

Лабораторная работа №6

Арифметические операции в NASM

Налобин Михаил Дмитриевич

Содержание

1	Цель работы	5
2	Ход работы	6
3	Выводы	23

Список иллюстраций

2.1	Создание каталога lab06	6
2.2	Код программы lab6-1.asm	7
2.3	Запуск программы lab6-1	8
2.4	Изменение кода программы lab6-1.asm	9
2.5	Запуск измененной программы lab6-1	10
2.6	Создание lab6-2.asm	10
2.7	Код программы lab6-2.asm	11
2.8	Запуск программы lab6-2	11
2.9	Изменение кода программы lab6-2.asm	12
2.10	Запуск измененной программы lab6-2	12
2.11	Замена <code>iprintLF</code> на <code>iprint</code> в файле lab6-2.asm	13
2.12	Запуск программы с заменой lab6-2	13
2.13	Создание lab6-3.asm	14
2.14	Код программы lab6-3.asm	15
2.15	Запуск программы lab6-3	16
2.16	Код измененной программы lab6-3.asm	17
2.17	Запуск измененной программы lab6-3	18
2.18	Создание variant.asm	18
2.19	Код программы variant.asm	19
2.20	Запуск программы variant	20
2.21	Код программы sr.asm	21
2.22	Запуск программы sr	22

Список таблиц

1 Цель работы

Освоение навыка использования арифметических инструкций языка ассемблера NASM.

2 Ход работы

Создали каталог lab06 для файлов лабораторной работы №6 и в нем файл lab6-1.asm (рис. 2.1).

```
mdnalobin@dk3n62 ~ $ cd work/arch-pc/  
mdnalobin@dk3n62 ~/work/arch-pc $ ls  
lab05  
mdnalobin@dk3n62 ~/work/arch-pc $ mkdir lab06  
mdnalobin@dk3n62 ~/work/arch-pc $ cd lab06  
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ touch lab6-1.asm
```

Рис. 2.1: Создание каталога lab06

Открыли с помощью gedit файл lab6-1.asm и переписали в него пример Листинга 6.1. (рис. 2.2).

```
%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 sprintfLF

    call quit
```

Рис. 2.2: Код программы lab6-1.asm

Далее создали исполняемый файл и запустили его (рис. 2.3).

```
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 2.3: Запуск программы lab6-1

Снова открыли файл lab6-1.asm и заменили символы на регистры числа, после чего создали исполняемый файл и запустили его. В итоге наблюдаем, что символ не отображается, так как **символ с кодом 10 является перевод на новую строку** (рис. 2.4 и рис. 2.5).


```
%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 sprintfLF

        call quit
```

Рис. 2.4: Изменение кода программы lab6-1.asm

```
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ./lab6-1

mdnalogin@dk3n62 ~/work/arch-pc/lab06 $
```

Рис. 2.5: Запуск измененной программы lab6-1

Создали файл lab6-2.asm и открыли для редактирования с помощью gedit (рис. 2.6).

```
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ touch lab6-2.asm
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ gedit lab6-2.asm
```

Рис. 2.6: Создание lab6-2.asm

Используя в качестве примера Листинг 6.2., заполнили lab6-2.asm (рис. 2.7).

```
%include 'in_out.asm'
```

```
SECTION .text
```

```
GLOBAL _start
```

```
    _start:
```

```
    mov eax, '6'
```

```
    mov ebx, '4'
```

```
    add eax, ebx
```

```
    call iprintLF
```

```
    call quit
```

Рис. 2.7: Код программы lab6-2.asm

Затем создали исполняемый файл lab6-2 и запустили его (рис. 2.8).

```
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ ./lab6-2
106
```

Рис. 2.8: Запуск программы lab6-2

Как и в прошлом заменили символы на регистры числа, но уже в програм-

ме lab06-2.asm, создали его исполняемый файл и запустили. **В результате на выводе получили число 10** (рис. 2.9 и рис. 2.10).

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

    mov eax, 6
    mov ebx, 4
    add eax, ebx
    call iprintLF

    call quit
```

Рис. 2.9: Изменение кода программы lab6-2.asm

```
mdnlobin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
mdnlobin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
mdnlobin@dk3n62 ~/work/arch-pc/lab06 $ ./lab6-2
10
```

