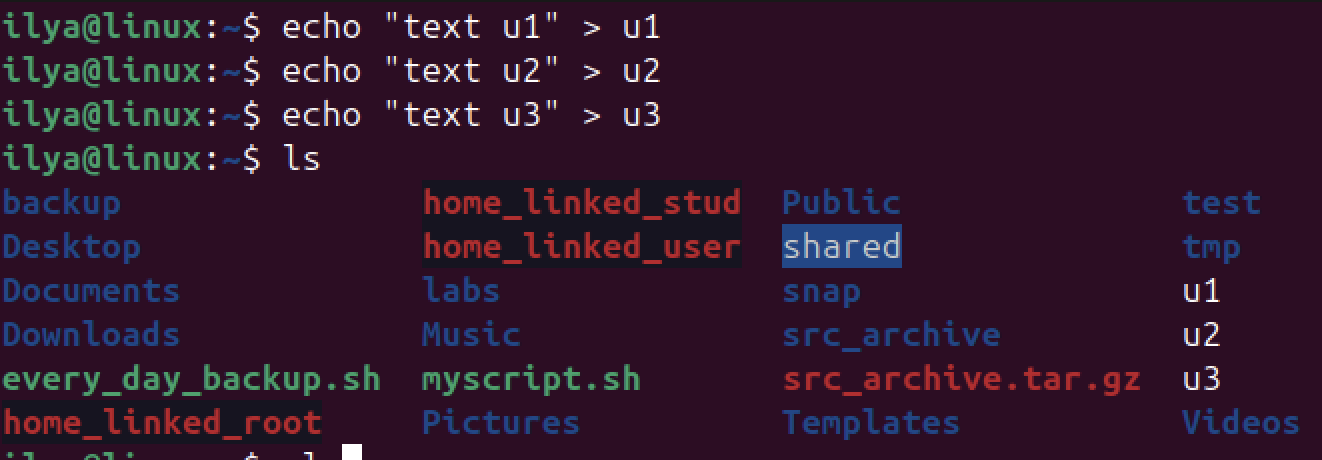
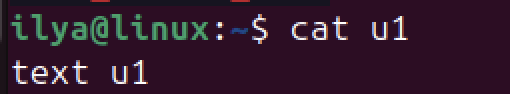
**LX3 Права и пользователи. Управление правами доступа**

**Цель работы** - Получить навыки по управлению правами доступа к файловым ресурсам системы, освоить основные команды - chmod, chown, chgrp.

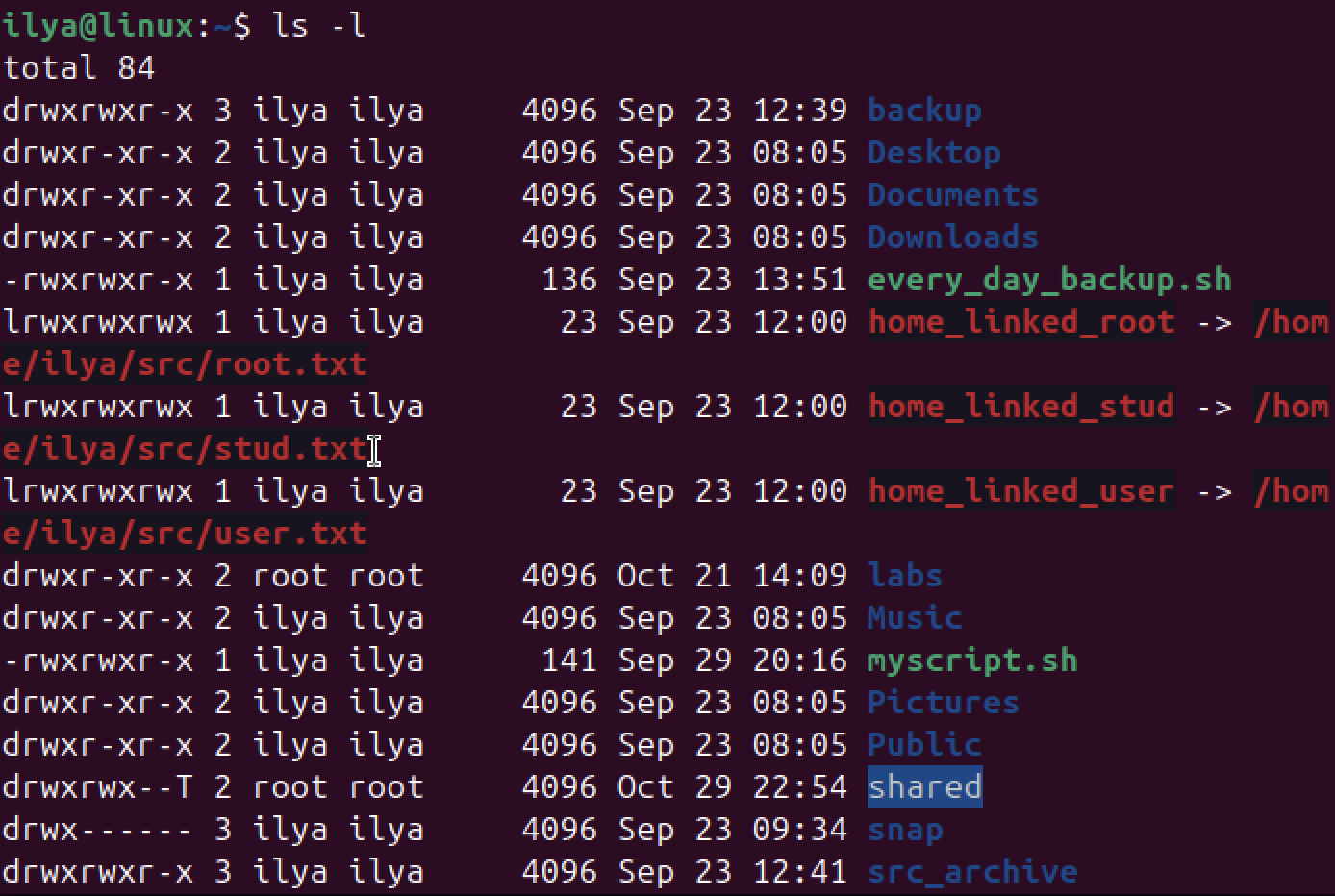
Создать в домашнем каталоге 2-3 файла произвольного содержания (имена файлов - u1, u2, u3). Воспользуемся командой echo ‘содержимое файла’ > u1.



