

# Лабораторная работа №7

## Архитектура компьютера и операционные системы

Бабенко Константин, НКАбд-01-23

### Содержание

1	Цель работы .....	1
2	Выполнение лабораторной работы.....	1
2.1	Домашняя работа.....	Ошибка! Закладка не определена.
3	Выводы .....	2

## 1 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.

## 2 Выполнение лабораторной работы

Скопирую файл `/usr/include/sys/io.h` в домашний каталог и называю его `equipment`. Если файла `io.h` нет, то использую любой другой файл в каталоге `/usr/include/sys/` вместо него. В домашнем каталоге создаю директорию `~/ski.plases`. Перемещаю файл `equipment` в каталог `~/ski.plases`. Переименую файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`. Создаю в домашнем каталоге файл `abc1` и копирую его в каталог `~/ski.plases`, назовите его `equiplist2`. Создаю каталог с именем `equipment` в каталоге `~/ski.plases`. Перемещаю файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment`. Создаю и перемещаю каталог `~/newdir` в каталог `~/ski.plases` и называю его `plans`.

```
babenkoka@LAPTOP-6USQFP0J:~$ cp /usr/include/aio.h ~/equipment
babenkoka@LAPTOP-6USQFP0J:~$ mkdir ski.plases
babenkoka@LAPTOP-6USQFP0J:~$ mv ~/equipment ~/ski.plases/equipment
babenkoka@LAPTOP-6USQFP0J:~$ mv ~/ski.plases/equipment ~/ski.plases/equiplist
babenkoka@LAPTOP-6USQFP0J:~$ touch abc1
babenkoka@LAPTOP-6USQFP0J:~$ cp ~/abc1 ~/ski.plases/equiplist2
babenkoka@LAPTOP-6USQFP0J:~$ mkdir ~/ski.plases/equipment
babenkoka@LAPTOP-6USQFP0J:~$ mv ~/ski.plases/equiplist ~/ski.plases/equipment/equiplist
babenkoka@LAPTOP-6USQFP0J:~$ mv ~/ski.plases/equiplist2 ~/ski.plases/equipment/equiplist2
babenkoka@LAPTOP-6USQFP0J:~$ mkdir ~/newdir
babenkoka@LAPTOP-6USQFP0J:~$ mv newdir ~/ski.plases/plans
babenkoka@LAPTOP-6USQFP0J:~$
```

Fig 1 - Выполняю действия

Определяю опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет.

```
abenkoka@LAPTOP-6USQFP0J:~$ chmod u+r australia
abenkoka@LAPTOP-6USQFP0J:~$ chmod g+w australia
abenkoka@LAPTOP-6USQFP0J:~$ chmod o+x australia
abenkoka@LAPTOP-6USQFP0J:~$ chmod u+r play
abenkoka@LAPTOP-6USQFP0J:~$ chmod g+w play
abenkoka@LAPTOP-6USQFP0J:~$ chmod o+x play
abenkoka@LAPTOP-6USQFP0J:~$ chmod u+r my_os
abenkoka@LAPTOP-6USQFP0J:~$ chmod g-x my_os
abenkoka@LAPTOP-6USQFP0J:~$ chmod o+x my_os
abenkoka@LAPTOP-6USQFP0J:~$ chmod u+r feathers
abenkoka@LAPTOP-6USQFP0J:~$ chmod g+w feathers
abenkoka@LAPTOP-6USQFP0J:~$ chmod o-x feathers
```

*Fig 2 - Выполняю действия*

Смотрю содержимое файла `/etc/passwd`.

```
babenkoka@LAPTOP-6USQFP0J:~$ 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
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/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
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105:/:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111:/:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:/nonexistent:/usr/sbin/nologin
uidd:x:106:112:/:/run/uidd:/usr/sbin/nologin
tcpdump:x:107:113:/:/nonexistent:/usr/sbin/nologin
babenkoka:x:1000:1000:,,,:/home/babenkoka:/bin/bash
rtkit:x:108:118:RealtimeKit,,,:/proc:/usr/sbin/nologin
```

