## **РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

### **Факультет физико-математических и естественных наук**

### **Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6**

дисциплина: Операционные системы

**Студент:**

Афтаева Ксения Васильевна

**Преподаватель:**

Велиева Т.В.

**Группа:** НПИбд-01-20

**МОСКВА** 2021 г.

### Цель работы:

Ознакомление с файловой системой Linux, ее структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

### Задачи:

Изучить теоретический материал, выполнить действия, описанные в задании, ответить на контрольные вопросы.

### Объект и предмет исследования:

ОС Linux, командная строка, файловая система.

### Техническое оснащение:

Ноутбук, на котором установлена виртуальная машина с линукс

### Теоретические вводные данные[[1]](https://pingvinus.ru/note/command-line-fundamentals)

**Файл** - именновагная область данных на носителе информации.

**Каталог** - поименнованная совокупность файлов и подкаталогов.

**Файловая система** в Linux определяет также организацию расположения файлов, по сути представляя собой иерархическую структуру **«дерева»**: начинается с корневого каталога «/» и разрастается ветвями в зависимости от работы системы.

### Условные обозначения:

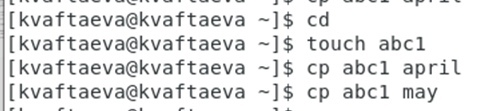
Права доступа:

* r - разрешены просмотр и копирование файлов, просмотр входящих в каталог файлов.
* w - разрешены изменение и переименование файлов, создание и удаление файлов в каталогах.
* x - разрешено выполнение файлов, доступ в каталог.

### Выполнение работы:

1. Выполнила все примеры, приведенные в первой части описания лабораторной работы:

**Копирование (команда cp):** - Скопировала файл **~/abc1** в файл **april** и в файл **may** (Рис.1).



*Рис.1 Копирование файла в текущем каталоге*

Видим, что копирование прошло успешно.

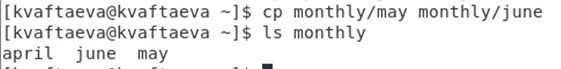
* Скопировала файлы **april** и **may** в каталог **monthly** (Рис.2).



*Рис.2 Копирование нескольких файлов в каталог*

Создать каталог не удалось, так он уже существовал. поэтому я просто перемещаю в него нужные файлы.

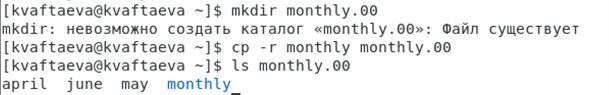
* Скопировала файл **monthly/may** в файл с именем **july** (Рис.3).



*Рис.3 Копирование файлов в произвольном каталоге*

Выводим содержимое каталога и видим, что копирование с нужным именем произошло успешно.

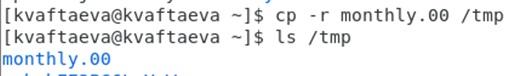
* Скопировала каталог **monthly** в каталог **monthly.00** (Рис.4).



*Рис.4 Копирование каталогов в текущем каталоге*

Каталог уже был на моем компьютере. Посмотрев содержимое каталога видим, что копирование прошло успешно и подкаталог **monthly** присутствует.

* Скопировала каталог **monthly.00** в каталог \*\*/tmp (Рис.5).



*Рис.5 Копирование каталогов в произвольном каталоге*

Смотрим содержимое каталога **/tmp**. В нем есть подкаталог **monthly.00**, значит, копирование прошло успешно.

**Перемещение и переименование (команды mv и mvdir):**

* Изменила название файла **april** на **july** в домашем каталоге (Рис.6).



*Рис.6 Переименование файла в текущем каталоге*

Смотрим содержимое домашнего каталога. Видим, что файла с названием **april** больше нет, так как он теперь называется **july**.

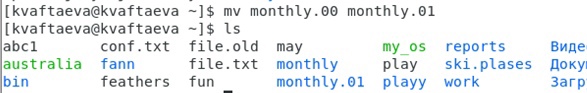
* Переместила файл july в каталог monthly.00 (Рис.7).



