

# **Лабораторная работа №6.**

**Арифметические операции в NASM.**

Бурыкина Софья Дмитриевна

# Содержание

|   |                                |    |
|---|--------------------------------|----|
| 1 | Цель работы                    | 5  |
| 2 | Теоретическое введение         | 6  |
| 3 | Выполнение лабораторной работы | 7  |
| 4 | Ответы на вопросы              | 13 |
| 5 | Самостоятельная работа         | 14 |
| 6 | Выводы                         | 15 |
|   | Список литературы              | 16 |

## Список иллюстраций

|      |                               |    |
|------|-------------------------------|----|
| 3.1  | Создания файла . . . . .      | 7  |
| 3.2  | Ввела программу . . . . .     | 8  |
| 3.3  | Запуск программы . . . . .    | 8  |
| 3.4  | Исправила текст . . . . .     | 9  |
| 3.5  | Запустила программу . . . . . | 9  |
| 3.6  | Созданный файл . . . . .      | 9  |
| 3.7  | Запуск программы . . . . .    | 10 |
| 3.8  | Запуск программы . . . . .    | 10 |
| 3.9  | Результат программы . . . . . | 10 |
| 3.10 | Результат программы . . . . . | 10 |
| 3.11 | Создание файла . . . . .      | 11 |
| 3.12 | Файл с текстом . . . . .      | 11 |
| 3.13 | Результат работы . . . . .    | 11 |
| 3.14 | Результат работы . . . . .    | 12 |
| 3.15 | Результат работы . . . . .    | 12 |
| 5.1  | Результат работы . . . . .    | 14 |
| 5.2  | Результат работы . . . . .    | 14 |

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - приобретения практических навыков арифметических инструкций языка ассемблера NASM.

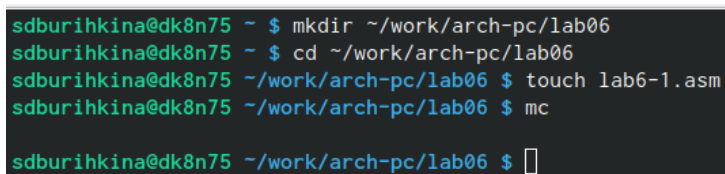
## 2 Теоретическое введение

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. Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации. Существует три основных способа адресации: • Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: `mov ax,bx`. • Непосредственная адресация – значение операнда задается непосредственно в команде, Например: `mov ax,2`. • Адресация памяти – операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию. Например, определим переменную `intg DD 3` – это означает, что задается область памяти размером 4 байта, адрес которой обозначен меткой `intg`. В таком случае, команда `mov eax,intg` копирует из памяти по адресу `intg` данные в регистр `eax`. В свою очередь команда `mov intg,eax` запишет в память по адресу `intg` данные из регистра `eax`. Также рассмотрим команду `mov eax,intg`

Более подробно об Unix см. в [1–6].

### 3 Выполнение лабораторной работы

Создала каталог для программ лабораторной работы No 6, перешла в него и создала файл lab6-1.asm (рис. 3.1).



```
sdburikhina@dk8n75 ~ $ mkdir ~/work/arch-pc/lab06
sdburikhina@dk8n75 ~ $ cd ~/work/arch-pc/lab06
sdburikhina@dk8n75 ~/work/arch-pc/lab06 $ touch lab6-1.asm
sdburikhina@dk8n75 ~/work/arch-pc/lab06 $ mc

sdburikhina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.1: Создания файла

Создала исполняемый файл (рис. 3.2).

```
lab6-1.asm [----] 13 L: [ 1+11 12/ 14] *(162 / 173b) 0010 0x00A
#include "lab6-1.asm"
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, '1'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintf
call quit
```

Рис. 3.2: Ввела программу

Создала исполняемый файл запустила его (рис. 3.3).

```
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ./lab6-1
j
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.3: Запуск программы

Изменила текст программы (рис. 3.4).



```
lab6-1.asm [-M--] 9 L: [ 1+13 14/ 14] *(169 / 169b) <EOF>
#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. 3.4: Исправила текст

Создала исполняемый файл после изменения и запустила его (рис. 3.5).

Ответ на вопрос: нет не отображается ничего

```
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ./lab6-1

sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.5: Запустила программу

Создала файл lab6-2.asm в каталоге (рис. 3.6).

