

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: Шабакowa Кaрина Баировна

с\б:1132242469

Группа:НКАбд-05-24

МОСКВА

2024г.

Введение:

1.Цель работы

2.Задание

3.Теоретическое введение

4. Выполнение лабораторной работы

5.Выводы

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. Задание

1. Перемещение по файловой системе
2. Создание пустых каталогов и файлов
3. Перемещение и удаление файлов или каталогов
4. Команда cat: вывод содержимого файлов
5. Задание для самостоятельной работы

3.Теоретическое введение

3.1. 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).

3.2. 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, 1 Демидова А. В. Архитектура ЭВМ 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 2 Демидова

А. В. Архитектура ЭВМ вариантов завершения команды. Нажав клавишу Tab ещё раз, можно получить список имён, начинающихся с mc: iivanova@dk4n31:~\$ mc mc mcd mcredit mclasserase mcookie mcview mcat mcdiff mcheck mcomp mcopy iivanova@dk4n31:~\$ mc

Более подробно о работе в операционной системе Linux см., например, в [13; 16].

3.3 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 систему
/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

Демидова А. В. 3 Архитектура ЭВМ

Каталог	Описание
/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.

3.4. Базовые команды *bash* В операционной системе GNU Linux взаимодействие пользователя с

системой обычно осуществляется с помощью командной строки посредством построчного ввода команд.

Общий формат команд можно представить следующим образом: 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. 4 Демидова А. В. Архитектура ЭВМ Таблица 1.2. Основные команды взаимодействия пользователя с файловой системой

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

Более подробно о работе в *bash* см. в [2; 5; 6; 8].

3.5. Полезные комбинации клавиш Для удобства и экономии времени при работе в терминале

существует большое количество сокращённых клавиатурных команд. Клавиши ↑ и ↓ позволяют увидеть историю предыдущих команд в *bash*. Количество хранимых строк определено в переменной окружения HISTSIZE. Клавиши ← и → 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.

4. Выполнение лабораторной работы

Перемещение по файловой системе

Открыла терминал. С помощью команды pwd, узнала путь домашнего каталога.

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

1.1Использую pwd

Перешла в подкаталог Документы домашнего каталога указав относительный путь

```
bash: ln-l: command not found...  
liveuser@localhost-live:~$ cd Documents  
liveuser@localhost-live:~/Documents$ ln  
ln: missing file operand  
Try 'ln --help' for more information.  
liveuser@localhost-live:~/Documents$ ls  
liveuser@localhost-live:~/Documents$ ls -l  
total 0
```

1.2 использовала cd

Перешла в каталог local – подкаталог usr корневого каталога указав абсолютный путь к нему (/usr/local):

```
total 0  
liveuser@localhost-live:~/Documents$ cd /usr/local  
liveuser@localhost-live:/usr/local$ cd  
liveuser@localhost-live:~$ cd /usr/local
```

1.3 команда cd

Вывела список файлов подкаталога Документы домашнего каталога указав относительный путь и

Вывела список файлов каталога /usr/local указав абсолютный путь к нему:

```
liveuser@localhost-live:~ x liveuser@localhost-live:/tmp
liveuser@localhost-live:~$ ls Documents
liveuser@localhost-live:~$ ls Ljrevtyns
ls: cannot access 'Ljrevtyns': No such file or directory
liveuser@localhost-live:~$ ls /usr/local
bin  etc  games  include  lib  lib64  libexec  sbin  share  src
liveuser@localhost-live:~$ mkdir
```

1.4 команда ls

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

Создание пустых каталогов и файлов

Создаю в домашнем каталоге подкаталог с именем parentdir

С помощью команды ls проверяю, что каталог создан. Создаю подкаталог в существующем каталоге.

