Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Козин Иван Евгеньевич

Содержание

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Задание

1. Реализация переходов в NASM
2. Изучение структуры файлов листинга
3. Самостоятельное написание программ по материалам лабораторной работы

3 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов: • условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия. • безусловный переход – выполнение передачи управления в определенную точку про- граммы без каких-либо условий.

4 Выполнение лабораторной работы

4.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы №7 (рис. 1).

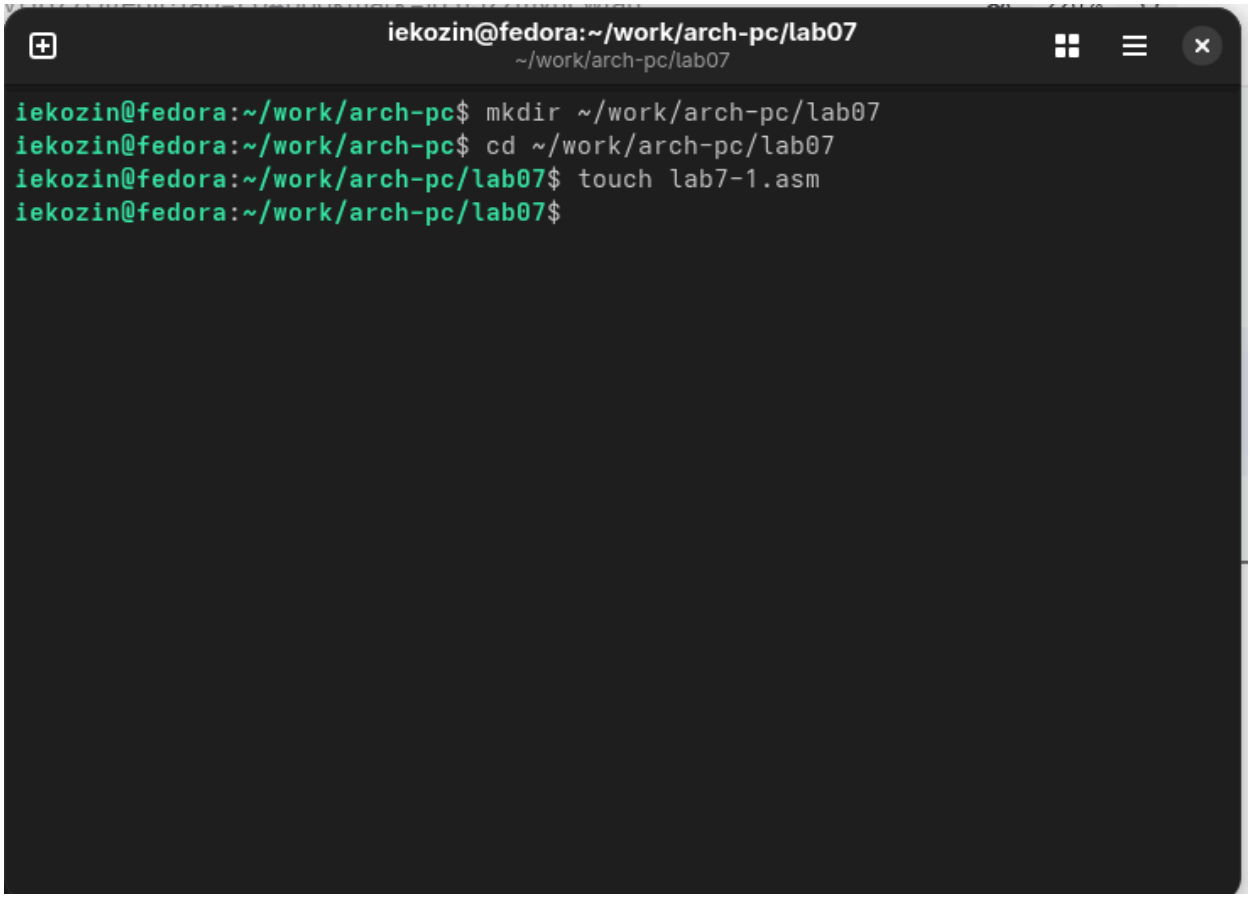
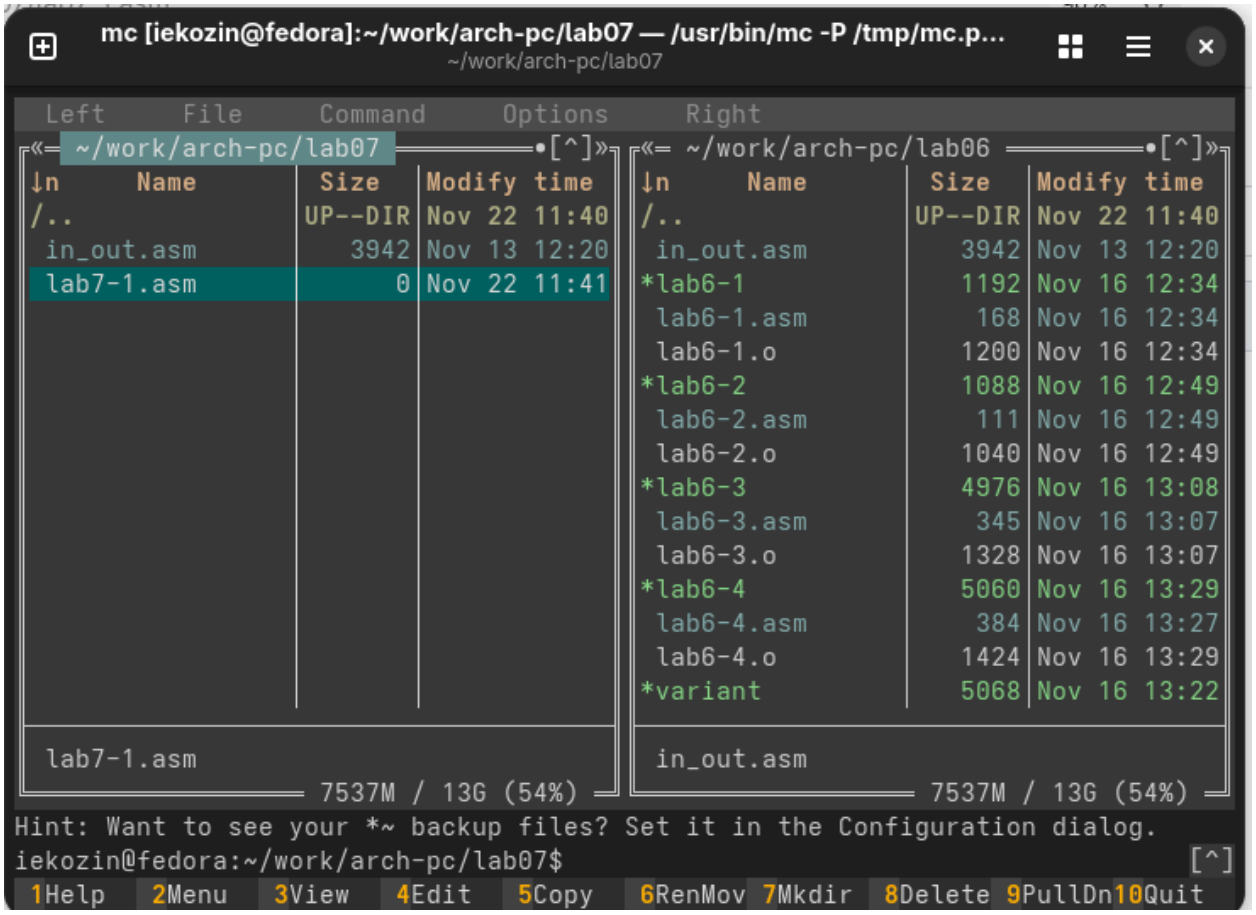


