

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Компьютерные и информационные науки

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Нечаева Кира

Группа: НКАбд-04-23

МОСКВА

2023__ г.

Содержание

1. Цель работы.....	3
2. Задание.....	4
3. Выполнение лабораторной работы.....	5
4. Выводы.....	15
5. Источники.....	16

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

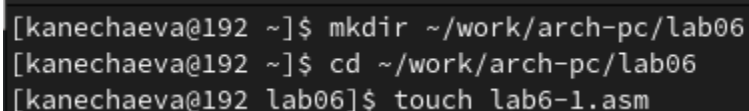
2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Ответы на вопросы
4. Задание для самостоятельной работы

3 Выполнение лабораторной работы

1. Символьные и численные данные в NASM

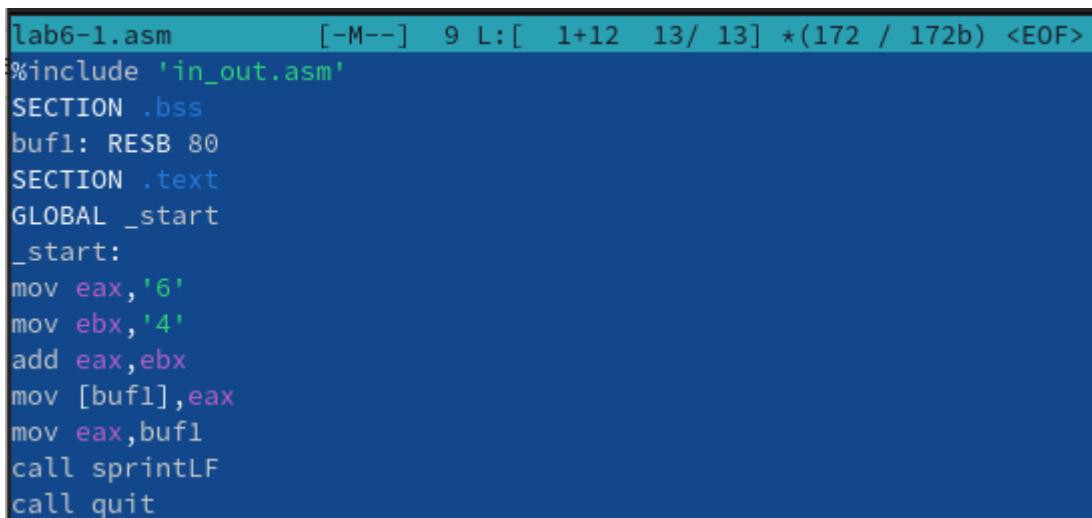
Для начала я создаю каталог для программ лабораторной работы № 6, перехожу в него и создаю файл lab6-1.asm. (рис. 1)



```
[kanechaeva@192 ~]$ mkdir ~/work/arch-pc/lab06
[kanechaeva@192 ~]$ cd ~/work/arch-pc/lab06
[kanechaeva@192 lab06]$ touch lab6-1.asm
```

Рис. 1. Создание рабочего пространства для выполнения лабораторной №6

Затем открываю Midnight Commander в терминале и с помощью функциональной клавиши F4 открываю файл lab6-1.asm для редактирования во встроенном редакторе. Затем ввожу текст программы из листинга 6.1, после чего сохраняю изменения и закрываю файл. (рис. 2)



```
lab6-1.asm  [-M--]  9  L:[  1+12  13/ 13]  *(172 / 172b) <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 sprintLF
call quit
```

Рис. 2. Редактирование файла

После копирования файла `in_out.asm` из файла `lab05` в файл `lab06` запускаю файл. (рис. 3)

```
[kanechaeva@192 lab06]$ nasm -f elf lab6-1.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[kanechaeva@192 lab06]$ ./lab6-1
j
[kanechaeva@192 lab06]$
```

Рис. 3. Запуск файла и получение результата

Далее изменяю текст программы. Вместо символов, записываю числа в регистры. (рис. 4)

```
lab6-1.asm [-M]
%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. Исправление файла

Теперь создаю исполняемый файл и запускаю его. (рис. 5)

```
[kanechaeva@192 lab06]$ nasm -f elf lab6-1.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[kanechaeva@192 lab06]$ ./lab6-1

[kanechaeva@192 lab06]$
```

Рис. 5. Запуск файла.

Теперь вывелся символ с кодом 10. В соответствии с ASCII таблицей это символ перевода строк и он не отображается при выводе на экран.

Теперь создаю файл `lab6-2.asm` в каталоге `~/work/arch-pc/lab06`. (рис. 6)

