

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 № 7

**«ОС Alt Linux. Файловые системы.**

**Совместное использование каталогов и файлов. Жесткие и символические  
ссылки на файлы и каталоги»**

**Москва**

**2024 г.**

## ОГЛАВЛЕНИЕ

<b>1</b>	<b>ЦЕЛЬ РАБОТЫ.....</b>	<b>3</b>
<b>2</b>	<b>ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....</b>	<b>3</b>
2.1	Структуры данных файлов и каталогов .....	3
2.2	Совместное использование файлов .....	4
2.2.1	Жесткие ссылки ( <i>hard link</i> ).....	4
2.2.2	Символическая ссылка ( <i>soft link, symbolic link</i> ) на файл.....	6
2.2.3	Недостатки жёстких и символических ссылок. ....	7
2.3	Создание жёстких и символических ссылок в командной строке.....	8
<b>3</b>	<b>ВЫПОЛНЕНИЕ РАБОТЫ .....</b>	<b>9</b>
3.1	Задание.....	9
3.2	Порядок выполнения работы.....	9
<b>4</b>	<b>КОНТРОЛЬНЫЕ ВОПРОСЫ .....</b>	<b>10</b>
<b>5</b>	<b>ЛИТЕРАТУРА.....</b>	<b>10</b>
<b>6</b>	<b>ПРИЛОЖЕНИЕ.....</b>	<b>11</b>
6.1	Основные команды для работы с файлами и каталогами .....	11
6.1.1	Получение информации о каталогах и файлах .....	11
6.1.2	Навигация по файловой системе.....	11
6.1.3	Создание каталога .....	11
6.1.4	Создание файла.....	11
6.2	Создание жёстких и символических ссылок .....	12

## 1 Цель работы

Целью работы является знакомство с физической реализацией каталогов и файлов ОС Alt Linux и создание жёстких и символических связей с данными для совместного использования файлов и каталогов.

## 2 Теоретическая часть

### 2.1 Структуры данных файлов и каталогов

Основными объектами файловой системы ОС Alt Linux являются **каталоги** и **файлы**.

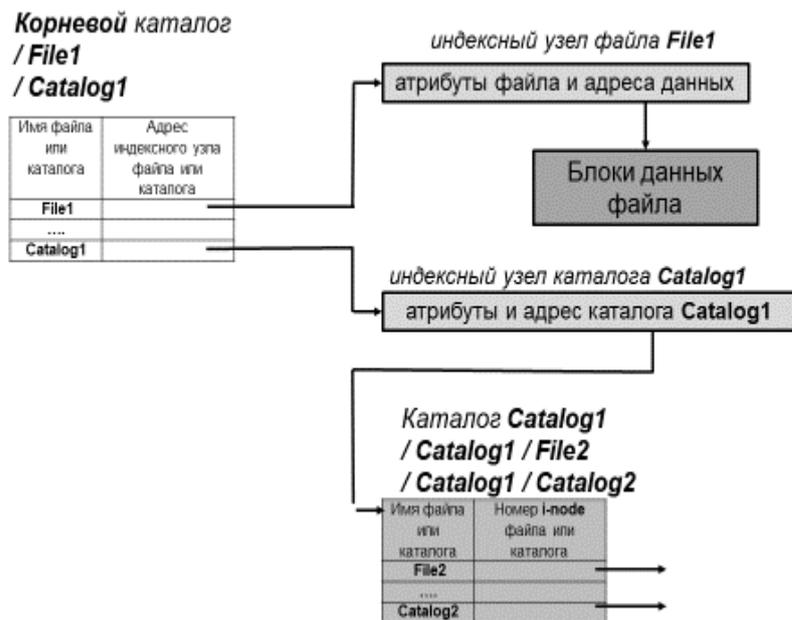
Каждый файл имеет набор данных (**данные**) и набор **атрибутов**, определяющих характеристики файла

Физическая реализация хранения файла заключается в хранении **атрибутов** и **данных файла**.

С каждым файлом связана структура данных, называемая **i-узлом** (**index node** – индексный узел). **i-узел** содержит **атрибуты** файла и **адреса блоков данных файла**.

**Каталоги** - это **системные файлы**, обеспечивающие поддержку структуры файловой системы. Каждый каталог содержит информацию о файлах и других каталогах (подкаталогах), входящих в данный каталог. Каждому файлу соответствует запись в каталоге. Эта запись содержит **имя файла** и **номер i-узла**.

В файловой системе принята иерархическая организация данных (дерево каталогов).



1

Рисунок 1. Структуры данных каталогов и файлов

## 2.2 Совместное использование файлов

Под совместным использованием файлов понимают ситуацию, когда один и тот же файл одновременно доступен в различных каталогах, принадлежащих одному и тому же или разным пользователям.

