

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

дисциплина: Архитектура компьютера

Шурыгин Илья Максимович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Ответы на вопросы:	11
5	Задание для самостоятельной работы:	13
6	Выводы	15

Список иллюстраций

3.1	Создали каталог lab07 в Midnight Commander и файл lab7-1.asm .	7
3.2	Вывод программы символа j	8
3.3	Вывод программы символа переноса строки	8
3.4	Вывод программы числа 106	8
3.5	Вывод программы числа 10	9
3.6	Вывод программы числа 10 без переноса строки	9
3.7	Результат работы программы $f(x) = (5 * 2 + 3)/3$	9
3.8	Результат работы программы $f(x) = (4 * 6 + 2)/5$	9
3.9	Результат работы программы вычисления варианта	10
5.1	Результат работы программы на тестах	13
5.2	Код программы	14

Список таблиц

1 Цель работы

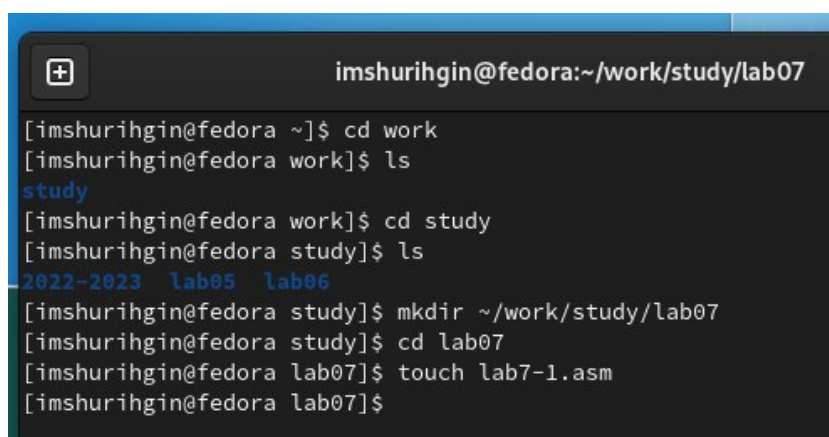
Приобрести практические навыки работы в Midnight Commander, освоить арифметические инструкции языка ассемблера NASM.

2 Задание

Необходимо написать программы, которые вычисляют функции.

3 Выполнение лабораторной работы

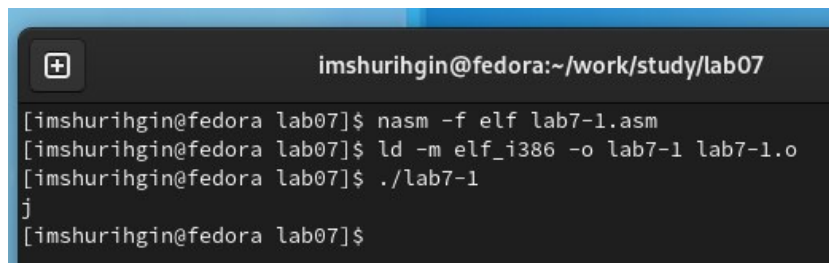
1. Откройте Midnight Commander и перейдем в каталог ~/work/study. Создадим каталог для программ лабораторной работы N7, перейдем в него и создадим файл lab7-1.asm.(рис. 3.1)



```
imshurihgin@fedora:~/work/study/lab07
[imshurihgin@fedora ~]$ cd work
[imshurihgin@fedora work]$ ls
study
[imshurihgin@fedora work]$ cd study
[imshurihgin@fedora study]$ ls
2022-2023 lab05 lab06
[imshurihgin@fedora study]$ mkdir ~/work/study/lab07
[imshurihgin@fedora study]$ cd lab07
[imshurihgin@fedora lab07]$ touch lab7-1.asm
[imshurihgin@fedora lab07]$
```

