Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тихоокеанский государственный университет»

Высшая школа кибернетики и цифровых технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

ПО КУРСУ «ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ»

«СОЗДАНИЕ УСТАНОВОЧНЫХ ПАКЕТОВ ФОРМАТА DEB ДЛЯ КОМПОНЕНТОВ ПРОГРАММНОГО КОМПЛЕКСА»

Выполнил: студент 2 курса группы ПО(б)-31

Магонов Александр Александрович

Проверила: ассистент ВШ КЦТ

Балабасова Алиса Дмитриевна

Хабаровск 2024 г.

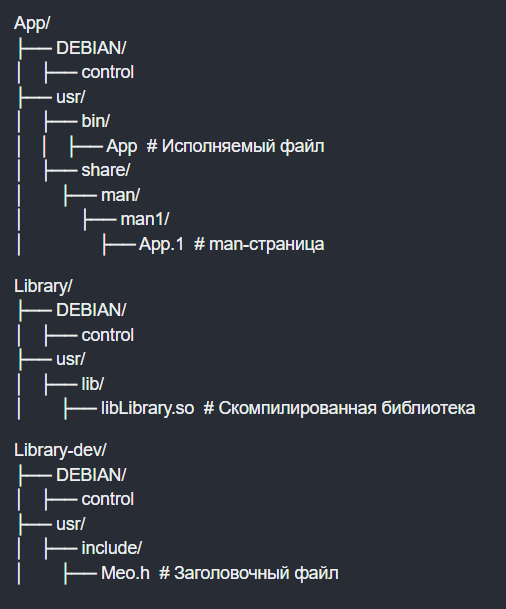
**Цель работы:**

Создать установочные пакеты формата DEB для компонентов программного комплекса, разработанного в лабораторной работе №1, с учётом стандартов размещения файлов в дистрибутивах Linux.

**Задание:**

1. Создать три DEB-пакета:
   * **Программа (App):** содержит исполняемый файл приложения и man-страницу, созданную в лабораторной работе №4.
   * **Библиотека (Library - основной пакет):** содержит откомпилированную версию библиотеки.
   * **Пакет разработчика (Library-dev):** содержит заголовочные файлы библиотеки.
2. Реализовать размещение файлов в соответствии с правилами:
   * Исполняемые файлы программы: /usr/bin.
   * Man-страницы: /usr/share/man/man1.
   * Двоичные файлы библиотек: /usr/lib.
   * Заголовочные файлы библиотек: /usr/include.
3. Настроить зависимости пакетов:
   * Пакет **App** должен зависеть от пакета **Library**.
   * **Library-dev** должен зависеть от **Library**.
4. Проверить корректность работы пакетов:
   * Установить пакеты в системе.
   * Убедиться в работе программы и библиотеки после установки.
   * Проверить работу man-страницы через команду man App.

Структура комплексного приложения: исполняемый файл (App), динамическая библиотека (Library), библиотека разработчика с заголовочным файлом (Library-dev):

  
Рисунок 1 – структура комплексного приложения

Создаем .so библиотеку вместо .dll для поддержки в OC Linux на рисунке 2.

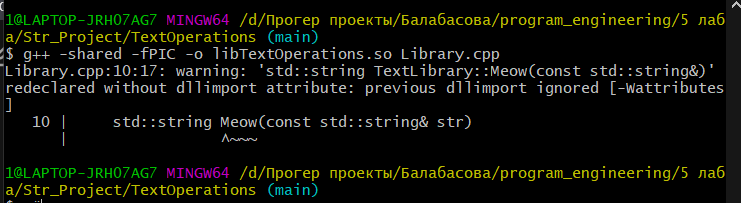


Рисунок 2 – сборка библиотеки TextOperations с помощью g++ в консоли MINGW64

Также напишем файлы control для каждого отдельного нашего каталога файлов в папке DEBIAN (они представляют из себя конфигурацию для deb формата) (рисунок 3).

