

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 материалы к лабораторным работам

по дисциплине

«Операционные системы»

(кафедра СГНЗ)

Лабораторная работа № 7

«ОС Alt Linux. Файловые системы.

Совместное использование каталогов и файлов»

Москва

2024 г.

ОГЛАВЛЕНИЕ

1	ЦЕЛЬ РАБОТЫ.....	3
2	ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	3
2.1	Структуры данных файлов и каталогов	3
2.2	Совместное использование файлов	4
2.2.1	Жесткие ссылки (hard link).....	4
2.2.2	Символическая ссылка (soft link, symbolic link) на файл	6
2.2.3	Недостатки жёстких и символических ссылок.	7
2.3	Создание жёстких и символических ссылок в командной строке.....	9
3	ВЫПОЛНЕНИЕ РАБОТЫ	9
3.1	Задание.....	9
3.2	Порядок выполнения работы.....	9
3.2.1	Создать учётную запись пользователя <i>user_ws</i>	9
3.2.2	Проверить и при необходимости отредактировать файл <i>sudoers</i>	10
3.2.3	Войти в систему под учётной записью <i>user_ws</i>	10
3.2.4	Создать каталог совместного использования	10
3.2.5	Создать файл в каталоге совместного использования	10
3.2.6	Создать в домашнем каталоге пользователя <i>admin_ws</i> символическую ссылку на каталог <i>common</i> 10	
3.2.7	Создать файл в каталоге по ссылке <i>soft_common</i>	10
3.2.8	Просмотреть информацию о созданных каталогах и файлах	10
3.2.9	Создать в домашнем каталоге пользователя <i>user_ws</i> символическую ссылку на каталог <i>common</i> ..	10
3.2.10	Создать в домашнем каталоге пользователя <i>user_ws</i> жёсткую ссылку на файл в каталоге <i>common</i> 11	
3.2.11	Проверить правильность создания символических и жёстких ссылок	11
4	КОНТРОЛЬНЫЕ ВОПРОСЫ	11
5	ЛИТЕРАТУРА.....	11
6	ПРИЛОЖЕНИЕ.....	13
6.1	Основные команды для работы с файлами и каталогами	13
6.1.1	Получение информации о каталогах и файлах	13
6.1.2	Навигация по файловой системе	13
6.1.3	Создание каталога	13
6.1.4	Создание файла	13
6.2	Создание жёстких и символических ссылок	14
6.3	Редактирование файла <i>sudoers</i>	14

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

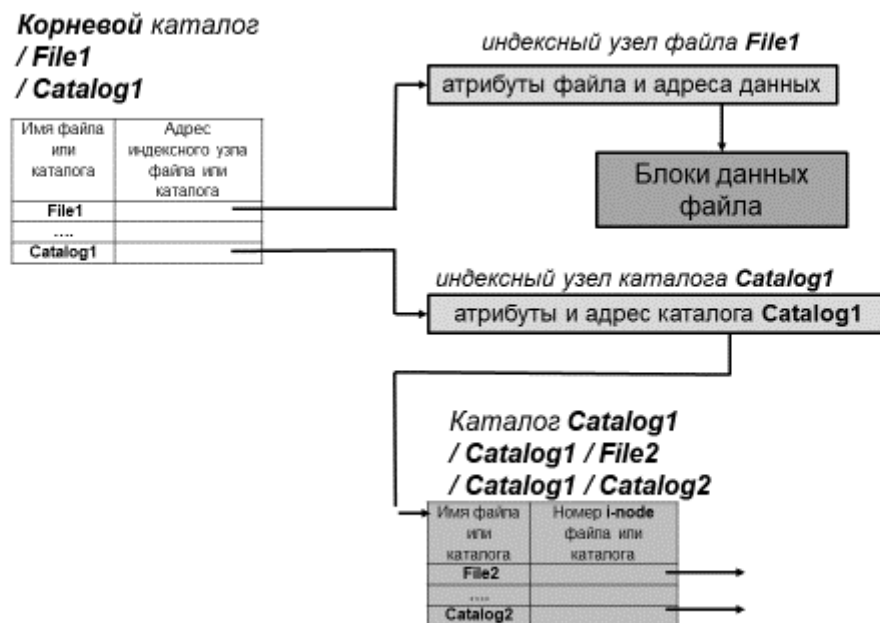
Основными объектами файловой системы ОС Alt Linux являются **файлы и каталоги**.

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

С каждым файлом связана структура данных, называемая **i-узлом (index node** – индексный узел), содержащей 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.

В файловой системе ОС Alt Linux принята иерархическая организация данных (дерево каталогов). Каждому файлу соответствует запись в каталоге. Эта запись содержит **имя файла и номер i-узла**.



1

adminws

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

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное ис-5 пользование каталогов и файлов)

первоначально являлось оригиналом (Любой файл всегда имеет как минимум одну жесткую ссылку – его имя, под которым он был создан.).

