# 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: Соболевский Денис Андреевич

Группа: НФИбд-02-20

**МОСКВА**

2021 г.

**Цель работы:**

Ознакомление с файловой системой Linux, её 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.

**Выполнение работы:**

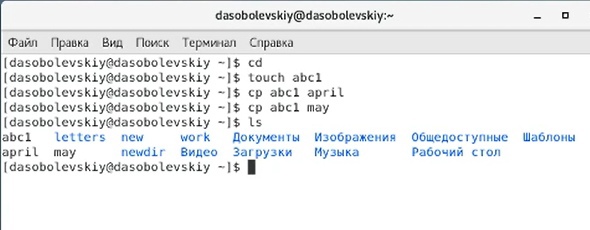
1. Выполняем все примеры, приведенные в материалах лабораторной работы №6.

• Копирование файлов и каталогов:

Примеры:

1. Копирование файла в текущем каталоге. Скопируем файл ~/abc1 в файл april и в файл may, предварительно создаем файл abc1 командой touch (рисунок 1). Само копирование осуществляется командой cp.

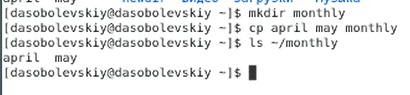
*Рисунок 1:*



Проверяем правильность действий командой ls - просматриваем содержимое домашнего каталога. Видим, что все новые файлы на месте.

1. Копирование нескольких файлов в каталог. Скопируем файлы april и may в каталог monthly командой cp (рисунок 2) (предварительно создали каталогn mothly командой mkdir):

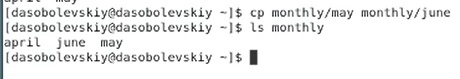
*Рисунок 2:*



Командой ls проверяем содержимое каталога mothly. Видим, что файлы april и may скопировались в наш каталог.

1. Копирование файлов в произвольном каталоге. Скопируем файл monthly/may в файл с именем june командой cp и проверим верность наших действий, просмотрев каталог monthly командой ls (рисунок 3):

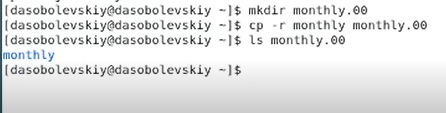
*Рисунок 3:*



Видим, что скопированный файл may отображается в каталоге как june, как и задумывалось.

1. Копирование каталогов в текущем каталоге. Скопируем каталог monthly в каталог monthly.00. Для этого воспользуемся командой cp -r, которая позволяет копировать каталоги вместе с входящими в них файлами и каталогами (рисунок 4), но санчала создадим каталог monthly.00 командой mkdir:

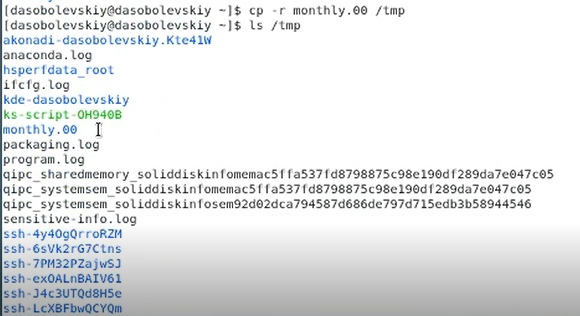
*Рисунок 4:*



Проверим верность действий командой ls - просмотрим содержимое каталога monthly.00. Видим, что каталог monthly был успешно скопирован в каталог monthly.00.

1. Копирование каталогов в произвольном каталоге. Скопируем каталог monthly.00 в каталог /tmp все той же командой cp -r (рисунок 5):

*Рисунок 5:*

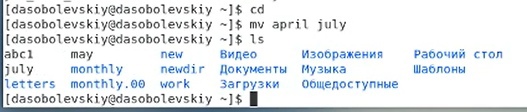


Просмотрим содержимое каталога /tmp (команда ls). Видим, что каталог monthly.00 теперь действительно находится в нем (он подсвечен синим).

