**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ**

(национальный исследовательский университет)

«**МАИ**»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Отчёт по практической работе

по дисциплине

«Операционные системы»

на тему:

«Архиватор»

Выполнили: студенты группы М3О-325Бк-19 Наштыков Д.Е.

Попов Н.П.

Проверил: доцент каф. 304, к.т.н. Грабовский М.Н.

Москва 2022

**Задание**

Создать базовый архиватор.

**Проектирование**

1. Формат файла архива:

Результирующий файл будет хранить в себе несжатые данные файлов вместе с метаинформацией, хранящейся в виде JSON-строки. Первые 4 байта файла представляют из себя без знаковое целое, представляющее из себя кол-во хранящихся в архиве файлов. Далее, последовательно, хранится мета информация и сами данные файлов.

В качестве интерфейса управления программой будут выступать аргументы командной строки:

* -d — если установлен, означает, что переданный входной файл-архив, и его необходимо разархивировать
* -i <название директории> - устанавливает входной файл
* -o <название выходной директории.файла> - устанавливает название выходного файла.архива

2. Управление кодом

В ходе разработки будет использована система контроля версий git. Так как над программой работают два человека, для командной работы будем использовать удаленный репозиторий GitHub(ссылка: <https://github.com/SHTYKOVYCH/OS-labs>).

Для сборки проекта используется система Cmake. Конфигурационный файл:

cmake\_minimum\_required(VERSION 3.0)

set(PROJECT arhv)

project(${PROJECT})

set(HEADERS

error\_codes.h

error\_handler.h

read-file.h

write\_file.h

parse-args.h

string-2-json.h

json\_stringify.h

read\_json\_from\_file.h

dir\_tree.h

archive.h

depack.h

)

set(SOURCES

main.c

error\_handler.c

read-file.c

write\_file.c

parse-args.c

string-2-json.c

json\_stringify.c

read\_json\_from\_file.c

dir\_tree.c

depack.c

archive.c

)

set(GXX\_FLAGS "-g -fsanitize=address")

set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} ${GXX\_FLAGS}")

add\_executable(${PROJECT} ${HEADERS} ${SOURCES})

**Результат**

out.dir

├── 3

│   └── 3

│   ├── 3

│   └── bash

├── rtw89

│   ├── cam.c

│   ├── cam.h

│   ├── coex.c

│   ├── coex.h

│   ├── core.c

│   ├── core.h

│   ├── debian

│   │   ├── changelog

│   │   ├── compat

│   │   ├── control

│   │   ├── copyright

│   │   ├── dkms.conf

│   │   ├── README.Debian

│   │   ├── README.source

│   │   ├── rtw89-dkms.postinst

│   │   ├── rtw89-dkms.prerm

│   │   ├── rules

│   │   └── source

│   │   └── format

│   ├── debug.c

│   ├── debug.h

│   ├── dkms.conf

│   ├── efuse.c

│   ├── efuse.h

│   ├── fw.c

│   ├── fw.h

│   ├── LICENSE

│   ├── mac80211.c

│   ├── mac.c

│   ├── mac.h

│   ├── Makefile

│   ├── pci.c

│   ├── pci.h

│   ├── phy.c

│   ├── phy.h

│   ├── ps.c

│   ├── ps.h

│   ├── README.md

│   ├── regd.c

│   ├── regd.h

│   ├── reg.h

│   ├── rtw8852a.c

│   ├── rtw8852a\_fw.bin

│   ├── rtw8852a.h

│   ├── rtw8852a\_rfk.c

│   ├── rtw8852a\_rfk.h

│   ├── rtw8852a\_rfk\_table.c

│   ├── rtw8852a\_rfk\_table.h

│   ├── rtw8852a\_table.c

│   ├── rtw8852a\_table.h

│   ├── sar.c

│   ├── sar.h

│   ├── ser.c

│   ├── ser.h

│   ├── suspend\_rtw89

│   ├── txrx.h

│   ├── util.c

│   └── util.h

├── test.dir

│   ├── test.dir2

│   │   └── test.file

│   └── test.file

├── test.dir2

│   └── test.file

└── test.file

Запуск программы:

./arhv -i out.dir

Запуск деархивации:

./arhv -d -i archive -o test.out