При задании нескольких аргументов создаётся несколько каталогов

Если требуется создать подкаталог в каталоге, отличном от текущего, указываю в явном виде

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

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

Для создания файлов может быть использована команда touch, которая имеет следующий синтаксис: touch

[опции] файл [файл...] Создаю файл test.txt в каталоге Проверю наличие файла с помощью команды ls


```
liveuser@localhost-live:~$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
liveuser@localhost-live:~$ mkmdir parentdir
bash: mkmdir: command not found...
liveuser@localhost-live:~$ mkdir parentdir
liveuser@localhost-live:~$ cd parentdir
liveuser@localhost-live:~/parentdir$ mkdir dir1 dir2 dir3
liveuser@localhost-live:~/parentdir$ mkdir ~/newdir
liveuser@localhost-live:~/parentdir$ ls
dir1  dir2  dir3
liveuser@localhost-live:~/parentdir$ ls ~
Desktop  Downloads  newdir  Pictures  Templates
Documents  Music  parentdir  Public  Videos
liveuser@localhost-live:~/parentdir$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~/parentdir$ mkdir -p ~/newdir/dir1/dir2
liveuser@localhost-live:~/parentdir$ cd
liveuser@localhost-live:~$ touch ~/newdir/dir1/dir2/text.txt
liveuser@localhost-live:~$ ls ~/newdir/dir1/dir2
```

1.5 команды mkdir,ls,touch

```
liveuser@localhost-live:~$ touch ~/newdir/dir1/dir2/text.txt
liveuser@localhost-live:~$ ls ~/newdir/dir1/dir2
text.txt
liveuser@localhost-live:~$ rm -i ~/newdir/dir1/dir2/*
```

Продолжение 1.5

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

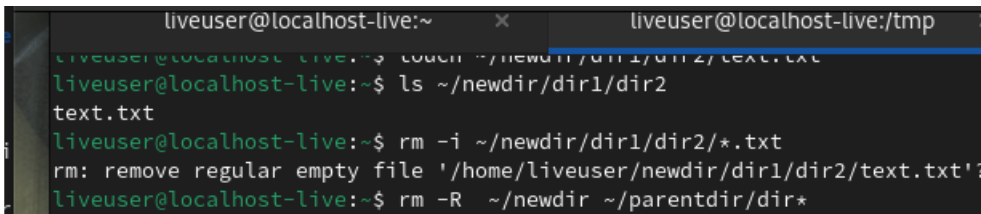
каждого файла в текущем каталоге, удалите в подкаталоге `/newdir/dir1/dir2/` все файлы с именами, заканчивающимися на `.txt`: `user@dk4n31:~$ rm -i ~/newdir/dir1/dir2/*.txt` Рекурсивно удалите из текущего каталога без запроса подтверждения на удаление каталог `newdir`, а также файлы, чьи имена начинаются с `dir` в

каталоге `parentdir`: `user@dk4n31:~$ rm -R ~/newdir ~/parentdir/dir*` 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: `ср [опции] [файл|каталог ...]` Некоторые опции команды `ср`: • `-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.

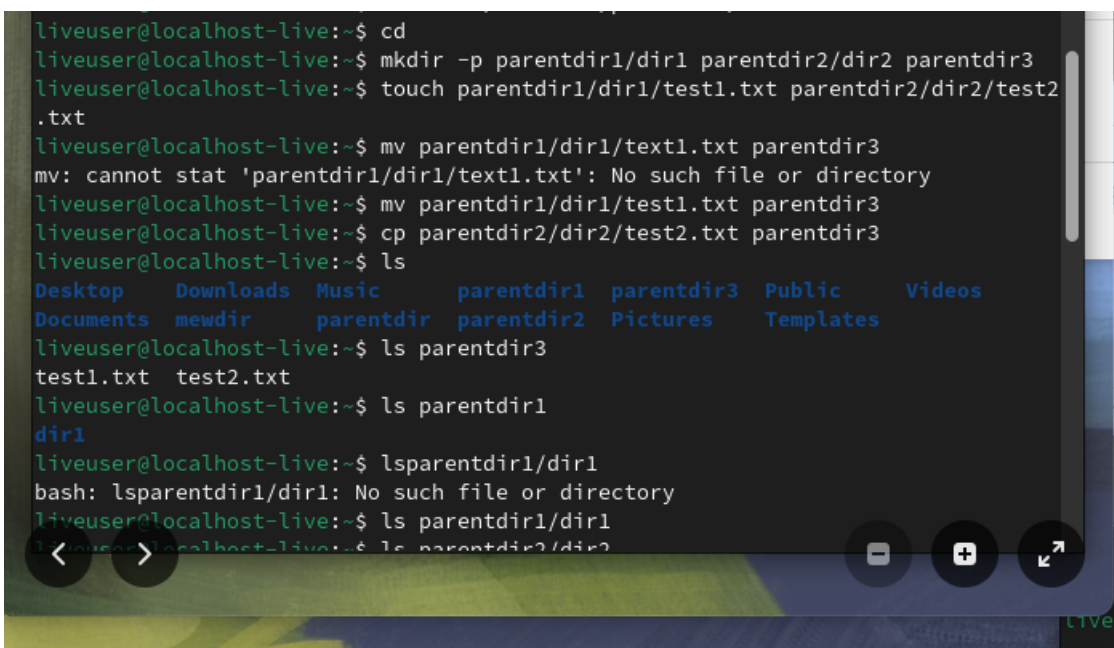


```
liveuser@localhost-live:~$ cp ~/newdir/dir1/dir2/text.txt
liveuser@localhost-live:~$ ls ~/newdir/dir1/dir2
text.txt
liveuser@localhost-live:~$ rm -i ~/newdir/dir1/dir2/*.txt
rm: remove regular empty file '/home/liveuser/newdir/dir1/dir2/text.txt'?
liveuser@localhost-live:~$ rm -R ~/newdir ~/parentdir/dir*
```