Рис. 3.1: Создали каталог lab07 в Midnight Commander и файл lab7-1.asm

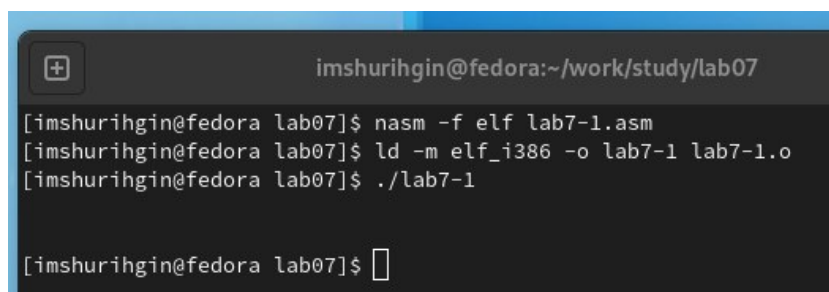
2. Запишем в файл lab6-1.asm код программы вывода значения регистра eax и запустим исполняемый файл. Выведется символ j, так как код символа 6 равен 00110110 в двоичном представлении(или 54 в десятичном представлении),а код символа 4 – 00110100(52), что в сумме даст 106, а это символ j по таблице ASCII.(рис. 3.2)



```
imshurihgin@fedora:~/work/study/lab07
[imshurihgin@fedora lab07]$ nasm -f elf lab7-1.asm
[imshurihgin@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[imshurihgin@fedora lab07]$ ./lab7-1
j
[imshurihgin@fedora lab07]$
```

Рис. 3.2: Вывод программы символа j

3. Изменим текст программы lab7-1.asm(вместо символов, запишем в регистры числа). Программа выведет символ с кодом 10 - перенос строки.(рис. 3.3)

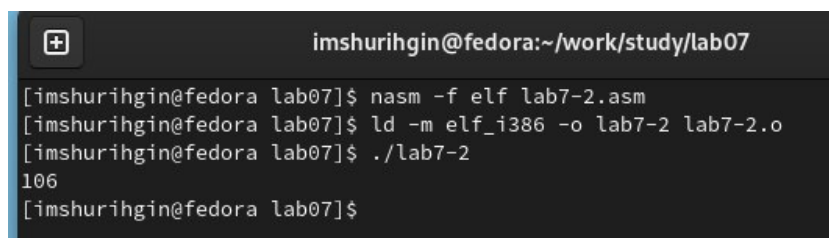


```
imshurihgin@fedora:~/work/study/lab07
[imshurihgin@fedora lab07]$ nasm -f elf lab7-1.asm
[imshurihgin@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[imshurihgin@fedora lab07]$ ./lab7-1

[imshurihgin@fedora lab07]$
```

Рис. 3.3: Вывод программы символа переноса строки

4. Создадим файл lab7-2.asm, в котором используем функцию iprintLF. В результате работы программы мы получим число 106.(рис. 3.4)



```
imshurihgin@fedora:~/work/study/lab07
[imshurihgin@fedora lab07]$ nasm -f elf lab7-2.asm
[imshurihgin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[imshurihgin@fedora lab07]$ ./lab7-2
106
[imshurihgin@fedora lab07]$
```

Рис. 3.4: Вывод программы числа 106

5. Изменим текст программы lab7-2.asm(изменим символы на числа). В результате работы программы мы получим число 10. Далее заменим функцию iprintLF на iprint, число 10 будет выводиться без переноса строки.(рис. 3.5)(рис. 3.6)


```
imshurihgin@fedora:~/work/study/lab07
[imshurihgin@fedora lab07]$ nasm -f elf lab7-2.asm
[imshurihgin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[imshurihgin@fedora lab07]$ ./lab7-2
10
[imshurihgin@fedora lab07]$
```

Рис. 3.5: Вывод программы числа 10

```
imshurihgin@fedora:~/work/study/lab07
[imshurihgin@fedora lab07]$ nasm -f elf lab7-2.asm
[imshurihgin@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[imshurihgin@fedora lab07]$ ./lab7-2
10[imshurihgin@fedora lab07]$
```

Рис. 3.6: Вывод программы числа 10 без переноса строки

6. Создадим файл lab7-3.asm и запишем в него программу вычисления выражения $f(x) = (5 * 2 + 3)/3$. Затем изменим текст программы так, чтобы она вычисляла выражение $f(x) = (4 * 6 + 2)/5$. (рис. 3.7)(рис. 3.8)