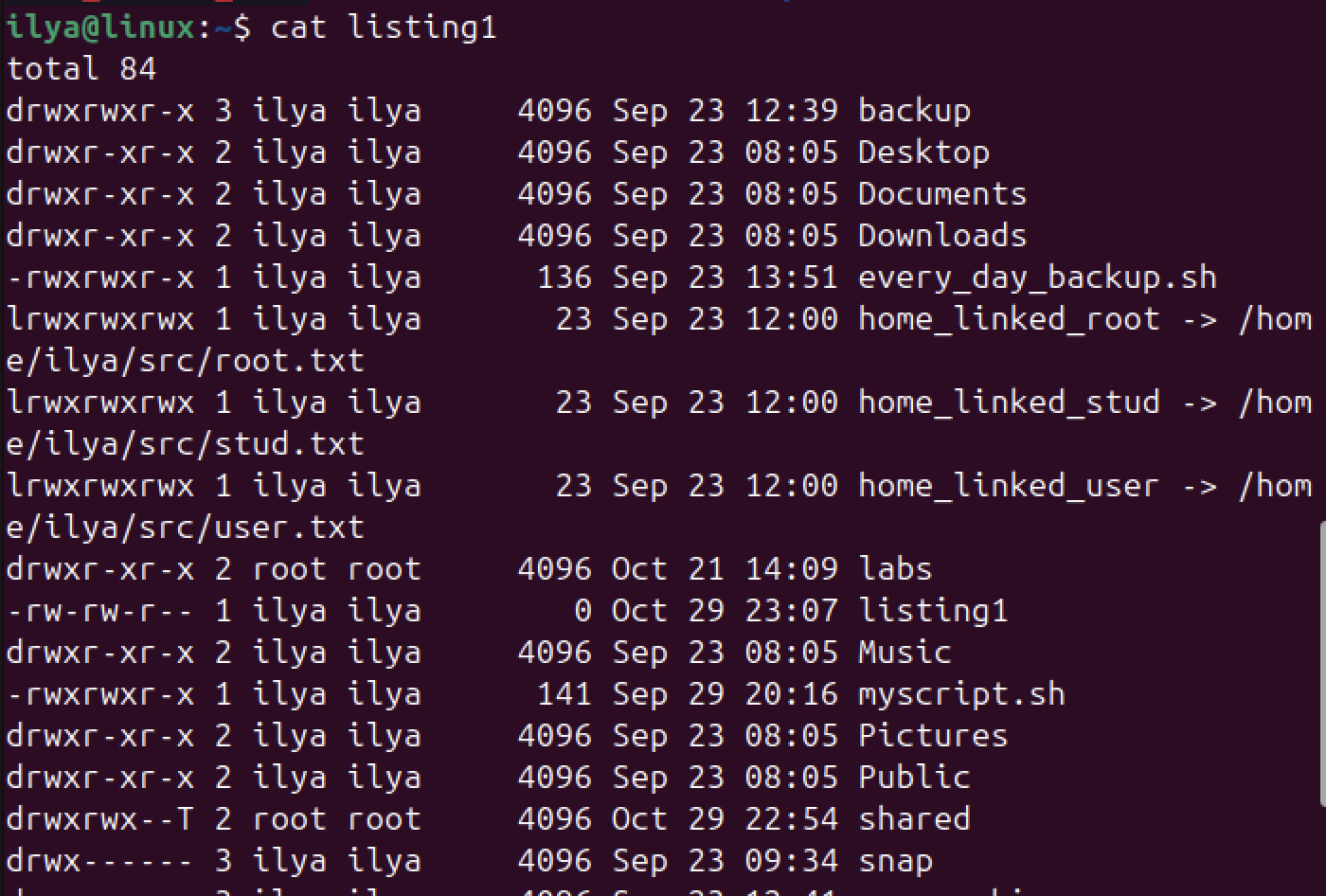
Видим, что файлы были успешно добавлены. Теперь проверим содержимое одного из них. Команда - cat u1



Получить развернутый список файлов домашнего каталога и сохранить его в файле listing1. Для получения развернутого списка файлов домашнего каталога пропишем команду ls -l.



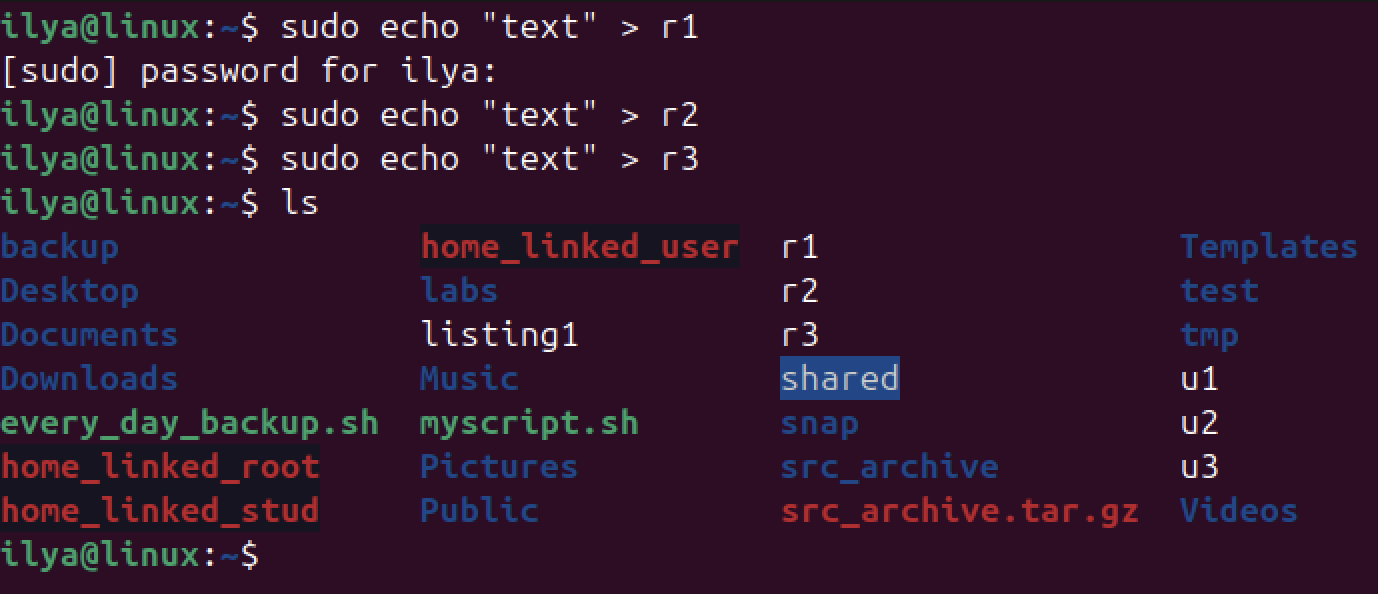
Далее сохраним этот список в файле listing1, который создадим в домашнем каталоге. Воспользуемся командой ls -l > listing1.



