

Лабораторная работа №6

Бондарь Алексей Олегович^{inst{1}}

11 мая 2021 год, Москва, Россия

¹RUDN University, Moscow, Russian Federation

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 лабораторной работы

Выполнение лабораторной работы

1) Для начала выполним примеры, описанные в первой части описания лабораторной работы. (рис. -fig. ??): 1. Скопируем файл ~/abc1 в файл april и в файл may. Для этого создадим файл abc1, используя команду «touch abc1», далее осуществим копирование с помощью команд «сrabс1 april» и «сrabс1 may». 2. Скопируем файлы april и may в каталог monthly, используя команды «mkdir monthly» – для создания каталога monthly и «сrapril may monthly» – для копирования. 3. Скопируем файл monthly/may в файл с именем june.

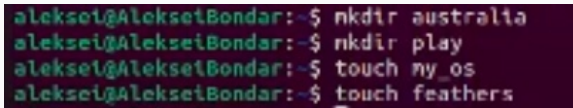
Выполним команды «ср monthly/may monthly/june» и «ls monthly» (для просмотра содержимого каталога). 4. Скопируем каталог monthly в каталог monthly.00. Для этого создадим каталог monthly.00 командой «mkdir monthly.00» и осуществим копирование, используя команду «ср -r monthly monthly.00» (команда ср с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами). 5. Скопируем каталог monthly.00 в каталог /tmp, используя команду «ср -r

Выполнение лабораторной работы

2) Выполняем следующие действия, отображенные на (рис. -fig. ??) и (рис. -fig. 2) : 1. Копируем файл `/usr/include/aio.h` (т.к. у меня нет каталога `/usr/include/sys/`, то беру произвольный файл из каталога `/usr/include/`) в домашний каталог (команда «`cp /usr/include/aio.h ~`») и называем его `equipment` (команда «`mv aio.h equipment`»). 2. В домашнем каталоге создаем директорию `~/ski.places` (команда «`mkdir ski.places`»). 3. Перемещаем файл `equipment` в каталог `~/ski.places` (команда «`mv equipment ski.places`»). 4. Переименовываем файл `~/ski.places/equipment` в `~/ski.places/equiplist` (команда «`mv ski.places/equipment ski.places/equiplist`»). ## Выполнение лабораторной работы 5. Создаем в домашнем каталоге файл `abc1` (команда «`touch abc1`») и копируем его в каталог `~/ski.places` (команда «`cp abc1 ski.places`»), называем его `equiplist2` (команда «`mv ski.places/abc1 ski.places/equiplist2`»). 6. Создаем каталог с именем `equipment` в каталоге `~/ski.places` (команда «`mkdir ski.places/equipment`»). 7.

Выполнение лабораторной работы

3) Определяем опции команды `chmod`, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «`mkdir australia`», «`mkdir play`», «`touch my_os`», «`touch feathers`». (рис. -fig. ??)

A screenshot of a terminal window with a dark background and light-colored text. It shows four lines of commands being executed by a user named 'aleksei@alekseiBondar'. The commands are: 'mkdir australia', 'mkdir play', 'touch my_os', and 'touch feathers'. Each line shows the prompt, the command, and a cursor at the end.

```
aleksei@alekseiBondar:~$ mkdir australia
aleksei@alekseiBondar:~$ mkdir play
aleksei@alekseiBondar:~$ touch my_os
aleksei@alekseiBondar:~$ touch feathers
```

Выполнение

лабораторной работы (рис. -fig. 3): `drwxr-r...` `australia`: команда «`chmod 744 australia`» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение). `drwx-x-x ...` `play`: команда «`chmod 711 play`» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение). ## Выполнение лабораторной работы `-r-xr-r...` `my_os`: команды «`chmod 544 my_os`» (это файл, владелец имеет право на чтение и

Выполнение лабораторной работы

4) Выполняем следующие действия, отображенные на (рис. -fig. ??) и (рис. -fig. ??): 1. Просмотрим содержимое файла `/etc/passwd` (команда `cat/etc/passwd`). 2. Копируем файл `~/feathers` в файл `~/file.old` (команда `cpfeathersfile.old`). 3. Переместим файл `~/file.old` в каталог `~/play` (команда `mvfile.ordplay`). 4. Скопируем каталог `~/play` в каталог `~/fun` (команда `cp-rplayfun`). ## Выполнение лабораторной работы 5.

Переместим каталог `~/fun` в каталог `~/play` (команда `mvfunplay`) и назовем его `games` (команда `mvplay/funplay/games`). 6. Лишим владельца файла `~/feathers` права на чтение (команда `chmod u-rfeathers`). 7. Если мы попытаемся просмотреть файл `~/feathers` командой `cat`, то получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на чтение данного файла. 8. Если мы попытаемся скопировать файл `~/feathers`, например, в каталог `monthly`, то получим отказ в доступе, по причине, описанной в предыдущем пункте. 9. Дадим владельцу файла `~/feathers` право на чтение (команда `chmodu+rfeathers`). ## Выполнение

Выполнение лабораторной работы

```
MOUNT(8)                                System Administration                                MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-fFnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount --make-([shared|slave|private|unbindable|rshared|rslave|rpri-
    vate|runbindable]) mountpoint

DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree,
    the file hierarchy, rooted at /. These files can be spread out over
    several devices. The mount command serves to attach the filesystem
    found on some device to the big file tree. Conversely, the umount(8)
    command will detach it again. The filesystem is used to control how
    data is stored on the device or provided in a virtual way by network

Manual page mount(8) line 1 (press h for help or q to quit)
```

Выполнение

лабораторной работы - Команда fsck:это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux.Он использует программы, специфичные для типа файловой системы, которую он проверяет.У командыfsckследующий синтаксис:fsck [параметр] -[параметры ФС] [. . .] Например, если нужно восстановить