

Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Бережной Иван Александрович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Разбор символьных и численных данных в NASM	7
3.2	Выполнение арифметических операций в NASM	11
3.3	Ответы на вопросы	14
3.4	Задание для самостоятельной работы	15
4	Выводы	17
	Список литературы	18

Список иллюстраций

3.1	Подготовка к работе	7
3.2	Копирование кода в lab6-1.asm	8
3.3	Запуск файла lab6-1.exe	8
3.4	Редактирование lab6-1.exe	9
3.5	Запуск изменённого файла lab6-1.exe	9
3.6	Копирование кода в lab6-2.asm	10
3.7	Запуск файла lab6-2.exe	10
3.8	Редактирование lab6-2.asm	11
3.9	Запуск изменённого файла lab6-2.exe	11
3.10	Запуск дважды изменённого файла lab6-2.exe	11
3.11	Копирование кода в lab6-3.asm	12
3.12	Запуск файла lab6-3.exe	12
3.13	Запуск изменённого файла lab6-3.exe	13
3.14	Копирование кода в variant.asm	13
3.15	Запуск файла variant.exe	14
3.16	Проверка работоспособности программы	15

Список таблиц

1 Цель работы

Освоить арифметические инструкции языка ассемблера NASM.

2 Задание

1. Разбор символьных и численных данных в NASM
2. Выполнение арифметических операций в NASM
3. Ответы на вопросы
4. Задание для самостоятельной работы

3 Выполнение лабораторной работы

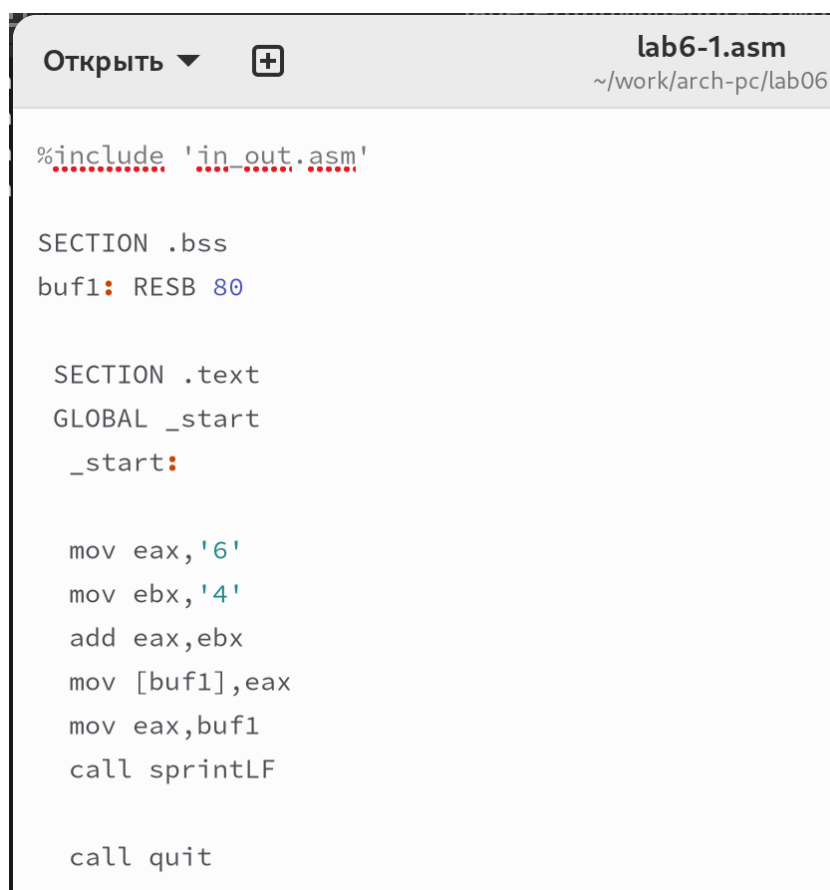
3.1 Разбор символьных и численных данных в NASM

Для начала создадим каталог lab06 для последующей работы. Для этого воспользуемся командой `mkdir`. Перейдём в него и создадим файл `lab6-1.asm` (рис. 3.1).

```
[iaberezhnoy@fedora ~]$ mkdir ~/work/arch-pc/lab06  
[iaberezhnoy@fedora ~]$ cd ~/work/arch-pc/lab06  
[iaberezhnoy@fedora lab06]$ touch lab6-1.asm
```

Рис. 3.1: Подготовка к работе

Скопируем предложенный листинг в новый файл (рис. 3.2). Создадим и запустим исполняемый файл (рис. 3.3). В выводе видим букву “j”.