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

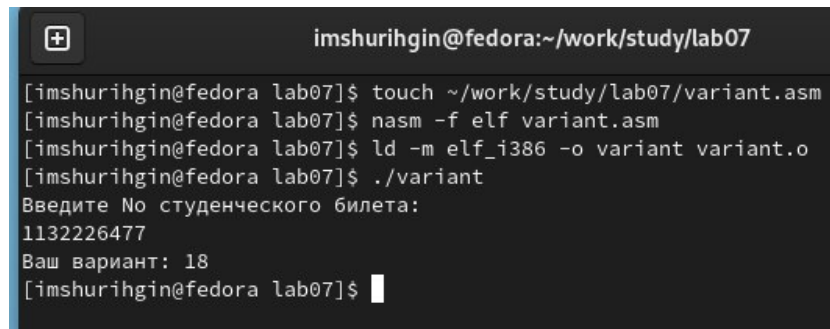
Рис. 3.7: Результат работы программы $f(x) = (5 * 2 + 3)/3$

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

Рис. 3.8: Результат работы программы $f(x) = (4 * 6 + 2)/5$

7. Создадим программу вычисления варианта для самостоятельной работы.

Создадим файл `variant.asm` и скопируем в нее код.(рис. 3.9)



```
imshurihgin@fedora:~/work/study/lab07
[imshurihgin@fedora lab07]$ touch ~/work/study/lab07/variant.asm
[imshurihgin@fedora lab07]$ nasm -f elf variant.asm
[imshurihgin@fedora lab07]$ ld -m elf_i386 -o variant variant.o
[imshurihgin@fedora lab07]$ ./variant
Введите No студенческого билета:
1132226477
Ваш вариант: 18
[imshurihgin@fedora lab07]$
```

Рис. 3.9: Результат работы программы вычисления варианта

4 Ответы на вопросы:

1. За вывод на экран сообщения 'Ваш вариант:' отвечают строки:

- `mov eax, msg`
- `call sprintLF`

2. Инструкция `mov ecx, x` используется для записывания в регистр `ecx` записываются данные из `x`. Инструкция `mov edx, 80` используется для указания длины переменной `x`. Инструкция `call sread` используется для считывания введенной информации.

3. Инструкция `call atoi` используется для преобразования ASCII кода в число.

4. За вычисления варианта отвечают:

- `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 прибавления 1.

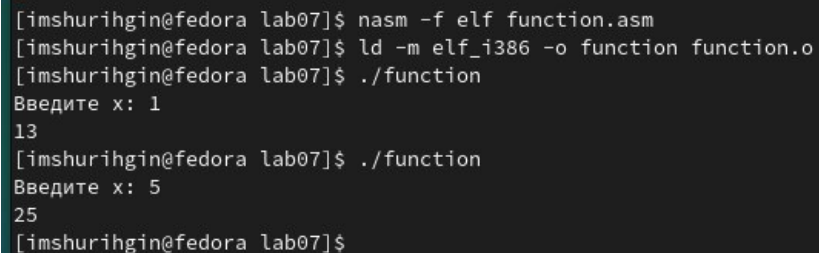
7. За вывод на экран результата вычислений отвечают:

- `mov eax,rem`

- call sprint
- mov eax,edx
- call iprintLF

5 Задание для самостоятельной работы:

1. Создадим файл `function.asm`, который будет вычислять функцию - $3(x+10)-20$. Проверим работу программы на тестах: $x = 1$, $x = 5$. (рис. 5.1)(рис. 5.2)



```
[imshurihgin@fedora lab07]$ nasm -f elf function.asm
[imshurihgin@fedora lab07]$ ld -m elf_i386 -o function function.o
[imshurihgin@fedora lab07]$ ./function
Введите x: 1
13
[imshurihgin@fedora lab07]$ ./function
Введите x: 5
25
[imshurihgin@fedora lab07]$
```

Рис. 5.1: Результат работы программы на тестах

```
mc [imshurihgin@fedora]:~/work/study/lab07
function.asm [----] 9 L: [ 1+ 9 10/ 28] *(208 / 682b)
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите x: ', 0h
div: DB 'Результат: ', 0

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax, buf1
call atoi
add eax, 10 ; EAX=EAX+10
mov ebx, 3
mul ebx
sub eax, 20
mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, edi ; вызов подпрограммы печати
call iprintLF
call quit ; вызов подпрограммы завершения

1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~
```

Рис. 5.2: Код программы

6 Выводы

В данной лабораторной работе я научился писать простые программы на языке ассемблера NASM, а именно: создал программу вычисления функции.