

Лабораторная работа №6

Бондарь Алексей Олегович НПМбд-01-20¹

10 мая, 2021, 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

1) Для начала выполним примеры, описанные в первой части описания лабораторной работы. (рис. 1):

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 `срabc1 april` и `срabc1 may`.
2. Скопируем файлы `april` и `may` в каталог `monthly`, используя команды `mkdir monthly` – для создания каталога `monthly` и `срapril may monthly` – для копирования.
3. Скопируем файл `monthly/may` в файл с именем `june`. Выполним команды `срmonthly/may monthly/june` и `ls monthly` (для просмотра содержимого каталога).
4. Скопируем каталог `monthly` в каталог `monthly.00`. Для этого

```
aleksei@alekseiBondar:~$ touch abc1
aleksei@alekseiBondar:~$ cp abc1 april
aleksei@alekseiBondar:~$ cp abc1 may
aleksei@alekseiBondar:~$ mkdir monthly
aleksei@alekseiBondar:~$ cp april may monthly
aleksei@alekseiBondar:~$ cp monthly/may monthly/june
aleksei@alekseiBondar:~$ ls monthly
april  june  may
aleksei@alekseiBondar:~$ mkdir monthly.00
aleksei@alekseiBondar:~$ cp -r monthly monthly.00
aleksei@alekseiBondar:~$ cp -r monthly.00 /tmp
aleksei@alekseiBondar:~$ clear
```

Figure 1: 1

(рис. 2):

1. Изменим название файла `april` на `july` в домашнем каталоге, используя команду `mv april july`.
2. Переместим файл `july` в каталог `monthly.00` с помощью команды `mv july monthly.00`. Проверим результат командой `ls monthly.00`.
3. Переименуем каталог `monthly.00` в `monthly.01`, используя команду `mv monthly.00 monthly.01`.
4. Переместим каталог `monthly.01` в каталог `reports`. Для этого создадим каталог `reports` с помощью команды `mkdir reports` и выполним перемещение командой `mv monthly.01 reports`.
5. Переименуем каталог `reports/monthly.01` в `reports/monthly` командой `mv reports/monthly.01 reports/monthly`

```
aleksei@AleksaiBondar:~$ cd
aleksei@AleksaiBondar:~$ mv april july
aleksei@AleksaiBondar:~$ mv july monthly.00
aleksei@AleksaiBondar:~$ ls monthly.00
july  monthly
aleksei@AleksaiBondar:~$ mv monthly.00 monthly.01
aleksei@AleksaiBondar:~$ mkdir reports
aleksei@AleksaiBondar:~$ mv monthly.01 reports
aleksei@AleksaiBondar:~$ mv reports/monthly.01 reports/monthly
aleksei@AleksaiBondar:~$ clear
```

Figure 2: 2

(рис. 3) и (рис. 4)

1. Создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may» (изменение прав), «ls -l may»
2. Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле).
3. Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly»(создание каталога), «chmod go -r monthly»(изменение прав).
4. Создаем файл ~/abc1 с правом записи для членов группы,


```
aleksei@AlekseiBondar:~$ cd
aleksei@AlekseiBondar:~$ touch may
aleksei@AlekseiBondar:~$ ls -l may
-rw-rw-r-- 1 aleksei aleksei 0 мая 15 20:29 may
aleksei@AlekseiBondar:~$ chmod u+x may
aleksei@AlekseiBondar:~$ ls -l may
-rwxrwx-r-- 1 aleksei aleksei 0 мая 15 20:29 may
aleksei@AlekseiBondar:~$ chmod u-x may
aleksei@AlekseiBondar:~$ ls -l may
-rw-rw-r-- 1 aleksei aleksei 0 мая 15 20:29 may
aleksei@AlekseiBondar:~$ cd
aleksei@AlekseiBondar:~$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: файл существует
```

Figure 3: 3

```
aleksei@AlekseiBondar:~$ chmod go-r monthly  
aleksei@AlekseiBondar:~$ cd  
aleksei@AlekseiBondar:~$ touch abc1  
aleksei@AlekseiBondar:~$ chmod g+w abc1
```

Figure 4: 4

2)Выполняем следующие действия, отображенные на (рис. 5) и (рис. 6) :

