Лабораторная работа №1

студента группы ИТ-222

Мокрищева Николая Павловича

Выполнение:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Защита: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ТЕМА**

Цель работы: ознакомление с операционной системой UNIX.

**Содержание работы**

Вариант №7

1. Загрузить операционную систему (ОС) семейства UNIX. Запустить эмулятор терминала (Konsole, GNOME Terminal и т. п.) или перейти в 1-й терминал по нажатию клавиш <Ctrl>+<Alt>+<F1>.

2. Выполнить команду, выводящую на экран полное имя текущей рабочей директории (обычно, при запуске терминала, это домашняя директория пользователя).

3. Вывести содержимое текущей директории, включая скрытые объекты (файлы и директории), с указанием параметров объектов (тип, права доступа, количество ссылок, имя владельца, имя группы, размер (в байтах), временной штамп и имя объекта).

4. Выполнить команду, которая позволит изменить текущую рабочую директорию на корневую. Затем, вернуть в качестве рабочей директории домашнюю. В обоих случаях указывать абсолютный (полный) путь к директориям. В правильности смены директорий убедиться, выводя каждый раз на экран полное имя текущей рабочей директории.

5. Выполнить команду cd ../.., а затем cd без параметров, просматривая после выполнения каждой команды полное имя текущей директории. Сравнить с результатами выполнения п. 4, сделав соответствующие выводы, в том числе о значении директории, имя которой состоит из двух точек.

6. Выполнить команду cd . (параметром команды cd является точка). Как изменилась рабочая директория? Сделать вывод о значении директории с именем, состоящим из одной точки.

7. Используя команду конкатенации файлов, в текущей (домашней) директории создать файл с именем file1, в который в качестве содержимого записать строку 'Nihil sine labore!', введенную с клавиатуры. Убедиться в правильности, выполнив команду, выводящую содержимое файла file1 на экран.

8. Используя механизм перенаправления данных, в текущей директории создать file2, содержащий имена (без дополнительной информации о типе, правах доступа и т. п.) нескрытых объектов текущей директории. Убедиться в правильности, выполнив команду, выводящую содержимое файла file2 на экран.

9. Создать в текущей директории файл с именем file3 и содержимым, являющимся результатом конкатенации информации из файлов file1 и file2. Убедиться в правильности, выполнив команду, выводящую содержимое файла file3 на экран.

10. Внутри текущей директории создать директории с именами dir1, dir2 и dir3. Убедиться в правильности создания, выведя содержимое текущей директории на экран.

11. Создать копию файла file1 с именем file4. Скопировать файлы file1, file2, file3, file4 в директорию dir1 одной командой без использования шаблона имен. Убедиться в успешности копирования, выведя содержимое директории dir1 на экран (текущую рабочую директорию не изменять).

12. Переместить файлы file1, file2, file3, file4 из текущей директории в директорию dir2 одной командой с использованием шаблона имен. Переименовать перемещенный файл file2 в abc.txt. Убедиться в успешности перемещения, выведя содержимое директории dir2 на экран (текущую рабочую директорию не изменять).

13. Скопировать директории dir1 и dir2 вместе с их содержимым внутрь директории dir3. Убедиться в успешности копирования, выведя содержимое директории dir3 на экран (текущую рабочую директорию не изменять).

14. Выполнить команду, позволяющую удалить файлы file1, file3 из директории dir1, используя при этом шаблон имен. Убедиться в успешности удаления, выведя содержимое директории dir1 на экран (текущую рабочую директорию не изменять).

15. Выполнить команду, которая удалит все содержимое директории dir2. Убедиться в успешности удаления, выведя содержимое директории dir2 на экран (текущую рабочую директорию не изменять).

16. Используя шаблон имен, удалить директории dir1, dir2 и dir3 из текущей. Убедиться в правильности создания, выведя содержимое текущей директории на экран.

17. Используя команду id вывести информацию о текущем пользователе: его имя, идентификатор, названия и идентификаторы группы по умолчанию и дополнительных групп. Аналогичную информацию вывести для пользователя root, выполнив команду id root.

18. В текущей директории создать файл с именем text, содержащий строку 'Experientia est optima magistra.'. Выполнить команду, позволяющую получить для созданного файла информацию о правах доступа.