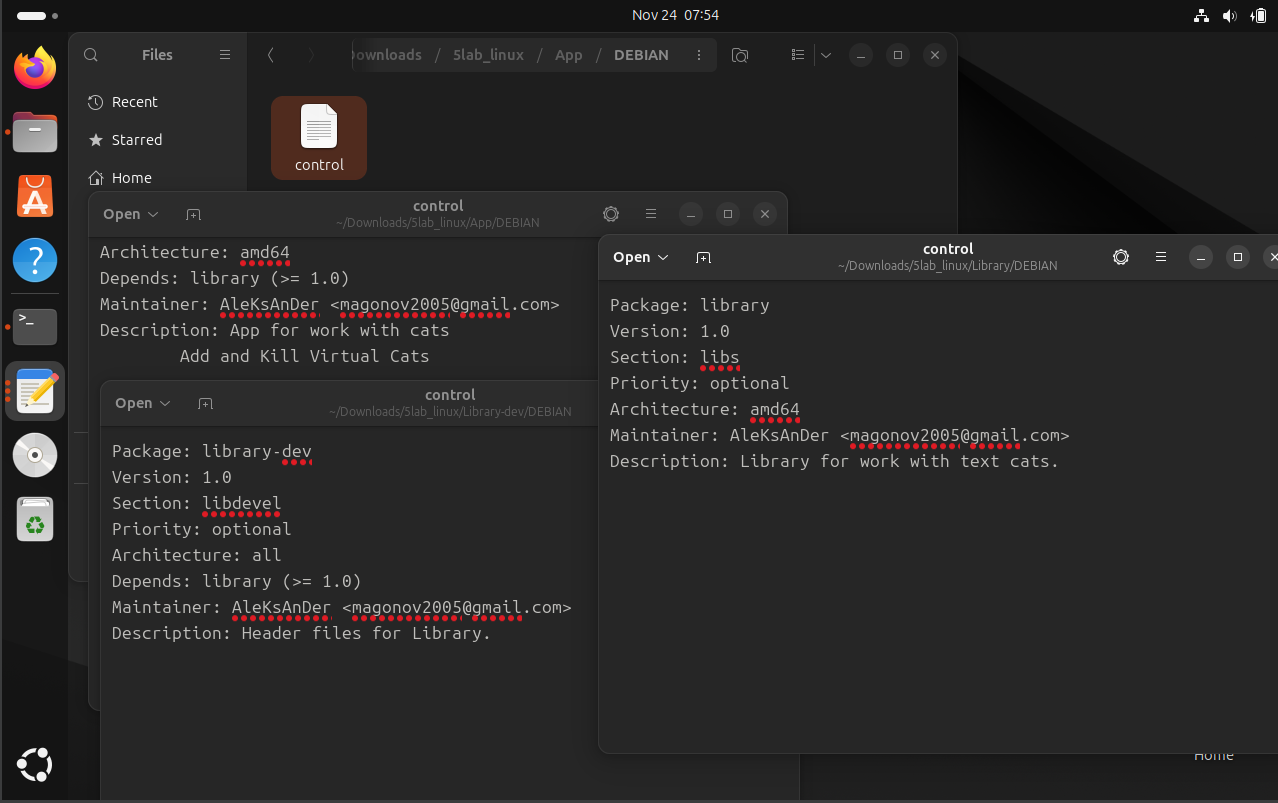


Рисунок 3 – написание конфигурации в файлы control (связывание библиотек и заголовочных файлов)

Для создания DEB-пакетов используем утилиту dpkg-deb :

App.deb — пакет с программой и документацией, Library.deb — основной пакет библиотеки, Library-dev.deb — пакет разработчика с заголовочными файлами.

