

# Лабораторная работа 2

## Декодирование PNG

### Цель работы

Изучить особенности работы с двоичными файлами, структурами, директивами препроцессора и сторонними библиотеками в С. Закрепить навык чтения спецификаций.

### Стандарт языка и требования к работе

С99 и новее. [Требования к работам](#)

### Описание

Необходимо реализовать программу, которая декодирует изображение в формате PNG в изображение в формате PNM.

***Должны поддерживаться:*** серые, цветные и палитровые изображения (Color Type 0, 2 и 3), 8 бит на канал.

***Не нужно поддерживать:*** работу с прозрачностью, interlacing и цветовую коррекцию (гамму и пр.).

Для разжатия deflate-потока (и только для этого) необходимо использовать одну из следующих библиотек: zlib, libdeflate или isa-l (на ваше усмотрение).

**Критерий “Все библиотеки”:** должно поддерживаться использование каждой из трёх библиотек: zlib, libdeflate и isa-l.

**Спецификация:** [Portable Network Graphics \(PNG\) Specification \(Third Edition\)](#) и [PNG \(Portable Network Graphics\) Specification, Version 1.2](#)

## Использование сторонних библиотек

Кроме zlib, libdeflate и isa-l запрещено использовать какие-либо другие сторонние библиотеки (стандартную библиотеку C использовать можно и нужно).

Библиотека	Версия	Подключение
zlib	1.2.13	#include <zlib.h>
libdeflate	1.17	#include <libdeflate.h>
isa-l	2.30	#include <include/igzip_lib.h>

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

```

/<github_repo_name>
├── main.c /* your src file*/
├── my_header.h /* [optional] your header file */
├── .gitignore
├── .clang-format
└── ...

```

**Внимание!** В репозитории не должно быть любых файлов zlib, libdeflate и isa-l. Вы можете их хранить рядом локально, но в этом случае они должны быть занесены .gitignore.

**Выбор библиотеки должен определяться макросом,** который указывается ключом компилятора (аналог #define). Гарантируется, что при компиляции будет указан ровно один из трёх макросов.

Библиотека	zlib	libdeflate	isa-l
Макрос	ZLIB	LIBDEFLATE	ISAL

В случае, если при компиляции был указан макрос, определяющий подключение библиотеки, которую ваша программа не поддерживает, то необходимо командой препроцессора #error() выводить сообщение о

том, что данная библиотека не поддерживается. Подробнее:

<https://en.cppreference.com/w/c/preprocessor/error>

#if defined(ZLIB) #error ...	#ifdef ZLIB #error ...
---------------------------------	---------------------------

## Формат аргументов командной строки

Аргументы программе передаются через командную строку:

<имя\_входного\_PNG\_файла> <имя\_выходного\_PNM\_файла>

## Входной файл

PNG-изображение. Не гарантируется корректность данных внутри файла (может быть любой файл, в том числе не PNG или файл с ошибками).

Для просмотра структуры png по блокам (chunks) можно использовать *tweakPNG* (Windows) и *pngcheck* (Unix/MacOS).

## Выходной файл

Формат выходных изображений: PNM (P5 или P6). Во всех PNM файлах (pgm, ppm) комментарии отсутствуют.

Тип входного файла (Color Type)	0	2	3
Тип выходного файла	P5	P6	P5, если возможно, иначе P6

Формат представлен ниже:

P5 (PGM)	P5\n<width> <height>\n255\n<Gray_data>
P6 (PPM)	P6\n<width> <height>\n255\n<RGB_data>

*Windows* можно установить [WIC Codec](#) для просмотра PNM через стандартный просмотрщик изображений или использовать Gimp и Photoshop (поддерживают PNM “из коробки”). *Linux / Mac OS*: поддерживают PNM “из коробки”.

## Автотесты

Начиная с этой работы, автотесты на Github можно запустить только вручную через вкладку Actions. Функционал с переоткрытием Pull Request отключен.

**Как запускать тесты описано в памятке:**

<https://skkv-itmo.gitbook.io/c-cpp-cource/organization/labs/cicd-github#run-autotests>

Поле	Возможные значения	Значение по умолчанию	Описание
grey_test	true false	true	Изображение в оттенках серого
rgb_test	true false	true	Изображение в rgb
neg_test	true false	true	Изображение, не соответствующее спецификации

Пример запуска (терминал открыт из папки корня репо):

```
gh workflow run BuildTest -f rgb_test=false -f grey_test=true
-f neg_test=false
```

Таким образом происходит запуск автотестов, где проверяется работа только с серыми картинками. Несколько полей могут принимать значение true.

**Подробнее о том, как можно смотреть запуски из вашей IDE:**

<https://skkv-itmo.gitbook.io/c-cpp-cource/ide-features/gh-actions-ide>