

Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Мария Данииловна Гольцова

Содержание

1	Цель работы	1
2	Выполнение лабораторной работы	1
3	Ответы на вопросы	6
4	Самостоятельная работа	7
5	Выводы	8
	Список литературы	8

1 Цель работы

Освоить арифметические инструкции языка ассемблера NASM.

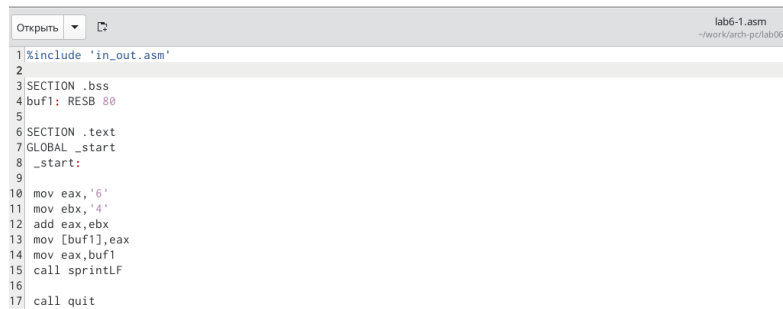
2 Выполнение лабораторной работы

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

```
mdgoljcova@dk5n60 ~ $ mkdir ~/work/arch-pc/lab06
mdgoljcova@dk5n60 ~ $ cd ~/work/arch-pc/lab06
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ touch lab6-1.asm
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $
```

Создание каталога и файла

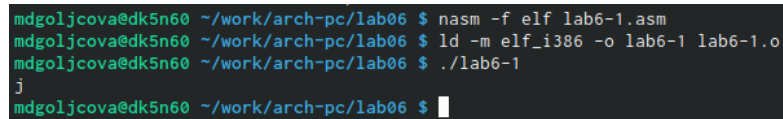
Ввела в файл lab6-1.asm текст программы из листинга 6.1 (рис. ??).



```
1 %include "in_out.asm"
2
3 SECTION .bss
4 buf1: RESB 80
5
6 SECTION .text
7 GLOBAL _start
8 _start:
9
10 mov eax, '6'
11 mov ebx, '4'
12 add eax, ebx
13 mov [buf1], eax
14 mov eax, buf1
15 call sprintLF
16
17 call quit
```

Ввод текста программы

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



```
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ./lab6-1
j
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $
```

Запуск файла

Далее изменила текст программы и вместо символов записала в регистры числа (рис. ??).

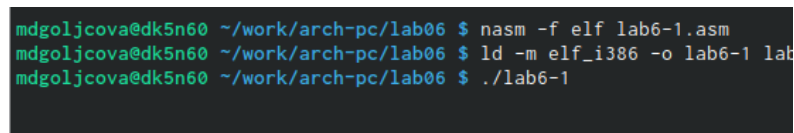


```
mov eax, 6
mov ebx, 4
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintLF

call quit
```

Изменение текста программы

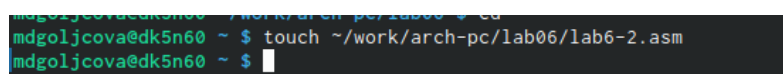
Создала исполняемый файл и запустила его (рис. ??).



```
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ./lab6-1
```

Запуск программы

Создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. ??).



```
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ cd
mdgoljcova@dk5n60 ~ $ touch ~/work/arch-pc/lab06/lab6-2.asm
mdgoljcova@dk5n60 ~ $
```

Создание файла

Ввела в него текст программы из листинга 6.2 (рис. ??).

```
lab6-2.asm [-M--] 1 L:[ 1+10 11/ 12] *(117 / 128b) 0010 0x
#include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Ввод текста программы

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

```
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $ ./lab6-2
106
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $
```

Запуск файла с программой

Аналогично предыдущему примеру изменила символы на числа (рис. ??).

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax, 6
mov ebx, 4
add eax, ebx
call iprintLF
call quit
```

Замена символов на числа

Создала исполняемый файл и запустила его. В результате было получено число 10 (рис. ??).

```
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $ ./lab6-2
10
mdgoljcovadk5n60 ~/work/arch-pc/lab06 $
```

Запуск изменённой программы

Заменяла функцию `iprintLF` на `iprint` (рис. ??).

```
lab6-2.asm      [----] 12 L:[ 1+ 9 10/ 12] *(109 / 122b) 0010 0x00A
%include "in_out.asm"

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
```

