

Отчёт по лабораторной работе №7

Дисциплина: Архитектура компьютера

Алиева Милена Арифовна

Содержание

- 1 Цель работы
- 2 Задания
- 3 Теоретическое введение
- 4 Выполнение лабораторной работы
- 5 Вывод

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

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

3 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.

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`.

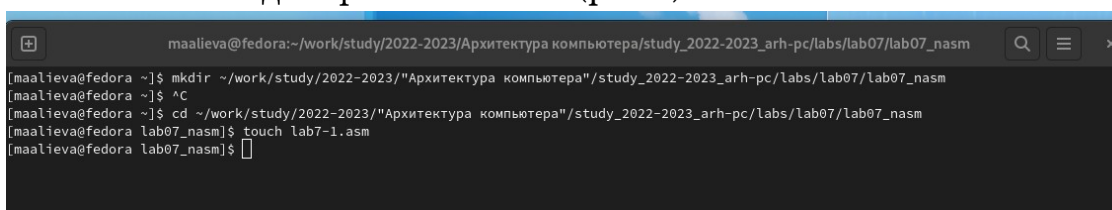
Адресация памяти – операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

Довольно часто при написании программ встречается операция прибавления или вычитания единицы. Прибавление единицы называется инкрементом, а вычитание — декрементом. Для этих операций существуют специальные команды: inc (от англ. increment) и dec (от англ. decrement), которые увеличивают и уменьшают на 1 свой операнд.

Команда neg рассматривает свой операнд как число со знаком и меняет знак операнда на противоположный. Операндом может быть регистр или ячейка памяти любого размера.

4 Выполнение лабораторной работы

1. Создаю каталог для программ лабораторной работы № 7, перехожу в него и создаю файл lab7-1.asm: (рис. 1)



```
maalieva@fedora:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab07/lab07_nasm
[maalieva@fedora ~]$ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab07/lab07_nasm
[maalieva@fedora ~]$ ^C
[maalieva@fedora ~]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab07/lab07_nasm
[maalieva@fedora lab07_nasm]$ touch lab7-1.asm
[maalieva@fedora lab07_nasm]$
```