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

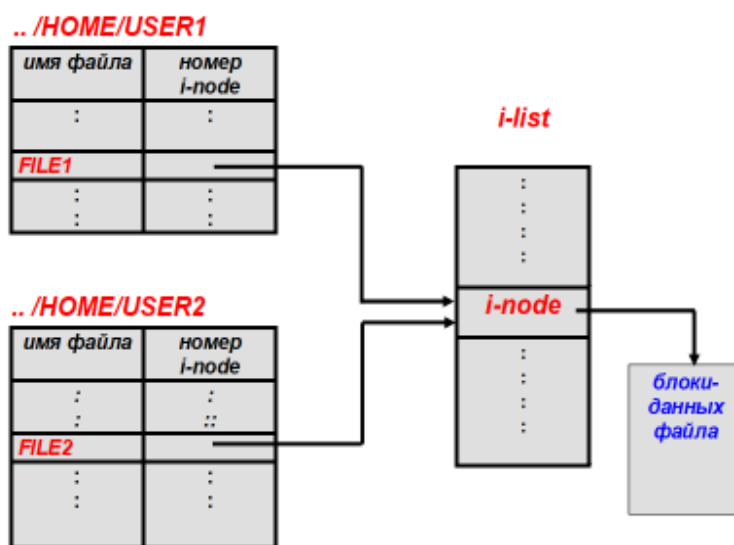


Рисунок 2. Жёсткие ссылки на атрибуты и данные файла

Таким образом, особенности 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 i-node и набор разрешений что и у исходного файла;*
- *разрешения на ссылку изменятся при изменении разрешений файла;*

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

Символическая или косвенная ссылка обеспечивает возможность вместо имени файла (с путем) или каталога указывать имя файла-ссылки; т.е. символическая ссылка представляет собой псевдоним (*текстовую подстановку*) для имени. Файл, на который указывает символическая ссылка, и сама ссылка представляют собой *разные* объекты файловой системы. Поэтому можно создавать ссылки на несуществующие файлы и каталоги, удалять оригинальные файлы, не удалив при этом ссылку. Можно создавать ссылки на ссылки и т.п.

Когда пользователь А устанавливает связь с одним из файлов пользователя В, система создает в каталоге пользователя А новый файл типа LINK (связь). Новый файл содержит просто имя пути к файлу, с которым он связан. Когда пользователь А читает данные из связанного файла, операционная система определяет, что обращение производится к файлу типа LINK, поэтому она открывает файл, с которым связан этот файл, и читает данные из него. Такой метод называется символическим связыванием.

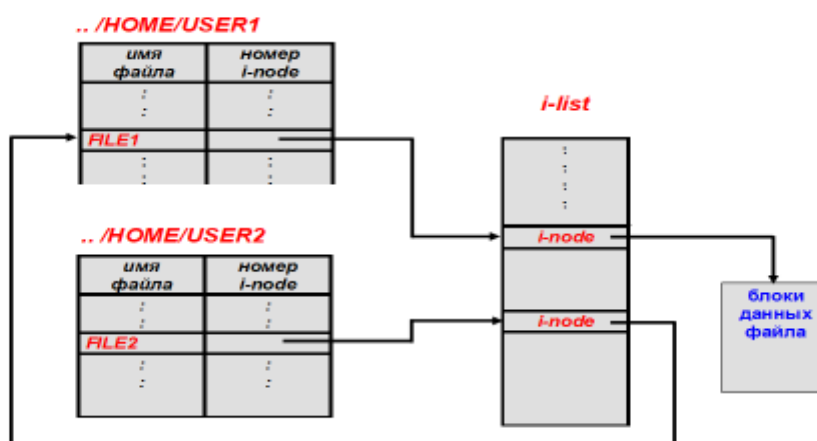


Рисунок 3. Символическая ссылка на атрибуты и данные файл

Основные особенности символических ссылок:

- *символические ссылки могут быть как на файлы, так и на каталоги;*

- *после удаления, перемещения или переименования файла становятся недействительными;*
- *права доступа и номер inode отличаются от исходного файла;*
- *при изменении прав доступа для исходного файла, права на ссылку останутся неизменными;*
- *можно ссылаться на другие разделы диска;*

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, когда пользователь В устанавливает связь с совместно используемым файлом, то в *i*-узле владельцем файла числится пользователь А, создавший файл. Создание связи не изменяет владельца файла, а только увеличивает счетчик *i*-узла, что позволяет системе отслеживать количество записей в каталогах, ссылающихся на этот файл. Если впоследствии пользователь А попытается удалить этот файл, операционная система столкнется с проблемой. Если она удалит файл и очистит *i*-узел, то у пользователя В окажется запись в каталоге, указывающая на неверный *i*-узел. Если этот *i*-узел впоследствии будет назначен другому файлу, связь в каталоге пользователя В будет ссылаться на неверный файл. По значению счетчика *i*-узла система сможет определить, что этот файл используется кем-то еще, но у нее нет способа найти все записи в каталогах, указывающие на этот файл, чтобы удалить их. Указатели на каталоги не могут храниться в *i*-узлах, так как число таких каталогов ничем не ограничивается.