*Fig 3 - Содержимое файла*

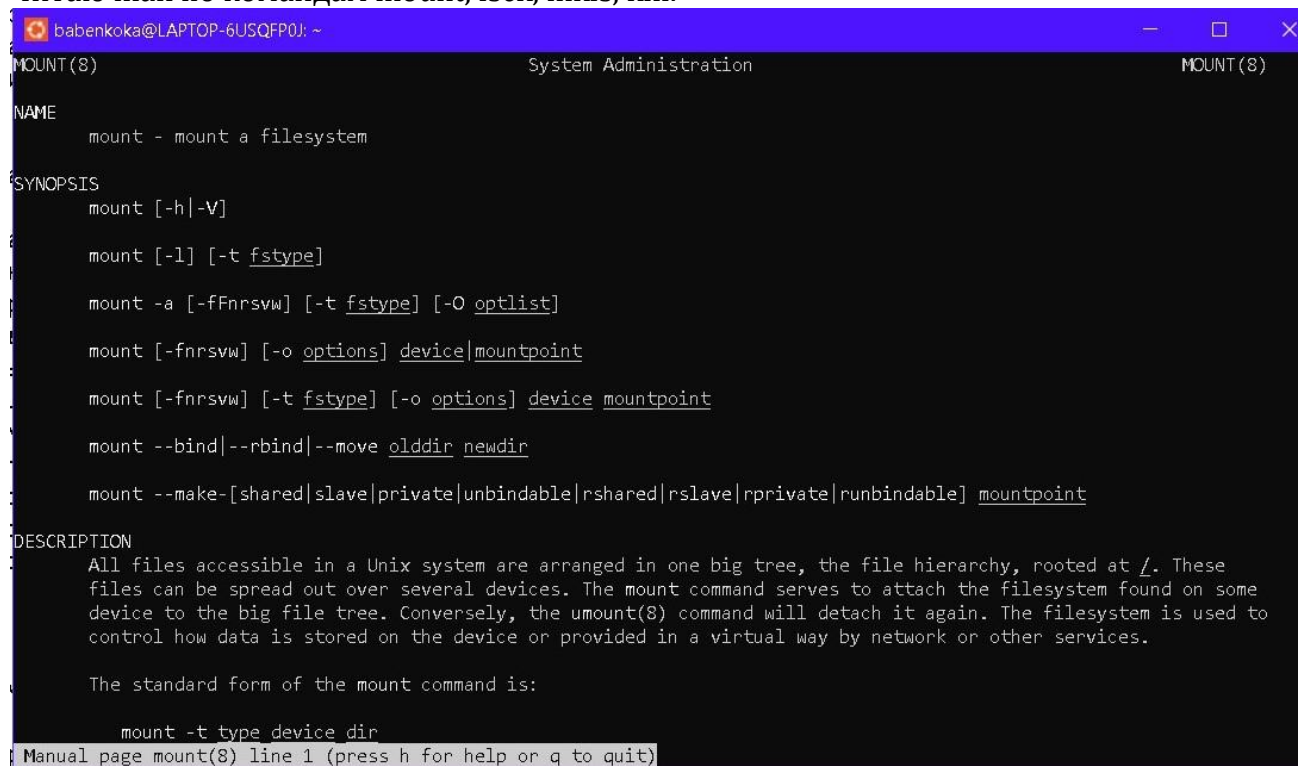
Копирую файл `~/feathers` в файл `~/file.old`. Перемещаю файл `~/file.old` в каталог `~/play`. Копирую каталог `~/play` в каталог `~/fun`. Перемещаю каталог `~/fun` в каталог `~/play` и называю его `games`. Лишаю владельца файла `~/feathers` права на чтение. Я не могу просмотреть файл `~/feathers` командой `cat`, скопировать файл `~/feathers`. Даю владельцу файла `~/feathers` право на чтение. Лишаю владельца каталога `~/play` права на выполнение. Я не могу перейти в каталог `~/play`. Даю владельцу каталога `~/play`

право на выполнение.

```
babenkoka@LAPTOP-6USQFP0J:~$ cp ~/feathers ~/file.old
babenkoka@LAPTOP-6USQFP0J:~$ mv ~/file.old ~/play/file.old
babenkoka@LAPTOP-6USQFP0J:~$ cp -r ~/play ~/fun
babenkoka@LAPTOP-6USQFP0J:~$ mv ~/fun ~/play/games
babenkoka@LAPTOP-6USQFP0J:~$ chmod u-r feathers
babenkoka@LAPTOP-6USQFP0J:~$ cat ~/feathers
cat: /home/babenkoka/feathers: Permission denied
babenkoka@LAPTOP-6USQFP0J:~$ cp ~/feathers ~/l
cp: cannot open '/home/babenkoka/feathers' for reading: Permission denied
babenkoka@LAPTOP-6USQFP0J:~$ chmod u+r feathers
babenkoka@LAPTOP-6USQFP0J:~$ chmod u-x play
babenkoka@LAPTOP-6USQFP0J:~$ cd ~/play
-bash: cd: /home/babenkoka/play: Permission denied
babenkoka@LAPTOP-6USQFP0J:~$ chmod u+x play
```

