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](#_Toc25360273)

[1 Цель работы с постановкой задачи 4](#_Toc25360274)

[1.1 Цель работы 4](#_Toc25360275)

[1.2 Задача работы 4](#_Toc25360276)

[1.3 Описание и пояснение к работе 5](#_Toc25360277)

[2 Исходные тексты программ (с комментариями в стиле системы doxygen) 6](#_Toc25360278)

[3 Содержимое скрипта configure 6](#_Toc25360279)

[3.1 configure 6](#_Toc25360280)

[3.2 сonfigure.ac 6](#_Toc25360281)

[3.3 Makefile.am 7](#_Toc25360282)

[3.4 build/Makefile.am 7](#_Toc25360283)

[4 Примеры работы программ в виде перехватов содержимого экрана 8](#_Toc25360284)

[4.1 Запуск №1 (с помощью loopback, WSL Ubuntu 18.04, GCC) 8](#_Toc25360285)

[4.1.1. Сборка с помощью autotools 8](#_Toc25360286)

[4.1.2. Запуск клиента без запущенного сервера 9](#_Toc25360287)

[4.1.3. Запуск сервера 9](#_Toc25360288)

[4.1.4. Запуск клиента 10](#_Toc25360289)

[4.1.5. Результат работы сервера 10](#_Toc25360290)

[4.2 Запуск №2 (с помощью локальной сети по TCP: Клиент – Ubuntu 19.10, GCC; сервер – Android (Termux)) 11](#_Toc25360291)

[4.2.1. Сборка на сервере 12](#_Toc25360292)

[4.2.2. Запуск сервера 15](#_Toc25360293)

[4.2.3. Запуск клиента 15](#_Toc25360294)

[4.2.4. Результат работы сервера 16](#_Toc25360295)

[4.3 Запуск №3 (с помощью глобальной сети по TCP: Клиент – Windows, MSVC; удалённый сервер – Ubuntu 16.04, GCC) 16](#_Toc25360296)

[4.3.1. Сборка с помощью autotools на удалённом сервере 16](#_Toc25360297)

[4.3.2. Сборка на клиенте с помощью CMake и msbuild 18](#_Toc25360298)

[4.3.3. Запуск клиента без запущенного сервера 19](#_Toc25360299)

[4.3.4. Запуск сервера 19](#_Toc25360300)

[4.3.5. Запуск клиента 20](#_Toc25360301)

[4.3.6. Результат работы сервера 20](#_Toc25360302)

# 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 16.** Клиент отсылает серверу элементы двух квадратных матриц

одинакового размера. Его, а также сами элементы матриц должен вводить пользователь. Сервер принимает две квадратные матрицы, а затем выводит на экран сумму матриц, а также определитель суммарной матрицы.

По варианту задания требуется использование дейтаграммных сокетов. Ранее в программе была возможность опционально использовать потоковые сокеты, поэтому в примерах работы программы 4.2 и 4.3 используются потоковые сокеты.

## 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], [2.0], [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

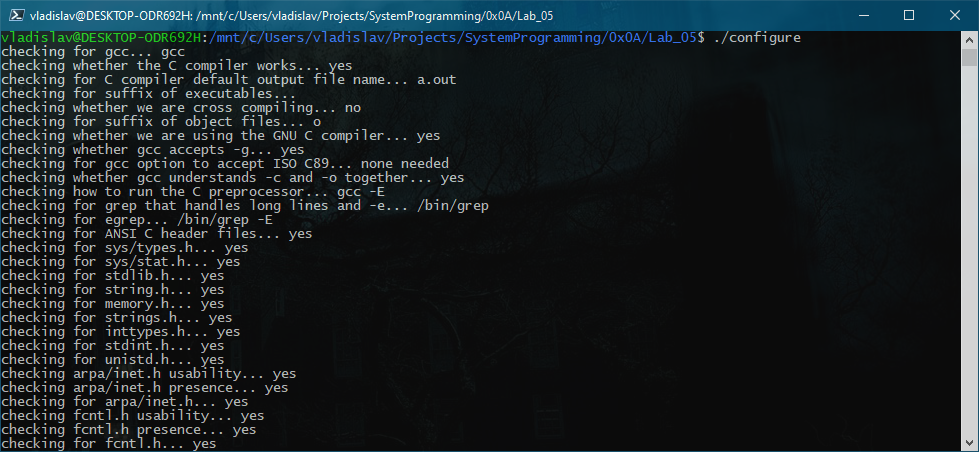
## build/Makefile.am

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\_MatrixLib\_a\_CFLAGS = -std=gnu99  
  
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 -std=gnu99  
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 -std=gnu99  
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 -std=gnu99  
Lab\_05\_Server\_LDADD = libLab\_05\_MatrixLib.a libLab\_05\_Lib.a