Просмотреть файл listing1, обратив внимание на поля прав доступа, владельца и группы. При обращении внимания на поля прав доступа владельца и группы. У директорий в правой колонке на первом место стоит буква d, которая указывает на то, что это директория. У файлов стоит на первом месте символ “-”. Также замечаем набор символов rwx, повторяющие трижды, которые показывают права доступа (чтение, запись, исполнение) для владельца, групп и остальных пользователей.

Повторить п. 2 от имени пользователя root в новом сеансе или по команде su (имена файлов - r1, r2, r3). Завершить сеанс root. Для этого будем использовать приписку sudo перед выполнением команд.

Создадим файлы произвольного содержания с названиями r1, r2, r3.

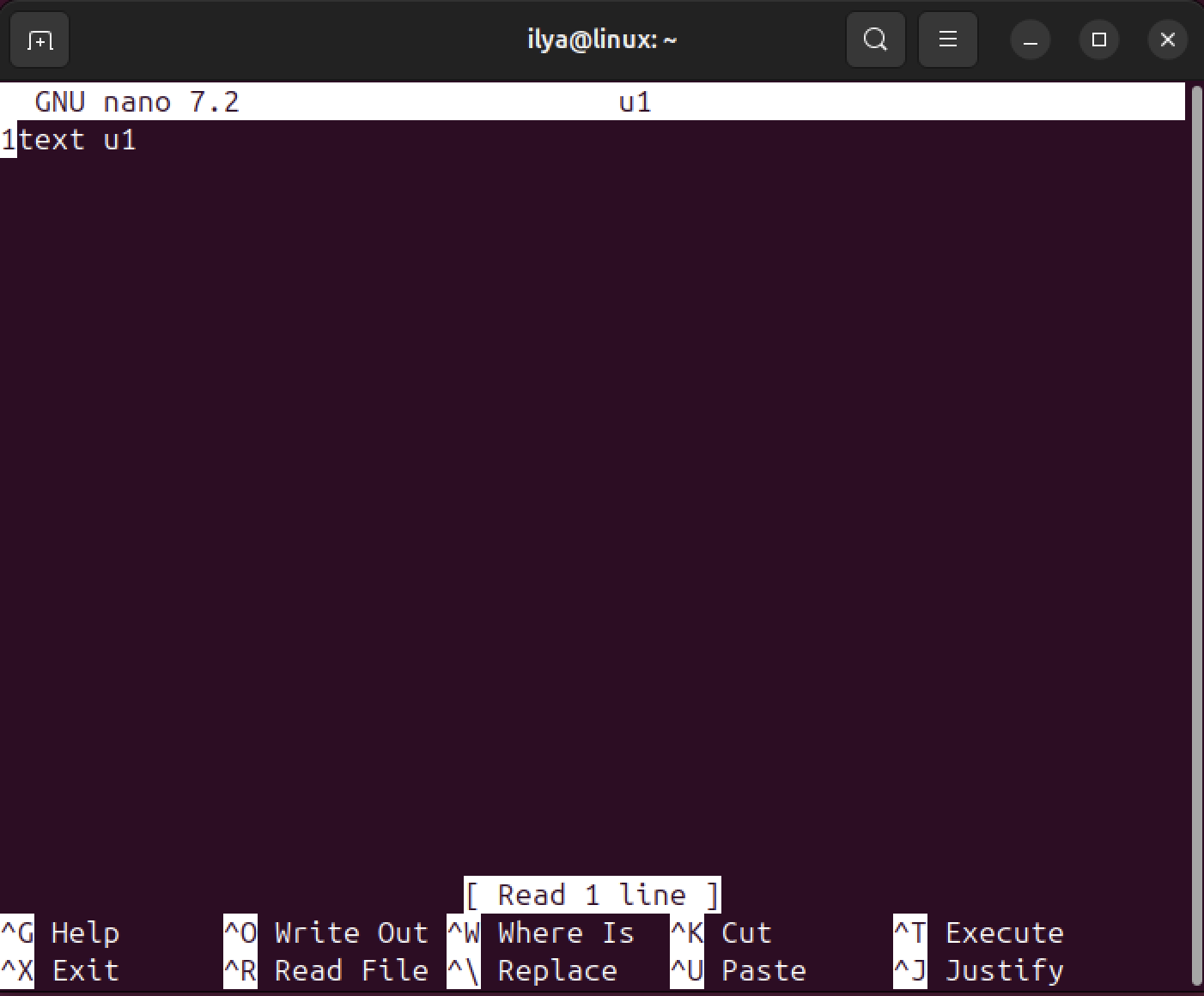


Запишем развернутый список файлов в файл listing1.



