

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

дисциплина: Архитектура компьютера

Шурыгин Илья Максимович

Содержание

| | | |
|----------|--|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Задание для самостоятельной работы: | 11 |
| 5 | Выводы | 15 |

Список иллюстраций

| | | |
|-----|--|----|
| 3.1 | Создали каталог lab08 в Midnight Commander и файл lab8-1.asm . | 7 |
| 3.2 | Вывод программы, которая использует инструкцию jmp | 7 |
| 3.3 | Код программы | 8 |
| 3.4 | Вывод программы | 8 |
| 3.5 | Вычисление наибольшей из 3 целочисленных переменных | 9 |
| 3.6 | Файл листинга программы lab8-2.asm | 10 |
| 3.7 | Выполняем трансляцию с получением файла листинга | 10 |
| 3.8 | Описание ошибки | 10 |
| 4.1 | Результат работы программы | 11 |
| 4.2 | Код программы | 12 |
| 4.3 | Результат работы программы | 13 |
| 4.4 | Код программы | 14 |

Список таблиц

1 Цель работы

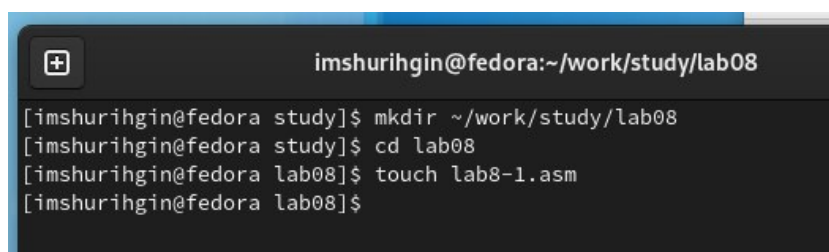
Изучить команды условного и безусловного переходов(инструкция jmp).

2 Задание

Необходимо написать программы, которые вычисляют значение функции, зависящее от вводных данных.

3 Выполнение лабораторной работы

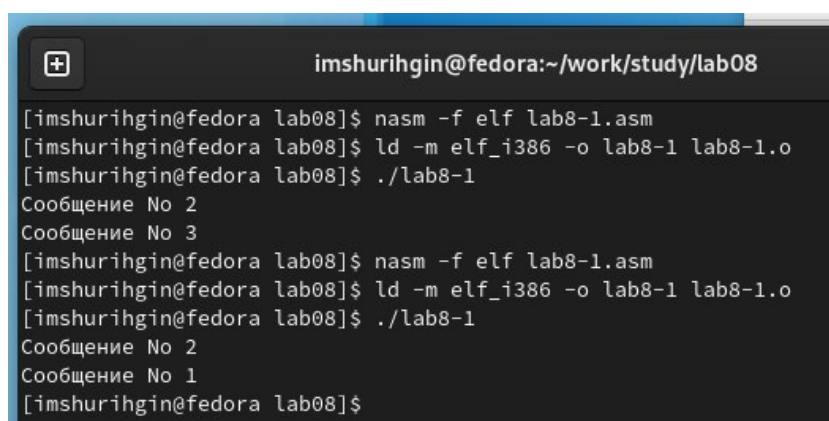
1. Откройте Midnight Commander и перейдем в каталог ~/work/study. Создадим каталог для программ лабораторной работы N8, перейдем в него и создадим файл lab8-1.asm.(рис. 3.1)



```
imshurihgin@fedora:~/work/study/lab08
[imshurihgin@fedora study]$ mkdir ~/work/study/lab08
[imshurihgin@fedora study]$ cd lab08
[imshurihgin@fedora lab08]$ touch lab8-1.asm
[imshurihgin@fedora lab08]$
```

