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 №3

ПО КУРСУ «ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ»

«СБОРКА ПРОГРАММНОГО КОМПЛЕКСА С ПОМОЩЬЮ CMAKE»

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

Магонов Александр Александрович

Проверила: ассистент ВШ КЦТ

Балабасова Алиса Дмитриевна

Хабаровск 2024 г.

**Цель работы:**  
Написать CMakeLists.txt файлы для программного комплекса, разработанного в лабораторной работе №1, обеспечивающие сборку и установку библиотеки и консольного приложения.

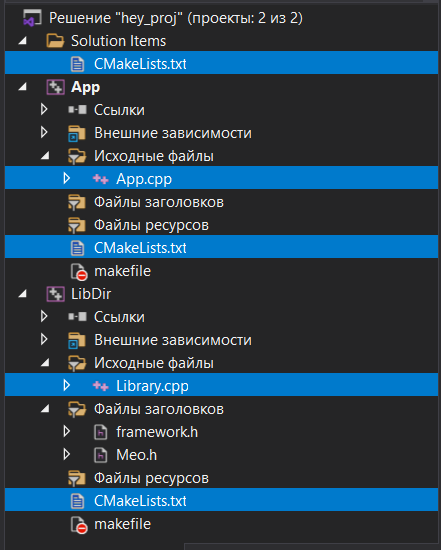
**Задание:**  
Для библиотеки и основного приложения создаются отдельные файлы CMakeLists.txt. В каждом файле должны быть предусмотрены следующие правила:

1. **compile** — Цель по умолчанию. Выполняет компиляцию части программного комплекса (динамической библиотеки или консольного приложения).
2. **install** — Установка части программного комплекса в заданную папку:
   * Для библиотеки:
     + Установка двоичного файла (.dll) в папку lib.
     + Установка заголовочного файла в папку include.
   * Для консольного приложения:
     + Установка исполняемого файла в папку bin.
3. **clean** — Очистка папки от ранее скомпилированных файлов (.dll, .exe, временных файлов сборки).
4. **uninstall** — Удаление ранее установленной части программного комплекса.

**Требования к CMake проекту:**

1. В файле CMakeLists.txt основной программы (Str\_App) должно быть указано местоположение библиотеки (TextOperations) через переменные пути.
2. Проект должен собираться через CMake с поддержкой сборки динамической библиотеки (.dll) и линковки приложения.
3. Установленные файлы должны сохраняться в соответствующие директории:
   * bin — для исполняемых файлов.
   * lib — для файлов библиотеки.
   * include — для заголовочных файлов библиотеки.

Структура комплексного приложения с консольным приложением (App.exe) и динамической библиотекой (Library.dll), а также CMakeLists для приложения, библиотеки и корня всего проекта.

  
Рисунок 1 – структура комплексного приложения с CMakeLists

Код файла CMakeLists с инструкцией для сборки dll-библиотеки Library на рисунке 2.

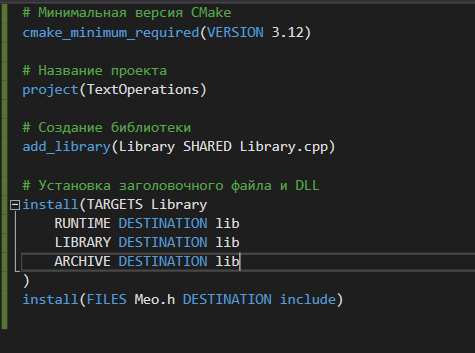


Рисунок 2 – код CMakeLists для динамической библиотеки

Код файла CMakeLists с инструкцией для сборки консольного приложения App на рисунке 3.

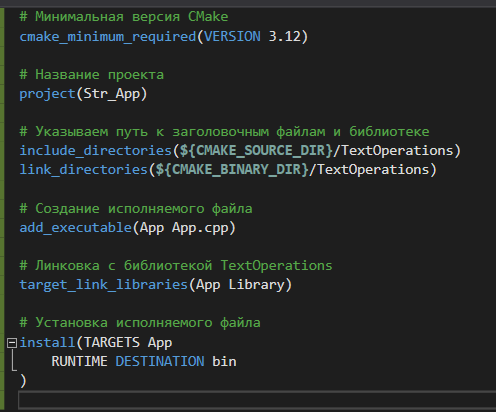


Рисунок 3 – код CMakeLists для консольного приложения

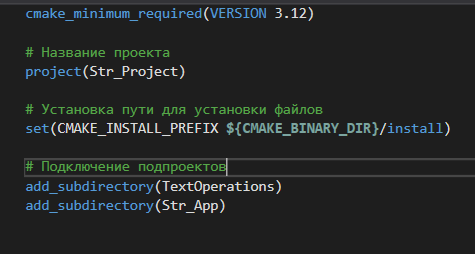


Рисунок 4 – код CMakeLists для всего проекта (главный)