*Рис.7 Перемещение файлов в другой каталог*

Выводим содержимое каталога **monthly.00** и видим, что файл был успешно перенесен.

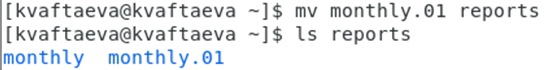
* Переименовала каталог **monthly.00** в **monthly.01** (Рис.8).



*Рис.8 Переименование каталогов в текущем каталоге*

Вывела содержимое. Видим, что каталог **monthly.00** был переименован в **monthly.01**.

* Переместила каталог **monthly.01** в каталог **reports** (Рис.9).



*Рис.9 Перемещение каталога в другой каталог*

Посмотрев содержимое, видим, что перемещение выполнено.

* Переименовала каталог **reports/monthly.01** в **reports/mounthly** (Рис.10).

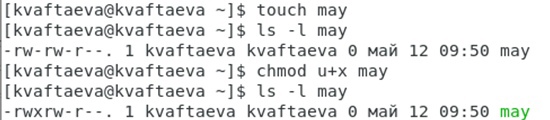


*Рис.10 Переименование каталога не являющегося текущим*

Выводим содержимое, видим, что переименовали каталог успешно.

**Изменение прав доступа:**

* Создала файл **~/may** с правом выполнения для владельца (Рис.11).



*Рис.11 Создание файла с правом выполнения для владельца*

Выводим командой ls -l подробную информацию до и после команды с изменением прав доступа. Видим, что добавили право выполнения для владельца.

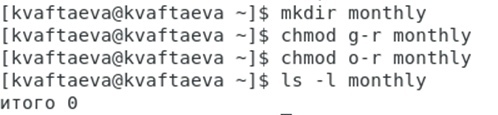
* Лишила владельца файла **~/may** права на выполнение (Рис.12).



*Рис.12 Лишение права на выполнение*

Выводим командой ls -l подробную информацию после команды с изменением прав доступа. Видим, что убрали право выполнения для владельца.

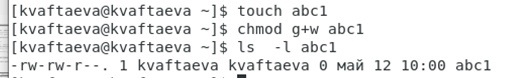
* Создала каталог **monthly** с запретом на чтение для членов группы всех остальных пользователей (Рис.13)



*Рис.13 Каталог с запретом на чтение*

Видим, что все выполнено верно.

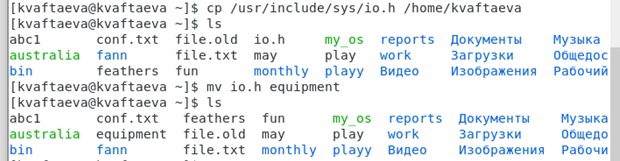
* Создала файл **~/abc1** с правом записи для членов группы (Рис.14).



*Рис.14 Файл с правом записи для членов группы*

Выводим подробную информацию о файле, видим, что все выполнено.

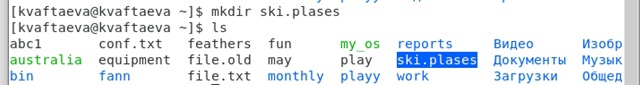
1. Скопировала файл **/usr/include/sys/io.h** в домашний каталог командой cp /usr/include/sys/io.h /home/user. Затем, с помощью команды mv io.h equipment назвала его согласно требованиям (Рис.15).



*Рис.15 Копирование файла в домашний каталог*

Вывожу содержимое домашнего каталога дважды. в первый раз мы видим, что файл был успешно скопирован. Во второй раз файл есть уже под другим названием. Значит, переименование прошло успешно.

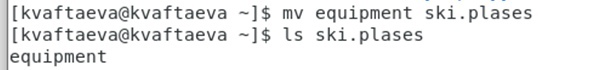
1. В домашнем каталоге создала директорию **~/ski.plases** с помощью команды mkdir (Рис.16).