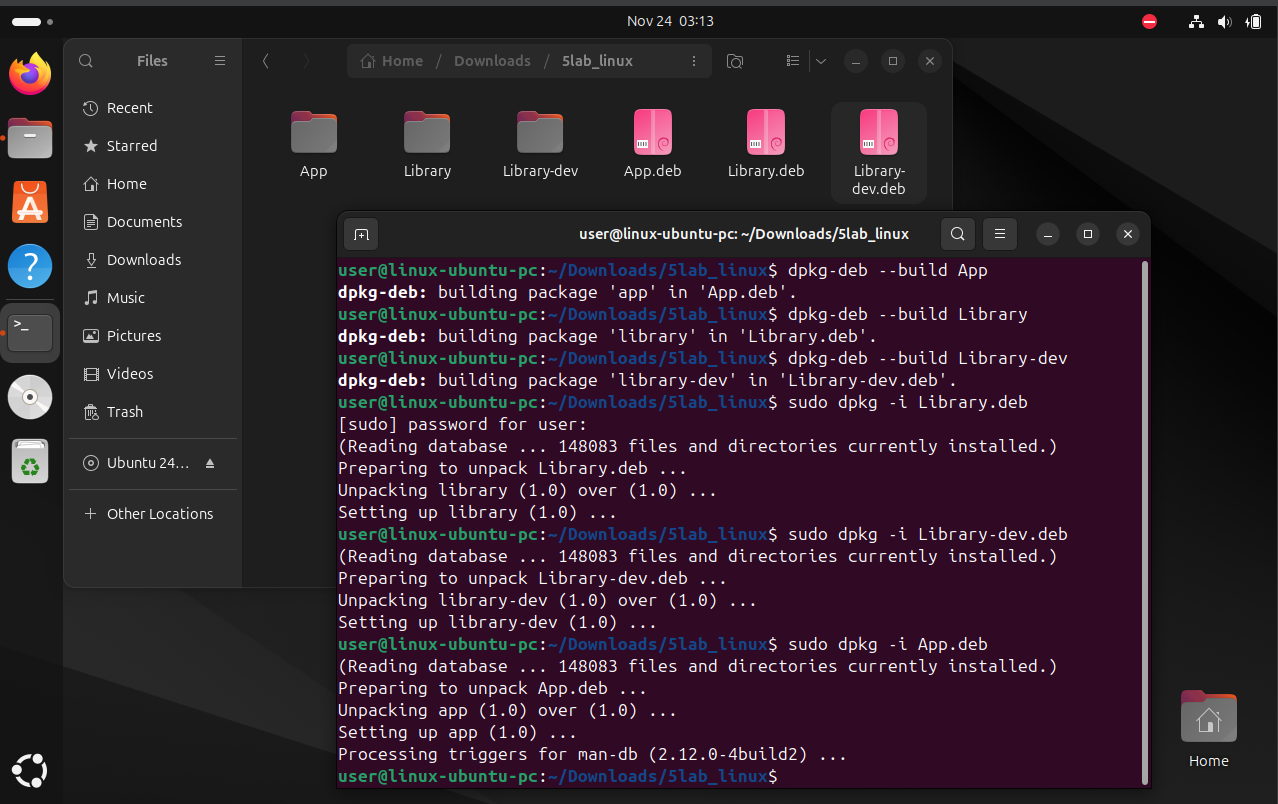


Рисунок 4 – создание DEB-пакетов и установка библиотек

Демонстрация работы man-file в консоли на OC Linux на рисунке 5.

