

# 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: *Панявкина Ирина*

Группа: *НКАбд-04-24*

МОСКВА

2024 г.

# Лабораторная работа №1. Основы интерфейса командной строки ОС GNU Linux

## Содержание

1. Цель работы
2. Теоретическое введение
  - 2.1. Введение в GNU Linux
  - 2.2. Введение в командную строку GNU Linux
  - 2.3. Файловая структура GNU Linux: каталоги и файлы
  - 2.4. Базовые команды bash
  - 2.5. Полезные комбинации клавиш
3. Выполнение лабораторной работы
4. Выполнение самостоятельной работы
5. Вопросы для самопроверки
6. Выводы
7. Список литературы

# 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. Теоретическое введение

### 2.1. Введение в GNU Linux

Операционная система (ОС)— это комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем. Сегодня наиболее известными операционными системами являются ОС семейства Microsoft Windows и UNIX-подобные системы.

GNU Linux — семейство переносимых, многозадачных и многопользовательских операционных систем, на базе ядра Linux, включающих тот или иной набор утилит и программ проекта GNU, и, возможно, другие компоненты. Как и ядро Linux, системы на его основе, как правило, создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения (Open-Source Software). Linux-системы распространяются в основном бесплатно в виде различных дистрибутивов.

Дистрибутив GNU Linux — общее определение ОС, использующих ядро Linux и набор библиотек и утилит, выпускаемых в рамках проекта GNU, а также графическую оконную подсистему X Window System. Дистрибутив готов для конечной установки на пользовательское оборудование. Кроме ядра и, собственно, операционной системы дистрибутивы обычно содержат широкий набор приложений, таких как редакторы документов и таблиц, мультимедийные проигрыватели, системы для работы с базами данных и т.д. Существуют дистрибутивы, разрабатываемые как при коммерческой поддержке (Red Hat / Fedora, SLED / OpenSUSE, Ubuntu), так и исключительно усилиями добровольцев (Debian, Slackware, Gentoo, ArchLinux).

## 2.2. Введение в командную строку GNU Linux

Работу ОС GNU Linux можно представить в виде функционирования множества взаимосвязанных процессов. При загрузке системы сначала запускается ядро, которое, в свою очередь, запускает оболочку ОС (от англ. shell «оболочка»). Взаимодействие пользователя с системой Linux (работа с данными и управление работающими в системе процессами) происходит в интерактивном режиме посредством командного языка. Оболочка операционной системы (или командная оболочка, интерпретатор команд) — интерпретирует (т.е. переводит на машинный язык) вводимые пользователем команды, запускает соответствующие программы (процессы), формирует и выводит ответные сообщения. Кроме того, на языке командной оболочки можно писать небольшие программы для выполнения ряда последовательных операций с файлами и содержащимися в них данными — сценарии (скрипты).

Из командных оболочек GNU Linux наиболее популярны `bash`, `csh`, `ksh`, `zsh`. Команда `echo $SHELL` позволяет проверить, какая оболочка используется. В качестве предустановленной командной оболочки GNU Linux используется одна из наиболее распространённых разновидностей командной оболочки — `bash` (Bourne again shell).

В GNU Linux доступ пользователя к командной оболочке обеспечивается через терминал (или консоль). Запуск терминала можно осуществить через главное меню: Приложения – Стандартные – Терминал (или Консоль) или нажав `Ctrl + Alt + t`.

Интерфейс командной оболочки очень прост. Обычно он состоит из приглашения командной строки (строки, оканчивающейся символом `$`), по которому пользователь вводит команды:

```
iivanova@dk4n31:~$
```

Это приглашение командной оболочки, которое несёт в себе информацию об имени пользователя `iivanova`, имени компьютера `dk4n31` и текущем каталоге, в котором находится пользователь, в данном случае это домашний каталог пользователя, обозначенный как `~`.

Команды могут быть использованы с ключами (или опциями) — указаниями, модифицирующими поведение команды. Ключи обычно начинаются с символа (-) или (--) и часто состоят из одной буквы. Кроме ключей после команды могут быть использованы аргументы (параметры) — названия объектов, для которых нужно выполнить команду (например, имена файлов и каталогов). Например, для подробного просмотра содержимого каталога documents может быть использована команда ls с ключом -l:

```
iivanova@dk4n31:~$ ls -l documents
```

В данном случае ls — это имя команды, l — ключ, documents — аргумент. Команды, ключи и аргументы должны быть отделены друг от друга пробелом.

Ввод команды завершается нажатием клавиши Enter, после чего команда передаётся оболочке на исполнение. Результатом выполнения команды могут являться сообщения о ходе выполнения команды или об ошибках. Появление приглашения командной строки говорит о том, что выполнение команды завершено.

Иногда в GNU Linux имена программ и команд слишком длинные, однако bash может завершать имена при их вводе в терминале. Нажав клавишу Tab, можно завершить имя команды, программы или каталога. Например, предположим, что нужно использовать программу mcedit. Для этого наберите в командной строке mc, затем нажмите один раз клавишу Tab. Если ничего не происходит, то это означает, что существует несколько возможных вариантов завершения команды. Нажав клавишу Tab ещё раз, можно получить список имён, начинающихся с mc:

```
iivanova@dk4n31:~$ mc
mc      mcd      mcedit   mclasserace  mcookie  mcview
mcat    mcdiff   mcheck   mcomp        mcopy
iivanova@dk4n31:~$ mc
```

### 2.3. Файловая структура GNU Linux: каталоги и файлы

Файловая система определяет способ организации, хранения и именования

данных на носителях информации в компьютерах и представляет собой иерархическую структуру в виде вложенных друг в друга каталогов (директорий), содержащих все файлы. В ОС Linux каталог, который является “вершиной” файловой системы, называется корневым каталогом, обозначается символом / и содержит все остальные каталоги и файлы.