Рис. 1: Создание каталога и файла для программы

Копирую код из листинга в файл будущей программы. (рис. 2).


mc [iekozin@fedora]:~/work/arch-pc/lab07 — /usr/bin/mc -P /tmp/mc.p...
~/work/arch-pc/lab07
Left File Command Options Right
~/work/arch-pc/lab07
↓n Name Size Modify time
/.. UP--DIR Nov 22 11:40
in_out.asm 3942 Nov 13 12:20
lab7-1.asm 0 Nov 22 11:41
lab7-1.asm
7537M / 13G (54%)
~/work/arch-pc/lab06
↓n Name Size Modify time
/.. UP--DIR Nov 22 11:40
in_out.asm 3942 Nov 13 12:20
*lab6-1 1192 Nov 16 12:34
lab6-1.asm 168 Nov 16 12:34
lab6-1.o 1200 Nov 16 12:34
*lab6-2 1088 Nov 16 12:49
lab6-2.asm 111 Nov 16 12:49
lab6-2.o 1040 Nov 16 12:49
*lab6-3 4976 Nov 16 13:08
lab6-3.asm 345 Nov 16 13:07
lab6-3.o 1328 Nov 16 13:07
*lab6-4 5060 Nov 16 13:29
lab6-4.asm 384 Nov 16 13:27
lab6-4.o 1424 Nov 16 13:29
*variant 5068 Nov 16 13:22
in_out.asm
7537M / 13G (54%)
Hint: Want to see your *~ backup files? Set it in the Configuration dialog.
iekozin@fedora:~/work/arch-pc/lab07$
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

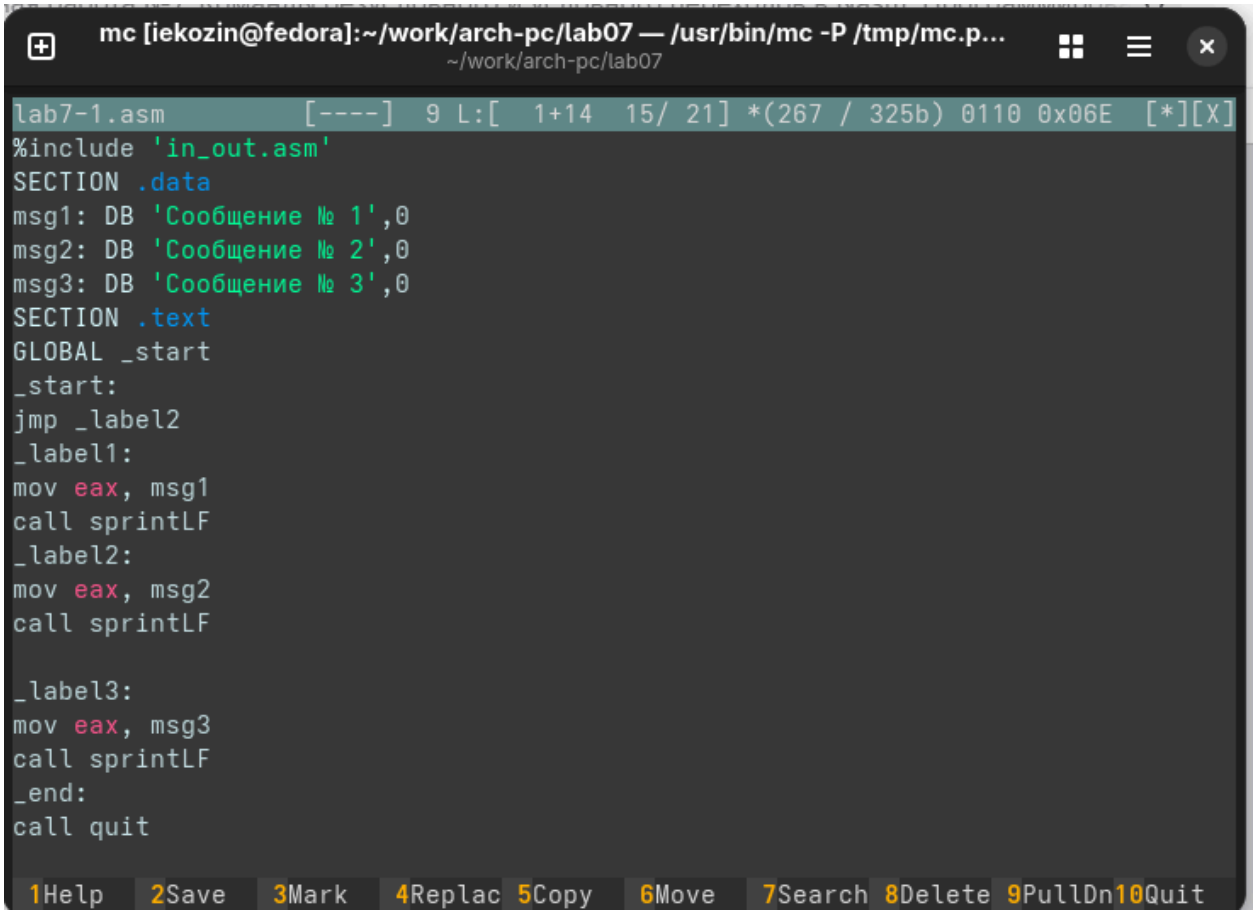


Рис. 2: Сохранение программы

При запуске программы я убедился в том, что неусловный переход действительно изменяет порядок выполнения инструкций (рис. 3).