| Имя        | Размер  | Время правки | Имя           |
|------------|---------|--------------|---------------|
| ..         | -BBERX- | ноя 15 12:19 | ..            |
| in_out.asm | 1842    | ноя 15 12:22 | ..cache       |
| *lab6-1    | 1108    | ноя 15 12:34 | ..config      |
| lab6-1.asm | 189     | ноя 15 12:33 | ..git         |
| lab6-1.o   | 1200    | ноя 15 12:33 | ..gnupg       |
| lab6-2.asm | 0       | ноя 15 12:36 | ..gphoto      |
|            |         |              | ..icewm       |
|            |         |              | ..local       |
|            |         |              | ..mozilla     |
|            |         |              | ..ssh         |
|            |         |              | ..texlive2022 |
|            |         |              | ..texlive2023 |
|            |         |              | ..GNUstep     |
|            |         |              | ..public      |
|            |         |              | ..public_html |
|            |         |              | ..work        |
|            |         |              | ..Видео       |
|            |         |              | ..Документы   |

Рис. 3.6: Созданный файл

Создала исполняемый файл (рис. 3.7).

```
sdburikhkina@dk8n75 ~ $ cd ~/work/arch-pc/lab06
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ./lab6-2

106
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.7: Запуск программы

Изменила содержимое файла (рис. 3.8).

```
lab6-2.asm [----] 9 L: [ 1+ 5 6/ 10] *(77 / 114b) 0010 0x00A
#include "stdio.h"
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 3.8: Запуск программы

Полученный результат 10 при исполнении программы (рис. 3.9).

```
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ./lab6-2
10
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.9: Результат программы

Заменяла функцию `iprintLF` на `iprint`. Получила результат. Отличается выводом строки (рис. 3.10).

```
sdburikhkina@dk8n33 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
sdburikhkina@dk8n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
sdburikhkina@dk8n33 ~/work/arch-pc/lab06 $ ./lab6-2
10sdburikhkina@dk8n33 ~/work/arch-pc/lab06 $
```

Рис. 3.10: Результат программы

Создала файл lab6-3.asm в каталоге (рис. 3.11).

```
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.11: Создание файла

Ввела текст программы (рис. 3.12).

```
lab6-3.asm [----] 41 L: [ 1+29 30/ 30] *(1363/1363b) <EOF>
; Программа вычисления выражения
; =====
%include "lab06.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. 3.12: Файл с текстом

Создала исполняемый файл и запустила его (рис. 3.13).

```
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.13: Результат работы

Изменила текст программы для вычисления выражения (рис. 3.14).

```
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.14: Результат работы

Создала файл variant.asm в каталоге (рис. 3.15).

```
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant.asm variant.o
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $ ./variant.asm
Введите No студенческого билета:
1132231986
Ваш вариант: 7
sdburikhkina@dk8n75 ~/work/arch-pc/lab06 $
```

Рис. 3.15: Результат работы

## 4 Ответы на вопросы

1 Ответ: `sprint` выводит сообщения на экран, перед вызовом `sprint` в регистр `eax` необходимо записать выводимое сообщение (`mov eax,msg`). Также `sprintLF` работает аналогично `sprint`, но при выводе на экран добавляет к сообщению символ перевода строки. 2 Ответ: `mov ecx, x` запись адреса переменной 'EAX' `mov edx, 80` запись длины вводимого сообщения в 'EBX' `call sread` вызов подпрограммы ввода сообщения 3 Ответ: `call` - вызов, `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,int`). 4 Ответ: `call sread` вывод программы ввода сообщения. 5 Ответ: процессор поделит число, старшие биты которого хранит регистр `edx`, а младшие `eax` на значение, хранящееся в регистре `ebx`. 6 Ответ: для вызова прерывания с указанным номером такой как `edx`. 7 Ответ: `call iprintLF`; `call quit`

## 5 Самостоятельная работа

Создала исполняемый файл и проверила его работу Ответ x1 (рис. 5.1).

```
sdburikhina@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf variant7.asm
sdburikhina@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant7 variant7.o
sdburikhina@dk3n33 ~/work/arch-pc/lab06 $ ./variant7
Введите значение переменной x: 3
Результат: 20
```

Рис. 5.1: Результат работы

Ответ x2 (рис. 5.2).

```
sdburikhina@dk3n33 ~/work/arch-pc/lab06 $ nasm -f elf variant7.asm
sdburikhina@dk3n33 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant7 variant7.o
sdburikhina@dk3n33 ~/work/arch-pc/lab06 $ ./variant7
Введите значение переменной x: 5
Результат: 80
```

Рис. 5.2: Результат работы

## 6 Выводы

Приобрела практические знания арифметических инструкций языка ассемблера NASM.

## Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.  
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.