

Лабораторная работа №6

Арифметические операции в NASM

Виеру Женифер

Содержание

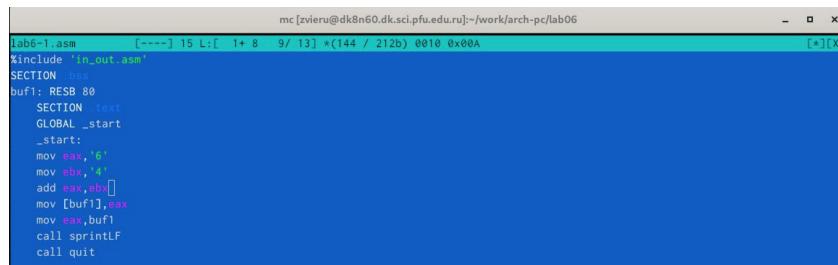
1 Цель работы	3
2 Выполнение лабораторной работы	4
3 Ответы на вопросы	11
4 Самостоятельная работа	12
5 Выводы	14

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

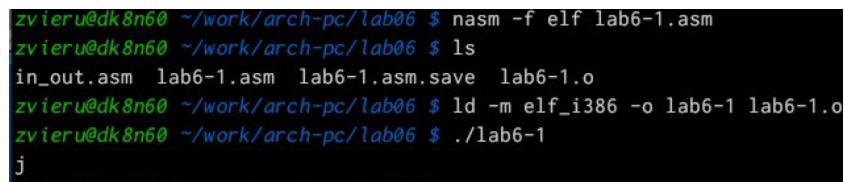
Создала каталог для программам лабораторной работы № 6, перешла в него и создала файл lab6-1.asm. Потом открыла его с помощью команды mc и горячай клавиши F4 и написала в нем следующий текст(рис. 2.1).



```
mc [zvieru@dk8n60.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab06
Lab6-1.asm [----] 15 L:[ 1+ 8 9/ 13] *(144 / 212b) 0010 0x00A
*include 'in_out.asm'
SECTION .bss
buf1: RESB 80
    SECTION .text
GLOBAL _start
_start:
    mov eax, '6'
    mov ebx, '4'
    add eax,ebx[]
    mov [buf1],eax
    mov eax,buf1
    call sprintLF
    call quit
```

Рис. 2.1: Программа вывода значения регистра eax

Создала исполняемый файл и запустила его (рис. 2.2).



```
zvieru@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ls
in_out.asm lab6-1.asm lab6-1.asm.save lab6-1.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 2.2: Запуск lab6-1.asm

Далее изменила текст программы и вместо символов, запишем в регистры числа. То есть вместе ,‘6’ и ,‘4’ я записала 6 и 4 (рис. 2.3).

```
mc [zvieru@dk8n60.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab06
lab6-1.asm      [---] 15 L:[ 1+ 8   9/ 13] *(140 / 208b) 0010 0x00A
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
    mov eax,6
    mov ebx,4
    add eax,ebx
    mov [buf1],eax
    mov eax,buf1
    call sprintLF
    call quit
```

Рис. 2.3: Код программы

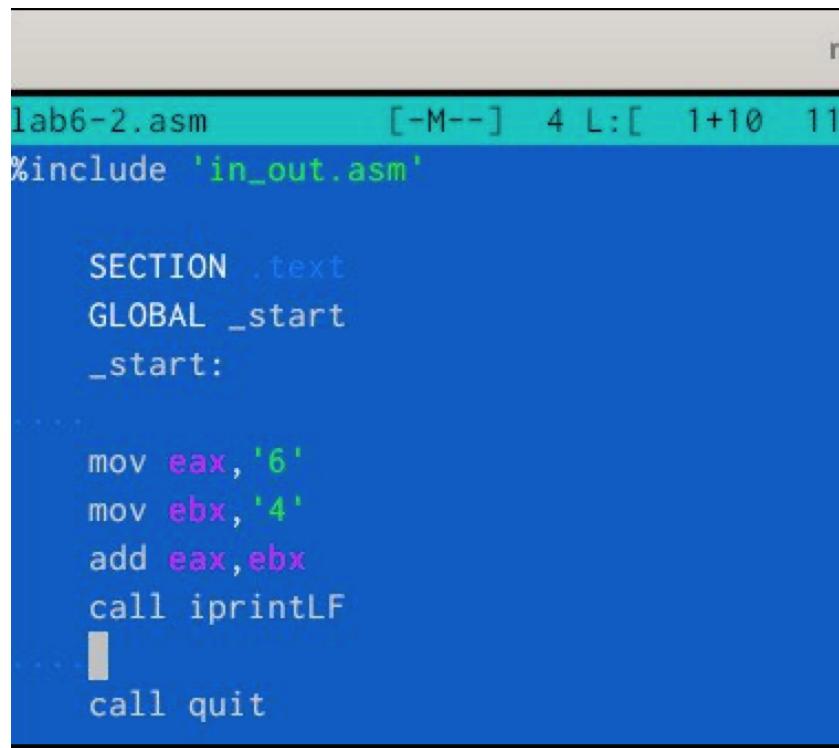
Создала исполняемый файл и запустила его (рис. 2.4).

```
zvieru@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1

zvieru@dk8n60 ~/work/arch-pc/lab06 $ mc
```

Рис. 2.4: Запуск lab6-1.asm

Создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввела в него следующий текст программы. (рис. 2.5).



