Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №8

на тему:

**«ВВЕДЕНИЕ В АРХИТЕКТУРУ ARM»**

БГУИР 6-05-0612-02 85

|  |
| --- |
| Выполнил студент группы 353503  МАРТИНОВИЧ Андрей Александрович |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 10.** Инвертировать регистры букв в введенной строке.  
 **Задание 2. Вариант 10.** Даны вещественные коэффициенты квадратного уравнения 𝑎𝑥2+𝑏𝑥+𝑐. Гарантируется, что детерминант этого квадратного уравнения >= 0. Найти корни данного квадратного уравнения и вывести их в экспоненциальной форме (разрешено пользоваться вызовом фукнции printf языка С).

**Задание 3. Вариант 10.** Пользователем вводится путь к файлу. Далее, без использования функции write, перенести содержимое файла в другой путь, введенный пользователем. Открыть файл по новому пути, и вывести в стандартный вывод в его содержимое (содержимое файла в формате UTF-8).

# 2 Выполнение работы

**Задание 1**

Листинг кода 1 – Исходный код

.global \_start

.section .data

input\_string:

.asciz "Enter string to invert:\n"

.section .bss

.lcomm buffer, 1024

.section .text

\_start:

LDR r0, =input\_string

BL print\_string

MOV r0, #0

LDR r1, =buffer

MOV r2, #1024

MOV r7, #3

SVC 0

LDR r0, =buffer

BL invert\_case

LDR r0, =buffer

BL print\_string

MOV r7, #1

SVC 0

invert\_case:

LDRB r1, [r0]

CMP r1, #0

BEQ end\_invert

CMP r1, #'A'

BLT next\_char

CMP r1, #'Z'

BGT next\_char

ADD r1, r1, #'a' - 'A'

B store\_char

next\_char:

CMP r1, #'a'

BLT store\_char

CMP r1, #'z'

BGT store\_char

SUB r1, r1, #'a' - 'A'

store\_char:

STRB r1, [r0]

ADD r0, r0, #1

B invert\_case

end\_invert:

BX lr

print\_string:

MOV r1, r0

MOV r2, #1024

MOV r7, #4

MOV r0, #1

SVC 0

BX lr

**Задание 2**

Листинг кода 2 – Исходный код

.global main

main:

STR LR, [SP, #-16]!

LDR X0, =a

LDR X1, =b

LDR X2, =c

LDR D0, [X0]

LDR D1, [X1]

LDR D2, [X2]

// D = b^2 - 4ac

FMUL D3, D1, D1 // D3 = b^2

FMOV D4, 4.0 // D4 = 4

FMUL D4, D4, D0 // D4 = 4a

FMUL D4, D4, D2 // D4 = 4ac

FSUB D5, D3, D4 // D5 = b^2 - 4ac

// x1 и x2

FSQRT D5, D5 // D5 = sqrt(D)

FNEG D6, D1 // D6 = -b

FSUB D7, D6, D5 // D7 = -b - sqrt(D)

FADD D8, D6, D5 // D8 = -b + sqrt(D)

FDIV D7, D7, D0 // x1 = (-b - sqrt(D)) / a

FDIV D8, D8, D0 // x2 = (-b + sqrt(D)) / a

FMOV D9, 2.0 // D9 = 2

FDIV D0, D7, D9 // x1 = x1 / 2

LDR X0, =fmt

BL printf

FDIV D0, D8, D9 // x2 = x2 / 2

LDR X0, =fmt

BL printf

MOV X0, #0

BL exit

LDR LR, [SP], #16

RET

.data

a: .double 1.0

b: .double -3.0

c: .double 2.0

fmt: .asciz "root = %e\n"

**Задание 3**

Листинг кода 3 – Исходный код

.include "macros.s"

.global main

main:

print enterNameMesage 16

mov x0, #0

ldr x1, =filename1

mov x2, #256

mov x8, #63

svc 0

ldr x0, =filename1

mov x1, #256

replace\_newline1:

ldrb w2, [x0], #1

cmp w2, #10

beq replace\_zero1

cmp w2, #0

beq end\_replace1

b replace\_newline1

replace\_zero1:

strb wzr, [x0, #-1]

b replace\_newline1

end\_replace1:

print enterNameMesage 16

mov x0, #0

ldr x1, =filename2

mov x2, #256

mov x8, #63

svc 0

ldr x0, =filename2

mov x1, #256

replace\_newline2:

ldrb w2, [x0], #1

cmp w2, #10

beq replace\_zero2

cmp w2, #0

beq end\_replace2

b replace\_newline2

replace\_zero2:

strb wzr, [x0, #-1]

b replace\_newline2

end\_replace2:

openFile filename1, S\_RDWR

MOV X11, X0

CMP X0, 0

openFile filename2, S\_RDWR

MOV X12, X0

CMP X0, 0

copy\_file:

MOV X0, X11

MOV X1, 0

MOV X2, X12

MOV X3, 0

MOV X4, -1

MOV X5, 0

MOV X8, #285

SVC 0

CMP X0, 0

BGE no\_error

error:

mov x1, x0

ldr x0, =err\_msg

bl printf

exit 0

no\_error:

print successMesage 16

close X11

close X12

openFile filename2, O\_RDONLY

MOV X11, X0

B.PL read

print errMessage 21

B finish

read:

readFile X11, output, 255

close X11

close X12

print output 255

finish:

exit 0

.data

err\_msg: .asciz "Error code: %d\n"

errMessage: .asciz "Fail\n"

successMesage: .asciz "Copy success\n"

enterNameMesage: .asciz "Enter file name:\n"

output: .fill 256, 1, 0

.bss

filename1: .skip 256

filename2: .skip 256

)

# Вывод

В ходе выполнения лабораторной работы были получены навыки работы с архитектурой Arm.