• Перемещение и переименование файлов и каталогов:

1. Переименование файлов в текущем каталоге. Изменим название файла april на july в домашнем каталоге. Воспользуемся командой mv (рисунок 6):

*Рисунок 6:*



Просматриваем содержимое каатлога и видим, что файл был успешно переименован (выделено черным).

1. Перемещение файлов в другой каталог. Переместим файл july в каталог monthly.00, вновь воспользовавшись командой mv (рисунок 7):

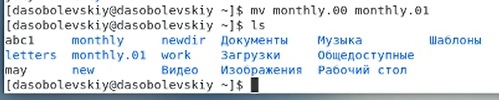
*Рисунок 7:*



Смотрим содержимое каталога monthly.00 и убеждаемся в том, что файл july был успешно перемещен в него.

1. Переименование каталогов в текущем каталоге. Переименуем каталог monthly.00 в monthly.01 командой mv (рисунок 8):

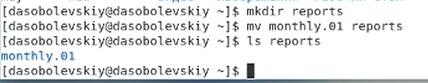
*Рисунок 8:*



Проверим содержимое домашнего каталога. Видим, что каталог monthly.00 был переименован в monthly.01 (подсвечено синим).

1. Перемещение каталога в другой каталог. Переместим каталог monthly.01 в каталог reports командой mv, предварительно создадим каталог reports командой mkdir (рисунок 9):

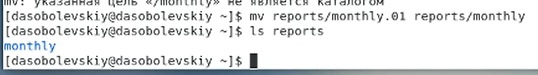
*Рисунок 9:*



Просматриваем каталог reports и видим, что каталог monthly.01 был успешно перемещен в него.

1. Переименование каталога, не являющегося текущим. Переименуем каталог reports/monthly.01 в reports/monthly (рисунок 10):

*Рисунок 10:*

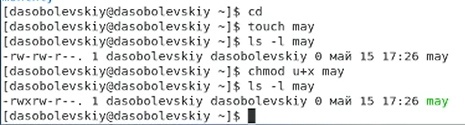


Просматриваем каталог reports и видим, что каталог monthly.01 был успешно переименован в monthly.

• Изменение прав доступа:

1. Создадим файл ~/may (команда touch) с правом выполнения для владельца (меняем права доступа командой chmod, используем комбинацию u+x) между делом проверяем свойства файла командой ls -l (рисунок 11):

*Рисунок 11:*



По выводам команды ls понимаем, что задание выполенено верно.

1. Лишим владельца файла ~/may права на выполнение комбинацией u-x и проверим правильность действий командой ls -l (рисунок 12):

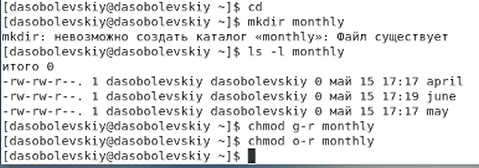
*Рисунок 12:*



По выводам команды ls понимаем, что задание выполенено верно.

1. Создадим каталог monthly (командой mkdir) с запретом на чтение для членов группы и всех остальных пользователей (g-r, o-r) (рисунок 13):

*Рисунок 13:*



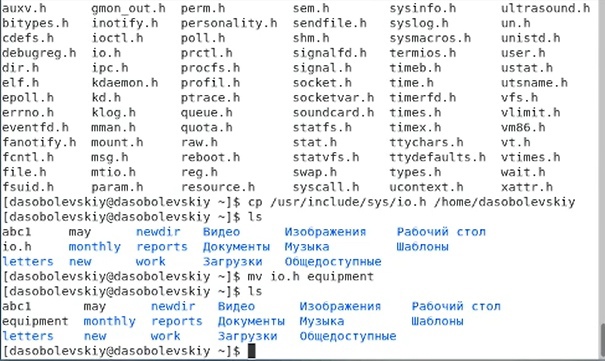
1. Создадим файл ~/abc1 (команда touch) с правом записи для членов группы (g+w) (рисунок 14):

*Рисунок 14:*



2.1 Убедимся в том, что искомый файл io.h существует и находится в кааталоге /usr/include/sys, для этого воспользуемся командой просмотра содержимого ls. Скопируем файл /usr/include/sys/io.h в домашний каталог (команда cp) и назовем его equipment - переименуем командой mv. (рисунок 15)

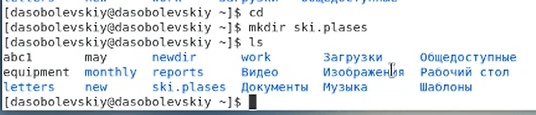
*Рисунок 15:*



Смотрим на промежуточные и конечный выводы команды ls, которую используем в качестве проверки наших дейсвтий, видим, что действия выполнены верно, и теперь в домашнем каатлоге содержится файл equipment.