*Рис.16 Создание директории*

Выводим содержимое домашнего каталога и видим, что директория была успешно создана.

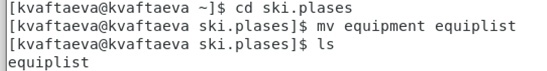
1. Переместила файл **equipment** в каталог **~/ski.plases** командой mv (Рис.17).



*Рис.17 Перемещение файла в каталог*

Вывожу содержимое каталога **ski.plases**, файл был успешно перемещен.

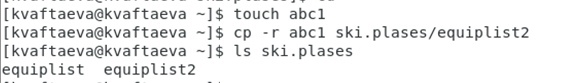
1. Переименовала файл **~/ski.plases/equipment** в **~/ski.plases/equiplist** командой mv (Рис.18).



*Рис.18 Переименование файла*

Проверяю содержимое каталога. Можем увидеть, что переименование произошло успешно.

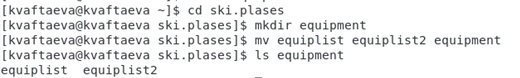
1. Создала в домашнем каталоге файл **abc1** командой touch. Скопировала его в каталог **~/ski.plases**, назвав **equiplist2**, командой cp -r (Рис.19).



*Рис.19 Копирование и переименование файла abc1*

Вывожу содержимое. Можем увидеть, чьл файл был скопирован с нужным именем.

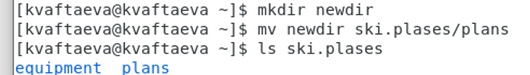
1. Создала в каталоге **~/ski.plases** каталог с именем **equipment** командой mkdir и переместила файлы **equiplist** и **equiplist2** в этот каталог командой mv (Рис.20).



*Рис.20 Перемешение файлов equiplist, equiplist2*

Вывела содержимое каталога. Видим, что перемещение прошло успешно.

1. Создала каталог **newdir** в домашнем каталоге командой mkdir. Переместила его в каталог **ski.plases** командой mv, назвав **plans** (Рис.21).



*Рис.21 Создание и перемещение каталога newdir*

Вывожу содержимое. Видим, что каталог был перемещен с нужным именем.

1. Определила опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа. Дала требуемые права для кадого файла (Рис.22).



*Рис.22 Права доступа*

Для каждого файла вывожу информацию о правах доступа. Видим, что все нужные права есть (есть и другие права, но убирать их по заданию не требовалось).

1. У меня в каталоге **/etc** нет файла **password**. Поэтому я вывела содержимое файла **passwd** из того же каталога командой ls (Рис.23).



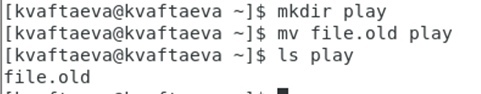
*Рис.23 Содержимое файла*

1. Скопировала файл **~/feathers** в файл **~/file.old** командой cp (Рис.24).



*Рис.24 Копирование файла в файл*

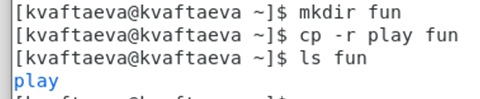
1. Скопировала файл **~/file.old** в каталог **~/play** командой mv (Рис.25).



*Рис.25 Копирование файла в каталог*

Открываем содержимое каталога **play**, видим, что файл был скопирован.

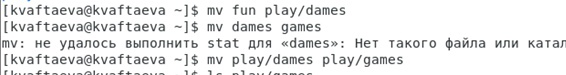
1. Скопировала каталог **~/play** в каталог **~/fun** командой cp -r (Рис.26).



*Рис.26 Копирование каталога в каталог*

Видим, что копирование прошло успешно.