19. Вывести содержимое файла text на экран. Выполнить команду chmod u-r text. Попытаться вывести на экран информацию из файла. Сделать вывод о результате выполнения команды chmod u-r text, предварительно получив для файла text информацию об измененных правах доступа.

20. Выполнить команду chmod 642 text. Для файла text вывести на экран информацию о правах доступа. Сделать вывод о том, как при выполнении chmod 642 text изменились права доступа для каждой из трех категорий пользователей. Какие действия теперь могут и не могут совершать пользователи каждой категории с файлом?

21. Создать файл lab1\_1.c, содержащий текст программы выводящей на экран при помощи системных вызовов getuid() и getgid() идентификаторы пользователя и группы по умолчанию для пользователя. Разрабатывать текст программы возможно с использованием удобных графических текстовых редакторов (gedit, kwrite и т. п.).

22. Скомпилировать текст разработанной программы в исполняемый файл с именем lab1\_1.

23. Запустив программу из файла lab1\_1 на выполнение, получить идентификатор и группу текущего пользователя.

24. Используя команду su, выполнить программу из файла lab1\_1 от имени пользователя root, таким образом, получить идентификатор и группу пользователя root.

**Ход работы**

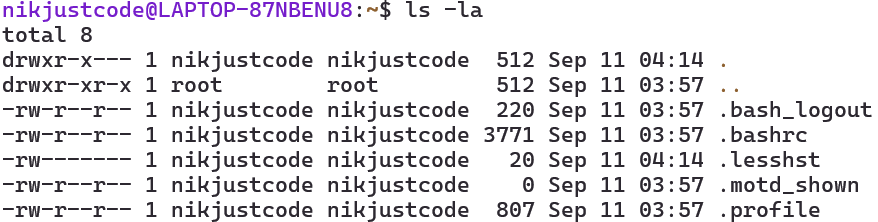
Вариант №7

2. pwd – команда для вывода на экран полное имя текущей рабочей директории. Прописываем ее в терминал (Рисунок 1).



1. Вывод полного имени текущей директории

3. Для того чтобы вывести содержимое текущей директории, включая скрытые объекты (файлы и директории), с указанием всех параметров, используйте команду ls -la (Рисунок 2).

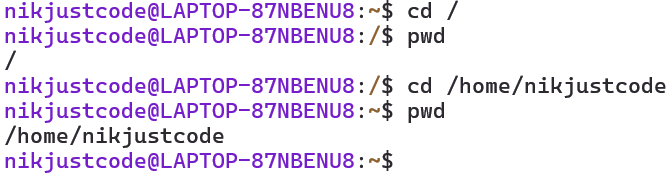


1. Вывод содержимого текущей директории со скрытыми объектами и параметрами объекта

4. В Linux, чтобы выполнить задачу по изменению текущей рабочей директории и проверить правильность выполнения, нужно использовать команды cd и pwd (Рисунок 3). Команда для перехода в корневую директорию: cd /

Команда для вывода текущей рабочей директории: pwd

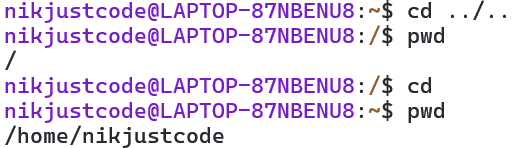
Команда для перехода в домашнюю директорию: cd /home/имя\_пользователя



1. Переходим к корневой директории и рабочей директории через абсолютный путь

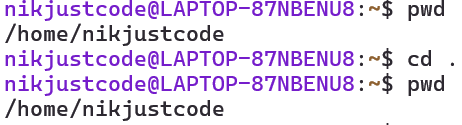
5. В прошлом задании мы перемещались в корневую директорию вручную по абсолютному пути, а в этом мы не прописывает полный путь, куда нам нужно попасть. В этом задании команда «cd ../..» (Рисунок 4) позволяет переходить в родительские директории относительно текущего местоположения делать два шага назад по родительским элементам. Она полезна для навигации в структуре файловых систем. Команда «cd» без параметров в обоих заданиях приводит нас к одному результату – она возвращает в домашнюю директорию.

Две точки (..) представляют родительскую директорию, и каждая пара точек поднимает нас на один уровень вверх в файловой системе. Команда «cd» без параметров – способ вернуться в домашнюю директорию без указания пути. Абсолютный путь (/home/имя\_пользователя) указывает на точное расположение, тогда как использование относительных путей (через ..) удобно для быстрого перемещения по структуре каталогов.