В большинстве Linux-систем поддерживается стандарт иерархии файловой системы (Filesystem Hierarchy Standard, FHS), унифицирующий местонахождение файлов и каталогов. Это означает, что в корневом каталоге находятся только подкаталоги со стандартными именами и типами данных, которые могут попасть в тот или иной каталог. Так, в любой Linux-системе всегда есть каталоги /etc, /home, /usr, /bin и т.п. В табл. 1.1 приведено краткое описание нескольких каталогов.

*Таблица 1.1. Описание некоторых каталогов файловой системы GNU Linux*

Каталог	Описание
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям (например: cat, ls, cp)
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей, таких как CD-ROM, DVD-ROM, flash
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя; содержит большинство пользовательских приложений и утилит, используемых в многопользовательском режиме; может быть смонтирована по сети только для чтения и быть общей для нескольких машин

Обратиться к файлу, расположенному в каком-то каталоге, можно указав путь к нему. Существует несколько видов путей к файлу:

- **полный или абсолютный путь** — начинается от корня (/), образуется

перечислением всех каталогов, разделённых прямым слешем (/), и завершается именем файла (например, полный путь к файлу addition.txt из каталога user в каталоге home, находящемся в корневом каталоге, будет иметь вид: /home/user/documents/addition.txt;

- **относительный путь** — так же как и полный путь, строится перечислением через (/) всех каталогов, но начинается от текущего каталога (каталога, в котором “находится” пользователь), т.е. пользователь, находясь в каталоге user, может обратиться к файлу addition.txt, указав относительный путь documents/addition.txt.

Таким образом, в Linux если имя объекта начинается с /, то системой это интерпретируется как полный путь, в любом другом случае — как относительный.

В Linux любой пользователь имеет **домашний каталог**, который, как правило, имеет имя пользователя. В домашних каталогах хранятся документы и настройки пользователя. Для обозначения домашнего каталога используется знак тильды (~). При переходе из домашнего каталога знак тильды будет заменён на имя нового текущего каталога.

## 2.4. Базовые команды bash

В операционной системе GNU Linux взаимодействие пользователя с системой обычно осуществляется с помощью командной строки посредством строчного ввода команд. Общий формат команд можно представить следующим образом:

<имя\_команды><разделитель><аргументы>

Первые задачи, которые приходится решать в любой системе это — работа с данными (обычно хранящимися в файлах) и управление работающими в системе программами (процессами). Для получения достаточно подробной информации по каждой из команд используйте команду man, например:

```
user@dk4n31:~$ man ls
```

В таблице 1.2 приведены основные команды взаимодействия пользователя с файловой системой в GNU Linux посредством командной строки.

*Таблица 1.2. Основные команды взаимодействия пользователя с файловой системой*

Команда		Описание
pwd	<b>Print Working Directory</b>	определение текущего каталога
cd	<b>Change Directory</b>	смена каталога
ls	<b>LiSt</b>	вывод списка файлов
mkdir	<b>MaKe DiRectory</b>	создание пустых каталогов
touch		создание пустых файлов
rm	<b>ReMove</b>	удаление файлов или каталогов
mv	<b>MoVe</b>	перемещение файлов и каталогов
cp	<b>CoPy</b>	копирование файлов и каталогов
cat		вывод содержимого файлов

## 2.5. Полезные комбинации клавиш

Для удобства и экономии времени при работе в терминале существует большое количество сокращённых клавиатурных команд.

Клавиши  $\uparrow$  и  $\downarrow$  позволяют увидеть историю предыдущих команд в `bash`. Количество хранимых строк определено в переменной окружения `HISTSIZE`.

Клавиши  $\leftarrow$  и  $\rightarrow$  перемещают курсор влево и вправо в текущей строке, позволяя редактировать команды.

Сочетания клавиш `Ctrl + a` и `Ctrl + e` перемещают курсор в начало и в конец текущей строки. Клавиши `Ctrl + k` удаляет всё от текущей позиции курсора до конца строки, а `Ctrl + w` или `Alt + Backspace` удаляют слово перед курсором.

Сочетание клавиш `Ctrl + d` в пустой строке служит для завершения текущего сеанса. Для завершения выполняющейся в данный момент команды можно использовать `Ctrl + c`. Также данное сочетание отменит редактирование командной строки и вернёт приглашение командной строки. `Ctrl + l` очищает экран.



## ***\*Примечание***

*Просматривая иллюстрации к отчету, Вы можете обратить внимание на то, что в определенный момент приглашение в вводной строке меняется. Это связано с тем, что изначально я работала на платформе полностью виртуально (однако в определенный момент программа перегружилась и перестала работать, поэтому мне пришлось ее переустановить и загрузить на жесткий диск, чтобы впредь такого не происходило, а потому поменялось имя пользователя), а затем с установленной на жесткий диск программой, используя учетную запись Fedora. Прошу понять, так как работа с Linux для меня в новинку.*

## **3. Выполнение лабораторной работы**

### **3.1 Техническое обеспечение**

Характеристики техники:

- 12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz, 16 GB оперативной памяти
- ОС Linux Gentoo (<http://www.gentoo.ru/>)

### **3.2 Перемещение по файловой системе**

Откройте терминал. По умолчанию терминал открывается в домашнем каталоге пользователя, который обозначается символом ~.