2.2. В домашнем каталоге создадим директорию ~/ski.plases командой mkdir и проверим правильность действий просмотром содержимого домашнего каталога (рисунок 16):

*Рисунок 16:*



По результату вывода команды ls видим, что теперь каталог ~/ski.plases содержится в домашнем каталоге (подсвечен синим).

2.3. Переместим файл equipment в каталог ~/ski.plases командой mv. Просмотрим содержимое каталога ~/ski.plases командой ls (рисунок 17).

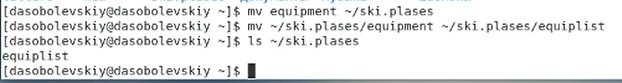
*Рисунок 17:*



Видим, что файл действительно находится на своем месте.

2.4. Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist командой mv (рисунок 18).

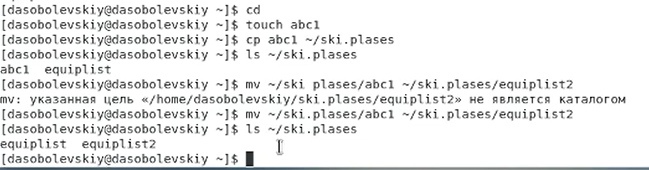
*Рисунок 18:*



Просмотрим содержимое каталога и убедимся в правильности действий - теперь катлог содержит файл equiplist.

2.5. Создаем в домашнем каталоге файл abc1 командой touch и копируем его в каталог ~/ski.plases командой cp, называем его equiplist2 - переименовываем в equiplist2 командой mv (рисунок 19).

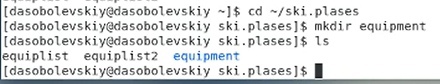
*Рисунок 19:*



Правильность действий проверяем периодическим вызовом команды ls для просмотра содержимого катлога ~/ski.plases. Смотрим на результаты выводов и видим, что мы выполнили все верно и теперь в каталоге содержится 2 разных файла equiplist.

2.6. Создадим каталог с именем equipment в каталоге ~/ski.plases с помощью команды mkdir, для этого сначал перйдем в каталог ~/ski.plases (рисунок 20).

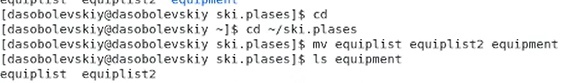
*Рисунок 20:*



Проверяем правильность действий просмотром содержимого каталога. Видим, что новый каатлог был в него добавлен (выделен синим)

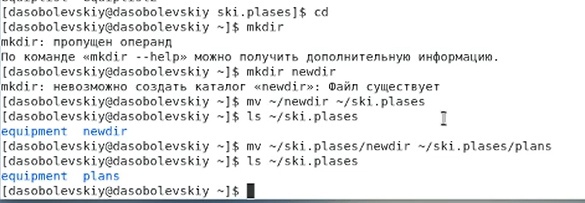
2.7. Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment с помощью команды mv и проверяем содержимое каталога equipment командой ls, видим, что все верно (рисунок 21).

*Рисунок 21:*



2.8. Создадим (mkdir) и переместим каталог ~/newdir в каталог ~/ski.plases, используя команду mv, и назовем его plans - переименуем при помощи той же команды mv (рисунок 22). По ходу действий проверяем верность выполнения просмотром содержимого каталога ls.

*Рисунок 22:*



Смотрим на выводы ls - работа выполнена верно.