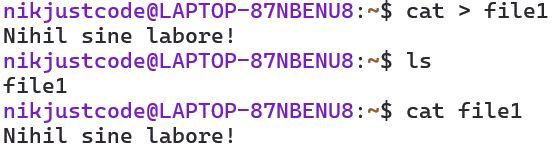
1. Переход в родительскую директорию относительно текущей

6. Точка (.) в Linux представляет текущую директорию. Выполнение команды «cd .» (Рисунок 5) означает переход в ту же самую директорию, в которой вы уже находитесь. Как видно из примера (), после выполнения команды «cd .», рабочая директория не изменилась – она осталась «/home/nikjustcode». Эта команда полезна, если необходимо явно указать на текущую директорию при выполнении скриптов или операций, где требуется указание пути, но физически она не перемещает вас по файловой системе.



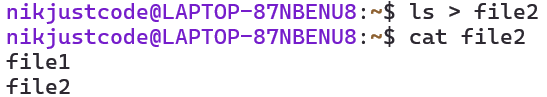
1. Возврат в домашнюю директорию

7. cat > file1 — команда cat ожидает ввода с клавиатуры и перенаправляет введённое содержимое в файл file1. Если файл file1 не существует, он будет создан.



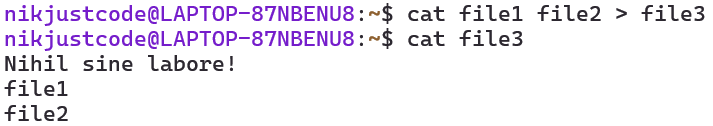
1. Запись в файл данных

8. Используем команду ls без флагов. Нужно использовать символ «>» для перенаправления вывода команды ls в файл file2 (Рисунок 7).



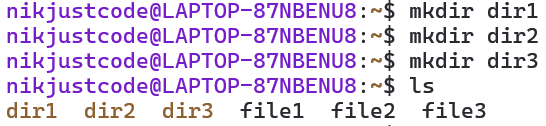
1. Запись нескрытых объектов в файл

9. Для конкатенации содержимого нескольких файлов можно использовать команду cat. Мы объединим содержимое файлов file1 и file2 и перенаправим результат в файл file3. Команда для объединения файлов file1 и file2: Используем команду cat для объединения их содержимого и создадим файл file3. cat file1 file2 — объединяет содержимое файлов file1 и file2. > — перенаправляет результат в новый файл file3. Если файл file3 уже существует, его содержимое будет перезаписано (Рисунок 8).



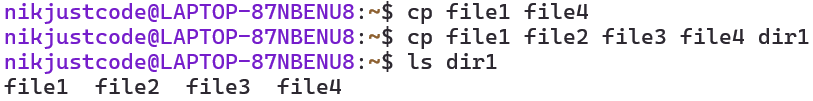
1. Запись содержимого с двух файлов в 1

10. Для создания нескольких директорий можно использовать команду mkdir (Рисунок 9).



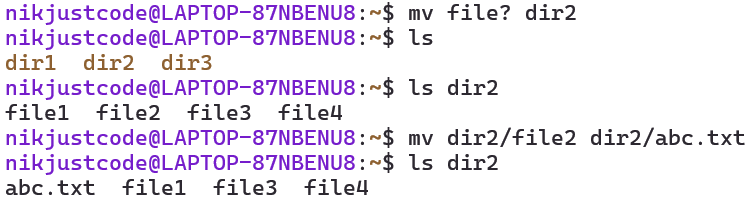
1. Создание директорий

11. Создание копии файла file1 с именем file4: Для создания копии файла можно использовать команду cp. cp — команда для копирования файлов. file1 — исходный файл. file4 — имя копии файла. Чтобы скопировать несколько файлов одновременно в директорию, используем команду cp с перечислением всех файлов, которые нужно скопировать, и указанием директории назначения.