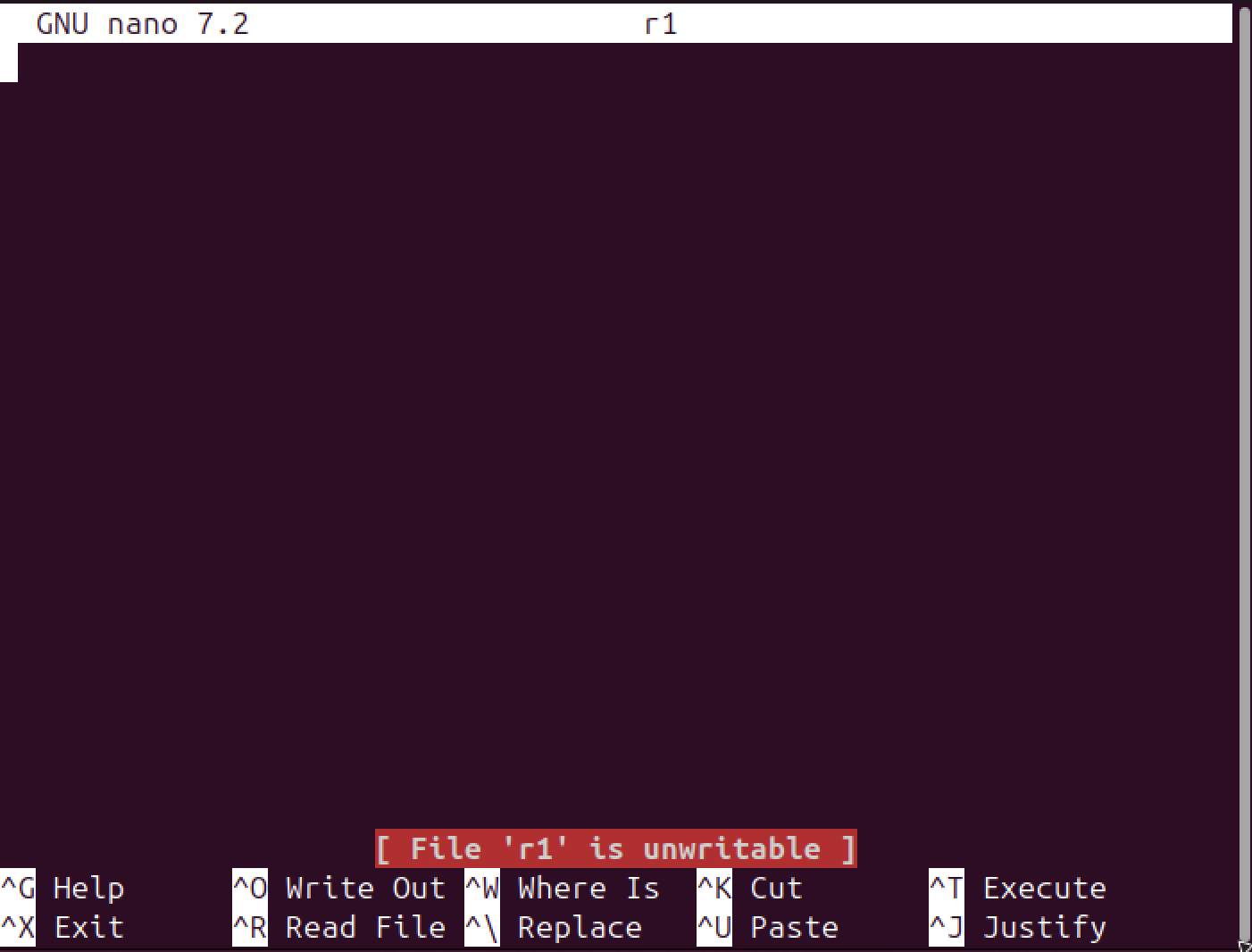
В графе пользователь теперь стоит root. Следовательно задание выполнено верно.

Изменить содержимое файлов, созданных вами и суперпользователем. Сохранить изменения. Используем команду nano для изменения содержимого фала. Сначала изменим файл u1 от имени пользователя, созданный мной.

