РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Семенченко Т. С.

Группа: НКАбд-05-25

МОСКВА

2025г.

Оглавление

1 Цель работы

Освоить арифметические инструкций языка ассемблера NASM.

2 Задачи

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

3 Выполнение лабораторной работы

3.1Символьные и численные данные в NASM

Создала каталог для программ лабораторной работы №6, перешла в него и создала файл lab6-1.asm (рис. 3.1.1)

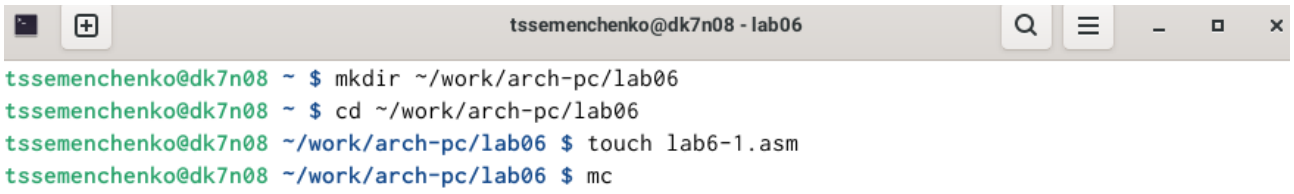

Рис. 3.1.1 Демонстрация написанных команд

Ввела в файл lab6-1.asm текст программы из листинга. Вывод программы отличается от предполагаемого изначально, ибо коды символов в сумме дают символ j по таблице ASCII. {#fig:003 width=70%} (рис. 3.1.2)

