Задание 1 (1 балл)

ТП

Дедлайн - 11.04.2021 20:59

Необходимо создать структуру проекта в CMake, поддерживающую следующие возможности

Алгоритм действия:

1. Исходный код лежит в трех папках A, B, C.
2. В папке A лежит скрипт на питоне, preparing.py, который генерирует хедер index.h.
3. В папке B лежит библиотека со своим собственным CMakeLists.txt, главный хедер - lib.h.
4. (Простой путь) В папке C лежит два файла main.h и main.cpp, в main.h есть строчка #include "A/index.h" и строчка "include "B/lib.h".
5. (Сложный путь) В папке C помимо файлов из пункта 4 лежит файл main_test.cpp и некоторые другие файлы, которые запускают сборку тестов через google test. https://ru.wikipedia.org/wiki/Google_C%2B%2B_Testing_Framework В файле main_test.cpp должен быть запуск тестов, а в другом cpp-файле, который собирается в один executable - должны быть реализованы 2 теста: один проверяет, что A/index.h делает корректные вещи, второй проверяет правильность реализации библиотеки.
6. После сборки проекта:
    - Должна быть создана папка bin, в которой должен быть лежать исполняемый файл C с главной точкой входа main.cpp
    - Должна быть создана папка lib, в которой должны лежать собранные динамические библиотеки.
    - В папке bin дополнительно должен лежать исполняемый файл CTest с запуском тестов, полученный после сборки C/main_test.cpp с исходными файлами, содержащие тесты.

Задание 2 (2 балла)

Поднять окружение

Дедлайн - 11.04.2021 20:59

Есть проект - https://github.com/akhtyamovpavel/PatternsCollection
Необходимо сделать окружение разработчика, в котором этот проект запускается. Для этого необходимо установить необходимые пакеты + описать сценарий, как эти пакеты должны ставиться.

Необходимо написать необходимые действия для работоспособности всех бинарных файлов.

Задание 3 (3 балла)

Кросс-компиляция

Дедлайн - 11.04.2021 20:59

Есть имеющийся проект на CMake, необходимо сделать этот проект cross-компилируемым!

Предыстория

Представим себе, что у нас есть небольшая плата, которая имеет arm-архитектуру, но собирать сложные библиотеки на плате - плохая затея. Поэтому на машину (ноутбук, персональный ноутбук) ставится Toolchain для сборки и Sysroot для эмуляции arm-архитектуры.

Вашей целью является адаптировать имеющийся CMake-проект под сборку для arm-архитектуры.

Сценарий исполнения:

- Собирается проект с указанием пути к Toolchain
- Готовится набор библиотек/хедеров/исполняемых файлов для переноса на машину под arm-архитектурой
- Файлы переносятся на плату (делается проверяющей системой автоматически)

- На плате собирается еще один проект, который подключает хедеры/библиотеки из предыдущего пункта

Алгоритм действия

1. Клонируйте репозиторий: https://github.com/akhtyamovpavel/TechProgSimpleLibrary (он собирается под Linux и Mac спокойно)
    a. В нем есть две цели: Main и MainLib, которые необходимо и запустить под ARM архитектурой!
2. Скачайте тулчейн по ссылке отсюда: https://releases.linaro.org/components/toolchain/binaries/latest-7/arm-linux-gnueabihf/gcc-linaro-7.5.0-2019.12-i686_arm-linux-gnueabihf.tar.xz
3. Научитесь добавлять скачанный компилятор для сборки CMake (см. раздел “Параметры запуска сборки проекта”)
4. Скачайте sysroot для проверки проекта: http://releases.linaro.org/components/toolchain/binaries/7.5-2019.12/armv8l-linux-gnueabihf/sysroot-glibc-linaro-2.25-2019.12-armv8l-linux-gnueabihf.tar.xz
5. Скачайте QEMU для виртуализации запуска программ (можно подставить через apt-get)
    a. Пользователям mac-а советуется поставить виртуальную машину для выполнения заданий
6. Для тестирования задания можно использовать команду: qemu-arm -L <path/to/sysroot> <path/to/executable>
7. Однако если пункт 6 у вас работает, то это не значит, что будет работать в любом месте и на любой архитектуре. Для дистрибуции пакетов необходимо выполнять команду **make install**, поддержку которой вам необходимо будет реализовать. *Запускаться код будет на другой машине через Main.*
8. После выполнения команды make install вы обнаружите, что у вас не хватает путей к библиотекам, необходимо исправить эту проблему: ключевое слово для гугла - rpath. *Тестирующая система запускает на другой машине (с ARM-архитектурой) файл MainLib с определенным путем, куда именно должен быть положен MainLib - можно посмотреть в оригинальном CMakeLists.txt или во втором задании*
9. Если вы прошли пункт 8, то необходимо правильно прописать опции установки. На машине с arm-архитектурой будет производиться сборка cmake-проекта, который знает о том, что файлы для сборки проекта лежит в определенной папке. Ваша цель - правильно указать параметры, чтобы сборка прошла успешно! Если получится - вы получаете полный балл за задание!

Задание 4 (2 балла)

Кодогенерация

Дедлайн 18.04.2021 20:59

https://github.com/magdasta/instagram-7.7.-decompiled-/blob/master/assets/filters/lo_fi/map.png

Это разрешение размером 3x256, это преобразование входной фотографии.

Необходимо сделать две вещи:

- Используя OpenCV, загрузить изображение из файла
- Для каждого пикселя, необходимо перевести красный цвет в значение пикселя в первой строке, зеленого - во второй, синего - в третьей
- После этого напишите кодогенерацию этого массива, чтобы использовать этот массив из g++
- Сравните времена работы этих подходов

Кодогенерация должна быть сделана через CMake.

Задание 5 (6 баллов)

Пересборка проекта под Bazel/Gradle

Дедлайн: 18.04.2021 20:59

Есть замечательный проект: https://github.com/akhtyamovpavel/PatternsCollection

Необходимо сделать так, чтобы он собирался через Bazel или Gradle!

- (2 балла) Собирается все, кроме Decorator под x86
- (2 балла) Собирается Decorator под x86
- (2 балла) Собирается проект под Arm

Оценка за курс Системы Сборки

Сумма баллов за задания - 14.

Оценка за курс Систем Сборки: сумма количества баллов, полученные за задания - 3.