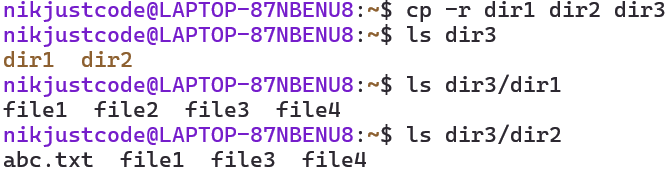
1. Копирование файлов

12. Чтобы переместить все файлы с именами, которые начинаются с file, используем команду mv с шаблоном имен file? (Рисунок 11). mv file? dir2 — перемещает все файлы, имена которых начинаются с file, в директорию dir2. Шаблон file? обозначает все файлы, начинающиеся с "file" (т.е. file1, file2, file3, file4). После перемещения переименуем файл file2 в abc.txt. mv dir2/file2 dir2/abc.txt — переименовывает файл file2 в директории dir2 в abc.txt



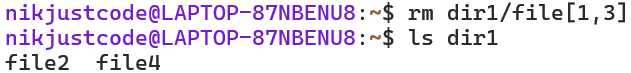
1. Перемещение файлов между директориями и переименование файла

13. Для копирования директорий вместе с их содержимым используем команду cp с флагом -r (или --recursive), который позволяет рекурсивно копировать директории и их содержимое. cp -r dir1 dir2 dir3 — копирует директории dir1 и dir2, включая их содержимое, в директорию dir3 (Рисунок 12).



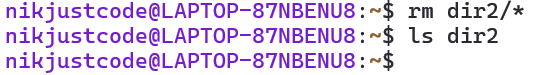
1. Копирование директорий

14. Для удаления файлов, которые начинаются с file, в директории dir1 можно использовать команду rm с шаблоном имен file[1,3]. (Рисунок 13).



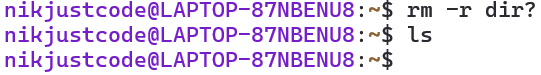
1. Удаление файлов с директории

15. Для удаления всех файлов и поддиректорий внутри директории dir2, можно использовать команду rm dir2/\* — указывает на все файлы и поддиректории внутри dir2 (Рисунок 14).



1. Удаление директории без подтверждения

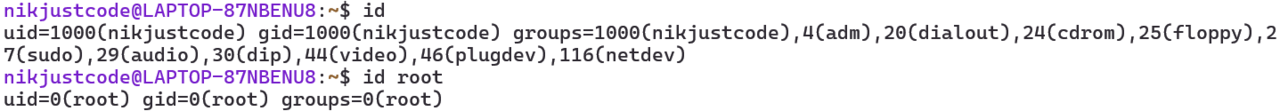
16. Для удаления нескольких директорий можно использовать команду rm с опцией -r (рекурсивное удаление) и шаблон имен для указания директорий. rm -r dir? — удаляет все директории и их содержимое, которые начинаются с dir (Рисунок 15).



1. Удаление всех директорий

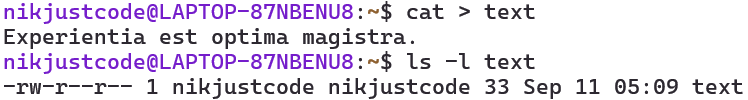
17. Информация о текущем пользователе: Для этого используйте команду id без параметров. Эта команда покажет: UID — идентификатор пользователя. GID — идентификатор основной группы. Группы — названия и идентификаторы всех групп, в которые входит пользователь.

Информация о пользователе root: Чтобы вывести информацию о пользователе root, используйте команду id с указанием имени пользователя root (Рисунок 16).



1. Информация о текущем пользователе и пользователе root

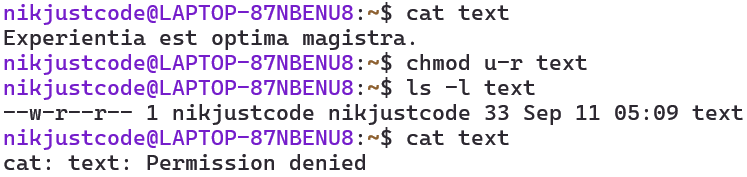
18. Создадим файл text и запишем в него строку текста. После, при помощи команды ls и его параметра -l узнаем информацию о правах доступа к файлу (Рисунок 17).



1. Информация о правах доступа

19. Выводим информацию из файла text. После, изменяем права доступа выполнив команду chmod u-r text, чтобы запретить пользователю (владельцу) читать файл. chmod u-r — запрещает владельцу файла право на чтение файла (u — владелец, -r — убрать право на чтение). Проверяем права доступа командой ls -l text. Опять стараемся вывести информацию из файла и получаем результат Рисунок 18.

В результате, при попытке снова вывести содержимое файла, было выдано сообщение об ошибке "Permission denied", так как у текущего пользователя нет прав на чтение файла.