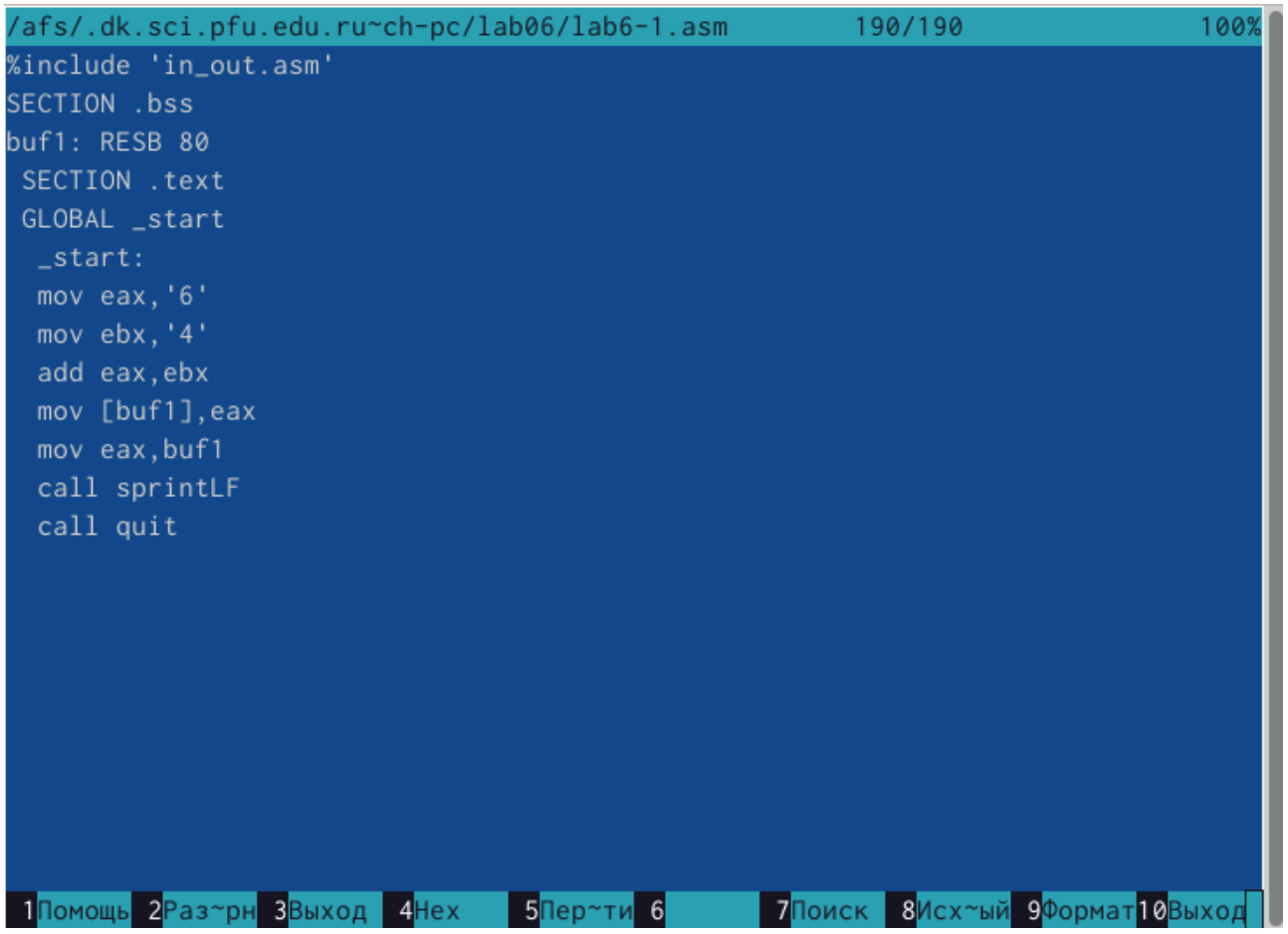

Рис. 3.1.2 Текст программы

Создала исполняемый файл и запустила его (рис. 3.1.3)

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./lab6-1
j
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.1.3 Демонстрация работы исполняемого файла

Изменила текст программы и вместо символов, записала в регистры числа. Исправила тест программы из листинга (рис. 3.1.4)

```
lab6-1.asm        [----] 11 L:[  1+12  13/ 13] *(186 / 186b) <EOF>                [*][X]
%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. 3.1.4 Демонстрация исправленной программы

Создала исполняемый файл и запустила его (рис. 3.1.4)

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./lab6-1
```

Рис. 3.1.4 Работа исполняемого файла. Данный символ не отображается на экране, , это связано с тем, что символ 10 означает переход на новую строку.

Создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввела в него текст программы из второго листинга (рис. 3.1.5, 3.1.6)

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-2.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.1.5 Создание файла

```
lab6-2.asm          [----] 11 L:[  1+ 8   9/  9] *(131 / 131b) <EOF>
%include 'in_out.asm'
 SECTION .text
 GLOBAL _start
  _start:
  mov eax,'6'
  mov ebx,'4'
  add eax,ebx
  call iprintLF
  call quit
```

Рис. 3.1.6 Демонстрация введенной программы

Создала исполняемый файл и запустила его. Программа, как и в первый раз, сложила коды символов, но вывела само число, а не его символ, благодаря замене функции вывода на iprintLF (рис. 3.1.7)

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./lab6-2
106
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.1.7 Результат работы программы

Аналогично предыдущему заданию изменила текст программы и вместо символов, записала в регистры числа. Создала исполняемый файл и запустила его (рис. 3.1.8, 3.1.9)

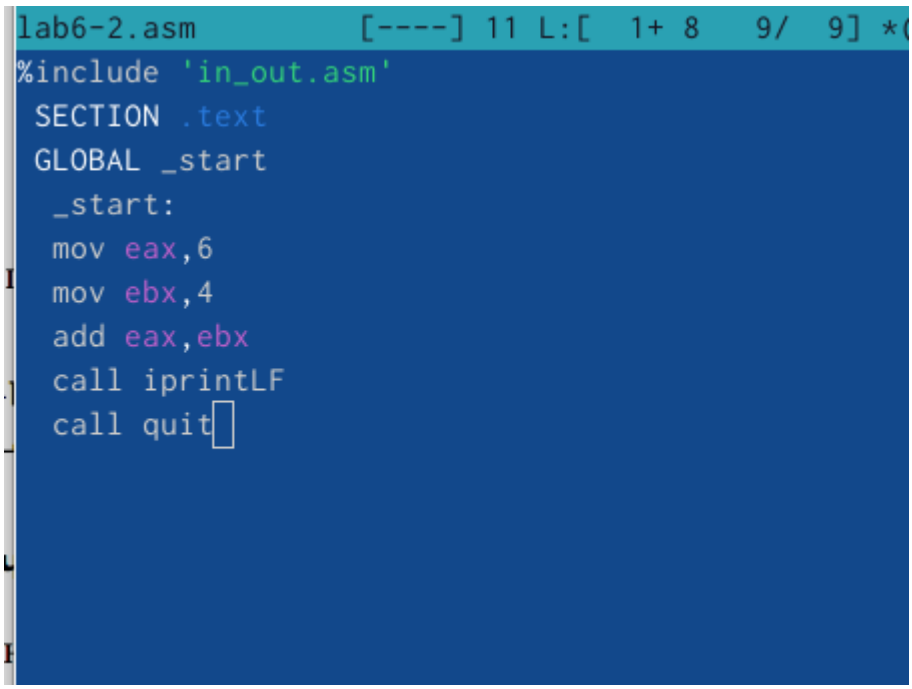

Рис. 3.1.8 Исправленная программа

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./lab6-2
10
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.1.9 Демонстрация работы программы

