

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

Дисциплина: Архитектура компьютера

Скрипникова София Дмитриевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Выполнение арифметических операций в NASM.	15
4.2	Ответы на вопросы	18
4.3	Задания для самостоятельной работы	19
5	Выводы	22
	Список литературы	23

Список иллюстраций

4.1	Создание, переход в lab07	8
4.2	Ввела нужный текст	9
4.3	Вывела результат	9
4.4	Изменения текста	11
4.5	Результат изменений	12
4.6	Ввела текст программы	12
4.7	Результат программы	13
4.8	Ввела изменения в текст	13
4.9	Результат изменений	13
4.10	Замена функции	14
4.11	Вывод результата	14
4.12	Введение текста	15
4.13	Результат программы	16
4.14	Изменение текста	16
4.15	Результат вычислений	16
4.16	Введенный текст	17
4.17	Результат программы	18
4.18	Текст программы	19
4.19	Текст программы	20
4.20	Результат	20

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Освоить арифметические инструкции языка ассемблера NASM.

3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти.

ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом.

Для выполнения лабораторных работ в файле `in_out.asm` реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Это:

- `iprint` – вывод на экран чисел в формате ASCII, перед вызовом `iprint` в регистр `eax` необходимо записать выводимое число (`mov eax,`).
- `iprintLF` – работает аналогично `iprint`, но при выводе на экран после числа добавляет к символ перевода строки.
- `atoi` – функция преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`, перед вызовом `atoi` в регистр `eax` необходимо записать число (`mov eax,`).

4 Выполнение лабораторной работы

1. Создала каталог для программ лабораторной работы №7, перешла в него и создала файл lab7-1.asm. (рис. 4.1)

```
[sdscripnikova@fedora ~]$ mkdir ~/work/arch-pc/lab07  
[sdscripnikova@fedora ~]$ cd ~/work/arch-pc/lab07  
[sdscripnikova@fedora lab07]$ touch lab7-1.asm
```

Рис. 4.1: Создание, переход в lab07

2. Ввела в файл lab7-1 нужный текст программы из листинга 7.1., создала исполняемый файл и вывела результат. (рис. 4.2; рис. 4.3)


```

%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. 4.2: Ввела нужный текст

```

[sdskripnikova@fedora lab07]$ nasm -f elf lab7-1.asm
[sdskripnikova@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[sdskripnikova@fedora lab07]$ ./lab7-1
j

```

Рис. 4.3: Вывела результат

3. Изменила текст программы и вместо символов, записала в регистры числа.

(рис. 4.4; рис. 4.5)

```
%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 sprintf

call quit
```

Рис. 4.4: Изменения текста

```
[sdscripnikova@fedora lab07]$ nasm -f elf lab7-1.asm
[sdscripnikova@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[sdscripnikova@fedora lab07]$ ./lab7-1

[sdscripnikova@fedora lab07]$
```

Рис. 4.5: Результат изменений

Этот символ не отображается при выводе на экран.

4. Создала файл lab7-2.asm в нужном каталоге, ввела в него текст программы из листинга 7.2. и вывела результат. (рис. [рис. 4.6; рис. 4.7])

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 4.6: Ввела текст программы

```
[sdscripnikova@fedora lab07]$ nasm -f elf lab7-2.asm
[sdscripnikova@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[sdscripnikova@fedora lab07]$ ./lab7-2
106
[sdscripnikova@fedora lab07]$
```

Рис. 4.7: Результат программы

5. Изменила символы на числа, создала исполняемый файл и вывела результат.
(рис. 4.8; рис. 4.9)

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 4.8: Ввела изменения в текст

```
[sdscripnikova@fedora lab07]$ nasm -f elf lab7-2.asm
[sdscripnikova@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[sdscripnikova@fedora lab07]$ ./lab7-2
10
```

Рис. 4.9: Результат изменений

6. Заменяла функцию `iprintLF` на `iprint` и вывела результат. (рис. 4.10; рис. 4.11)

```
%include 'in_out.asm'  
SECTION .text  
GLOBAL _start  
_start:  
mov eax,6  
mov ebx,4  
add eax,ebx  
call iprint  
call quit
```

Рис. 4.10: Замена функции

```
[sdscripnikova@fedora lab07]$ nasm -f elf lab7-2.asm  
[sdscripnikova@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o  
[sdscripnikova@fedora lab07]$ ./lab7-2  
10[sdscripnikova@fedora lab07]$
```

Рис. 4.11: Вывод результата

Вывод функции `iprintLF` и `iprint` отличаются выводом результата, в одном варианте результат выводиться на отдельной строчке, а в другом нет.

4.1 Выполнение арифметических операций в NASM.

1. Создала файл lab7-3.asm в нужном каталоге, ввела нужный текст и вывела результат. (рис. 4.12; рис. 4.13)