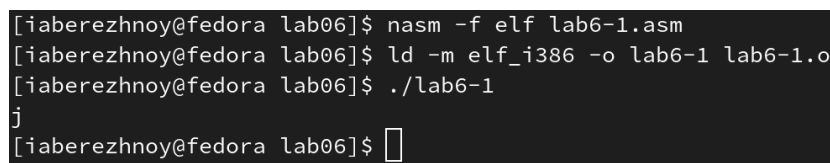
```
Открыть ▾  lab6-1.asm  
~/work/arch-pc/lab06  
  
%include 'in_out.asm'  
  
SECTION .bss  
buf1: RESB 80  
  
SECTION .text  
GLOBAL _start  
_start:  
  
    mov eax, '6'  
    mov ebx, '4'  
    add eax, ebx  
    mov [buf1], eax  
    mov eax, buf1  
    call sprintLF  
  
    call quit
```

Рис. 3.2: Копирование кода в lab6-1.asm



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

Рис. 3.3: Запуск файла lab6-1.exe

Немного изменим текст программы, убрав одинарные кавычки в строках `mov eax, '6'` и `mov ebx, '4'` (рис. 3.4). Также пересоздадим и запустим исполняемый файл - получили перевод строки (рис. 3.5).

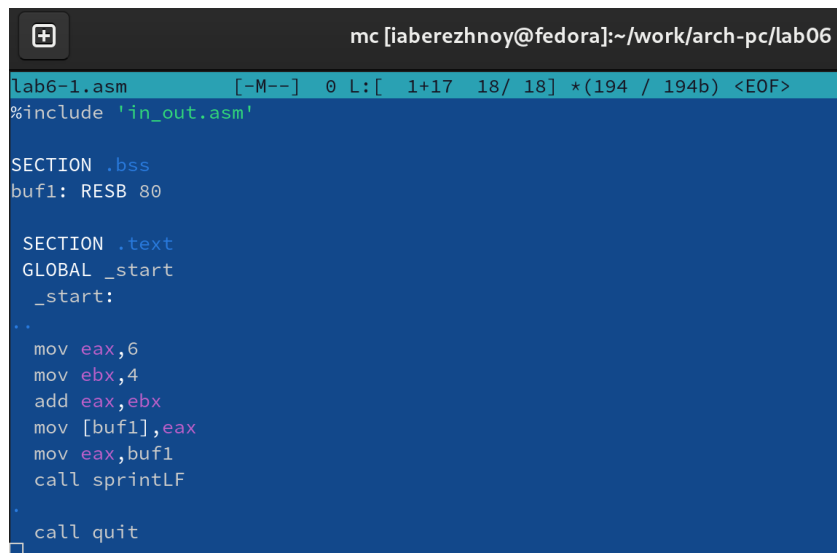
A screenshot of a text editor window titled 'mc [iaberezhnoy@fedora]:~/work/arch-pc/lab06'. The editor shows the file 'lab6-1.asm' with a status bar indicating line 1 of 17, column 18, and a buffer size of 194 bytes. The code includes a header file 'in_out.asm', defines a .bss section for a buffer 'buf1' of size 80, and a .text section starting at '_start'. The assembly code moves the value 6 into 'eax', moves 4 into 'ebx', adds 'ebx' to 'eax', stores the result in 'buf1', moves 'buf1' into 'eax', calls 'sprintf', and finally calls 'quit'.