Рис. 3.1: Создали каталог lab08 в Midnight Commander и файл lab8-1.asm

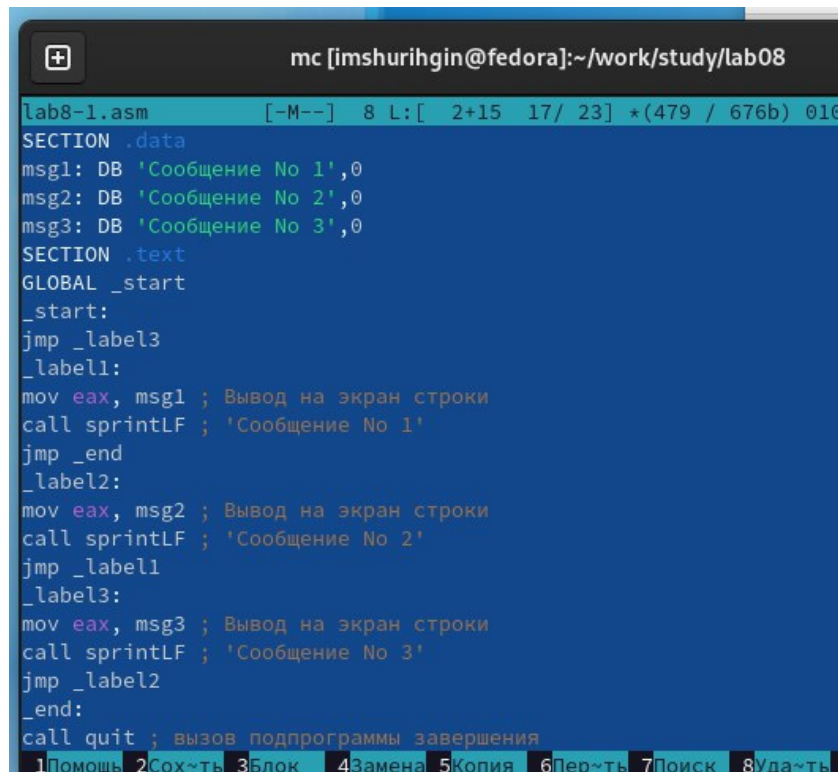
2. Запишем в файл lab8-1.asm текст программы из листинга 8.1, она будет выводить: сообщение 2, затем - 3. Изменим текст программы на код из листинга 8.2. Теперь программа выводит: сообщение 2, затем - 1.(рис. 3.2)



```
imshurihgin@fedora:~/work/study/lab08
[imshurihgin@fedora lab08]$ nasm -f elf lab8-1.asm
[imshurihgin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[imshurihgin@fedora lab08]$ ./lab8-1
Сообщение No 2
Сообщение No 3
[imshurihgin@fedora lab08]$ nasm -f elf lab8-1.asm
[imshurihgin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[imshurihgin@fedora lab08]$ ./lab8-1
Сообщение No 2
Сообщение No 1
[imshurihgin@fedora lab08]$
```

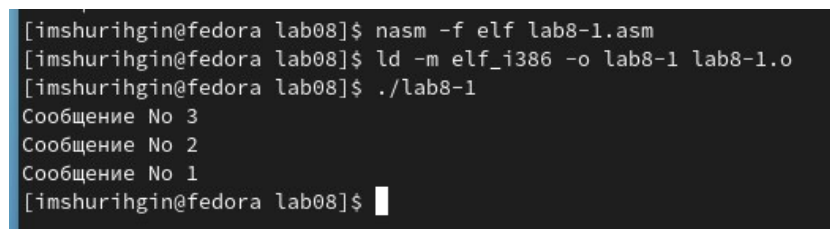
Рис. 3.2: Вывод программы, которая использует инструкцию jmp

3. Изменим текст программы lab8-1.asm так, чтобы вывод программы был следующим: сообщение 3, затем - 2, потом - 1.(рис. 3.3)(рис. 3.4)



```
mc [imshurihgin@fedora]:~/work/study/lab08
lab8-1.asm  [-M--]  8 L:[ 2+15 17/ 23] *(479 / 676b) 010
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение No 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение No 3'
jmp _label2
_end:
call quit ; вызов подпрограммы завершения
```

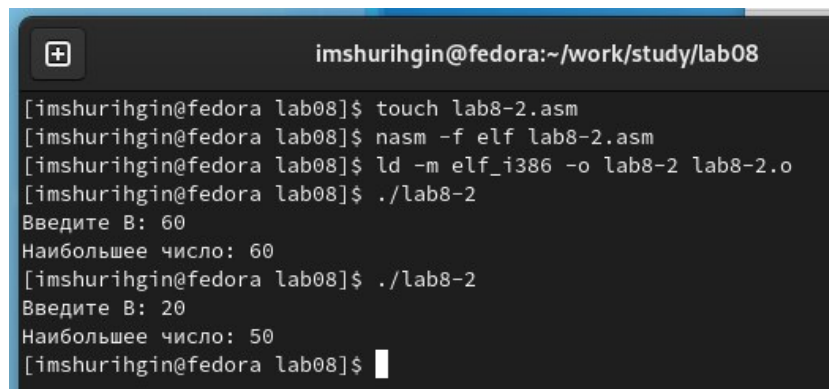
Рис. 3.3: Код программы



```
[imshurihgin@fedora lab08]$ nasm -f elf lab8-1.asm
[imshurihgin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[imshurihgin@fedora lab08]$ ./lab8-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
[imshurihgin@fedora lab08]$
```