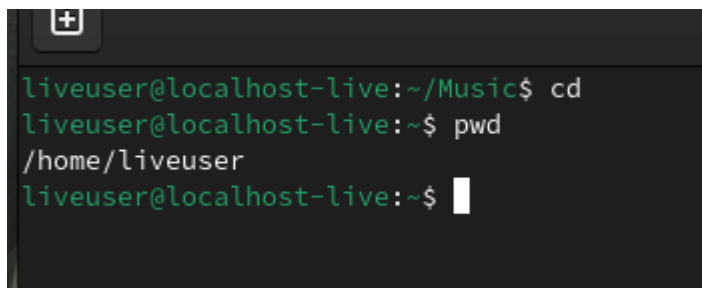
```
user@dk4n31:~$
```

Убедитесь, что Вы находитесь в домашнем каталоге. Если это не так, перейдите в него. Это можно сделать с помощью команды `cd` без аргументов.

```
user@dk4n31:/tmp$ cd user@dk4n31:~$
```

С помощью команды `pwd` узнайте полный путь к Вашему домашнему каталогу (Рисунок 3.1).

```
user@dk4n31:~$ pwd
```



```
liveuser@localhost-live:~/Music$ cd
liveuser@localhost-live:~$ pwd
/home/liveuser
liveuser@localhost-live:~$
```

Рисунок 3.1 – Полный путь к домашнему каталогу

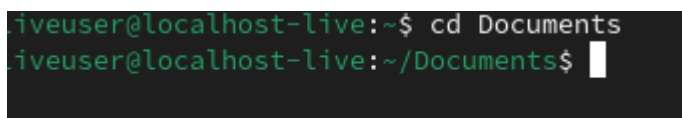
Команда `cd` позволяет сменить текущий каталог на другой, указав путь к нему в качестве параметра. Формат команды:

`cd [путь_к_каталогу]`

Команда `cd` работает как с абсолютными, так и с относительными путями. Перейдите в подкаталог Документы Вашего домашнего каталога указав относительный путь (Рисунок 3.2)

```
user@dk4n31:~$ cd Документы
```

```
user@dk4n31:~/Документы$
```



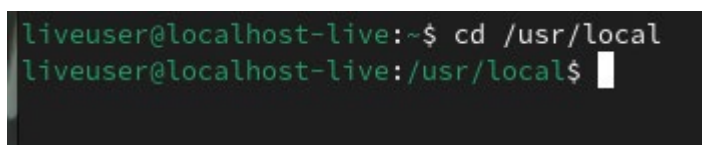
```
liveuser@localhost-live:~$ cd Documents
liveuser@localhost-live:~/Documents$
```

Рисунок 3.2 – Переход к папке Документы (Documents)

Перейдите в каталог `local` – подкаталог `usr` корневого каталога указав абсолютный путь к нему (`/usr/local`): (Рисунок 3.3)

```
user@dk4n31:~$ cd /usr/local
```

```
user@dk4n31:~/usr/local$
```



```
liveuser@localhost-live:~$ cd /usr/local
liveuser@localhost-live:/usr/local$
```

Рисунок 3.3 – Переход в папку `local` с помощью абсолютного пути

Обратите внимание абсолютный путь всегда начинается от корневого каталога (т.е. с символа `/`).

Можно использовать комбинацию `'cd -'` для возвращения в последний посещённый пользователем каталог. А `'cd ..'` используется для перехода на один

каталог выше по иерархии.

Введите последовательно эти команды. В каком каталоге Вы находитесь?

После последовательного ввода команд, мы попадаем в каталог «home» (Рисунок 3.4).

```
liveuser@localhost-live:/usr/local$ cd -  
/home/liveuser  
liveuser@localhost-live:~$ cd ..  
liveuser@localhost-live:/home$
```

Рисунок 3.4 – Последовательный ввод команд «cd -» и «cd ..»

Команда ls выдаёт список файлов указанного каталога и имеет следующий синтаксис: ls [опции] [каталог] [каталог...]

Для просмотра списка файлов текущего каталога может быть использована команда ls без аргументов.

Перейдите в домашний каталог

```
user@dk4n31:~$ cd ~
```

Выведите список файлов Вашего домашнего каталога. (Рисунок 3.5).

```
user@dk4n31:~$ ls
```

```
liveuser@localhost-live:~$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos  
liveuser@localhost-live:~$
```

Рисунок 3.5 – Вывод файлов домашнего каталога

Откройте домашний каталог с помощью файлового менеджера графического окружения Вашей ОС (Рисунок 3.6)

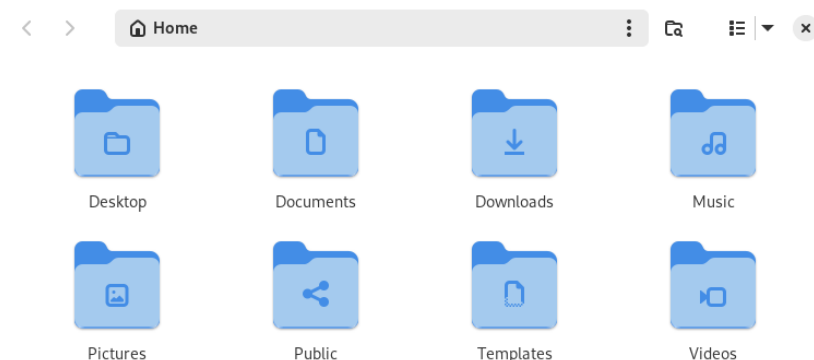
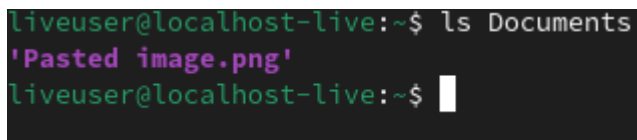


Рисунок 3.6 – Домашний каталог

Убедитесь в том, что список файлов, полученных с помощью команды `ls`, совпадает с файлами, отображающимися в графическом файловом менеджере.

