МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Вычислительная техника»

Дисциплина «Организация ЭВМ и систем»

**Лабораторная работа №6**

**««Программирование задач средней сложности на языке ассемблера под архитектуру Intel»**

**Вариант 12**

Выполнил:

студент группы ИВТАПбд-31

Кондратьев П.С.

Проверила:

Лылова А. В.

Ульяновск, 2018

**Техническое задание**

* Программный код, полученный в ходе выполнения лабораторной работы №5, изучить и перевести в программный код языка ассемблер для архитектуры ARM, учитывая синтаксис, структуру программы и команды для данной архитектуры.

**Исходный программный код на языке Assembler для архитектуры Intel**

.data

mes db 0ah,0dh,'arr - ','$'

mas db 77 dup (?)

i db 0

.code

main:

mov ax,@data

mov ds,ax

xor ax,ax

mov cx,3

mov si,0

push cx

go:

mov bh,i

mov mas[si],bh

inc i

inc si

loop go

pop cx

mov si,2

mov ah,09h

lea dx,mes

int 21h

add cx,1

cmp si, cx

jc show

show:

mov dl,mas[si]

cmp dl,0

je next

mov ah,02h

call print

mov dl, ' '

int 21h

mov ax,si

mul ax

mov di,ax

cmp di, cx

jc A

A:

mov mas[di],0

add di,si

cmp di, cx

jc A

next:

inc si

cmp si, cx

jc show

jmp exit

print:

push cx

mov ax, dx

xor cx, cx

mov bx, 10

oi2:

xor dx,dx

div bx

push dx

inc cx

test ax, ax

jnz oi2

mov ah, 02h

oi3:

pop dx

add dl, '0'

int 21h

loop oi3

pop cx

ret

exit:

mov ax,4c00h

int 21h

end main

**Программный код на языке Assembler для архитектуры ARM**

.data

mes: 'arr - ','$'

i: 0

mas: .byte

eoa:

start:

MOV R1,@data

MOV ds,R1

MOV R1,0

MOV R0,77

MOV R4,0

PUSH {R0}

go: CMP cx, R6

MOV R6,i

MOV mas[R4],R6

ADD i,i,1

ADD R4,R4,1

BEQ go

POP {R0}

MOV R4,2

WRITEWAIT LDR R3,mes

TST R3,#8

BEQ WRITEWAIT

SWI 0

ADD R0,R0,1

cmp R4, R0

BVS show

show:

MOV R5,mas[R4]

cmp R5,#0

BEQ next

MOV ah,02h

B print

getout:

MOV R5, ' '

SWI 0

MOV R1,R4

MUL R1,R1,R1

MOV di,R1

cmp di,R0

BVS

BVS A

A:

MOV mas[di],0

ADD di,di,R4

cmp di,R0

BVS A

next:

ADD R4,R4,1

cmp R4,R0

BVS show

B exit

print:

PUSH {R0}

MOV R1, R3

MOV R0,0

MOV R2,10

oi2:

MOV R3,0

DivMod R0,R3,R2

PUSH {R3}

ADD R0,R0,1

TEQ R1, R1

B oi2

WRITEWAIT LDR R5

TST R5,#8

BEQ WRITEWAIT

oi3:

POP {R3}

ADD R5,R5, '0'

SWI 0

BEQ oi3

POP {R0}

B getout

exit:

B exit

**Вывод:**

Архитектура Intel основана на архитктуре CISC, в то время как архитектура ARM основана на архитектуре RISC. В целом, есть множество сходных команд, таких как: mov, cmp и т.д. Однако методы адресации в командах различаются также различаются условия переходов. В целом, можно сказать, что архитектура ARM гораздо проще читать, нежели архитектура Intel.