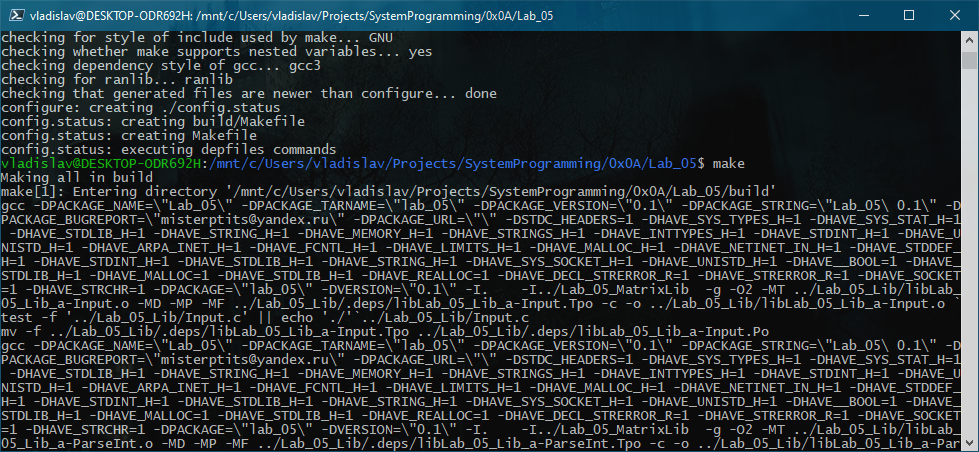
# Примеры работы программ в виде перехватов содержимого экрана

