

Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Пономарева Татьяна Александровна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Символьные и численные данные в NASM	6
2.2	Выполнение арифметических операций в NASM	9
3	Задание для самостоятельной работы	14
4	Выводы	16
	Список литературы	17

Список иллюстраций

2.1	Терминал. Создание каталога lab06 и файла lab6-1.asm	6
2.2	Окно Midnight Commander. Содержание файла lab6-1.asm	6
2.3	Терминал. Создание исполняемого файла lab6-1 и его запуск . . .	6
2.4	Окно Midnight Commander. Содержание файла lab6-1.asm	7
2.5	Терминал. Создание исполняемого файла lab6-1 и его запуск . . .	7
2.6	Терминал. Создание файла lab6-2.asm	7
2.7	Окно Midnight Commander. Содержание файла lab6-2.asm	8
2.8	Терминал. Создание исполняемого файла lab6-2 и его запуск . . .	8
2.9	Окно Midnight Commander. Содержание файла lab6-2.asm	8
2.10	Терминал. Создание исполняемого файла lab6-2 и его запуск . . .	8
2.11	Окно Midnight Commander. Содержание файла lab6-2.asm	9
2.12	Терминал. Создание исполняемого файла lab6-2 и его запуск . . .	9
2.13	Терминал. Создание файла lab6-3.asm	9
2.14	Окно Midnight Commander. Содержание файла lab6-3.asm	10
2.15	Терминал. Создание исполняемого файла lab6-3 и его запуск . . .	10
2.16	Окно Midnight Commander. Содержание файла lab6-3.asm	10
2.17	Терминал. Создание исполняемого файла lab6-3 и его запуск . . .	11
2.18	Терминал. Создание файла variant.asm	11
2.19	Окно Midnight Commander. Содержание файла variant.asm	11
2.20	Терминал. Создание исполняемого файла variant и его запуск . . .	11
3.1	Терминал. Создание файла lab6-4.asm	14
3.2	Окно Text Editor. Содержание файла lab6-4.asm	15
3.3	Терминал. Проверка вычисления программы варианта 3.	15

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

2.1 Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы № 6 , далее перехожу в него и создаю файл lab6-1.asm (рис. 2.1).

```
taponomareva@2c7fe9w:~$ mkdir ~/work/arch-pc/lab06
taponomareva@2c7fe9w:~$ cd ~/work/arch-pc/lab06
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ touch lab6-1.asm
```

Рис. 2.1: Терминал. Создание каталога lab06 и файла lab6-1.asm

Ввожу в файл lab6-1.asm текст программы из листинга 6.1 (рис. 2.2).

```
GNU nano 7.2 /home/taponomareva/work/arch-pc/lab06/lab6-1.asm Modified
#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. 2.2: Окно Midnight Commander. Содержание файла lab6-1.asm

Создаю исполняемый файл lab6-1 и запускаю его (рис. 2.3).

```
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ./lab6-1
j
```

Рис. 2.3: Терминал. Создание исполняемого файла lab6-1 и его запуск

Далее изменяю текст программы и вместо символов, пишу в регистры числа (рис. 2.4).



```
GNU nano 7.2 /home/taponomareva/work/arch-pc/lab06/lab6-1.asm Modified
#include 'in_out.asm'

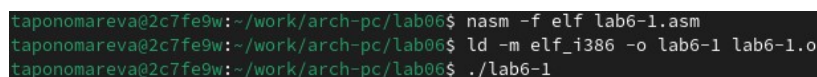
SECTION .bss
buf1: RESB 80

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. 2.4: Окно Midnight Commander. Содержание файла lab6-1.asm

Создаю исполняемый файл lab6-1 и запускаю его (рис. 2.5).



```
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ./lab6-1
```

Рис. 2.5: Терминал. Создание исполняемого файла lab6-1 и его запуск

Согласно таблице ASCII, код 10 соответствует символу перевода строки (newline), который обозначается как LF (Line Feed). Символ с кодом 10 (перевод строки) не отображается визуально при выводе на экран. Вместо этого он вызывает действие — переход на новую строку.

Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 при помощи команды touch ~/work/arch-pc/lab06/lab6-2.asm (рис. 2.6).



```
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
```

Рис. 2.6: Терминал. Создание файла lab6-2.asm

Ввожу в него текст программы из листинга 6.2 (рис. 2.7).

```

GNU nano 7.2 /home/taonomareva/work/arch-pc/lab06/lab6-2.asm Modified
#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: Окно Midnight Commander. Содержание файла lab6-2.asm

Создаю исполняемый файл lab6-2 и запускаю его (рис. 2.8).

```

taonomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ./lab6-2
106

```

Рис. 2.8: Терминал. Создание исполняемого файла lab6-2 и его запуск

Аналогично предыдущему примеру изменяю символы на числа (рис. 2.9).