Выведите список файлов подкаталога Документы Вашего домашнего каталога указав относительный путь (Рисунок 3.7)

```
user@dk4n31:~$ ls Документы
```

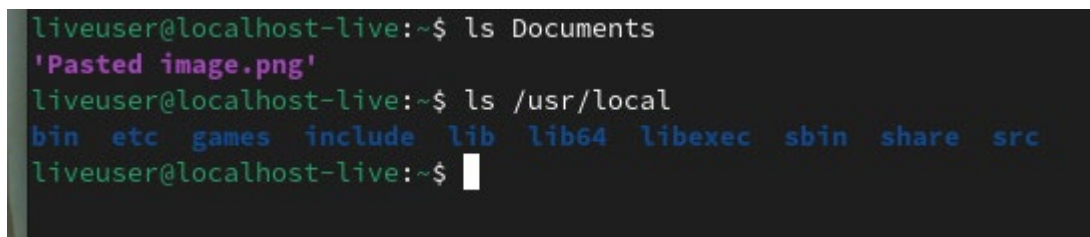


```
liveuser@localhost-live:~$ ls Documents
'Pasted image.png'
liveuser@localhost-live:~$
```

Рисунок 3.7 – Файлы из каталога Документы (Documents)

Выведите список файлов каталога `/usr/local` указав абсолютный путь к нему (Рисунок 3.8):

```
user@dk4n31:~$ ls /usr/local
```



```
liveuser@localhost-live:~$ ls Documents
'Pasted image.png'
liveuser@localhost-live:~$ ls /usr/local
bin  etc  games  include  lib  lib64  libexec  sbin  share  src
liveuser@localhost-live:~$
```

Рисунок 3.8 – Файлы из каталога `local` с помощью абсолютного пути

Для данной команды существует довольно много опций (ключей), ниже дано описание некоторых из них.

Таблица 1.3 – 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, владельцы 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

## Примеры (Рисунок 3.9)

- команда `ls -R` рекурсивно выводит список содержимого текущего каталога;
- команда `ls -ls images/ ..` выводит список файлов каталога `images` и родительского по отношению к текущему каталога, при этом для каждого файла указан номер inode и его размер в килобайтах;
- команда `ls -l images/*.png` выводит список всех файлов в каталоге `images`, чьи имена заканчиваются на `.png`, включая скрытый файл `.png`, если таковой существует.

```
liveuser@localhost-live:~$ ls -a
.          .bashrc      Documents  Pictures   .vboxclient-draganddrop-tty2-control.pid  Videos
..         .bashrc.save  Downloads  Public     .vboxclient-draganddrop-tty2-service.pid
.bash_history .cache        .local     Templates  .vboxclient-hostversion-tty2-control.pid
.bash_logout .config       .mozilla   .vboxclient-clipboard-tty2-control.pid    .vboxclient-seamless-tty2-control.pid
.bash_profile Desktop       Music      .vboxclient-clipboard-tty2-service.pid    .vboxclient-seamless-tty2-service.pid
liveuser@localhost-live:~$

liveuser@localhost-live:~$ ls -R
.:
Desktop Documents Downloads Music Pictures Public Templates Videos

./Desktop:

./Documents:
'Pasted image.png'

./Downloads:

./Music:

./Pictures:
Screenshots

./Pictures/Screenshots:
'Screenshot from 2024-09-14 10-33-09.png' 'Screenshot from 2024-09-14 10-34-29.png' 'Screenshot from 2024-09-14 10-57-14.png'

./Public:

./Templates:

./Videos:
liveuser@localhost-live:~$
```

Рисунок 3.9 – Примеры использования команды `ls` с ключами «-a» и «-R»

## 3.3 Создание пустых каталогов и файлов

Для создания каталогов используется команда `mkdir`. Её синтаксис имеет вид: `mkdir [опции] [каталог...]`

Создайте в домашнем каталоге подкаталог с именем `parentdir`

```
user@dk4n31:~$ cd
```

```
user@dk4n31:~$ mkdir parentdir
```

С помощью команды `ls` проверьте, что каталог создан. (Рисунок 3.10)

```
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ ls
Desktop Documents Downloads Music parentdir Pictures Public Templates Videos
liveuser@localhost-live:~$
```

Рисунок 3.10 – Создание каталога «parentdir»

Создайте подкаталог в существующем каталоге (Рисунок 3.11):

```
user@dk4n31:~$ mkdir parentdir/dir
```

```
liveuser@localhost-live:~$ mkdir parentdir/dir
liveuser@localhost-live:~$ ls parentdir
dir
liveuser@localhost-live:~$
```

Рисунок 3.11 – Создание подкаталога «dir» через каталог «parentdir»

При задании нескольких аргументов создаётся несколько каталогов (Рисунок 3.12):

```
user@dk4n31:~$ cd parentdir
```

```
user@dk4n31:~$ mkdir dir1 dir2 dir3
```

```
liveuser@localhost-live:~$ cd parentdir
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$ ls
dir dir1 dir2 dir3
```

Рисунок 3.12 – Создание нескольких каталогов: «dir1», «dir2», «dir3»

Если требуется создать подкаталог в каталоге, отличном от текущего, то путь к нему требуется указать в явном виде:

```
user@dk4n31:~$ mkdir ~/newdir
```

Эта команда должна создать каталог newdir в домашнем каталоге (~). Проверьте это с помощью команды (Рисунок 3.13)

```
user@dk4n31:~$ ls ~
```

```
liveuser@localhost-live:~$ mkdir ~/newdir
liveuser@localhost-live:~$ ls ~
Desktop Documents Downloads Music newdir parentdir Pictures Public Templates Videos
liveuser@localhost-live:~$
```

Рисунок 3.13 – Создание каталога «newdir» в домашнем каталоге

Опция – parents (краткая форма -p) позволяет создавать иерархическую цепочку подкаталогов, создавая все промежуточные каталоги. Создайте следующую последовательность вложенных каталогов newdir/dir1/dir2 в домашнем каталоге

```
user@dk4n31:~$ mkdir -p ~/newdir/dir1/dir2
```

```
liveuser@localhost-live:~$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~$ ls ~/newdir/dir1
dir2
```

Рисунок 3.14 – Создание цепочки каталогов

Для создания файлов может быть использована команда touch, которая имеет следующий синтаксис (Рисунок 3.15):

touch [опции] файл [файл...]