## Запуск №1 (с помощью loopback, WSL Ubuntu 18.04, GCC)

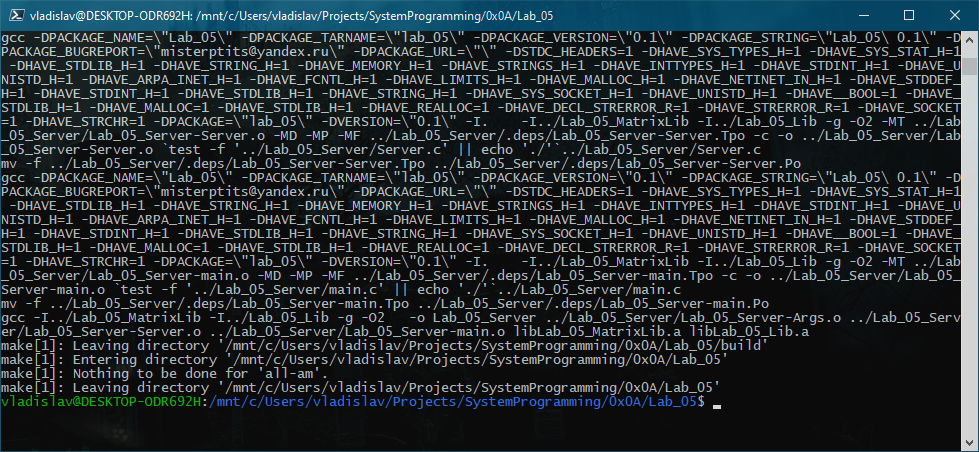
### Сборка с помощью autotools



1. Запуск ./configure

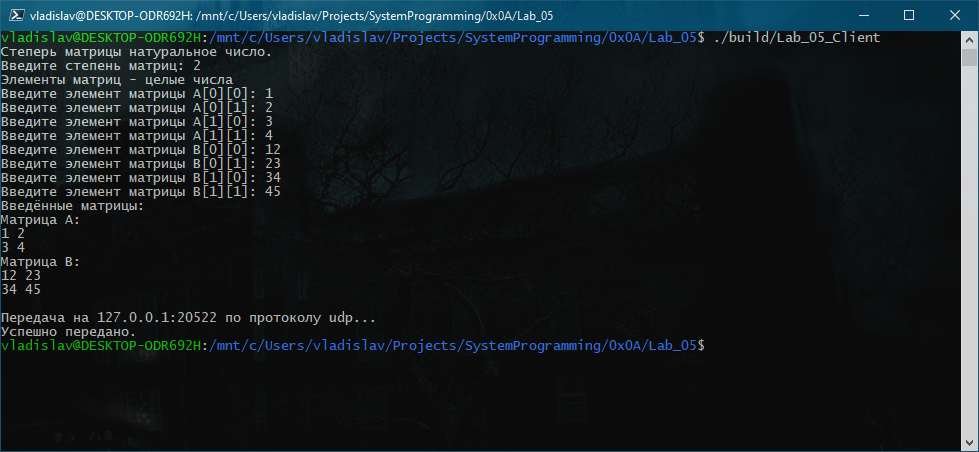


1. Конец работы ./configure, запуск make



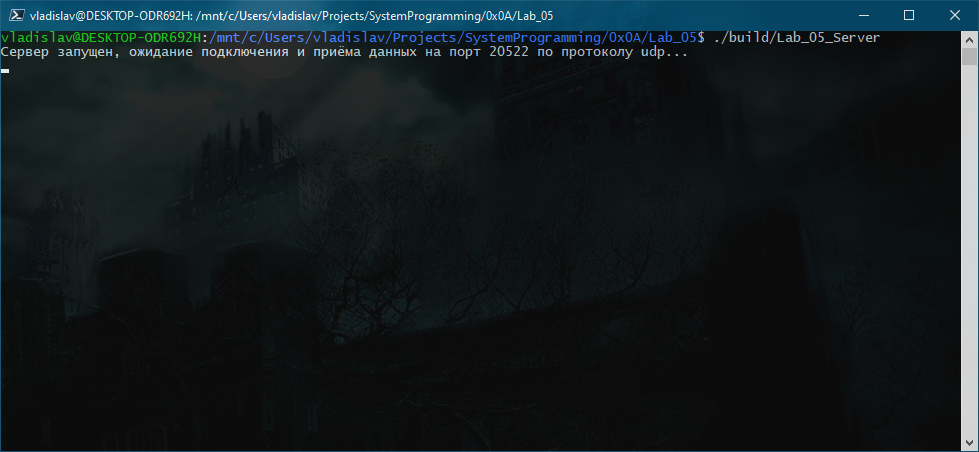
1. Результат работы make

### Запуск клиента без запущенного сервера



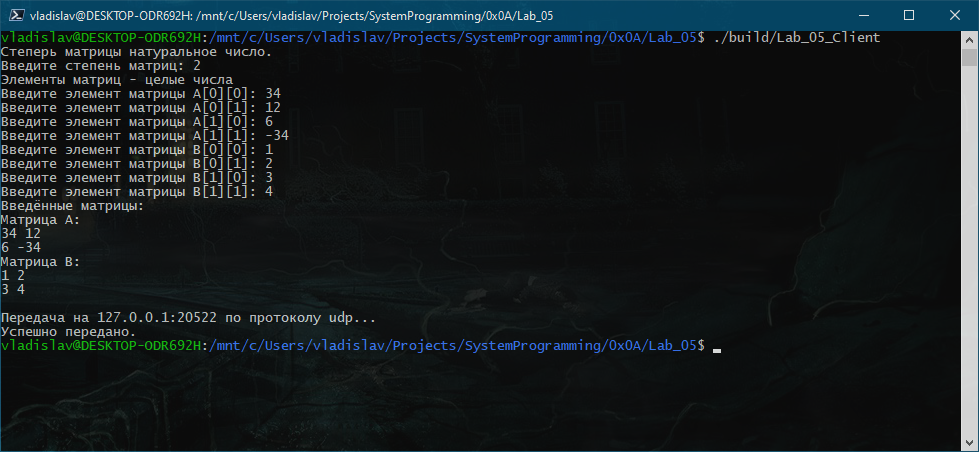
1. Матрицы передаются, но клиент не знает, что переданные данные потерялись, потому что используются дейтаграммные сокеты

### Запуск сервера



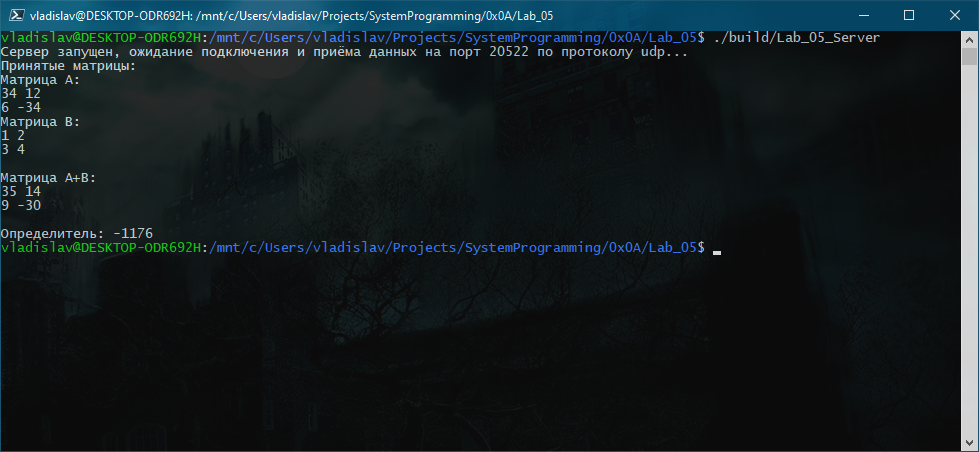
1. Сервер ожидает подключения от клиента

### Запуск клиента



1. Клиент считал данные, передал и успешно завершился

### Результат работы сервера



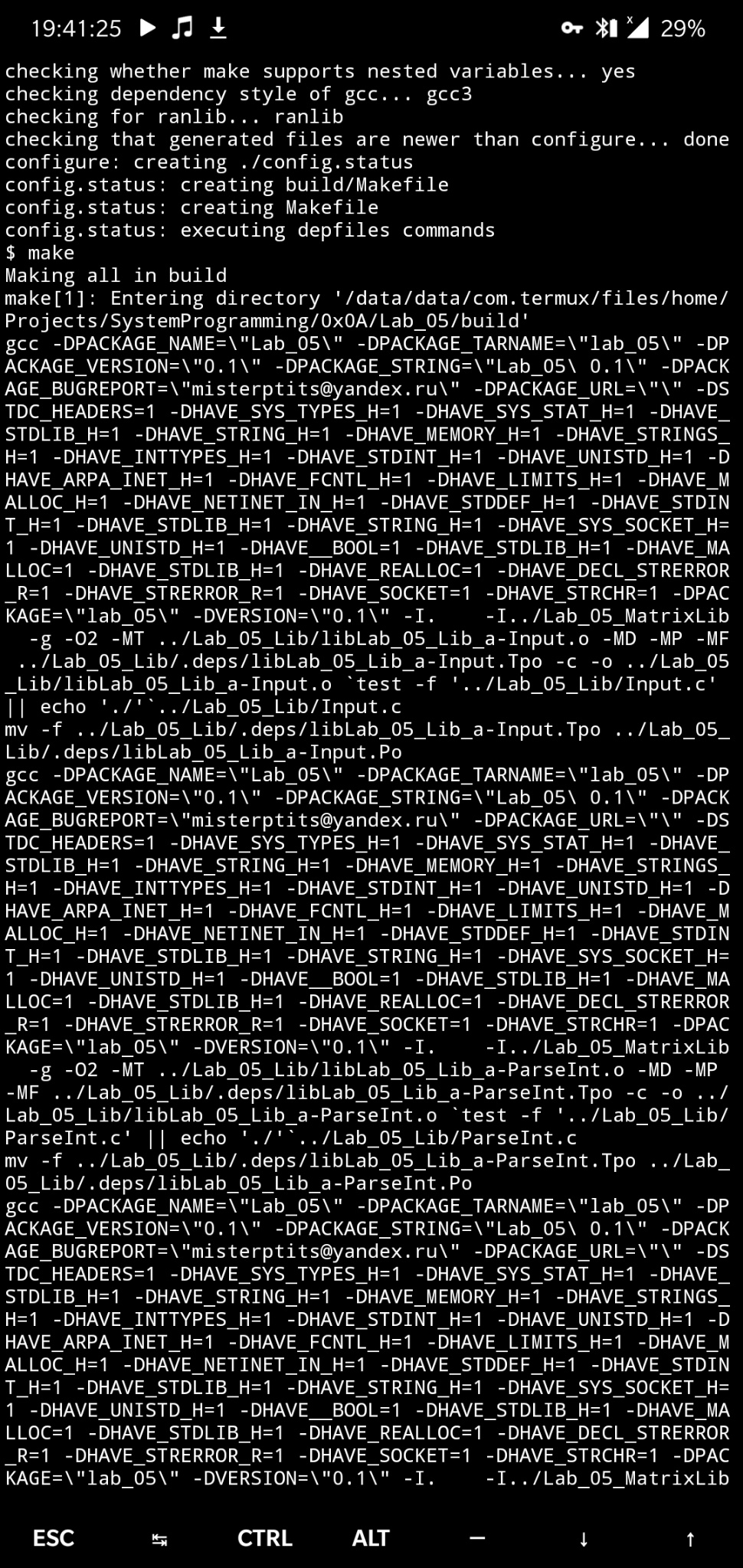
1. Сервер принял, вычислил, вывел результат и успешно завершился

## Запуск №2 (с помощью локальной сети по TCP: Клиент – Ubuntu 19.10, GCC; сервер – Android (Termux))

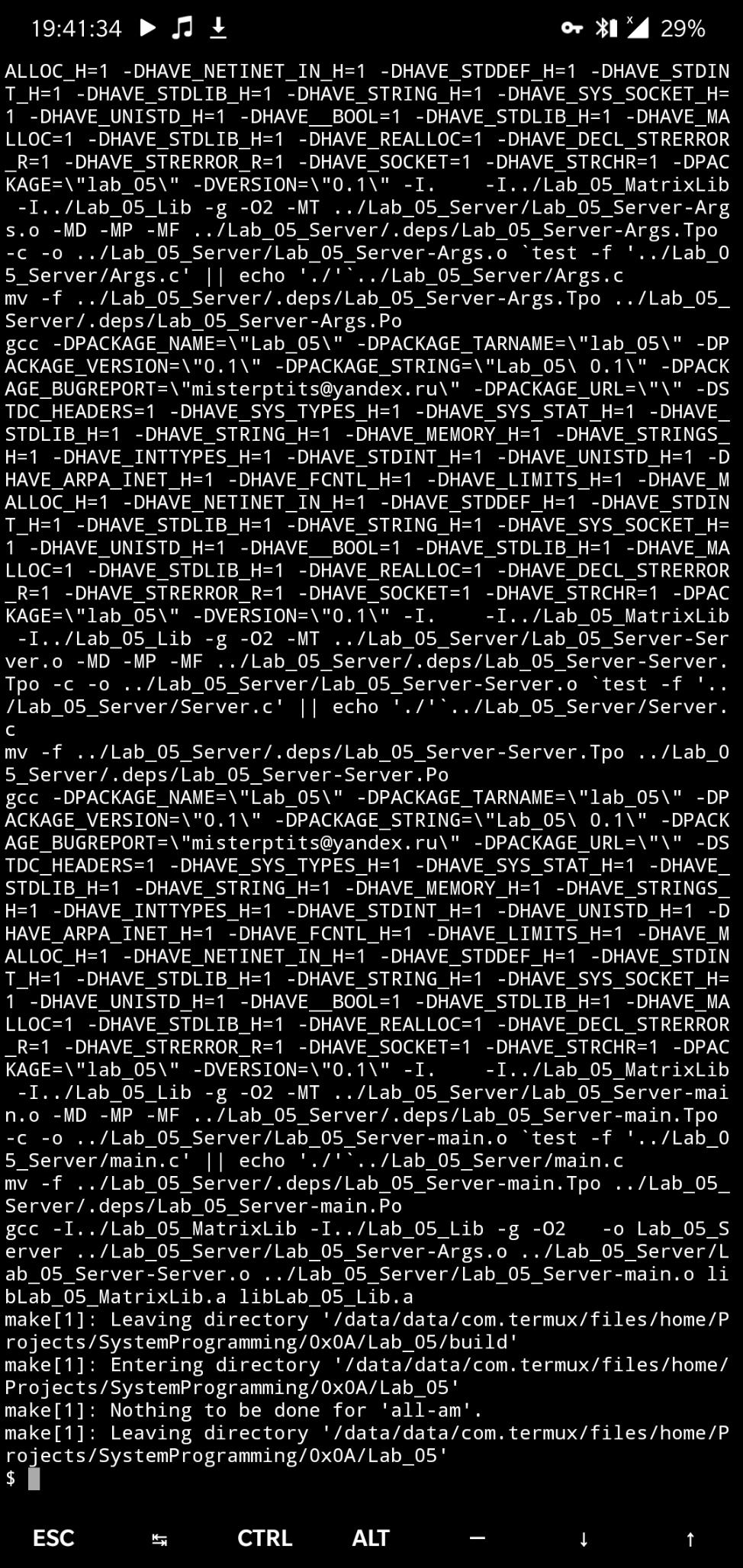
### Сборка на сервере



1. Запуск ifconfig для определения адреса сервера в локальной сети и запуск ./configure

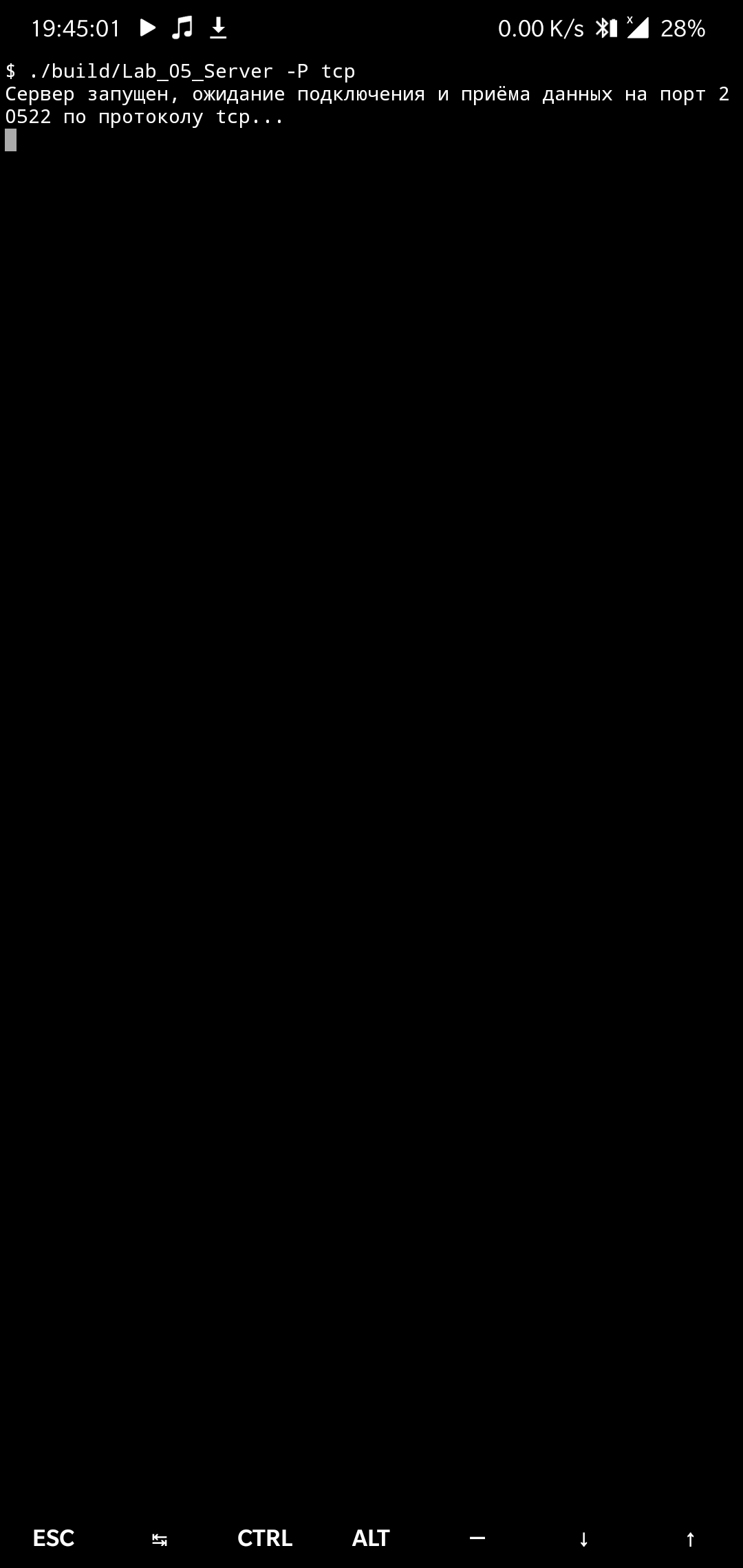


1. Результат работы ./configure и запуск make



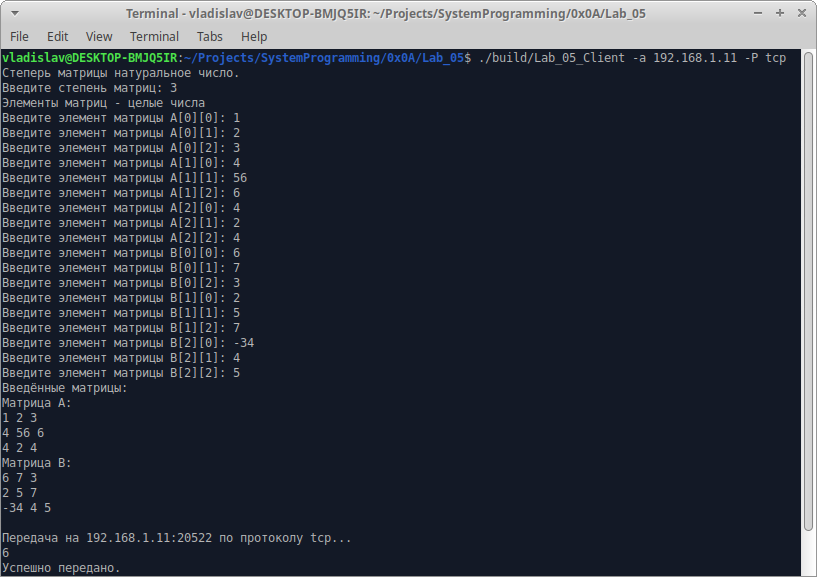
1. Результат работы make

### Запуск сервера



1. Запуск сервера

### Запуск клиента



1. Запуск клиента

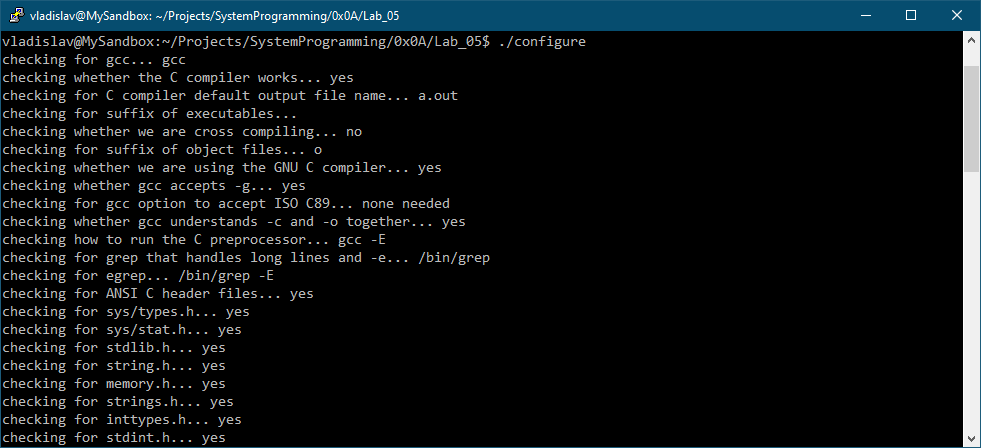
### Результат работы сервера



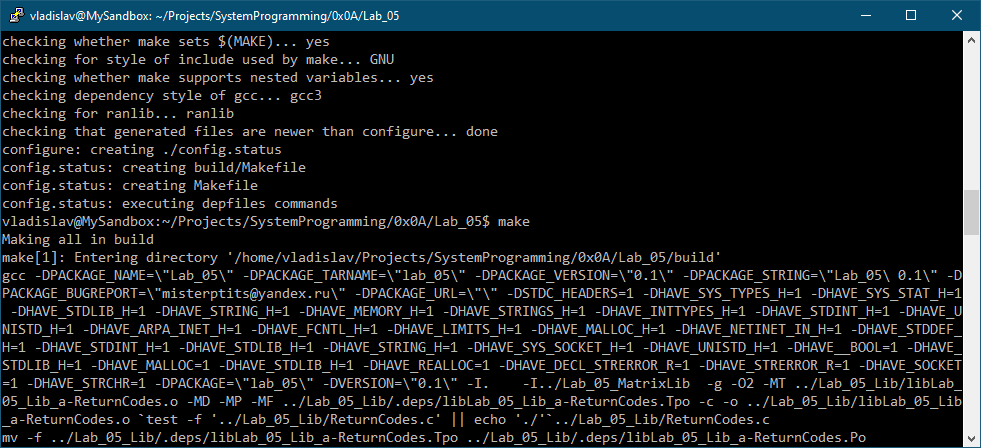
1. Результат работы сервера

## Запуск №3 (с помощью глобальной сети по TCP: Клиент – Windows, MSVC; удалённый сервер – Ubuntu 16.04, GCC)

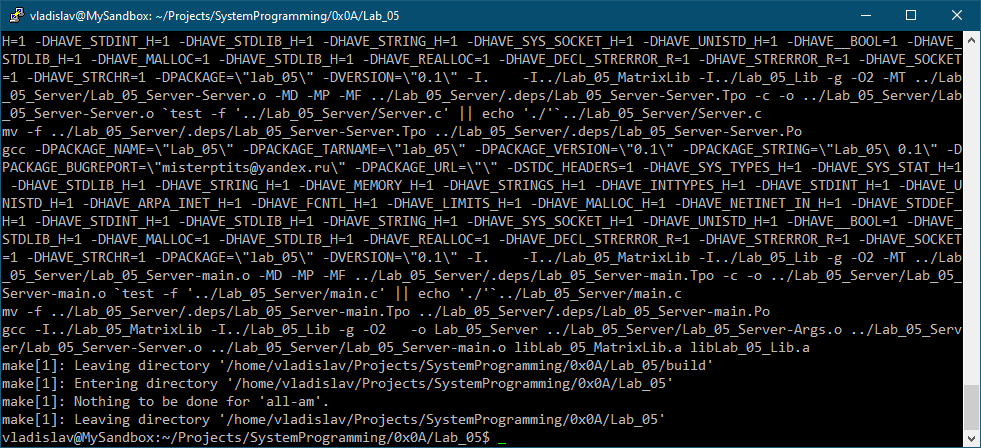
### Сборка с помощью autotools на удалённом сервере



1. Запуск ./configure

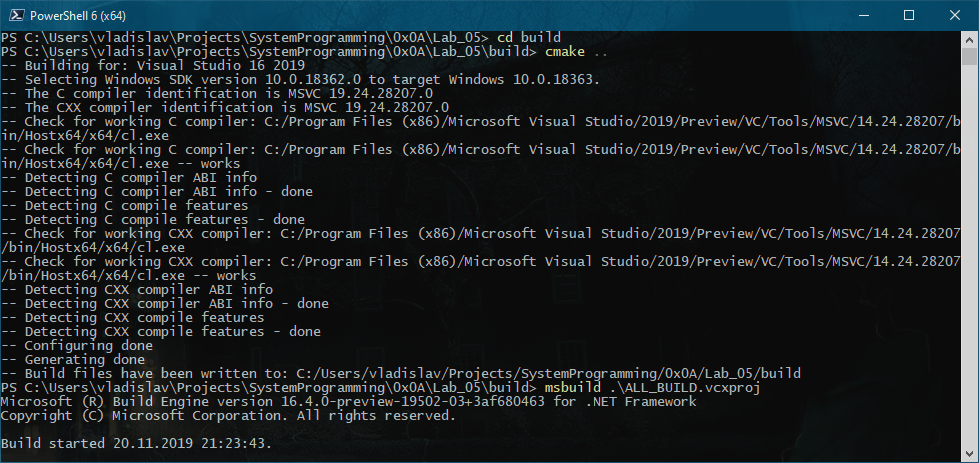


1. Запуск make



1. Результат работы make

### Сборка на клиенте с помощью CMake и msbuild

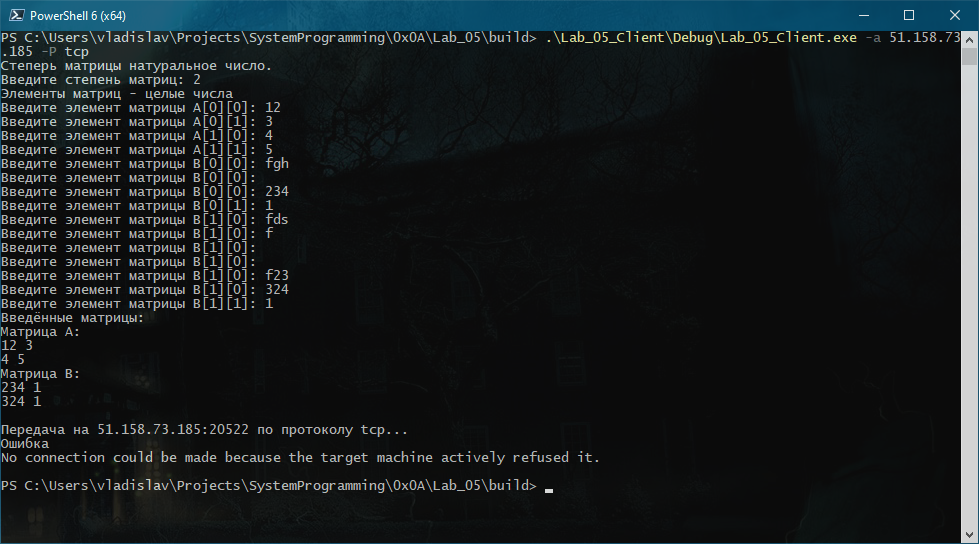


1. Запуск cmake и msbuild



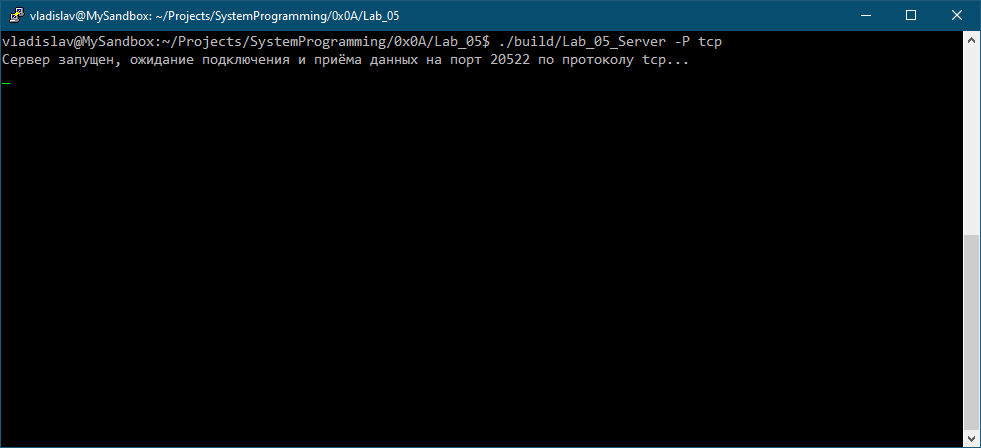
1. Конец работы msbuild

### Запуск клиента без запущенного сервера



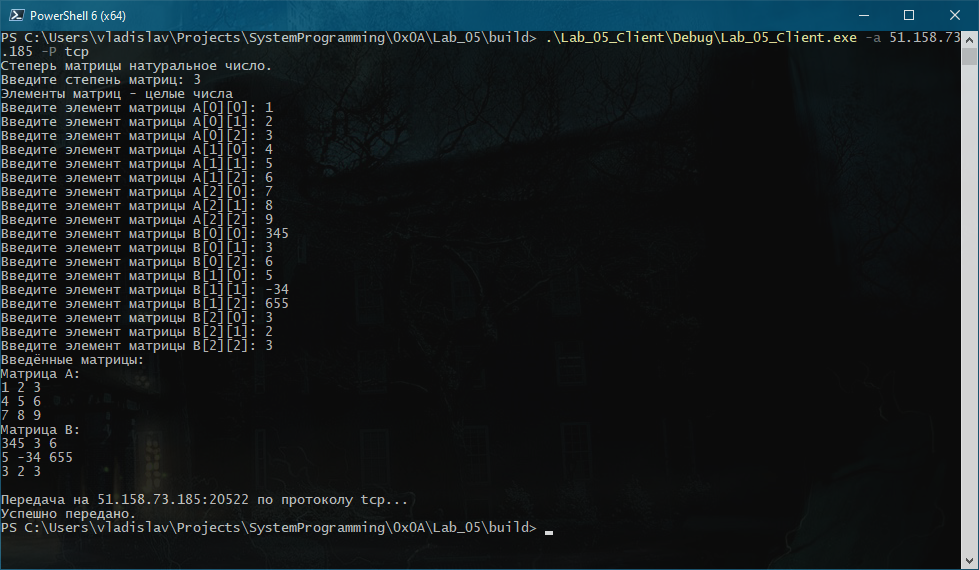
1. Ошибка если сервер не запущен

### Запуск сервера



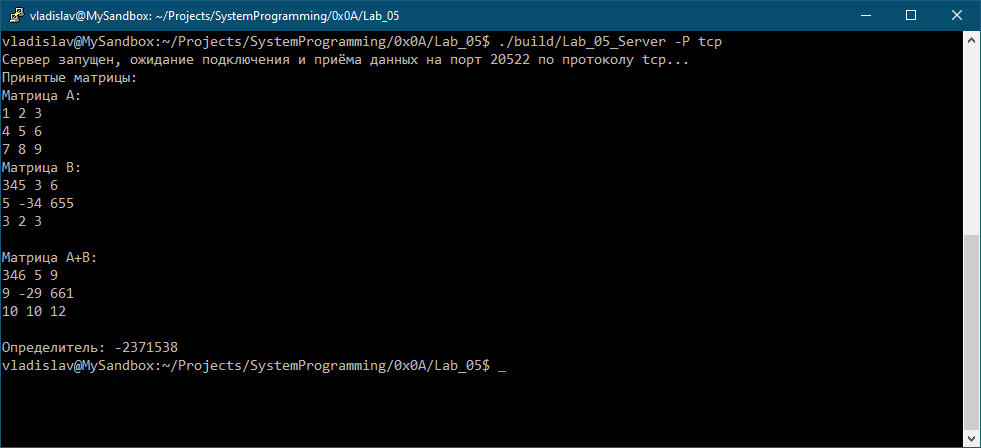
1. Сервер ожидает подключения от клиента

### Запуск клиента



1. Клиент считал данные, передал и успешно завершился

### Результат работы сервера



1. Сервер принял, вычислил, вывел результат и успешно завершился