

Отчёт по лабораторной работе №5

Дисциплина: Операционные системы

Татьяна Александровна Лебединец

Содержание

Цель работы	5
Задание	6
Выполнение лабораторной работы	7
Выводы	15
Список литературы	16

Список иллюстраций

Список таблиц

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

Задание

Проверить работу команд, представленных в методичке.

Выполнение лабораторной работы

Шаг 1 Выполним все примеры, представленные в первой части описания лр. (рис. [fig:001]) (рис. [fig:002]) (рис. [fig:003]) (рис. [fig:004]) (рис. [fig:005]) (рис. [fig:006]) (рис. [fig:007]) (рис. [fig:008]) (рис. [fig:009]) (рис. [fig:010]) (рис.

[fig:011])

```
[talebedinec@fedora ~]$ cd  
[talebedinec@fedora ~]$ touch abc1  
[talebedinec@fedora ~]$ cp abc1 april  
[talebedinec@fedora ~]$ cp abc1 may
```

{#fig:001 width=70%}

```
[talebedinec@fedora ~]$ mkdir monthly  
[talebedinec@fedora ~]$ cp april may monthly  
[talebedinec@fedora ~]$ cp monthly/may monthly/june  
[talebedinec@fedora ~]$ ls monthly  
april june may
```

{#fig:002 width=70%}

```
[talebedinec@fedora ~]$ mkdir monthly.00  
[talebedinec@fedora ~]$ cp -r monthly monthly.00  
[talebedinec@fedora ~]$ cp -r monthly.00 /tmp  
[talebedinec@fedora ~]$
```

{#fig:003 width=70%}

```
[talebedinec@fedora ~]$ mkdir monthly.00
[talebedinec@fedora ~]$ cp -r monthly monthly.00
[talebedinec@fedora ~]$ cp -r monthly.00 /tmp
[talebedinec@fedora ~]$ cd
[talebedinec@fedora ~]$ mv april july
[talebedinec@fedora ~]$ mv july monthly.00
[talebedinec@fedora ~]$ ls monthly.00
july  monthly
[talebedinec@fedora ~]$ mv monthly.00 monthly.01
[talebedinec@fedora ~]$ mkdir reports
[talebedinec@fedora ~]$ mv monthly.01 reports
[talebedinec@fedora ~]$ mv reports/monthly.01 reports/monthly
[talebedinec@fedora ~]$
```

{#fig:004 width=70%}

```
[talebedinec@fedora ~]$ cd
[talebedinec@fedora ~]$ touch may
[talebedinec@fedora ~]$ ls -l may
-rw-rw-r--. 1 talebedinec talebedinec 0 мая  5 15:49 may
[talebedinec@fedora ~]$ chmod u+x may
[talebedinec@fedora ~]$ ls -l may
-rwxrw-r--. 1 talebedinec talebedinec 0 мая  5 15:49 may
[talebedinec@fedora ~]$
```

{#fig:005 width=70%}

```
[talebedinec@fedora ~]$ chmod u-x may
[talebedinec@fedora ~]$ ls -l may
-rw-rw-r--. 1 talebedinec talebedinec 0 мая  5 15:49 may
[talebedinec@fedora ~]$ cd
[talebedinec@fedora ~]$ mkdir monthly
```

{#fig:006 width=70%}

```
[talebedinec@fedora ~]$ chmod o-r monthly
[talebedinec@fedora ~]$ chmod g-r monthly
[talebedinec@fedora ~]$ ls -l monthly
-rw-rw-r--. 1 talebedinec talebedinec 0 мая  5 15:41 april
-rw-rw-r--. 1 talebedinec talebedinec 0 мая  5 15:42 june
-rw-rw-r--. 1 talebedinec talebedinec 0 мая  5 15:41 may
[talebedinec@fedora ~]$ cd
[talebedinec@fedora ~]$ chmod q+w abc1
chmod: неверный режим: «q+w»
по команде «chmod --help» можно получить дополнительную информацию.
[talebedinec@fedora ~]$ chmod g+w abc1
```


{#fig:007 width=70%}

```
[talebedinec@fedora ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=986256k,nr_inodes=246564,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=402660k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/sda2 on / type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache,subvol=258,subvol=/root)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=31,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=15329)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
/dev/sda2 on /home type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache,subvol=256,subvol=/home)
/dev/sda1 on /boot type ext4 (rw,relatime,seclabel)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,nr_inodes=409600,inode64)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=201328k,nr_inodes=50332,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
```

