Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Факультет Безопасности (ФБ)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

КОМБИНИРОВАННЫЕ ПРОГРАММЫ. СВЯЗЫВАНИЕ РАЗНОЯЗЫКОВЫХ МОДУЛЕЙ

Отчёт по практической работе № 3 по дисциплине  
«Системное программирование»

Выполнил

Студент группы 726

\_\_\_\_\_\_\_\_\_\_ Качалов З.А.

\_\_\_\_\_\_\_\_\_\_

Принял

к.т.н., доцент кафедры КИБЭВС

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ Мещеряков Я.Е.

оценка

\_\_\_\_\_\_\_\_\_\_

дата

**1 Введение**

Цель работы: познакомиться с основными способами передачи параметров подпрограмм, особенностями передачи управления между модулями, научиться писать комбинированные программы, в которых модули Ассемблера вызываются из модулей, написанных на высокоуровневых языках программирования.

В ходе работы необходимо ознакомиться с теоретическим материалом по связыванию разных языковых модулей, разработать и написать модули программы, отладить программы, оформить отчет и защитить у преподавателя.

Задание:

1. Изучить теоретические сведения, самостоятельно изучить способы

связывания разных языковых модулей.

1. Разработать и написать программу на языке C++ с ассемблерной вставкой. Код программы на языке C++ и Assembler должен решать задачу согласно варианту 14. Задание 14 варианта:

Напишите программу, в которой создается двумерный числовой массив и заполняется случайными числами. Для каждой строки (или столбца) двумерного массива определяется наибольший (или наименьший) элемент, и из таких элементов создается одномерный числовой массив.

3. На moodle загрузить архив, содержащий отчет и все файлы проекта.

4. Написать отчет и защитить у преподавателя.

**2 Ход работы**

В ходе работы была написана программа на языке С++ с ассемблерной вставкой в соответствии с вариантом 14 (Приложение А). Код докер-файла представлен в Приложении B.

Результат компиляции, а также вывод результата в консоль представлен на рисунке 2.1.

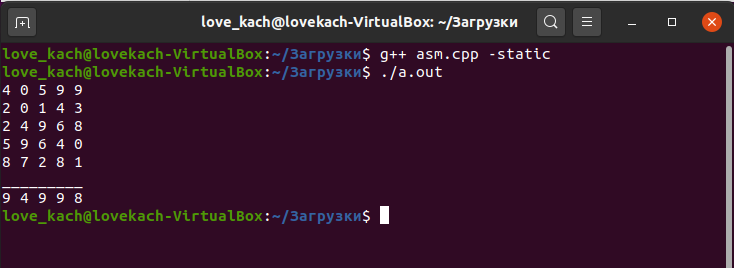


Рисунок 2.1 – Сборка и запуск программы

**3 Заключение**

В ходе работы были изучены основные способы передачи параметров подпрограмм, особенности передачи управления между модулями. Также была написана комбинированная программа.

Ссылка на репозиторий - <https://github.com/7261kza/SysProg_Lab3>.

Приложение А

Код основной программы

#include <iostream>#include <random>#include <ctime>

using namespace std;

extern "C" { const int size = 5; int tempSize = size; int newArray[size]; int array[size][size];};

int main(){ srand(time(0));

for (int i = 0; i < size; ++i) { for (int j = 0; j < size; ++j) { array[i][j] = 0 + rand() % 10; } } \_\_asm (

Листинг А.1 – Основной код программы (начало)

R"( .intel\_syntax noprefix ;//{ mov esi, 0 ;// этот регистр будет хранить адрес элементов одномерного массива mov ecx, 0 ;// этот регистр будет указывать адрес элементов двумерного массива mov dl, 0 ;// этот регистр будет считать количество пройденных строк

StartNewLine: mov dh, 0 ;// этот регистр считает количество пройденных элементов массива mov ebx, 0 ;// сюда будем класть макс элемент

Main: mov eax, array[ecx] ;// здесь будет лежать текущий (проверяемый) элемент cmp eax, ebx jg MaxElement

jmp Incrementing

MaxElement: mov ebx, eax

Incrementing: add ecx, 4

Листинг А.1 – Основной код программы (продолжение)

add dh, 1

CheckEndLine: cmp dh, BYTE PTR tempSize jl Main

mov newArray[esi], ebx add esi, 4

CheckEndArray: add dl, 1 cmp dl, BYTE PTR tempSize jl StartNewLine ;//} .att\_syntax )" );

for (int i = 0; i < size; i++) { for (int j = 0; j < size; j++) { cout << array[i][j] << " "; } cout << endl; } cout << "\_\_\_\_\_\_\_\_\_" << endl;

Листинг А.1 – Основной код программы (продолжение)

for (int i = 0; i < size; i++) { cout << newArray[i] << " "; } cout << endl;

return 0;}

Листинг А.1 – Основной код программы (окончание)

Приложение В

Код Dockerfile

FROM Ubuntu: 20200122

WORKDIR /home

RUN apt update

RUN apt-get install gcc build-essential gcc-multilib

COPY ./asm.cpp/home

Листинг B.1 – Содержимое Docker-файла