

Лабораторная работа № 6

Дисциплина: Архитектура компьютера

Дедова Виктория Сергеевна

Contents

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13

Список таблиц

Список иллюстраций

2.1 1 6

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM. |

2 Выполнение лабораторной работы

Создайте каталог для программ лабораторной работы No 7, перейдите в него и создайте файл lab7-1.asm:

```
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ mkdir lab07
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ cd lab07
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ touch lab7-1.asm
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ls
lab7-1.asm
```

Рис. 2.1: 1

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. Вводим в файл

```
GNU nano 6.2 /home/v
#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
```

lab7-1.asm текст программы из листинга 7.1

Создаём исполняемый файл и запускаем его.

```
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера
```

```
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-1
j
```

UP -- DIR	дек 23 17:43
in_out.asm	3942 дек 23 18:12
*lab7-1	5160 дек 23 18:14
lab7-1.asm	173 дек 23 17:49
lab7-1.o	1200 дек 23 18:14

```
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab7-1.asm
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-1
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$
```

Далее изменим текст программы и вместо символов, запишем в реги- стры

```
%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
```

числа, mov eax, '6' mov ebx, '4' на строки mov eax,6 mov ebx,4

```
%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
```

Код 10 соответствует символу переходу строки.

Создаём файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и вводим в него текст

.n	Name	Size
1	..	UP - -DIP
2	in_out.asm	3942
3	*lab7-1	5160
4	lab7-1.asm	169
5	lab7-1.o	1200
6	lab7-2.asm	0

программы из листинга 7.2.

В результате работы программы получаем 106.

Замените строки mov eax,'6' mov ebx,'4' на строки mov eax,6 mov ebx,4 Создаём

исполняемый файл и запускаем его.

Результат будет 10. Заменяем функцию iprintLF на iprint. Создаёте исполняемый


```

GNU nano 6.2 /
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit

```

```

vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3
10vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3

```

файл и запустите его.

Вывод функций отличается переходом на другую строку.

Приведем программу вычисления арифметического выражения $(5 \times (2 + 3))/3$. Создаём файл lab7-3.asm в каталоге ~/work/arch-pc/lab07 и вводим текст

.n	Name	Size	Modify time
./..		UP - -DIR	дек 23 17:43
	in_out.asm	3942	дек 23 18:12
*lab7-1		5160	дек 23 18:39
	lab7-1.asm	169	дек 23 18:19
	lab7-1.o	1200	дек 23 18:34
*lab7-2		5088	дек 23 18:50
	lab7-2.asm	112	дек 23 18:50
	lab7-2.o	1040	дек 23 18:50
	lab7-3.asm	0	дек 23 18:53

листинга 7.3.бщ

```

vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ nasm -f elf lab7-3.asm
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3
Результат: 4
Остаток от деления: 1

```

Измените текст программы для вычисления выражения $(4 \times (6 + 2))/5$.

```

vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./lab7-3
Результат: 5
Остаток от деления: 1

```

Создаём исполняемый файл и проверяем его работу.

```
./..
*Dedova-1
*Dedova.asm
Dedova.o
in_out.asm
*lab7-1
lab7-1.asm
lab7-1.o
*lab7-2
lab7-2.asm
lab7-2.o
*lab7-3
lab7-3.asm
lab7-3.o
*variant.asm
variant.o
```

Создаём файл variant.asm. вводим текст из листинга 7.4.

```
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ nasm -f elf variant.asm
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ld -m elf_i386 -o variant.asm variant.o
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07$ ./variant.asm
Введите No студенческого билета:
113222832
Ваш вариант: 13
```

Вопросы: 1. mov eax and rem call sprint 2. mov ecx,x-запись входной переменной в регистр ecx; mov edx,80 -запись размера переменной в регистре edx; call sread-вызов процедуры чтения данных. 3.call atoi - функция преобразующая ASCII код символа в целое число и записывающая результат в регистр eax 4.xor edx,edx mov ebx,20 div ebx, inc edx. 5.div ebx-ebx 6.inc- используется для увеличения операнда на единицу 7.mov eax,rem call sprint mov eax,edx call iprintLV

#Задание для самостоятельной работы

Напишем программу вычисления выражения $\Box = \Box(\Box)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения \Box , вычислять заданное выражение в зависимости от введенного \Box , выводить результат вычислений. Вид функции $\Box(\Box)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создаём исполняемый файл и проверьте его работу для

```

GNU nano 6.2 /home/vsdedov
#include 'in_out.asm' ; подключение внешнего
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Введите переменную x= : ',0
rem1: DB 'x= : ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax, rem
call sprintLF
mov eax, rem1
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax,x
call atoi
mov ebx,8
mul ebx
sub eax,6
xor edx,edx
mov ebx,10
mul ebx
mov edi,eax
mov eax,div
call sprint
mov eax,edi
call iprintLF
call quit

```

значений x_1 и x_2 из 6.3

```

vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-p
c/lab07$ nasm -f elf Dedova.asm
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-p
c/lab07$ ld -m elf_i386 -o Dedova-1 Dedova.o
vsdedova@brenker-NBLK-WAX9X:~/work/study/2022-2023/Архитектура компьютера/arch-p
c/lab07$ ./Dedova-1
Введите переменную x= :
x= : 5
Результат: 340

```

Left	File	Command	Options
<-	... компьютера/arch-pc/lab07	-.[^]>	
.n	Name	Size	Modify time
/..		UP - -DIR	дек 23 17:43
*Dedova-1		9168	дек 24 11:55
*Dedova.asm		541	дек 24 11:53
Dedova.o		1472	дек 24 11:54
in_out.asm		3942	дек 23 18:12
*lab7-1		5160	дек 23 18:35
lab7-1.asm		169	дек 23 18:19
lab7-1.o		1200	дек 23 18:34
*lab7-2		5088	дек 23 18:50
lab7-2.asm		112	дек 23 18:50
lab7-2.o		1040	дек 23 18:50
*lab7-3		9072	дек 23 19:03
lab7-3.asm		1366	дек 23 19:03
lab7-3.o		1328	дек 23 19:03
*variant.asm		9160	дек 23 19:07
variant.o		1440	дек 23 19:06

3 Выводы

На данной лабораторной работе мы освоили арифметические инструкции языка ассемблера NASM