Задача совместного использования файлов в файловых системах, использующих i-узлы, имеет два решения - создание “жестких ссылок” (hard link) и “символических ссылок” (soft link) на файл.

### 2.2.1 Жесткие ссылки (hard link).

Метод жёстких ссылок предполагает создание ссылок из нескольких каталогов на один и тот же индексный узел, описывающий какой-либо файл

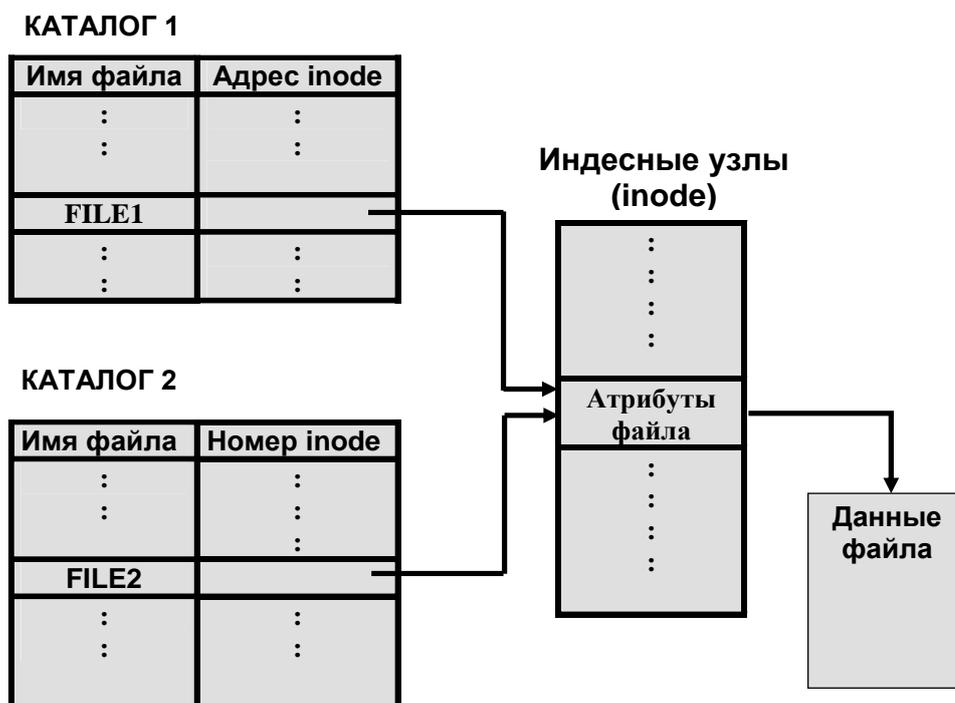
Жесткие ссылки можно создавать только для файлов (а не для файлов и каталогов) и только в пределах одной файловой системы.

Жесткие ссылки для операционных систем UNIX и Linux и реальное имя файла идентичны, после создания жесткой ссылки нельзя определить, какое имя первоначально являлось оригиналом (Любой файл всегда имеет как минимум одну жесткую ссылку – его имя, под которым он был создан.).

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное использование каталогов и файлов. Жесткие и символические ссылки на файлы и каталоги)

При создании жёсткой ссылке счётчик жёстких ссылок индексного узла увеличивается на единицу.

При удалении жесткой ссылки подсчитывается оставшееся количество ссылок, указывающих на данный индексный узел, и блоки данных файла не освобождаются до тех пор, пока не удалится его последняя жесткая ссылка.



**Рисунок 2. Жёсткие ссылки на атрибуты и данные файла**

При создании жёсткой ссылки **FILE2** в каталоге **КАТАЛОГ2** на файл **FILE1** в каталоге **КАТАЛОГ1**, в каталоге **КАТАЛОГ2** создаётся запись для **FILE2**, содержащая номер **inode** файла **FILE1**.

Свойства жёстких ссылок следующие:

- *жёсткие ссылки возможны только в пределах одной файловой системы;*
- *жёсткие ссылки возможны только на файлы, но не на каталоги;*
- *жёсткие ссылки имеют тот же inode и набор разрешений доступа что и у исходного файла;*

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное использование каталогов и файлов. Жесткие и символические ссылки на файлы и каталоги)

- **разрешения доступа ссылки изменятся при изменении разрешений файла;**

### 2.2.2 Символическая ссылка (soft link, symbolic link) на файл

Символическая или косвенная ссылка даёт возможность для доступа к файлу или каталогу вместо имени файла или каталога указывать имя файла-ссылки на данный файл или каталог.

Таким образом, символическая ссылка представляет собой псевдоним (текстовую подстановку).