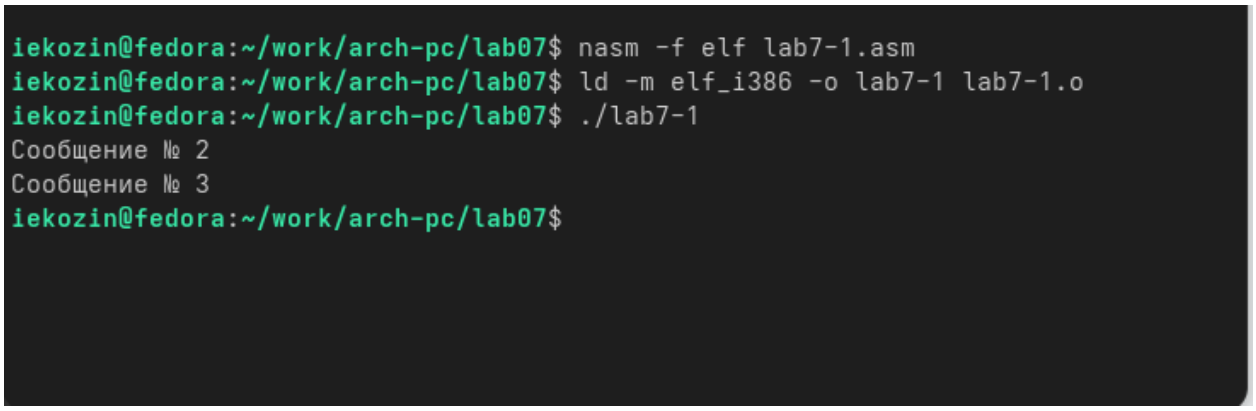


Рис. 3: Запуск программы

Изменяю программу таким образом, чтобы поменялся порядок выполнения функций (рис. 4).

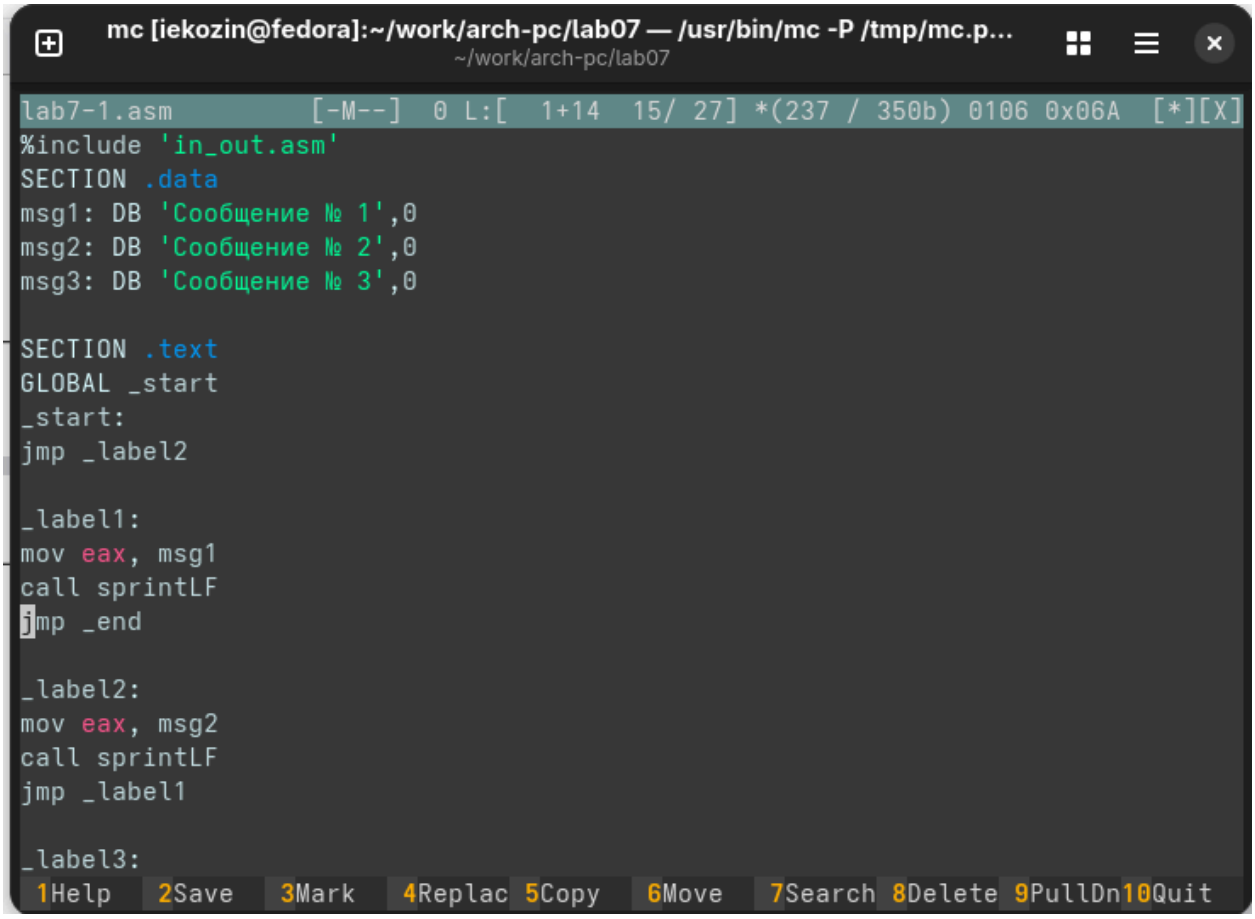


Рис. 4: Изменение программы

Запускаю программу и проверяю, что примененные изменения верны (рис. 5).

