МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 6**

по дисциплине: «Системное программирование»

на тему: «Архивация, сжатие и резервное копирование в Linux»

Выполнила: ст. гр. 10702221

Багапов Д.М.

Матвейчук К.А.

Яцук З.Р.

Проверил: Давыденко Н.В.

Минск 2023

**ЛАБОРАТОРНАЯ РАБОТА #1б**

**Архивация, сжатие и резервное копирование в Linux**

**Цель работы :**

Научиться создавать архивы, сжатие и распаковку файлов, создавать резервные копии данных, синхронизировать файлы и каталоги.

Рассмотрим несколько программ, часто используемых для управления коллекциями файлов.

**Задание 1**

Сделайте архивную копию вашего ранее созданного программного проекта.

Перенесите копию на другое устройство (можно использовать флешнакопитель) или другую учетную запись, и распакуйте архивную копию.

Сравните размеры полученных файлов.

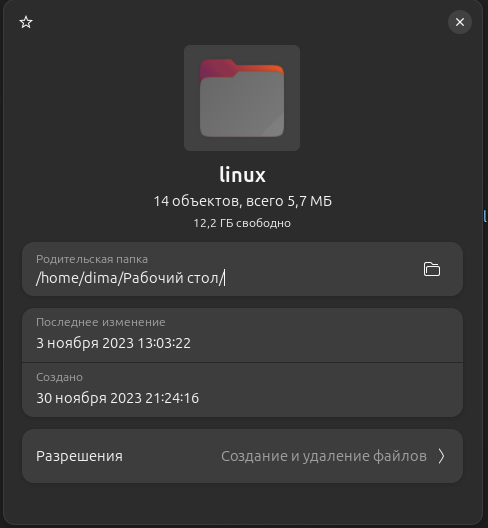
Затем проведите сжатие вашего ранее созданного программного проекта.

Перенесите сжатую версию проекта в другое пространство имен и распакуйте

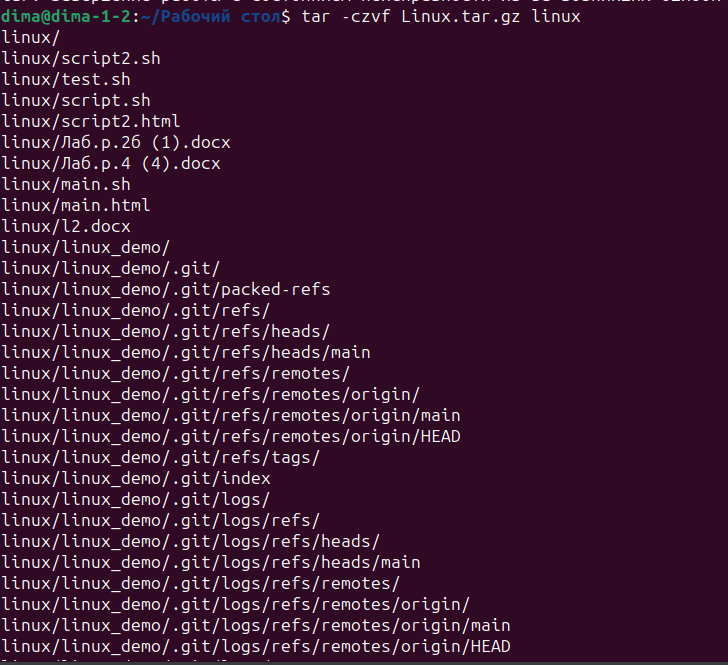
проект. Сравните размеры полученных файлов, а также сравните резултаты с

предыдущими результатами задания.

