

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

«ОСНОВЫ ИНТЕРФЕЙСА КОМАНДНОЙ СТРОКИ ОС GNU LINUX»

дисциплина: Архитектура компьютера

Студент: Осокин Г.И

Группа: НММбд-02-22

МОСКВА

2022 г.

Цели и Задачи

- Приобрести практические навыки работы с операционной системой на уровне командной строки
- Научиться взаимодействовать с файловой системой UNIX-подобных систем
- Освоить базовые команды командной оболочки bash

Создание и переключение пользователя

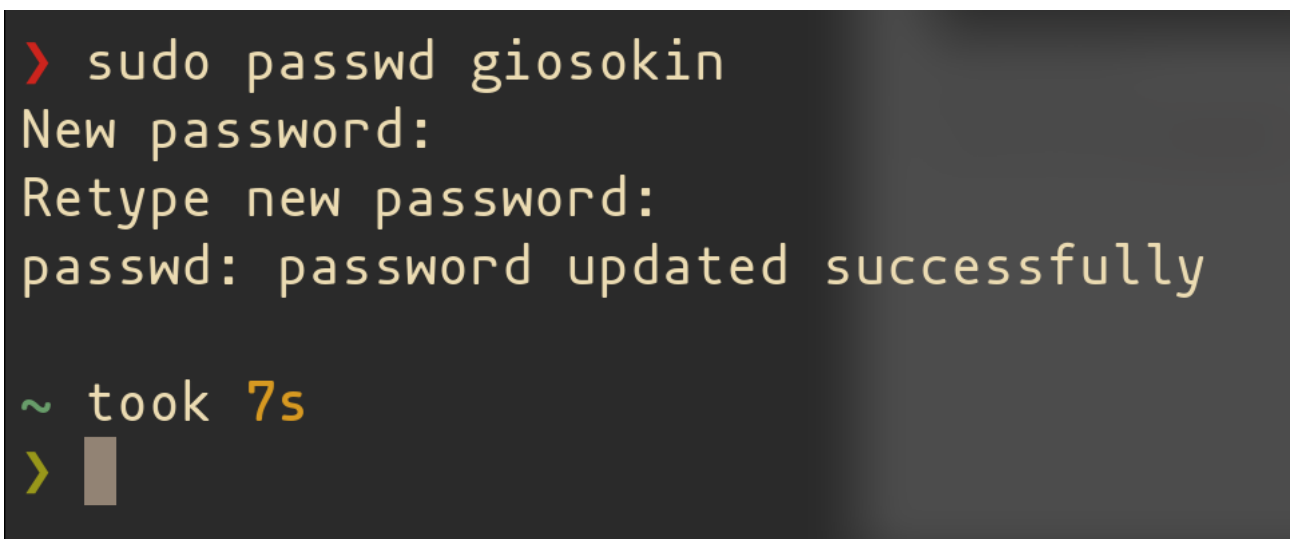
1. Создадим нового пользователя **giosokin** командой

A terminal window with a dark background. A green tilde (~) indicates the current directory is the user's home. A yellow prompt character (>) is followed by the command 'sudo useradd giosokin' in yellow text.

```
~  
> sudo useradd giosokin
```

Скриншот 1: Создание нового пользователя

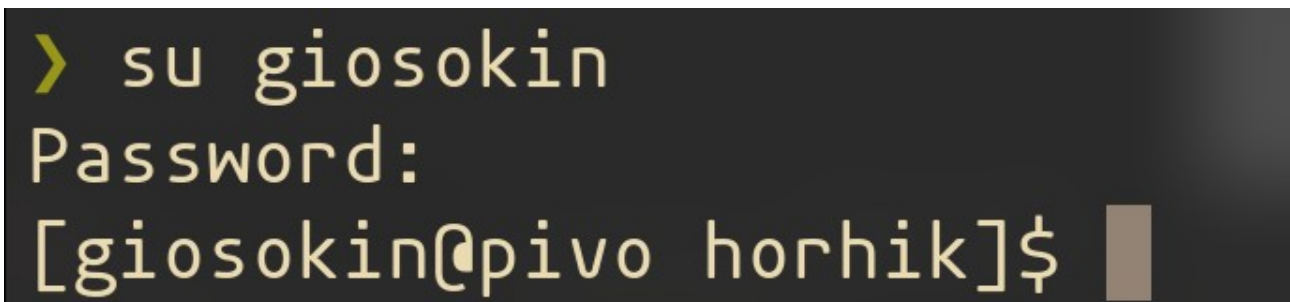
2. Зададим пользователю пароль

A terminal window with a dark background. A red prompt character (>) is followed by the command 'sudo passwd giosokin' in yellow text. The output shows 'New password:', 'Retype new password:', and 'passwd: password updated successfully' in yellow text. Below this, a green tilde (~) is followed by 'took 7s' in yellow text. A yellow prompt character (>) is followed by a grey rectangular cursor.

```
> sudo passwd giosokin  
New password:  
Retype new password:  
passwd: password updated successfully  
  
~ took 7s  
> █
```

Скриншот 2: Задание пароля для пользователя *giosokin*

3. Переключимся на пользователя **giosokin** командой

A terminal window with a dark background. A yellow prompt character (>) is followed by the command 'su giosokin' in yellow text. The output shows 'Password:' in yellow text. Below this, the prompt '[giosokin@pivo horhik]\$' is shown in yellow text, followed by a grey rectangular cursor.

```
> su giosokin  
Password:  
[giosokin@pivo horhik]$ █
```

Скриншот 3: Переключение пользователя

Выполнение лабораторной работы

1. Откроем терминал (в моем случае это консольный эмулятор **kitty**) и перейдем в домашнюю директорию нашего пользователя **giosokin**, командой **cd**

```
[giosokin:~]$ cd
[giosokin:~]$
```

Скриншот 4: Команда **cd**

2. Узнаем путь к нашему домашнему каталогу командой **pwd**

```
[giosokin:~]$ pwd
/home/giosokin
[giosokin:~]$
```

Скриншот 5: команда **pwd**

3. Попробуем зайти в папку **Documents**, которая находится в домашней директории. Так как мы уже находимся в ней, мы можем просто написать **cd Documents** или же **cd ./Documents**

```
[giosokin:~]$ cd Documents
bash: cd: Documents: No such file or directory
```

Скриншот 6: Ошибка при переходе в директорию **Documents**

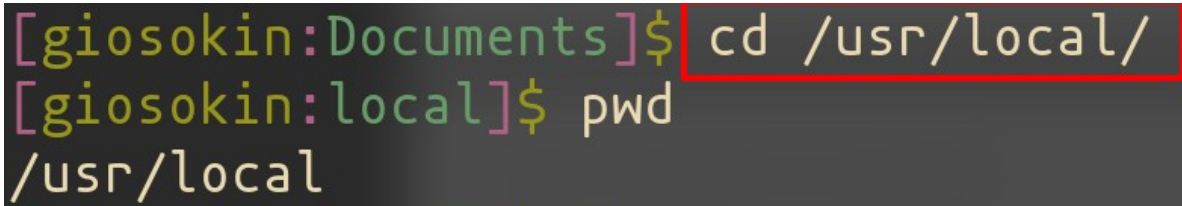
Наблюдаем, что такой директории не существует. Создадим ее командой **mkdir Documents**

```
[giosokin:~]$ mkdir Documents
[giosokin:~]$ cd Documents
[giosokin:Documents]$
```

Скриншот 7: Создание папки **Documents**

Теперь мы можем перейти в созданный каталог **Documents**

4. Перейдем в подкаталог **local** корневого каталога **usr** (**/usr/local**)



```
[giosokin:Documents]$ cd /usr/local/  
[giosokin:local]$ pwd  
/usr/local
```

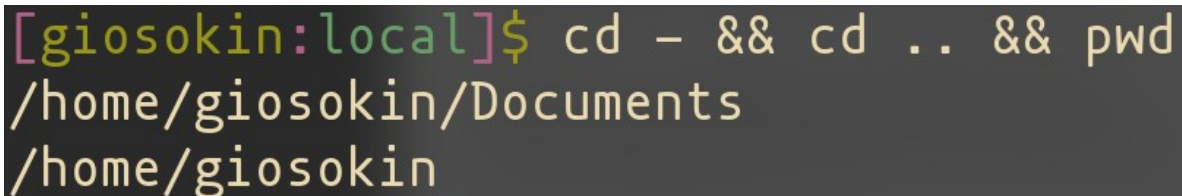
Скриншот 8: Переход в каталог **/usr/local**

Теперь мы находимся в директории **/usr/local**

Команда **cd** может работать и с относительными и с абсолютными файлами (как на скриншоте ... и ...)

5. Выполним последовательность команд **cd -** и **cd ..** , первая возвращает в последний посещенный каталог, а вторая поднимается по иерархии каталогов на один выше.

Введем в консоли команду **cd -&& cd .. && pwd** (**pwd** что бы сразу увидеть, абсолютный путь к директории, в которую мы попали)



```
[giosokin:local]$ cd - && cd .. && pwd  
/home/giosokin/Documents  
/home/giosokin
```

Скриншот 9: Использование различных опций команды **cd**

Мы переместились в домашнюю директорию. Т.к. ранее мы находились в **~/Documents** и еще поднялись выше по иерархии

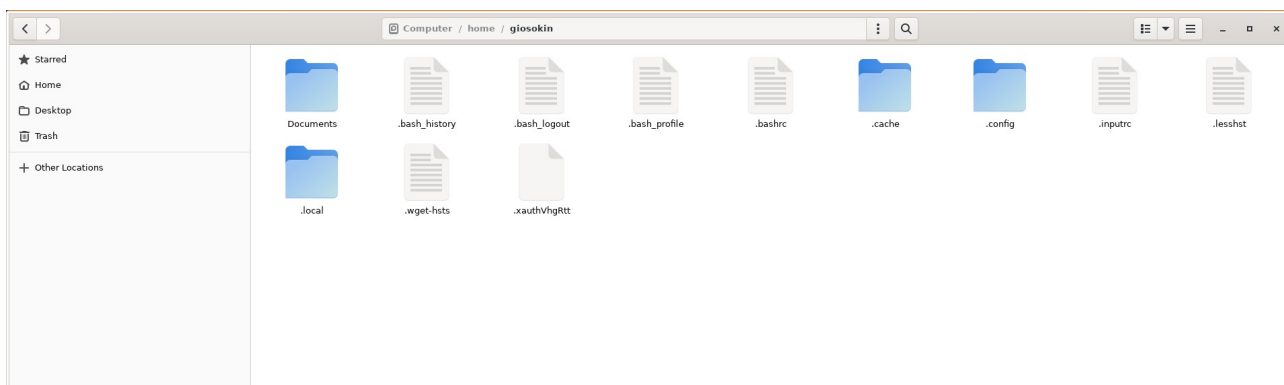
6. Выведем содержимое домашней директории, командой **ls**

```
[giosokin:~]$ ls
Documents
```

Скриншот 10: Команда *ls*

В **~** находится только **Documents**, потому что я создавал пользователя через команду **useradd** и использовал ее без каких-либо дополнительных аргументов. Поэтому в домашней директории и не создались дополнительные каталоги, такие как **Documents**, **Downloads**, **Pictures**, **е.т.с...**

7. Откроем файловый менеджер **nautilus** и посмотрим содержимое домашней директории через него.



Скриншот 11: Файловый менеджер *Nautilus*

Nautilus показывает намного больше файлов и папок, чем команда **ls** в терминале. Это из-за того, что в настройках включено отображение скрытых файлов и папок.

8. Мы можем вывести их в консоль, командой `ls -a` (all)

```
[giosokin:~]$ ls -a
.          .bash_profile  .inputrc    .xauthVhgRtt
..         .bashrc        .lessht     Documents
.bash_history .cache        .local
.bash_logout .config       .wget-hsts
```

Скриншот 12: Вывод ls -a

Теперь можем убедиться, что и в терминале и в файловом менеджере все отображается одинаково

9. Воспользуемся командой **ls -R** что бы рекурсивно вывести все содержимое директории

```
[giosokin:~]$ ls -Ra
.:
.  .bash_profile  .inputrc  .xauthVhgRtt
.. .bashrc       .lessht   Documents
.bash_history  .cache    .local
.bash_logout   .config   .wget-hsts

./cache:
.  .. nvim  starship

./cache/nvim:
.  .. log

./cache/starship:
.  ..

./config:
.  .. starship.toml

./local:
.  .. share
```

Скриншот 13: Вывод ls -R

Я ввел **ls -Ra** что бы также учесть и скрытые файлы, потому что не скрытой является только директория **Documents** и будет не наглядно.

10 . Можем посмотреть подробную информацию командой `ls -l` (так же введем `ls -la`)

```
[giosokin:~]$ ls -la
total 32
drwx----- 1 giosokin giosokin 222 Sep 21 13:48 .
drwxr-xr-x 1 root      root      46 Sep 20 06:41 ..
-rw----- 1 giosokin giosokin 2727 Sep 20 12:07 .bash_history
-rw-r--r-- 1 giosokin giosokin  18 May  3 01:20 .bash_logout
-rw-r--r-- 1 giosokin giosokin  89 May  3 01:20 .bash_profile
-rw-r--r-- 1 giosokin giosokin 248 Sep 20 07:39 .bashrc
drwx----- 1 giosokin giosokin  24 Sep 20 07:02 .cache
drwxrwxr-x 1 giosokin giosokin  26 Sep 20 07:02 .config
-rw-r--r-- 1 giosokin giosokin  22 May  3 01:20 .inputrc
-rw----- 1 giosokin giosokin 133 Sep 20 07:50 .lessht
drwx----- 1 giosokin giosokin  10 Sep 20 06:55 .local
-rw-rw-r-- 1 giosokin giosokin 180 Sep 20 07:30 .wget-hsts
-rw----- 1 giosokin giosokin  49 Sep 21 13:41 .xauthVhgRtt
drwxrwxr-x 1 giosokin giosokin   0 Sep 21 13:48 Documents
[giosokin:~]$
```

Скриншот 14: вывод `ls -la`

11. Введем `ls -lah` и посмотрим, в чем разница с `ls -la`

```
[giosokin:~]$ ls -la
total 32
drwx----- 1 giosokin giosokin 222 Sep 21 13:48 .
drwxr-xr-x 1 root     root     46 Sep 20 06:41 ..
-rw----- 1 giosokin giosokin 2727 Sep 20 12:07 .bash_history
-rw-r--r-- 1 giosokin giosokin 18 May 3 01:20 .bash_logout
-rw-r--r-- 1 giosokin giosokin 89 May 3 01:20 .bash_profile
-rw-r--r-- 1 giosokin giosokin 248 Sep 20 07:39 .bashrc
drwx----- 1 giosokin giosokin 24 Sep 20 07:02 .cache
drwxrwxr-x 1 giosokin giosokin 26 Sep 20 07:02 .config
-rw-r--r-- 1 giosokin giosokin 22 May 3 01:20 .inputrc
-rw----- 1 giosokin giosokin 133 Sep 20 07:50 .lessht
drwx----- 1 giosokin giosokin 10 Sep 20 06:55 .local
-rw-rw-r-- 1 giosokin giosokin 180 Sep 20 07:30 .wget-hsts
-rw----- 1 giosokin giosokin 49 Sep 21 13:41 .xauthVhgRtt
drwxrwxr-x 1 giosokin giosokin 0 Sep 21 13:48 Documents
[giosokin:~]$ ls -lah
total 32K
drwx----- 1 giosokin giosokin 222 Sep 21 13:48 .
drwxr-xr-x 1 root     root     46 Sep 20 06:41 ..
-rw----- 1 giosokin giosokin 2.7K Sep 20 12:07 .bash_history
-rw-r--r-- 1 giosokin giosokin 18 May 3 01:20 .bash_logout
-rw-r--r-- 1 giosokin giosokin 89 May 3 01:20 .bash_profile
-rw-r--r-- 1 giosokin giosokin 248 Sep 20 07:39 .bashrc
drwx----- 1 giosokin giosokin 24 Sep 20 07:02 .cache
drwxrwxr-x 1 giosokin giosokin 26 Sep 20 07:02 .config
-rw-r--r-- 1 giosokin giosokin 22 May 3 01:20 .inputrc
-rw----- 1 giosokin giosokin 133 Sep 20 07:50 .lessht
drwx----- 1 giosokin giosokin 10 Sep 20 06:55 .local
```

Скриншот 15: Сравнение `ls -la` и `ls -lah`

Без аргумента `-h` размеры файлов были указаны в битах, а с `-h` размер указан удобнее (тут 2700 бит заменил 2.7 килобайт)

12. Создадим директорию `parentdir` с помощью команды `mkdir`

```
[giosokin:~]$ mkdir parentdir
[giosokin:~]$ ls
Documents parentdir
```

Скриншот 16: Команда `mkdir`

Убедимся что директория создана с помощью команды `ls`

13. Создадим директорию **dir** в каталоге **parentdir**

```
[giosokin:~]$ mkdir parentdir/dir
```

Скриншот 17: создание поддиректории *parentdir*

14. Перейдем в **parentdir** и создадим три папки

```
[giosokin:~]$ cd parentdir/  
[giosokin:parentdir]$ mkdir dir1 dir2 dir3  
[giosokin:parentdir]$ ls  
dir dir1 dir2 dir3
```

Скриншот 18: Создание нескольких папок командой *mkdir*

15. Создадим папку в домашней директории указывая не относительный путь

```
[giosokin:parentdir]$ mkdir ~/newdir  
[giosokin:parentdir]$ ls ..  
Documents newdir parentdir
```

Скриншот 19: Команда *mkdir*

16. Используем аргумент **-p** что бы создать иерархическую структуру

```
[giosokin:parentdir]$ mkdir -p ~/tree/branch/branch/branch/branch  
[giosokin:parentdir]$
```

Скриншот 20: Опция *-p* для создание вложенных директорий

17. Установим (если она не установлена) и воспользуемся утилитой **tree** что бы наглядно посмотреть как вложены директории.

```
[giosokin:parentdir]$ sudo xbps-install tree  
Password:  
Package `tree' already installed.
```

Скриншот 21: Установка пакета *tree*

```
[giosokin:parentdir]$ tree ~/tree/
/home/giosokin/tree/
├── branch
│   ├── branch
│   │   └── branch
│   └── branch
└──
```

4 directories, 0 files

Скриншот 22: Вывод команды tree

18. Создадим файл **leaf** в директории
~/tree/branch/branch/branch/branch

```
[giosokin:parentdir]$ touch ~/tree/branch/branch/branch/branch/leaf
[giosokin:parentdir]$ tree ~/tree/
/home/giosokin/tree/
├── branch
│   ├── branch
│   │   ├── branch
│   │   └── leaf
└──
```

4 directories, 1 file

Скриншот 23: Создание пустого файла командой touch

Как видим, файл создан.

19. Выполним команду **rm -riv ~/parentdir** что бы рекурсивно удалить содержимое директории и при этом выводить подробную информацию и спрашивать подтверждение при удалении каждого файла

```
[giosokin:~]$ rm -riv ~/parentdir/
rm: descend into directory '/home/giosokin/parentdir/'? y
rm: remove directory '/home/giosokin/parentdir/dir'? y
removed directory '/home/giosokin/parentdir/dir'
rm: remove directory '/home/giosokin/parentdir/dir1'? y
removed directory '/home/giosokin/parentdir/dir1'
rm: remove directory '/home/giosokin/parentdir/dir2'? y
removed directory '/home/giosokin/parentdir/dir2'
rm: remove directory '/home/giosokin/parentdir/dir3'? y
removed directory '/home/giosokin/parentdir/dir3'
rm: remove directory '/home/giosokin/parentdir/'? y
removed directory '/home/giosokin/parentdir/'
[giosokin:~]$ ls ~/parentdir
ls: cannot access '/home/giosokin/parentdir': No such file or directory
```

Скриншот 24: Удаление командой rm -riv

Как видим, директории **parentdir** теперь не существует

20. Теперь удалим директорию `tree` командой `rm -rf tree`, опция `-f` принудительно удалит все ее содержимое и не будет спрашивать нашего подтверждения

```
[giosokin:~]$ rm -rf tree/  
[giosokin:~]$ ls tree  
ls: cannot access 'tree': No such file or directory
```

Скриншот 25: Удаление командой `rm -rf`

21. Создадим файл `to_move`

```
[giosokin:~]$ touch to_move  
[giosokin:~]$
```

Скриншот 26: Создание файла командой `touch`

Мы можем использовать команду `mv` что бы «переименовать» файл (т. е. Просто переместить его в несуществующий файл с другим названием)

```
[giosokin:~]$ mv to_move new_name  
[giosokin:~]$ ls  
Documents new_name newdir
```

Скриншот 27: Переименование файла командой `mv`

22. Создадим два файла с разным содержимым и попробуем переместить один в другой, добавив опцию `-v` (verbose)

```
[giosokin:~]$ mv -v letter-1 letter-2  
renamed 'letter-1' -> 'letter-2'
```

Скриншот 28: Вывод `mv` с опцией `-v`

По сути мы переименовали **letter-1** в **letter-2**, утилита нам и сообщила об этом.

23 . Заново создадим файл **letter-1** и переместим его командой **mv -vi letter-2**

```
[giosokin:~]$ echo "word1" > letter-1
[giosokin:~]$ mv -vi letter-1 letter-2
mv: overwrite 'letter-2'? y
renamed 'letter-1' -> 'letter-2'
```

Скриншот 29: Команда *echo*

24 . Создадим директорию **book** и в ней файл **page-1**

```
[giosokin:~]$ mkdir book
[giosokin:~]$ touch book/page-1
[giosokin:~]$
```

Скриншот 30: Команда *mkdir book*

25 . Добавим в файл **page-1** содержимое .

```
[giosokin:~]$ echo "first chapter" > book/page-1
[giosokin:~]$ cat book/page-1
first chapter
```

Скриншот 31: Команда *echo*

26 . Зайдем в каталог **book** и перекопируем этот файл командой **cp**

```
[giosokin:book]$ cp page-1 page-2
[giosokin:book]$ ls
page-1  page-2
[giosokin:book]$ cat *
first chapter
first chapter
```

Скриншот 32: Копирование командой *cp*

27 . Просмотрим содержимое всех файлов находящихся в **book**
Как видим , содержание одинаковое , файл **page-2** копия **page-1**
скопируем папку **~/newdir** в папку **~/newestdir**

```
[giosokin:~]$ cp -r ~/newdir/ ~/newestdir  
[giosokin:~]$ ls  
Documents book letter-2 new_name newdir newestdir text
```

Скриншот 33: Рекурсивное копирование с опцией **-r**

Мы использовали опцию **-r** и поэтому папка рекурсивно перекопировалась

28 . Выведем содержимое файла **/etc/hosts** командой **cat**

```
[giosokin:RUDN]$ cat /etc/hosts  
#  
# /etc/hosts: static lookup table for host names  
#  
#<ip-address>          <hostname.domain.org>  <hostname>  
127.0.0.1               localhost.localdomain  localhost  
::1                     localhost.localdomain  localhost ip6-localhost  
# End of file
```

Скриншот 34: Результат вывода команды **cat /etc/hosts**

Задания для самостоятельной работы

1. Узнаем путь к домашней директории. Зайдем в нее и исполним команду `pwd`



```
[giosokin:~]$ cd  
[giosokin:~]$ pwd  
/home/giosokin
```

Скриншот 35: команда `pwd`

2. Выполним набор команд :



```
[giosokin:~]$ cd  
[giosokin:~]$ mkdir tmp  
[giosokin:~]$ cd tmp  
[giosokin:tmp]$ pwd  
/home/giosokin/tmp  
[giosokin:tmp]$ cd /tmp  
[giosokin:tmp]$ pwd  
/tmp  
[giosokin:tmp]$
```

Скриншот 36: Различный вывод у команд `pwd`

Команда `pwd` дает разные выводы, потому что, в первом случае, мы перешли в директорию `tmp` в домашнем каталоге, так как указывали относительный путь до нее и изначально находились в домашней директории. Во втором случае мы переходим в каталог `/tmp` лежащий в корневой директории, поэтому абсолютный путь, выводимый `pwd` различается

3. Просмотрим содержимое корневого каталога

```
[giosokin:tmp]$ ls /
bin      etc      lib32    media    proc     sbin     usr
boot     home     lib64    mnt      root     sys      var
dev      lib      lost+found  opt      run      tmp
```

Скриншот 37: Содержимое корневого каталога

Выводятся директории, лежащие в корне файловой системы

4. Просмотрим содержимое домашнего каталога

```
[giosokin:tmp]$ ls ~
```

Documents	labs	new_name	newestdir	text
book	letter-2	newdir	temp	tmp

Скриншот 38: Содержимое домашнего каталога

Видим директории, которые были созданы нами в ходе лабораторной работы

5. Выведем содержимое /etc

[@iosokin:tmp]\$ ls /etc/	bluetooth	environment	hosts	lvm	passwd	rpc	subgid	wgetrc
image4gicx-7	ca-certificates	fonttypes	inputrc	lvm	passwize	rsyncd.conf	subuid	wireguard
hostcloud	ca-certificates.conf	fonts	inputrc2	lynx.cfg	passwd	rsyslog.conf	subuid	wireguard.conf
lvmCL	colors	freerds	iptables	lynx.lss	passwd-w	rsyslog.d	subuid-sudo	wpa_supplicant
lvmPower	cron.daily	fstab	issue	man.conf	passwd.OLD	runit	sudo.conf	xattr.conf
X11	crypttab	fuse.conf	issue.new-0.142_11	mke2fs.conf	pkcs11	sane.d	sudoers	xkb.d
acpi	cups	gai.conf	julia	wdrprobe.d	pkcs11-1	securetty	sudoers.d	xorg
adjtime	dbus-1	geoclue	kernel.d	wmodules-load.d	profile	security	sudoers.dist	xinetd.d
alsa	default	gimp	ld.so.cache	apw	profile.d	services	sudoers.new-1.9.10_1	xm1
animator.d	depmod.d	gnucash	ld.so.conf	mtab	protocols	shadow	sudoers.new-1.9.11p3_1	zsh
arducpstream.conf	dhcpd.conf	group	lib	netconfig	rc.conf	shadow	sway	zshrc
arduc6	dillo	libao	libao.conf	ngio	rc.conf.new-20220329_1	shells	sysctl.conf	zzz.d
avahi	dkms	group.new-0.142_13	libltdldev	nix	rc.local	signon-ui	ts.conf	
avrdude.conf	dns	gsub.d	libinput	nsiswitch.conf	rc-shutdown	signond.conf	udev	
bash	docker	gshadow	libinput-gestures.conf	ntpd.conf	rc_maps.conf	skel	udisks2	
bash.bashrc	dracut.conf	gshadow-w	libl	odbc.ini	resolv.conf	s1sh.rc	speech-dispatcher	ufw
bash.bashrc.backup-before-nix	dracut.conf.d	gtk-2.0	libappser.d	odbcinst.ini	resolv.conf	ssh	ssh	udev
bashrc	e2scrub.conf	gtk-3.0	locale.conf	openbsd	resolv.conf	ssh	ssh	vpdpu_wrapper.cfg
bindresvport-blacklist	element-desktop	hostname	localtime	opensc.conf	rhashrc	ssl		
[@iosokin:tmp]\$ ll	elogind	hostname	login.defs	os-release	rofi-pass.conf			

Скриншот 39: Содержимое каталога /etc

6. Выведем содержимое `/usr/local`

```
[giosokin:tmp]$ ls /usr/local/  
bin include lib sbin share
```

Скриншот 40: Содержимое `/usr/local`

7. Создадим каталоги `temp` и `labs`, внутри каталога `labs` добавим папки `lab1`, `lab2`, `lab3`, а в `temp` `text1.txt`, `text2.txt`, `text3.txt`.

Так как эти директории были у меня созданы, для начала я их удалю.

```
[giosokin:~]$ rm -rf labs/ temp/  
[giosokin:~]$ mkdir labs temp  
[giosokin:~]$ cd labs  
[giosokin:labs]$ mkdir lab1 lab2 lab3  
[giosokin:labs]$ cd ..  
[giosokin:~]$ cd temp/  
[giosokin:temp]$ touch text{1,2,3}.txt  
[giosokin:temp]$ cd ..  
[giosokin:~]$ ls labs temp/  
labs:  
lab1 lab2 lab3  
  
temp/:  
text1.txt text2.txt text3.txt
```

Скриншот 41: Создание нескольких папок и нескольких файлов одной командой `mkdir` и `touch`

Папки были созданы командой `mkdir`, а файлы `touch`. Для того что бы не писать `text_.txt` три раза я воспользовался записью фигурными скобками.

8. Воспользуемся консольным текстовым редактором VIM и впишем имя, фамилию и группу в файлы `text1` `text2` и `text3`.

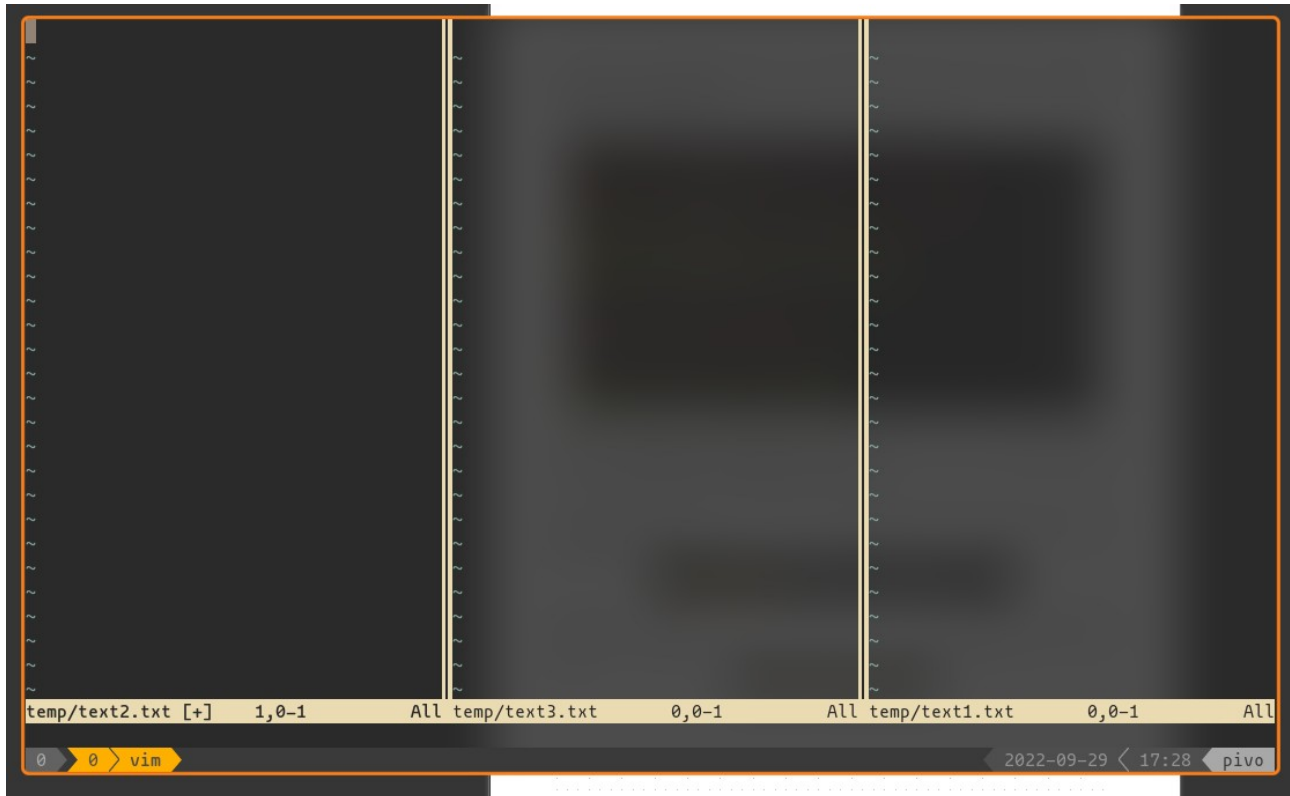
```
[giosokin:~]$ man vim  
[giosokin:~]$ vim temp/text1.txt
```

Скриншот 42: Вход в VIM

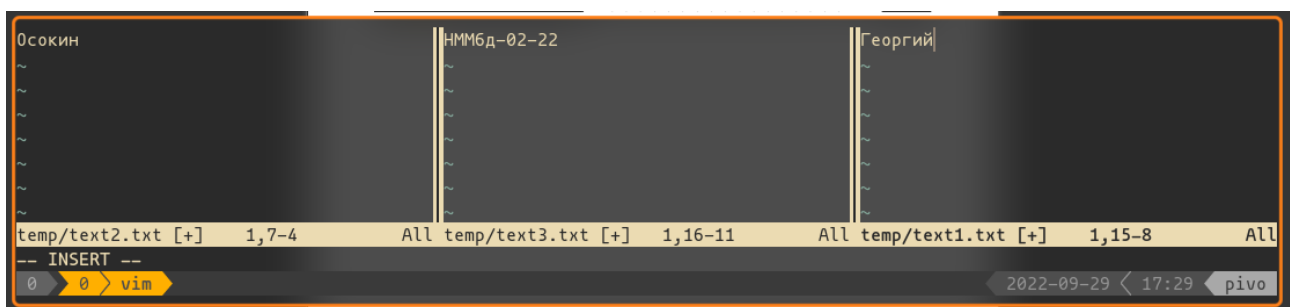
в vim введем `:vsplit temp/text2.txt` что б редактировать два файла параллельно . Аналогично для text3

```
temp/text2.txt  
:vsplit temp/text2.txt
```

Скриншот 43: Разделение экрана редактора для параллельного



Скриншот 44: Внешний вид текстового редактора VIM с тремя открытыми файлами
Отредактируем .



Скриншот 45: Измененные файлы

Напишем :wqа что сохранить все и выйти из редактора

Проверим, что файлы изменились

```
[giosokin:~]$ cat temp/*
Георгий
Осокин
НММбд-02-22
```

Скриншот 46: Вывод команды cat для всех файлов в директории temp

Видим, что

вывелся текст, который мы добавили в файлы

9. Скопируем все .txt файлы из каталога ~/temp в labs и соответственно переименуем в text1, text2, text3 в firstname, lastname и в id-group

```
[giosokin:~]$ cp temp/*.txt labs/
[giosokin:~]$ cd labs/
[giosokin:labs]$ mv text1.txt firstname
[giosokin:labs]$ mv text2.txt lastname
[giosokin:labs]$ mv text3.txt id-group
```

Скриншот 47: Копирование файлов из temp в labs

С помощью команды cat убедимся, что содержимое сохранилось и названия соответствующие

```
[giosokin:labs]$ cat firstname lastname id-group
Георгий
Осокин
НММбд-02-22
```

Скриншот 48: Вывод содержимого переименованных файлов

10. Удалим все файлы и директории, созданные за время лабораторной работы. (воспользуемся флагом -i что бы подтверждать удаление и не удалить лишние)

```
[giosokin:~]$ cd
[giosokin:~]$ rm -ri *
rm: remove directory 'Documents'? n
rm: descend into directory 'book'? y
rm: remove regular file 'book/page-1'? y
rm: remove regular file 'book/page-2'? y
rm: remove directory 'book'? y
rm: descend into directory 'labs'? y
rm: remove directory 'labs/lab1'? y
rm: remove directory 'labs/lab2'? y
rm: remove directory 'labs/lab3'? y
rm: remove regular file 'labs/firstname'? y
rm: remove regular file 'labs/lastname'? y
rm: remove regular file 'labs/id-group'? y
rm: remove directory 'labs'? y
rm: remove regular file 'letter-2'? y
rm: remove regular empty file 'new_name'? y
rm: remove directory 'newdir'? y
rm: descend into directory 'newstdir'? y
rm: remove directory 'newstdir/newdir'? y
rm: remove directory 'newstdir'? y
rm: descend into directory 'temp'? y
rm: remove regular file 'temp/text1.txt'? y
rm: remove regular file 'temp/text2.txt'? y
rm: remove regular file 'temp/text3.txt'? y
rm: remove directory 'temp'? y
rm: remove regular file 'text'? y
rm: remove directory 'tmp'? y
```

Скриншот 49: Процесс рекурсивного удаления с командой `rm -ri`

Выводы

В ходе лабораторной работы:

- Мы приобрели практические навыки работы с операционной системой на уровне командной строки. Научились создавать через нее пользователей, взаимодействовать с файловой системой и пользоваться базовыми командами `bash`.
- Мы познакомились с такими утилитами как `adduser`, `mv`, `ls`, `cat`, `echo`, `mkdir`, `touch`, `vim`, `rm` и некоторыми их опциями
- Мы научились перемещаться по файловой системе с помощью команды `cd`, используя абсолютные и относительные пути