Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ**

Лабораторная работа No 5. Взаимодействие процессов в ОС GNU/Linux

тема

Преподаватель \_\_\_\_\_\_\_\_\_ А.С. Кузнецов

подпись, дата инициалы, фамилия

Студент КИ18-17/1б 031831229 \_\_\_\_\_\_\_\_\_ В.А. Прекель

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2019

Содержание

[Содержание 2](#_Toc24305909)

[1 Цель работы с постановкой задачи 4](#_Toc24305910)

[1.1 Цель работы 4](#_Toc24305911)

[1.2 Задача работы 4](#_Toc24305912)

[1.3 Описание и пояснение к работе 4](#_Toc24305913)

[2 Описание использованных при выполнении задания функций Linux API управления файлами 5](#_Toc24305914)

[3 Примеры использования (вызова) этих функций в представленном программном коде 8](#_Toc24305915)

[3.1 Обертки над функциями низкоуровнего ввода-вывода из VS (например, \_open()) или Linux API (например open()) (Lab\_04\_Lib/IOWrapper.c): 8](#_Toc24305916)

[3.2 Использование lseek() для перемещение позиции в файле на место указанной записи (Lab\_04\_Lib/RecordFile.c): 9](#_Toc24305917)

[3.1 Использование write() для записи записи в файл с записями (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305918)

[3.2 Использование close() для закрытия файла с записями (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305919)

[3.1 Использование read() для чтения записи из файла с записями (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305920)

[3.2 Использование ftruncate() для отсечения записей в файле (Lab\_04\_Lib/RecordFile.c): 10](#_Toc24305921)

[3.3 Использование access() для проверки файла на существование, читаемость и записываемость (Lab\_04\_Lib/RecordFile.c): 11](#_Toc24305922)

[4 Содержимое файла makefile. 11](#_Toc24305923)

[4.1 ./Makefile 11](#_Toc24305924)

[4.2 ./Lab\_04\_Lib/Makefile 11](#_Toc24305925)

[4.1 ./Lab\_04\_Console/Makefile 12](#_Toc24305926)

[5 Примеры работы программ в виде перехватов содержимого экрана 12](#_Toc24305927)

[5.1 Снимок экрана запуска № 1 (WSL Ubuntu 18.04, make, GCC) 12](#_Toc24305928)

[5.2 Снимок экрана запуска № 2 (Windows 10, mingw32-make, MSVC) 13](#_Toc24305929)

[5.3 Запуск №3 (WSL Ubuntu 18.04, make, Clang) 14](#_Toc24305930)

[5.4 Шестнадцатеричный дамп после запуска №4 17](#_Toc24305931)

# Цель работы с постановкой задачи

## Цель работы

Изучение особенностей межпроцессного взаимодействия в ОС GNU/Linux.

## Задача работы

Требуется: разработать две программы: первая реализует серверную часть, вторая — клиентскую часть. Обмен данными между ними организуется посредством механизма Internet-сокетов и протокола TCP либо UDP. Результат выполнения выводится на терминал/консоль. Должен использоваться интерфейс командной строки (CLI). При реализации обязательно использование изученных в лекционном курсе системных вызовов (ОС Linux), предназначенных для работы с сокетами. Программный код, относящийся к пользовательскому интерфейсу, должен быть физически отделен от кода, реализующего межпроцессное взаимодействие, и оба они, в свою очередь, отделены от кода реализации основной логики, например, вычислений. Допускается реализация одной программы в форме Windows-приложения, авторой — в форме Linux-приложения. Далее оговаривается функционал клиентской и серверной частей. Обе части должны быть устойчивы к некорректному пользовательскому вводу. В нечетных вариантах заданий используются потоковые сокеты, в четных — дейтаграммные сокеты.

**Вариант 17.** Клиент отсылает серверу элементы введенной пользователем квадратной матрицы, а также два введенных им числа – номер столбца и номер строки. Сервер принимает матрицу и номер столбца N и номер строки M, выводит ее на экран, затем «вычеркивает» из матрицы столбец с номером N и строку с номером M, выводит измененную матрицу на экран, вычисляет и выводит на экран значение ее определителя.

## Описание и пояснение к работе

Используется система сборки Autotools на Linux и CMake на Windows и Linux. Используется интерфейс командной строки. Для сборки, вывода справки клиента и сервера требуется:

*# linux, autotools, make, gcc*./configure  
make  
./build/Lab\_05\_Client -h  
./build/Lab\_05\_Server -h  
  
*# linux, autotools, make, clang*./configure CC=clang  
make  
./build/Lab\_05\_Client -h  
./build/Lab\_05\_Server -h  
  
*# linux, cmake, make, gcc*cd build  
cmake ..  
make  
./Lab\_05\_Client/Lab\_05\_Client -h  
./Lab\_05\_Server/Lab\_05\_Server -h  
  
