

Отчет по лабораторной работе 6

**Основы работы с midnight commander. Структура программы на языке
ассемблера NASM**

Симко Сергей Евгеньевич

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Выполнение лабораторной работы | 6 |
| 3 | Ответы на вопросы | 11 |
| 4 | Самостоятельная работа | 12 |
| 5 | Выводы | 14 |

Список иллюстраций

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

```
[sesimko@fedora ~]$ mkdir ~/work/arch-pc/lab07
[sesimko@fedora ~]$ cd ~/work/arch-pc/lab07
[sesimko@fedora lab07]$ touch lab7-1.asm
[sesimko@fedora lab07]$
```

Создаем директорию для нашей лаб. работы и в ней создаем .asm файл.

```
%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
```

Пишем код.

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

Собираем и запускаем.

```

_start:
mov eax, 6
mov ebx, 4

```

Заменяем в коде символы

на числа.

```

[sesimko@fedora lab07]$ nasm -f elf lab7-1.asm
[sesimko@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[sesimko@fedora lab07]$ ./lab7-1

[sesimko@fedora lab07]$

```

Снова собираем и запускаем.

Открыть ▾

lab7-2.asm
~/work/arch-pc/lab07

lab7-1.asm

```

%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit

```

Создаем новый .asm и

пишем в него код.

```

[sesimko@fedora lab07]$ nasm -f elf lab7-2.asm
[sesimko@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[sesimko@fedora lab07]$ ./lab7-2
106
[sesimko@fedora lab07]$

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

```

Собираем и запускаем.

Заменяв символы на числа,

наконец, получаем интересующий нас результат!

```

add eax,ebx

call iprint
call quit

```

Заменяем в программе

функцию iprintLF на iprint

```
[sesimko@fedora lab07]$ nasm -f elf lab7-2.asm
[sesimko@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[sesimko@fedora lab07]$ ./lab7-2
10[sesimko@fedora lab07]$
```

Теперь вывод отличается от

предыдущего, в нем нет переноса строки.

```
%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
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
```

Пишем новую программу.

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

Собираем и запускаем.

Результат соответствует ожидаемому.


```

%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
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx
mov edi,eax
; ---- Вывод результата на экран
mov eax,div
call sprint
mov eax,edi |
call iprintLF
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit

```

Немного меняем формулу.

```

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

```

Получаем новый результат.

```
Открыть ▾ + variant.asm
~/work/arch-pc/lab07

lab7-1.asm | lab7-2.asm | lab7-3.asm

%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
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call iprintf
call quit
```

Пишем программу

генерации варианта на основе номера студ. билета.

```
[sesimko@fedora lab07]$ nasm -f elf variant.asm
[sesimko@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[sesimko@fedora lab07]$ ./variant
Введите No студенческого билета:
1132222841
Ваш вариант: 2
[sesimko@fedora lab07]$
```

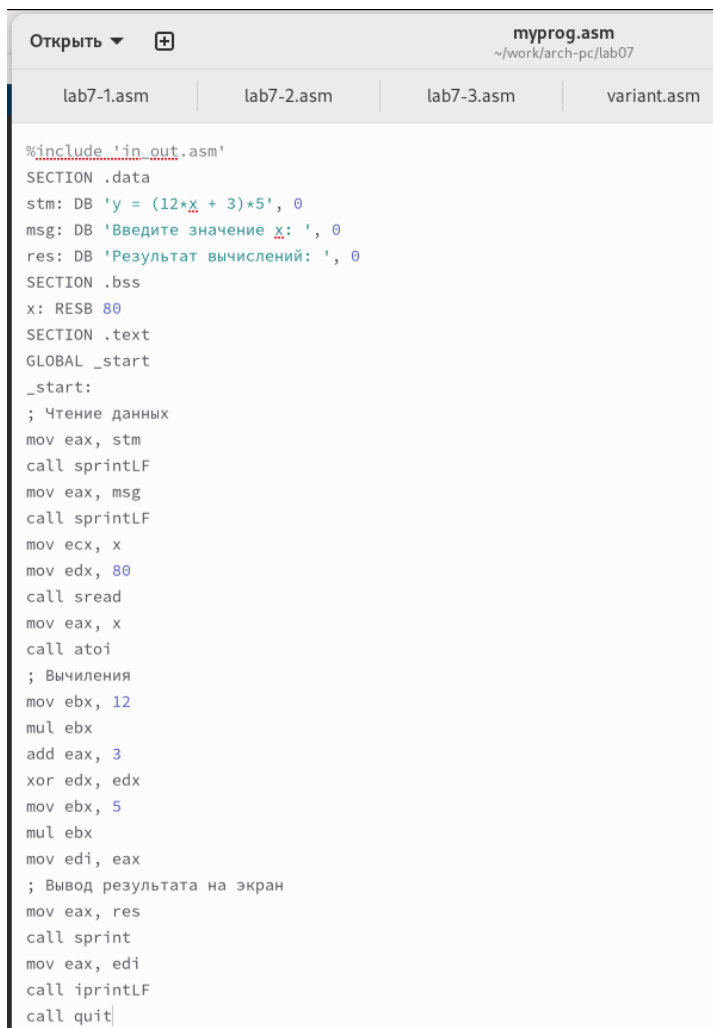
Собираем и запускаем.


Посчитав вручную, удостоверяемся, что результат правильный.

3 Ответы на вопросы

1. Первые две строки во входной функции.
2. Инструкции отвечают за ввод студ. билета.
3. Для преобразования ASCII в число (aSCII to iNTEGER).
4. `xor edx,edx; mov ebx,20; div ebx; inc edx`
5. `edx`
6. Прибавить к значению в `edx` единицу.
7. `mov eax,rem; call sprint; mov eax,edx; call iprintLF`

4 Самостоятельная работа



```
Открыть ▾  myprog.asm
~/work/arch-pc/lab07

lab7-1.asm | lab7-2.asm | lab7-3.asm | variant.asm

%include 'in_out.asm'
SECTION .data
stm: DB 'y = (12*x + 3)*5', 0
msg: DB 'Введите значение x: ', 0
res: DB 'Результат вычислений: ', 0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
; Чтение данных
mov eax, stm
call sprintLF
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
; Вычисления
mov ebx, 12
mul ebx
add eax, 3
xor edx, edx
mov ebx, 5
mul ebx
mov edi, eax
; Вывод результата на экран
mov eax, res
call sprint
mov eax, edi
call iprintLF
call quit
```

Пишем программу

```

1 [sesimko@fedora lab07]$ nasm -f elf myprog.asm
2 [sesimko@fedora lab07]$ ld -m elf_i386 -o myprog myprog.o
3 [sesimko@fedora lab07]$ ./myprog
4 y = (12*x + 3)*5
5 Введите значение x:
6 1
7 Результат вычислений: 75
8 [sesimko@fedora lab07]$ ./myprog
9 y = (12*x + 3)*5
10 Введите значение x:
11 6
12 Результат вычислений: 375
13 [sesimko@fedora lab07]$

```

Проверяем результат для

интересующих значений

5 Выводы

Мы изучили арифметические операции в `asm`.