{#fig:008 width=70%}

```
[talebedinec@fedora ~]$ cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Apr 28 07:50:41 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=9d54732d-bc4b-40ef-accd-5d9126f0b944 / btrfs subvol=root,compress=zstd:1 0 0
UUID=332f4b85-d9dd-44cf-926d-f00ac5302e80 /boot ext4 defaults 1 2
UUID=9d54732d-bc4b-40ef-accd-5d9126f0b944 /home btrfs subvol=home,compress=zstd:1 0 0
[talebedinec@fedora ~]$
```

{#fig:009 width=70%}

```
[talebedinec@fedora ~]$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs           986256            0   986256            0% /dev
tmpfs              1006644          10504   996140            2% /dev/shm
tmpfs              402660           1384   401276            1% /run
/dev/sda2          40893440        9665652  30080428          25% /
/dev/sda2          40893440        9665652  30080428          25% /home
/dev/sda1          996780          175308   752660           19% /boot
tmpfs              1006644          34584   972060            4% /tmp
tmpfs              201328           160    201168            1% /run/user/1000
[talebedinec@fedora ~]$
```

{#fig:010 width=70%}

```
[talebedinec@fedora ~]$ fsck /dev/sda1
fsck из util-linux 2.37.2
e2fsck 1.46.3 (27-Jul-2021)
/dev/sda1 is mounted.

WARNING!!! The filesystem is mounted.  If you continue you ***WILL***
cause ***SEVERE*** filesystem damage.

Do you really want to continue<n>? no
check aborted.
[talebedinec@fedora ~]$
```

{#fig:011 width=70%}

Шаг 2 2.1 Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовем его `equipment`. 2.2. В домашнем каталоге создаем директорию `~/ski.places`. 2.3. Переместим файл `equipment` в каталог `~/ski.places`. 2.4. Переименуем файл `~/ski.places/equipment` в `~/ski.places/equiplist`. 2.5. Создаем в домашнем каталоге файл `abc1` и копируем его в каталог `~/ski.places`, назовем его `equiplist2`. 2.6. Создаем каталог с именем `equipment` в каталоге `~/ski.places`. 2.7. Переместим файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`. 2.8. Создаем и переместим каталог `~/newdir` в каталог `~/ski.places` и назовем его `plans`(рис. [fig:012])

```
[talebedinec@fedora ~]$ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
[talebedinec@fedora ~]$ cp /usr/include/sys/io.h equipment
[talebedinec@fedora ~]$ mkdir ski.places
[talebedinec@fedora ~]$ mv equipment ski.places
[talebedinec@fedora ~]$ ls ski.places
equipment
[talebedinec@fedora ~]$ mv /ski.places/equipment /ski.places/equiplist
mv: не удалось выполнить stat для '/ski.places/equipment': Нет такого файла или каталога
[talebedinec@fedora ~]$ mv ~/ski.places/equipment ~/ski.places/equiplist
[talebedinec@fedora ~]$ ls ski.places
equiplist
[talebedinec@fedora ~]$ touch abc1
[talebedinec@fedora ~]$ cp abc1 ~/ski.places/equiplist2
[talebedinec@fedora ~]$ ls ski.places
equiplist  equiplist2
[talebedinec@fedora ~]$ mkdir ~/ski.places/equipment
[talebedinec@fedora ~]$ mv ~/ski.places/equiplist ~/ski.places/equipment
[talebedinec@fedora ~]$ ls ski.places/equipment
equiplist
[talebedinec@fedora ~]$ mkdir newdir
[talebedinec@fedora ~]$ mv ~/newdir ~/ski.places/plans
[talebedinec@fedora ~]$ ls ski.places
equiplist2  equipment  plans
[talebedinec@fedora ~]$ chmod 744 australia
chmod: невозможно получить доступ к 'australia': Нет такого файла или каталога
```

{#fig:012 width=70%}

Шаг 3 Определим опции команды `chmod`,необходимые для того,чтобы присвоить перечис-
ленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет: 3.1. `drwxr-r-` ... `australia` 3.2. `drwx-x-x` ... `play` 3.3. `-r-xr-r-` ...
`my_os` 3.4. `-rw-rw-r-` ... `feathers` (рис. [fig:013])

```
[talebedinec@fedora ~]$ touch australia play my_os feathers
[talebedinec@fedora ~]$ chmod 744 australia
[talebedinec@fedora ~]$ chmod 711 play
[talebedinec@fedora ~]$ chmod 554 my_os
[talebedinec@fedora ~]$ chmod 664 feathers
[talebedinec@fedora ~]$ ls -l play my_os australia feathers
-rwxr--r--. 1 talebedinec talebedinec 0 мая  5 16:23 australia
-rw-rw-r--. 1 talebedinec talebedinec 0 мая  5 16:23 feathers
-r-xr-xr--. 1 talebedinec talebedinec 0 мая  5 16:23 my_os
-rwx--x--x. 1 talebedinec talebedinec 0 мая  5 16:23 play
[talebedinec@fedora ~]$
```

