Задание 3. Кросс-компиляция (кроме 120)

Дедлайн: 01 апреля 2021 г., 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/aarch64-linux-gnu/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar.xz
3. Научитесь добавлять скачанный компилятор для сборки CMake (см. раздел “Параметры запуска сборки проекта”)
4. Скачайте sysroot для проверки проекта: sysroot-glibc-linaro-2.25-2019.12-aarch64-linux-gnu.tar.xz
5. Скачайте QEMU для виртуализации запуска программ (можно подставить через apt-get: qemu-arm)

    a. Пользователям mac-а советуется поставить виртуальную машину для выполнения заданий
6. Для тестирования задания можно использовать команду: qemu-aarch64 -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. *Тестирующая система запускает на другой машине Main с определенным путем, как именно путь устроен - можно посмотреть во втором задании*
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. (Подсказка: CMAKE_INSTALL_PREFIX - путь, куда будет установлены файлы после запуска команды make install. Важно: не исполняйте команду от sudo - иначе потом придется удалять файлы - а это неприятно)*
9. Если вы прошли пункт 8 (за это вы получите 0.3 балла), 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 лежат в определенной папке. Ваша цель - правильно указать параметры, чтобы сборка прошла успешно! Если получится - вы получаете полный балл за задание! *Подсказка: попробуйте собрать код, используя собранную библиотеку в совершенно стороннем CMake-файле.*
10. **ВАЖНО! Система работает нестабильно на этом задании, поэтому время между запусками проверяющей системы будет увеличено! Просьба выполнять задание заранее!**

Требования

1. Используйте приватный репозиторий на github.com, при помощи которого вы сдали задание 0. Не забудьте проверить, что в коллабораторах есть techprogchecker.
2. В приватном репозитории создайте ветку task3, в нем создайте папку task3.
3. Положите созданный и работающий CMake проект в папку task3 в ветке task3.
4. Создайте pull request из ветки task3 в ветку master, добавьте ревьюера в pull request и не сливайте этот pull request!

Параметры запуска сборки проекта

Один из вариантов сборки проекта будет запускаться следующими командами:

```
mkdir build
cd build
cmake -DARM_BUILD=ON -DTOOLCHAIN=<path/to/toolchain/bin> -
DCMAKE_INSTALL_PREFIX=<path/to/install/prefix> ..
make
make install
```

При этом проверяющая система передает эти параметры, не стоит костылить путь руками

Критерии оценивания

- На ARM-машине запускается Main - 0.15 балла
- На ARM-машине запускается MainLib - 0.15 балла
- На ARM-машине собирается и запускается код - 0.15 балла
- Если первые три пункта выполнены, то добавляется еще 0.05 балла

В таблицу ставится балл, умноженный на 10.

Полезная информация

- https://github.com/victor-yacovlev/mipt-diht-caos/tree/master/practice/arm - как работать с qemu и эмуляцией под arm
- https://github.com/akhtyamovpavel - здесь можно поискать некоторые подсказки