

Лабораторная работа №7.Арифметические

Архитектура ЭВМ

Плескачева Елизавета Андреевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Замена символов на числа	8
3.1	Вывод на экран значения <code>eah</code>	8
3.2	Замена <code>iprintLf</code> на <code>iprint</code>	10
3.3	Программа на выполнение арифметической операции	11
3.4	Программа вычисляющая $(4*6 + 2/5)$	12
3.5	Программа для вывода варианта	12
3.5.1	Ответы на вопросы	13
4	Задание для самостоятельной работы	15
5	Выводы	17
	Список литературы	18

Список иллюстраций

2.1	Создание рабочей папки и файлов	6
2.2	Содержимое lab7-1.asm	6
2.3	Вывод программы в консоль	7
3.1	Измененная часть lab7-1.asm	8
3.2	Запуск измененной lab7-2	8
3.3	Код из листинга 7.2	9
3.4	Вывод программы lab7-2	9
3.5	Замена символов на числа	10
3.6	Вывод измененной программы	10
3.7	Вывод измененной программы lab7-2	10
3.8	Код из листинга 7.3	11
3.9	Вывод программы lab7-3 на экран	12
3.10	Измененная програма по вычислению выражения	12
3.11	Вывод вычислений программы	12
3.12	Создание файла variant.asm	12
3.13	Текст листинга 7.4	13
3.14	Вывод программы по номеру студенческого билета	13
4.1	Основная часть кода lab7-4.asm	15
4.2	Проверка правильного исполнения программы	15

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

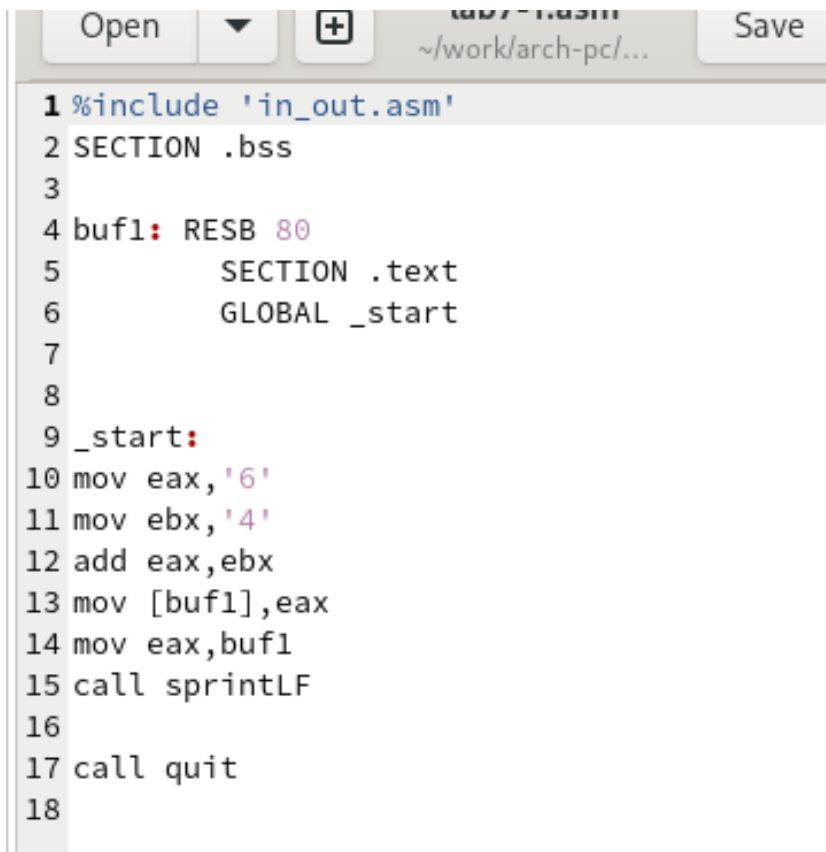
2 Выполнение лабораторной работы

Создадим папку, перейдем в нее и создадим файлы

```
[eapleskacheva@localhost ~]$ mkdir ~/work/arch-pc/lab07  
[eapleskacheva@localhost ~]$ cd ~/work/arch-pc/lab07  
[eapleskacheva@localhost lab07]$ touch lab7-1.asm  
[eapleskacheva@localhost lab07]$ gedit lab7-1.asm
```