1. Копируем файл `/usr/include/aio.h`(т.к. у меня нет каталога `/usr/include/sys/`, то беру произвольный файл из каталога `/usr/include/`)вдомашнийкаталог(команда«`cp /usr/include/aio.h ~`») и назоваем 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«`mvski.places/equipment ski.places/equiplist`»).
5. Создаем в домашнем каталоге файл `abc1`(команда«`touchabc1`») и

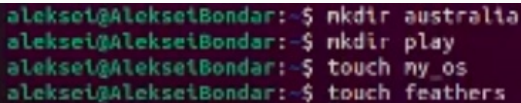
```
aleksei@AlekseiBondar:~$ cp /usr/include/stdio.h -
aleksei@AlekseiBondar:~$ mv stdio.h equipment
aleksei@AlekseiBondar:~$ mkdir ski.places
aleksei@AlekseiBondar:~$ mv equipment ski.places
aleksei@AlekseiBondar:~$ mv ski.places/equipment ski.places/equiplist
mv: невозможно переместить 'ski.places/equipment' в 'ski.places/equiplist'
такого файла или каталога
aleksei@AlekseiBondar:~$ mv ski.places/equipment ski.places/equiplist
aleksei@AlekseiBondar:~$ touch abc1
aleksei@AlekseiBondar:~$ cp abc1 ski.places
aleksei@AlekseiBondar:~$ mv ski.places/abc1 ski.places/equiplist2
aleksei@AlekseiBondar:~$ mkdir ski.places/equipment
```

Figure 5: 5

```
aleksei@AlekseiBondar:~$ mv skl.places/equip1list skl.places/equip1list2 skl.places/equipment
aleksei@AlekseiBondar:~$ mkdir newdir
aleksei@AlekseiBondar:~$ mv newdir skl.places
aleksei@AlekseiBondar:~$ mv skl.places/newdir skl.places/plans
aleksei@AlekseiBondar:~$
```

Figure 6: 6

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. 7)



```
aleksei@AlekseiBondar:~$ mkdir australia
aleksei@AlekseiBondar:~$ mkdir play
aleksei@AlekseiBondar:~$ touch ny_os
aleksei@AlekseiBondar:~$ touch feathers
```

Figure 7: 7

(рис. 8):

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). -r-xr-r... my_os: команды «chmod 544 my_os» (это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные – только чтение). -rw-rw-r... feathers: команды «chmod 664 feathers» (это файл, владелец и группа владельца имеют право на чтение и запись, остальные – только чтение). Командой «ls -l» проверяем правильность выполненных действий.

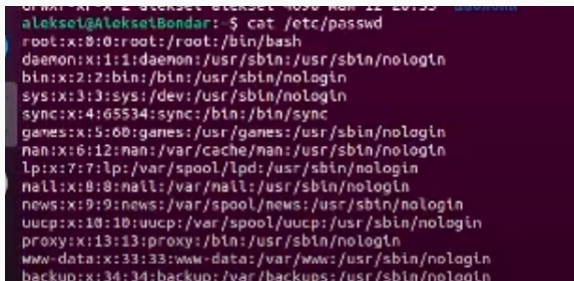
Выполнение лабораторной работы

```
aleksei@alekseiBondar:~$ chmod 744 australia
aleksei@alekseiBondar:~$ chmod 711 play
aleksei@alekseiBondar:~$ chmod 544 ny_os
aleksei@alekseiBondar:~$ chmod 664 feathers
aleksei@alekseiBondar:~$ ls -l
итого 68
-rw-rw-r-- 1 aleksei aleksei  0 мая 15 20:36 abc1
drwxr--r-- 2 aleksei aleksei 4096 мая 15 20:42 australia
-rw-rw-r-- 1 aleksei aleksei  0 мая 15 20:42 feathers
drwxrwxr-x 3 aleksei aleksei 4096 мая 13 22:36 lab3
-rw-rw-r-- 1 aleksei aleksei  0 мая 15 20:29 may
drwx-mx--x 2 aleksei aleksei 4096 мая 15 20:23 monthly
-r-xr--r-- 1 aleksei aleksei  0 мая 15 20:42 ny_os
drwxrwxr-x 2 aleksei aleksei 4096 мая 14 12:54 pandoc-crossref
drwx--x--x 2 aleksei aleksei 4096 мая 15 20:42 play
drwxrwxr-x 3 aleksei aleksei 4096 мая 15 20:28 reports
drwxrwxr-x 4 aleksei aleksei 4096 мая 15 20:41 ski_places
drwxr-xr-x 3 aleksei aleksei 4096 мая 14 01:18 snap
drwxrwxr-x 4 aleksei aleksei 4096 мая 14 00:04 work
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Видео
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Документы
drwxr-xr-x 3 aleksei aleksei 4096 мая 14 13:26 Загрузки
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Изображения
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Музыка
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Общедоступные
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 'Рабочий стол'
drwxr-xr-x 2 aleksei aleksei 4096 мая 12 20:35 Кабаны
```

