Лабораторная работа №2

**Использование библиотек**

## Инструментарий и требования к работе

Работа выполняется на C (C11 и новее). На сервере сборка под C17.

# Задание

Необходимо написать программу, которая позволит найти временное смещение одного аудио-файла относительно другого методом кросс-корреляции ([cross-correlation](https://en.wikipedia.org/wiki/Cross-correlation)).

Аргументы программе передаются через командную строку в одном из следующих форматов:

| # | Формат | Пояснение |
| --- | --- | --- |
| 1 | **<file>** | Один двухканальный файл. Если каналов не 2, то необходимо сообщить о неверных данных и завершиться с ненулевым кодом. |
| 2 | **<file1> <file2>** | Два файла. Для обработки берётся первый канал из каждого файла. |

Для работы с данными (например, чтения аудио-файлов) необходимо использовать библиотеку ffmpeg. Аудио-файлы могут быть следующих форматов:

1. FLAC (Free Lossless Audio Codec)
2. MP2 (MPEG audio layer 2)
3. MP3 (MPEG audio layer 3)
4. Opus
5. AAC (Advanced Audio Coding)

Разница выводится в миллисекундах относительно первого файла/канала. Если второй файл/канал позже первого, то значение положительно, иначе – отрицательно.

Помимо найденной разницы необходимо вывести sample rate. Если у входных файлов sample rate разный, то:

* либо необходимо завершиться с сообщением о том, что это не поддерживается (и ненулевым кодом возврата),
* либо выполнить передискретизацию в больший sample rate.

Формат вывода: "delta: %i samples\nsample rate: %i Hz\ndelta time: %.3f ms\n"

Пример:

delta: 5644800 samples

sample rate: 44100 Hz

delta time: 128.000 ms

Кроме ffmpeg разрешено также использовать библиотеку fftw. Другие сторонние библиотеки использовать запрещено (стандартную библиотеку С использовать можно и нужно).

| **Библиотека** | **Версия** | **Подключение (пример)** |
| --- | --- | --- |
| ffmpeg | 6.1 | #include <libavformat/avformat.h> |
| fftw | 3.3 | #include <fftw3.h> |

Для вашего репозитория (на github) исходный файл, содержащий функцию main, должен лежать в корне репозитория.

**Внимание!** В репозитории не должно быть любых файлов ffmpeg и fftw. Вы можете их хранить рядом локально, но в этом случае они должны быть занесены .gitignore, а лучше в .git/info/exclude.

Пример:

| /<github\_repo\_name>  ├── main.c /\* your src file \*/  ├── my\_src.c /\* [optional] your src file \*/  ├── src /\* [optional] source dir \*/  ├── my\_src\_file1.c /\* [optional] your src file \*/  ├── my\_src\_file2.c /\* [optional] your src file \*/  ├── my\_header.h /\* [optional] your header file \*/  ├── include /\* [optional] header dir \*/  ├── my\_header\_file1.h /\* [optional] your hdr file \*/  ├── my\_header\_file2.h /\* [optional] your hdr file \*/  ├── .gitignore  ├── .clang-format  ├── … |
| --- |

Должно присутствовать разумное разделение на файлы и правильно сформированные заголовочные файлы с минимальными зависимостями.

Если на вход программе поданы некорректные данные, то необходимо завершаться с человекочитаемым сообщением о возникшей ошибке (по-английски) и правильным кодом из return\_codes.h.

Полезности:

* [Download FFmpeg](https://ffmpeg.org/download.html#get-sources) – ссылка на скачивание, нужны shared исходники.
* [FFTW Download Page](http://www.fftw.org/download.html) – ссылка на скачивание.
* <https://skkv-itmo.gitbook.io/ct-c-cpp-course/build/building-program>
* <https://skkv-itmo.gitbook.io/ct-c-cpp-course/build/build-systems>
* Компиляция из командной строки: <https://www.youtube.com/watch?v=t2iHVBZdhdA>
* Подключение внешних библиотек из командной строки: <https://www.youtube.com/watch?v=5w3CZvsanKE>
* Системы сборки и подключение внешних библиотек: <https://skkv-itmo.gitbook.io/ct-c-cpp-course/build/libs>

Защита:

На защите помимо объяснения своего кода и как он у вас собирается будет необходимо продемонстрировать навыки сборки вашего кода на стороннем компьютере.

Также принимающий может дать задание на защите, заключающееся в модификации кода.