Рис. 2.10: Запуск измененной программы lab6-2

Заменили iprintLF на iprint в файле lab6-2.asm, после чего создали исполняемый

файл, проверили работу и заметили, что **iprint не переводит на новую строку, а работа продолжается на прежней** (рис. 2.11 и рис. 2.12).

```
%include 'in_out.asm

SECTION .text
GLOBAL _start
_start:

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

    call quit
```

Рис. 2.11: Замена iprintLF на iprint в файле lab6-2.asm

```
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ gedit lab6-2.asm
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ ./lab6-2
10mdnalobin@dk3n62 ~/work/arch-pc/lab06 $
```

Рис. 2.12: Запуск программы с заменой lab6-2

Создали файл lab6-3.asm (рис. 2.13).

```
mdn@alobin@dk3n62 ~/work/arch-pc/lab06 $ touch lab6-3.asm
```

Рис. 2.13: Создание lab6-3.asm

Ввели в lab6-3.asm содержимое Листинга 6.3., создали исполняемый файл и запустили его (рис. 2.14 и рис. 2.15).

```
%include 'in_out.asm'

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

    mov eax,5
    mov ebx,2
    mul ebx
    add eax,3
    xor edx,edx
    mov ebx,3
    div ebx

    mov edi,eax

    mov eax,div
    call sprint
    mov eax,edi
    call iprintLF

    mov eax,rem
    call sprint
    mov eax,edx
    call iprintLF

    call quit
```

Рис. 2.14: Код программы lab6-3.asm

```
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 2.15: Запуск программы lab6-3

Далее изменили текст программы для вычисления следующего выражения $f(x)$
 $= (4 \cdot 6 + 2)/5^*$ (рис. 2.16 и рис. 2.17).


```

#include 'in_out.asm'

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

    mov eax,4
    mov ebx,6
    mul ebx
    add eax,2
    xor edx,edx
    mov ebx,5
    div ebx

    mov edi,eax

    mov eax,div
    call sprint
    mov eax,edi
    call iprintLF

    mov eax,rem
    call sprint
    mov eax,edx
    call iprintLF

    call quit

```

Рис. 2.16: Код измененной программы lab6-3.asm

```
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 2.17: Запуск измененной программы lab6-3

В качестве другого примера создали файл variant.asm (рис. 2.18).

```
mdnalogin@dk3n62 ~/work/arch-pc/lab06 $ touch variant.asm
```

Рис. 2.18: Создание variant.asm

Записали в него текст из Листинга 6.4. (рис. 2.19).

```

#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
    call atoi

    xor edx, edx
    mov ebx, 20
    div ebx
    inc edx

    mov eax, rem
    call sprintf
    mov eax, edx
    call iprintLF

    call quit

```

Рис. 2.19: Код программы variant.asm

Создали исполняемый файл variant и запустили его. Получили свой вариант, в моем случае - 6 (рис. 2.20).

```
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
mdnalobin@dk3n62 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132236105
Ваш вариант: 6
```

Рис. 2.20: Запуск программы variant

Ответы на вопросы: 1. `mov eax,rem call sprint` 2. Они используются, чтобы в переменную `x` поместить число, введенное с клавиатуры. 3. Данная инструкция нужна для перевода символа числа в десятичную символьную запись. 4. `mov ebx,20 div ebx inc edx` 5. Остаток от деления записывает в регистр `EDX`. 6. `inc edx` увеличивает свой операнд на 1. 7. `mov eax,edx call iprintLF`

##Самостоятельная работа

В новосозданном файле `sr.asm` и написали в нем код программы для выполнения выражения $(x^3)/2+1$ (рис. 2.21).

```

SECTION .data
msg:    DB 'Введите значение x: ',0
div:    DB 'Результат: ',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
    call atoi

    mov ebx, eax
    mul ebx
    mul ebx
    mov ebx, 2
    div ebx
    inc eax

    mov edi, eax

    mov eax, div
    call sprint
    mov eax, edi
    call iprintLF

    call quit

```

Рис. 2.21: Код программы sr.asm

Затем сделали исполняемый файл и проверили работу программы для значений 2 и 5 (рис. 2.22).

```
[mdnalogin@mdnalogin lab06]$ ./sr
Введите значение x:
2
Результат: 5
[mdnalogin@mdnalogin lab06]$ ./sr
Введите значение x:
5
Результат: 63
```

Рис. 2.22: Запуск программы sr

3 Выводы

В ходе данной лабораторной работы освоили арифметические инструкции языка ассемблера NASM и на основе приобретенных знаний создали свою программу для вычисления выражения.

...