{#fig:013 width=70%}

Шаг 4 Прделаем приведённые ниже упражнения, записывая в отчёт по лабора-
торной работе используемые при этом команды: 4.1. Просмотрим содержимое файла
`/etc/passwd`. (рис. [fig:015]) 4.2. Скопируем файл `~/feathers` в файл `~/file.old`.
(рис. [fig:015]) 4.3. Переместим файл `~/file.old` в каталог `~/play`. (рис. [fig:015]) 4.4.
Скопируем каталог `~/play` в каталог `~/fun`. (рис. [fig:015]) 4.5. Переместим каталог
`~/fun` в каталог `~/play` и назовите его `games`. (рис. [fig:015]) 4.6. Лишим владельца
файла `~/feathers` права на чтение. (рис. [fig:015]) 4.7. Узнаем, что произойдёт,если
мы попытаемся просмотреть файл `~/feathers` командой `cat`? (рис. [fig:015]) 4.8.
Узнаем, что произойдёт,если мы попытаемся скопировать файл `~/feathers`? 4.9. Да-
дим владельцу файла `~/feathers` право на чтение. (рис. [fig:016]) 4.10. Лишим
владельца каталога `~/play` права на выполнение. (рис. [fig:016]) 4.11. Перейдем
в каталог `~/play`.Что произошло? (рис. [fig:016]) 4.12. Дадим владельцу каталога
`~/play` право на выполнение. (рис. [fig:016])

```

ts: невозможно получить доступ к '/reports/monthly': нет такого файла или каталога
[talebenedinec@fedora ~]$ ls ~/reports/monthly
july  monthly
[talebenedinec@fedora ~]$ touch file.old
[talebenedinec@fedora ~]$ cp ~/feathers ~/file.old
[talebenedinec@fedora ~]$ mv ~/file.old ~/play
[talebenedinec@fedora ~]$ cp ~/play ~/fun
[talebenedinec@fedora ~]$ mv ~/fun ~/play/games
mv: не удалось получить доступ к '/home/talebenedinec/play/games': Это не каталог
[talebenedinec@fedora ~]$ mv ~/fun ~/play
[talebenedinec@fedora ~]$ mv fun games
mv: не удалось выполнить stat для 'fun': Нет такого файла или каталога
[talebenedinec@fedora ~]$ mv ~/fun games
mv: не удалось выполнить stat для '/home/talebenedinec/fun': Нет такого файла или каталога
[talebenedinec@fedora ~]$ chmod u-r ~/feathers
[talebenedinec@fedora ~]$ cat ~/feathers
cat: /home/talebenedinec/feathers: Отказано в доступе
[talebenedinec@fedora ~]$ 

```

{#fig:015 width=70%}

```

[talebenedinec@fedora ~]$ chmod u-r ~/feathers
[talebenedinec@fedora ~]$ cat ~/feathers
cat: /home/talebenedinec/feathers: Отказано в доступе
[talebenedinec@fedora ~]$ cp ~/feathers
cp: после '/home/talebenedinec/feathers' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
[talebenedinec@fedora ~]$ cp feathers
cp: после 'feathers' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
[talebenedinec@fedora ~]$ cp feathers monthly
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[talebenedinec@fedora ~]$ 

```

{#fig:016 width=70%}

Шаг 5 Прочитаем man по командам mount,fsck,mkfs,kill и кратко их охарактеризуйте, приведя примеры. 1. mount (рис. [fig:018]) 2. fsck (рис. [fig:019]) 3. mkfs (рис. [fig:020]) 4. kill (рис. [fig:021])

```
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|rprivate|runbindable] mountpoint

DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /.
    Manual page mount(8) line 1 (press h for help or q to quit)
```

{#fig:018 width=70%}

```
talebedinec@fedora:~ — man fsck
FSCK(8)                                     System Administration                                     FSCK(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=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=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
```

{#fig:019 width=70%}

```

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.fstype) 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

```

{#fig:020 width=70%}

```

tatebedinec@redora: ~$ man kill

KILL(1)                                     User Commands                                     KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...

    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to the specified processes or process groups.

    If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --queue options, and the possibility to specify processes by

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

```

{#fig:021 width=70%}

Выводы

Я ознакомилась с файловой системой, приобрела практические навыки по применению команд для работы с файлами и каталогами.

Список литературы