Рис. 3.4: Редактирование lab6-1.exe

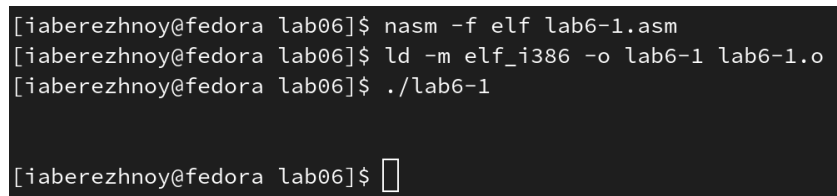
A screenshot of a terminal window showing the compilation and execution of the assembly file. The commands and their outputs are: 'nasm -f elf lab6-1.asm' (no output), 'ld -m elf_i386 -o lab6-1 lab6-1.o' (no output), and './lab6-1' (no output). The prompt returns to the shell.

Рис. 3.5: Запуск изменённого файла lab6-1.exe

Создадим файл lab6-2.asm и скопируем в него второй листинг (рис. 3.6). Для него тоже создадим исполняемый файл и запустим его. В выводе получили 106 (рис. 3.7).

```

%include 'in_out.asm'

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

```

Рис. 3.6: Копирование кода в lab6-2.asm

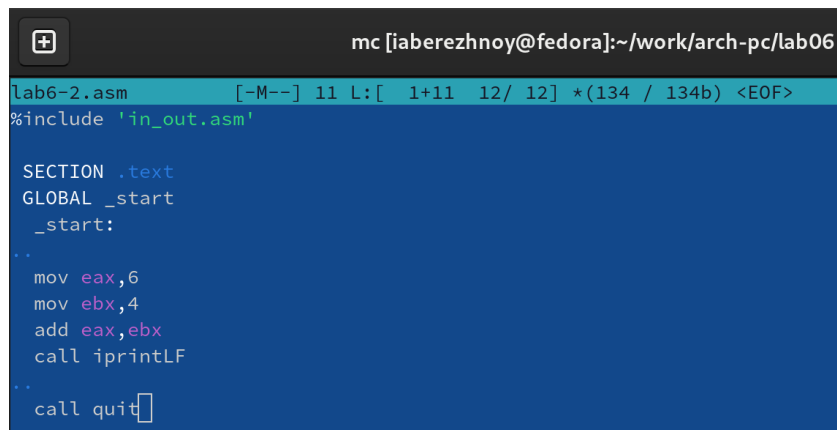
```

[iaberezhnoy@fedora lab06]$ nasm -f elf lab6-2.asm
[iaberezhnoy@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[iaberezhnoy@fedora lab06]$ ./lab6-2
106
[iaberezhnoy@fedora lab06]$ 