Рис. 1: Создание каталога lab7-1.asm и файла для работы

2. Открываю файл lab7-1.asm и вставляю в него программу вывода значений, записанных в регистр eax (рис. 2)



```
Открыть ▾ + lab7-1.asm Стр. 14, Поз. 1
~/work/study/2022-2023/... pc/labs/lab07/lab07_nasm

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

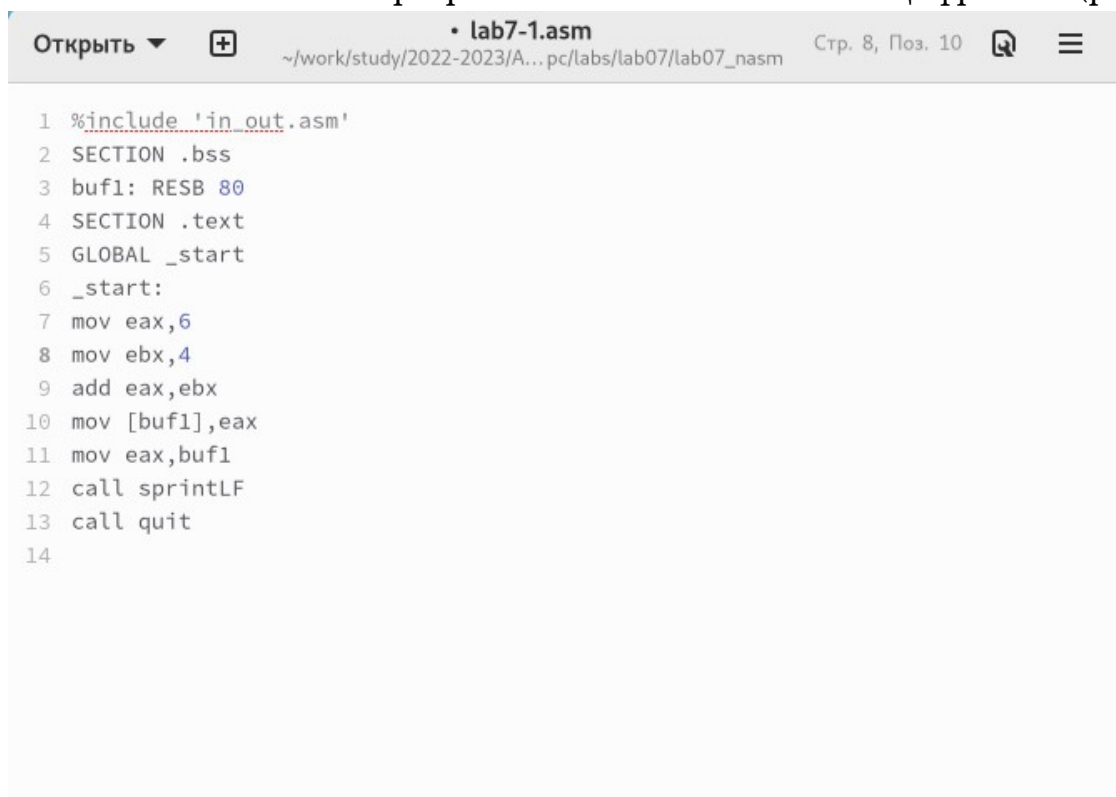
Рис. 2: Текст программы

3. Создаю исполняемый файл и запускаю его, предварительно добавив в каталог подключаемый файл in_out.asm. Результатом будет символ j. Это происходит потому, что код символа 6 равен 00110110 в двоичном представлении, а код символа 4 – 00110100. Команда add eax,ebx запишет в регистр eax сумму кодов – 01101010 (106 в десятичном представлении), что в свою очередь является кодом символа j по таблице ASCII (рис. 3)

```
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-1.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[maalieva@fedora lab07_nasm]$ ./lab7-1
j
[maalieva@fedora lab07_nasm]$
```

Рис. 3: Создание исполняемого файла и запуск программы

4. Заменяю в тексте программы символы “6” и “4” на цифры 6 и 4(рис. 4)



```
• lab7-1.asm
~/work/study/2022-2023/A...pc/labs/lab07/lab07_nasm  Стр. 8, Поз. 10

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

Рис. 4: Изменённый текст программы

5. Создаю новый исполняемый файл программы и запускаю его. Результатом является пустой вывод(так как вывелся символ с кодом 10, а это символ перевода строки, этот символ не отображается при выводе на экран (рис. 5))

```
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-1.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[maalieva@fedora lab07_nasm]$ ./lab7-1
[maalieva@fedora lab07_nasm]$
```

Рис. 5: Вывод изменённой программы

6. Создаю файл lab7-2.asm с помощью утилиты touch. Ввожу в файл текст программы для вывода значения регистра eax (рис. 6)



```
lab7-2.asm
~/work/study/2022-2023/A...pc/labs/lab07/lab07_nasm  Стр. 9, Поз. 10

1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,'6'
6 mov ebx,'4'
7 add eax,ebx
8 call iprintLF
9 call quit
```

Рис. 6: Программа для вывода значения регистра eax

7. Создаю исполняемый файл и запускаю его. Результатом является число 106(это происходит потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”) (рис. 7)

```
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-2.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[maalieva@fedora lab07_nasm]$ ./lab7-2
106
[maalieva@fedora lab07_nasm]$
```

Рис. 7: Вывод программы(файл lab7-2)

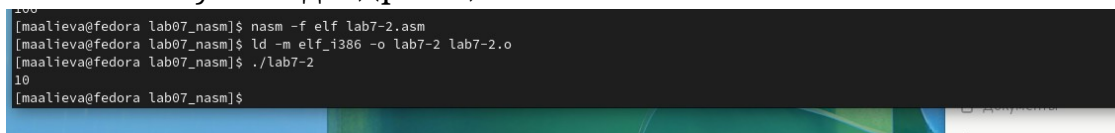
8. Заменяю в тексте программы в файле lab7-2.asm символы “6” и “4” на числа 6 и 4 (рис. 8)