1.6 команды ls, rm

Для демонстрации работы команд `ср` и `mv` приведем следующие примеры. Создаю следующие файлы и каталоги в домашнем каталоге. Используя команды `ср` и `mv` файл `test1.txt` скопируйте, а `test2.txt` переместила в каталог `parentdir3`.

С помощью команды `ls` проверяю корректность выполненных команд.



```
liveuser@localhost-live:~$ cd
liveuser@localhost-live:~$ mkdir -p parentdir1/dir1 parentdir2/dir2 parentdir3
liveuser@localhost-live:~$ touch parentdir1/dir1/test1.txt parentdir2/dir2/test2
.txt
liveuser@localhost-live:~$ mv parentdir1/dir1/test1.txt parentdir3
mv: cannot stat 'parentdir1/dir1/test1.txt': No such file or directory
liveuser@localhost-live:~$ mv parentdir1/dir1/test1.txt parentdir3
liveuser@localhost-live:~$ cp parentdir2/dir2/test2.txt parentdir3
liveuser@localhost-live:~$ ls
Desktop  Downloads  Music      parentdir1  parentdir3  Public      Videos
Documents  mewdir     parentdir  parentdir2  Pictures    Templates
liveuser@localhost-live:~$ ls parentdir3
test1.txt  test2.txt
liveuser@localhost-live:~$ ls parentdir1
dir1
liveuser@localhost-live:~$ ls parentdir1/dir1
bash: ls parentdir1/dir1: No such file or directory
liveuser@localhost-live:~$ ls parentdir1/dir1
liveuser@localhost-live:~$ ls parentdir2/dir2
```

1.7 демонстрация команд ср и mv

```
liveuser@localhost-live:~ x liveuser@localhost-live:/tmp x
liveuser@localhost-live:~$ ls parentdir1/dir1
liveuser@localhost-live:~$ ls parentdir2/dir2
test2.txt
```