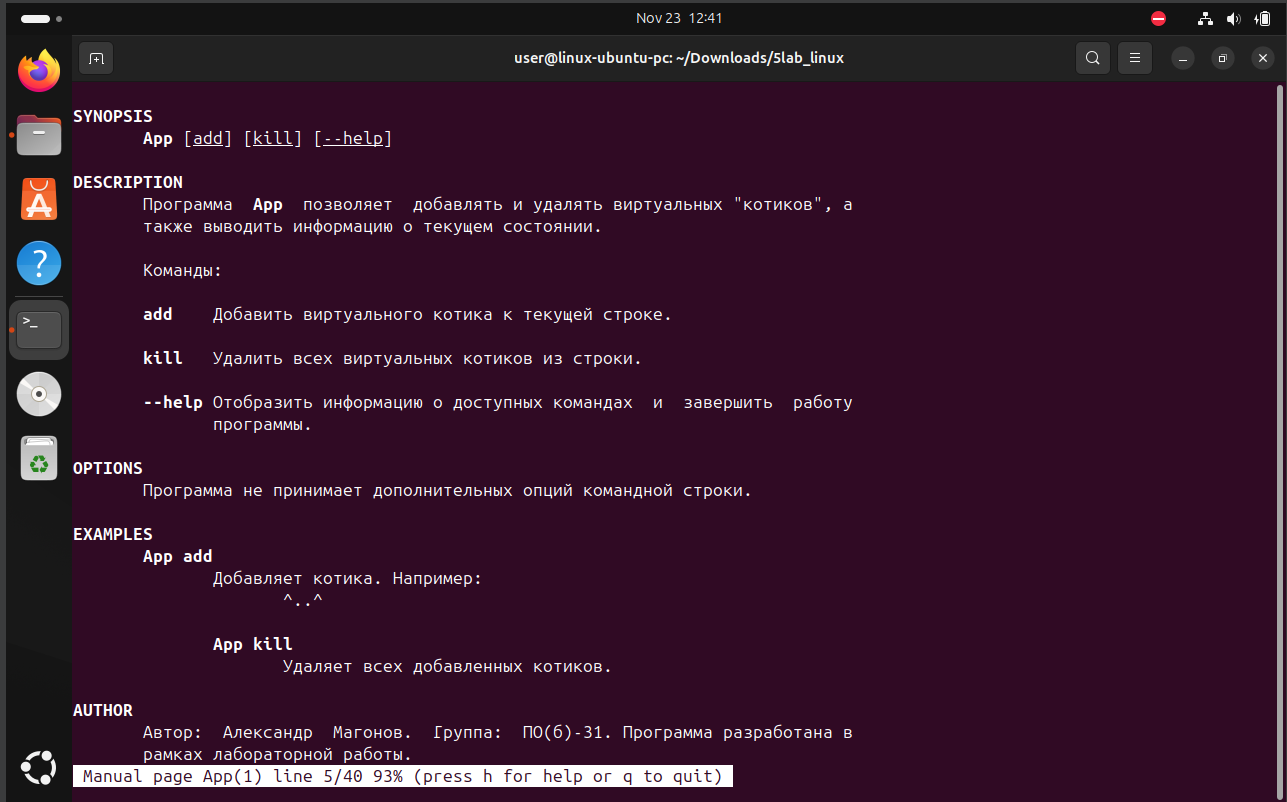


Рисунок 5 – вызов команды “man App”

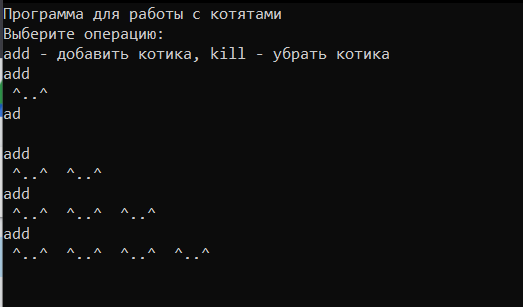


Рисунок 6 – демонстрация работы консольного приложения

**Вывод:**

Цель лабораторной работы заключалась в создании установочных пакетов формата DEB для программного комплекса, состоящего из консольного приложения и динамической библиотеки. В ходе работы были выполнены следующие задачи:

* Разработаны три DEB-пакета:
  + Пакет с программой **App**.
  + Основной пакет библиотеки **Library**.
  + Пакет разработчика с заголовочными файлами **Library-dev**.
* Реализована структура пакетов в соответствии с правилами дистрибутивов Linux:
  + Исполняемый файл программы размещён в /usr/bin.
  + Man-страница размещена в /usr/share/man/man1.
  + Библиотека размещена в /usr/lib.
  + Заголовочные файлы размещены в /usr/include.
* Установлены зависимости между пакетами:
  + **App** зависит от **Library**.
  + **Library-dev** зависит от **Library**.
* Проведено тестирование:
  + Установка пакетов на Linux-систему прошла успешно.
  + Программа и библиотека работоспособны после установки.
  + Man-страница отображается корректно через команду man App.

В результате работы был создан полностью функциональный программный комплекс, готовый к развёртыванию в дистрибутивах Linux. Лабораторная работа продемонстрировала практическое применение навыков создания пакетов и работы с зависимостями в системе.