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

\_\_\_\_\_\_\_\_\_\_ М.И. Васильева

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

Принял

к.т.н., доцент кафедры КИБЭВС

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

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, а также модуль на Assembler и вызвать процедуру из основного кода программы. Код программы на языке C и Assembler должен решать задачу согласно варианту 8. Задание 8 варианта:

Напишите программу, в которой создается одномерный числовой массив. После заполнения значениями (например, случайными числами) массива в нем нужно выполнить циклическую перестановку элементов. Количество позиций для циклической перестановки вводится пользователем с клавиатуры.

3. На moodle загрузить архив, содержащий: Отчет, все файлы проекта.

4. Написать отчет и защитить у преподавателя.

2 Ход работы

В ходе работы была написана программа на языке С (Приложение А) и на Ассемблере (Приложение Б) в соответствии с вариантом 8.

Результат компиляции и линковки программы для получения одного исполняемого файла, а также вывод результата в консоль (рисунок 1).

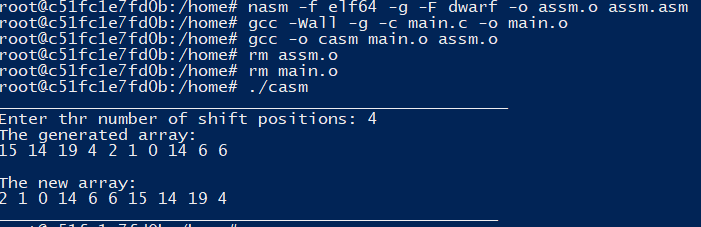


Рисунок 1 – Сборка и запуск программы

3 Заключение

В ходе работы были изучены основные способы передачи параметров подпрограмм, особенности передачи управления между модулями. Также была написана комбинированная программа, в которой модуль Ассемблера вызывается из модуля, написанного на языке С.

Ссылка на репозиторий - <https://github.com/mariaigorevna> . Название – Lab3\_moduls.

Приложение А

Код модуля программы С

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

extern void \_reverse**(**int **\***a**,** int **\***b**);**

int main**()**

**{**

int i**,** n**=**10**,**m**;**

int mas**[**n**];**

printf**(**"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"**);**

printf**(**"Enter the number of shift positions: "**);**

scanf**(**"%d"**,** **&**m**);**

srand**(**time**(**0**));**

printf**(**"The generated array:\n"**);**

**for** **(**i **=** 0**;** i **<** n**;** i**++)** **{**

mas**[**i**]** **=** rand**()** **%** 20**;**

printf**(**"%d "**,** mas**[**i**]);**

**}**

putchar**(**'\n'**);**

\_reverse**(**mas**,**mas**+**m**-**1**);**

\_reverse**(&**mas**[**m**],**mas**+**n**-**1**);**

\_reverse**(**mas**,**mas**+**n**-**1**);**

printf**(**"The new array:\n"**);**

**for(**i **=** 0**;** i **<** 10**;** i**++)**

printf**(**"%d "**,** mas**[**i**]);**

putchar**(**'\n'**);**

printf**(**"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"**);**

putchar**(**'\n'**);**

**return** 0**;**

**}**

Листинг А.1 – Основной код программы

Приложение Б

Код модуля программы ассемблер

section **.text**

global \_reverse

\_reverse**:**

**push** **rbp**

**push** **rsp**

**mov** **rbp,** **rsp**

**sub** **rsp,** 16

**mov** **QWORD** **[rbp-**8**],** **rdi**

**mov** **QWORD** **[rbp-**16**],** **rsi**

**mov** **rax,** **QWORD** **[rbp-**8**]**

**cmp** **rax,** **QWORD** **[rbp-**16**]**

**jnb** .L3

**mov** **rax,** **QWORD** **[rbp-**8**]**

**mov** **edx,** **DWORD** **[rax]**

**mov** **rax,** **QWORD** **[rbp-**16**]**

**mov** **eax,** **DWORD** **[rax]**

**add** **edx,** **eax**

**mov** **rax,** **QWORD** **[rbp-**8**]**

**mov** **DWORD** **[rax],** **edx**

**mov** **rax,** **QWORD** **[rbp-**8**]**

**mov** **edx,** **DWORD** **[rax]**

**mov** **rax,** **QWORD** **[rbp-**16**]**

**mov** **eax,** **DWORD** **[rax]**

**sub** **edx,** **eax**

**mov** **rax,** **QWORD** **[rbp-**16**]**

**mov** **DWORD** **[rax],** **edx**

**mov** **rax,** **QWORD** **[rbp-**8**]**

**mov** **edx,** **DWORD** **[rax]**

**mov** **rax,** **QWORD** **[rbp-**16**]**

**mov** **eax,** **DWORD** **[rax]**

**sub** **edx,** **eax**

**mov** **rax,** **QWORD** **[rbp-**8**]**

**mov** **DWORD** **[rax],** **edx**

**mov** **rax,** **QWORD** **[rbp-**16**]**

**lea** **rdx,** **[rax-**4**]**

**mov** **rax,** **QWORD** **[rbp-**8**]**

**add** **rax,** 4

**mov** **rsi,** **rdx**

Листинг Б.1 – Код модуля программы ассемблер (начало)

**mov** **rdi,** **rax**

**call** \_reverse

.L3**:**

**mov** **rsp,** **rbp**

POP **rsp**

POP **rbp**

ret

Листинг Б.2 – Код модуля программы ассемблер (конец)

Приложение В

Код Dockerfile

FROM debian

COPY ./assm.asm /home/assm.asm

COPY ./main.c /home/main.c

RUN apt-get update && apt-get -y install gcc

RUN apt-get install -y nano

RUN apt-get install -y gdb

RUN apt-get install -y nasm

ENTRYPOINT /bin/bash

Листинг В.1 – Код Dockerfile