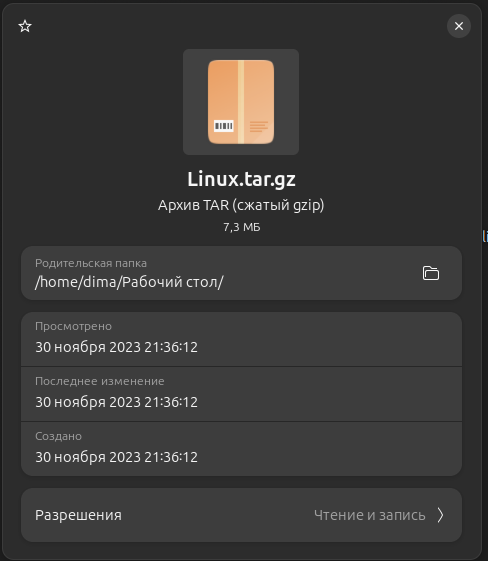
При выполнении задания предпочтительным является передача файлов между

системами по сети.

# Папка до архивации



Успешное архивирование папки



Свойства архива

Размер папки после переноса на другое устройство не изменился.

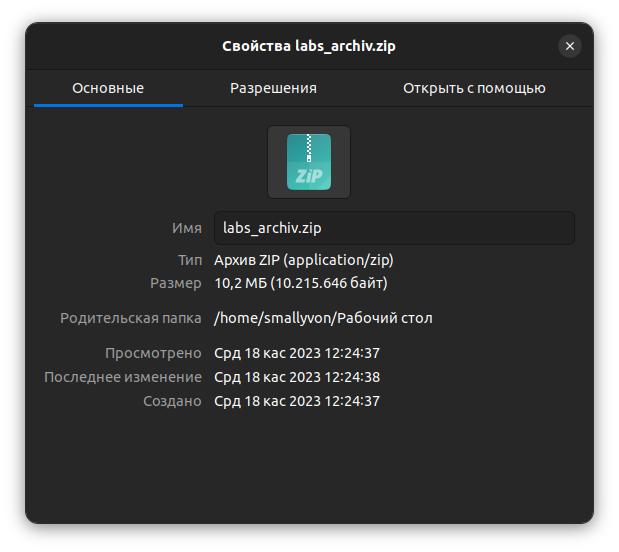
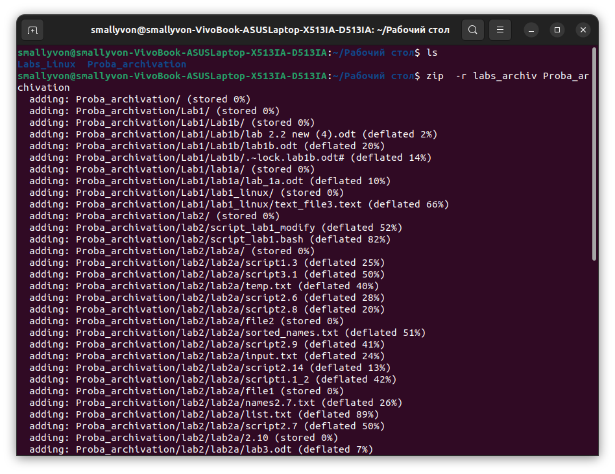
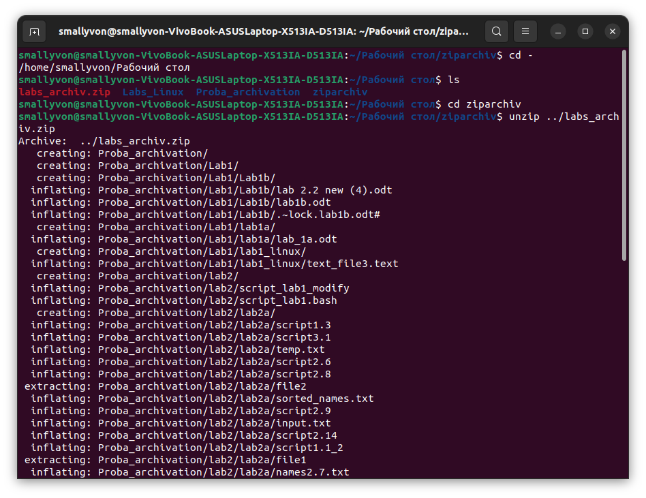
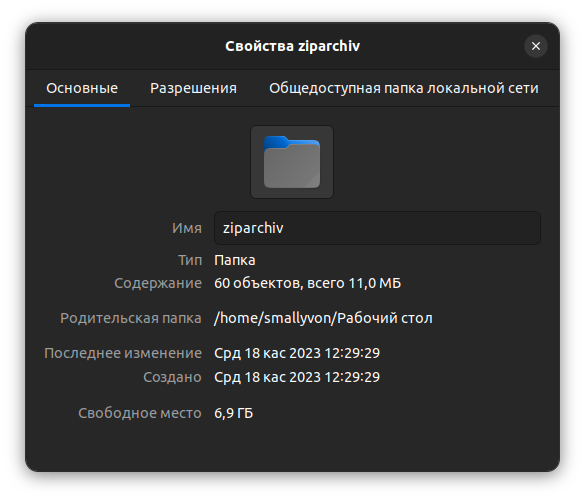
Он мог измениться если бы были потери в сети или если бы отличался блочный размер на другом устройстве