Заменила функцию iprintLF на iprint. Создала исполняемый файл и запустила его (рис. 3.1.10, 3.1.11)

Рис. 3.1.10 Замена функции в программе

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./lab6-2
10tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.1.11 Демонстрация работы программы. Функция iprintLF выводит результат на отдельной строке, iprint выводит результат без переноса строки.

3.2Выполнение арифметических операций в NASM

В качестве примера выполнения арифметических операций в NASM привела программу вычисления арифметического выражения $f(x) = (5 * 2 + 3)/3$.

Создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06. Ввела в него текст программы из третьего листинга. Создала исполняемый файл и запустила его (рис. 3.2.1)

```
10tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-3.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ mc

tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.2.1 Создание файла, создание исполняемого файла, результат работы программы.

Изменила текст программы для вычисления выражения $f(x) = (4 * 5 + 2)/5$. Создала исполняемый файл и проверила его работу (рис. 3.2.2)

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 2
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.2.2 Результат работы программы для заданного выражения

Создала файл variant.asm в каталоге ~/work/arch-pc/lab06. Изучила тест программы из четвертого листинга и ввела его в файл (рис. 3.2.3, 3.2.4, 3.2.5)

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
```

Рис. 3.2.3 Создание файла

```
variant.asm        [----] 11 L:[  1+24  25/ 26] *(426 / 428b) 0010 0x00A
%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 sprintLF
  mov ecx, x
  mov edx, 80
  call sread
  mov eax, x
  call atoi
  xor edx,edx
  mov ebx,20
  div ebx
  inc edx
  mov eax,rem
  call sprint
  mov eax,edx
  call iprintLF
  call quit
```

Рис. 3.2.4 Демонстрация программы

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1032253509
Ваш вариант: 10
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 3.2.5 Создание исполняемого файла и результат его работы. Я запустила программу и ввела свой номер студенческого билета, получила номер своего варианта.

Ответы на вопросы:

1. mov eax,rem

call sprint

Данные строки отвечают за вывод сообщение «Ваш вариант»

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 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 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 Задание для самостоятельной работы

В соответствии с выбранным вариантом, я реализую программу для подсчета функции f(x) = 5(x + 18) − 28, проверка на нескольких переменных показывает корректное выполнение программы (рис. 4.1)

```
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ nasm -f elf variant10.asm
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant10 variant10.o
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./variant10
Введите значение переменной: 2
72
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $ ./variant10
Введите значение переменной: 3
77
tssemenchenko@dk7n08 ~/work/arch-pc/lab06 $
```

Рис. 4.1 Запуск, проверка и результаты программы

```
/afs/.dk.sci.pfu.edu.ru/home/t/s/tss~ko/work/arch-pc/lab06/variant
%include 'in_out.asm'
 SECTION .data
 msg: DB 'Введите значение переменной: ',0
 rem: DB 'Результат: ',0
 SECTION .bss
 x:      RESB 80
 SECTION .text
 GLOBAL _start
  _start:
  mov eax, msg
  call sprint
  mov ecx, x
  mov edx, 80
  call sread
  mov eax, x
  call atoi
  add eax,18
  mov ebx,5
  mul ebx
  sub eax,28
  mov edi,eax
  xor eax,rem
  call sprint
  mov eax,edi
  call iprintLF
  call quit
```

Рис. 4.2 Текст программы для подсчета функции

Прикрепляю текст программы:

```
%include 'in_out.asm'
 SECTION .data
 msg: DB 'Введите значение переменной: ',0
 rem: DB 'Результат: ',0
 SECTION .bss
 x:      RESB 80
 SECTION .text
 GLOBAL _start
  _start:
```

```
mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
add eax,18
mov ebx,5
mul ebx
sub eax,28
mov edi,eax
xor eax,rem
call sprint
mov eax,edi
call iprintLF
call quit
```

5 Выводы

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