

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Мошаров Денис Максимович

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Порядок выполнения лабораторной работы	6
3.2	Ответы на вопросы	12
3.3	Задание для самостоятельной работы	12
4	Выводы	16

Список иллюстраций

3.1	Создаём каталог	6
3.2	Заполняем программу	6
3.3	Результат	7
3.4	Подмена	7
3.5	Результат	7
3.6	Заполняем	8
3.7	Результат	8
3.8	Подмена	8
3.9	Результат	8
3.10	Меняем	9
3.11	Результат	9
3.12	Пишем программу	9
3.13	Результат	10
3.14	Меняем выражение	10
3.15	Результат	10
3.16	Пишем программу	11
3.17	Результат	11

1 Цель работы

Научиться писать и анализировать ассемблерный код с арифметическими операциями и понять синтаксис. Работа поможет развить навыки низкоуровневого программирования и понимания работы процессора.

2 Задание

Написать несколько программ для вычислений.

3 Выполнение лабораторной работы

3.1 Порядок выполнения лабораторной работы

Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm

```
dmmosharov@dmmosharov:~$ mkdir ~/work/arch-pc/lab06
dmmosharov@dmmosharov:~$ cd ~/work/arch-pc/lab06
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ touch lab6-1.asm
```

Рис. 3.1: Создаём каталог

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax

```
; ---- Вычисление выражения
mov eax,4      ; EAX=4
mov ebx,6      ; EBX=6
mul ebx        ; EAX=EAX*EBX
add eax,2      ; EAX=EAX+2
xor edx,edx    ; обнуляем EDX для корректной работы div
mov ebx,5      ; EBX=5
div ebx        ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax    ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
```

Рис. 3.2: Заполняем программу

```
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./lab6-1
j
```

Рис. 3.3: Результат

Изменим текст программы и вместо символов, запишем в регистры числа.

```
mov eax,6
mov ebx,4
```

Рис. 3.4: Подмена

```
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./lab6-1
dmmosharov@dmmosharov:~/work/arch-pc/lab06$
```

Рис. 3.5: Результат

Преобразуем текст программы из Листинга 6.1 с использованием этих функций.

```

; ---- Вычисление выражения
mov eax,4      ; EAX=4
mov ebx,6      ; EBX=6
mul ebx        ; EAX=EAX*EBX
add eax,2      ; EAX=EAX+2
xor edx,edx    ; обнуляем EDX для корректной работы div
mov ebx,5      ; EBX=5
div ebx        ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax    ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран

```

Рис. 3.6: Заполняем

```

dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./lab6-2
106

```

Рис. 3.7: Результат

Изменим символы на числа

```

mov eax,6
mov ebx,4

```

Рис. 3.8: Подмена

```

dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./lab6-2
10

```

Рис. 3.9: Результат

Замените функцию `iprintLF` на `iprint`. Создайте исполняемый файл и запустите его.