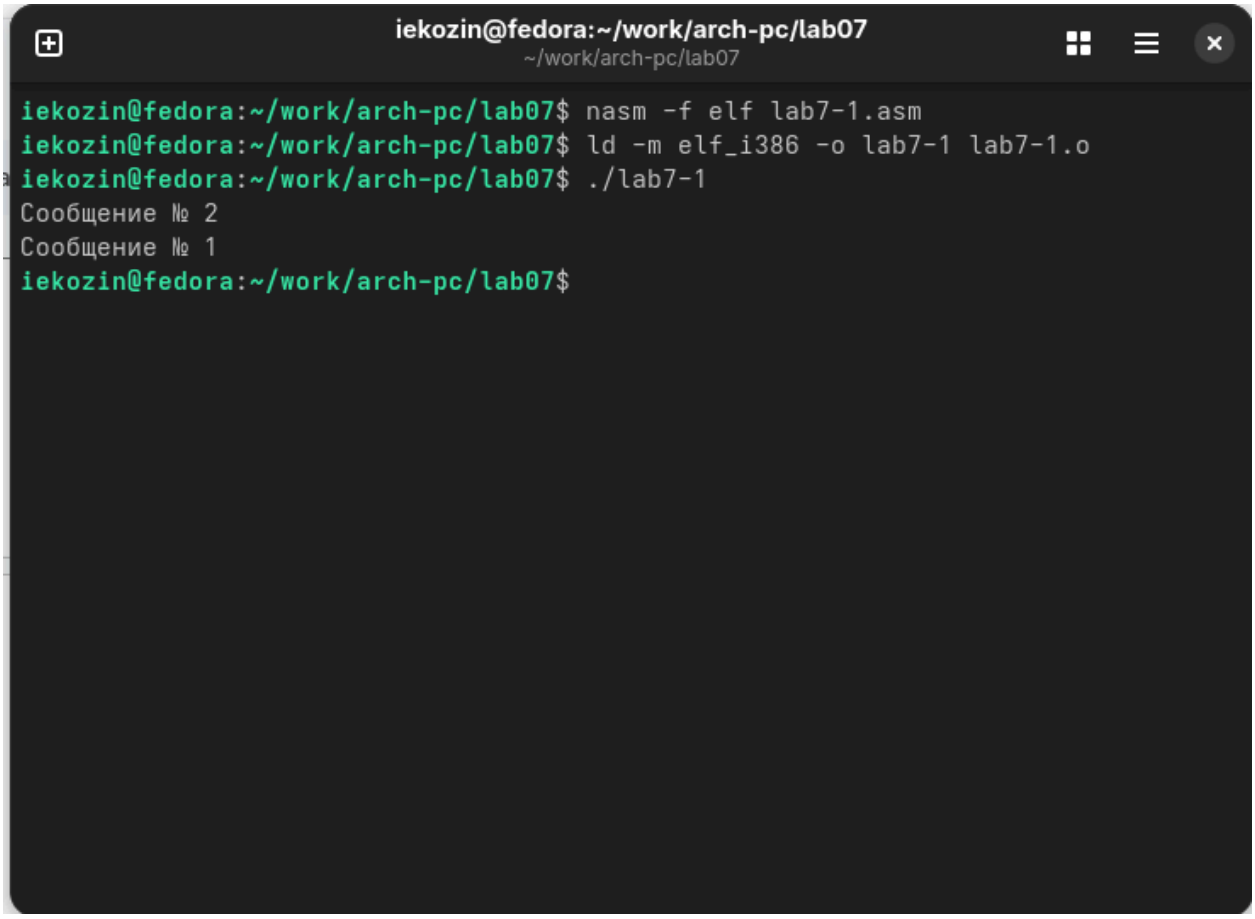


Рис. 5: Запуск измененной программы

Теперь изменяю текст программы так, чтобы все три сообщения вывелись в обратном порядке (рис. 6).

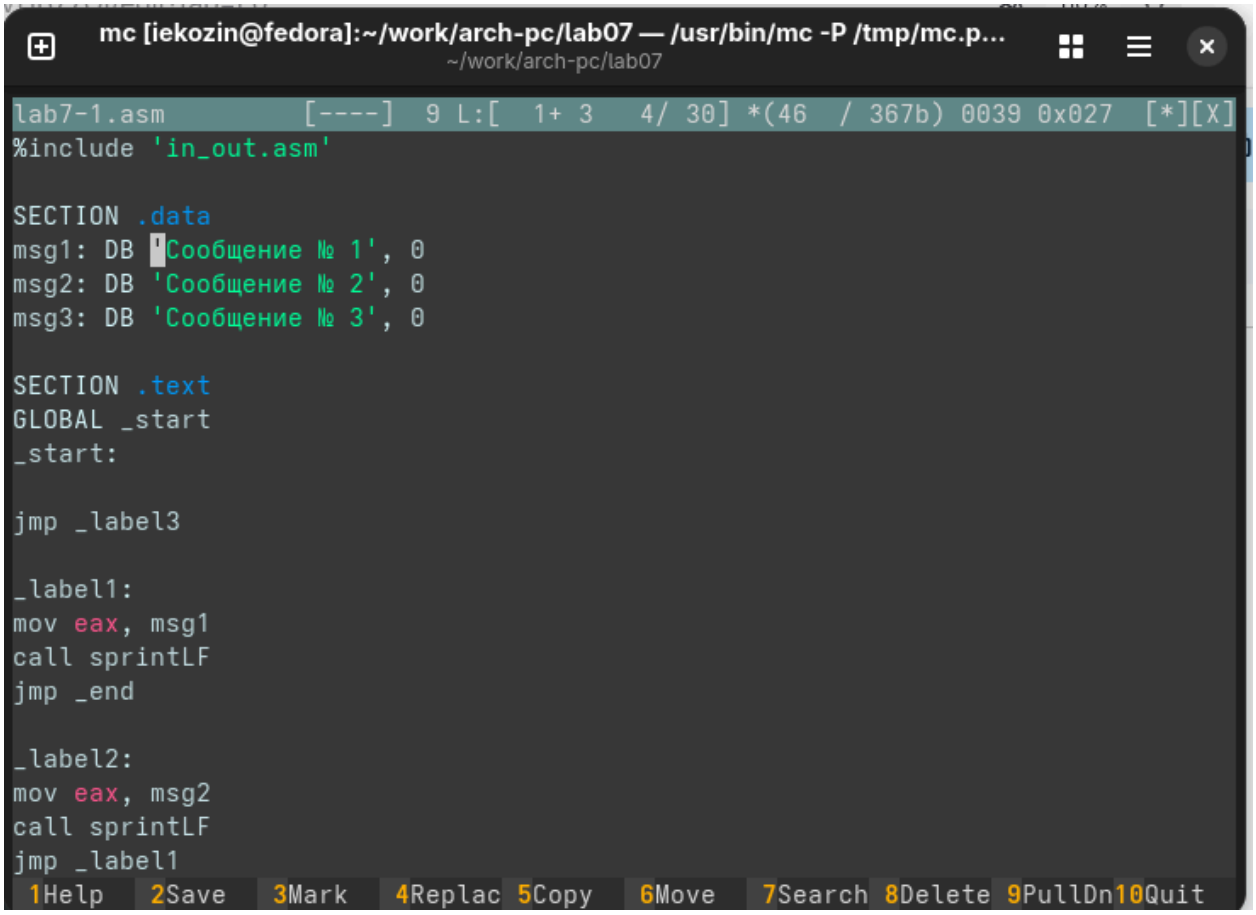


Рис. 6: Изменение программы

Работа выполнена корректно, программа в нужном мне порядке выводит сообщения (рис. 7).