Замена функции `iprintLF` на `iprint`

В результате было выведено число 10 не на отдельной строке (рис. ??).

```
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ./lab6-2
10mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $
```

Результат функции `iprint`

Создала файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06` (рис. ??).

```
mdgoljcova@dk5n60 ~ $ touch ~/work/arch-pc/lab06/lab6-3.asm
mdgoljcova@dk5n60 ~ $
```

Создание файла в каталоге

Внимательно изучила текст программы из листинга 6.3 и ввела в `lab6-3.asm` (рис. ??).

```
lab6-3.asm      [-M--] 32 L:[ 1+ 0 1/ 29] *(32 /1365b) 0045 0x02D
;-----
; Программа вычисления выражения
;-----
%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
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перемена
```

Ввод программы

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

```
mdgoljcova@dk5n60 ~ $ cd ~/work/arch-pc/lab06
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $
```

Запуск файла

Изменила текст программы для вычисления выражения $\frac{4 * 6 + 2}{5}$ (рис. ??).

```
lab6-3.asm      [-M--]  9 L:[ 5+16  21/ 29] *(763 /1365b) 0105 0x069
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,0 ; 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
```

Изменённая программа

Создала исполняемый файл и проверила его работу. Получено: результат - 15, остаток от деления - 1 (рис. ??).

```
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $
```

Работа программы

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

```
mdgoljcova@dk5n60 ~/work/arch-pc/lab06 $ cd
mdgoljcova@dk5n60 ~ $ touch ~/work/arch-pc/lab06/variant.asm
mdgoljcova@dk5n60 ~ $
```

Создание файла

Внимательно изучила текст программы из листинга 6.4 и ввела в файл variant.asm (рис. ??).

```

variant.asm      [-M--] 33 L:[ 1+ 0 1/ 28] *(33 / 617b) 0010 0x00A
;-----
; Программа вычисления варианта
;-----
%include "in_out.asm"
SECTION .data
msg: DB "Введите No студенческого билета: ",0
rem: DB "Ваш вариант: ",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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
xor edx, edx
mov ebx, 20

```

Ввод текста программы

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

```

mdgoljcovae@dk5n60 ~ $ cd ~/work/arch-pc/lab06
mdgoljcovae@dk5n60 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
mdgoljcovae@dk5n60 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
mdgoljcovae@dk5n60 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1132236080
Ваш вариант: 1

```

Работа программы

3 Ответы на вопросы

1. За вывод сообщения "Ваш вариант" отвечают строки кода `mov eax, rem` `call sprint`
2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры.
3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`.
4. За вычисления варианта отвечают строки: `xor edx, edx`; обнуление `eax` для корректной работы `div` `mov ebx, 20`; `ebx = 20` `div ebx`; `eax = eax / 20`, `edx` - остаток от деления `inc edx`; `edx = edx + 1`
5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`.
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1.

7. За вывод на экран результатов вычислений отвечают строки: `mov eax,edx call iprintLF`

4 Самостоятельная работа

Создала файл `sr.asm` в каталоге `~/work/arch-pc/lab06` (рис. ??).

```
mdgoljcov@dk1n22 ~ $ touch ~/work/arch-pc/lab06/sr.asm
```

Создание файла

Написала программу вычисления выражения $\frac{10 + 2x}{3}$ (рис. ??).

```
GNU nano 7.2 /afs/.dk.sci.pfu.edu.ru/home/m/d/mdgoljcov/work/a
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите значение x: ',0
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, eax=x
mov ebx,2 ; EBX=2
mul ebx;EAX=EAX*EBX=x*2
add eax,10; eax=eax+10=x*2+10
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

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

[ Прочитано 36 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполни
```

Программа

Подставила значение $x1 = 1$ (рис. ??).

```
mdgoljcova@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf sr.asm
mdgoljcova@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o sr sr.o
mdgoljcova@dk1n22 ~/work/arch-pc/lab06 $ ./sr
Введите значение x:
1
Результат: 4
Остаток от деления: 0
```

$x1 = 1$

Подставила значение $x2 = 10$ (рис. ??).

```
mdgoljcova@dk1n22 ~/work/arch-pc/lab06 $ nasm -f elf sr.asm
mdgoljcova@dk1n22 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o sr sr.o
mdgoljcova@dk1n22 ~/work/arch-pc/lab06 $ ./sr
Введите значение x:
10
Результат: 10
Остаток от деления: 0
```

$x1 = 10$

5 Выводы

Я освоила арифметические инструкции языка ассемблера NASM.

Список литературы