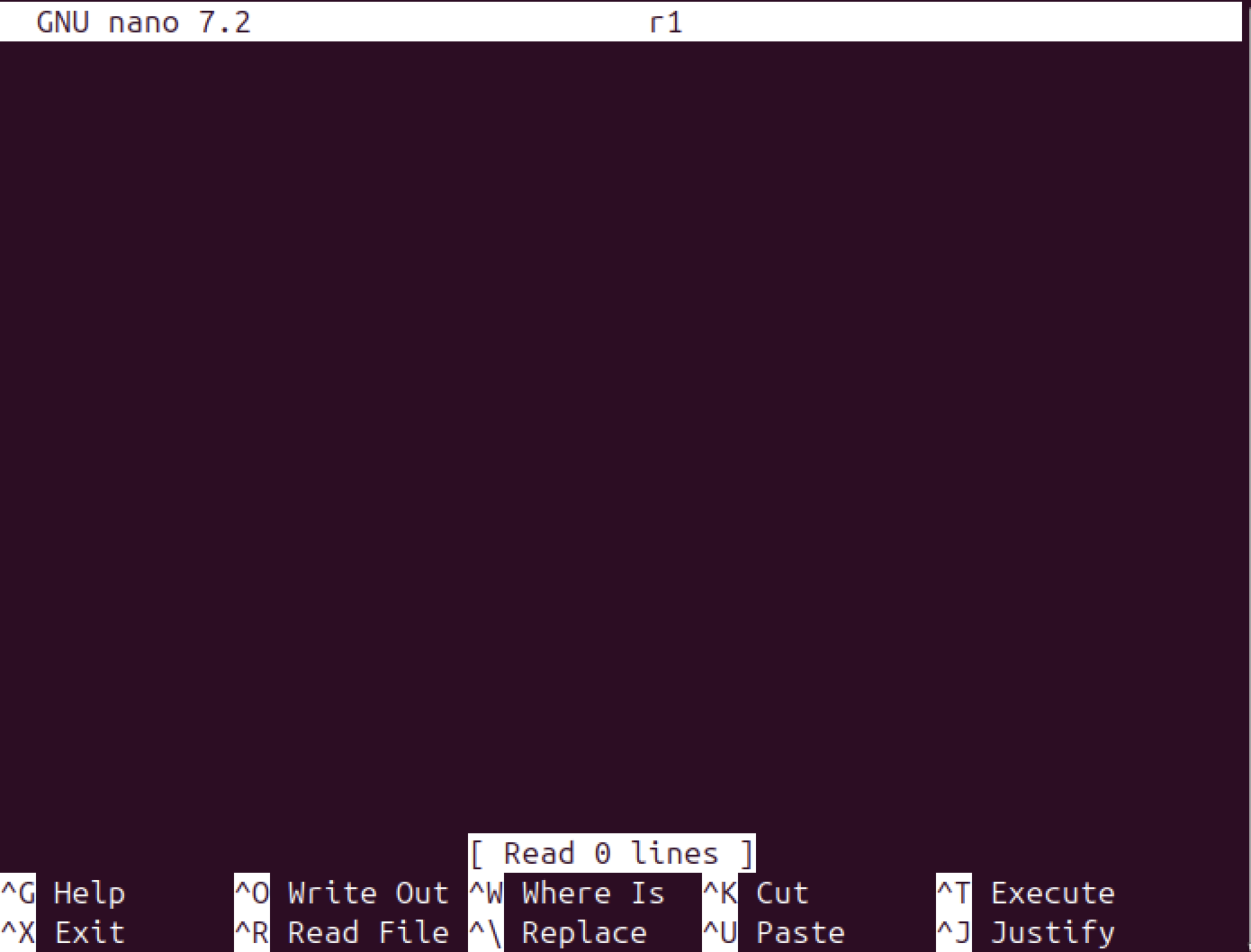


Появляется данное окно, в котором мы спокойно можем редактировать содержимое файла.

Теперь попробуем изменить содержимое для фала r1, прописав ту же команду.



Видим небольшую красную табличку, которая указывает, что файл r1 не записываемый. Это потому, что мы редактируем данный файл не от имени суперпользователя, а от имени пользователя. Чтобы это исправить, перед тем как вводить команду nano r1, нужно добавить sudo.

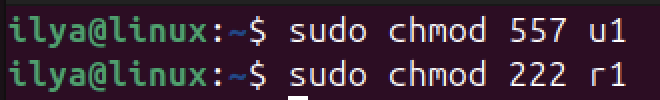


Как мы видим, добавив команду sudo, мы спокойно можем редактировать содержимое.

Открыть сеанс root. Перейти в каталог /home/ваша\_учетная\_запись. Изменить права доступа к файлам u1 и r1 следующим образом:

u1: запретить запись для владельца и группы

r1: разрешить запись для всех



Для файла u1 запретили запись для владельца и группы, путем вычитания 2 из 7, получили 5.

Нужно помнить, что

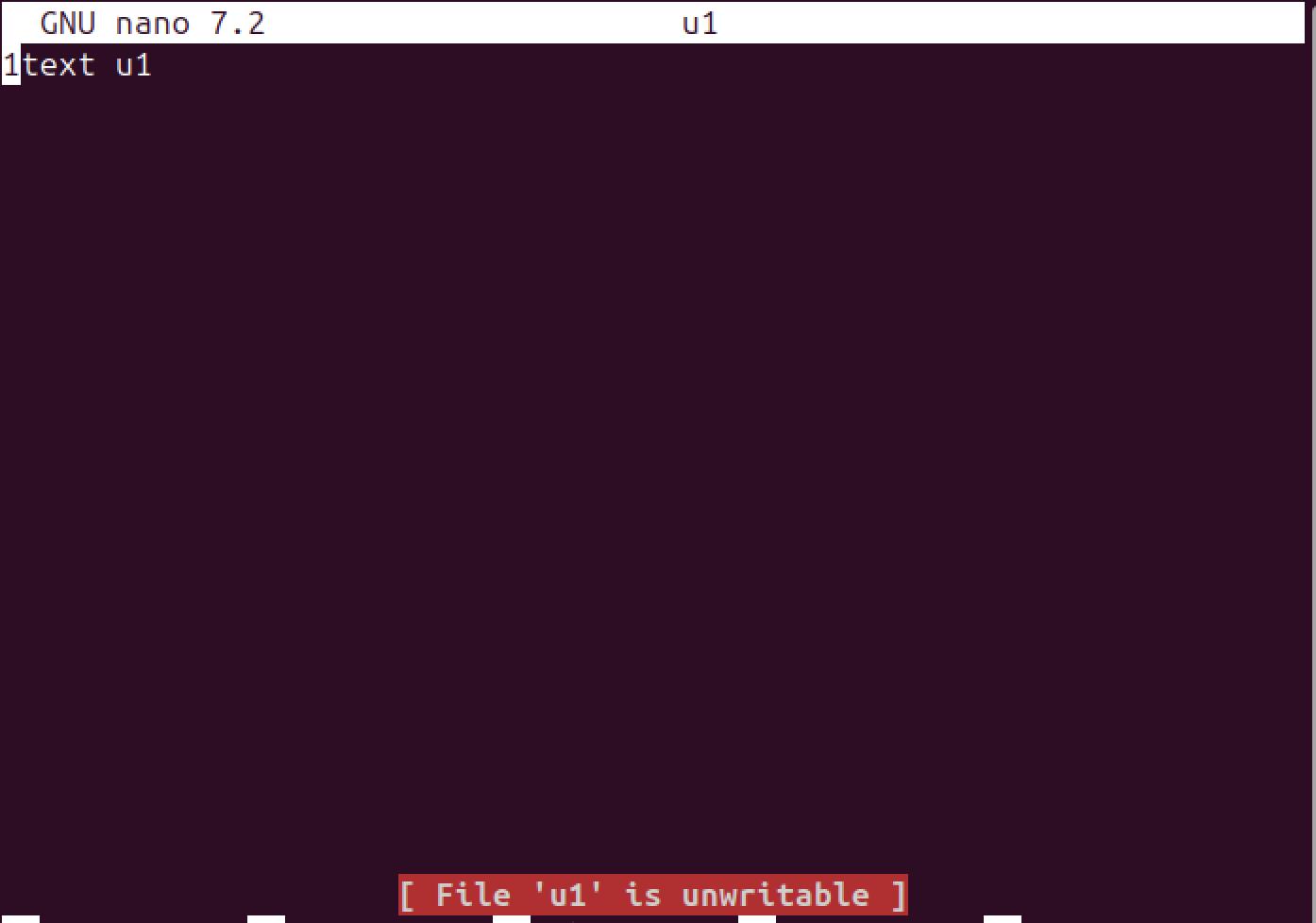
4 — разрешение на чтение (read).

2 — разрешение на запись (write).