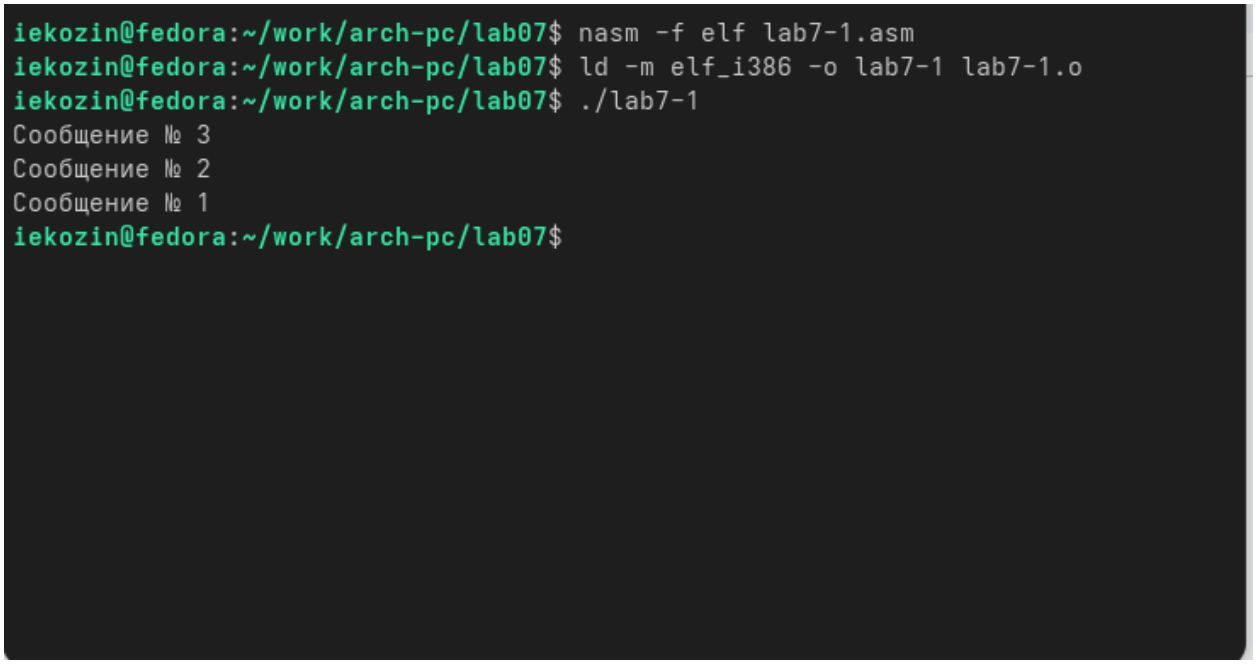


Рис. 7: Проверка изменений

Создаю новый рабочий файл и вставляю в него код из следующего листинга (рис. 8).

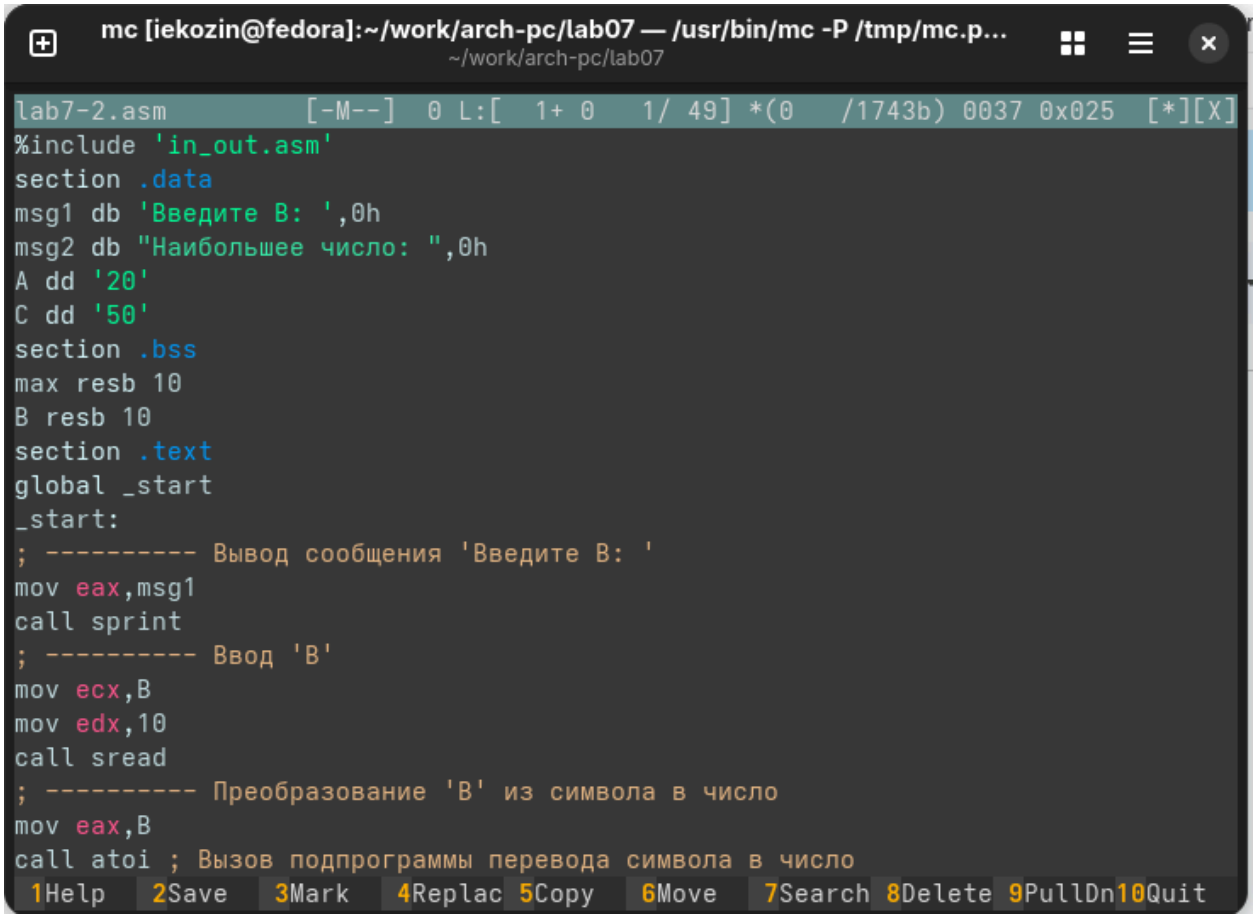


Рис. 8: Сохранение новой программы

Программа выводит значение переменной с максимальным значением, проверяю работу программы с разными входными данными (рис. 9).

```
iekozin@fedora:~/work/arch-pc/lab07$ touch lab7-2.asm
iekozin@fedora:~/work/arch-pc/lab07$ mc

iekozin@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
iekozin@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
iekozin@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 30
Наибольшее число: 50
iekozin@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 56
Наибольшее число: 56
iekozin@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 10
Наибольшее число: 50
iekozin@fedora:~/work/arch-pc/lab07$
```