Figure 8: 8

4)Выполняем следующие действия, отображенные на (рис. 9) и (рис. 10):

1. Просмотримсодержимое файла `/etc/passwd` (команда «`cat/etc/passwd`»).
2. Копируемфайл `~/feathers` в файл `~/file.old`(команда «`cp feathers file.old`»).
3. Переместимфайл `~/file.old` в каталог `~/play`(команда «`mv file.old play`»).
4. Скопируемкаталог `~/play` в каталог `~/fun`(команда «`cp -r play fun`»).
5. Переместимкаталог `~/fun` в каталог `~/play`(команда «`mv funplay`»)и назовемего `games`(команда «`mv play/fun play/games`»).
6. Лишимвладельца файла `~/feathers` права на чтение(команда «`chmod u-r feathers`»).

A terminal window with a dark background and light-colored text. The prompt is 'aleksei@AleksaiBondar:~\$'. The command 'cat /etc/passwd' has been executed, displaying the contents of the /etc/passwd file. The output lists system and regular users with their IDs, names, shells, and home directories.

```
aleksei@AleksaiBondar:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
nan:x:6:12:nan:/var/cache/nan:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
nail:x:8:8:nail:/var/nail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

Figure 9: 9

```
aleksei@AlekselBondar:~$ cp feathers file.old
aleksei@AlekselBondar:~$ mv file.old play
aleksei@AlekselBondar:~$ cp -r play fun
aleksei@AlekselBondar:~$ mv fun play
aleksei@AlekselBondar:~$ mv play/fun play/games
aleksei@AlekselBondar:~$ chmod u-r feathers
aleksei@AlekselBondar:~$ cat feathers
cat: feathers: Отказано в доступе
aleksei@AlekselBondar:~$ cp feathers monthly
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
aleksei@AlekselBondar:~$ chmod u+r feathers
aleksei@AlekselBondar:~$ chmod u-x play
aleksei@AlekselBondar:~$ cd play
bash: cd: play: Отказано в доступе
aleksei@AlekselBondar:~$ chmod u+x play
aleksei@AlekselBondar:~$
```

Figure 10: 10

5)Используя команды «man mount», «man fsck»,«man mkfs»,«man kill», получим информацию о соответствующих командах ((рис. 11), (рис. 12), (рис. 13), (рис. 14)).

- Команда `mount`: предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается как `/`. Физически файлы могут располагаться на различных устройствах. Команда `mount` служит для подключения файловых систем разных устройств к этому большому дереву. Наиболее часто встречающаяся форма команды `mount` выглядит следующим образом: `mount -t vfstype device dir`. Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа `vfstype`, расположенную на устройстве `device`, к заданному каталогу `dir`, который часто называют точкой монтирования.

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 --nake--(shared|slave|private|unbindable|rshared|rslave|private|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)
```