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
```

Рис. 8: Замена в тексте программы “6” и “4” на числа 6 и 4

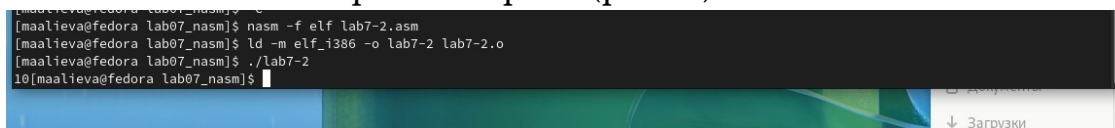
9. Создаю исполняемый файл и запускаю его. Результатом является число 106(это происходит потому что теперь программа складывает не соответствующие символы коды в системе ASCII, а сами числа, поэтому вывод 10)(рис. 9)



```
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-2.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[maalieva@fedora lab07_nasm]$ ./lab7-2
10
[maalieva@fedora lab07_nasm]$
```

Рис. 9: Запуск изменённой программы в файле lab7-2

10. Заменяю в тексте программы функцию iprintLF на iprint. Создаю исполняемый файл и запускаю его. Результатом является исчезновение переноса строки (рис. 10)



```
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-2.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[maalieva@fedora lab07_nasm]$ ./lab7-2
10[maalieva@fedora lab07_nasm]$
```

Рис. 10: Замена функции iprintLF на iprint

11. Создаю файл lab7-3.asm с помощью утилиты touch. Ввожу в созданный файл текст программы для вычисления значения выражения $f(x) = (5 * 2 + 3)/3$. Создаю исполняемый файл и запускаю его. Получаю корректный результат (рис. 11)

```
10[maalieva@fedora lab07_nasm]$ touch lab7-3.asm
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-3.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[maalieva@fedora lab07_nasm]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[maalieva@fedora lab07_nasm]$
```

Рис. 11: Программа по вычислению выражения $f(x) = (5 * 2 + 3)/3$

12. Изменяю программу так, чтобы она вычисляла значение выражения $f(x) = (4 * 6 + 2)/5$ (рис. 12)

```
Открыть + lab7-3.asm
~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab07/lab07_nasm Стр. 15, Поз. 20

1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 div: DB 'Результат: ',0
4 rem: DB 'Остаток от деления: ',0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 ; ---- Вычисление выражения
9 mov eax,4 ; EAX=4
10 mov ebx,6 ; EBX=6
11 mul ebx ; EAX=EAX*EBX
12 add eax,2 ; EAX=EAX+2
13 xor edx,edx ; обнуляем EDX для корректной работы div
14 mov ebx,5 ; EBX=5
15 div ebx ; EAX=EAX/5, EDX=остаток от деления
16 mov edi,eax ; запись результата вычисления в 'edi'
17 ; ---- Вывод результата на экран
18 mov eax,div ; вызов подпрограммы печати
19 call sprint ; сообщения 'Результат: '
20 mov eax,edi ; вызов подпрограммы печати значения
21 call iprintfLF ; из 'edi' в виде символов
22 mov eax,rem ; вызов подпрограммы печати
23 call sprint ; сообщения 'Остаток от деления: '
24 mov eax,edx ; вызов подпрограммы печати значения
25 call iprintfLF ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения
```

Рис. 12: Программа по вычислению выражения $f(x) = (4 * 6 + 2)/5$

13. Создаю исполняемый файл и запускаю его. Вывод результата и остатка от деления. Программа отработала верно (рис. 13)

```
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-3.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[maalieva@fedora lab07_nasm]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[maalieva@fedora lab07_nasm]$
```

Рис. 13: Создание исполняемого файла и вывод программы

14. Создаю файл variant.asm с помощью утилиты touch. Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 14)