```

GNU nano 7.2 /home/taonomareva/work/arch-pc/lab06/lab6-2.asm Modified
#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: Окно Midnight Commander. Содержание файла lab6-2.asm

Создаю исполняемый файл lab6-2 и запускаю его. В терминале выводится 10 (рис. 2.10).

```

taonomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ./lab6-2
10

```

Рис. 2.10: Терминал. Создание исполняемого файла lab6-2 и его запуск

Заменяю функцию iprintLF на iprint (рис. 2.11).


```

GNU nano 7.2 /home/taonomareva/work/arch-pc/lab06/lab6-2.asm
#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: Окно Midnight Commander. Содержание файла lab6-2.asm

Создаю исполняемый файл lab6-2 и запускаю его. `iprintLF` добавляет перевод строки после числа, а `iprint` выводит число без переноса строки (рис. 2.12).

```

taonomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ./lab6-2
10taonomareva@2c7fe9w:~/work/arch-pc/lab06$

```

Рис. 2.12: Терминал. Создание исполняемого файла lab6-2 и его запуск

2.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm в каталоге `~/work/arch-pc/lab06` при помощи команды `touch ~/work/arch-pc/lab06/lab6-3.asm` (рис. 2.13).

```

taonomareva@2c7fe9w:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm

```

Рис. 2.13: Терминал. Создание файла lab6-3.asm

Ввожу в lab6-3.asm текст программы из листинга 6.3 (рис. 2.14).

```

GNU nano 7.2 /home/taonomareva/work/arch-pc/lab06/lab6-3.asm Modified
#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. 2.14: Окно Midnight Commander. Содержание файла lab6-3.asm

Создаю исполняемый файл lab6-3 и запускаю его (рис. 2.15).

```

taonomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
taonomareva@2c7fe9w:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 2.15: Терминал. Создание исполняемого файла lab6-3 и его запуск

Изменяю текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (рис. 2.16).

```

GNU nano 7.2 /home/taonomareva/work/arch-pc/lab06/lab6-3.asm Modified
#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=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
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. 2.16: Окно Midnight Commander. Содержание файла lab6-3.asm

Создаю исполняемый файл lab6-3 и проверяю его работу (рис. 2.17).

```
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 2.17: Терминал. Создание исполняемого файла lab6-3 и его запуск

Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 2.18).

```
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
```

Рис. 2.18: Терминал. Создание файла variant.asm

Ввожу в variant.asm текст программы из листинга 6.4 (рис. 2.19).

```
GNU nano 7.2 /home/taponomareva/work/arch-pc/lab06/variant.asm Modified
;-----
; Программа вычисления варианта
;-----
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
r: 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 iprintf
call quit
```

Рис. 2.19: Окно Midnight Commander. Содержание файла variant.asm

Создаю исполняемый файл variant и запускаю его. Ввожу свой номер студенческого билета - 1132246742. Получаю вариант 3 (рис. 2.20).

```
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ nasm -f elf variant.asm
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
taponomareva@2c7fe9w:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246742
Ваш вариант: 3
```

Рис. 2.20: Терминал. Создание исполняемого файла variant и его запуск

Программа делит номер студенческого билета на 20, сохраняет остаток в регистре `edx`, а затем увеличивает его на 1 с помощью инструкции `inc edx`, что приводит к тому, что результат деления с остатком дает вариант 3.

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

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

Эти строки загружают строку “Ваш вариант:” в регистр `eax` и затем вызывают подпрограмму `sprint`, которая отвечает за вывод строки на экран.

2)

```
mov ecx, x  
mov edx, 80  
call sread
```

Эти инструкции загружают адрес переменной для ввода данных и максимальную длину строки, затем вызывают подпрограмму `sread` для чтения ввода с клавиатуры.

3) Инструкция `call atoi` вызывается для преобразования строки, содержащей ASCII-коды, в целое число, которое сохраняется в регистре `eax`.

4) За вычисления варианта отвечают следующие строки:

```
mov eax,x           ; Загружаем число из переменной x в регистр eax  
call atoi           ; Преобразуем строку в число (номер студенческого билета)  
xor edx,edx         ; Обнуляем регистр edx, чтобы он использовался для остатка от деления  
mov ebx,20          ; Загружаем делитель (20) в регистр ebx  
div ebx             ; Делим eax на ebx, результат в eax, остаток в edx  
inc edx             ; Увеличиваем остаток на 1, получаем вариант
```

- 5) Остаток от деления при выполнении инструкции `div ebx` записывается в регистр `edx`.
- 6) Инструкция `inc edx` используется для увеличения остатка от деления в регистре `edx` на 1.
- 7) За вывод на экран результата вычислений отвечают следующие строки:

```
mov eax,rem      ; Загружаем адрес строки 'Ваш вариант:' в регистр eax
call sprint      ; Выводим строку "Ваш вариант:"
mov eax,edx      ; Загружаем результат (вариант) из регистра edx в eax
call iprintLF    ; Выводим значение из eax (номер варианта)
```

3 Задание для самостоятельной работы

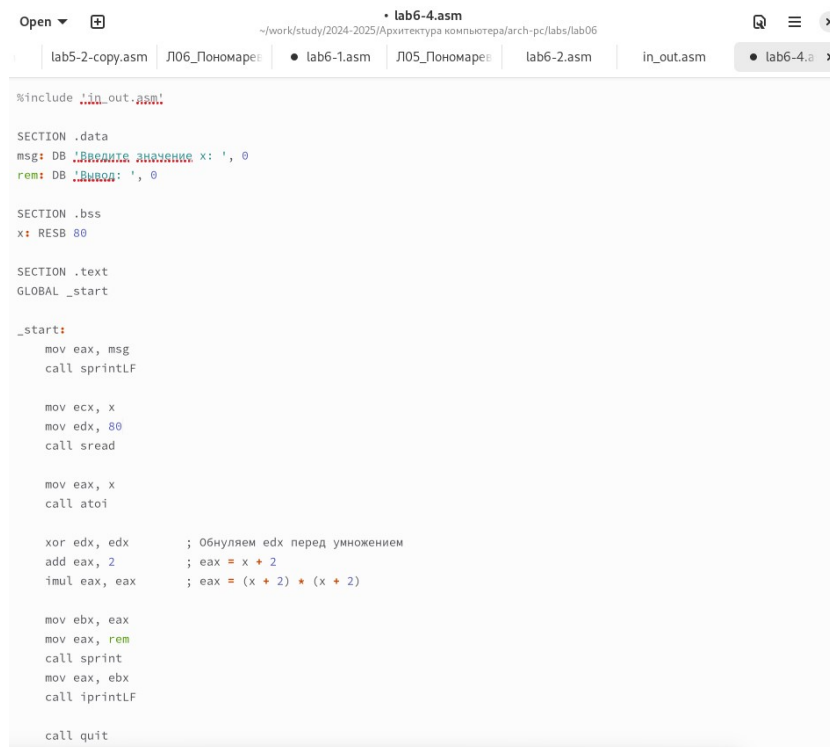
Создаю файл lab6-4.asm в каталоге ~/work/arch-pc/lab06 при помощи touch ~/work/arch-pc/lab06/lab6-4.asm (рис. 3.1).

A terminal window with a dark background. The prompt is 'caronomaeva@2c7fe9w:~/work/arch-pc/lab06\$'. The command entered is 'touch ~/work/arch-pc/lab06/lab6-4.asm'.

```
caronomaeva@2c7fe9w:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-4.asm
```

Рис. 3.1: Терминал. Создание файла lab6-4.asm

Вписываю текст программы в файл lab6-4.asm для вычисления выражения $y=f(x)$. У меня вариант 3, т.е. $y=(2+x)^2$, $x_1=2$, $x_2=8$ (рис. 3.2).



```
%include 'in_out.asm'

SECTION .data
msg: DB "Введите значение x: ", 0
rem: DB "Вывод: ", 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

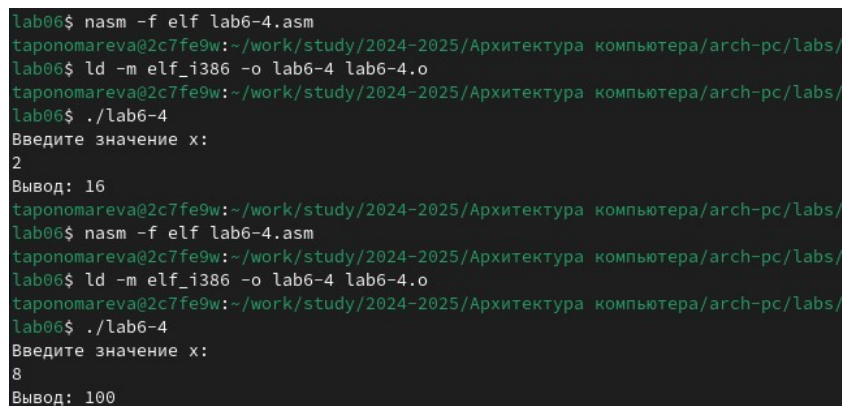
    xor edx, edx        ; Обнуляем edx перед умножением
    add eax, 2          ; eax = x + 2
    imul eax, eax       ; eax = (x + 2) * (x + 2)

    mov ebx, eax
    mov eax, rem
    call sprintf
    mov eax, ebx
    call iprintf

    call quit
```

Рис. 3.2: Окно Text Editor. Содержание файла lab6-4.asm

Ответ. При $x_1=2$, имеем 16, при $x_2=8$, имеем 100 (рис. 3.3).



```
lab06$ nasm -f elf lab6-4.asm
taronomareva@2c7fe9w:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
taronomareva@2c7fe9w:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab06$ ./lab6-4
Введите значение x:
2
Вывод: 16
taronomareva@2c7fe9w:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab06$ nasm -f elf lab6-4.asm
taronomareva@2c7fe9w:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
taronomareva@2c7fe9w:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/
lab06$ ./lab6-4
Введите значение x:
8
Вывод: 100
```

Рис. 3.3: Терминал. Проверка вычисления программы варианта 3.

Загружаю на GitHub.

4 Выводы

Были освоены арифметические инструкции языка ассемблера NASM.

Список литературы

1. Курс на ТУИС
2. Лабораторная работа №6