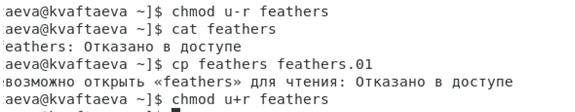
1. Переместила каталог **~/fun** в каталог **~/play** командой mv и назвала его **games** (Рис.27).



*Рис.27 Перемещение каталога в каталог*

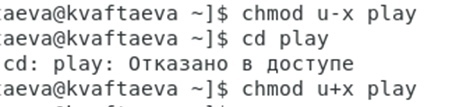
Открыв содержимое каталога, видим, что перемещение выполнено.

1. Лишила владельца файла **~/feathers** права на чтение командой chmod u-r. Попробовала посмотреть содержимое файла командой cat. Видим, что сделать этого нельзя. Аналогично нельзя скопировать этот файл. Вернула владельцу право на чтение командой chmod u+r (Рис.28).



*Рис.28 Право на чтение для файла feathers*

1. Лишила владельца каталога **~/play** права на выполнение командой chmod u-x. Видим, что теперь в него нельзя перейти. Дала владельцу права на выполнение этого каталога (Рис.29).



*Рис.29 Право на выполнение каталога play*

1. Прочитала man по командам mount, fsck, mkfs, kill. Их характеристика:

**Mount** – это команда для монтирования в линукс вручную.

Опции:

* t - задать файловую систему, которая будет использована
* o - задать различные параметры монтирования, например, монтировать только для чтения и т.д.
* V - вывести версию утилиты;
* h - вывести справку;
* v - подробный режим;
* a - примонтировать все устройства, описанные в fstab;
* F - создавать отдельный экземпляр mount для каждого отдельного раздела;
* f - не выполнять никаких действий, а только посмотреть что собирается делать утилита;
* n - не записывать данные о монтировании в /etc/mtab;
* l - добавить метку диска к точке монтирования;
* c - использовать только абсолютные пути;
* r - монтировать раздел только для чтения;
* w - монтировать для чтения и записи;
* L - монтировать раздел по метке;
* U - монтировать раздел по UUID;
* T - использовать альтернативный fstab;
* B - монтировать локальную папку;
* R - перемонтировать локальную папку.

**fsck** – проверка файловой системы linux и исправление найденных ошибок

Опции: -l - не выполнять другой экземпляр fsck для этого жесткого диска, пока текущий не завершит работу. Для SSD параметр игнорируется; - t - задать типы файловых систем, которые нужно проверить. - A - проверить все файловые системы из /etc/fstab. - C - показать прогресс проверки файловой системы; - M - не проверять, если файловая система смонтирована; - N - ничего не выполнять, показать, что проверка завершена успешно; - R - не проверять корневую файловую систему; - T - не показывать информацию об утилите; - V - подробный вывод. - a - во время проверки исправить все обнаруженные ошибки, без каких-либо вопросов. - n - выполнить только проверку файловой системы, ничего не исправлять; - r - спрашивать перед исправлением каждой ошибки, используется по умолчанию для файловых систем ext; - y - отвечает на все вопросы об исправлении ошибок утвердительно; - c - найти и занести в черный список все битые блоки на жестком диске. Доступно только для ext3 и ext4; - f - принудительная проверка файловой системы, даже если по журналу она чистая; - b - задать адрес суперблока, если основной был поврежден; - p - выполняет проверку и исправление автоматически. Для этой цели можно использовать одну из трех опций: p, a, y.

**mkfs** – позволяет создать файловую систему Linux.

Опции: - t - Указывает тип создаваемой файловой системы. - с - проверить устройство на наличие битых секторов - b - размер блока файловой системы - j - использовать журналирование для ext3 - L - задать метку раздела - v - показать подробную информацию о процессе работы - V - версия программы

**kill** –для прерывания процесса.

Опции: - l - Выдать все значения сигналов, поддерживаемые в данной реализации, если операнды не указаны. - s - задает сигнал, который надо послать, используя одно из символьных имен, заданных в заголовочном файле .