1. Определяем опции команды chmod:

* 3.1. drwxr--r-- ... australia (рисунок 23) Нам нужно обозначить неограниченные права для владельца каталога и право только на чтение для группы и других пользователей.
* Для этого сначала создадим катлог australia командой mkdir (создаем каталог, поскольку первый знаком является d, следовательно australia - это каталог). Проверим его свойства и зададим новые права доступа командой chmod и следующими комбинациями: u+r+w+x - для пользователя, g+r-w-x/ o+r-w-x - для группы и остальных пользователей.

*Рисунок 23:*



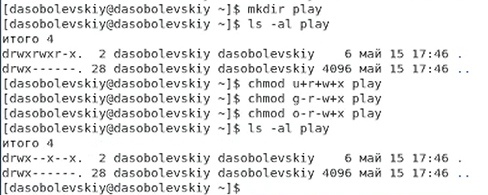
Проверим свойства командой ls. Видим, что права доступа стали именно такими, как требовалось (рисунок 23).

3.2. drwx--x--x ... play (рисунок 24)

Нам нужно обозначить неограниченные права для владельца каталога и право только на выполнение для группы и других пользователей.

Для этого сначала создадим катлог play командой mkdir (создаем каталог, поскольку первый знаком является d, следовательно play - это каталог). Проверим его свойства и зададим новые права доступа командой chmod и следующими комбинациями: u+r+w+x - для пользователя, g-r-w+x/ o-r-w+x - для группы и остальных пользователей.

*Рисунок 24:*



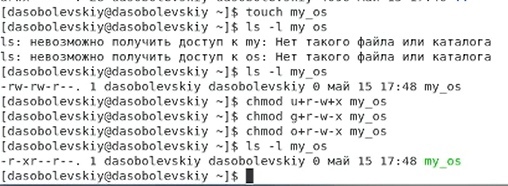
Проверим свойства командой ls. Видим, что права доступа стали именно такими, как требовалось (рисунок 24).

3.3. -r-xr--r-- ... my\_os (рисунок 25)

Нам нужно обозначить права для владельца файла на его чтение и выполнение и право только на чтение для группы и других пользователей.

Для этого сначала создадим файл my\_os командой touch (создаем файл, поскольку первый знаком является -, следовательно my\_os - это файл). Проверим его свойства и зададим новые права доступа командой chmod и следующими комбинациями: u+r-w+x - для пользователя, g+r-w-x/ o+r-w-x - для группы и остальных пользователей.

*Рисунок 25:*



Проверим свойства командой ls. Видим, что права доступа стали именно такими, как требовалось (рисунок 25).

3.4. -rw-rw-r-- ... feathers (рисунок 26)

Нам нужно обозначить права для владельца файла и группы на его чтение и запись и право только на чтение для других пользователей.

Для этого сначала создадим файл feathers командой touch (создаем файл, поскольку первый знаком является -, следовательно feathers - это файл). Проверим его свойства и зададим новые права доступа командой chmod и следующими комбинациями: u+r+w-x/ g+r+w-x - для пользователя и группы, o+r-w-x - для остальных пользователей.

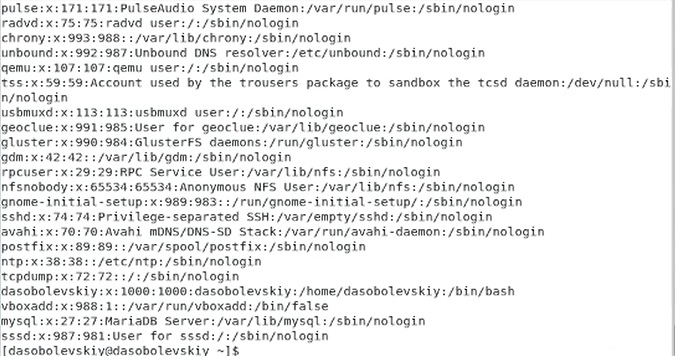
*Рисунок 26:*



Проверим свойства командой ls. Видим, что права доступа стали именно такими, как требовалось (рисунок 26).

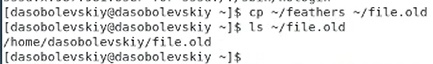
4.1. Просмотрим содержимое файла /etc/passwd с помощью команды cat (рисунок 27).