Создайте файл test.txt в каталоге ~/newdir/dir1/dir2

```
user@dk4n31:~$ touch ~/newdir/dir1/dir2/test.txt
```

Проверьте наличие файла с помощью команды

```
user@dk4n31:~$ ls ~/newdir/dir1/dir2
```

```
liveuser@localhost-live:~$ touch ~/newdir/dir1/dir2/test.txt
liveuser@localhost-live:~$ ls ~/newdir/dir1/dir2
test.txt
```

Рисунок 3.15 – Создание файла через цепочку каталогов

### 3.4 Перемещение и удаление файлов или каталогов

Команда rm удаляет файлы и (или) каталоги и имеет следующий синтаксис:

rm [опции] [файл|каталог...]

Опции команды rm:

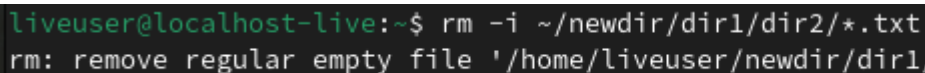
- -r или -R: рекурсивное удаление (это обязательная опция для удаления любого каталога, пустого или содержащего файлы и (или) подкаталоги);
- -i: запрос подтверждения перед удалением;
- -v: вывод подробной информации при выполнении команды;
- -f: принудительное удаление файлов или каталогов.

Для удаления пустых каталогов можно воспользоваться командой rmdir.

Запросив подтверждение на удаление каждого файла в текущем каталоге,

удалите в подкаталоге /newdir/dir1/dir2/ все файлы с именами, заканчивающимися на .txt (Рисунок 3.16):

```
user@dk4n31:~$ rm -i ~/newdir/dir1/dir2/*.txt
```



```
liveuser@localhost-live:~$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/liveuser/newdir/dir1/dir2/test.txt'? █
```

Рисунок 3.16 – Удаление всех файлов в формате .txt

Рекурсивно удалите из текущего каталога без запроса подтверждения на удаление каталог newdir, а также файлы, чьи имена начинаются с dir в каталоге parentdir (Рисунок 3.17):

```
user@dk4n31:~$ rm -R ~/newdir ~/parentdir/dir*
```



```
liveuser@localhost-live:~$ rm -R ~/newdir ~/parentdir/dir*
liveuser@localhost-live:~$ █
```

Рисунок 3.17 – Рекурсивное удаление пустых каталогов

Команда rm удаляет файлы безвозвратно, и не существует способа для их восстановления. Команда mv служит для перемещения файлов и каталогов и имеет следующий синтаксис: mv [опции] [файл|каталог...]

Некоторые опции:

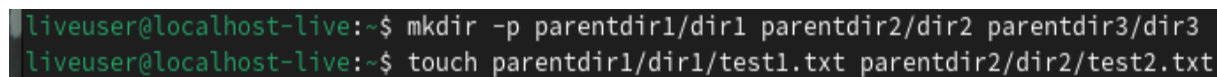
- -f: принудительное выполнение операции (предупреждение не будет выводиться даже при перезаписи существующего файла);
- -i: запрашивается подтверждение перед перезаписью существующего файла;
- -v: подробный режим, который сообщает обо всех изменениях и действиях при выполнении команды. Команда cp копирует файлы и каталоги и имеет следующий синтаксис: cp [опции] [файл|каталог ...] Некоторые опции команды cp:
- -R: рекурсивное копирование; является обязательной опцией для копирования каталогов;
- -i: запрос подтверждения перед перезаписью любых файлов;
- -f: заменяет любые существующие файлы без запроса подтверждения;
- -v: подробный режим, сообщает обо всех изменениях и действиях.



Для демонстрации работы команд `cp` и `mv` преведем следующие примеры.

Создайте следующие файлы и каталоги в домашнем каталоге (Рисунок 3.18):

```
user@dk4n31:~$ cd
user@dk4n31:~$ mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3
user@dk4n31:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2/test2.txt
```



```
liveuser@localhost-live:~$ mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3/dir3
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2/test2.txt
```

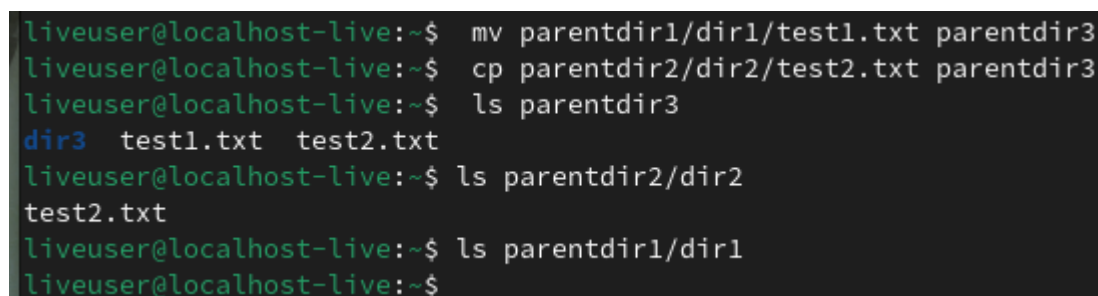
Рисунок 3.18 – Создание каталогов и файлов в этих каталогах

Используя команды `cp` и `mv` файл `test1.txt` скопируйте, а `test2.txt` переместите в каталог `parentdir3`:

```
user@dk4n31:~$ mv parentdir1/dir1/test1.txt parentdir3
user@dk4n31:~$ cp parentdir2/dir2/test2.txt parentdir3
```