```

1  %include 'in_out.asm'
2  SECTION .data
3  msg: DB 'Введите № студенческого билета: ',0
4  rem: DB 'Ваш вариант: ',0
5  SECTION .bss
6  x: RESB 80
7  SECTION .text
8  GLOBAL _start
9  _start:
10 mov eax, msg
11 call sprintf
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax,x ; вызов подпрограммы преобразования
16 call atoi ; ASCII кода в число, `eax=x`
17 xor edx,edx
18 mov ebx,20
19 div ebx
20 inc edx
21 mov eax,rem
22 call sprint
23 mov eax,edx

```

Рис. 14: Текст программы по вычислению варианта

Ответы на вопросы:

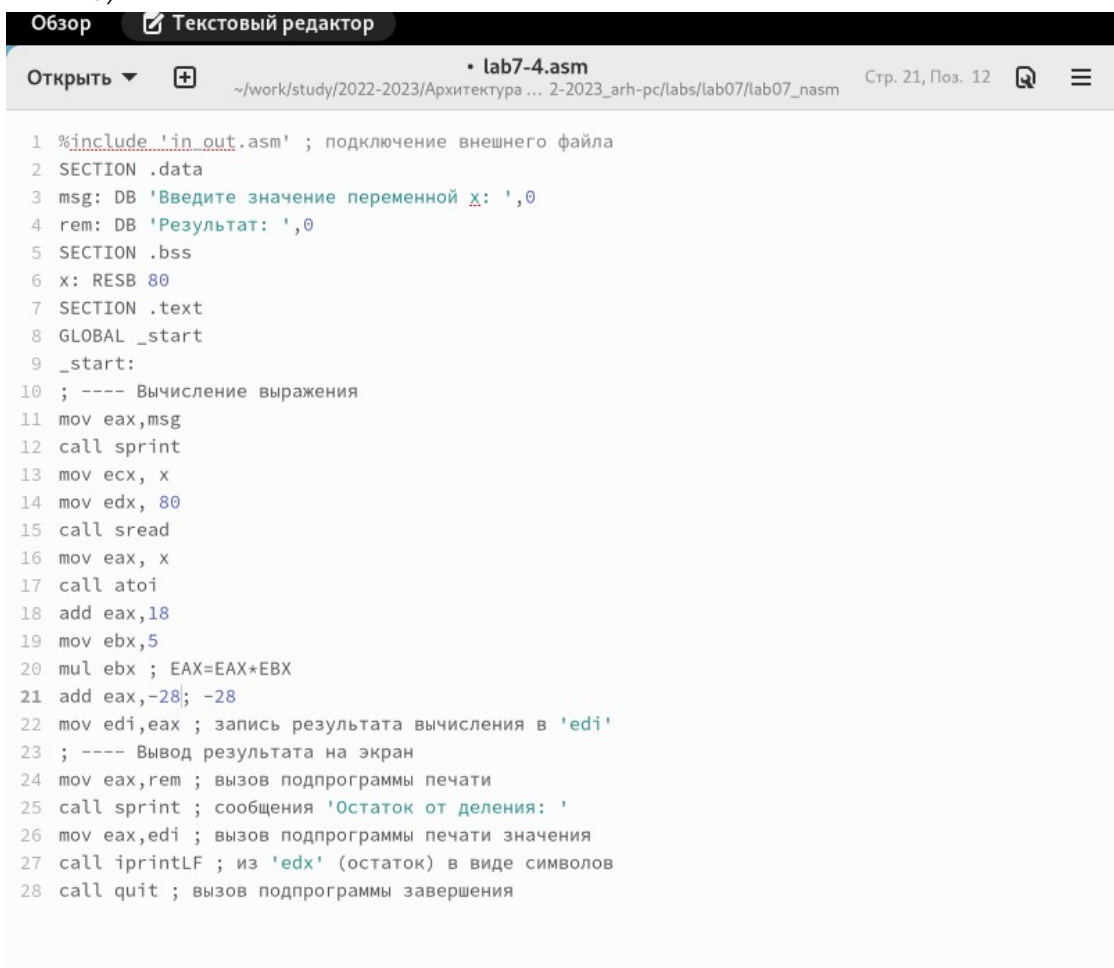
- 1) За вывод на экран сообщения 'Ваш вариант:' отвечают строки `mov eax,rem` `call sprint`
- 2) `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
- 3) Инструкция `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 вычисление варианта отвечают строки `xor edx,edx` `mov ebx,20` `div ebx` `inc edx`
- 5) Остаток от деления при выполнении инструкции "div ebx" записывается в регистр `edx`
- 6) Инструкция "inc edx" используется для увеличения значения регистра `edx` на 1
- 7) За вывод на экран результата вычислений отвечают строки `mov eax,edx` `call iprintLF`

15. Создаю исполняемый файл и запускаю его. Получаю номер варианта, при проверке всё сходится (рис. 15)

