

**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: Карвецкий Всеволод Анатольевич

Группа: НКНбд-01-20

**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.

## Задание

Изучить основные команды для работы с файловой системой Linux, её структурой, именами и содержанием каталогов

## Выполнение лабораторной работы

1. Выполнил все примеры
2.
  1. У меня не было каталога sys, поэтому взял файл из другого каталога(см. 1.1)

```
[vakarvetskiy@10 ~]$ cd usr/include
[vakarvetskiy@10 include]$ ls
gnome-boxes  kde4  KF5  python3.6m  X11
[vakarvetskiy@10 include]$ cd gnome-boxes
[vakarvetskiy@10 gnome-boxes]$ ls
govf  gtk-frdp
[vakarvetskiy@10 gnome-boxes]$ cd govf
[vakarvetskiy@10 govf]$ ls
govf-disk.h  govf.h  govf-package.h
[vakarvetskiy@10 govf]$ sudo govf.h /equipment.h
sudo: govf.h: command not found
[vakarvetskiy@10 govf]$ sudo cp govf.h /equipment.h
[vakarvetskiy@10 govf]$ █
```

2. Создал директорию ski.plases(см. 1.2)

```
[vakarvetskiy@10 ~]$ cd usr/include
[vakarvetskiy@10 include]$ ls
gnome-boxes  kde4  KF5  python3.6m  X11
[vakarvetskiy@10 include]$ cd gnome-boxes
[vakarvetskiy@10 gnome-boxes]$ ls
govf  gtk-frrdp
[vakarvetskiy@10 gnome-boxes]$ cd govf
[vakarvetskiy@10 govf]$ ls
govf-disk.h  govf.h  govf-package.h
[vakarvetskiy@10 govf]$ sudo govf.h /equipment.h
sudo: govf.h: command not found
[vakarvetskiy@10 govf]$ sudo cp govf.h /equipment.h
[vakarvetskiy@10 govf]$ cd /
[vakarvetskiy@10 ~]$ cd ~
[vakarvetskiy@10 ~]$ mkdir ski.plases
[vakarvetskiy@10 ~]$ ls
Desktop  Downloads  Music  Pictures  ski.plases  Videos
Documents  equipment.h  newdir  Public  templates
```

3. Переместил файл equipment в каталог ski.plases(см. 1.3)

```
Documents  equipment.h  newdir  Public  Templates
[vakarvetskiy@10 ~]$ sudo mv /equipment.h ~/ski.plases/equipment.h
```

4. Переименовал файл(см. 1.4)

```
[vakarvetskiy@10 ~]$ mv ski.plases/equipment.h ski.plases/equiplist
[vakarvetskiy@10 ~]$
```

5. Создал файл abc1 и скопировал его в каталог ski.plases, назвав его equiplist2(см. 1.5)

```
[vakarvetskiy@10 ~]$ touch abc1; cp abc1 ski.plases/equiplist2
[vakarvetskiy@10 ~]$
```

6. Создал каталог equipment в каталоге ski.plases(см. 1.6)

```
[vakarvetskiy@10 ~]$ mkdir ski.plases/equipment
[vakarvetskiy@10 ~]$
```

7. Переместил файлы equiplist и equiplist2 в каталог equipment(см. 1.7)

```
[vakarvetskiy@10 ~]$ cd ski.plases
[vakarvetskiy@10 ski.plases]$ mv equiplist equiplist2 equipment
[vakarvetskiy@10 ski.plases]$ ls
equipment
[vakarvetskiy@10 ski.plases]$
```

8. Создал и переместил каталог newdir в каталог ski.plases и назвал его plans(см. 1.8)

```
[vakarvetskiy@10 ~]$ mkdir newdir; mv newdir ski.places/plans
mkdir: cannot create directory 'newdir': File exists
[vakarvetskiy@10 ~]$ rm -r newdir
rm: cannot remove 'newdir': No such file or directory
[vakarvetskiy@10 ~]$ rmdir newdir
rmdir: failed to remove 'newdir': No such file or directory
[vakarvetskiy@10 ~]$ ls
abc1      Documents  equipment.h  Pictures  ski.places  Videos
Desktop  Downloads  Music        Public    Templates
[vakarvetskiy@10 ~]$ cd ski.places
[vakarvetskiy@10 ski.places]$ ls
equipment  plans
[vakarvetskiy@10 ski.places]$
```

3. Определил опции команды `chmod`, необходимые для присвоения файлам и каталогам необходимых прав доступа. Создал файлы(см. 3.1)

```
[vakarvetskiy@10 ~]$ mkdir australia; chmod 744 australia
[vakarvetskiy@10 ~]$ mkdir play; chmod 711 play
[vakarvetskiy@10 ~]$ touch my_os; chmod 544 my_os
[vakarvetskiy@10 ~]$ touch feathers; chmod 664 feathers
[vakarvetskiy@10 ~]$
```

- 4.

1. Просмотрел содержимое файла `etc/passwd`(см. 4.1)

```
[vakarvetskiy@10 etc]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
geoclue:x:997:995:User for geoclue:/var/lib/geoclue:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
```

2. Скопировал файл `feathers` в файл `file.old`(см. 4.2)

```
[vakarvetskiy@10 ~]$ cp feathers file.old
[vakarvetskiy@10 ~]$
```

3. Переместил `file.old` в каталог `play`(см. 4.3)

```
[vakarvetskiy@10 ~]$ mv file.old play/file.old
[vakarvetskiy@10 ~]$
```

4. Скопировал каталог play в каталог fun(см. 4.4)  

```
[vakarvetskiy@10 ~]$ cp -r play fun  
[vakarvetskiy@10 ~]$
```
5. Переместил каталог fun в каталог play и назвал его games(см. 4.5)  

```
[vakarvetskiy@10 ~]$ mv fun play/games  
[vakarvetskiy@10 ~]$
```
6. Лишил владельца прав на чтение файла feathers(см. 4.6)  

```
[vakarvetskiy@10 ~]$ chmod u-r feathers  
[vakarvetskiy@10 ~]$
```
7. Файл не просматривается(см. 4.7)  

```
[vakarvetskiy@10 ~]$ cat feathers  
cat: feathers: Permission denied  
[vakarvetskiy@10 ~]$
```
8. Файл не копируется(см. 4.8)  

```
[vakarvetskiy@10 ~]$ cp feathers feathers1  
cp: cannot open 'feathers' for reading: Permission denied  
[vakarvetskiy@10 ~]$
```
9. Дал владельцу право на чтение(см. 4.9)  

```
[vakarvetskiy@10 ~]$ chmod u+r feathers  
[vakarvetskiy@10 ~]$
```
10. Лишил владельца права на выполнение каталога play(см. 4.10)  

```
[vakarvetskiy@10 ~]$ chmod u-x play  
[vakarvetskiy@10 ~]$
```
11. Доступ в каталог запрещен(см. 4.11)  

```
[vakarvetskiy@10 ~]$ cd play  
bash: cd: play: Permission denied  
[vakarvetskiy@10 ~]$
```
12. Дал владельцу право на выполнение(см. 4.12)  

```
[vakarvetskiy@10 ~]$ chmod u+x play  
[vakarvetskiy@10 ~]$
```

5. Прочитал ман по командам:

1. **mount** - монтирует файловую систему(см. [mount](#))

MOUNT(8)	System Administration	MOUNT(8)
<b>NAME</b> mount - mount a filesystem		
<b>SYNOPSIS</b> mount [-l -h -V]  mount -a [-fFnrsvw] [-t <u>fstype</u> ] [-O <u>optlist</u> ]  mount [-fnrsvw] [-o <u>options</u> ] <u>device</u>   <u>dir</u>  mount [-fnrsvw] [-t <u>fstype</u> ] [-o <u>options</u> ] <u>device</u> <u>dir</u>		
<b>DESCRIPTION</b> 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 <b>mount</b> command serves to attach the filesystem found on some device to the big file tree. Conversely, the <b>umount(8)</b> 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 another services.		
Manual page mount(8) line 1 (press h for help or q to quit)		

2. **fsck** - проверка файловой системы на целостность(см. [fsck](#))

FSCK(8)	System Administration	FSCK(8)
<b>NAME</b> fsck - check and repair a Linux filesystem		
<b>SYNOPSIS</b> fsck [-lsAVRTMNP] [-r [ <u>fd</u> ]] [-C [ <u>fd</u> ]] [-t <u>fstype</u> ] [ <u>filesystem...</u> ] [--] [ <u>fs-specific-options</u> ]		
<b>DESCRIPTION</b> <b>fsck</b> is used to check and optionally repair one or more Linux filesystems. <u>filesystems</u> can be a device name (e.g. <u>/dev/hdc1</u> , <u>/dev/sdb2</u> ), a mount point (e.g. <u>/</u> , <u>/usr</u> , <u>/home</u> ), or an ext2 label or UUID specifier (e.g. UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the <b>fsck</b> 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 <b>-A</b> option is not specified, <b>fsck</b> will default to checking filesystems in <u>/etc/fstab</u> serially. This is equivalent to the <b>-As</b> options.  The exit code returned by <b>fsck</b> is the sum of the following conditions:		

### 3. mkfs - создание новой файловой системы(см. [mkfs](#))

MKFS(8)	System Administration	MKFS(8)
<b>NAME</b>		
mkfs - build a Linux filesystem		
<b>SYNOPSIS</b>		
mkfs [options] [-t <u>type</u> ] [ <u>fs-options</u> ] <u>device</u> [ <u>size</u> ]		
<b>DESCRIPTION</b>		
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 <u>device</u> argument is either the device name (e.g. /dev/hda1, /dev/sdb2), or a regular file that shall contain the filesystem. The <u>size</u> argument is the number of blocks to be used for the filesystem.		
The exit code 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		

### 4. kill - убивает фоновой процесс(см. [kill](#))

KILL(1)	User Commands	KILL(1)
<b>NAME</b>		
kill - terminate a process		
<b>SYNOPSIS</b>		
kill [- <u>signal</u>  -s <u>signal</u>  -p] [-q <u>value</u> ] [-a] [--] <u>pid</u>   <u>name</u> ...		
kill -l [ <u>number</u> ]   -L		
<b>DESCRIPTION</b>		
The command kill sends the specified <u>signal</u> 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.		

## Выводы

Я научился использовать основные команды для работы с файловой системой Linux, её структурой, именами и содержанием каталогов