

Индивидуальное задание

Библиотека для тестирования производительности различных алгоритмов свертки, входящих в состав сверточных нейронных сетей.

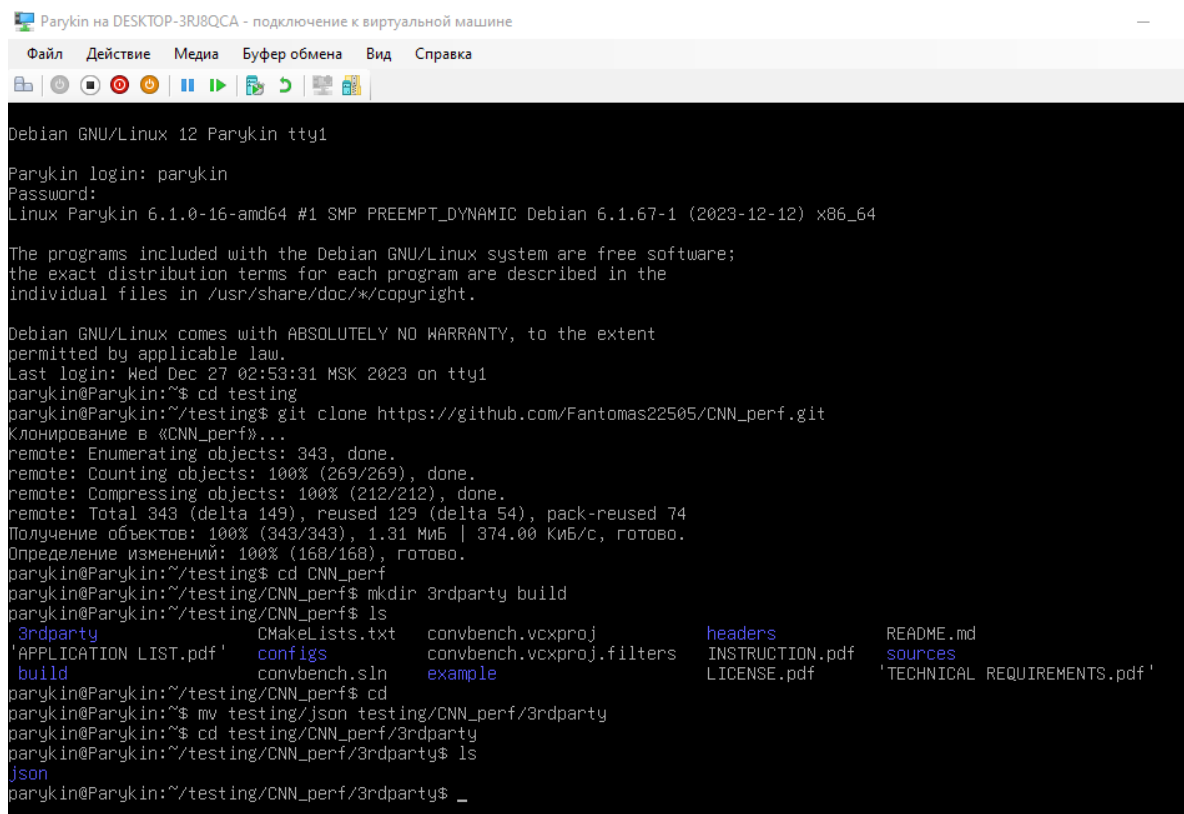
Цель работы:

1. Выполнить *Fork* библиотеки в свою учетную запись;
2. Изучить материалы открытой библиотеки;
3. Провести установку библиотеки на свой локальный ПК;
4. Провести апробацию (например, с использованием представленных в репозитории примеров).