С помощью команды `ls` проверьте корректность выполненных команд (Рисунок 3.19)

```
user@dk4n31:~$ ls parentdir3
test1.txt test2.txt
user@dk4n31:~$ ls parentdir1/dir1
user@dk4n31:~$ ls parentdir2/dir2 test2.txt
```



```
liveuser@localhost-live:~$ mv parentdir1/dir1/test1.txt parentdir3
liveuser@localhost-live:~$ cp parentdir2/dir2/test2.txt parentdir3
liveuser@localhost-live:~$ ls parentdir3
dir3 test1.txt test2.txt
liveuser@localhost-live:~$ ls parentdir2/dir2
test2.txt
liveuser@localhost-live:~$ ls parentdir1/dir1
liveuser@localhost-live:~$
```

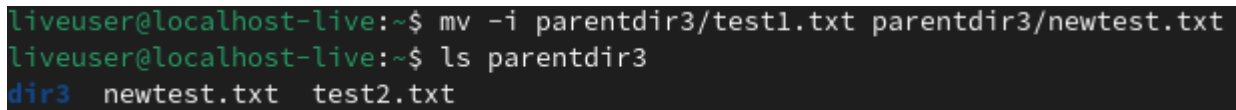
Рисунок 3.19 – Перемещение файлов и их копирование

Также команда `mv` может быть использована для переименования файлов и каталогов, а команда `cp` позволяет сделать копию файла с новым именем

Переименуйте файл `test1.txt` из каталога `parentdir3` в `newtest.txt`, запрашивая подтверждение перед перезаписью (Рисунок 3.20):

```
user@dk4n31:~$ ls parentdir3
test1.txt test2.txt
```

```
user@dk4n31:~$ cp parentdir3/test2.txt parentdir3/subtest2.txt
user@dk4n31:~$ mv -i parentdir3/test1.txt parentdir3/newtest.txt
user@dk4n31:~$ ls parentdir3
newtest.txt subtest2.txt test2.txt
```

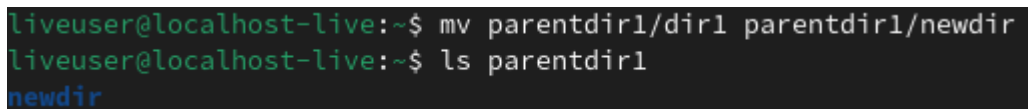
A terminal window with a dark background and light green text. It shows the execution of two commands: 'mv -i parentdir3/test1.txt parentdir3/newtest.txt' and 'ls parentdir3'. The output of the second command is 'dir3 newtest.txt test2.txt', where 'dir3' is highlighted in blue.

```
liveuser@localhost-live:~$ mv -i parentdir3/test1.txt parentdir3/newtest.txt
liveuser@localhost-live:~$ ls parentdir3
dir3 newtest.txt test2.txt
```

Рисунок 3.20 – Переименование файла

Переименуйте каталог dir1 в каталоге parentdir1 в newdir (Рисунок 3.21):

```
user@dk4n31:~$ cd parentdir1
user@dk4n31:~/parentdir1$ ls
dir1
user@dk4n31:~/parentdir1$ mv dir1 newdir
user@dk4n31:~/parentdir1$ ls
newdir
```

A terminal window with a dark background and light green text. It shows the execution of two commands: 'mv parentdir1/dir1 parentdir1/newdir' and 'ls parentdir1'. The output of the second command is 'newdir', which is highlighted in blue.

```
liveuser@localhost-live:~$ mv parentdir1/dir1 parentdir1/newdir
liveuser@localhost-live:~$ ls parentdir1
newdir
```

Рисунок 3.21 – Переименование каталога

### 3.5 Команда cat: вывод содержимого файлов

Команда cat объединяет файлы и выводит их на стандартный вывод — обычно это экран (Рисунок 3.22):

```
user@dk4n31:~$ cat /etc/hosts
#
# /etc/hosts: static lookup table for host names
#
# 127.0.0.1 localhost.localdomain localhost
# End of file
```

```
liveuser@localhost-live:~$ cat /etc/hosts
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.example.org foo
# 192.168.1.13 bar.example.org bar
```

Рисунок 3.22 – Осуществление команды “cat”

## 4. Самостоятельная работа

1. Воспользовавшись командой `pwd`, узнайте полный путь к своей домашней директории (Рисунок 4.1).

```
liveuser@localhost-live:~$ pwd
/home/liveuser
```

Рисунок 4.1 – Полный путь к домашней директории

2. Введите следующую последовательность команд (Рисунок 4.2)

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

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

```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir tmp
liveuser@localhost-live:~$ cd tmp
liveuser@localhost-live:~/tmp$ pwd
/home/liveuser/tmp
liveuser@localhost-live:~/tmp$ cd /tmp
liveuser@localhost-live:/tmp$ pwd
/tmp
liveuser@localhost-live:/tmp$
```

Рисунок 4.2 – Выполнение последовательности команд

Когда в команду «`cd`» мы добавляем `/`, то она, при переходе на 1 каталог, используется как абсолютный путь. Таким образом, после выполнения данной команды, путь терминала начнется с папки назначения.