Fig 4 - Выполняю следующие действия

Читаю ман по командам mount, fsck, mkfs, kill.



```
babenkoka@LAPTOP-6USQFP0J: ~
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 /. 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 or other services.

    The standard form of the mount command is:

        mount -t type device dir

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

Fig 5 - cd

```
babenkoka@LAPTOP-6USQFP0J: ~
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

    1
        Filesystem errors corrected

    2
        System should be rebooted

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

Fig 6 - pwd

```
babenkoka@LAPTOP-6USQFP0J: ~
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.

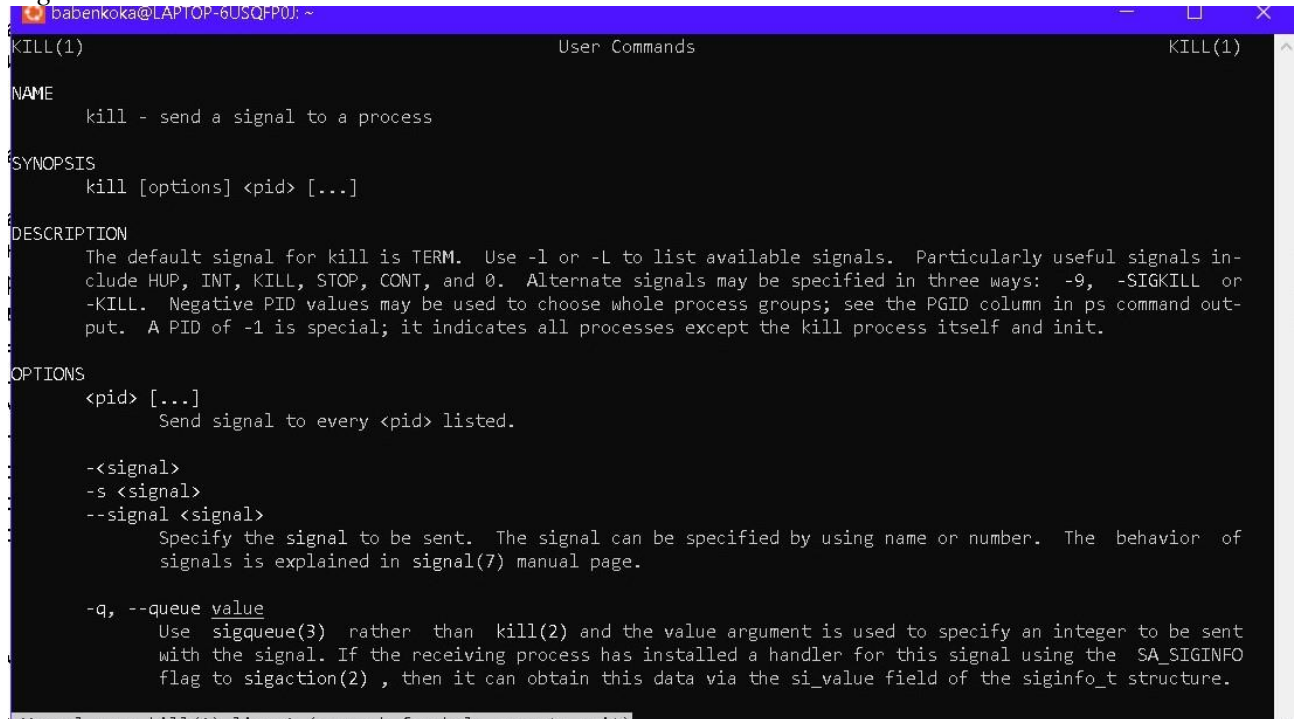
    In actuality, mkfs is simply a front-end for the various filesystem builders (mkfs.fstype) available under
    Linux. The filesystem-specific builder is searched for via your PATH environment setting only. Please see the
    filesystem-specific builder manual pages for further details.

OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2)
        is used.

    fs-options
        Filesystem-specific options to be passed to the real filesystem builder.

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

Fig 7 - mkdir



```
babenkoka@LAPTOP-6USQFP0J: ~
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.

    -q, --queue value
        Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer to be sent with the signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(2), then it can obtain this data via the si_value field of the siginfo_t structure.
```

Fig 8 - rmdir

### 3 Выводы

Я 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.