```

<- ~/work/arch-pc/lab06 .[^]>
.и      Имя      Размер  Время правки
/..      -ВВЕРХ-   ноя 18 10:36
in_out.asm      3942 ноя 11 15:11
*lab6-1         5160 ноя 18 11:16
lab6-1.asm      168  ноя 18 11:15
lab6-1.o        1200 ноя 18 11:15
-ВВЕРХ-
378G / 475G (79%)

<- ~ .[^]>
.и      Имя      Размер  Время правки
/..      -ВВЕРХ-   сен 25 22:34
.cache    500  ноя 11 14:03
.config   584  ноя 11 14:03
.java     30   окт 12 22:05
.local    32   окт 13 12:44
.mozilla  48   сен 25 22:47
.pki      10   сен 30 08:20
.ssh      84   сен 26 12:31
.texlive2022 18   окт 13 14:34
.var       6   сен 25 22:36
/work     24   окт 24 12:30
/Видео    0     сен 25 22:35
/Документы 882  ноя 18 11:23
/Загрузки 1252 ноя 11 19:18
/Изображения 102  ноя 11 19:17
-ВВЕРХ-
378G / 475G (79%)

Совет: Вы можете задать имя пользователя в команде: 'cd ftp://user@machine.edu'.
[kanechaeva@192 lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm

```

Рис. 6. Создание файла lab6-2.asm

После этого ввожу в него текст программы из листинга 6.2, затем создаю исполняемый файл и запускаю его. (рис. 7)

```

[kanechaeva@192 lab06]$ nasm -f elf lab6-2.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[kanechaeva@192 lab06]$ ./lab6-2
106
[kanechaeva@192 lab06]$

```

Рис. 7. Запуск файла lab6-2.asm

Аналогично предыдущему примеру изменяю символы на числа. Запускаю файл. (рис. 8) Рис. 8. Запуск измененного файла

```

[kanechaeva@192 lab06]$ nasm -f elf lab6-2.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[kanechaeva@192 lab06]$ ./lab6-2
10
[kanechaeva@192 lab06]$

```

Рис. 8. Запуск измененного файла

Теперь при исполнении программы будет получен не соответствующий символам код в системе ASCII, а само число, то есть 10.

Далее заменяю функцию `iprintLF` на `iprint`. Создаю исполняемый файл и запускаю его. (рис. 9)

```
[kanechaeva@192 lab06]$ nasm -f elf lab6-2.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[kanechaeva@192 lab06]$ ^C
[kanechaeva@192 lab06]$ ./lab6-2
10[kanechaeva@192 lab06]$
```

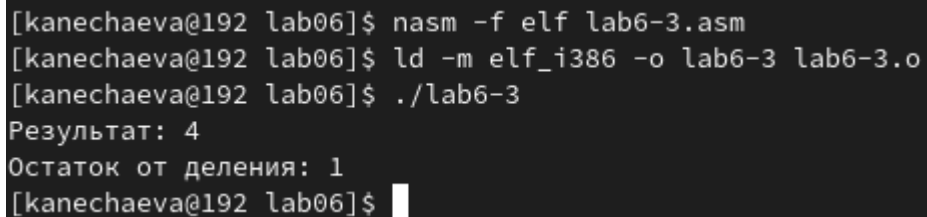
Рис. 9. Запуск исправленного файла

Изменилось то, что исчез символный перенос строк, который был засчет ограничения LF для `iprint`.

2. Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 и ввожу в файл текст программы из листинга 6.3. Создаю исполняемый файл и запускаю его.

(рис. 10)

A terminal window with a dark background and light gray text. The prompt is [kanechaeva@192 lab06]\$. The first command is nasm -f elf lab6-3.asm. The second command is ld -m elf_i386 -o lab6-3 lab6-3.o. The third command is ./lab6-3. The output shows 'Результат: 4' and 'Остаток от деления: 1'. The prompt returns to [kanechaeva@192 lab06]\$.

```
[kanechaeva@192 lab06]$ nasm -f elf lab6-3.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[kanechaeva@192 lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[kanechaeva@192 lab06]$
```

Рис. 10. Запуск файла lab6-3.asm

Затем изменяю текст программы для вычисления выражения $\square(\square) = (4 * 6 + 2)/5$. (рис. 11)

```
%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 di
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, 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. 11. Изменение программы

Создаю исполняемый файл и проверяю его работу. (рис. 12)

```
[kanechaeva@192 lab06]$ nasm -f elf lab6-3.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[kanechaeva@192 lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[kanechaeva@192 lab06]$
```

Рис. 12. Запуск измененного файла

В качестве другого примера рассмотрю программу вычисления варианта задания по номеру студенческого билета, работающую по следующему

алгоритму:

- вывести запрос на введение № студенческого билета
- вычислить номер варианта по формуле: $(\square\square \bmod 20) + 1$, где $\square\square$ – номер студенческого билета (В данном случае $\square \bmod \square$ – это остаток от деления \square на \square).
- вывести на экран номер варианта.