Рис. 9: Проверка программы из листинга

4.2 Изучение структуры файла листинга

Создаю файл листинга с помощью флага -l команды nasm и открываю его с помощью текстового редактора mousepad (рис. 10).

```
iekozin@fedora:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
iekozin@fedora:~/work/arch-pc/lab07$ mc

iekozin@fedora:~/work/arch-pc/lab07$
```

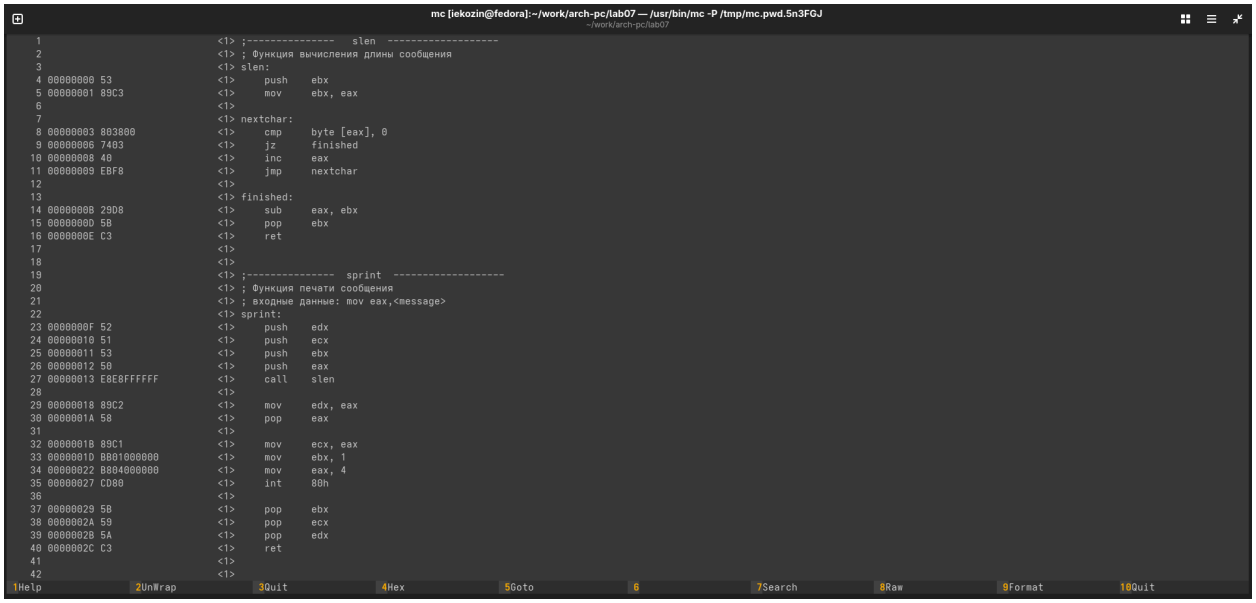


Рис. 10: Проверка файла листинга

Первое значение в файле листинга - номер строки, и он может вовсе не совпадать с номером строки изначального файла. Второе вхождение - адрес, смещение машинного кода относительно начала текущего сегмента, затем непосредственно идет сам машинный код, а заключает строку исходный текст прогарммы с комментариями.

Удаляю один операнд из случайной инструкции, чтобы проверить поведение файла листинга в дальнейшем (рис. 11).

Рис. 11: Удаление операнда из программы

В новом файле листинга показывает ошибку, которая возникла при попытке трансляции файла. Никакие выходные файлы при этом помимо файла листинга не создаются. (рис. 12).