Рис. 2.1: Создание рабочей папки и файлов

Введем текст из листинга 7.1 в файл



```
1 %include 'in_out.asm'  
2 SECTION .bss  
3  
4 buf1: RESB 80  
5 SECTION .text  
6 GLOBAL _start  
7  
8  
9 _start:  
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  
18
```

Рис. 2.2: Содержимое lab7-1.asm

Создадим исполняемый файл и запустим

```
[eapleskacheva@localhost lab07]$ ./lab7-1.c  
[eapleskacheva@localhost lab07]$ nasm -f elf lab7-1.asm  
ld -m elf_i386 -o lab7-1 lab7-1.o  
./lab7-1  
j
```

Рис. 2.3: Вывод программы в консоль

Программа вывела j так как мы складывали символы

3 Замена символов на числа

Уберем кавычки

```
9 _start:  
10 mov eax, 6  
11 mov ebx, 4  
12 add eax, ebx
```

Рис. 3.1: Измененная часть lab7-1.asm

Снова скомпилируем и запустим программу

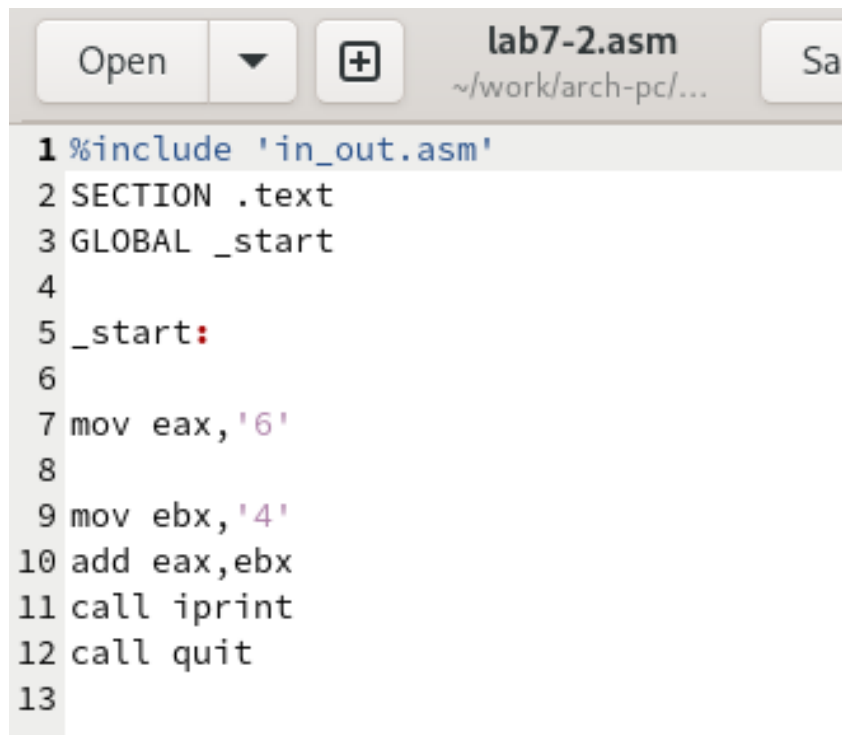
```
[eapleskacheva@localhost lab07]$ nasm -f elf lab7-1.asm  
ld -m elf_i386 -o lab7-1 lab7-1.o  
./lab7-1  
  
[eapleskacheva@localhost lab07]$
```

Рис. 3.2: Запуск измененной lab7-2

Теперь на экран вывелся перенос строки (его символ 10)

3.1 Вывод на экран значения eax

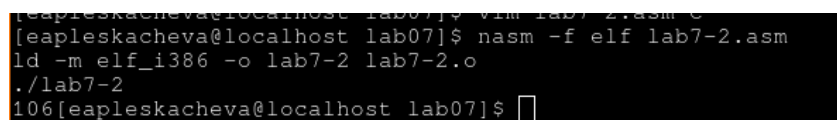
Создадим файл lab7-2.asm и откроем его в gedit. Введем код из листинга 7.2



```
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4
5 _start:
6
7 mov eax, '6'
8
9 mov ebx, '4'
10 add eax, ebx
11 call iprint
12 call quit
13
```