Для этого создаю файл variant.asm в каталоге и ввожу в него текст программы из листинга 6.4. (рис. 13) Рис. 13.

```
;-----  
; Программа вычисления варианта  
;-----  
%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 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  
div ebx  
inc edx  
mov eax, rem  
call sprintf  
mov eax, edx  
call iprintLF  
call quit
```

Рис. 13. Редактирование файла

Создаю исполняемый файл и запускаю его. Программа вывела, что мой вариант - 12. (рис. 14)

```
[kanechaeva@192 lab06]$ nasm -f elf variant.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o variant variant.o
[kanechaeva@192 lab06]$ ./variant
Введите № студенческого билета:
1132236031
Ваш вариант: 12
[kanechaeva@192 lab06]$
```

Рис. 14. Запуск файла

3. Ответы на вопросы

1. Для вывода сообщения «Ваш вариант» предоставляются строки кода:

```
mov eax,rem
```

```
call sprint
```

2. Инструкция `mov ecx, x` используется, чтобы поставить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, вводы сообщений с клавиатуры

3. Вызов `atoi` используется для вызова подпрограмм из внешнего файла, который преобразует символы `ascii`-кода в имена чисел и записывает результат в регистр `eax`.

4. Для обработки решений предоставляются следующие предложения:

```
xor edx,edx ; обнуление edx 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 остаток деления `div ebx` записывается в регистр `edx`.

6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1

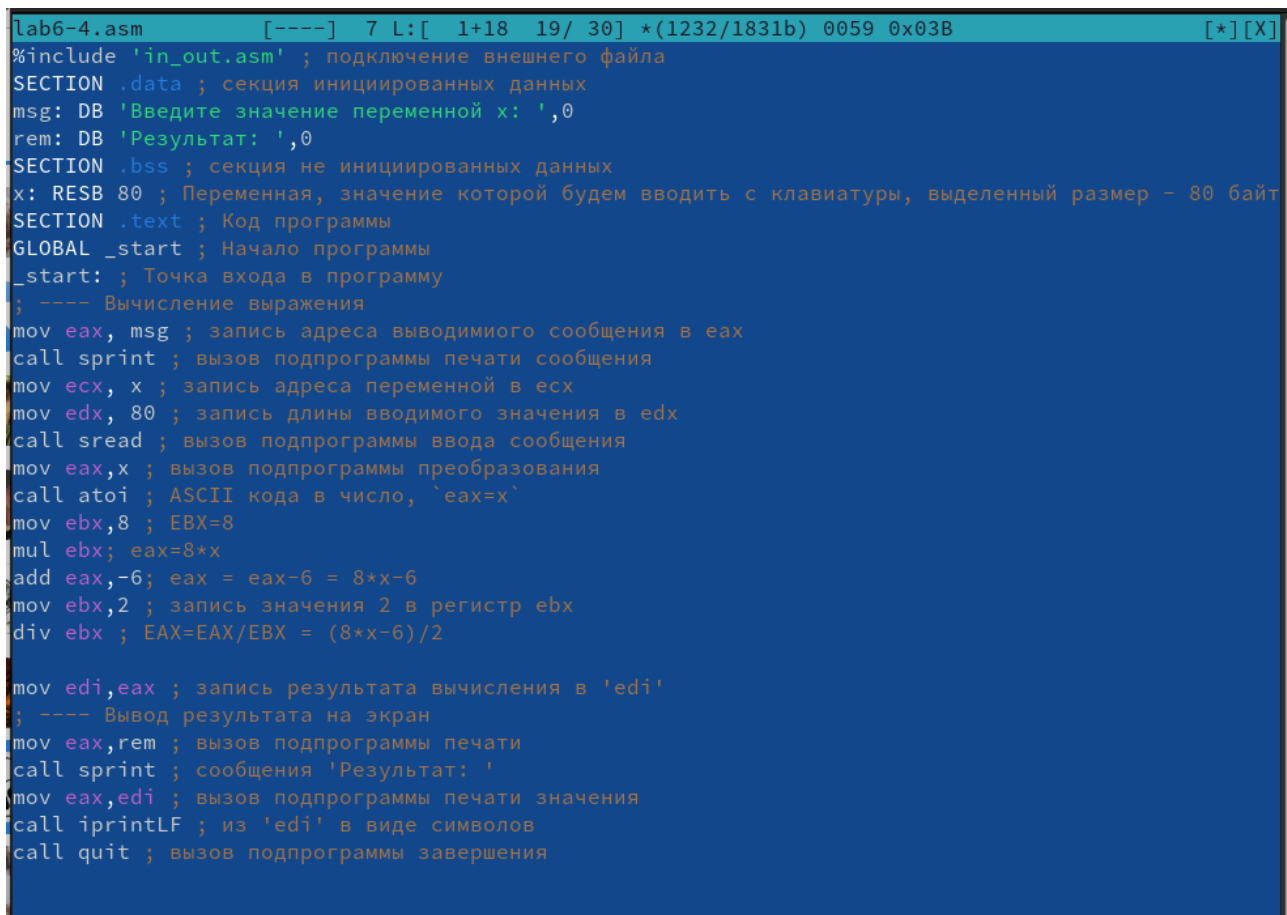
7. Для вывода результатов на экране компьютерных представлений:

```
mov eax,edx
```

```
call iprintLF
```

4. Задание для самостоятельной работы

Создаю файл lab6-4.asm и ввожу в него текст программы для вычисления значений выражений $(8x - 6)/2$. Это выражение было в варианте 12. (рис. 15)

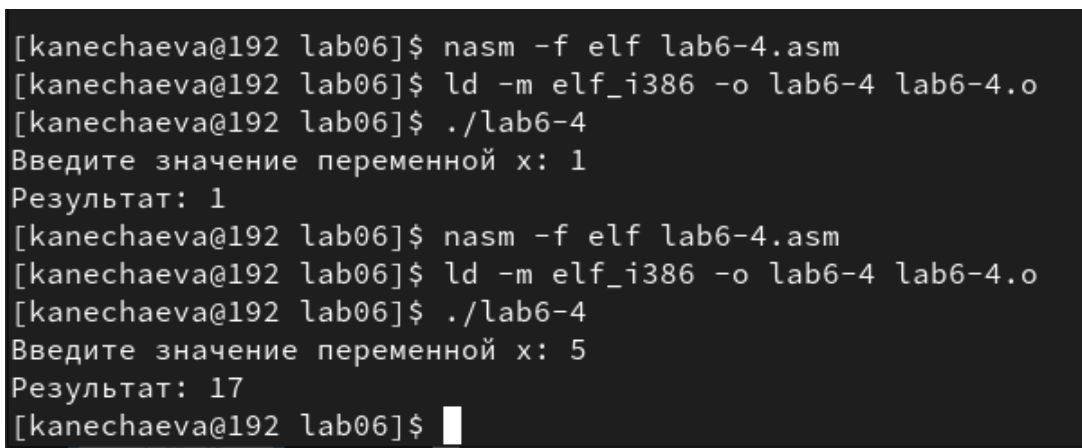


```
lab6-4.asm      [----] 7 L:[ 1+18 19/ 30] *(1232/1831b) 0059 0x03B      [*] [X]
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; секция инициированных данных
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss ; секция не инициированных данных
x: RESB 80 ; Переменная, значение которой будем вводить с клавиатуры, выделенный размер - 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
; ---- Вычисление выражения
mov eax, msg ; запись адреса выводимого сообщения в eax
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
mov ebx, 8 ; EBX=8
mul ebx ; eax=8*x
add eax, -6 ; eax = eax-6 = 8*x-6
mov ebx, 2 ; запись значения 2 в регистр ebx
div ebx ; EAX=EAX/EBX = (8*x-6)/2

mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 15. Программа для вычисления

Запускаю файл. (рис. 16)



```
[kanechaeva@192 lab06]$ nasm -f elf lab6-4.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[kanechaeva@192 lab06]$ ./lab6-4
Введите значение переменной x: 1
Результат: 1
[kanechaeva@192 lab06]$ nasm -f elf lab6-4.asm
[kanechaeva@192 lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[kanechaeva@192 lab06]$ ./lab6-4
Введите значение переменной x: 5
Результат: 17
[kanechaeva@192 lab06]$
```

Рис. 16. Запуск файла

4 Вывод

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

5 Источники

1. ТУИС – Архитектура ЭВМ – [Электронный ресурс] -
<https://esystem.rudn.ru/mod/resource/view.php?id=1030554>