Figure 11: 11

- Команда `fsck`: это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Она использует программы, специфичные для типа файловой системы, которую она проверяет. У команды `fsck` следующий синтаксис: `fsck [параметр] -[параметры ФС] [. . .]`. Например, если нужно восстановить («починить») файловую систему на некотором устройстве `/dev/sdb2`, следует воспользоваться командой: «`sudo fsck -y /dev/sdb2`». Опция `-y` необходима, т.к. при её отсутствии придётся слишком часто давать подтверждение.


```
FCK(8)                                System Administration                                FCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--]
    [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=8868abf6-88c5-4a93-98b8-bfc24057f7bd or LA-BEL=root). Normally, the fsck program will try to handle filesystems on different physical disk drives in parallel to reduce the total amount of time needed to check all of them.

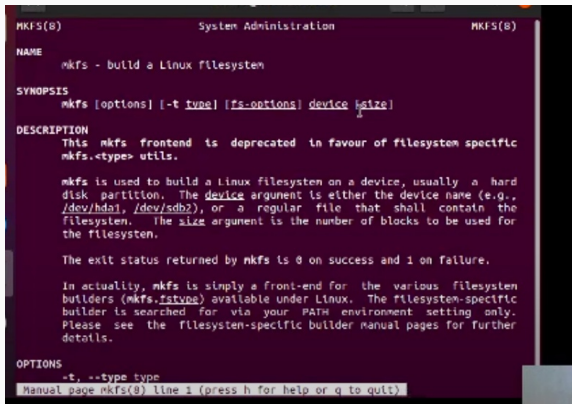
    If no filesystems are specified on the command line, and the -A option is not specified, fsck will default to checking filesystems in /etc/fstab serially. This is equivalent to the -As options.

    The exit status returned by fsck is the sum of the following conditions:

        0      No errors
        1      Filesystem errors corrected
        2      System should be rebooted
```

Figure 12: 12

- Команда `mkfs`: создаёт новую файловую систему Linux. Имеет следующий синтаксис: `mkfs[-V] [-tfstype] [fs-options] filesys[blocks]` `mkfs` используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента `filesys` для файловой системы может выступать или название устройства (например, `/dev/hda1`, `/dev/sdb2`) или точка монтирования (например, `/usr`, `/home`). Аргументом `blocks` указывается количество блоков, которые выделяются для использования этой файловой системой. По окончании работы `mkfs` возвращает 0 - в случае успеха, а 1 - при неудачной операции. Например, команда «`mkfs -t ext2 /dev/hdb1`» создаёт файловую систему типа `ext2` в разделе `/dev/hdb1` (второй жёсткий диск).



```

MKFS(8)                                System Administration          MKFS(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [fs-options] device size

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific
    mkfs.<type> utils.

    mkfs is used to build a Linux filesystem on a device, usually a hard
    disk partition. The device argument is either the device name (e.g.,
    /dev/hda1, /dev/sdb2), or a regular file that shall contain the
    filesystem. The size argument is the number of blocks to be used for
    the filesystem.

    The exit status returned by mkfs is 0 on success and 1 on failure.

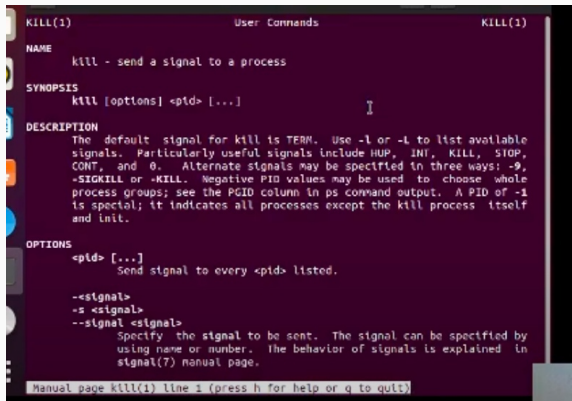
    In actuality, mkfs is simply a front-end for the various filesystem
    builders (mkfs.<type>) available under Linux. The filesystem-specific
    builder is searched for via your PATH environment setting only.
    Please see the filesystem-specific builder manual pages for further
    details.

OPTIONS
    -t, --type type
Manual page mkfs(8) line 1 (press h for help or q to quit)

```

Figure 13: 13

- Команда `kill`: посылает сигнал процессу или выводит список допустимых сигналов. Имеет следующий синтаксис: `kill[опции] PID`, где PID – это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам. Например, команда «`kill -KILL 3121`» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс.



```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available
    signals. Particularly useful signals include HUP, INT, KILL, STOP,
    CONT, and 0. Alternate signals may be specified in three ways: -9,
    -SIGKILL or -KILL. Negative PID values may be used to choose whole
    process groups; see the PGID column in ps command output. A PID of -1
    is special; it indicates all processes except the kill process itself
    and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by
        using name or number. The behavior of signals is explained in
        signal(7) manual page.

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

Figure 14: 14

В ходе выполнения данной лабораторной работы я ознакомился файловой системой Linux, её структурой, именами и содержанием каталогов, получила навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.