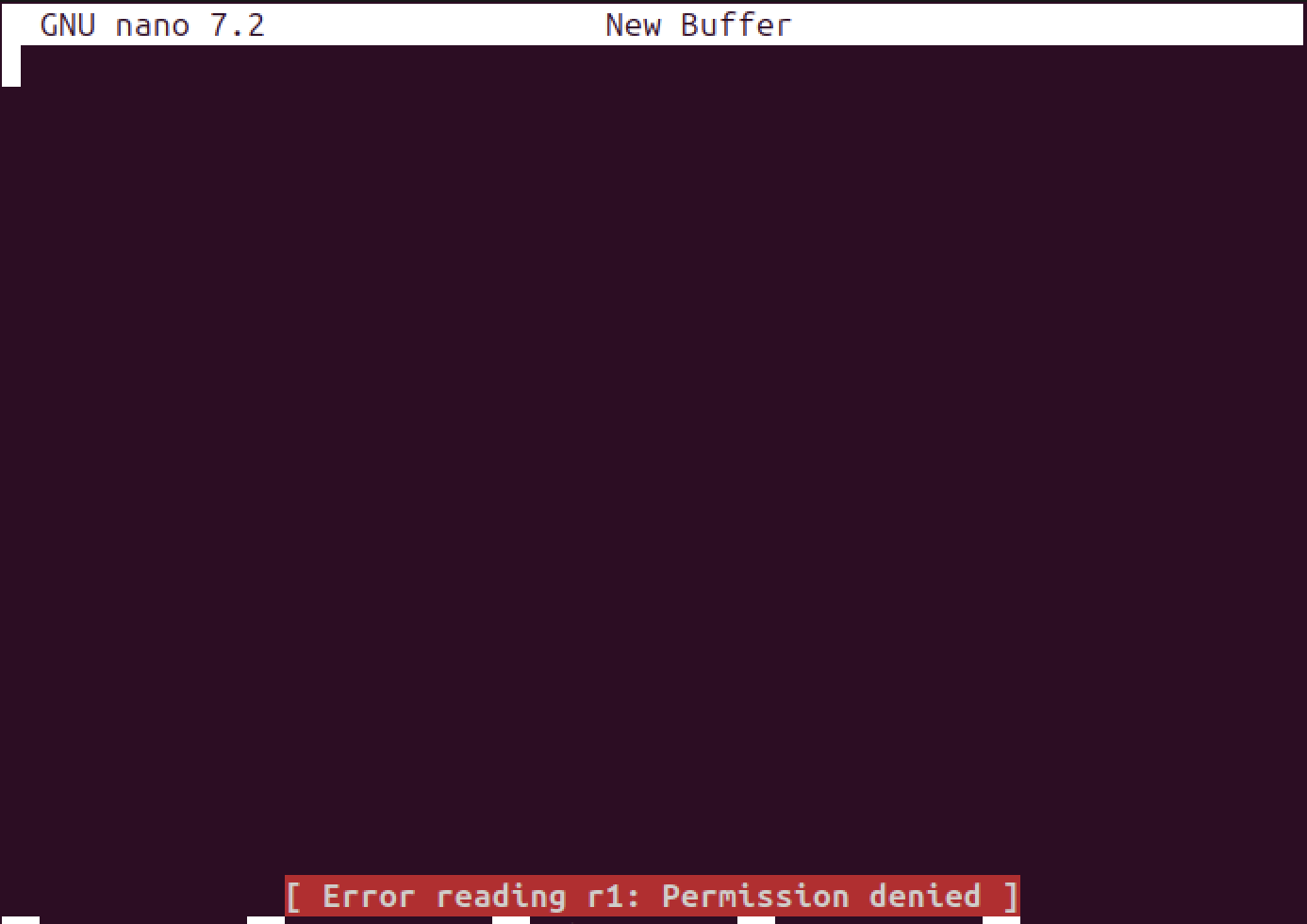
1 — разрешение на выполнение (execute).

Переключиться в сеанс пользователя и изменить содержимое файлов u1 и r1. Сохранить изменения

Проверим результаты работы.

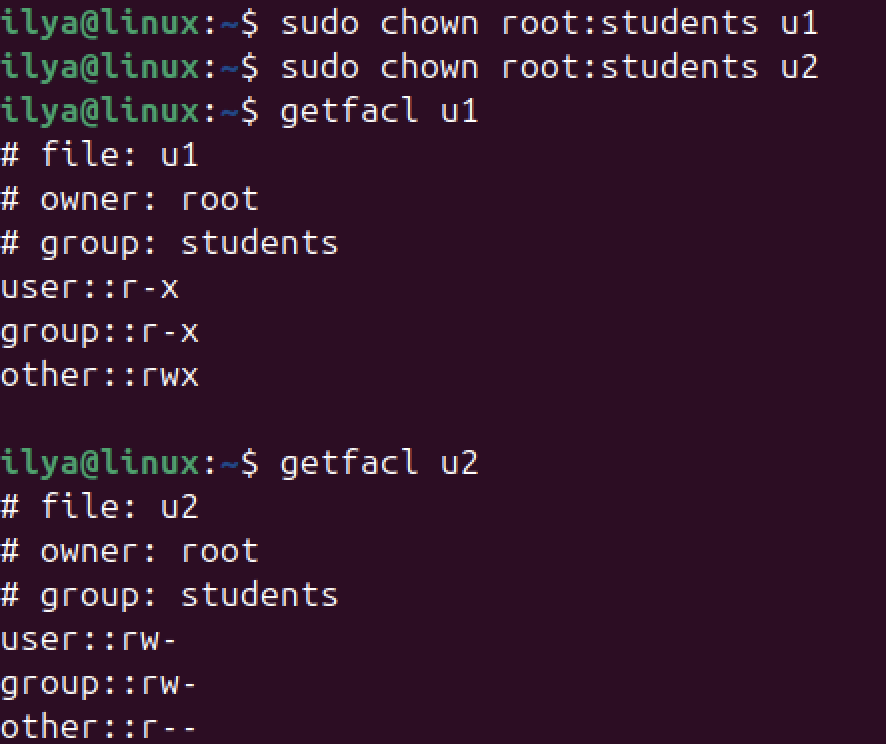


Для файла u1 видим, что в него мы не можем ничего записать. Все верно.



Для файла r1 видим ошибку чтения, это потому, что мы разрешили только запись для этого фала, но чтение запретили.

Перейти в сеанс root и изменить владельца файлов u1 и u2 на root, а группу - на students. Для этого пропишем команду sudo chown root:students u1 u2.



