

Лабораторная Работа №6. Освоение арифметических инструкций языка ассемблера NASM.

Дисциплина: Архитектура ЭВМ

Иван Шевырев

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Символьные и численные данные в NASM	6
2.1.1	Изменим текстовый файл, заменив символы числами . . .	7
2.2	Программа для вывода значения eax	8
2.2.1	Заменим символы на числа	10
2.2.2	Заменим <code>iprint</code> на <code>iprintLF</code>	11
2.3	Выполнение арифметических операций в NASM	11
2.4	Программа вычисления варианта	12
3	Задания для самостоятельной работы	14
4	Выводы	16

Список иллюстраций

2.1	Создание исполняемого файла lab6-1	7
2.2	Изменение файла lab6-1.asm	7
2.3	Вывод измененной программы lab6-1	8
2.4	Код lab6-2.asm	9
2.5	Запуск lab6-2	9
2.6	Изменение программы lab6-2.asm	10
2.7	Вывод lab6-2	10
2.8	Изменение исходного текста lab6-2	11
2.9	Запуск измененной lab6-2	11
2.10	Открытие файла в nano	11
2.11	Запуск программы	12
2.12	Измененная часть программы lab6-2.asm	12
2.13	Вывод результата программой	12
2.14	Код variant.asm	13
2.15	Вычисление варианта по номеру студенческого билета	13
3.1	Код созданной программы lab6-4.asm	14
3.2	Проверка программы на данных значениях	15

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

2.1 Символьные и численные данные в NASM

Создадим каталог lab06 и перейдем в него, используя команды:

```
mkdir ~/work/arch-pc/lab06
```

```
cd touch lab6-1.asm
```

Создадим текстовый файл lab6-1.asm и введем в него листинг 7.1

```
%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 sprintf
call quit
```

Создадим исполняемый файл и запустим его

```
[idshevihryov@teapod lab06]$  
nasm -f elf ./lab6-1.asm  
ld -m elf_i386 -o ./lab6-1 ./lab6-1.o  
./lab6-1  
  
j  
[idshevihryov@teapod lab06]$
```

Рис. 2.1: Создание исполняемого файла lab6-1

На экран вывелось j

2.1.1 Изменим текстовый файл, заменив символы числами

```
_start:  
mov eax,6  
mov ebx,4  
add eax,ebx  
mov [buf1],eax  
mov eax,buf1  
call sprintf  
  
call quit
```

Рис. 2.2: Изменение файла lab6-1.asm

Создадим исполняемый файл и запустим его.

```
[idshevihryov@teapod lab06]$ nano lab6-1.asm
[idshevihryov@teapod lab06]$
nasm -f elf ./lab6-1.asm
ld -m elf_i386 -o ./lab6-1 ./lab6-1.o
./lab6-1

[idshevihryov@teapod lab06]$
```

Рис. 2.3: Вывод измененной программы lab6-1

Видим, что вывелся перенос строки, потому что теперь программа выводит, не символ с кодом '6' + '4' = 106, а с кодом 6+4 = 10

2.2 Программа для вывода значения eax

Введем код из листинга 7.2


```
%include 'in_out.asm'
SECTION .text
GLOBAL _start

_start:

mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprint
call quit
```

Рис. 2.4: Код lab6-2.asm

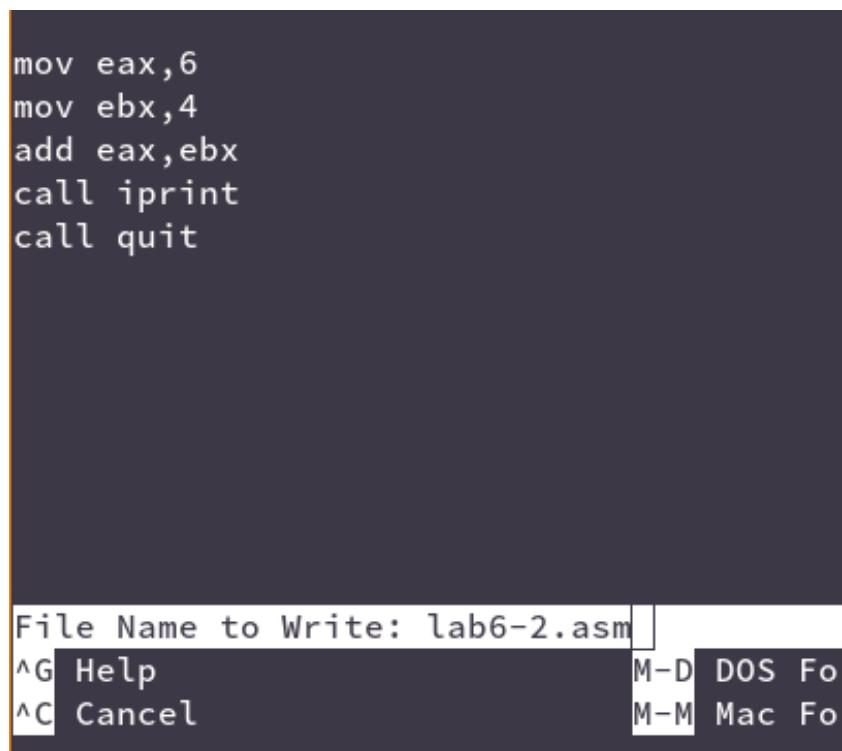
Запустим программу

```
[idshevihryov@teapod lab06]$
nasm -f elf ./lab6-2.asm
ld -m elf_i386 -o ./lab6-2 ./lab6-2.o
./lab6-2

106[idshevihryov@teapod lab06]$
```