Рис. 3.4: Вывод программы

4. Создадим файл lab8-2.asm и запишем в него код из листинга 8.3(Программа, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C).(рис. 3.5)

A terminal window with a dark background and light text. The title bar shows the user 'imshurihgin' on a 'fedora' machine in the directory '~/work/study/lab08'. The terminal shows the following commands and output:

```
[imshurihgin@fedora lab08]$ touch lab8-2.asm
[imshurihgin@fedora lab08]$ nasm -f elf lab8-2.asm
[imshurihgin@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[imshurihgin@fedora lab08]$ ./lab8-2
Введите B: 60
Наибольшее число: 60
[imshurihgin@fedora lab08]$ ./lab8-2
Введите B: 20
Наибольшее число: 50
[imshurihgin@fedora lab08]$
```

Рис. 3.5: Вычисление наибольшей из 3 целочисленных переменных

5. Создадим файл листинга для программы из файла lab8-2.asm с помощью команды: `nasm -f elf -l lab8-2.lst lab8-2.asm`. (рис. 3.6)

На строке 14 записан ее номер, адрес - 000000E8, машинный код - B8000000 и сама команда - `mov eax,msg1`, которая записывает содержимое msg1 в ячейку памяти eax.

На строке 15 записан ее номер, адрес - 000000ED, машинный код - E81DFFFFFF и сама команда - `call sprint`, которая выводит данные

На строке 19 записан ее номер, адрес - 000000FC, машинный код - E842FFFFFF и сама команда - `call sread`, которая считывает входные данные

```

imshurihgin@fedora:~/work/study/lab08 — /usr/bin/mc -P /var/...
lab8-2.lst [----] 0 L:[179+ 0 179/225] *(10993/14458b) 0032 0x020[*]
4 0000002E D0BBD0BE3A2000....
5 00000035 32300000          A dd '20'
6 00000039 35300000          C dd '50'
7                                section .bss
8 00000000 <res Ah>          max resb 10
9 0000000A <res Ah>          B resb 10
10                               section .text
11                               global _start
12                               _start:
13                               ; ----- Вывод сообщения 'Введите
14 000000E8 B8[00000000]      mov eax,msg1
15 000000ED E81DFFFFFF      call sprint
16                               ; ----- Ввод 'B'
17 000000F2 B9[0A000000]      mov ecx,B
18 000000F7 BA0A000000      mov edx,10
19 000000FC E842FFFFFF      call sread
20                               ; ----- Преобразование 'B' из с
21 00000101 B8[0A000000]      mov eax,B
22 00000106 E891FFFFFF      call atoi ; Вызов подпрограммы перево
23 0000010B A3[0A000000]      mov [B],eax ; запись преобразованного
24                               ; ----- Записываем 'A' в перемен
25 00000110 8B0D[35000000]      mov ecx,[A] ; 'ecx = A'
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Выдать 9МенюMC10Вых

```

Рис. 3.6: Файл листинга программы lab8-2.asm

- В файле с программой lab8-2.asm удалим один операнд у инструкции с двумя операндами. На строке 18 удалим операнд у mov и оставим только edx. Попытаемся выполнить трансляцию с получением файла листинга.(рис. 3.7)(рис. 3.8)

```

imshurihgin@fedora:~/work/study/lab08
[imshurihgin@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
[imshurihgin@fedora lab08]$ mcedit lab8-2.lst

[imshurihgin@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
lab8-2.asm:18: error: invalid combination of opcode and operands
[imshurihgin@fedora lab08]$

```

Рис. 3.7: Выполняем трансляцию с получением файла листинга

```

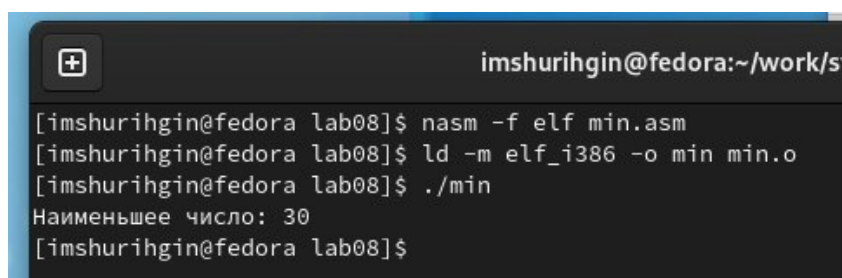
16                               ; ----- Ввод 'B'
17 000000F2 B9[0A000000]      mov ecx,B
18                               mov edx
18                               *****
18                               error: invalid combination of opcode and operands
19 000000F7 E847FFFFFF      call sread
20                               ; ----- Преобразование 'B' из с