```
%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
; ---- Вывод результата на экран
mov eax,div
call sprint
mov eax,edi
call iprintLF
mov eax,rem
call sprint
```

Рис. 4.12: Введение текста

```
[sdscripnikova@fedora lab07]$ nasm -f elf lab7-3.asm
[sdscripnikova@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[sdscripnikova@fedora lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[sdscripnikova@fedora lab07]$
```

Рис. 4.13: Результат программы

2. Изменила текст программы для вычисления выражения $f(x)=(4*6+2)/5$. (рис. 4.14; рис. 4.15)

```
%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
; ---- Вывод результата на экран
mov eax,div
call sprint
mov eax,edi
call iprintLF
mov eax,rem
call sprint
```

Рис. 4.14: Изменение текста

```
[sdscripnikova@fedora lab07]$ nasm -f elf lab7-3.asm
[sdscripnikova@fedora lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[sdscripnikova@fedora lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[sdscripnikova@fedora lab07]$
```

Рис. 4.15: Результат вычислений

3. Изучила текст программы из листинга 7.4., ввела его в нужный файл и вывела результат. (рис. 4.16; рис. 4.17)

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите No студенческого билета'
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 преобразо
call atoi ; ASCII кода в число, `eax=x`
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
```

Рис. 4.16: Введенный текст

```
[sdscripnikova@fedora lab07]$ nasm -f elf variant.asm
[sdscripnikova@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[sdscripnikova@fedora lab07]$ ./variant
Введите No студенческого билета:
1132226523
Ваш вариант: 4
[sdscripnikova@fedora lab07]$
```

Рис. 4.17: Результат программы

4.2 Ответы на вопросы

1. Какие строки листинга 7.4. отвечают за вывод на экран сообщения ‘Ваш вариант:’?

```
mov eax, msg call sprintfLF
```

2. Для чего используются следующие инструкции? `nasm mov ecx, x mov edx, 80 call sread`

Эти инструкции используются для ввода переменной X с клавиатуры и сохранения введенных данных.

3. Для чего используется инструкция “call atoi”?

Эта инструкция используется для преобразования кода переменной ASCII в число.

4. Какие строки листинга 7.4. отвечают за вычисления варианта?

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

5. В какой регистр записывается остаток от деления при выполнении инструкции “div edx”?

В регистре ebx.

6. Для чего используется инструкция “inc edx”?

Для увеличения значения edx на 1.

7. Какие строки листинга 7.4. отвечают за вывод на экран результата вычислений?

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

4.3 Задания для самостоятельной работы

1. Я написала программу, которая будет решать выражение и выводить ответ при введенных X. Для решения мне попало уравнение из 4 варианта:
 $f(x) = 4/3(x-1) + 5$. (рис. 4.18; рис. 4.19; рис. 4.20)

```
%include 'in_out.asm'
SECTION .data
rem: DB 'Вычислить значение выражения 4/3(x-1)+5', 0
msg: DB 'Введите x: ', 0
div: DB 'Результат: ', 0

SECTION .bss

x:      RESB 80
rez:     RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, rem
call sprintLF

mov eax, msg
call sprintLF

mov ecx, x
mov edx, 80
```

Рис. 4.18: Текст программы

```

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

mov ebx, 1
sub eax, ebx
mov ebx, 4
mul ebx
mov ebx, 3
div ebx
add eax, 5

mov[rez], eax

mov eax, div
call sprintLF
mov eax, [rez]
call iprintLF

call quit

```

Рис. 4.19: Текст программы

```

[sdskripnikova@fedora lab07]$ nasm -f elf sr07.asm
[sdskripnikova@fedora lab07]$ ld -m elf_i386 -o sr07 sr07.o
[sdskripnikova@fedora lab07]$ ./sr07
Вычислить значение выражения 4/3(x-1)+5
Введите x:
4
Результат:
9
[sdskripnikova@fedora lab07]$ ./sr07
Вычислить значение выражения 4/3(x-1)+5
Введите x:
10
Результат:
17
[sdskripnikova@fedora lab07]$

```

Рис. 4.20: Результат

Данные изменения можно проверить по ссылке: <https://github.com/sdskripniko>

va/study_2022-2023_arh-pc/tree/master/labs/lab07

5 Выводы

Освоила арифметические инструкции языка ассемблера NASM.

Список литературы

1. https://esystem.rudn.ru/pluginfile.php/1584388/mod_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%967.pdf