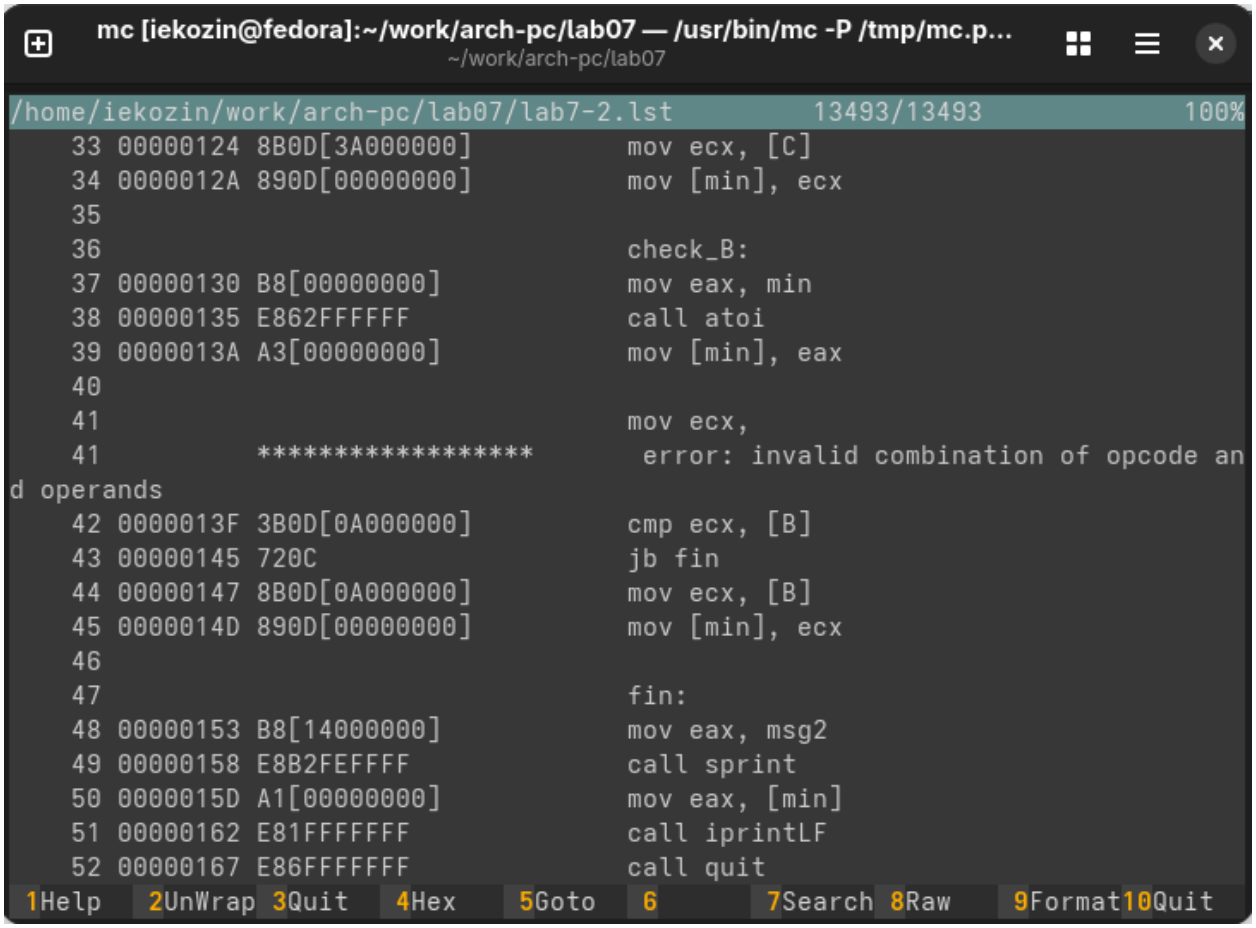
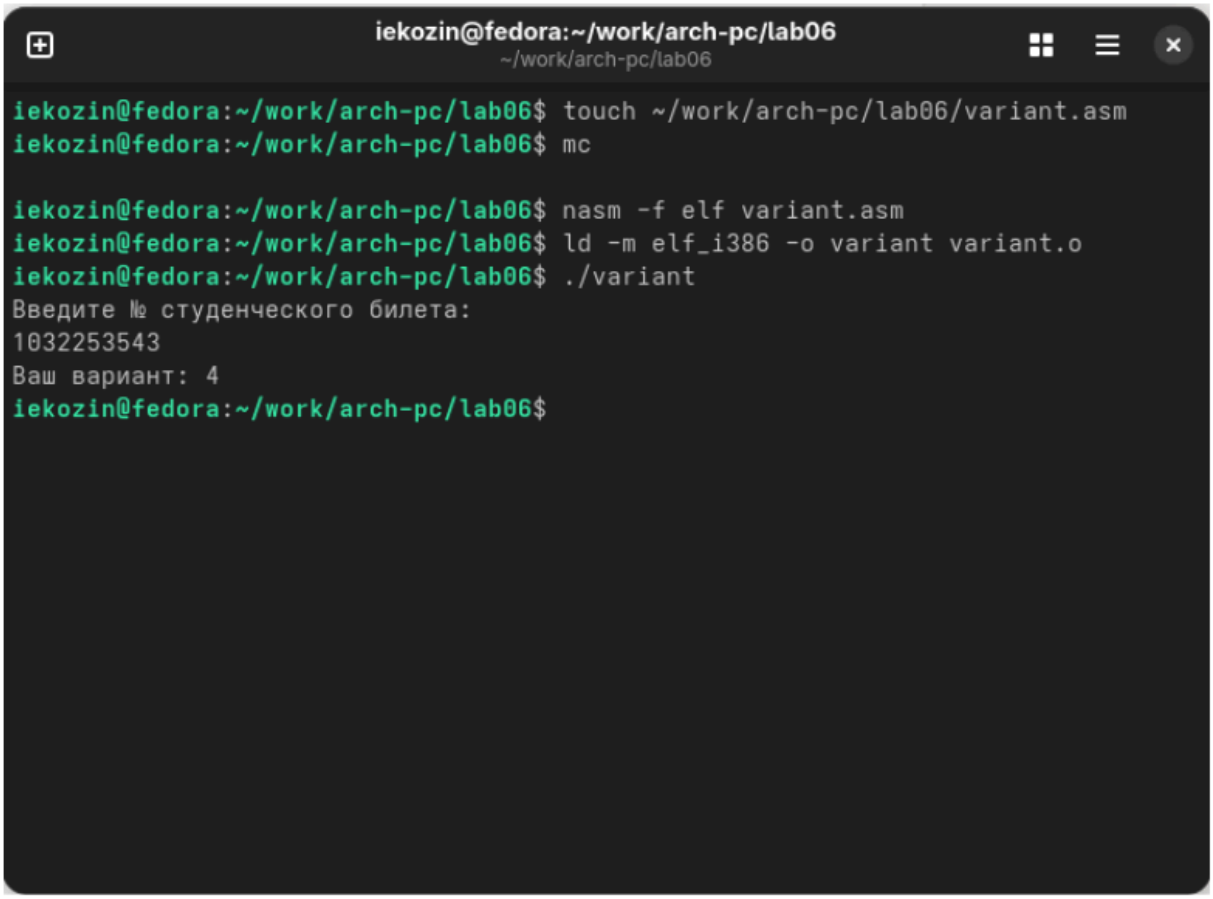


Рис. 12: Просмотр ошибки в файле листинга

4.3 Задания для самостоятельной работы

Возвращаю операнд к функции в программе и изменяю ее так, чтобы она выводила переменную с наименьшим значением (рис. 13).



Скорее всего мой вариант и в 7 лабе это четвертый

Рис. 13: Первая программа самостоятельной работы

Код первой программы:

```
%include "in_out.asm"

SECTION .data
msgA db "Введите A: ", 0
msgB db "Введите B: ", 0
msgC db "Введите C: ", 0
msgMin db "Наименьшее число: ", 0

SECTION .bss
A resb 10
B resb 10
C resb 10
min resd 1

SECTION .text
GLOBAL _start
_start:

; ===== Ввод A =====
mov eax, msgA
call sprint

mov ecx, A
mov edx, 10
call sread

mov eax, A
```

```
call atoi
mov [A], eax

; ===== Ввод B =====
mov eax, msgB
call sprint

mov ecx, B
mov edx, 10
call sread

mov eax, B
call atoi
mov [B], eax

; ===== Ввод C =====
mov eax, msgC
call sprint

mov ecx, C
mov edx, 10
call sread

mov eax, C
call atoi
mov [C], eax

; ===== Поиск минимума =====
```

```
mov eax, [A]     ; min = A
mov [min], eax

mov eax, [B]
cmp eax, [min]
jge check_C
mov [min], eax

check_C:
mov eax, [C]
cmp eax, [min]
jge finish
mov [min], eax

finish:
mov eax, msgMin
call sprint

mov eax, [min]
call iprintLF

call quit
```

Проверяю корректность написания первой программы (рис. 14).