Выполнение работы:

1. Произвел установку необходимых зависимостей с помощью команд:

```
sudo apt-get install build-essential  
sudo apt-get install cmake  
sudo apt-get install git
```
2. Выполнил *fork* и клонирование [репозитория проекта](#) и [библиотеки json](#). Подготовил необходимые каталоги, как показано на рисунке 1.



```
Debian GNU/Linux 12 Parykin tty1  
Parykin login: parykin  
Password:  
Linux Parykin 6.1.0-16-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.67-1 (2023-12-12) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Dec 27 02:53:31 MSK 2023 on tty1  
parykin@Parykin:~$ cd testing  
parykin@Parykin:~/testing$ git clone https://github.com/Fantomas22505/CNN_perf.git  
Клонирование в «CNN_perf»...  
remote: Enumerating objects: 343, done.  
remote: Counting objects: 100% (269/269), done.  
remote: Compressing objects: 100% (212/212), done.  
remote: Total 343 (delta 149), reused 129 (delta 54), pack-reused 74  
Получение объектов: 100% (343/343), 1.31 Миб | 374.00 Киб/с, готово.  
Определение изменений: 100% (168/168), готово.  
parykin@Parykin:~/testing$ cd CNN_perf  
parykin@Parykin:~/testing/CNN_perf$ mkdir 3rdparty build  
parykin@Parykin:~/testing/CNN_perf$ ls  
3rdparty      CMakeLists.txt      convbench.vcxproj      headers          README.md  
'APPLICATION LIST.pdf'  configs              convbench.vcxproj.filters  INSTRUCTION.pdf  sources  
build          convbench.sln        example                 LICENSE.pdf      'TECHNICAL REQUIREMENTS.pdf'  
parykin@Parykin:~/testing/CNN_perf$ cd  
parykin@Parykin:~$ mv testing/json testing/CNN_perf/3rdparty  
parykin@Parykin:~$ cd testing/CNN_perf/3rdparty  
parykin@Parykin:~/testing/CNN_perf/3rdparty$ ls  
json  
parykin@Parykin:~/testing/CNN_perf/3rdparty$ _
```

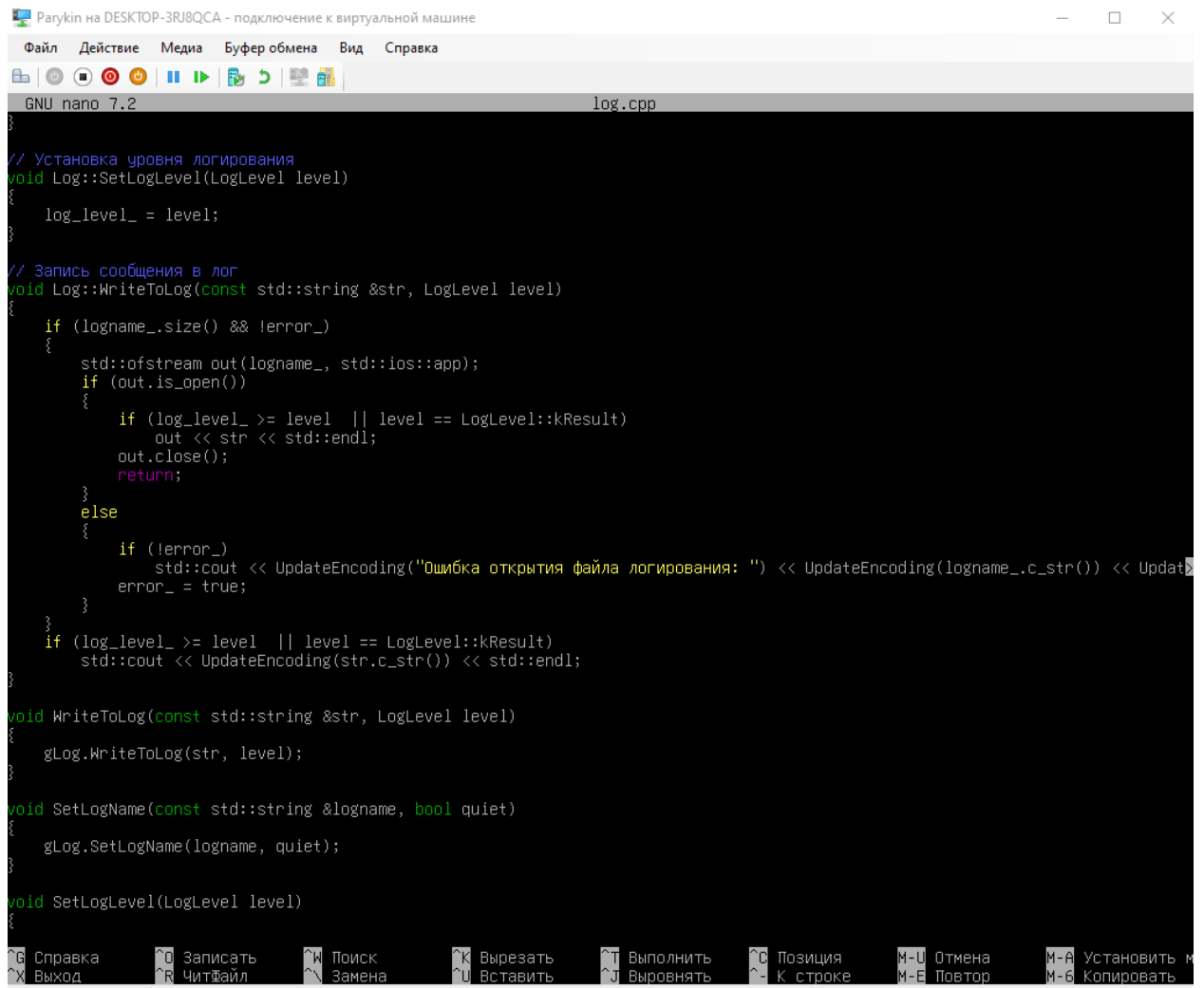
Рисунок 1 - Подготовка к работе с проектом