```
lab6-2.asm      [-M--] 4 L:[ 1+10 11
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

    mov eax, '6'
    mov ebx, '4'
    add eax,ebx
    call iprintLF
    |
call quit
```

Рис. 2.5: Код программы lab6-2.asm

Далее изменила текст программы и вместо символов, запишем в регистры числа. То есть вместе ,‘6’ и ,‘4’ я записала 6 и 4 (рис. 2.6).

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:
        mov eax,6
        mov ebx,4
        add eax,ebx
        call iprintLF
        call quit
```

Рис. 2.6: Код программы lab6-2.asm

Создала исполняемый файл и запустила его (рис. 2.7).

```
zvieru@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10
zvieru@dk8n60 ~/work/arch-pc/lab06 $ |
```

Рис. 2.7: Запуск lab6-2.asm

Поменяла iprintLF на iprint и создала исполняемый файл и запустила его (рис. 2.8).

```

zvieru@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-2
10zvieru@dk8n60 ~/work/arch-pc/lab06 $

```

Рис. 2.8: Запуск lab6-1.asm

Создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 и написала программу для вычисления выражения (рис. 2.9).

```

lab6-3.asm      [---] 43 L:[ 1+14 15/ 26] *(491 /1177b) 0010
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit

```

Рис. 2.9: Программа вычисления выражения $f(5)$

Создала исполняемый файл и запустила его (рис. 2.10).

```

zvieru@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 2.10: Запуск lab6-3.asm

Изменила текст программы ля вычисления выражения $\square(\square)$ (рис. 2.11).

```
lab6-3.asm      [-M--] 9 L:[ 1+17 18/ 26] *(634 /1177b) 0105 0x069
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit
```

Рис. 2.11: Программа

Создала исполняемый файл и запустила его (рис. 2.12).

```
zvieru@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ./lab6-1

zvieru@dk8n60 ~/work/arch-pc/lab06 $ mc
```

Рис. 2.12: Запуск lab6-3.asm

Создала файл variant.asm в каталоге ~/work/arch-pc/lab06 и написала программу вычисления варианта задания по номеру студенческого билета : (рис. 2.13).

```
variant.asm      [-M--] 7 L:[ 1+19 20/ 25] *(391 / 45
%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 sprintLF
    mov ecx, x
    mov edx, 80
    call sread
    mov eax,x ; вызов подпрограммы преобразования
    call atoi
    xor edx,edx
    mov ebx,20
    div ebx
    inc edx[]
    mov eax,rem
    call sprint
    mov eax,edx
    call iprintLF
    call quit
```

Рис. 2.13: Программа вычисления вычисления варианта задания по номеру студенческого билета

Создала исполняемый файл и запустила его (рис. 2.14).

```
zvieru@dk8n60 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
zvieru@dk8n60 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132246785
```

Рис. 2.14: Запуск variant.asm

3 Ответы на вопросы

1 Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? mov eax,msg call sprintLF 2 Для чего используется следующие инструкции?Эти инструкции используются для ввода переменной X с клавиатуры и сохранения введенных данных. 3 Для чего используется инструкция “call atoi”?Эта инструкция используется для преобразования Кода переменной ASCII в число. 4 Какие строки листинга 6.4 отвечают за вычисления варианта? mov ebx,20 div ebx inc edx 5 В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? В регистре ebx 6 ля чего используется инструкция “inc edx”?Для увеличения значения edx на 1. 7 акие строки листинга 6.4 отвечают за вывод на экран результата вычислений? mov eax,edx call iprintLF

4 Самостоятельная работа

Создала файл lab6-0.asm в каталоге ~/work/arch-pc/lab06 и написала программу вычисления выражения $f(x)$ (рис. 4.1).

The screenshot shows a terminal window with the following assembly code:

```
Lab6-0.asm      [---] 41 L: 1+ 0   3/ 39] *(57  /115cb) 1077 0x435
include "in_out.asm"; подключение внешнего файла
SECTION data
f1: DB '(x*3)/2 + 1', 0
msg: DB "Введите значение переменной x: ", 0
SECTION code
GLOBAL _start
_start:
; ---- Вычисление выражения
    mov eax, f1
    call sprintF
    mov eax, msg
    call sprintF
    mov eax, 0
    mov edx, 80
    call sread
    mov ebx, eax ; вызов подпрограммы преобразования
    call atoi ; ASCII код в число, 'eax'
    mov ebx, eax
    mul ebx
    mov ebx, 2
    div ebx
    add eax, 1
    mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
    mov eax, 1
    call print ; сообщение 'Результат: '
    mov eax, edi ; вызов подпрограммы печати значения
    call iprintlf ; из 'edi' в виде символов
    call quit ; вызов подпрограммы завершения
```

Рис. 4.1: Программа вычисления выражения $f(x)$

Создала исполняемый файл, запустила его и проверила если программы работает (рис. 4.2).

The screenshot shows a terminal window with a dark background. The user has run the following commands:

```
zvleruldk3n55 ~/work/arch-pc/lab6 $ nasm -f elf lab6-0.asm
zvleruldk3n55 ~/work/arch-pc/lab6 $ ld -m elf_i386 -o lab6-0 lab6-0.o
zvleruldk3n55 ~/work/arch-pc/lab6 $ ./lab6-0
(x*3)/2 + 1
Введите значение переменной x:
2
Результат: 5
zvleruldk3n55 ~/work/arch-pc/lab6 $ nasm -f elf lab6-0.asm
zvleruldk3n55 ~/work/arch-pc/lab6 $ ld -m elf_i386 -o lab6-0 lab6-0.o
zvleruldk3n55 ~/work/arch-pc/lab6 $ ./lab6-0
(x*3)/2 + 1
Введите значение переменной x:
5
Результат: 63
```

Рис. 4.2: Запуск lab6-0.asm

5 Выводы

Выполнив данную лабораторную работу я обрела теоретические и практические знания в NASM. Я научилась выполнять самые разные арифметические операции