```
[maalieva@fedora lab07_nasm]$ nasm -f elf variant.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o variant variant.o
[maalieva@fedora lab07_nasm]$ ./variant
Введите № студенческого билета:
1132226430
Ваш вариант: 11
[maalieva@fedora lab07_nasm]$
```

Рис. 15: Вывод программы из файла variant.asm

16. Приступаю к выполнению заданий для самостоятельной работы. Создаю файл lab7-4.asm с помощью утилиты touch. Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения под номером 10: $(x + 18) * 5 - 28$ (рис. 16)



```
Обзор  Текстовый редактор
Открыть  +  • lab7-4.asm  Стр. 21, Поз. 12
~/work/study/2022-2023/Архитектура ... 2-2023_arh-pc/labs/lab07/lab07_nasm

1  %include 'in_out.asm' ; подключение внешнего файла
2  SECTION .data
3  msg: DB 'Введите значение переменной x: ',0
4  rem: DB 'Результат: ',0
5  SECTION .bss
6  x: RESB 80
7  SECTION .text
8  GLOBAL _start
9  _start:
10 ; ---- Вычисление выражения
11 mov eax,msg
12 call sprint
13 mov ecx, x
14 mov edx, 80
15 call sread
16 mov eax, x
17 call atoi
18 add eax,18
19 mov ebx,5
20 mul ebx ; EAX=EAX*EBX
21 add eax,-28; -28
22 mov edi,eax ; запись результата вычисления в 'edi'
23 ; ---- Вывод результата на экран
24 mov eax,rem ; вызов подпрограммы печати
25 call sprint ; сообщения 'Остаток от деления: '
26 mov eax,edi ; вызов подпрограммы печати значения
27 call iprintLF ; из 'edx' (остаток) в виде символов
28 call quit ; вызов подпрограммы завершения
```

Рис. 16: Текст программы для вычисления $(x + 18) * 5 - 28$

17. Создаю исполняемый файл и запускаю его. При вводе $x = 2$ получаю ответ 72, при вводе $x = 3$ получаю ответ 77. Делаю проверку вручную. Оба полученных значения являются корректными (рис. 17)


```
Результат: 102
[maalieva@fedora lab07_nasm]$ nasm -f elf lab7-4.asm
[maalieva@fedora lab07_nasm]$ ld -m elf_i386 -o lab7-4 lab7-4.o
[maalieva@fedora lab07_nasm]$ ./lab7-4
Введите значение переменной x: 2
Результат: 72
[maalieva@fedora lab07_nasm]$ ./lab7-4
Введите значение переменной x: 3
Результат: 77
[maalieva@fedora lab07_nasm]$
```

Рис. 17: Вывод программы по вычислению $(x + 18) * 5 - 28$

5 Выводы

При выполнении лабораторной работы освоила арифметические инструкции языка ассемблера NASM.

Список литературы