3. Пользуясь командами `cd` и `ls`, посмотрите содержимое корневого каталога, домашнего каталога, каталогов `/etc` и `/usr/local` (Рисунки 4.3, 4.4, 4.5, 4.6.)

```
liveuser@localhost-live:/tmp$ cd
liveuser@localhost-live:~$ ls /
afs bin boot dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
liveuser@localhost-live:~$
```

Рисунок 4.3 – Просмотр содержимого корневого каталога и его файлы с помощью команды `ls /`

```
liveuser@localhost-live:~$ ls ~/
Desktop Downloads parentdir parentdir2 Pictures Templates Videos
Documents Music parentdir1 parentdir3 Public tmp
liveuser@localhost-live:~$
```

Рисунок 4.4 – Просмотр содержимого домашнего каталога и его файлы с помощью команды `ls ~/`

```
liveuser@localhost-live:~$ ls /etc
abrt fedora-release ld.so.cache PackageKit sos
adjtime filesystems ld.so.conf pam.d speech-dispatcher
aliases firefoxf ld.so.conf.d paperspecs ssh
alsa firewallld libaudit.conf passim.conf ssl
alternatives flatpak libblockdev passwd sssd
anaconda fonts libb1bverbs.d passwd statetab.d
anthy-unicode.conf fprintd.conf libn1 subgid
asound.conf fstab libreport pinforc subgid
audit fuse.conf libssh pkcs11 subuid
authselect fwupd libuser.conf pkgconfig sudo.conf
avahi gcrypt libvirt polkit-1 sudoers
bash_completion.d gdbinit locale.conf plymouth sudoers.d
bashrc gdbinit.d localtime pm swid
bindresvport.blacklist gdm login.defs popt.d swtpm-localca.conf
binfmt.d geoclue logrotate.conf rpm swtpm-localca.options
bluezdbus glvnd logrotate.d printcap sysconfig
brlapi.key group machine-id profile sysctl.conf
brltty.conf groff mailcap protocols sysctl.d
ceph group- makedumpfile.conf.sample pulse systemd
chkconfig.d grub2.cfg mcelog qemu system-release
chromium grub2-efi.cfg mdevctl.d rc0.d system-release-cpe
cifs-utils grub.d mime.types rc1.d terminfo
colord gshadow mke2fs.conf rc2.d thermald
containers gss modprobe.d rc3.d tmpfiles.d
credstore gssproxy modules-load.d rc4.d tpm2-tss
crypto-encrypted host.conf motd rc5.d Trolltech.conf
crypto-policies hosts motd.d rc6.d trusted-key.key
csh.cshrc htpd mtab rc.d udev
csh.login httpd mttools.conf rc.d-rc.conf.d udisk2
cups cupsd my.cnf redhat-release unbound
dbus-1 idmapd.conf my.cnf.d request-key.conf updatedb.conf
dbusconf ImageMagick-7 nanorc request-key.d UPower

debuginfo init.d ndctl resolv.conf uresourced.conf
default inittab ndctl.conf.d rpc usb_modeswitch.conf
depmod.d inputrc netconfig rpm vconsole.conf
dhcp ipp-usb NetworkManager rsyncd.conf vdpau_wrapper.cfg
DIR_COLORS iscsi networks rwtab.d virc
DIR_COLORS.lightbgcolor issue nfs.conf rygel.conf vmware-tools
dleyna-server-service.conf issue.d nfsmount.conf samba vpl
dnf issue.net nftables sane.d vpnc
dnsmasq.conf java nilfs_cleaner.conf sasl2 vulkan
dnsmasq.d jvm nsswitch.conf security whois.conf
dracut.conf jvm-common nvme selinux wireplumber
dracut.conf.d kdump openldap services wpa_supplicant
egl kdump.conf opensc.conf sestatus.conf X11
environment kernel opencsc-x86_64.conf sgml xattr.conf
ethertypes keys openvpn shadow xdg
exports keyutils opt shadow.xml
exports.d krb5.conf os-release shells yum.repos.d
favicon.png krb5.conf.d ostree skel zfs-fuse
liveuser@localhost-live:~$
```

Рисунок 4.5 – Просмотр содержимого каталога `/etc` и его файлы с помощью команды `ls`

```
liveuser@localhost-live:~$ ls /usr/local
bin etc games include lib lib64 libexec sbin share src
liveuser@localhost-live:~$
```

Рисунок 4.6 – Просмотр содержимого каталога `/usr/local` и его файлы с помощью команды `ls`

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 создайте файлы text1.txt, text2.txt, text3.txt. Пользуясь командой ls, убедитесь, что все действия выполнены успешно – каталоги и файлы созданы (Рисунок 4.7).

```
liveuser@localhost-live:~$ mkdir -p temp labs/lab1 labs/lab2 labs/lab3
liveuser@localhost-live:~$ touch temp/text1.txt temp/text2.txt temp/text3.txt
liveuser@localhost-live:~$ ls
Desktop  Documents  Downloads  labs  Music  parentdir  parentdir1  parentdir2
parentdir3  Pictures  Public  temp  Templates  tmp  Videos
liveuser@localhost-live:~$ ls labs
lab1  lab2  lab3
liveuser@localhost-live:~$ ls temp
text1.txt  text2.txt  text3.txt
liveuser@localhost-live:~$
```

Рисунок 4.7 – Создание файлов и цепочки каталогов с помощью команды “mkdir” и ключа “-p”

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 (Рисунок 4.8).

```
irina_panyavkina@fedora:~$ cat temp/text1.txt temp/text2.txt temp/text3.txt
Ирина
Панявкина
НКАбд-04-24
irina_panyavkina@fedora:~$
```

Рисунок 4.8 – Вывод содержимого файлов «text1.txt», «text2.txt», «text3.txt»