Демонстрация сборки (компиляции) проекта с помощью CMakeLists через Git Bash (MINGW64):

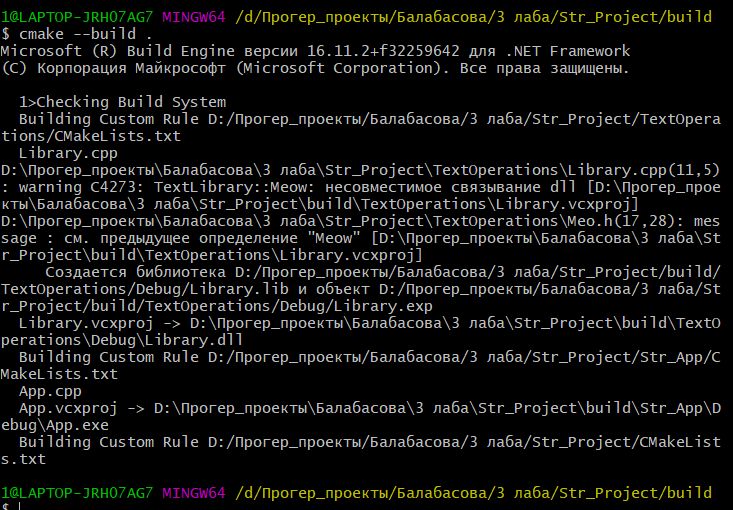


Рисунок 5 – пример выполнения команды сборки CMakeLists

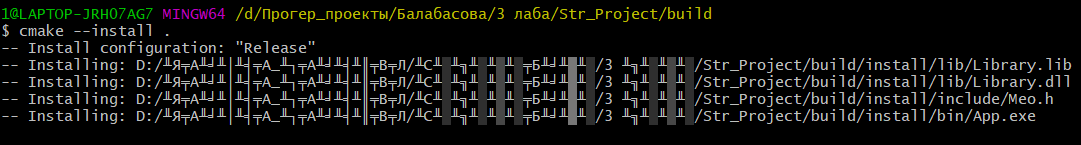


Рисунок 6 – пример выполнения команды установки CMakeLists

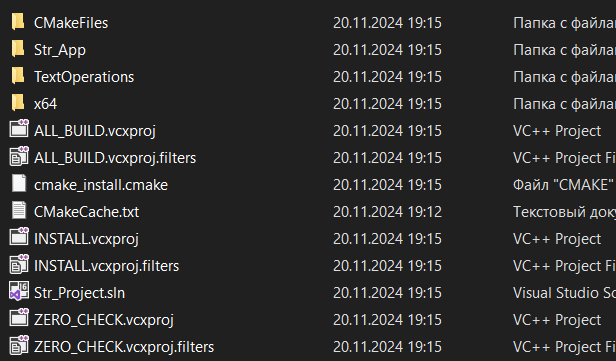


Рисунок 7 – демонстрация сборки в проводнике

**Вывод:**

Цель лабораторной работы заключалась в создании программного комплекса, состоящего из консольного приложения и динамической библиотеки для выполнения операций с текстовыми файлами, включая такие операции, как соединение файлов, разрезание слов по длине и изменение порядка слов. В ходе работы были выполнены следующие задачи:

• Разработаны **CMakeLists.txt** файлы для автоматизации сборки, установки, очистки и удаления компонентов программного комплекса. CMake файлы для библиотеки и основного приложения созданы отдельно, что позволяет гибко управлять каждой частью проекта и облегчить конфигурацию сборки.

• Создана динамическая библиотека **TextOperations**, содержащая функции для работы с текстом.

• Реализованы цели в **CMake** для библиотеки:

* **compile** — выполняет компиляцию библиотеки и создает динамическую библиотеку .dll.
* **install** — устанавливает библиотеку и заголовочные файлы в указанные папки lib и include.
* **clean** — очищает папку от ранее скомпилированных файлов.
* **uninstall** — удаляет установленную библиотеку и заголовочные файлы.

• Написано консольное приложение **Str\_App** с соответствующим **CMakeLists.txt** файлом, который позволяет:

* Скомпилировать приложение (compile).
* Установить приложение в указанную папку (install).
* Очистить скомпилированные файлы (clean).
* Удалить установленное приложение (uninstall).

• Приложение поддерживает интерактивный ввод данных от пользователя, включая ввод команд для выполнения операций с текстом, а также вывод справки по командам через ключ --help.

• Обеспечена обработка ошибок: проверка корректности входных данных, обработка ошибок при некорректных входных данных.

В результате выполнения лабораторной работы был создан программный комплекс, соответствующий требованиям задания. Комплекс включает приложение и динамическую библиотеку, обеспечивающие выполнение операций с текстом в интерактивном режиме с обработкой ошибок и выводом результатов. Лабораторная работа продемонстрировала практическое применение концепций модульной разработки, использования динамических библиотек и автоматизации сборки с помощью **CMake**.