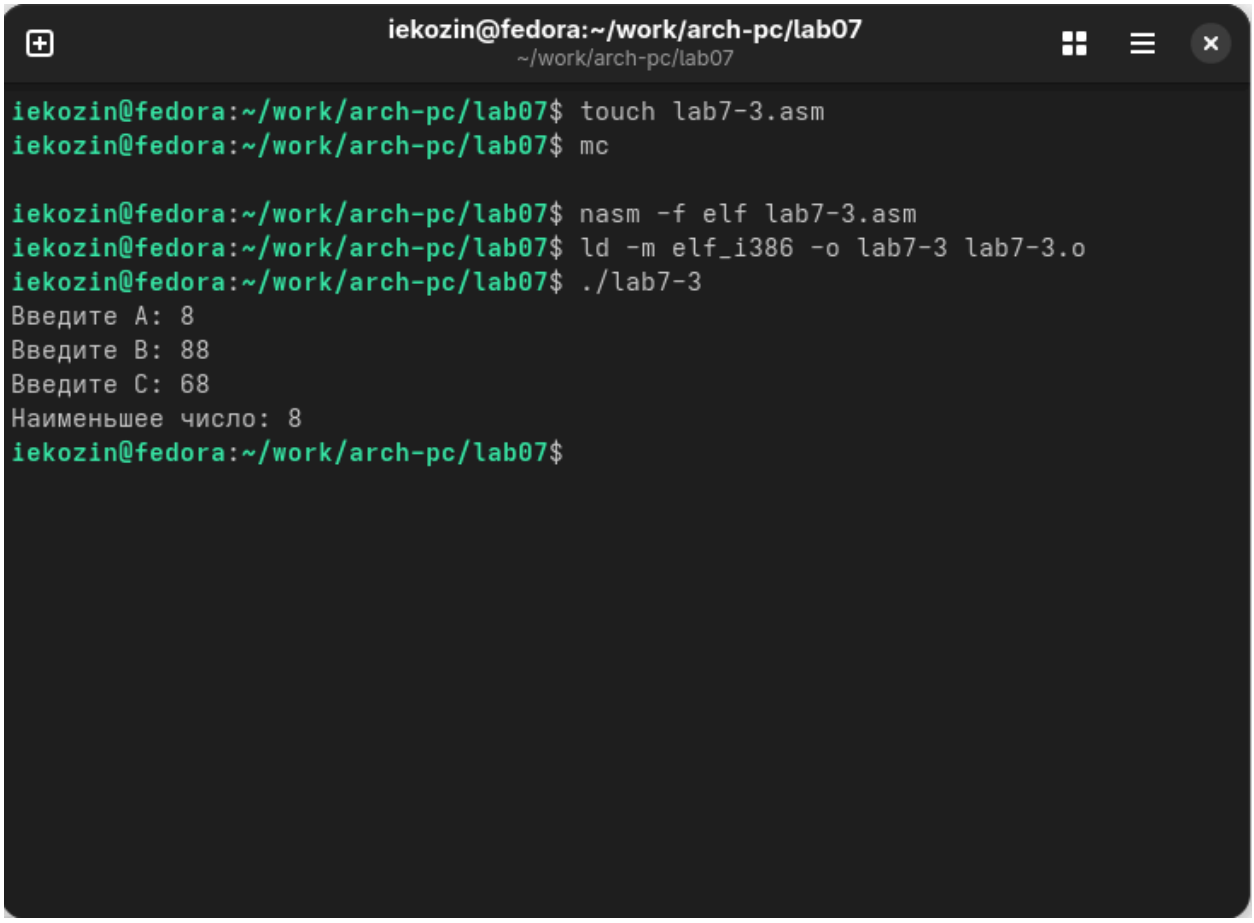


Рис. 14: Проверка работы первой программы

Пишу программу, которая будет вычислять значение заданной функции согласно моему варианту для введенных с клавиатурых переменных a и x (рис. 15).

Рис. 15: Вторая программа самостоятельной работы

Код второй программы:

```
%include 'in_out.asm'


SECTION .data_
msg_x: DB 'Введите значение переменной x: ', 0
msg_a: DB 'Введите значение переменной a: ', 0
res: DB 'Результат: ', 0


SECTION .bss
x: RESB 80
a: RESB 80


SECTION .text
GLOBAL _start
_start:
    ; Ввод x
    mov eax, msg_x
```

```
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov edi, eax          ; edi = x

; Ввод a
mov eax, msg_a
call sprint
mov ecx, a
mov edx, 80
call sread
mov eax, a
call atoi
mov esi, eax          ; esi = a

; Проверка a != 0
cmp esi, 0
je a_is_zero

; a != 0 => f(x) = 2*x + a
mov eax, edi          ; eax = x
shl eax, 1            ; eax = 2*x
add eax, esi          ; eax = 2*x + a
jmp print_result
```

```
a_is_zero:
    ; a == 0 => f(x) = 2*x + 1
    mov eax, edi          ; eax = x
    shl eax, 1            ; eax = 2*x
    add eax, 1            ; eax = 2*x + 1

print_result:
    mov edi, eax          ; сохраняем результат
    mov eax, res
    call sprint
    mov eax, edi
    call iprintLF

    call quit
```

Транслирую и компоную файл, запускаю и проверяю работу программмы для различных значений a и x (рис. 16).

```
iekozin@fedora:~/work/arch-pc/lab07$ touch lab7-4.asm
iekozin@fedora:~/work/arch-pc/lab07$ mc

iekozin@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-4.asm
iekozin@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-4 lab7-4.o
iekozin@fedora:~/work/arch-pc/lab07$ ./lab7-4
Введите значение переменной x: 3
Введите значение переменной a: 0
Результат: 0
iekozin@fedora:~/work/arch-pc/lab07$
```

```
iekozin@fedora:~/work/arch-pc/lab07$ ./lab7-4
Введите значение переменной x: 3
Введите значение переменной a: 2
Результат: 2
iekozin@fedora:~/work/arch-pc/lab07$
```

Рис. 16: Проверка работы второй программы

5 Выводы

При выполнении лабораторной работы я изучил команды условных и безусловных переходво, а также приобрел навыки написания программ с использованием перходов, познакомился с назначением и структурой файлов листинга.

Список литературы

1. Курс на ТУИС
2. Лабораторная работа №7
3. Программирование на языке ассемблера NASM Столяров А. В.