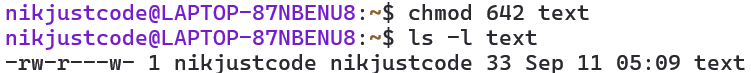


1. Изменение прав доступа к файлу

20. Команда chmod 642 устанавливает права доступа по числовому коду для каждой категории пользователей (Рисунок 19).

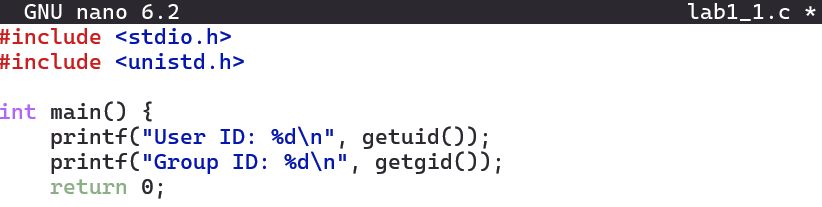
Числовая форма команды chmod определяет права доступа через трёхзначное число, где каждая цифра соответствует категории пользователей: Первая цифра (6) — права на чтение и запись для владельца файла (user). Вторая цифра (4) — права на чтение для группы (group). Третья цифра (2) — права на запись для всех остальных пользователей (others).

Таким образом, команда chmod 642 изменила права доступа так, что разные категории пользователей могут выполнять только определённые действия с файлом, ограничивая доступ к его содержимому для некоторых пользователей.



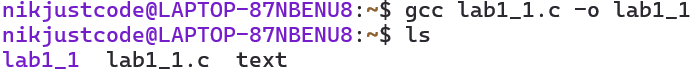
1. Изменение прав доступа

21. Код программы для вывода на экран идентификаторов пользователя и групп по умолчанию для пользователя (Рисунок 20).



1. Код на С

22. Для компиляции программы используем компилятор gcc (Рисунок 21).



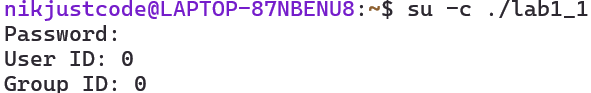
1. Компиляция программы

23. Код программы для вывода на экран идентификаторов пользователя и групп по умолчанию для пользователя (Рисунок 22).



1. Результат работы программы

24. После выполняем запуск скомпилированной программы с правами root (Рисунок 23)



1. Результат работы программы с правами root

Вывод: в ходе работы были изучены основные команды, необходимые для взаимодействия с файловой системой Linux, включая навигацию по директориям, создание и удаление файлов и папок, а также работу с абсолютными и относительными путями. Мы также разобрали принципы использования символов текущей и родительской директории (одна и две точки), что способствовало лучшему пониманию организации файловой структуры.

Ключевым аспектом стало применение команд для создания, копирования, перемещения и удаления файлов, что дало нам практический опыт в управлении данными в Linux. Мы освоили механизм перенаправления вывода команд, что позволило автоматизировать запись информации в файлы. Значительное внимание было уделено работе с шаблонами имен (wildcards), что упростило взаимодействие сразу с несколькими файлами.

Подробно была рассмотрена система управления правами доступа к файлам с использованием команды «chmod», позволяющей изменять права для различных категорий пользователей: владельцев, групп и «остальных». Это дало возможность глубже понять, как устроена система безопасности в Linux и как правильно настраивать доступ в разных ситуациях.

Работа с пользователями и группами также оказалась важной. Мы применяли команды для получения информации о текущем пользователе и научились запускать программы от имени суперпользователя root, что дало нам представление о работе привилегированных пользователей в системе. Также были написаны программы на языке C, которые взаимодействовали с системными вызовами для получения идентификаторов пользователя и группы, что углубило наше понимание взаимодействия между операционной системой и программным обеспечением на низком уровне.

Работа с WSL (Windows Subsystem for Linux) продемонстрировала возможности использования Linux-окружения на платформе Windows, что позволяет сочетать преимущества обеих систем. Этот опыт обеспечил практическое понимание работы с правами доступа, файлами и программами в Linux, что является важным навыком для системных администраторов и разработчиков.

Таким образом, в результате данной работы были приобретены обширные навыки, охватывающие взаимодействие с файловой системой, управление правами доступа, пользователями и программированием на уровне операционной системы.