-R**, можно рекурсивно изменять права доступа ко всем файлам каталога, начиная с той или иной точки в файловой системе.

### 2.5.4 Изменение владельца файла или каталога

При создании файла или каталога учетная запись пользователя присваивается этому файлу или каталогу. То же происходит и в отношении первичной группы. Являясь суперпользователем, можно изменить владение (пользователя) и группу, присвоенную файлу, на другого пользователя и/или группу с помощью команд **chown** и **chgrp**.

## 3 Задание на выполнение работы

1. Войти в систему под учётной записью studXX (XX – индекс группы).
2. Запустить программу Oracle VM VirtualBox
3. Запустить виртуальную машину Ubuntu
4. Войти в систему под учётной записью studXX (XX – индекс группы)
5. В каталоге **myfs2** создать папку **Документы студентов**
6. В папке **Документы студентов** создать папки **Группа ИУ5-51 - Группа ИУ5-54** и папку **Студенты 3 курса**.
7. В папках с помощью текстового редактора создать текстовые файлы
8. Установить разрешения для файлов и папок:
  - К папке **Документы студентов** разрешить доступ пользователям Stud51, Stud52, Stud53, Stud54
  - К папкам групп разрешить доступ для изменения файлов и добавления новых только для студентов соответствующих групп
  - К папке **Студенты 3 курса** разрешить полный доступ всем пользователям
9. Проверить правильность установки разрешений
10. Ответить на контрольные вопросы

## 4 Контрольные вопросы

1. Какие типы файлов поддерживает ОС Ubuntu?
2. Что такое файловые разрешения и файловые атрибуты?
3. Какие категории пользователей определяют файловые разрешения?

## 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. Негус К. Ubuntu и Debian Linux для продвинутых. 2-е изд. – СПб.: Питер, 2014. -384 с.: ил.

## 6 Приложение. Основные команды для работы с файлами и каталогами

### 6.1 Создание файлов и каталогов

**mkdir** - создание каталога

**cd** - переход в домашний каталог

**cd /home** - переход в каталог /home

**touch** <путь>имя файла - создание пустого файла

(**touch /tmp/newfile1.txt** - создание пустого файла)

**ls** - вывод списка файлов и каталогов по порядку

**ls -laX** - выводит список файлов и директорий в форматированном варианте, в том числе показывает скрытые

**ls -li file1** - вывод информации о файле(включая информацию о индексном дескрипторе

**pwd** - вывести на экран текущий каталог

**cp** – копирование файла

**cat** - вывести на экран файл (**cat <имя файла>** ) или ввод с консоли (**cat >> <имя файла>**). (остановить ввод - **Ctrl+Z** )



## 6.2 Создание жёстких и символических ссылок

**ln file1 file1-hard** - создание жёсткой ссылки

**ln -s /home/fire/file1 /home/file1-link** - создание символической ссылки **file1-link** на файл **file1**

## 6.3 Изменение прав доступа

**chmod** – изменение прав доступа к файлам и каталогам

**umask** – настройка прав доступа при создании файла или каталога

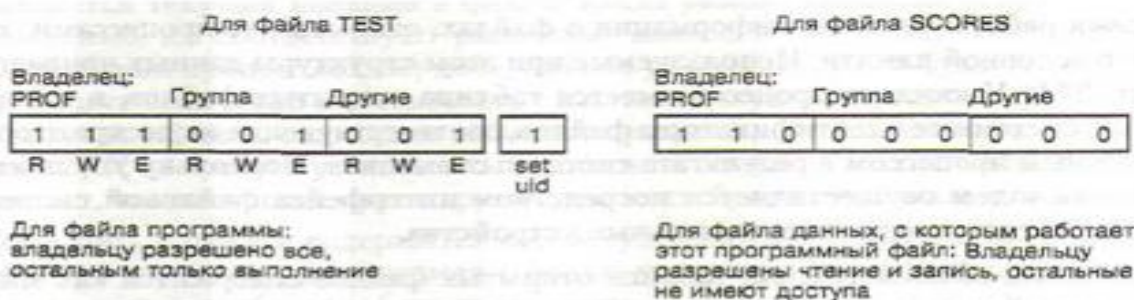
**sudo chmod 0777 /home/** - изменение права доступа к каталогу только для /home.

0777 – разрешение на чтение/запись/исполнение для всех групп;

**sudo chmod -R 0777 /home/** - рекурсивное изменение прав доступа к каталогу /home. 0777 – разрешение на чтение/запись/исполнение для всех групп. Все вложенные каталоги и файлы будут иметь такие же права 0777;



### Разрешение доступа для обычного файла



### Применение бита установки идентификатора пользователя

<b>User owner</b>	Владелец-пользователь файла. Владелец-пользователем вновь созданного файла является пользователь создавший файл.
<b>Group owner</b>	Владелец-группа файла. Порядок назначения зависит от версии.(м.б первичная группа пользователя, создавшего файл)

### Изменение владельца-пользователя

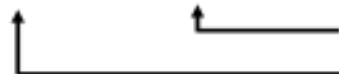
**chown**      **user1** **file1.c** **file2.c**



**список файлов**  
**имя пользователя**

### Изменение владельца-группы

**chgrp**      **group1** **file1.c** **file2.c**



**список файлов**  
**имя группы**

## Владельцы файлов

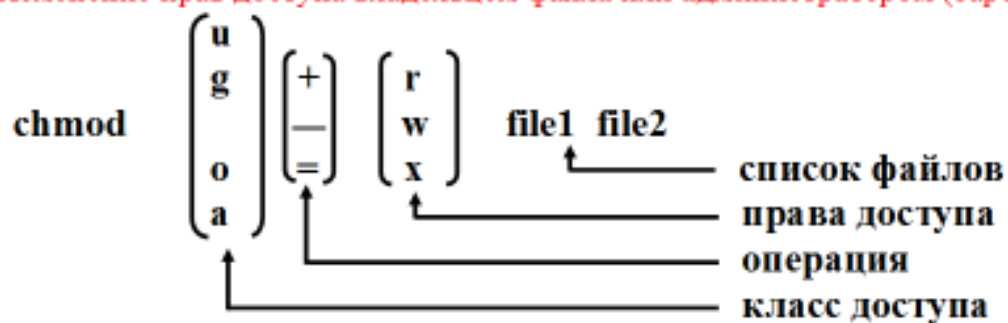
2

<b>User access(u)</b>	Для владельца-пользователя файла
<b>Group access(g)</b>	Для членов группы, являющейся владельцем файла
<b>Other access(o)</b>	Для остальных пользователей( кроме суперпользователя)

### Права доступа для каждого класса

<b>read(r)</b>	чтение
<b>write(w)</b>	запись
<b>execute(x)</b>	выполнение

Изменение прав доступа владельцем файла или администратором (superuser):



## Классы и права доступа к файлу

3