Рис. 3.3: Код из листинга 7.2

Скомпилируем код и запустим его



```
[eapleskacheva@localhost lab07]$ vim lab7-2.asm
[eapleskacheva@localhost lab07]$ nasm -f elf lab7-2.asm
ld -m elf_i386 -o lab7-2 lab7-2.o
./lab7-2
106[eapleskacheva@localhost lab07]$
```

Рис. 3.4: Вывод программы lab7-2

На экран вывелся код символа j

Теперь снова уберем из кода кавычки

```

3 GLOBAL _start
4
5 _start:
6
7 mov eax, 6
8
9 mov ebx, 4
10 add eax, ebx
11 call iprint
12 call quit

```

Рис. 3.5: Замена символов на числа

Запустим измененную программу

```

[eapleskacheva@localhost lab07]$ nasm -f elf lab7-2.asm
[eapleskacheva@localhost lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[eapleskacheva@localhost lab07]$ ./lab7-2
10[eapleskacheva@localhost lab07]$ 

```

Рис. 3.6: Вывод измененной программы

Теперь программа выводит просто сумму чисел $6+4 = 10$

3.2 Замена iprintLf на iprint

Заменяем функцию iprintLF на iprint

Запустим теперь программу

```

[eapleskacheva@localhost lab07]$ ./lab7-2
10
[eapleskacheva@localhost lab07]$ 

```

Рис. 3.7: Вывод измененной программы lab7-2

Теперь 10 вывелось с переносом строки, из-за того, что мы изменили функцию

3.3 Программа на выполнение арифметической операции

Создадим файл lab7-3.asm и введем в него код из листинга 7.3

```
1 %include 'in_out.asm' ; подключение внешнего файла
2
3 SECTION .data
4 div: DB 'Результат: ',0
5 rem: DB 'Остаток от деления: ',0
6
7 SECTION .text
8 GLOBAL _start
9 _start: ; ---- Вычисление выражения
10 mov eax,5 ; EAX=5
11 mov ebx,2 ; EBX=2
12 mul ebx ; EAX=EAX*EBX
13
14
15 add eax,3 ; EAX=EAX+3
16 xor edx,edx ; обнуляем EDX для корректной работы div
17 mov ebx,3 ; EBX=3
18 div ebx ; EAX=EAX/3, EDX=остаток от деления
19
20 mov edi,eax ; запись результата вычисления в 'edi'
21
22 ; ---- Вывод результата на экран
23
24 mov eax,div ; вызов подпрограммы печати
25 call sprint ; сообщения 'Результат: '
26 mov eax,edi ; вызов подпрограммы печати значения
27 call iprintLF ; из 'edi' в виде символов
28
29 mov eax,rem ; вызов подпрограммы печати
30 call sprint ; сообщения 'Остаток от деления: '
31 mov eax,edx ; вызов подпрограммы печати значения
32 call iprintLF ; из 'edx' (остаток) в виде символов
33
34 call quit ; вызов подпрограммы завершения
35
36
```

Рис. 3.8: Код из листинга 7.3

Скомпилируем программу и запустим ее

```
[eapleskacheva@localhost lab07]$ nasm -f elf ./lab7-3.asm
ld -m elf_i386 -o ./lab7-3 ./lab7-3.o
./lab7-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.9: Вывод программы lab7-3 на экран

На экран выводится целая часть и остаток от деления выражения $(5*2 + 3)/3$

3.4 Программа вычисляющая $(4*6 + 2/5)$

Изменим код программы так, что бы она вычисляла значение выражения $(4*6 + 2)/5$

```
10 mov eax,4 ; EAX=5
11 mov ebx,6 ; EBX=2
12 mul ebx ; EAX=EAX*EBX
13
14
15 add eax,2 ; EAX=EAX+3
16 xor edx,edx ; обнуляем EDX для корректной работы div
17 mov ebx,5 ; EBX=3
18 div ebx ; EAX=EAX/3, EDX=остаток от деления
```

Рис. 3.10: Измененная программа по вычислению выражения

```
[eapleskacheva@localhost lab07]$ nasm -f elf ./lab7-3.asm
ld -m elf_i386 -o ./lab7-3 ./lab7-3.o
./lab7-3
Результат: 5
Остаток от деления: 1
```

Рис. 3.11: Вывод вычислений программы

3.5 Программа для вывода варианта

Создадим файл variant.asm и введем в него код из листинга 7.4