Единственное, что может сделать операционная система - это удалить запись в каталоге пользователя А, но оставить на месте *i*-узел, уменьшив значение счетчика на 1. Таким образом, мы теперь получили ситуацию, в которой пользователь В является единственным пользователем, имеющим ссылку на файл, вла-

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное ис-8
пользование каталогов и файлов)

дельцем которого считается пользователь А. Если система ведет учет или выделяет ограниченные квоты на использование дискового пространства, пользователь А будет продолжать получать счета за этот файл до тех пор, пока пользователь В не решит, наконец, удалить его. В этом случае счетчик *i*-узла уменьшится до нуля и система удалит 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. 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 является необходимость накладных расходов. 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 диску.

Кроме того, для каждой символической связи требуется дополнительный *i*-узел, а также дополнительный блок диска для хранения пути к файлу, хотя, если имя пути короткое, то в качестве оптимизации система может хранить его в самом *i*-узле. 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.

Использование связей создает еще одну проблему. При использовании связей у одного файла оказывается несколько путей. Программы, сканирующие каталоги со всеми подкаталогами, могут наткнуться на один и тот же файл несколько раз. Если такая программа, например, записывает все файлы на магнитную ленту, она запишет на ленту несколько копий одного и того же файла. Более

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное ис-⁹ пользование каталогов и файлов)

того, если запись с этой ленты будет потом прочитана на другом компьютере, этот файл окажется скопированным на диск несколько раз, если только считывающая ленту программа не догадается, что это один и тот же файл.

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 каталог для совместного использования различными пользователями системы.
2. В домашних каталогах пользователей создать символические ссылки на каталог совместного пользования.
3. Создать файлы в каталоге совместного использования.
4. Создать в домашних каталогах пользователей жесткие и символические ссылки на совместно используемые файлы.

3.2 Порядок выполнения работы.

1. Войти в систему под учётной записью **user2**.
2. Запустить программу виртуализации **Oracle VM VirtualBox**
3. Запустить виртуальную машину **Alt10-XX**.
4. Войти в систему под учётной записью **root/adminroot**.

3.2.1 Создать учётную запись пользователя *user_ws*

1. Запустить программу **Терминал**.
2. Перейти на виртуальную консоль **tty2 (Ctrl+Alt+F2)**
3. Войти под учётной записью **root/adminroot**
4. Создать учётную запись пользователя **user_ws**, задать пароль (**userws**) и включить пользователя в группу **wheel**

3.2.2 Проверить и при необходимости отредактировать файл `sudoers`

Разрешить пользователям группы **wheel** выполнение всех команд с помощью утилиты **sudo**.

3.2.3 Войти в систему под учётной записью `user_ws`

1. Перейти на виртуальную консоль **tty3**
2. Войти в систему под учётной записью **user_ws/userws**

3.2.4 Создать каталог совместного использования

1. Перейти на виртуальную консоль **tty2** (пользователь **root**)
2. Создать каталог **/home/common**
3. Установить права доступа к каталогу **2777**

3.2.5 Создать файл в каталоге совместного использования

Используя команду **touch** создать в каталоге **common** текстовый файл **test_root.txt**.

3.2.6 Создать в домашнем каталоге пользователя `admin_ws` символическую ссылку на каталог `common`

1. Перейти на виртуальную консоль **tty1** (пользователь **admin_ws**)
2. Создать в домашнем каталоге пользователя символическую ссылку **soft_common** на каталог **/home/common**

3.2.7 Создать файл в каталоге по ссылке `soft_common`

Используя команду **touch** создать по ссылке **soft_common** текстовый файл **test_admin.txt**.

3.2.8 Просмотреть информацию о созданных каталогах и файлах

Используя команду **ln** просмотреть информацию и права доступа к каталогу **common**, ссылке **soft_common** и файлу **test_admin.txt**

3.2.9 Создать в домашнем каталоге пользователя `user_ws` символическую ссылку на каталог `common`

1. Перейти на виртуальную консоль **tty3** (пользователь **user_ws**)

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное ис-11 пользование каталогов и файлов)

2. Создать в домашнем каталоге пользователя символическую ссылку **soft_common** на каталог **/home/common**

3.2.10 Создать в домашнем каталоге пользователя *user_ws* жёсткую ссылку на файл в каталоге *common*

1. Перейти на виртуальную консоль **tty3** (пользователь **user_ws**)

2. Создать в домашнем каталоге пользователя жёсткую ссылку **hard_test_root.txt** на файл **/home/common/test_root.txt**

3.2.11 Проверить правильность создания символических и жёстких ссылок

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

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное ис-12
пользование каталогов и файлов)

Linux. Управление пользователями. Администрирование дисковой подсистемы: Учебно-методическое пособие. –М.: Издательство «Спутник+», 2023. -78 с.

4. 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

Операционные системы Лаб. работа №7 (ОС Alt Linux. Файловые системы. Совместное ис-14
пользование каталогов и файлов)

\$ 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**

6.3 Редактирование файла *sudoers*

Разрешить пользователям группы **wheel** выполнение всех команд с помощью утилиты **sudo**.

EDITOR=nano visudo

убрать символ **#** в строке **WHEEL_USERS ALL=(ALL) ALL**

Просмотр правил **sudo** для данного пользователя

sudo -l