Продолжение 1.7

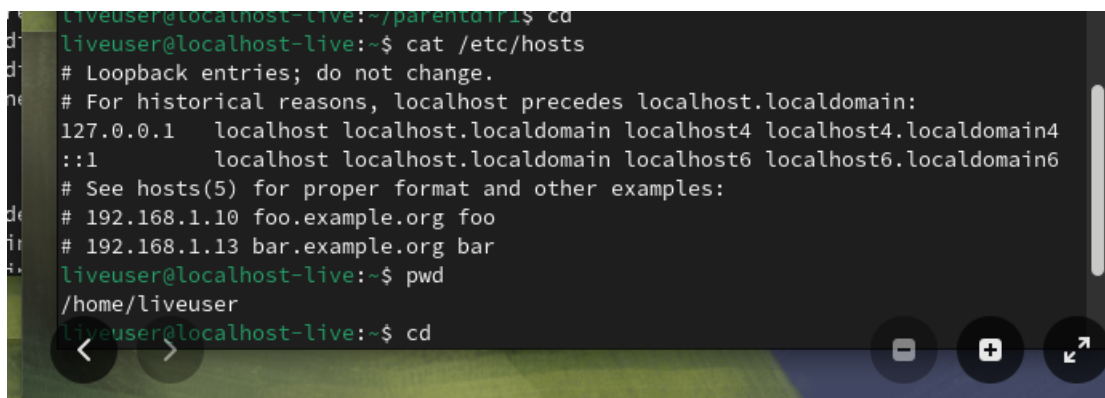
Также команда `mv` может быть использована для переименования файлов и каталогов, а команда `cp` позволяет сделать копию файла с новым именем. Переименую файл `test1.txt` из каталога `parentdir3` в `newtest.txt`, запрашивая подтверждение перед перезаписью и переименую каталог `dir1` в каталоге `parentdir1` в `newdir`:

```
liveuser@localhost-live:~ x liveuser@localhost-live:/tmp x
liveuser@localhost-live:~$ ls parentdir1/dir1
liveuser@localhost-live:~$ ls parentdir2/dir2
test2.txt
liveuser@localhost-live:~$ ls parentdir3
test1.txt test2.txt
liveuser@localhost-live:~$ cp parentdir3/test2.txt parentdir3/subtest2.txt
liveuser@localhost-live:~$ mv -i parentdir3/test1.txt parentdir3/newtest.txt
liveuser@localhost-live:~$ ls parentdir3
newtest.txt subtest2.txt test2.txt
liveuser@localhost-live:~$ cd parentdir1
bash: cd: parentdir1: No such file or directory
liveuser@localhost-live:~$ cd parentdir1
liveuser@localhost-live:~/parentdir1$ ls dir 1
ls: cannot access 'dir': No such file or directory
ls: cannot access '1': No such file or directory
liveuser@localhost-live:~/parentdir1$ ls dir1
liveuser@localhost-live:~/parentdir1$ mv dir1 newdir
liveuser@localhost-live:~/parentdir1$ ls newdir
```

1.8 команды `cp`, `mv` и `ls`

Команда cat: вывод содержимого файлов Команда cat объединяет файлы и выводит их на стандартный вывод

(обычно это экран):

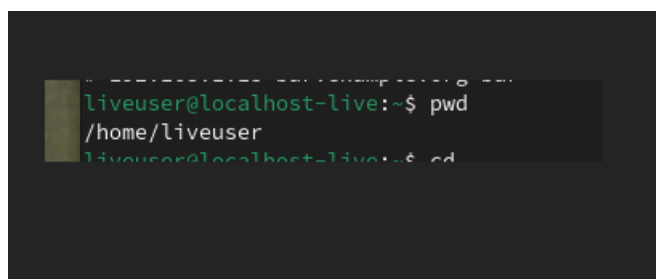


```
liveuser@localhost-live:~/parentdir1$ cd
liveuser@localhost-live:~$ cat /etc/hosts
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain 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
liveuser@localhost-live:~$ pwd
/home/liveuser
liveuser@localhost-live:~$ cd
```

1.9 команда cat

5.Задание для самостоятельной работы

1. Воспользовавшись командой pwd, узнайте полный путь к своей домашней директории.



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

2. Введите следующую последовательность команд

```
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$
```

Объясняю, когда находимся в домашнем каталоге ~/tmp\$ выводит: /home/liveuser/tmp , /home/liveuser – домашний каталог

Когда мы сразу вводим cd /tmp: то мы не находимся в домашнем каталоге:/tmp\$, вывод: /tmp – без домашнего каталога