*# linux, cmake, make, clang*cd build  
cmake -DCMAKE\_C\_COMPILER=clang ..  
make  
./Lab\_05\_Client/Lab\_05\_Client -h  
./Lab\_05\_Server/Lab\_05\_Server -h  
  
*# windows, cmake, msbuild, msvc*cd build  
cmake ..  
msbuild .\ALL\_BUILD.vcxproj  
.\Lab\_05\_Client\Debug\Lab\_05\_Client.exe -h  
.\Lab\_05\_Server\Debug\Lab\_05\_Server.exe -h  
  
*# windows, cmake, mingw32-make, mingw*cd build  
cmake -G **"MinGW Makefiles"** ..  
mingw32-make  
.\Lab\_05\_Client\Lab\_05\_Client.exe -h  
.\Lab\_05\_Server\Lab\_05\_Server.exe -h  
  
*# windows, cmake, mingw32-make, clang*cd build  
cmake -G **"MinGW Makefiles"** -DCMAKE\_C\_COMPILER=clang -DCMAKE\_CXX\_COMPILER=clang++ ..  
mingw32-make  
.\Lab\_05\_Client\Lab\_05\_Client.exe -h  
.\Lab\_05\_Server\Lab\_05\_Server.exe -h

# Исходные тексты программ (с комментариями в стиле системы doxygen)

Исходные тексты программ предоставлены в архиве.

# Содержимое скрипта configure

## configure

Предоставлен в архиве.

## сonfigure.ac

# -\*- Autoconf -\*-  
# Process this file with autoconf to produce a configure script.  
  
AC\_PREREQ([2.69])  
AC\_INIT([Lab\_05], [0.1], [misterptits@yandex.ru])  
AC\_CONFIG\_SRCDIR([Lab\_05\_Client/main.c])  
#AC\_CONFIG\_HEADERS([config.h])  
  
# Checks for programs.  
AC\_PROG\_CC  
  
# Checks for libraries.  
  
# Checks for header files.  
AC\_CHECK\_HEADERS([arpa/inet.h fcntl.h limits.h malloc.h netinet/in.h stddef.h stdint.h stdlib.h string.h sys/socket.h unistd.h])  
  
# Checks for typedefs, structures, and compiler characteristics.  
AC\_CHECK\_HEADER\_STDBOOL  
AC\_TYPE\_SIZE\_T  
AC\_TYPE\_UINT16\_T  
AC\_TYPE\_UINT32\_T  
  
# Checks for library functions.  
AC\_FUNC\_MALLOC  
AC\_FUNC\_REALLOC  
AC\_FUNC\_STRERROR\_R  
AC\_CHECK\_FUNCS([socket strchr])  
  
AM\_INIT\_AUTOMAKE  
AC\_PROG\_RANLIB  
  
AC\_CONFIG\_FILES([build/Makefile  
 Makefile])  
AC\_OUTPUT

## Makefile.am

SUBDIRS = build

## build/Makefile.am

AUTOMAKE\_OPTIONS = subdir-objects  
  
noinst\_LIBRARIES = libLab\_05\_Lib.a libLab\_05\_MatrixLib.a   
bin\_PROGRAMS = Lab\_05\_Client Lab\_05\_Server  
  
libLab\_05\_MatrixLib\_a\_SOURCES = \  
 ../Lab\_05\_MatrixLib/Matrix.c  
  
libLab\_05\_Lib\_a\_SOURCES = \  
 ../Lab\_05\_Lib/Input.c \  
 ../Lab\_05\_Lib/ParseInt.c \  
 ../Lab\_05\_Lib/ReturnCodes.c \  
 ../Lab\_05\_Lib/LastErrorMessage.c \  
 ../Lab\_05\_Lib/Request.c \  
 ../Lab\_05\_Lib/Socket.c  
libLab\_05\_Lib\_a\_CFLAGS = -I../Lab\_05\_MatrixLib   
libLab\_05\_Lib\_a\_LIBADD = libLab\_05\_MatrixLib.a  
  
Lab\_05\_Client\_SOURCES = \  
 ../Lab\_05\_Client/Args.c \  
 ../Lab\_05\_Client/Client.c \  
 ../Lab\_05\_Client/main.c  
Lab\_05\_Client\_CFLAGS = -I../Lab\_05\_MatrixLib -I../Lab\_05\_Lib  
Lab\_05\_Client\_LDADD = libLab\_05\_MatrixLib.a libLab\_05\_Lib.a  
  
Lab\_05\_Server\_SOURCES = \  
 ../Lab\_05\_Server/Args.c \  
 ../Lab\_05\_Server/Server.c \  
 ../Lab\_05\_Server/main.c  
Lab\_05\_Server\_CFLAGS = -I../Lab\_05\_MatrixLib -I../Lab\_05\_Lib  
Lab\_05\_Server\_LDADD = libLab\_05\_MatrixLib.a libLab\_05\_Lib.a

# Примеры работы программ в виде перехватов содержимого экрана

## Запуск