Рис. 2.5: Запуск lab6-2

2.2.1 Заменим символы на числа



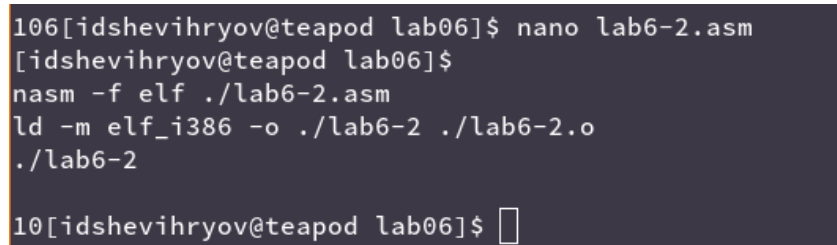
```
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

File Name to Write: lab6-2.asm

^G Help M-D DOS Fo
^C Cancel M-M Mac Fo

Рис. 2.6: Изменение программы lab6-2.asm

Создадим исполняемый файл и запустим



```
106[idshevihryov@teapod lab06]$ nano lab6-2.asm
[idshevihryov@teapod lab06]$
nasm -f elf ./lab6-2.asm
ld -m elf_i386 -o ./lab6-2 ./lab6-2.o
./lab6-2

10[idshevihryov@teapod lab06]$
```

Рис. 2.7: Вывод lab6-2

Вывелось 10, т.е. код переноса строки

2.2.2 Заменим `iprint` на `iprintLF`

```
mov ebx, 4  
add eax, ebx  
call iprintLF  
call quit
```

Рис. 2.8: Изменение исходного текста lab6-2

Запустим программу заново

```
[idshevihryov@teapod lab06]$  
nasm -f elf ./lab6-2.asm  
ld -m elf_i386 -o ./lab6-2 ./lab6-2.o  
./lab6-2  
  
10  
[idshevihryov@teapod lab06]$
```

Рис. 2.9: Запуск измененной lab6-2

Вывелось 10 с переносом строки

2.3 Выполнение арифмитических операций в NASM

Создадим и откроем файл lab6-3.asm в текстовом редакторе (nano)

```
[idshevihryov@teapod lab06]$ nano lab6-3.asm
```

Рис. 2.10: Открытие файла в nano

Введем код из листинга, создадим исполняемый файл и запустим программу

```
[idshevihryov@teapod lab06]$
nasm -f elf ./lab6-3.asm
ld -m elf_i386 -o ./lab6-3 ./lab6-3.o
./lab6-3

Результат: 4
Остаток от деления: 1
[idshevihryov@teapod lab06]$
```

Рис. 2.11: Запуск программы

Как видим, программа вывела $(5 * 2 + 3) / 3 = 13 / 3 = 4.333 = 4$

Изменим программу так, что бы она вычисляла значение $(4 * 6 + 2) / 5$

```
GLOBAL _start
_start: ; ---- 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/3, EDX=остаток от деления
```

Рис. 2.12: Измененная часть программы lab6-2.asm

Странслируем, слинкуем и запустим программу:

```
nasm -f elf ./lab6-3.asm
ld -m elf_i386 -o ./lab6-3 ./lab6-3.o
./lab6-3

Результат: 5
Остаток от деления: 1
```

Рис. 2.13: Вывод результата программой

$$(4 * 6 + 2) / 5 = 26 / 5 = 5$$

2.4 Программа вычисления варианта

Создадим файл variant.asm и введем в него текст из листинга 7.4

```

;-----
; 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          ; адресс `x` в ecx
    mov edx,80         ; размер буфера в edx
    call sread         ; чтение ввода в `x`

    mov eax,x ; вызов подпрограммы преобразования
    call atoi ; ASCII кода в число, `eax=x`

    xor edx,edx        ; очищение edx
    mov ebx,20         ; 20 в ebx
    div ebx            ; делим значение из eax на 20
    inc edx            ; увеличиваем остаток от деления на 1

    mov eax,rem         ; Запись адреса строки 'Ваш вариант' в eax
    call sprint         ; Вызов вывода строки без переноса строки
    mov eax,edx         ; перенос вычисленного остатка в eax
    call iprintf        ; Вывод вычисленного остатка с переносом строки

    call quit

```

Рис. 2.14: Код variant.asm

Запустим программу и введем в нее номер нашего студенческого билета.