```
[eapleskacheva@localhost lab07]$ touch variant.asm
[eapleskacheva@localhost lab07]$ gedit variant.asm
```

Рис. 3.12: Создание файла variant.asm

```

1 ;-----
2 ; Программа вычисления варианта
3 ;-----
4 %include 'in_out.asm'
5
6 SECTION .data
7 msg: DB 'Введите № студенческого билета: ',0
8 rem: DB 'Ваш вариант: ',0
9
10 SECTION .bss
11 x:    RESB 80
12
13 SECTION .text
14 GLOBAL _start
15 _start:
16     mov eax, msg
17     call sprintf
18
19     mov ecx, x          ; адресс `x` в ecx
20     mov edx, 80         ; размер буфера в edx
21     call sread          ; чтение ввода в `x`
22
23     mov eax, x ; вызов подпрограммы преобразования
24     call atoi ; ASCII кода в число, `eax=x`
25
26     xor edx,edx         ; очищение edx
27     mov ebx,20          ; 20 в ebx
28     div ebx            ; делим значение из eax на 20
29     inc edx            ; увеличиваем остаток от деления на 1
30
31     mov eax,rem         ; Запись адреса строки 'Ваш вариант' в eax
32     call sprint         ; Вызов вывода строки без переноса строки
33     mov eax,edx         ; перенос вычисленного остатка в eax
34     call iprintLF       ; Вывод вычисленного остатка с переносом строки
35
36     call quit
37

```

Рис. 3.13: Текст листинга 7.4

Скомпилируем и запустим программу. Введем туда свой студенческий билет.

```

[eapleskacheva@localhost lab07]$
nasm -f elf ./variant.asm
ld -m elf_i386 -o ./variant ./variant.o
./variant
Введите № студенческого билета:
1132226461
Ваш вариант: 2
[eapleskacheva@localhost lab07]$

```

Рис. 3.14: Вывод программы по номеру студенческого билета

Программа вывела вариант: 2

3.5.1 Ответы на вопросы

1. За вывод на экран “Ваш вариант” отвечают стоки 31-32
2. инструкции на строка 19-21 используется для ввода числа из консоли в переменную x

3. `call atoi` используется для преобразования ASCII кода в число
4. За вычисление варианта отвечают строки 26-29
5. Остаток от деления записывается в `edx`
6. `inc edx` увеличивает значение `edx` на 1
7. За вывод результата вычислений отвечают строки 33-34

4 Задание для самостоятельной работы

Мой вариант, 2, поэтому надо написать программу вычисляющую значение $(12x + 3) * 5$

Изменим программу выше, подставив вместо вычислений варианта по студенческому билету, вычисление функции

```
18  
19     mov ecx, x           ; адресс `x` в ecx  
20     mov edx, 10          ; размер буфера в edx  
21     call sread           ; чтение ввода в `x`  
22  
23  
24  
25  
26     mov eax, x ; вызов подпрограммы преобразования  
27     call atoi ; ASCII кода в число, `eax=x`  
28  
29  
30     mov ecx, 12          ; ecx = 12  
31     mul ecx              ; eax = 12*x  
32     add eax, 3           ; eax = 12*x + 3  
33     mov ecx, 5           ; ecx = 5  
34     mul ecx              ; eax = 5*(12*x + 3)  
35  
36
```

Рис. 4.1: Основная часть кода lab7-4.asm

Скомпилируем и запустим программу. Введем значения $x = 1$ и $x = 6$

```
[eapleskacheva@localhost lab07]$ nasm -f elf ./lab7-4.asm  
[eapleskacheva@localhost lab07]$ ld -m elf_i386 -o ./lab7-4 ./lab7-4.o  
[eapleskacheva@localhost lab07]$ ./lab7-4  
Введите X:  
1  
Ответ: 75  
[eapleskacheva@localhost lab07]$ ./lab7-4  
Введите X:  
6  
Ответ: 375  
[eapleskacheva@localhost lab07]$
```

Рис. 4.2: Проверка правильного исполнения программы

Программа вычисляет правильно

5 Выводы

Мы освоили арифметические операции на языке ассемблера NASM и написали программу, вычисляющую значение функции

Список литературы