```

Рис. 3.7: Запуск файла lab6-2.exe

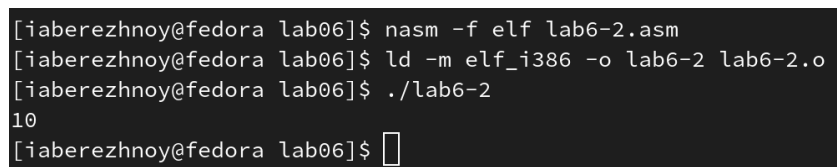
Аналогично предыдущей программе изменим в этой паре строк (рис. 3.8) и запустим уже изменённый файл. В результате получили 10 (рис. 3.9).



```
mc [iaberezhnoy@fedora]:~/work/arch-pc/lab06
lab6-2.asm  [-M--] 11 L: [ 1+11 12/ 12] *(134 / 134b) <EOF>
#include 'in_out.asm'

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

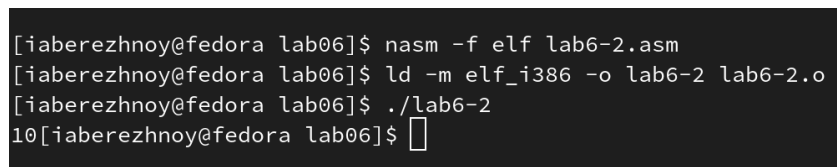
Рис. 3.8: Редактирование lab6-2.asm



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

Рис. 3.9: Запуск изменённого файла lab6-2.exe

Теперь заменим всё в этом же файле функцию iprintLF на iprint. Создадим исполняемый файл и запустим его. Получили всё тот же результат, но теперь отсутствует перевод строки (рис. 3.10).



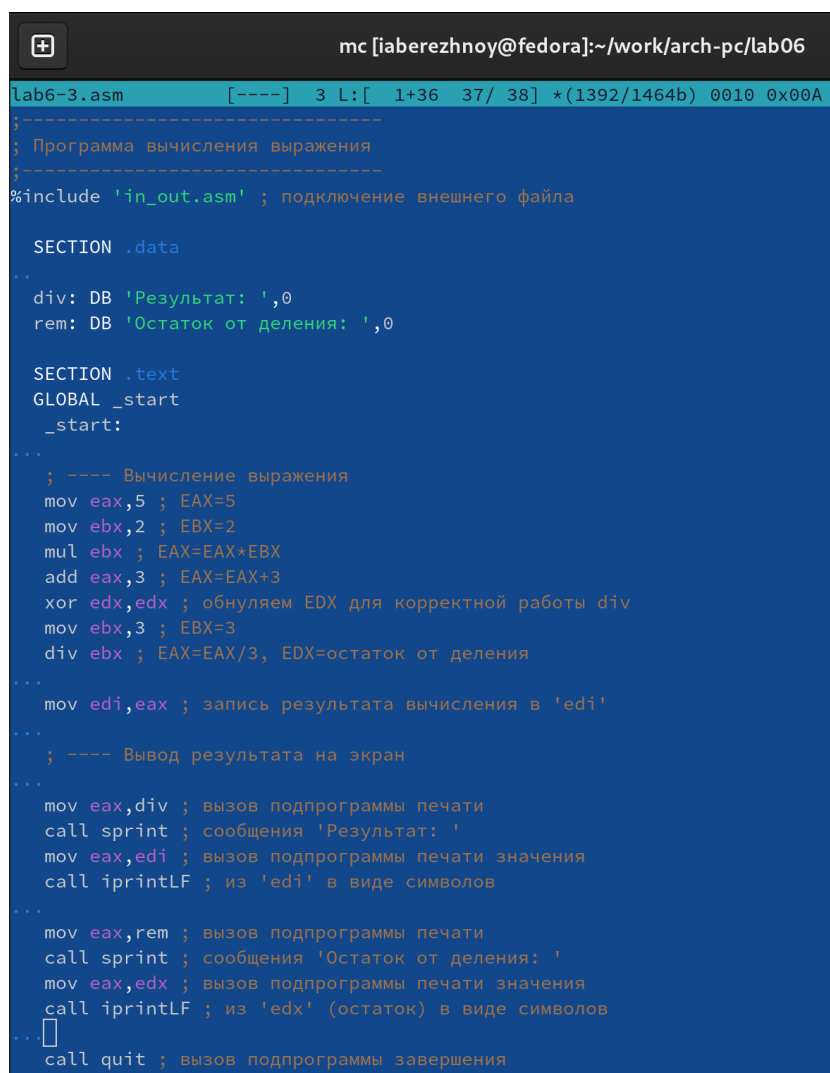
```
[iaberezhnoy@fedora lab06]$ nasm -f elf lab6-2.asm
[iaberezhnoy@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[iaberezhnoy@fedora lab06]$ ./lab6-2
10[iaberezhnoy@fedora lab06]$
```

Рис. 3.10: Запуск дважды изменённого файла lab6-2.exe

3.2 Выполнение арифметических операций в NASM

Создадим файл lab6-3.asm и скопируем в него третий код (рис. 3.11). Создадим и запустим исполняемый файл - в результате получили 4, а остаток от деления 1

(рис. 3.12).

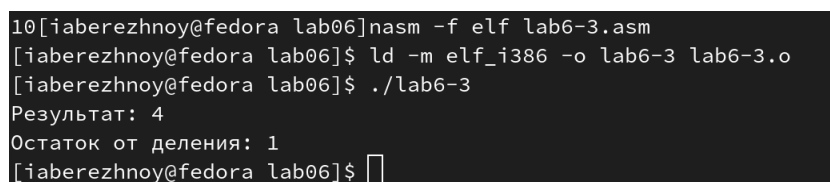


```
mc [iaberezhnoy@fedora]:~/work/arch-pc/lab06
lab6-3.asm [----] 3 L: [ 1+36 37/ 38] *(1392/1464b) 0010 0x00A
;-----
; Программа вычисления выражения
;-----
#include 'in_out.asm' ; подключение внешнего файла

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 ; вызов подпрограммы завершения
```

Рис. 3.11: Копирование кода в lab6-3.asm



```
10[iaberezhnoy@fedora lab06]nasm -f elf lab6-3.asm
[iaberezhnoy@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[iaberezhnoy@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[iaberezhnoy@fedora lab06]$
```

Рис. 3.12: Запуск файла lab6-3.exe

Изменим текст программы так, чтобы вычислялось значение выражения

$(4*6+2)/5$. Проверим работу файла (рис. 3.13).