3. При попытке сборки проекта обнаружил ошибку функции *UpdateEncoding* файла *log.cpp*, как показано на рисунке 2.

```
parykin@Parykin:~/testing/CNN_perf/3rdparty$ ls
json
parykin@Parykin:~/testing/CNN_perf/3rdparty$ cd ..
parykin@Parykin:~/testing/CNN_perf$ cd build
parykin@Parykin:~/testing/CNN_perf/build$ cmake ..
-- The CXX compiler identification is GNU 12.2.0
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using the multi-header code from /home/parykin/testing/CNN_perf/3rdparty/json/include/
-- Configuring done
-- Generating done
-- Build files have been written to: /home/parykin/testing/CNN_perf/build
parykin@Parykin:~/testing/CNN_perf/build$ make
[ 9%] Building CXX object CMakeFiles/convbench.dir/sources/args.cpp.o
[ 18%] Building CXX object CMakeFiles/convbench.dir/sources/avx512conv.cpp.o
[ 27%] Building CXX object CMakeFiles/convbench.dir/sources/avxconv.cpp.o
[ 36%] Building CXX object CMakeFiles/convbench.dir/sources/baseconv.cpp.o
[ 45%] Building CXX object CMakeFiles/convbench.dir/sources/jsonargs.cpp.o
[ 54%] Building CXX object CMakeFiles/convbench.dir/sources/log.cpp.o
/home/parykin/testing/CNN_perf/sources/log.cpp: In member function 'void Log::WriteToLog(const std::string&, LogLevel)':
/home/parykin/testing/CNN_perf/sources/log.cpp:49:30: error: 'UpdateEncoding' was not declared in this scope
 49 |         std::cout << UpdateEncoding("Ошибка открытия файла логирования: ") << UpdateEncoding(logname_.c_str())
    |                               ^~~~~~
< UpdateEncoding(" вывод осуществляется в консоль.") << std::endl;
    |                               ^~~~~~
/home/parykin/testing/CNN_perf/sources/log.cpp:54:22: error: 'UpdateEncoding' was not declared in this scope
 54 |         std::cout << UpdateEncoding(str.c_str()) << std::endl;
    |                       ^~~~~~
make[2]: *** [CMakeFiles/convbench.dir/build.make:146: CMakeFiles/convbench.dir/sources/log.cpp.o] Ошибка 1
make[1]: *** [CMakeFiles/Makefile2:99: CMakeFiles/convbench.dir/all] Ошибка 2
make: *** [Makefile:91: all] Ошибка 2
parykin@Parykin:~/testing/CNN_perf/build$
```

Рисунок 2 - Ошибка файла *log.cpp*

4. Попытался устранить ошибку путем замены *UpdateEncoding* в файле *log.cpp* строкой текста, как показано на рисунках 3-4.



```
GNU nano 7.2 log.cpp
// Установка уровня логирования
void Log::SetLogLevel(LogLevel level)
{
    log_level_ = level;
}

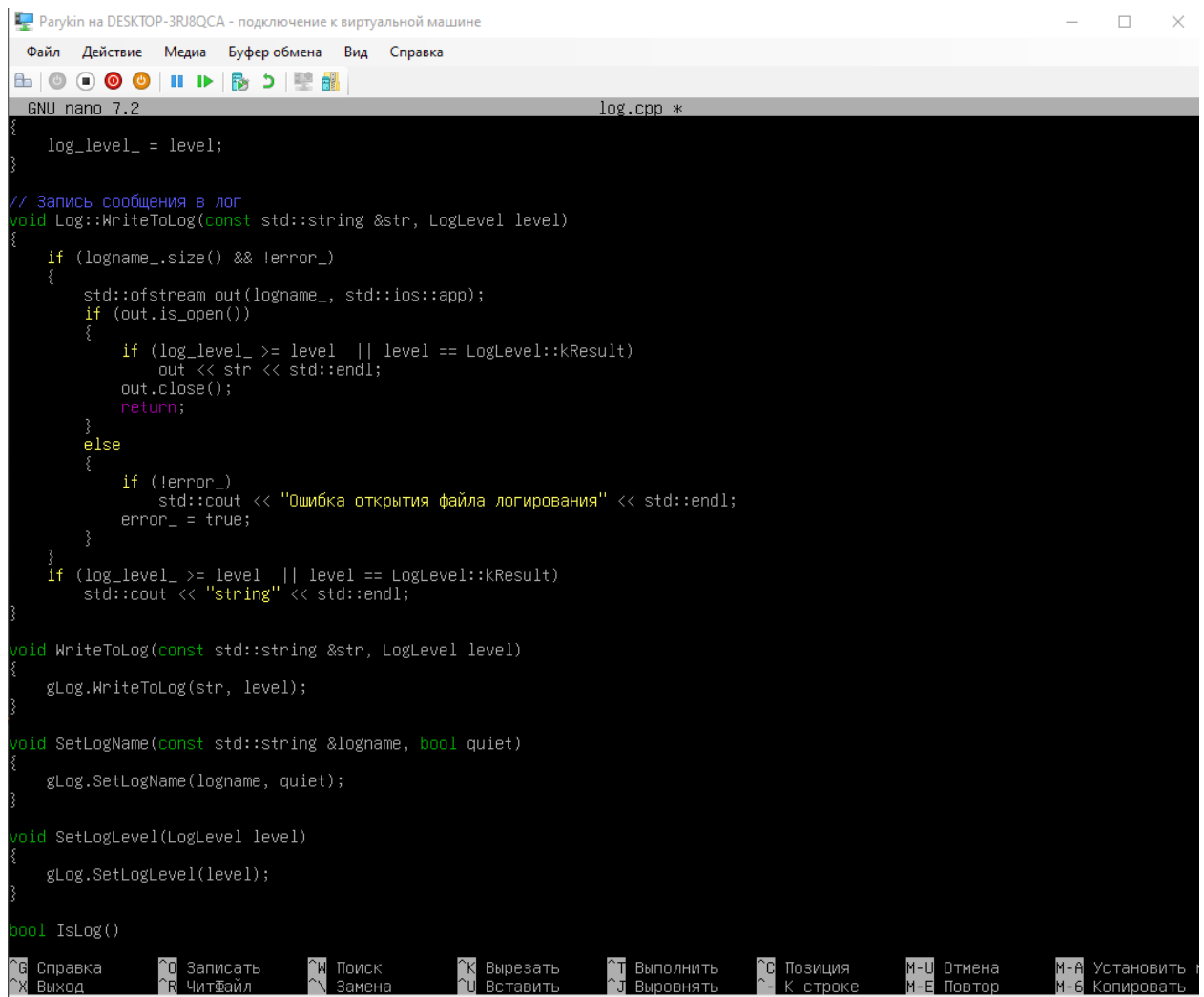
// Запись сообщения в лог
void Log::WriteToLog(const std::string &str, LogLevel level)
{
    if (logname_.size() && !error_)
    {
        std::ofstream out(logname_, std::ios::app);
        if (out.is_open())
        {
            if (log_level_ >= level || level == LogLevel::kResult)
                out << str << std::endl;
            out.close();
            return;
        }
        else
        {
            if (!error_)
                std::cout << UpdateEncoding("Ошибка открытия файла логирования: ") << UpdateEncoding(logname_.c_str()) << UpdateEncoding("\n");
            error_ = true;
        }
    }
    if (log_level_ >= level || level == LogLevel::kResult)
        std::cout << UpdateEncoding(str.c_str()) << std::endl;
}

void WriteToLog(const std::string &str, LogLevel level)
{
    gLog.WriteToLog(str, level);
}

void SetLogName(const std::string &logname, bool quiet)
{
    gLog.SetLogName(logname, quiet);
}

void SetLogLevel(LogLevel level)
{
}
```

Рисунок 3 - Исходный код



```
GNU nano 7.2 log.cpp *
{
    log_level_ = level;
}

// Запись сообщения в лог
void Log::WriteToLog(const std::string &str, LogLevel level)
{
    if (logname_.size() && !error_)
    {
        std::ofstream out(logname_, std::ios::app);
        if (out.is_open())
        {
            if (log_level_ >= level || level == LogLevel::kResult)
                out << str << std::endl;
            out.close();
            return;
        }
        else
        {
            if (!error_)
                std::cout << "Ошибка открытия файла логирования" << std::endl;
            error_ = true;
        }
    }
    if (log_level_ >= level || level == LogLevel::kResult)
        std::cout << "string" << std::endl;
}

void WriteToLog(const std::string &str, LogLevel level)
{
    gLog.WriteToLog(str, level);
}

void SetLogName(const std::string &logname, bool quiet)
{
    gLog.SetLogName(logname, quiet);
}

void SetLogLevel(LogLevel level)
{
    gLog.SetLogLevel(level);
}

bool IsLog()
```

Рисунок 4 - Измененный код

5. После внесения изменений ошибка файла *log.cpp* была устранена, однако возникла аналогичная ошибка в файле *main.cpp*, как показано на рисунке 5.

```
Parykin на DESKTOP-3RJ8QCA - подключение к виртуальной машине
Файл Действие Медиа Буфер обмена Вид Справка

if (!error_)
    std::cout << "Ошибка открытия файла логирования" << std::endl;
    error_ = true;
}
}
if (log_level_ >= level || level == LogLevel::kResult)
    std::cout << "string" << std::endl;
}

void WriteToLog(const std::string &str, LogLevel level)
{
    gLog.WriteToLog(str, level);
}

void SetLogName(const std::string &logname, bool quiet)
{
    gLog.SetLogName(logname, quiet);
}

void SetLogLevel(LogLevel level)
{
    gLog.SetLogLevel(level);
}

bool IsLog()

parykin@Parykin:~/testing/CNN_perf/sources$ cd ..
parykin@Parykin:~/testing/CNN_perf$ cd build
parykin@Parykin:~/testing/CNN_perf/build$ cmake ..
ma-- Using the multi-header code from /home/parykin/testing/CNN_perf/3rdparty/json/include/
-- Configuring done
-- Generating done
-- Build files have been written to: /home/parykin/testing/CNN_perf/build
parykin@Parykin:~/testing/CNN_perf/build$ make
[ 9%] Building CXX object CMakeFiles/convbench.dir/sources/log.cpp.o
[ 18%] Building CXX object CMakeFiles/convbench.dir/sources/main.cpp.o
/home/parykin/testing/CNN_perf/sources/main.cpp: In function 'int main(int, char**)':
/home/parykin/testing/CNN_perf/sources/main.cpp:31:26: error: 'UpdateEncoding' was not declared in this scope
 31 |         std::cout << UpdateEncoding(kHelpMessage) << std::endl;
    |                         ^~~~~~
/home/parykin/testing/CNN_perf/sources/main.cpp:82:22: error: 'UpdateEncoding' was not declared in this scope
 82 |         std::cout << UpdateEncoding(kErrorMessage) << err.what() << std::endl;
    |                      ^~~~~~
make[2]: *** [CMakeFiles/convbench.dir/build.make:160: CMakeFiles/convbench.dir/sources/main.cpp.o] Ошибка 1
make[1]: *** [CMakeFiles/Makefile2:99: CMakeFiles/convbench.dir/all] Ошибка 2
make: *** [Makefile:91: all] Ошибка 2
parykin@Parykin:~/testing/CNN_perf/build$
```

Рисунок 5 - Ошибка файла *main.cpp*

6. Внес аналогичные изменения, как показано на рисунках 6-7.

```
#include <iostream>
#include "headers/args.h"
#include "headers/jsonargs.h"
#include "headers/baseconv.h"
#include "headers/log.h"

#ifdef _WIN32
#include "headers/updencoding.h"
#endif

float gEpsilon = kDefEps;

int main(int argc, char *argv[])
{
#ifdef _WIN32
    setlocale(LC_ALL, "ru-RU");
#endif

    //SetLogLevel(LogLevel::kDebug);

    try {
        // Инициализация программного компонента разбора параметров командной строки
        std::shared_ptr<BaseArgs> arg = std::make_shared<CmdArgs>(argc, argv);
        std::string str;

        // Проверка на ключ затребуемый вывод справки
        if (((CmdArgs *)arg.get())->FindHelp() || argc==1)
        {
            std::cout << UpdateEncoding(kHelpMessage) << std::endl;
            return 0;
        }

        // Проверить на наличие Json-конфигурации
        std::string config, logname;
        if (((CmdArgs *)arg.get())->FindJsonConfig(config))
            arg = std::make_shared<JsonArgs>(config);

        // Проверить необходимость логирования и тихого режима, установить их для лога
        bool quiet = arg->CheckLog(logname);
        SetLogName(logname, quiet);

        // Описание процессора
```




Рисунок 6 - Исходный код

```

#include <iostream>
#include "headers/args.h"
#include "headers/jsonargs.h"
#include "headers/baseconv.h"
#include "headers/log.h"

#ifdef _WIN32
#include "headers/updencoding.h"
#endif

float gEpsilon = kDefEps;

int main(int argc, char *argv[])
{
#ifdef _WIN32
    setlocale(LC_ALL, "ru-RU");
#endif

    //SetLogLevel(LogLevel::kDebug);

    try {
        // Инициализация программного компонента разбора параметров командной строки
        std::shared_ptr<BaseArgs> arg = std::make_shared<CmdArgs>(argc, argv);
        std::string str;









        // Проверка на ключ требующий вывод справки
        if (((CmdArgs *)arg.get())->FindHelp() || argc==1)
        {
            std::cout << "kHelpMessage" << std::endl;
            return 0;
        }

        // Проверить на наличие Json-конфигурации
        std::string config, logname;
        if (((CmdArgs *)arg.get())->FindJsonConfig(config))
            arg = std::make_shared<JsonArgs>(config);

        // Проверить необходимость логирования и тихого режима, установить их для лога
        bool quiet = arg->CheckLog(logname);
        SetLogName(logname, quiet);

        // Описание процессора
    }
}

```

 Справка
  Записать
  Поиск
  Вырезать
  Выполнить
  Позиция
  Отмена
  Установить м








 Выход
  ЧитФайл
  Замена
  Вставить
  Выровнять
  К строке
  Повтор
  Копировать

Рисунок 7 - Измененный код

7. Сборка проекта затруднена в связи с возникновением ошибок в других файлах проекта.

```

    {
        std::cout << "kErrorString" << err.what() << std::endl;
        if (IsLog())
            LOG_RESULT(std::string(kErrorString) + err.what() + "\n");
        return 1;
    }

    return 0;
}

parykin@Parykin:~/testing/CNN_perf/sources$ cd..
-bash: cd..: команда не найдена
parykin@Parykin:~/testing/CNN_perf/sources$ cd ..
parykin@Parykin:~/testing/CNN_perf$ cd build
parykin@Parykin:~/testing/CNN_perf/build$ cmake ..
-- Using the multi-header code from /home/parykin/testing/CNN_perf/3rdparty/json/include/
-- Configuring done
-- Generating done
-- Build files have been written to: /home/parykin/testing/CNN_perf/build
parykin@Parykin:~/testing/CNN_perf/build$ make
[ 9%] Building CXX object CMakeFiles/convbench.dir/sources/main.cpp.o
[ 18%] Building CXX object CMakeFiles/convbench.dir/sources/simpleconv.cpp.o
[ 27%] Building CXX object CMakeFiles/convbench.dir/sources/sseconv.cpp.o
[ 36%] Building CXX object CMakeFiles/convbench.dir/sources/Utils.cpp.o
/home/parykin/testing/CNN_perf/sources/Utils.cpp: In function 'std::string string_format(std::string, ...)':
/home/parykin/testing/CNN_perf/sources/Utils.cpp:19:9: error: 'strcpy_s' was not declared in this scope; did you mean 'strcpy'?
   19 |         strcpy_s(&formatted[0], n, fmt_str.c_str());
      |         ~~~~~^
      |         strcpy
/home/parykin/testing/CNN_perf/sources/Utils.cpp: In function 'std::string GetCpuInfo()':
/home/parykin/testing/CNN_perf/sources/Utils.cpp:160:25: error: variable 'std::array<int, 4> integerBuffer' has initializer but incomplete type
   160 |         std::array<int, 4> integerBuffer = {};
      |         ~~~~~^
/home/parykin/testing/CNN_perf/sources/Utils.cpp:163:27: error: variable 'std::array<char, 64> charBuffer' has initializer but incomplete type
   163 |         std::array<char, 64> charBuffer = {};
      |         ~~~~~^
/home/parykin/testing/CNN_perf/sources/Utils.cpp:167:44: error: variable 'constexpr const std::array<unsigned int, 3> functionIds' has initializer but incomplete type
   167 |         constexpr std::array<unsigned int, 3> functionIds = {
      |         ~~~~~^
make[2]: *** [CMakeFiles/convbench.dir/build.make:202: CMakeFiles/convbench.dir/sources/Utils.cpp.o] Ошибка 1
make[1]: *** [CMakeFiles/Makefile2:99: CMakeFiles/convbench.dir/all] Ошибка 2
make: *** [Makefile:91: all] Ошибка 2
parykin@Parykin:~/testing/CNN_perf/build$

```

Рисунок 8 - Сборка проекта

8. Осуществил *push* репозитория на *GitHub*, как показано на рисунке 9.

```

parykin@Parykin:~/testing/CNN_perf$ git remote add upstream https://github.com/Fantomas22505/CNN_perf.git
error: внешний репозиторий upstream уже существует
parykin@Parykin:~/testing/CNN_perf$ git add *
parykin@Parykin:~/testing/CNN_perf$ git commit -m "Ошибки везде ошибки"
Текущая ветка: main
Ваша ветка опережает «origin/main» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

ничего коммитить, нет изменений в рабочем каталоге
parykin@Parykin:~/testing/CNN_perf$ git push origin main
Username for 'https://github.com': Fantomas22505
Password for 'https://github.com':
Перечисление объектов: 71, готово.
Подсчет объектов: 100% (71/71), готово.
Сжатие объектов: 100% (62/62), готово.
Запись объектов: 100% (67/67), 709.30 КиБ | 3.64 МБ/с, готово.
Всего 67 (изменений 24), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (24/24), completed with 4 local objects.
To https://github.com/Fantomas22505/CNN_perf.git
   f4bb832..d534c6a  main -> main
parykin@Parykin:~/testing/CNN_perf$

```

Рисунок 9 - Выполнение *git push*

Вывод: выполнил *fork* библиотеки в свою учетную запись, изучил материалы открытой библиотеки, произвел клонирование репозитория на свой ПК, в процессе сборки проекта столкнулся с файловыми ошибками.