Задание 2

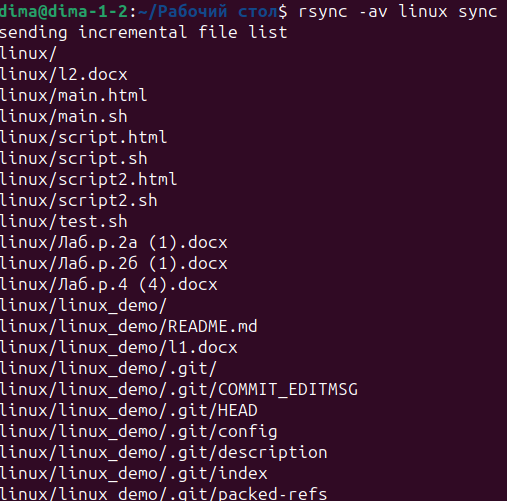
Проведите упаквку и сжатие вашего ранее созданного программного проекта.

Перенесите сжатую версию проекта в другое пространство имен и распакуйте

проект. Сравните размеры полученных файлов, а также сравните резултаты с

результатами предыдущих заданий.

Задание 3 Сделайте синхронизацию каталогов или файлов программы, разработанной вами ранее. Минимальным требованием является синхронизация дистрибутива в локальной системе.



Успешное выполнение синхронизации

**Контрольный вопросы :**

**2) Что такое относительны и абсолютынй путь к файлу?**

В Unix (и в других операционных системах) существуют два типа путей к файлам: абсолютные и относительные. Они определяют, как операционная система находит конкретный файл в файловой системе.

1. **\*\*Абсолютный путь (Absolute Path)\*\*:**

- Это полный и точный путь к файлу или каталогу от корневого каталога файловой системы.

- В Unix корневой каталог обозначается символом "/", и абсолютный путь всегда начинается с этого символа.

- Пример абсолютного пути: `/home/user/documents/myfile.txt`

- Абсолютный путь уникален и указывает на однозначное расположение файла или каталога в файловой системе.

**2. \*\*Относительный путь (Relative Path)\*\*:**

- Это путь к файлу или каталогу относительно текущего рабочего каталога (текущей директории).

- Он не начинается с символа "/", и он интерпретируется относительно текущего положения в файловой системе.

- Пример относительного пути: `../parent\_directory/file.txt`

- Относительный путь может изменяться в зависимости от текущей директории, в которой вы находитесь.

Пример использования:

Предположим, ваш текущий каталог (текущая директория) - это `/home/user/documents`, и вы хотите обратиться к файлу `myfile.txt`. Вы можете использовать абсолютный путь `/home/user/documents/myfile.txt` или относительный путь `myfile.txt`. Если вы смените текущий каталог на `/home/user`, то относительный путь также изменится на `documents/myfile.txt`.

Важно помнить, что абсолютные пути всегда указывают на одну и ту же локацию в файловой системе, в то время как относительные пути зависят от текущей директории и могут изменяться при её изменении.