```

Рис. 3.8: Описание ошибки

4 Задание для самостоятельной работы:

1. Напишем программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Данные значения: 83,73,30.(рис. 4.1)(рис. 4.2)



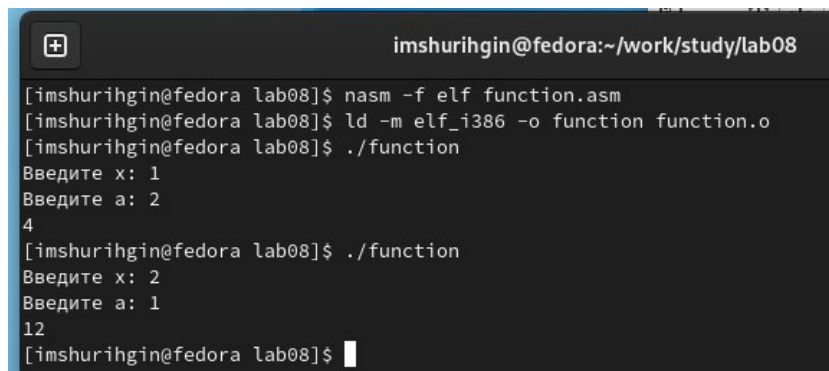
```
imshurihgin@fedora:~/work/s
[imshurihgin@fedora lab08]$ nasm -f elf min.asm
[imshurihgin@fedora lab08]$ ld -m elf_i386 -o min min.o
[imshurihgin@fedora lab08]$ ./min
Наименьшее число: 30
[imshurihgin@fedora lab08]$
```

Рис. 4.1: Результат работы программы

```
mc [imshurihgin@fedora]:~/work/study/lab
min.asm [----] 11 L:[ 1+ 8 9/ 45] *(169 /1419b) 0010
#include 'in_out.asm'
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наименьшее число: ",0h
A dd '83'
B dd '73'
C dd '30'
section .bss
max resb 10
section .text
global _start
_start:
; ----- Преобразование 'A,B,C'
mov eax,A
call atoi ; Вызов подпрограммы перевода символа в число
mov [A],eax ; запись преобразованного числа в 'A'
mov eax,B
call atoi
mov [B],eax
mov eax,C
call atoi
mov [C], eax
; ----- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C'
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jnb check_B ; если 'A<C', то переход на метку 'check_B',
mov ecx,[C] ; иначе 'ecx = C'
mov [max],ecx ; 'max = C'
; ----- Преобразование 'max(A,C)' из символа в число
check_B:
```

Рис. 4.2: Код программы

2. Напишем программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Данная функция выводит a^2 , $a \neq 1$ и $10 + x$, $a = 1$.(рис. 4.3)(рис. 4.4)



```
imshurihgin@fedora:~/work/study/lab08
[imshurihgin@fedora lab08]$ nasm -f elf function.asm
[imshurihgin@fedora lab08]$ ld -m elf_i386 -o function function.o
[imshurihgin@fedora lab08]$ ./function
Введите x: 1
Введите a: 2
4
[imshurihgin@fedora lab08]$ ./function
Введите x: 2
Введите a: 1
12
[imshurihgin@fedora lab08]$
```

Рис. 4.3: Результат работы программы

```
mc [imshurihgin@fedora]:~/work/study/lab0
function.asm [----] 11 L: [ 1+ 7 8/ 45] *(199 / 645b) 0010 0
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Введите x: ',0
msg2: DB 'Введите a: ',0
section .bss
X resb 10
A resb 10
res resb 10
section .text
global _start
_start:
mov eax, msg1
call sprint
mov ecx, X
mov edx, 10
call sread
mov eax, X
call atoi
mov [X], eax
mov eax, msg2
call sprint
mov ecx, A
mov edx, 10
call sread
mov eax, A
call atoi
mov [A], eax

mov ecx, [A]
cmp ecx, 1
je _check_F
mov eax, [A]
mul eax
mov [res], eax
jmp _end

_check_F:
mov eax, [X]
add eax, 10
mov [res], eax

_end:
mov eax, [res]
call iprintLF
call quit ; вызов подпрограммы завершения
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск
```

Рис. 4.4: Код программы

5 Выводы

В данной лабораторной работе я научился писать простые программы на языке ассемблера NASM, а именно: создал программу которая вычисляет значение функции, зависящее от вводных данных и программу нахождения наименьшей из 3 целочисленных переменных.