6. Скопируйте все файлы, чьи имена заканчиваются на .txt, из каталога ~/temp в каталог labs. После этого переименуйте файлы каталога labs и переместите их: text1.txt переименуйте в firstname.txt и переместите в подкаталог lab1, text2.txt в lastname.txt в подкаталог lab2, text3.txt в id-group.txt в подкаталог lab3. Пользуясь командами ls и cat, убедитесь, что все действия выполнены верно (Рисунок 4.9).

```
irina_panyavkina@fedora:~/temp$ mv text1.txt firstname.txt
irina_panyavkina@fedora:~/temp$ mv firstname.txt lab1
irina_panyavkina@fedora:~/temp$ mv text2.txt lastname.txt
irina_panyavkina@fedora:~/temp$ mv lastname.txt lab2
irina_panyavkina@fedora:~/temp$ mv text3.txt id-group.txt
irina_panyavkina@fedora:~/temp$ mv id-group.txt lab3
irina_panyavkina@fedora:~/temp$ cat lab1 lab2 lab3
Ирина
Панявкина
НКАбд-04-24
irina_panyavkina@fedora:~/temp$
```

Рисунок 4.9 – Переименование и перемещение текстовых файлов

7. Удалите все созданные в ходе выполнения лабораторной работы файлы и каталоги (Рисунок 4.10).

```
irina_panyavkina@fedora:~$ ls
Desktop Documents Downloads labs Music parentdir Pictures Public temp Templates tmp Videos
irina_panyavkina@fedora:~$ rm -r labs temp tmp parentdir
irina_panyavkina@fedora:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
irina_panyavkina@fedora:~$
```

Рисунок 4.10 – Удаление всех созданных ранее файлов и каталогов

## 5. Вопросы для самопроверки

1. Дайте определение командной строки. Приведите примеры.

Командная строка — текстовый интерфейс пользователя для взаимодействия с операционной системой компьютера и/или другим программным обеспечением с помощью команд, вводимых с клавиатуры. В системе Linux используется терминал, а в Windows – командная строка.

2. Как получить информацию об интересующей вас команде?

Для получения информации об интересующей команде, необходимо ввести команду «-help».

3. Чем относительный путь к файлу отличается от абсолютного?

Относительный путь – путь от текущего каталога, абсолютный путь – путь от корневого каталога.

4. Как определить абсолютный путь к текущей директории?

Определить абсолютный путь к текущей директории можно с помощью команды «pwd»

5. При помощи каких команд можно удалить файл и каталог? Можно ли это

сделать одной и той же командой?

Файл и каталог возможно удалить при помощи команды `rm` с ключом «-R». Также, файлы удаляются командой «`rm`», а каталоги – командой «`rmdir`».

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

Запустить несколько команд в одной строке можно воспользовавшись следующими служебными символами: «`;`», «`&&`», «`||`». Символ «`;`» осуществляет команды последовательно, независимо от успешности выполнения предыдущей. Символ «`&&`», обозначающий логическое «И», осуществляет команды последовательно только если предыдущая была выполнена с успехом. Символ «`||`», обозначающий логическое «ИЛИ», осуществляет команды последовательно если предыдущая не была выполнена.



```
irina_panyavkina@fedora:~$ mkdir example1 && rm -R example1; mkdir example1; rm -R example1
irina_panyavkina@fedora:~$
```

*Пример (Рисунок 5.1)*

Рисунок 5.1 – Результат создания и удаления папки «example1» с помощью использования служебных символов «`;`» и «`&&`» в одной строке двумя командами – «`mkdir`» и «`rm`» с ключом «-R».

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

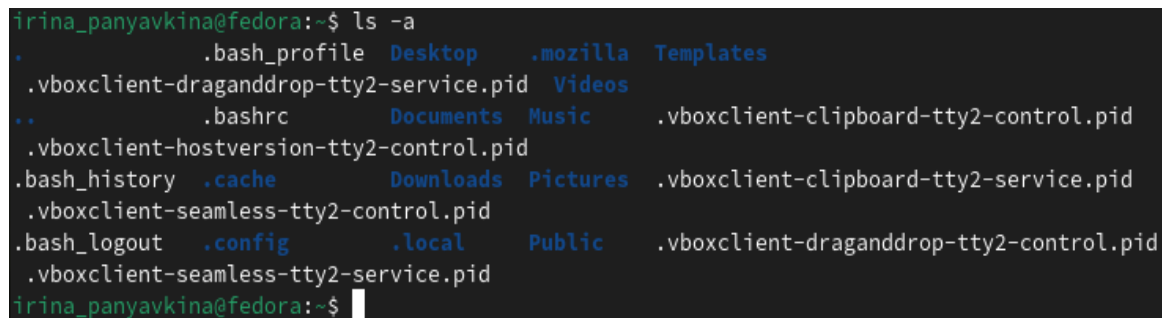
Если используется опция `-l` в команде `ls`, то отображается дополнительная информация о содержащихся файлах – размеры файлов, их группу, время создания, владельца, информацию о правах доступа и о том, является ли файл символьной ссылкой на другой файл и на какой именно.

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

Отобразить информацию о скрытых файлах возможно благодаря команде



«ls» с ключом «-a» (Рисунок 5.2)



```
irina_panyavkina@fedora:~$ ls -a
.          .bash_profile Desktop  .mozilla  Templates
.vboxclient-draganddrop-tty2-service.pid Videos
..         .bashrc      Documents Music      .vboxclient-clipboard-tty2-control.pid
.vboxclient-hostversion-tty2-control.pid
.bash_history .cache      Downloads Pictures   .vboxclient-clipboard-tty2-service.pid
.vboxclient-seamless-tty2-control.pid
.bash_logout .config     .local   Public    .vboxclient-draganddrop-tty2-control.pid
.vboxclient-seamless-tty2-service.pid
irina_panyavkina@fedora:~$
```

Рисунок 5.2 – Отображение информации о скрытых файлах в домашнем каталоге

9. Какая клавиша или комбинация клавиш служит для автоматического дополнения вводимых команд?

Клавиша «TAB» служит для автоматического дополнения вводимых команд.

## 6. Выводы

Я приобрела практические навыки работы с операционной системой на уровне командной строки (организация файловой системы, навигация по файловой системе, создание и удаление файлов и директорий).



## 7. Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix).
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).