Файл, на который указывает символическая ссылка, и сама ссылка представляют собой разные объекты файловой системы. Поэтому можно удалять файл, на который создана символическая ссылка, не удалив при этом ссылку. Можно создавать ссылки на ссылки и т.п.

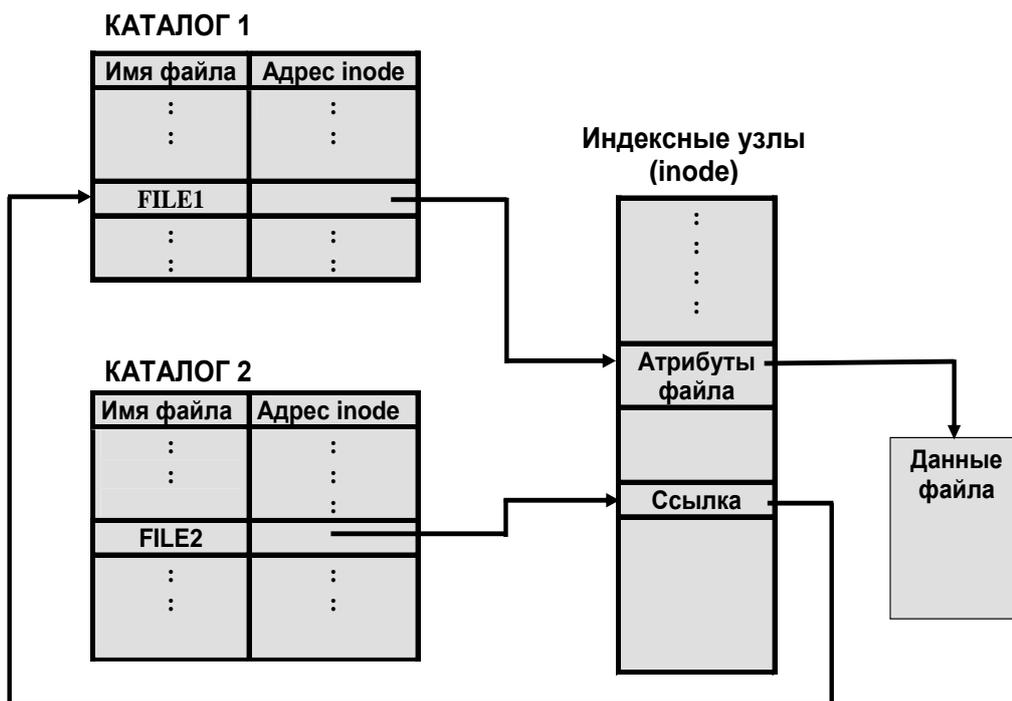


Рисунок 3. Символическая ссылка на атрибуты и данные файл

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное использование каталогов и файлов. Жесткие и символические ссылки на файлы и каталоги)

Когда устанавливается символическая ссылка **FILE2** из каталога **КАТАЛОГ2** на файл **FILE1** каталога **КАТАЛОГ1**, то в каталоге **КАТАЛОГ2** создается новый файл типа **link (связь)**. Индексный узел файла типа **link** содержит путь к файлу, с которым установлена символическая связь (если путь к файлу большой длины, и индексного узла недостаточно для записи этого пути, то выделяются дополнительные блоки хранения данных)

Когда происходит обращение пользователя к файлу **FILE2** в каталоге **КАТАЛОГ2**, то операционная система определяет, что это файл типа **link**, поэтому будет происходить переадресация к файлу, с которым связан этот файл **link**, т.е. обращение будет осуществляться к файлу **FILE1** в каталоге **КАТАЛОГ1**.

Основные особенности символических ссылок:

- *символические ссылки могут быть как на файлы, так и на каталоги;*
- *после удаления, перемещения или переименования файла ссылки становятся недействительными;*
- *права доступа к файлу и номер *inode* отличаются от исходного файла;*
- *при изменении прав доступа для исходного файла, права на ссылку останутся неизменными;*
- *можно ссылаться на другие файловые системы;*

При символической ссылке не используются биты прав доступа (они всегда отображаются, как *gwxgwxgwx*). Вместо этого, права доступа к файлу, полученному символической ссылкой, определяются правами доступа к файлу, на который он ссылается

### **2.2.3 Недостатки жёстких и символических ссылок.**

В методе жёстких ссылок, при создании файла в *i*-узле владельцем файла будет числиться пользователь, создавший файл. Создание новой жёсткой ссылки не изменяет владельца файла, а только увеличивает счетчик жёстких ссылок в *i*-узле, что позволяет системе отслеживать количество записей в каталогах, ссылающихся на этот файл.

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное использование каталогов и файлов. Жесткие и символические ссылки на файлы и каталоги)

Если система ведет учет или выделяет ограниченные квоты на использование дискового пространства, то пользователь, создавший файл, будет продолжать получать счета за этот файл до тех пор, пока удалятся все ссылки на данный файл.

