

Отчёт по лабораторной работе №6

Дисциплина: Архитектура Компьютера

Иванов Сергей Владимирович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	16

Список иллюстраций

2.1	Создание lab6-1.asm	5
2.2	Текст из листинга 6.1	5
2.3	Исполняемый файл	6
2.4	Изменяем текст программы	6
2.5	Запуск исполняемого файла	7
2.6	Текст из листинга 6.2	7
2.7	Запуск исполняемого файла	8
2.8	Редактируем программу	8
2.9	Замена функции	8
2.10	Файл lab6-3.asm	9
2.11	Запускаем lab6-3	9
2.12	Изменение текста программы	10
2.13	Запускаем отредактированную программу	10
2.14	Файл variant.asm	11
2.15	Запуск variant.asm	11
2.16	Создаем lab6-4.asm	13
2.17	Проверка lab6-4.asm	14

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 6, перейдем в него и создадим файл lab6-1.asm: (Рис. 2.1)

```
svivanov1@svivanov1:~$ mkdir ~/work/arch-pc/lab06
svivanov1@svivanov1:~$ cd ~/work/arch-pc/lab06
svivanov1@svivanov1:~/work/arch-pc/lab06$ touch lab6-1.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$
```

Рис. 2.1: Создание lab6-1.asm

Введем в файл lab6-1.asm текст программы из листинга 6.1 (Рис. 2.2)

```
GNU nano 7.2 /home/svivanov1/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. 2.2: Текст из листинга 6.1

Создадим исполняемый файл и запустим его. (Рис. 2.3)

```
svivanov1@svivanov1:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-1
j
svivanov1@svivanov1:~/work/arch-pc/lab06$
```

Рис. 2.3: Исполняемый файл

Изменим текст программы и вместо символов запишем в регистры числа. (Рис. 2.4)

```
GNU nano 7.2 /home/svivanov
#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: Изменяем текст программы

Создадим исполняемый файл и запустим его. (Рис. 2.5)

```
svivanov1@svivanov1:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-1

svivanov1@svivanov1:~/work/arch-pc/lab06$
```

Рис. 2.5: Запуск исполняемого файла

Пользуясь таблицей ASCII определяем, что это символ LF, перевод строки, он не отображается при выводе на экран.

Создадим файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введем в него текст программы из листинга 6.2 (Рис. 2.6)

```
GNU nano 7.2 /home/svivanov
%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: Текст из листинга 6.2

Создадим исполняемый файл и запустим его. (Рис. 2.7)

```
svivanov1@svivanov1:~$ touch ~/work/arch-pc/lab06/lab6-2.asm
svivanov1@svivanov1:~$ mc

svivanov1@svivanov1:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-2
106
svivanov1@svivanov1:~/work/arch-pc/lab06$
```

Рис. 2.7: Запуск исполняемого файла

Аналогично предыдущему примеру изменим символы на числа. Создадим исполняемый файл и запустим его. Получаем число 10. (Рис. 2.8)

```
svivanov1@svivanov1:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-2
10
svivanov1@svivanov1:~/work/arch-pc/lab06$
```

Рис. 2.8: Редактируем программу

Заменяем функцию `iprintLF` на `iprint`. Создадим исполняемый файл и запустим его. (Рис. 2.9)

```
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-2
10svivanov1@svivanov1:~/work/arch-pc/lab06$ █
```

Рис. 2.9: Замена функции

Отличие функций в том, что во втором случае строка не переносится.

Создадим файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06` и введем в него текст программы из листинга 6.3 (Рис. 2.10)


```

GNU nano 7.2 /home/svivanov1/work/arch-pc/lab06/lab6-3.asm
%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
[ Прочитано 26 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^_ К строке

```

Рис. 2.10: Файл lab6-3.asm

Создадим исполняемый файл и запустим его. (Рис. 2.11)

```

svivanov1@svivanov1:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
svivanov1@svivanov1:~/work/arch-pc/lab06$

```

Рис. 2.11: Запускаем lab6-3

Изменим текст программы для вычисления выражения $f(x) = (4*6+2)/5$. (Рис. 2.12)

```

#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/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

```

Рис. 2.12: Изменение текста программы

Создадим исполняемый файл и запустим его. (Рис. 2.13)

```

svivanov1@svivanov1:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 2.13: Запускаем отредактированную программу

Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06 и введем в него текст программы из листинга 6.4 (Рис. 2.14)

```

%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 ; ASCII кода в число, `eax=x`
xor edx,edx
mov ebx,20
div ebx
inc edx

```

Рис. 2.14: Файл variant.asm

Создадим исполняемый файл и запустим его. (Рис. 2.15)

```

svivanov1@svivanov1:~/work/arch-pc/lab06$ nasm -f elf variant.asm
svivanov1@svivanov1:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132236127
Ваш вариант: 8
svivanov1@svivanov1:~/work/arch-pc/lab06$

```

Рис. 2.15: Запуск variant.asm

Решив аналитически получаем: $1132236127/20 = 56611806$, остаток 7. Прибавляем +1, получаем 8. Ответ сошелся.

Ответы на вопросы:

- 1) `mov eax,rem`
`call sprint`
- 2) Для считывания переменной x, которую вводит пользователь
- 3) Для перевода ASCII в число

4) xor edx,edx

mov ebx,20

div ebx

inc edx

5) В регистр edx

6) Увеличивает значение edx на 1

7) mov eax,edx

call iprintLF

Создадим файл lab6-4.asm. Напишем программу из таблицы 6.3 для варианта
8. $((11+x)*2-6)$ (Рис. 2.16)

```

#include 'in_out.asm' ;
section .data
    prompt: DB 'Введите значение x: ', 0
    msg: DB 'Выражение (11+x)*2-6: ', 0
    div: DB 'Результат: ', 0
section .bss
    x: RESB 80
section .text
global _start
_start:
    mov eax, msg
    call sprintf
    mov eax, prompt
    call sprintf
    mov ecx, x
    mov edx, 80
    call sread
    mov eax, x
    call atoi
    mov ebx, 2
    add eax, 11
    mul ebx
    sub eax, 6
    call iprintLF
    call quit

```

Рис. 2.16: Создаем lab6-4.asm

Создадим исполняемый файл и запустим его. Проверим работу программы для $x=1$ и $x=9$. (Рис. 2.16)

```

svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-4
Выражение (11+x)*2-6:
Введите значение x: 1
18
svivanov1@svivanov1:~/work/arch-pc/lab06$ ./lab6-4
Выражение (11+x)*2-6:
Введите значение x: 9
34
svivanov1@svivanov1:~/work/arch-pc/lab06$

```

Рис. 2.17: Проверка lab6-4.asm

Код программы lab6-4

```

#include 'in_out.asm' ;

section .data
    prompt: DB 'Введите значение x: ', 0
    msg: DB 'Выражение (11+x)*2-6: ', 0
    div: DB 'Результат: ', 0

section .bss
    x: RESB 80

section .text
global _start
_start:
mov eax, msg
call sprintf
mov eax, prompt
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov ebx, 2

```

```
add eax, 11
mul ebx
sub eax, 6
call iprintLF
call quit
```

3 Выводы

В результате выполнения лабораторной работы мы освоили арифметические инструкции языка ассемблера NASM.