```

nasm -f elf ./variant.asm
ld -m elf_i386 -o ./variant ./variant.o
./variant

Введите № студенческого билета:
1132229881
Ваш вариант: 2
[idshevihryov@teapod lab06]$ 

```

Рис. 2.15: Вычисление варианта по номеру студенческого билета

Наш номер - 2

3 Задания для самостоятельной работы

Так как наш вариант: 2, нам нужно написать программу, которая будет вычислять значение функции $f(x) = (12x + 3)5$

Напишем эту программу

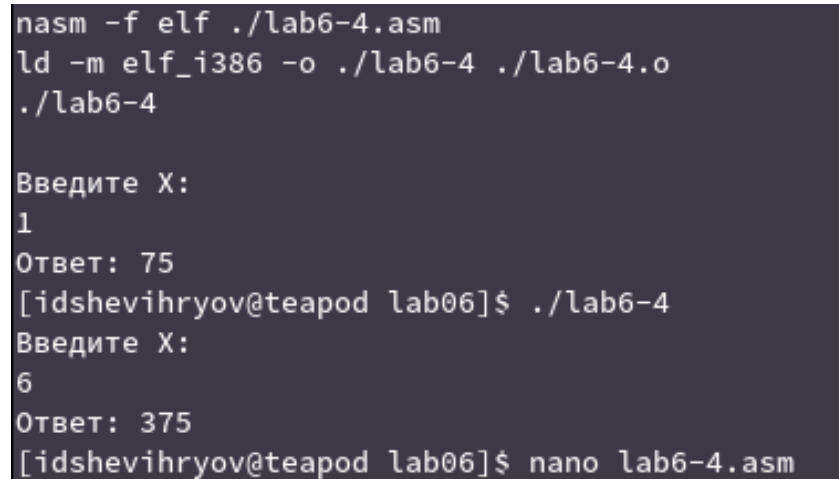
```
-----  
; Программа вычисления варианта  
-----  
%include 'in_out.asm'  
  
SECTION .data  
msg: DB 'Введите X: ',0  
rem: DB 'Ответ: ',0  
  
SECTION .bss  
x:    RESB 10  
  
SECTION .text  
GLOBAL _start  
_start:  
    mov eax, msg  
    call sprintf  
  
    mov ecx, x          ; x в ecx  
    mov edx, 10         ; размер edx  
    call sread          ; ввод в x  
  
    mov eax, x ; вызов подпрограммы преобразования  
    call atoi ; ASCII кода в число, `eax=x`  
  
    mov ecx, 12          ; ecx = 12  
    mul ecx              ; eax *= x  
    add eax, 3           ; eax += 3  
    mov ecx, 5           ; ecx = 5  
    mul ecx              ; eax *= 5  
  
    mov edx, eax  
    xor eax, eax  
  
    mov eax, rem         ; Запись адреса строки 'Ваш вариант' в eax  
    call sprint          ; Вызов вывода строки без переноса строки  
    mov eax, edx         ; перенос вычисленного остатка в eax  
    call iprintLF        ; Вывод вычисленного остатка с переносом строки  
  
    call quit
```

Рис. 3.1: Код созданной программы lab6-4.asm

Создадим исполняемый файл и проверим данную программу на указанных значениях

$x1 = 1$

$x2 = 6$



```
nasm -f elf ./lab6-4.asm
ld -m elf_i386 -o ./lab6-4 ./lab6-4.o
./lab6-4

Введите X:
1
Ответ: 75
[idshevihryov@teapod lab06]$ ./lab6-4
Введите X:
6
Ответ: 375
[idshevihryov@teapod lab06]$ nano lab6-4.asm
```

Рис. 3.2: Проверка программы на данных значениях

Функция вычисляется верно

$$f(1) = (12 + 3) * 5 = 75$$

$$f(6) = (12 * 6 + 3)5 = 75 * 5 = 375$$

4 Выводы

Мы освоили арифметические инструкции языка ассемблера NASM и написали программу по вычислению значения функции $f(x) = (12x + 3)^5$