### Контрольные вопросы: [[2]](https://losst.ru/tipy-fajlovyh-sistem-dlya-linux)

1. **devtmpfs**- она была разработана для решения проблемы с доступностью устройств во время загрузки. Ядро создает файлы устройств по мере надобности, а также уведомляет менеджер **udevd** о том, что доступно новое устройство. После получения такого сигнала менеджер **udevd** не создает файлы устройств, а выполняет инициализацию устройства и отправляет уведомление процессу. Кроме того, он создает несколько символических ссылок в каталоге **/dev** для дальнейшей идентификации устройств.

**Tmpfs** — временное файловое хранилище в Unix . Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо физического диска. Подобная конструкция является RAM диском. Все данные в **Tmpfs** являются временными, в том смысле, что ни одного файла не будет создано на жёстком диске. После перезагрузки все данные, содержащиеся в **Tmpfs**, будут утеряны.

**Fourth extended file system** сокр. ext4, или ext4fs — журналируемая ФС, используемая в ОС с ядром Linux. Основана на ФС ext3, ранее использовавшейся по умолчанию во многих дистрибутивах **GNU/Linux**.

**AFS** - Распределенная сетевая файловая система (Andrew File System) внедрена с целью создания единого файлового пространства пользователей при работе на различных фермах или рабочих станциях ОИЯИ, где домашний каталог пользователя определен как каталог в **-AFS**.

1. Все каталоги можно разделить на две группы: для статической (редко меняющейся) информации – **/bin, /usr** и динамической (часто меняющейся) информации – **/var, /tmp**. Исходя из этого администраторы могут разместить каждый из этих каталогов на собственном носителе, обладающем соответствующими характеристиками.

* **Корневой каталог**. Корневой каталог / является основой любой ФС UNIX. Все остальные каталоги и файлы располагаются в рамках структуры (дерева), порождённой корневым каталогом, независимо от их физического местонахождения.
* **/bin**. В этом каталоге находятся часто употребляемые команды и утилиты системы общего пользования. Сюда входят все базовые команды, доступные даже если была примонтирована только корневая файловая система. Примерами таких команд являются:Ls,cp и т.д.
* **/boot**. Директория содержит всё необходимое для процесса загрузки операционной системы: программу-загрузчик, образ ядра операционной системы и т.п..
* **/dev**. Каталог содержит специальные файлы устройств, являющиеся интерфейсом доступа к периферийным устройствам. Наличие такого каталога не означает, что специальные файлы устройств нельзя создавать в другом месте, просто достаточно удобно иметь один каталог для всех файлов такого типа.
* **/etc**. В этом каталоге находятся системные конфигурационные файлы. В качестве примеров можно привести файлы /etc/fstab, содержащий список монтируемых файловых систем, и /etc/resolv.conf, который задаёт правила составления локальных DNS-запросов. Среди наиболее важных файлов – скрипты инифиализации и деинициализации системы. В системах, наследующих особенности UNIX System V, для них отведены каталоги с /etc/rc0.d по /etc/rc6.d и общий для всех файл описания – /etc/inittab.
* **/home** (необязательно). Директория содержит домашние директории пользователей. Её существование в корневом каталоге не обязательно и её содержимое зависит от особенностей конкретной UNIX-подобной операционной системы.
* **/lib**. Каталог для статических и динамических библиотек, необходимых для запуска программ, находящихся-в директориях/bin,/sbin.
* **/mnt**. Стандартный каталог для временного монтирования файловых систем – например, гибких и флэш-дисков, компакт-дисков и т. п.
* **/root** (необязательно). Директория содержит домашюю директорию суперпользователя. Её существование в корневом каталоге не обязательно.
* **/sbin**. В этом каталоге находятся команды и утилиты для системного администратора. Примерами таких команд являются: route, halt, init и др. Для аналогичных целей применяются директории /usr/sbin и /usr/local/sbin.
* **/usr**. Эта директория повторяет структуру корневой директории – содержит каталоги /usr/bin, /usr/lib, /usr/sbin, служащие для аналогичных целей. Каталог /usr/include содержит заголовочные файлы языка C для всевозможные библиотек, расположенных в системе.
* **/usr/local** является следующим уровнем повторения корневого каталога и служит для хранения программ, установленных администратором в дополнение к стандартной поставке операционной системы.
* **/usr/share** хранит неизменяющиеся данные для установленных программ. Особый интерес представляет каталог /usr/share/doc, в который добавляется документация ко всем установленным программам.
* **/var, /tmp**. Используются для хранения временных данных процессов – системных и пользовательских соответственно.

1. Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой mount без параметров.
2. Причины:

* Из-за прерывания операций ввода-вывода выполняемых непосредственно с диском;
* Сбоя питания;
* Краха ОС;
* Нарушения работы дискового КЭШа;

Устранение поврежденных файлов: в большинстве случаев, проверка файловой системы способна обнаружить и выполнить ремонт такой ошибки автоматически, и после завершения процесс начальной загрузки продолжится как обычно. Если проблема файловой системы более серьезна, проверка файловой системы не может решить проблему автоматически. В этом случае процесс надо будет запустить вручную.

1. Обычно при установке Linux создание файловых систем - компетенция инсталлятора, который осуществляет его с некоторыми опциями по умолчанию. Изменить характеристики, определенные для файловой системы при ее создании, невозможно без повторного выполнения этого процесса.

Файловая система Ext2fs может быть создана любой из следующих команд - /sbin/mke2fs, /sbin/mkfs, /sbin/mkfs.ext2 с указанием файла устройства в качестве аргумента.

Для создания XFS -mkfs.xfs (из пакета xfsprogs).

Для создания файловой системы ext3fs -mke2fs с опцией j.

Файловая система ReiserFS - /sbin/mkreiserfs из пакета reiserfsprogs.

* Для просмотра небольших файлов –cat (cat имя-файла)
* Для просмотра больших файлов-less (less имя-файла)
* Для просмотра начала файла-head. По умолчанию она выводит первые 10 строк файла. (head [-n] имя-файла), n — количество выводимых строк
* Команда tail. Выводит несколько (по умолчанию 10) последних строк файла. (tail [-n] имя-файла), n — количество выводимых строк.

1. При помощи команды cp осуществляется копирование файлов и каталогов (cp[-опции] исходный\_файл целевой\_файл)

Возможности команды ср:

* копирование файла в текущем каталоге
* копирование нескольких файлов в каталог
* копирование файлов в произвольном каталоге
* опция i в команде cp поможет избежать уничтожения информации в случае, если на место целевого файла вы поставите имя уже существующего файла(т.е. система попросит подтвердить, что вы хотите перезаписать этот файл)
* команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами.

1. Команды mv и mvdir предназначены для перемещения и переименования файлов и каталогов. (mv [-опции] старый\_файл новый\_файл)

Для получения предупреждения перед переписыванием файла стоит использовать опцию i.

1. **Права доступа** - совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации.

Для изменения прав доступа к файлу или каталогу используется команда - chmod. ( chmod режим имя\_файла)

Права доступа к файлу может поменять только владелец и администратор.

Режим (в формате команды) имеет следующую структуру и способ записи:

* "=" установить право
* "-" лишить права
* "+" дать право
* "r" чтение
* "w" запись
* "x" выполнение
* "u" (user) владелец файла
* "g" (group) группа, к которой принадлежит владелец файла
* "o" (others) все остальные.

### Заключение:

Таким образом, все пункты задания были выполнены, навыки работы с файлами и каталогами получены.

### Вывод:

Я ознакомилась с файловой системой Linux, ее структурой, именами и содержанием каталогов. Приобрела практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

### Библиографический список:

[1]: [Основы работы в командной строке линукс](https://pingvinus.ru/note/command-line-fundamentals)

[2]: [Файловые системы линукс](https://losst.ru/tipy-fajlovyh-sistem-dlya-linux)