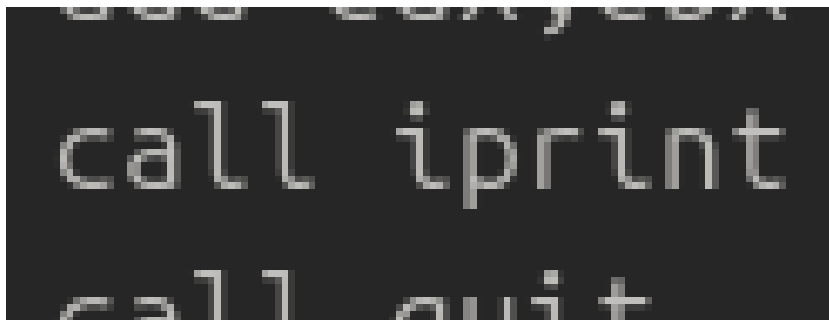


Рис. 3.10: Меняем

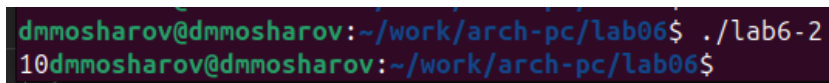


Рис. 3.11: Результат

В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $f(x) = (5 * 2 + 3) / 3$

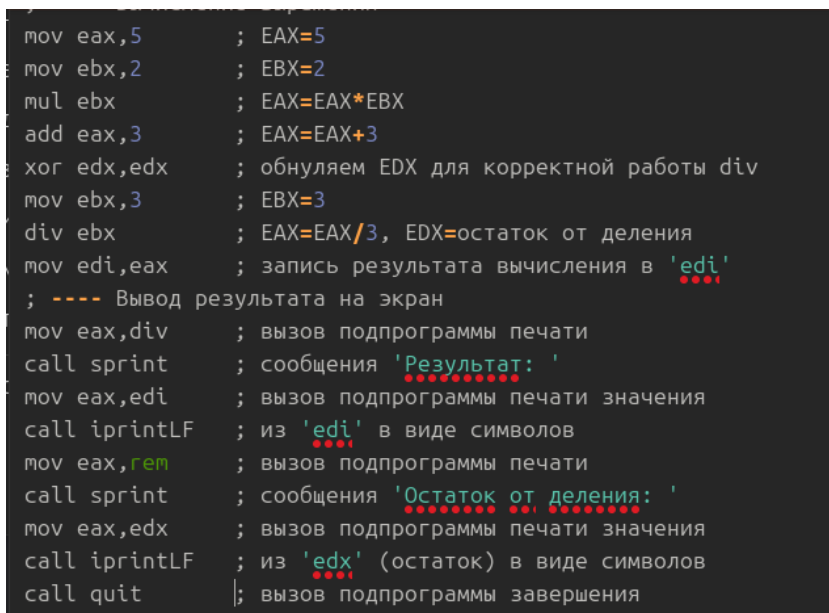


Рис. 3.12: Пишем программу

```
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.13: Результат

Измените текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$. Создайте исполняемый файл и проверьте его работу.

```
; ---- Вычисление выражения
mov eax,4      ; EAX=4
mov ebx,6      ; EBX=6
mul ebx        ; EAX=EAX*EBX
add eax,2      ; EAX=EAX+2
xor edx,edx    ; обнуляем EDX для корректной работы div
mov ebx,5      ; EBX=5
div ebx        ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax    ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
```

Рис. 3.14: Меняем выражение

```
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 3.15: Результат

Рассмотрим программу вычисления варианта задания по номеру студенческого билета

```

; Программа вычисления варианта
;-----
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintLF
call quit

```

Рис. 3.16: Пишем программу

```

dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1114425
Ваш вариант: 6

```

Рис. 3.17: Результат

3.2 Ответы на вопросы

1. Строка “moveax,rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’.
 2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре есх, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре еdx. Затем вызывается процедура sread, которая выполняет чтение строки.
 3. Инструкция “call atoi” используется для преобразования строки в целое число. Она принимает адрес строки в регистре еах и возвращает полученное число в регистре еах. Строка “xoredx,edx” обнуляет регистр. edx перед выполнением деления. Строка “movebx,20” загружает значение 20 в регистр ebx. Строка “divebx” выполняет деление регистра еах на значение регистра ebx с сохранением частного в регистре еах и остатка в регистре edx,
 4. Остаток от деления записывается в регистр edx.
 5. Инструкция “inc edx” используется для увеличения значения в регистре edx на
 6. В данном случае, она увеличивает остаток от деления на 1. 1з
 7. Строка “mov еах,edx” передает значение остатка от деления в регистр еах.
- 36 Строка “call iprintLF” вызывает процедуруiprintLF для вывода значения на экран вместе с переводом строки.

3.3 Задание для самостоятельной работы

Написать программу вычисления выражения $y = f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат

вычислений. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2

```

#include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
div: DB 'Результат: ',0
SECTION .bss
rez: RESB 80
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi

add eax, 2
mul eax
mov [rez], eax

mov eax, div
call sprint
mov eax, [rez]
call iprintLF

```

```
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./rabota
Введите x:
2
Результат: 16
dmmosharov@dmmosharov:~/work/arch-pc/lab06$ ./rabota
Введите x:
8
Результат: 100
dmmosharov@dmmosharov:~/work/arch-pc/lab06$
```

4 Выводы

В работе были изучены арифметические операции в языке ассемблера NASM. Был рассмотрен синтаксис и были написаны и проанализированы программы на ассемблере, которые используют арифметические операции для решения различных задач.