*Рисунок 27:*



4.2. Скопируем файл ~/feathers в файл ~/file.old командой cp, проверяем командой ls, что такой файл появился (рисунок 28).

*Рисунок 28:*



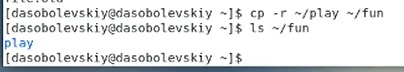
4.3. Переместим файл ~/file.old в каталог ~/play с помомщью команды mv, проверим правильность действий просмотром содержимого каталога, по выводу команды ls видим, что мы все сделали верно (рисунок 29).

*Рисунок 29:*



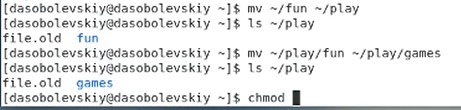
4.4. Скопируем каталог ~/play в каталог ~/fun командой cp, проверим правильность действий просмотром содержимого каталога, по выводу команды ls видим, что мы все сделали верно (рисунок 30).

*Рисунок 30:*



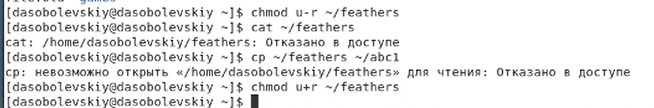
4.5. Переместим каталог ~/fun в каталог ~/play командой mv и назовите его games - переименуем его командой mv в games (рисунок 31). Проверим правильность действий просмотром содержимого каталога, по выводу команды ls видим, что мы все сделали верно (рисунок 31).

*Рисунок 31:*



4.6. Лишим владельца файла ~/feathers права на чтение командой chmod u-r ... (рисунок 32).

*Рисунок 32:*



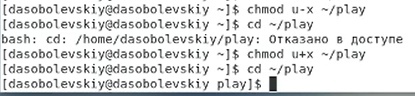
4.7. Посмотрим, что произойдёт, если мы попытаемся просмотреть файл ~/feathers командой cat (рисунок 32). Видим, что мы не можем просмотреть этот файл, система выдает сообщение об отказе в доступе, поскольку ранее мы его ограничили, следовательно мы все сделали правильно.

4.8. Теперь проверим, что произойдёт, если мы попытаемся скопировать файл ~/feathers в файл, например, ~/abc1 (рисунок 32). Система выдает сообщение о том, что мы не можем открыть файл для чтение - отказано в доступе - все верно.

4.9. Возвращаем владельцу файла ~/feathers право на чтение r командой chmod (рисунок 32).

4.10. Лишим владельца каталога ~/play права на выполнение командой chmod u-x ... (рисунок 33).

*Рисунок 33:*

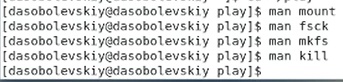


4.11. Пробуем перейти в каталог ~/play (рисунок 33). У нас не получается этого сделать, поскольку мы ограничили права в пункте 4.10 - отказано в доступе.

4.12. Вернем владельцу каталога ~/play право на выполнение через команду chmod и снова попробуем перейти в него (рисунок 33). На этот раз это происходит успешно, поскольку мы вернули владельцу права на выполнение.

1. Прочитаем man по командам mount, fsck, mkfs, kill (рисунок 34).

*Рисунок 34:*



Теперь мы можем дать краткую характеристику по данным командам:

• mount - утилита командной строки в UNIX-подобных операционных системах, которая применяется для монтирования файловых систем:

$ mount <файл\_устройства> <папка\_назначения>  
  
или  
  
$ mount <опции> -t <файловая\_система> -o <опции\_монтирования> <файл\_устройства> <папка\_назначения>

• fsck - команда UNIX, которая проверяет и устраняет ошибки в файловой системе:

$ fsck [опции] [опции\_файловой\_системы] [раздел\_диска]

• mkfs - позволяет создать файловую систему Linux:

$ sudo mkfs -t <тип> <устройство>

• kill - посылает указанный сигнал указанному процессу:

kill [ -s сигнал | -p ] [ -a ] pid...  
  
или  
  
kill -l [ сигнал ]

**Вывод:**

Ознакомился с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрел практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы