

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

*дисциплина: Архитектура компьютера*

Студент: Ермаков А.В.

Группа: НФПбд-01-21

№ ст. билета: 1032213202

**МОСКВА**

**2023 г.**

## Оглавление

Список иллюстраций .....	3
Список таблиц.....	4
1 Цель работы .....	5
2 Задание .....	6
3 Теоретическое введение.....	7
4 Выполнение лабораторной работы.....	16
Создание пустых каталогов и файлов .....	18
Перемещение и удаление файлов или каталогов .....	19
Задание для самостоятельной работы .....	20
Вопросы для самопроверки.....	24
5 Выводы .....	28
Список литературы.....	29

## Список иллюстраций

Рис. 4.1: Пример использования команд pwd и cd	15
Рис. 4.2: Пример использования команды nautilus	15
Рис. 4.3: Примеры использования команды ls с разными ключами	16
Рис. 4.4: Примеры создания пустых каталогов	17
Рис. 4.5: Примеры создания пустых файлов	17
Рис. 4.6: Примеры удаление каждого файла в текущем каталоге	18
Рис. 4.7: Примеры копирования и перемещения файлов в другой каталог	19
Рис. 4.8: Выполнение задания №1	20
Рис. 4.9: Выполнение задания №2	20
Рис. 4.10: Выполнение задания №3	20
Рис. 4.11: Выполнение задания №4	21
Рис. 4.12: Выполнение задания №5	22
Рис. 4.13: Выполнение задания №6	22

### **Список таблиц**

Таблица 3.1 Стандарт иерархии файловой системы(FHS)	9
Таблица 3.2. Основные команды bash	11
Таблица 3.3 Комбинации клавиш управления	13
Таблица 4.1. Опции команды ls	16
Таблица 4.2. Опции команды rm	18
Таблица 4.3. Опции команды cp	18

## **1 Цель работы**

Приобретение практических навыков работы с операционной системой на уровне командной строки (организация файловой системы, навигация по файловой системе, создание и удаление файлов и директорий).

## **2 Задание**

Пользуясь изученными консольными командами произвести навигацию по файловой системе, создать, скопировать, переместить и удалить назначенные файлы и директории.

### **3 Теоретическое введение**

**GNU/Linux** — это операционная система: набор программ, которые позволяют работать с компьютером и запускать другие программы.

Операционная система (ОС) состоит из нескольких основных программ, которые нужны вашему компьютеру, чтобы он мог общаться и получать инструкции от пользователей; читать и записывать данные на жёсткие диски, ленты и принтеры; контролировать использование памяти; запускать другие программы. Самая важная часть ОС — ядро. В системе GNU/Linux ядром является Linux. Оставшаяся часть системы состоит из других программ, многие из которых написаны проектом GNU или для проекта GNU. Поскольку одно ядро Linux — это ещё не вся работающая ОС, мы предпочитаем использовать термин «GNU/Linux» для обозначения систем, которые многие люди ошибочно называют «Linux».

GNU/Linux сделана по образцу операционной системы Unix. С самого начала GNU/Linux разрабатывалась как многозадачная, многопользовательская система. Этих факторов достаточно, чтобы сделать GNU/Linux отличной от других широко известных операционных систем. Однако, GNU/Linux отличается намного сильнее, чем вы можете себе представить. В отличие от других операционных систем, GNU/Linux никому не принадлежит. Значительная часть системы разработана добровольцами бесплатно.

Ядро Linux - очень большая программа, представляющая следующие функции:

#### **Хранение данных:**

RAM - Чтение и запись данных в оперативную память.

Постоянное хранение - Чтение и запись данных на постоянные запоминающие устройства.

Виртуальная файловая система.

#### **Доступ к сетям:**

Работа с физическими сетями (Ethernet, Wireless, LTE, коммутаторы, Bluetooth).

Виртуальные сети.

#### **Планирование задач:**

Совместное использование процессорного времени программами.

Балансировка нагрузки процессора и приоритеты.

### **Поддержка внешних устройств:**

Съемные USB носители.

Веб-камеры.

Мыши и клавиатуры.

### **Безопасность:**

Права доступа к файлам для пользователей и групп;

Разрешение доступа к ресурсам системы;

**Командная оболочка** — это программа, которая принимает команды, введенные с клавиатуры, и передает их операционной системе для выполнения. Практически все дистрибутивы Linux поставляются с командной оболочкой из проекта GNU, которая называется bash. Имя bash — это аббревиатура от названия Bourne Again Shell, отражающего тот факт, что bash является улучшенной заменой sh, первоначальной командной оболочки для Unix, написанной Стивом Борном (Steve Bourne).

Взаимодействие с компьютером осуществляется посредством команд, которые имеют следующий вид:

[название команды] <опции> [аргументы]

Некоторые команды требуют прав суперпользователя. На самом деле, большинство команд - это программы, все так же с открытым исходным кодом, который можно редактировать.

### **Файловая структура GNU Linux**

В отличие от Windows в GNU/Linux используется ФС ext4. Одним из основных ее отличий является поддержка прав доступа к файлам и папкам

Права бывают трех типов:

Read(r) - чтение файлов или папок.

Write(w) - запись в файл или папку.

Execute(x) - запуск программ.

Правами могут обладать:

Владелец файла или папки.



Группа пользователей.

Все остальные, не входящие в первые 2 группы.

Суперпользователь имеет неограниченный доступ ко всем файлам.

### **Суперпользователь**

root - суперпользователь системы. А если более точно, то это пользователь с идентификатором 0. Имя здесь не особо важно. Хотя по умолчанию это общее имя пользователя с нулевым идентификатором пользователя для всех unix-like операционных систем. Это пользователь обладает наивысшими привилегиями в ОС.

### **Мандатное управление доступом**

Мандатное управление доступом (англ. Mandatory access control, MAC) — разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности. Также иногда переводится как Принудительный контроль доступа.

Для файловых систем, оно может расширять или заменять дискреционный контроль доступа и концепцию пользователей и групп.

Реализации в Linux:

SELinux - реализация системы принудительного контроля доступа, которая может работать параллельно с классической избирательной системой контроля доступа. Разработан АНБ США, включен в состав ядра (начиная с 2.6).

AppArmor - программный инструмент упреждающей защиты, основанный на политиках безопасности, которые определяют, к каким системным ресурсам и с какими привилегиями может получить доступ то или иное приложение.

В сертифицированной в системах сертификации Минобороны России и ФСТЭК России операционной системе специального назначения Astra Linux Special Edition,

механизм мандатного разграничения доступа реализован, как и механизм дискреционного разграничения доступа в ядре ОС и СУБД.

**Таблица 3.1 Стандарт иерархии файловой системы(FHS)**

FHS предлагает следующую иерархию подкаталогов в корневом /:	
/bin	- основные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям.
/boot	- неизменяемые файлы, необходимые для загрузки системы.
/dev	- файлы устройств.
/etc	- файлы конфигурации системы на данном компьютере.
/home	- домашние каталоги пользователей.
/lib	- основные разделяемые библиотеки и модули ядра.
/mnt	- точку монтирования для временно подключаемых файловых систем.
/root	- домашний каталог пользователя root.
/opt	- дополнительное программное обеспечение.
/sbin	- основные системные программы для администрирования и настройки системы.
/tmp	- временные файлы.
/usr	- вторичная иерархия для данных пользователя. Содержит большинство пользовательских приложений и утилит, используемых в

	многопользовательском режиме.
/usr/b in	- дополнительные программы для всех пользователей, не являющиеся необходимыми в однопользовательском режиме.

Изначально /usr использовался для хранения домашних директорий. Когда стало нехватать места под bin, было решено создать пользователя bin.

/var - изменяемые файлы, такие как файлы регистрации, временные почтовые файлы.

## Systemd

Systemd — подсистема инициализации Linux — демон (программа, работающая в фоне) для запуска других демонов в Linux и управления ими в процессе работы системы. Название происходит от принятого в Unix добавления суффикса -d к демонам.

Разработана Леннартом Поттерингом взамен используемого ранее демона init. В свое время вызвала огромное количество не утихающий до сих пор споров. Сейчас используется по умолчанию во всех популярных дистрибутивах.

Помимо простого запуска и контроля сервисов, systemd предлагает некоторые другие удобные функции, для использования которых ранее системным администраторам приходилось прибегать к помощи дополнительных программ-демонов. Среди таких функций:

Сокет-активация служб (заменяет inetd).

Запуск сервисов по расписанию (заменяет cron).

Работа с аппаратным сторожевым таймером (заменяет watchdog).

Смена корня (заменяет chroot).

Автомонтирование томов и сетевых ресурсов (заменяет mount и fstab).

Удобная работа с логами посредством journald (заменяет syslog)

**Таблица 3.2. Основные команды bash**

man <команда>	подробная информация о команде <команда>
pwd	текущая директория
cd <директория>	сменить текущую директорию на <директория>
~	домашняя директория пользователя
ls	<p>список файлов и папок в текущей директории</p> <p>ls -l - подробный список(включает права, владельца, группу, размер и время редактирования)</p> <p>ls -a - скрытые файлы и папки(которые начинаются с .)</p> <p>ls -la - две опции вместе</p> <p>ls &lt;дир&gt; - список файлов в директории &lt;дир&gt;</p>
touch <файл>	создает файл <файл>
mkdir <дир>	<p>создает директорию &lt;дир&gt;</p> <p>mkdir -p &lt;дир&gt; - создает директорию &lt;дир&gt; и все вышележащие директории</p> <p>например, mkdir -p /test/test2/test3 - Создаст директории test, test2, test3</p>
cp <файл1> <файл2>	<p>копировать &lt;файл1&gt; в &lt;файл2&gt;</p> <p>cp -r &lt;дир1&gt; &lt;дир2&gt; - копировать директорию &lt;дир1&gt; в &lt;дир2&gt;</p>
mv <1> <2>	<p>переместить файл или директорию &lt;1&gt; в &lt;2&gt;</p> <p>rm &lt;файл&gt; - удаляет файл &lt;файл&gt;</p> <p>rm -rf &lt;дир&gt; - удаляет директорию &lt;дир&gt; и все ее содержимое</p>
echo <текст>	печатает <текст>

	<p>echo &lt;текст&gt; &gt; &lt;файл&gt; - заменяет содержимое файла &lt;файл&gt; текстом &lt;текст&gt;</p> <p>echo &lt;текст&gt; &gt;&gt; &lt;файл&gt; - добавляет текст &lt;текст&gt; в конец файла &lt;файл&gt;</p> <p>echo &lt;текст&gt; 1&gt;&amp;2 - перенаправление stdout в stderr</p> <p>Если целевого файла нет, он будет создан</p>
cat <файл>	<p>печатает содержимое файла &lt;файл&gt;</p> <p>cat &lt;файл1&gt; &gt; &lt;файл2&gt; - заменяет содержимое файла &lt;файл1&gt; содержимым &lt;файл2&gt;</p> <p>cat &lt;файл1&gt; &gt;&gt; &lt;файл2&gt; - добавляет текст файла &lt;файл1&gt; в конец файла &lt;файл2&gt;</p> <p>Если целевого файла нет, он будет создан</p>
ping <хост>	проверить доступность <хост> в сети
ip	<p>позволяет выполнять настройку сетевой подсистемы</p> <p>ip link - информация о сетевых интерфейсах</p> <p>ip addr - информация о сетевых интерфейсах и адресах</p> <p>ip route - таблица маршрутизации</p> <p>ip neigh - текущая ARP-таблица(таблица соответствия IP- и MAC-адресов)</p>
useradd <пользователь>	<p>создать пользователя &lt;пользователь&gt;</p> <p>useradd -m &lt;пользователь&gt; - создать пользователя и его домашнюю директорию(/home/&lt;пользователь&gt;)</p>
passwd <пользователь>	задать пароль пользователю <пользователь>
su <пользователь>	авторизоваться под пользователем <пользователь>
sudo <команда>	выполнить команду <команда> от имени суперпользователя root

exit	выйти/деавторизоваться
ps	вывести список активных процессов текущего пользователя  ps -e - список всех активных процессов
top	вывести в виде таблицы перечень запущенных процессов и оценить, какой объем ресурсов они потребляют
htop	аналогично. Более удобный. Обычно используется вместо
kill <pid>	отправляет сигнал процессу с идентификатором <pid>
ssh <пользователь>@<хост>	удаленно подключиться к компьютеру <хост> и зайти под пользователем <пользователь>
scp <файл> <пользователь>@<хост>:<дир>	отправить файл <файл> по протоколу
hostname	имя компьютера
dnf install <пакет>	установка пакета <пакет> пакетным менеджером dnf
apt install <пакет>	установка пакета <пакет> пакетным менеджером apt

**Таблица 3.3 Комбинации клавиш управления**

Ctrl + C	посылает запущенному в терминале процессу SIGINT (сигнал для остановки процесса)
Ctrl + D	комбинация, которая отправляет в консоль EOF (End Of File). Работает схожим образом с exit
Ctrl + Z	отправляет процессу SIGSTOP (сигнал, посылаемый для принудительной приостановки выполнения процесса)
Ctrl + a и Ctrl + e	перемещают курсор в начало и в конец текущей строки

Ctrl + k	удаляет всё от текущей позиции курсора до конца строки
Ctrl + d	в пустой строке служит для завершения текущего сеанса

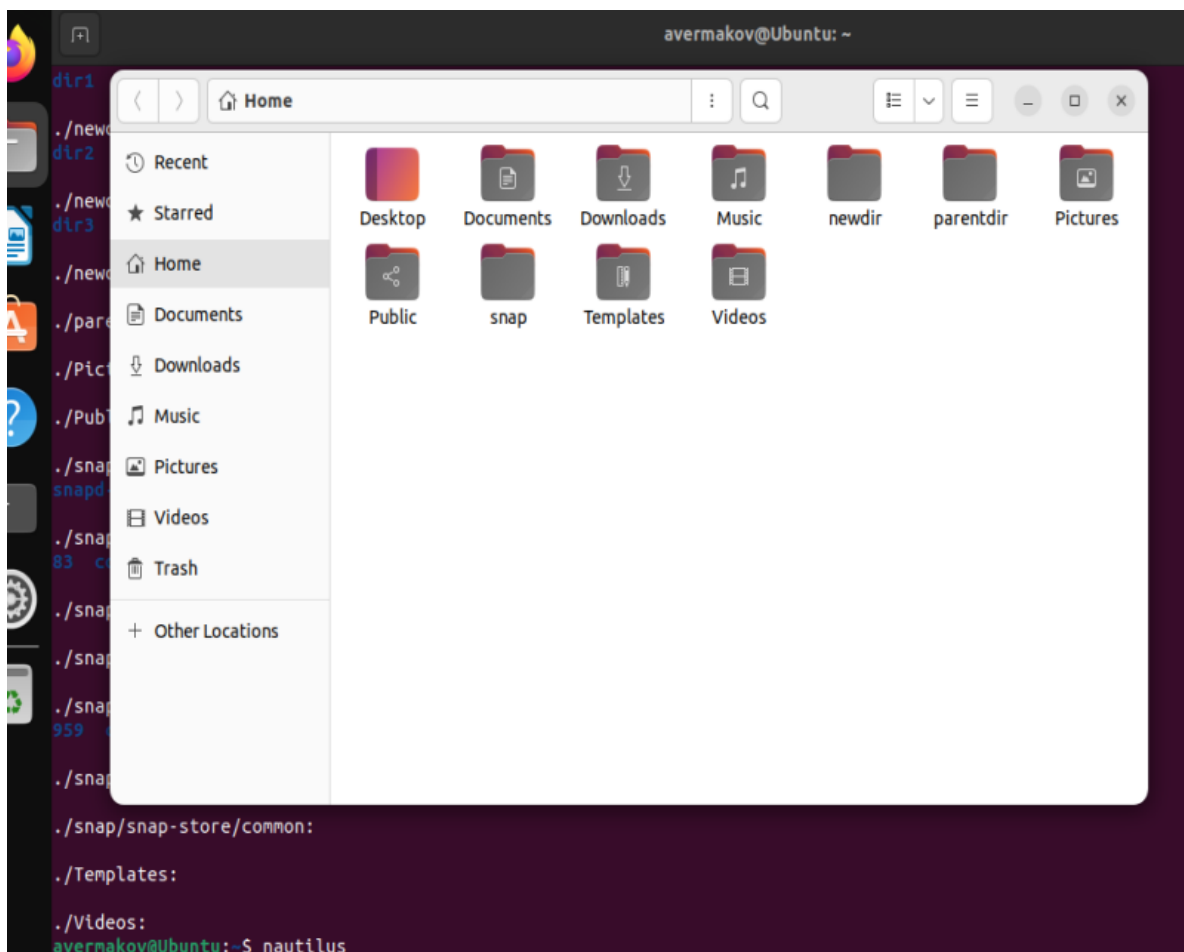
## 4 Выполнение лабораторной работы

С помощью команды `pwd` узнали полный путь к домашнему каталогу и осуществили навигацию по файловой системе (рис. 4.1).

```
avermakov@Ubuntu:~/parentdir$ pwd
/home/avermakov/parentdir
avermakov@Ubuntu:~/parentdir$ cd
avermakov@Ubuntu:~$ pwd
/home/avermakov
avermakov@Ubuntu:~$ 
avermakov@Ubuntu:~$ cd parentdir
avermakov@Ubuntu:~/parentdir$ cd
avermakov@Ubuntu:~$ cd parentdir
```

**Рис. 4.1: Пример использования команд `pwd` и `cd`**

Перешли в домашний каталог с помощью файлового менеджера графического окружения ОС командой `nautilus` (рис. 4.2).





## Рис. 4.2: Пример использования команды nautilus

Для просмотра списка файлов текущего каталога может быть использована команда `ls` (рис. 4.3).

```
avermakov@Ubuntu:~$ ls
Desktop  Documents  Downloads  Music  newdir  parentdir  Pictures  Public  snap  Templates  Videos
avermakov@Ubuntu:~$ ls -a
.  .bash_history  .bashrc  .config  Documents  .gnupg  .local  newdir  Pictures  Public  .ssh  Videos
.. .bash_logout  .cache  Desktop  Downloads  .lessht  Music  parentdir  .profile  snap  Templates
avermakov@Ubuntu:~$ ls -h
Desktop  Documents  Downloads  Music  newdir  parentdir  Pictures  Public  snap  Templates  Videos
avermakov@Ubuntu:~$ ls -l
1441817 Desktop      1441818 Downloads  1442543 newdir     1441823 Pictures  1441956 snap      1441824 Videos
1441821 Documents  1441822 Music      1442537 parentdir  1441820 Public   1441819 Templates
avermakov@Ubuntu:~$ ls -d
.
avermakov@Ubuntu:~$ ls -R
.:
Desktop  Documents  Downloads  Music  newdir  parentdir  Pictures  Public  snap  Templates  Videos

./Desktop:
'New Folder'

'./Desktop/New Folder':

./Documents:

./Downloads:

./Music:

./newdir:
dir1

./newdir/dir1:
dir2
```

## Рис. 4.3: Примеры использования команды `ls` с разными ключами

Основные опции команды `ls` отражены в таблице 4.1

**Таблица 4.1. Опции команды `ls`**

Ключ	Описание
-a	вывод списка всех файлов, включая скрытые файлы (в Linux названия скрытых файлов начинаются с точки)
-R	рекурсивный вывод списка файлов и подкаталогов
-h	вывод для каждого файла его размера
-l	вывод дополнительной информации о файлах (права доступа, владелец и группы, размеры файлов и время последнего доступа)
-i	вывод уникального номера файла (inode) в файловой системе перед каждым файлом

-d	обработка каталогов, указанных в командной строке, так, как если бы они были обычными файлами, вместо вывода списка их файлов
----	---

### Создание пустых каталогов и файлов

В домашнем каталоге создаётся подкаталог с именем parentdir. С помощью команды ls производится проверка, что каталог создан. Создаётся подкаталог в существующем каталоге dir и несколько каталогов dir1 dir2 dir3. С помощью команды ls производится проверка (рис. 4.4).

```

avernakov@Ubuntu:~$ mkdir parentdir
avernakov@Ubuntu:~$ ls
Desktop Documents Downloads Music parentdir Pictures Public snap Templates Videos
avernakov@Ubuntu:~$ cd parentdir
avernakov@Ubuntu:~/parentdir$ mkdir dir1 dir2 dir3
avernakov@Ubuntu:~/parentdir$ ls
dir1 dir2 dir3
avernakov@Ubuntu:~/parentdir$ mkdir ~/newdir
avernakov@Ubuntu:~/parentdir$ ls -R
.:
dir1 dir2 dir3

./dir1:

./dir2:

./dir3:

```

**Рис. 4.4: Примеры создания пустых каталогов**

Создаётся файл test.txt в каталоге ~/newdir/dir1/dir2. С помощью команды ls производится проверка (рис. 4.5).

```

avernakov@Ubuntu:~$ touch ~/newdir/dir1/dir2/dir3/test.txt
avernakov@Ubuntu:~$ cd dir3
bash: cd: dir3: No such file or directory
avernakov@Ubuntu:~$ ls ~/newdir/dir1/dir2/dir3/test.txt
/home/avernakov/newdir/dir1/dir2/dir3/test.txt
avernakov@Ubuntu:~$

```

**Рис. 4.5: Примеры создания пустых файлов**

## Перемещение и удаление файлов или каталогов

Команда `rm` удаляет файлы и (или) каталоги и имеет следующие опции (табл.4.2)

**Таблица 4.2. Опции команды `rm`**

• <code>-r</code> или <code>-R</code>	рекурсивное удаление (это обязательная опция для удаления любого каталога, пустого или содержащего файлы и (или) подкаталоги);
<code>-i</code>	запрос подтверждения перед удалением;
<code>-v</code>	вывод подробной информации при выполнении команды;
<code>-f</code>	принудительное удаление файлов или каталогов.

Запросив подтверждение на удаление каждого файла в текущем каталоге, удаляем в подкаталоге `/newdir/dir1/dir2/` все файлы с именами, заканчивающимися на `.txt`. Рекурсивно удаляем из текущего каталога без запроса подтверждения на удаление каталог `newdir`, а также файлы, чьи имена начинаются с `dir` в каталоге `parentdir` (рис. 4.6)

```
avermakov@Ubuntu:~$ rm -R ~/newdir ~/parentdir/*dir
avermakov@Ubuntu:~/parentdir$ cd
avermakov@Ubuntu:~$ touch ~/newdir/dir1/dir2/*.txt
avermakov@Ubuntu:~$ ls ~/newdir/dir1/dir2/*.txt
'/home/avermakov/newdir/dir1/dir2/*.txt'
avermakov@Ubuntu:~$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/avermakov/newdir/dir1/dir2/*.txt'?
avermakov@Ubuntu:~$ ls ~/newdir/dir1/dir2/*.txt
'/home/avermakov/newdir/dir1/dir2/*.txt'
avermakov@Ubuntu:~$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/avermakov/newdir/dir1/dir2/*.txt'? y
avermakov@Ubuntu:~$ ls ~/newdir/dir1/dir2/*.txt
ls: cannot access '/home/avermakov/newdir/dir1/dir2/*.txt': No such file or directory
avermakov@Ubuntu:~$ rm -R ~/parentdir/dir*
avermakov@Ubuntu:~$ ls ~ /parentdir
ls: cannot access '/parentdir': No such file or directory
```

**Рис. 4.6: Примеры удаление каждого файла в текущем каталоге**

Команда `cp` копирует файлы и каталоги и имеет следующие опции (табл.4.3)

**Таблица 4.3. Опции команды `cp`**

• <code>-R</code>	рекурсивное копирование; является обязательной опцией для копирования каталогов;
• <code>-i</code>	запрос подтверждения перед перезаписью любых файлов;

• -f	заменяет любые существующие файлы без запроса подтверждения;
• -v	подробный режим, сообщает обо всех изменениях и действиях.

Используя команды `cp` и `mv` копируем и перемещаем файлы в другой каталог (рис.4.7)

```

avermakov@Ubuntu:~$ ls ~/parentdir
avermakov@Ubuntu:~$ mkdir ~/parentdir/dir2 ~/parentdir/dir1
avermakov@Ubuntu:~$ ls ~/parentdir
dir1 dir2
avermakov@Ubuntu:~$ touch ~/parentdir/dir1/test.txt
touch: cannot touch '/home/avermakov/parentdir/dir1/test.txt': No such
avermakov@Ubuntu:~$ touch ~/parentdir/dir1/test.txt
avermakov@Ubuntu:~$ ls -R ~/parentdir
/home/avermakov/parentdir:
dir1 dir2

/home/avermakov/parentdir/dir1:
test.txt

/home/avermakov/parentdir/dir2:
avermakov@Ubuntu:~$ mv ~/parentdir/dir1/test.txt ~/parentdir/dir2
avermakov@Ubuntu:~$ ls -R ~/parentdir
/home/avermakov/parentdir:
dir1 dir2

/home/avermakov/parentdir/dir1:

/home/avermakov/parentdir/dir2:
test.txt
avermakov@Ubuntu:~$ cp ~/parentdir/dir2/test.txt ~/parentdir/dir1
avermakov@Ubuntu:~$ ls -R ~/parentdir
/home/avermakov/parentdir:
dir1 dir2

/home/avermakov/parentdir/dir1:
test.txt

/home/avermakov/parentdir/dir2:
test.txt

```

**Рис. 4.7: Примеры копирования и перемещения файлов в другой каталог**

#### **Задание для самостоятельной работы**

1. Воспользовавшись командой `pwd`, узнайте полный путь к своей домашней директории. Выполнение задания (рис.4.8.)

```
avermakov@Ubuntu:~$ pwd
/home/avermakov
```

**Рис. 4.8: Выполнение задания №1**

2. Введите следующую последовательность команд

```
cd
mkdir tmp
cd tmp
pwd
cd /tmp
pwd
```

Выполнение задания (рис.4.9.) pwd выводит путь к рабочей папке, cd переводит в корневую папку, mkdir tmp создаёт папку tmp, cd tmp переводит в папку tmp, pwd выводит путь к папке tmp

```
avermakov@Ubuntu:~$ pwd
/home/avermakov
avermakov@Ubuntu:~$ cd
avermakov@Ubuntu:~$ mkdir tmp
avermakov@Ubuntu:~$ pwd
/home/avermakov
avermakov@Ubuntu:~$ cd tmp
avermakov@Ubuntu:~/tmp$ pwd
/home/avermakov/tmp
avermakov@Ubuntu:~/tmp$
```

**Рис. 4.9: Выполнение задания №2**

Объясните, почему вывод команды pwd при переходе в каталог tmp дает разный результат.

Команда pwd означает рабочий каталог печати. При вызове команда печатает полный путь к текущему рабочему каталогу. Так как текущий рабочий каталог изменился, то и результат различен.

3. Пользуясь командами cd и ls, посмотрите содержимое разных каталогов. Выполнение задания (рис.4.10.) Сначала перешли из tmp в корневую папку, ls посмотрели содержание корневой папки, посмотрели tmp, там пусто, поэтому посмотрели содержание папки snap, в ней были подкаталоги, потом посмотрели папки



рабочего стола

```
avermakov@Ubuntu:~/tmp$ cd
avermakov@Ubuntu:~$ ls
Desktop Documents Downloads Music parentdir Pictures Public snap Templates tmp Videos
avermakov@Ubuntu:~$ ls ~/tmp
avermakov@Ubuntu:~$ ls ~/snap
snapd-desktop-integration snap-store
avermakov@Ubuntu:~$ ls ~/Desktop
'New Folder'
```

**Рис. 4.10: Выполнение задания №3**

4. Пользуясь изученными консольными командами, в своём домашнем каталоге создайте каталог temp и каталог labs с подкаталогами lab1, lab2 и lab3 одной командой. В каталоге temp создали файлы text1.txt, text2.txt, text3.txt. Пользуясь командой ls, убедитесь, что все действия выполнены успешно (каталоги и файлы созданы). Выполнение задания (рис.4.11.) Создали каталог temp и каталог labs с подкаталогами lab1, lab2 и lab3 одной командой. Пользуясь командой ls, убедились, что все действия выполнены успешно. В каталоге temp создали файлы test1.txt, test2.txt, test3.txt. Пользуясь командой ls, убедились, что все действия выполнены успешно

```
avermakov@Ubuntu:~$ ls
Desktop Documents Downloads Music parentdir Pictures Public snap Templates tmp Videos
avermakov@Ubuntu:~$ mkdir temp labs labs/lab1 labs/lab2 labs/lab3
avermakov@Ubuntu:~$ ls
Desktop Documents Downloads labs Music parentdir Pictures Public snap temp Templates tmp Videos
avermakov@Ubuntu:~$ ls labs
lab1 lab2 lab3
avermakov@Ubuntu:~$ touch temp/test1.txt temp/test2.txt temp/test3.txt
avermakov@Ubuntu:~$ ls temp
test1.txt test2.txt test3.txt
avermakov@Ubuntu:~$
```

**Рис. 4.11: Выполнение задания №4**

5. С помощью любого текстового редактора (например, редактора mcedit) запишите в файл text1.txt свое имя, в файл text2.txt фамилию, в файл text3.txt учебную группу. Выведите на экран содержимое файлов, используя команду cat. Выполнение задания (рис.4.12.) С помощью cat записали в файл text1.txt свое имя, в файл text2.txt фамилию, в файл text3.txt учебную группу. Так как фамилию записали два раза, то с помощью текстового редактора отредактировали запись. Вывели на экран содержимое файлов, используя команду cat.

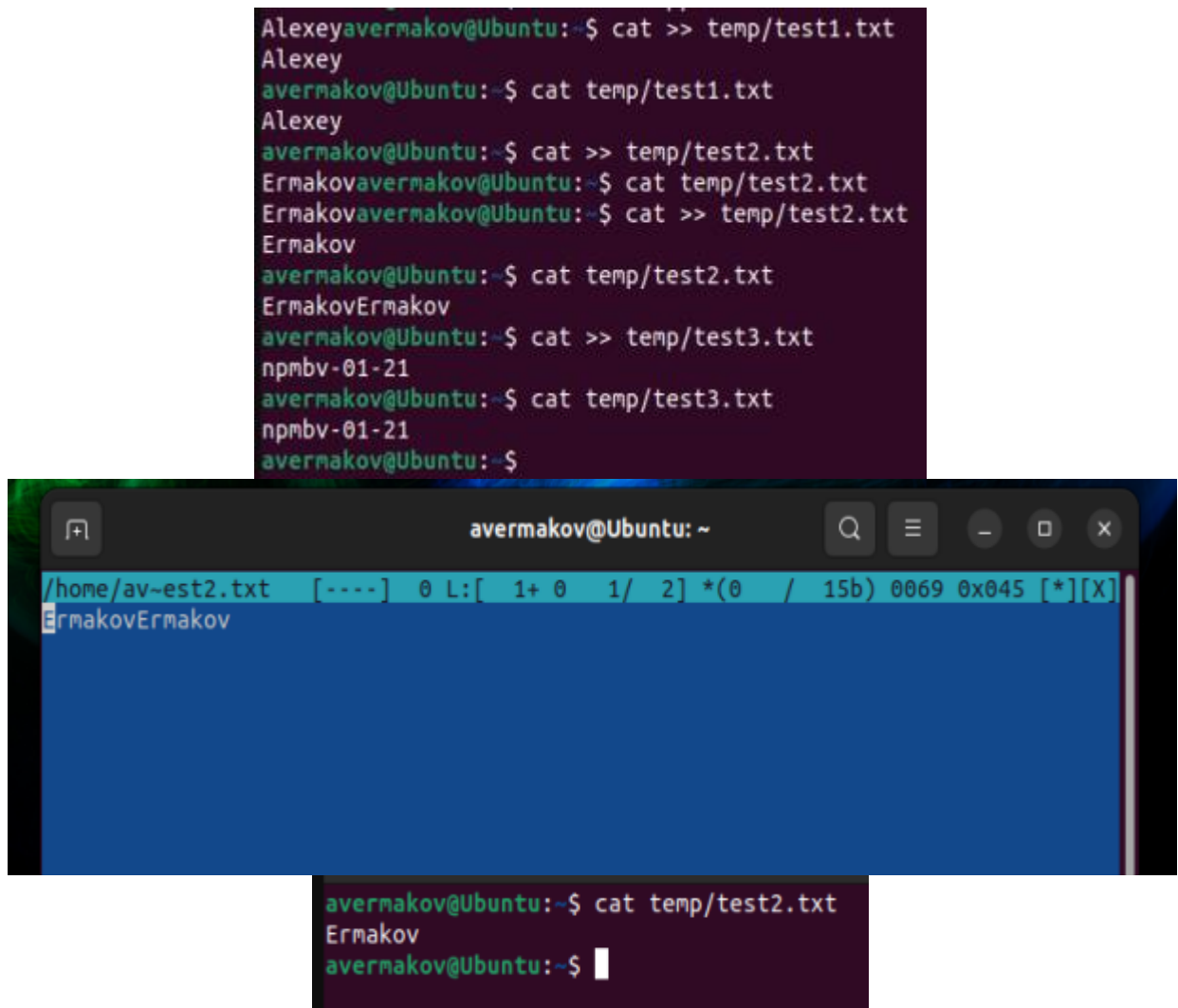


Рис. 4.12: Выполнение задания №5

6. Скопируйте все файлы, чьи имена заканчиваются на .txt, из каталога ~/temp в каталог labs. После этого переименуйте файлы каталога labs и переместите их: test1.txt переименуйте в firstname.txt и переместите в подкаталог lab1, test2.txt в lastname.txt в подкаталог lab2, test3.txt в id-group.txt в подкаталог lab3. Пользуясь командами ls и cat, убедитесь, что все действия выполнены верно. Удалите все созданные в ходе выполнения лабораторной работы файлы и каталоги. Выполнение задания (рис.4.13.) Скопировали все файлы, чьи имена заканчиваются на .txt, из каталога ~/temp в каталог labs. После этого переименовали файлы каталога labs и переместили их: test1.txt переименовали в firstname.txt и переместили в подкаталог lab1, test2.txt в lastname.txt в подкаталог lab2, test3.txt в id-group.txt в подкаталог lab3. Пользуясь командами ls и cat, убедились, что все действия выполнены верно. Удалили все созданные в ходе выполнения лабораторной работы файлы и каталоги

```
avermakov@Ubuntu:~$ ls labs
lab1 lab2 lab3 test1.txt test2.txt test3.txt
avermakov@Ubuntu:~$ mv labs/test1.txt labs/lab1/firstname.txt
avermakov@Ubuntu:~$ mv labs/test2.txt labs/lab2/lastname.txt
avermakov@Ubuntu:~$ rm labs/lab1/lastname.txt labs/lab1/id-group.txt
avermakov@Ubuntu:~$ mv labs/test3.txt labs/lab3/id-group.txt
avermakov@Ubuntu:~$ ls -R labs
labs:
lab1 lab2 lab3

labs/lab1:
firstname.txt

labs/lab2:
lastname.txt

labs/lab3:
id-group.txt
avermakov@Ubuntu:~$ rm -r labs temp
avermakov@Ubuntu:~$ ls
Desktop Documents Downloads Music parentdir Pictures Public snap Templates
avermakov@Ubuntu:~$
```

Рис. 4.13: Выполнение задания №6

### Вопросы для самопроверки

1. Дайте определение командной строки. Приведите примеры.

*Терминал Линукс (Linux Terminal) — это программа, которая используется для взаимодействия с командной оболочкой. Ее второе название — командная строка Линукс. Терминал предназначен для администрирования ОС: установки и удаления программ, работы с файловой системой и др. Открыть окно терминала можно с помощью комбинации клавиш Ctrl + Alt + T. Например: Объединить содержимое двух файлов и отобразите результат в терминале: cat file1.txt file2.txt*

2. Как получить информацию об интересующей вас команде?

*man — справка по командам и программам, доступным на рабочей машине, а также по системным вызовам и стандартной библиотеке C. Например: man grep, man atoi, man chdir, man man.*

3. Чем относительный путь к файлу отличается от абсолютного?

*Полный, абсолютный путь linux от корня файловой системы - этот путь начинается от корня "/" и описывает весь путь к файлу*

*Относительный путь linux - это путь к файлу относительно текущей папки*

*Путь относительно домашней папки текущего пользователя. - путь в файловой*



системе, только не от корня, а от папки текущего пользователя

4. Как определить абсолютный путь к текущей директории?

*В терминале выполнить команду `pwd`*

5. При помощи каких команд можно удалить файл и каталог? Можно ли это сделать одной и той же командой?

*rm - это утилита командной строки для удаления файлов и каталогов.*

6. Как можно запустить несколько команд в одной строке? Приведите примеры.

*Выполнить сначала одну команду. Затем, когда она отработает (вернет управление в командную строку), нужно запустить вторую команду. Для этого служит символ точка с запятой. Таким образом, если в терминале: `тусоттmand1; тусоттmand2; тусоттmand3` то сначала запустится первая команда `тусоттmand1`, а когда управление вернется в терминал, то запустится вторая и так далее. Это называется последовательным выполнением команд. Вне зависимости от того, как выполнялась предыдущая команда, вторая команда должна будет запуститься. Можно выполнять команды в зависимости от того, как отработала предыдущая для этого служат символы `&&` и `||`. Символы `&&` и `||` означают логическое И и ИЛИ. Поэтому `тусоттmand1 && тусоттmand2 && тусоттmand3` означает, что команда `тусоттmand2` будет выполнена только в том случае, если команда `тусоттmand1` была выполнена успешно. Каждая следующая команда выполняется только при успешном выполнении предыдущей. `тусоттmand1 || тусоттmand2 || тусоттmand3` - команда `тусоттmand2` будет выполнена только, если `тусоттmand1` вернула ошибку. Каждая следующая команда запускается только если предыдущая вернула ошибку.*

7. Какая информация выводится на экран о файлах и каталогах, если используется опция `-l` в команде `ls`?

*`-l` – вывести длинный список с подробной информацией.*

8. Каким образом отобразить информацию о скрытых файлах? Приведите примеры.

*Скрытые файлы можно найти с помощью этой команды `ls -`*

*`$ ls -a`*

*При этом будут показаны скрытые файлы, а также обычные файлы.*

*`.cache .emacs.d .profile Videos`*

*`.. .config ff Public .viminfo`*

```
.aws dd .gnupg results.txt wd
```

```
** many more files **
```

```
.bash_history Desktop .local .ssh
```

```
.bash_logout Documents Music .sudo_as_admin_successful
```

```
.bashrc Downloads Pictures Templates
```

Для отображения скрытых файлов с помощью команды `ls` используем `-a`. Это обозначает "все" и предписывает команде `"ls"` перечислить все файлы (включая скрытые файлы). Для отображения дополнительной информации о каждом файле можем использовать `-l`, что означает "длинный список". Флаг `-l` отображает тип файла, разрешения, группу, размер, владельца и время модификации.

```
$ ls -al
```

Эта команда отобразит все файлы, включая скрытые, в текущем каталоге, вместе с подробной информацией по каждому файлу.

```
total 112
```

```
drwxr-xr-x 17 papan papan 4096 Mar 17 05:39 .
```

```
drwxr-xr-x 3 root root 4096 Feb 17 20:53 ..
```

```
drwxrwxr-x 2 -- -- 4096 Feb 25 01:13 .aws
```

```
** many more files... **
```

```
-rw----- 1 papan papan 6101 Mar 16 02:48 .bash_history
```

```
-rw-r--r-- 1 papan papan 220 Feb 17 20:53 .bash_logout
```

```
-rw-r--r-- 1 papan papan 3771 Feb 17 20:53 .bashrc
```

```
с помощью команды Find
```

Эта команда является полезным инструментом для поиска файлов и каталогов в Linux. Чтобы найти скрытые файлы с помощью команды `find`, можем использовать опцию `-name` для указания имени файла. В Linux скрытые файлы начинаются с точки, поэтому мы можем использовать опцию `-name` `"*"`, это приведет к поиску всех файлов, присутствующих в нашей системе, которые начинаются с точки. Здесь использовали опцию `-type f` только для поиска файлов.

```
$ find . -name "*" -type f
```

Эта команда будет выполнять рекурсивный поиск всех скрытых файлов, а также отображать их имена и пути.

```
./.profile
```

*./Desktop/cbl/.1.cbl.swp*  
*./bashrc*  
*./emacs.d/auto-save-list/.save-6894-ubuntu~*  
*\*\* many more folders \*\**  
*./bash\_history*  
*./bash\_logout*  
*./viminfo*  
*./sudo\_as\_admin\_successful*

*Чтобы найти только скрытые каталоги в Linux, нужно использовать некоторые параметры для фильтрации результатов. Параметр “-name” сопоставляет шаблон с именем файлов или каталогов, используя “.” для поиска имен файлов, начинающихся с точки, что является общепринятым для скрытых файлов в Linux. -type d выполняет поиск только по каталогам, в то время как -maxdepth ограничивает поиск текущим каталогом, предотвращая получение большого количества результатов. “2> /dev/null” перенаправляет сообщения об ошибках на нулевое устройство, избегая беспорядка в выходных данных.*

9. Какая клавиша или комбинация клавиш служит для автоматического дополнения

вводимых команд?

*Tab используется для дополнения команд или имени файлов в командной строке., если вводить название файла, система автоматически дополнит название файла до полного имени.*

## **5 Выводы**

В результате выполнения работы был сделан вывод о достаточно высоком пороге входа для работы с данной ОС (даже командную строку пришлось загружать нестандартными методами, хотя в другой ОС она доступна по умолчанию). Были приобретены практические навыки работы с ОС на уровне командной строки (организация файловой системы, навигация по файловой системе, создание и удаление файлов и директорий).

### **Список литературы**

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.  
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с