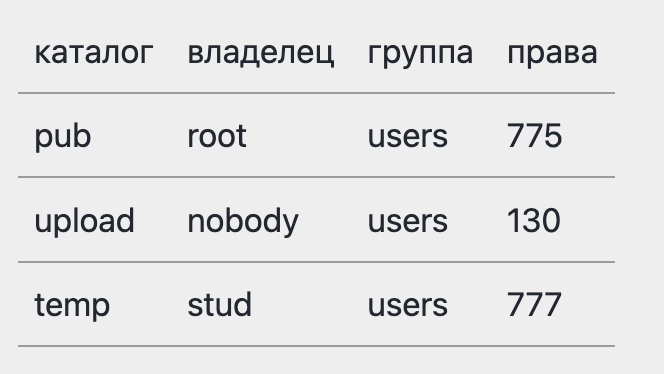
С помощью команды getfacl проверим владельца файлов и группу. Как мы видим, указаны root и students.

Из tty1 попробовать изменить файл u2.

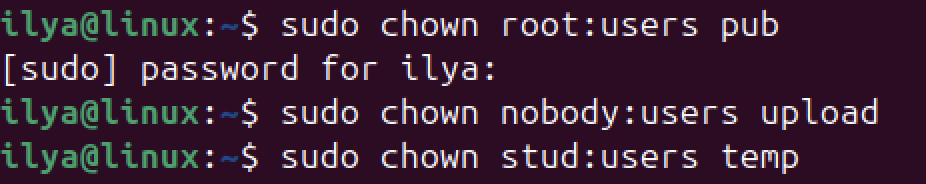
Перейдем в tty1 с помощью Ctrl+Alt+F1. Некоторые элементы не отображаются, но из сообщения понятно, что u2 не доступен для записи. Так что изменять содержимое мы не можем.

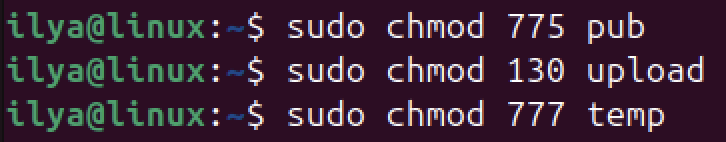


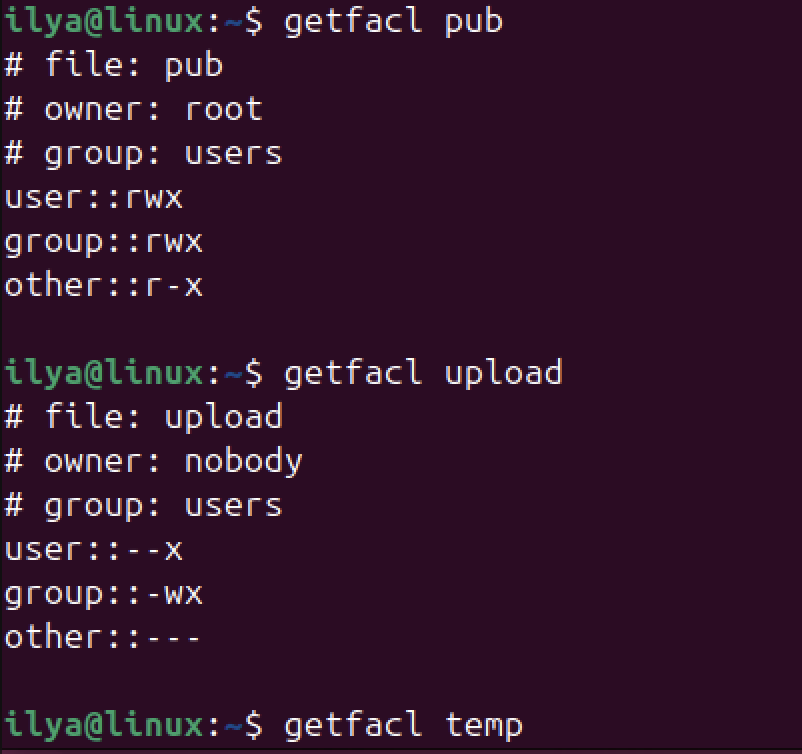
Из tty2 создать каталоги /home/shared, home/shared/pub, /home/shared/upload, /home/shared/temp. Установить на них следующие права:



После создания перейдем в графический терминал и проверим правильность созданных каталогов и проверим их права:

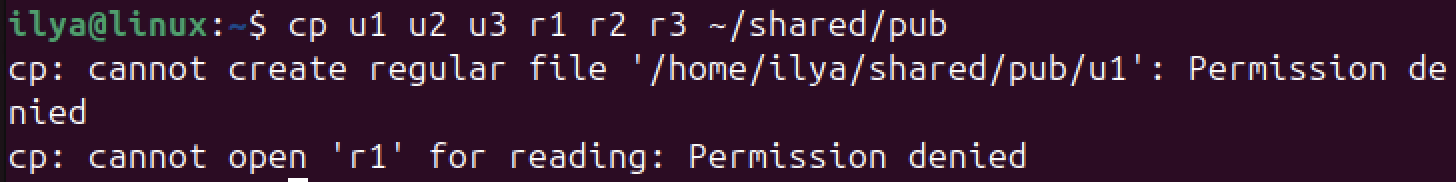




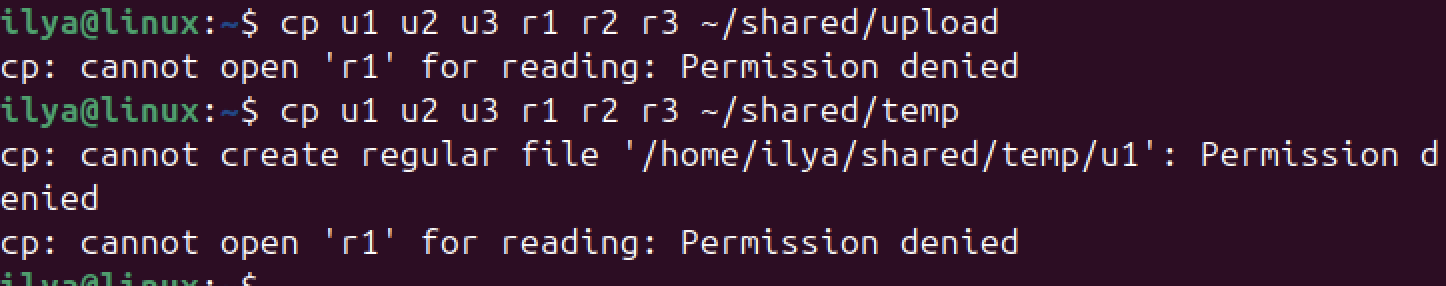


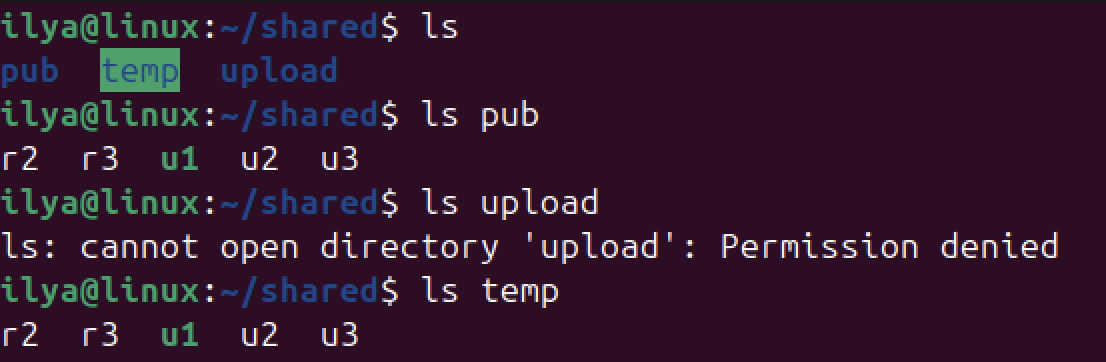
Выполнить копирование, чтение файлов u1, u2, u3, r1, r2, r3 в каталоги, созданные в п. 15 из сеансов root, stud и вашего. Сравнить и проанализировать результаты.

Воспользуемся командой cp для копирования фалов в созданные каталоги. Получаем следующее:



Получаем такое сообщение, так как в ходе работы для фала u1 мы задавали ограничения на запись, а для r1 мы разрешили только запись, но чтение произойти не может. Аналогично при копировании в другие каталоги.





Если рассматривать каталоги на чтение, то все файлы, кроме r1 в каталоге pub отображаются. В каталоге upload недоступное чтение файлов, так как мы задали значение для пользователя 130, что не предусматривает чтение этого каталога. Для каталога temp аналогично pub не отображается только r1, так как у этого файла права 222 - только запись.

**Контрольные вопросы.**

1. Зачем у файла нужны атрибуты доступа?

Атрибуты доступа к файлам играют важную роль в обеспечении безопасности и управления доступом. Они определяют, какие пользователи и группы могут выполнять определенные действия с файлами.

2. Для каких трех категории пользователей задаются права доступа каждого конкретного файла?

Для владельца, групп и других пользователей.

3. Для чего в UNIX-подобных системах используются пользовательские группы?

Они используются для упрощения управления правами доступа. Они позволяют объединять пользователей с общими задачами, что облегчает назначение прав на файлы и каталоги.

4. Какие действия с файлами регламентируются правами доступа?

Чтение (r), запись (w), исполнение (x).

5. На какой системе счисления основывается числовое представление прав доступа?

На восьмеричной системе счисления. Каждое право представляет собой число:

• Чтение = 4

• Запись = 2

• Исполнение = 1

Эти значения суммируется и образуют набор прав для файла или каталога. (max = 7).

6. Чем отличается назначение прав доступа к простому файлу и к директории?

• Для файлов права определяют возможность их чтения, записи или исполнения.

• Для директорий права определяют возможность просмотра содержимого (чтение), добавления или удаления файлов (запись) и входа в директорию (исполнение).

7. Кто может менять права доступа определенного файла?

Права доступа к файлам могут изменять только их владельцы или суперпользователь (root).

8. Каково назначение специальных битов прав доступа?

Специальные биты прав доступа используются для обеспечения более тонкого управления доступом и безопасности в системе:

• setuid: позволяет исполняемому файлу запускаться с правами владельца файла.

• setgid: позволяет исполняемому файлу запускаться с правами группы владельца файла.

• sticky bit: применяется к директориям и предотвращает удаление файлов другими пользователями, кроме владельца файла.

**Задания для самостоятельного выполнения.**

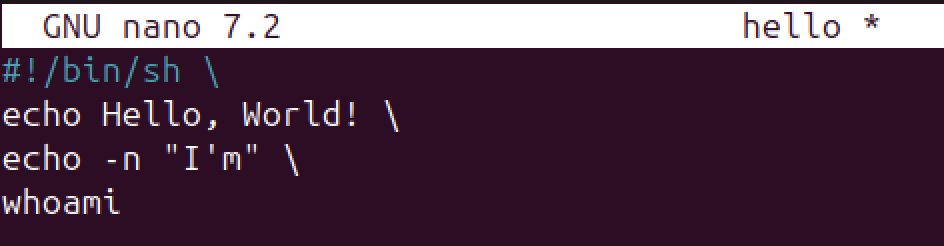
В текущей папке создать файл hello следующего содержания:

#!/bin/sh \

echo Hello, World! \

echo -n "I'm " \

whoami

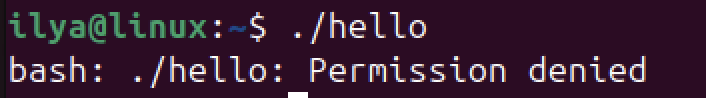


Выполнить следующие действия и проанализировать результаты:

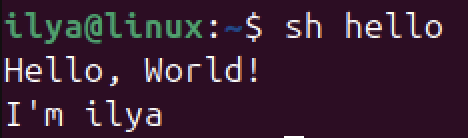
набрать в командной строке имя файла hello и нажать Enter

набрать в командной строке sh hello и нажать Enter

установить для файла hello права на исполнение (x), ввести имя файла в командной строке (./hello) и нажать Enter



Не получилось, т.к. у файла нет прав на исполнение. Попробуем через sh.



Добавим этому файлу право на исполнение с помощью команды chmod +x hello :