```
[iaberezhnoy@fedora lab06]$ nasm -f elf lab6-3.asm
[iaberezhnoy@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[iaberezhnoy@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[iaberezhnoy@fedora lab06]$
```

Рис. 3.13: Запуск изменённого файла lab6-3.exe

Создадим файл variant.asm командой touch, скопируем в него листинг №4 (рис. 3.14) и проверим работы исполняемого файла. Для этого введём 1132236041 - получили 2 (рис. 3.15), а значит, в задании для самостоятельной работы мы будем выполнять вариант под номером 2.

```
mc [iaberezhnoy@fedora]:~/work/arch-pc/lab06
variant.asm [-M--] 9 L:[ 1+27 28/ 28] *(511 / 511b) <EOF>
;-----
; 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 sprintLF
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 sprint
mov eax, edx
call iprintLF
call quit
```

Рис. 3.14: Копирование кода в variant.asm

```
[iaberezhnoy@fedora lab06]$ nasm -f elf variant.asm
[iaberezhnoy@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[iaberezhnoy@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236041
Ваш вариант: 2
[iaberezhnoy@fedora lab06]$
```

Рис. 3.15: Запуск файла variant.exe

3.3 Ответы на вопросы

1. За вывод на экран сообщения “Ваш вариант:” отвечают строки

```
mov eax,rem
call sprint
```

2. Первая инструкция используется, чтобы скопировать адрес вводимой строки в регистр еsx; Вторая инструкция используется, чтобы установить в регистр edx длину вводимой строки; Третья инструкция используется, чтобы вызвать подпрограмму из внешнего файла, обеспечивающей ввод текста с клавиатуры.
3. Инструкция call atoi используется, чтобы преобразовать ascii-код символа в целое число.
4. За вычисление варианта в листинге 6.4 отвечают следующие строки:

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

5. Остаток от деления при выполнении инструкции div ebx записывается в регистр edx.
6. Инструкция inc edx используется для увеличения значения в регистре edx на 1.

7. За вывод результата вычислений на экран в листинге 6.4 отвечают следующие строки:

```
mov eax,edx
call iprintLF
```

3.4 Задание для самостоятельной работы

Напишем программу для вычисления выражения $y = (12x + 3)^5$ в файле lab6-4.asm (листинг 6.1). Проверим её работоспособность (рис. 3.16)

```
[iaberezhnoy@fedora lab06]$ nasm -f elf lab6-4.asm
[iaberezhnoy@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[iaberezhnoy@fedora lab06]$ ./lab6-4
Введите значение переменной x: 1
Результат: 75[iaberezhnoy@fe./lab6-4
Введите значение переменной x: 6
Результат: 375[iaberezhnoy@fedora lab06]$
```

Рис. 3.16: Проверка работоспособности программы

Листинг 6.1. Программа вычисления выражения $y = (12x + 3)^5$

```
%include 'in_out.asm'

SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
```

```

_start:
    ; ---- Ввод значения переменной x
    mov eax, msg
    call sprint
    mov ecx, x
    mov edx, 80
    call sread

    ; ---- Вычисление выражения  $(12x + 3)5$ 
    mov eax, x
    call atoi
    mov ebx, 12
    imul eax, ebx
    add eax, 3
    imul eax, 5
    mov edi, eax

    ; ---- Вывод результата на экран
    mov eax, rem
    call sprint
    mov eax, edi
    call iprint
    call quit

```


4 Выводы

В ходе выполнения лабораторной работы мы разобрали символьные и численные данные в NASM, а также освоили арифметические операции языка ассемблера NASM.

Список литературы

::: Архитектура ЭВМ