3. Пользуясь командами cd и ls, посмотрите содержимое корневого каталога, домашнего каталога, каталогов /etc и /usr/local.

```
liveuser@localhost-live:/usr
liveuser@localhost-live:~$ pwd
/home/liveuser
liveuser@localhost-live:~$ /$
bash: /$: No such file or directory
liveuser@localhost-live:~$ /etc
bash: /etc: Is a directory
liveuser@localhost-live:~$ /usr/local
bash: /usr/local: Is a directory
liveuser@localhost-live:~$ cd /etc
liveuser@localhost-live:/etc$ pwd
/etc
liveuser@localhost-live:/etc$ cd /usr/local
liveuser@localhost-live:/usr/local$ pwd
/usr/local
liveuser@localhost-live:/usr/local$
```

Посмотрели содержимое с помощью команд cd и 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).

```
liveuser@localhost-live:/usr/local$ mkdir -p ~/temp/lab1/lab2/lab3
liveuser@localhost-live:/usr/local$ mkdir -p ~/labs/lab1/lab2/lab3
liveuser@localhost-live:/usr/local$

liveuser@localhost-live:/usr/local$ touch ~/temp/lab1/lab2/lab3/text1.txt
liveuser@localhost-live:/usr/local$ touch ~/temp/lab1/lab2/lab3/text2.txt
liveuser@localhost-live:/usr/local$ touch ~/temp/lab1/lab2/lab3/text3.txt
liveuser@localhost-live:/usr/local$ ls ~/temp/lab1/lab2/lab3
text1.txt text2.txt text3.txt
liveuser@localhost-live:/usr/local$
```

Создали каталоги с помощью mkdir и создали файлы touch командой

6. 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.

```
cat: text1.txt/text2.txt/text3.txt: Not a directory
liveuser@localhost-live:~/temp$ cat text1.txt
Karina

liveuser@localhost-live:~/temp$ cat text2.txt
Shabakova

liveuser@localhost-live:~/temp$ cat text3.txt
NKAbd-05-24
liveuser@localhost-live:~/temp$
```

С помощью командой cat вывели содержимое файла

1. 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.

```
liveuser@localhost-live:~/temp$ cd
liveuser@localhost-live:~$ cp ~/temp/*.txt labs
liveuser@localhost-live:~$ mv labs/text1.txt labs/firstname.txt
liveuser@localhost-live:~$ mv labs/firstname.txt labs/lab1
liveuser@localhost-live:~$ ls labs/lab1
firstname.txt lab2
```

```
liveuser@localhost-live:~$ mv labs/lab2 labs/lab1/lab2
liveuser@localhost-live:~$ ls labs/lab1/lab2
lab3 lastname.txt
```

```
liveuser@localhost-live:~$ mv labs/text3.txt labs/id-group.txt
liveuser@localhost-live:~$ mv labs/id-group.txt labs/lab1/lab2/lab3
liveuser@localhost-live:~$ ls labs/lab1/lab2/lab3
id-group.txt
liveuser@localhost-live:~$
```

С помощью команд rm – переименовали и переместили файлы и с помощью cp - скопировали файлы указав *.txt

2. Удалите все созданные в ходе выполнения лабораторной работы файлы и каталоги.

```
liveuser@localhost-live:~$ rm -r ~/mewdir
rm: failed to remove '/home/liveuser/mewdir': Directory not empty
liveuser@localhost-live:~$ rm -R ~/mewdir
liveuser@localhost-live:~$ rm -R ~/parentdir
liveuser@localhost-live:~$ rm -R ~/temp/*.txt
liveuser@localhost-live:~$ rm -R ~/temp
liveuser@localhost-live:~$ rm -R ~/parentdir*
liveuser@localhost-live:~$ rm -R ~/labs/temp
liveuser@localhost-live:~$ rm -R ~/labs/*.txt
rm: cannot remove '/home/liveuser/labs/*.txt': No such file or directory
liveuser@localhost-live:~$ rm -R ~/labs
liveuser@localhost-live:~$ rm -R ~/tmp
liveuser@localhost-live:~$ ls ~
Desktop Documents Downloads Music Pictures Public Templates Videos
liveuser@localhost-live:~$
```

Команда rm – удаляет все каталоги и файлы

5. Выводы

Приобрела практические навыки с операционной системой и командной строки ОС

GNU Linux(организация файловой системы, навигация по файловой системе, создание и удаление

файлов и директорий)

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.
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).