При символическом связывании такой проблемы не возникает, так как указатель на *i*-узел есть только у владельца файла. У остальных пользователей, установивших связь с этим файлом, есть только пути к файлу, а не указатели на *i*-узел. Когда владелец удаляет файл, файл уничтожается, то последующие попытки использовать этот файл при помощи символической связи не будут иметь успеха, так как системе не сможет найти файл. Удаление символической связи никоим образом не повлияет на файл.

Недостатком символической связи являются накладные расходы. Чтобы получить доступ к *i*-узлу, следует сначала прочитать файл, содержащий путь. Затем следует пройти по этому пути, открывая каталог за каталогом, пока, наконец, не будет получен нужный *i*-узел. Все эти действия могут потребовать существенного количества обращений к внешним устройствам.

Кроме того, для каждой символической связи требуется дополнительный *i*-узел, а также дополнительный блок данных для хранения пути к файлу, если имя пути длинное.

Преимущество символических связей состоит в том, что они могут использоваться для ссылок на файлы, расположенные на удаленных компьютерах. Для этого, кроме обычного пути файла, нужно всего лишь указать сетевой адрес машины, на которой файл располагается.

### **2.3 Создание жёстких и символических ссылок в командной строке**

Для создания как жёстких, так и символических ссылок используется утилита командной строки **ln**

Примеры использования утилиты **ln** приведены в приложении

### 3 Выполнение работы

#### 3.1 Задание.

1. Создать в каталоге **home** файловой системы каталог **СТЕНА**, в котором будут находиться каталоги и файлы для совместного использования пользователями системы.
2. Создать в каталоге **СТЕНА** файл **лента.txt**. Пользователи системы должны иметь полный доступ к этому файлу, используя символические ссылки на данный файл в своих домашних каталогах.
3. Создать в каталоге **СТЕНА** каталог **РАСПИСАНИЕ**, содержащий файлы: **расписание\_первого\_курса.txt**, **расписание\_второго\_курса.txt**, **расписание\_третьего\_курса.txt**.

Пользователи **admin\_kaf** и **admin\_dek** должны иметь полный доступ к данным файлам, используя жёсткие ссылки.

Остальные пользователи могут читать расписания, используя символические ссылки.

#### 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 **Alt10**
4. Войти в систему под учётной записью **admin\_kaf/adminkaf**.
5. Создать необходимые каталоги и файлы в соответствии с заданием. Установить владельцев, группы и права доступа.
6. Создать жесткие и символические ссылки на файлы и каталоги в соответствии с заданием.
7. Проверить правильность выполнения задания
8. Ответить на контрольные вопросы

#### 4 Контрольные вопросы

1. Что такое индексный узел?
2. Какая информация о файле находится в каталоге и какая – в индексном узле?
3. В чём разница между жёсткими и символическими ссылками?

#### 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 Основные команды для работы с файлами и каталогами**

#### **6.1.1 Получение информации о каталогах и файлах**

**\$ pwd** - вывод на экран пути к текущему каталогу

**\$ ls** - вывод списка файлов и каталогов текущего каталога

**\$ ls -l** - вывод списка файлов и каталогов в форматированном виде

**\$ ls -la** - вывод списка файлов и каталогов в форматированном виде, в том числе начинающихся с точки

#### **6.1.2 Навигация по файловой системе**

**\$ cd** - переход в домашний каталог пользователя

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

**\$ cd ..** - переход в родительский каталог данного каталога

**\$ cd /** - переход в корневой каталог файловой системы

#### **6.1.3 Создание каталога**

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

**# chown <Владелец>: <Имя каталога или файла>** - изменение владельца каталога или файла

**# chgrp <Владелец>: <Имя каталога или файла>** - изменение группы каталога или файла

#### **6.1.4 Создание файла**

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

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

**\$ ls -li <имя файла>** - вывод информации о файле(включая информацию о индексном дескрипторе

**\$ 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 <параметры> <file1> <file2>**

**file1** - существующая жёсткая или символическая ссылка

**file2** - создаваемая жёсткая или символическая ссылка

Параметры утилиты:

- **-d** – разрешить создавать жесткие ссылки для директорий суперпользователю;
- **-f** – удалять существующие ссылки;
- **-i** – спрашивать нужно ли удалять существующие ссылки;
- **-P** – создать жесткую ссылку;
- **-r** – создать символическую ссылку с относительным путем к файлу;
- **-s** – создать символическую ссылку.

**# ln file1 hard\_file1** - создание жёсткой ссылки **hard\_file1** на файл **file1**

**# ln -s /home/file1 /home/soft\_file1** - создание символической ссылки **soft\_file1** на файл **file1**