Лабораторная работа №8.

Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Боровиков Даниил Александрович

Содержание

# 1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навы- ков написания программ с использованием переходов. Знакомство с назначе- нием и структурой файла листинга.

# 2 Выполнение лабораторной работы

Создадим каталог для программам лабораторной работы № 8, перейдем в него и создадим файл lab8-1.asm(рис. 1)

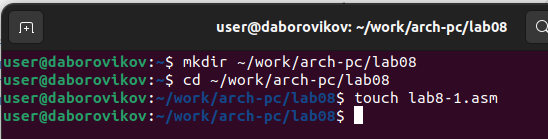


Рис. 1: Создание файла lab8-1.asm в соответствующем каталоге

Введем в файл lab8-1.asm текст программы из листинга 8.1.(рис. 2)

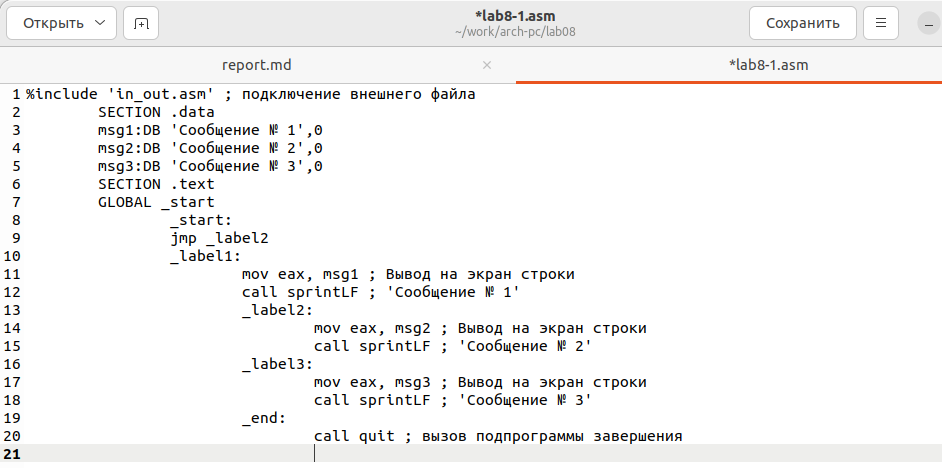


Рис. 2: Текст программы из листинга 8.1.

Создадим исполняемый файл и запустим его.(рис. 3)

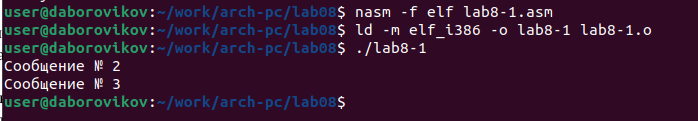


Рис. 3: Запуск исполняемого файла lab8-1.asm

Далее изменим текст программы в соответствии с листингом 8.2(рис. 4)

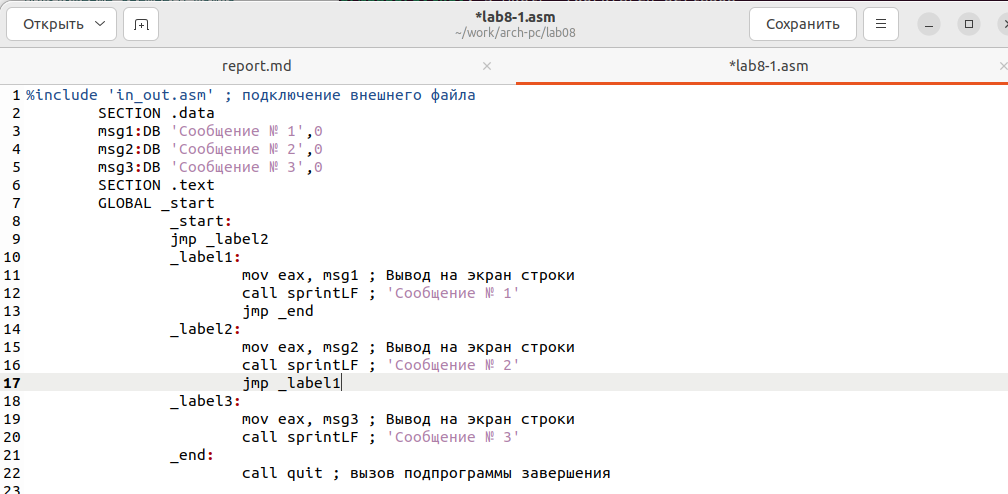


Рис. 4: Текст измененной программы

Создадим исполняемый файл исправленного текста программы lab8-1.asm и запустите его.(рис. 5)

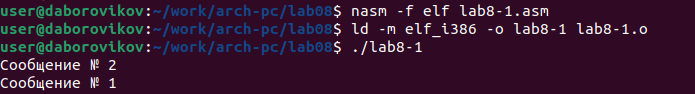


Рис. 5: Запуск исправленного исполняемого файла lab8-1.asm

Текст программы для следующего вывода:

user@dk4n31:~$ ./lab8-1

Сообщение № 3

Сообщение № 2

Сообщение № 1

user@dk4n31:~$(рис. 6)

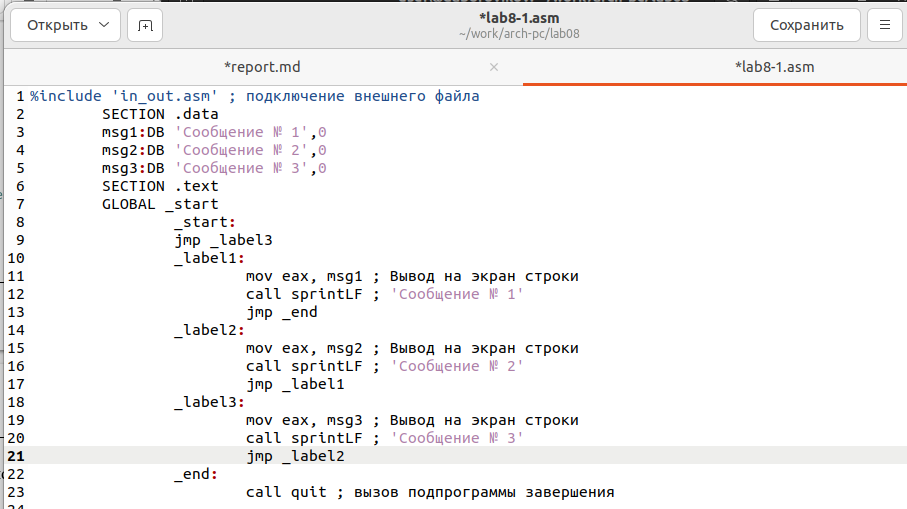


Рис. 6: Листинг программы с требуемым выводом

Создадим исполняемый файл и запустим его(рис. 7)

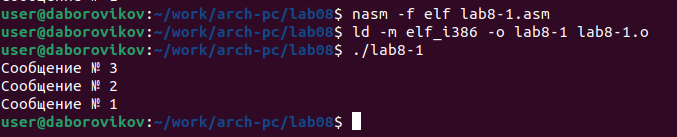


Рис. 7: Запуск измененного исполняемого файла lab8-1.asm

Создадим файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. Внимательно изучим текст программы из листинга 8.3 и введем в lab8-2.asm.(рис. 8)

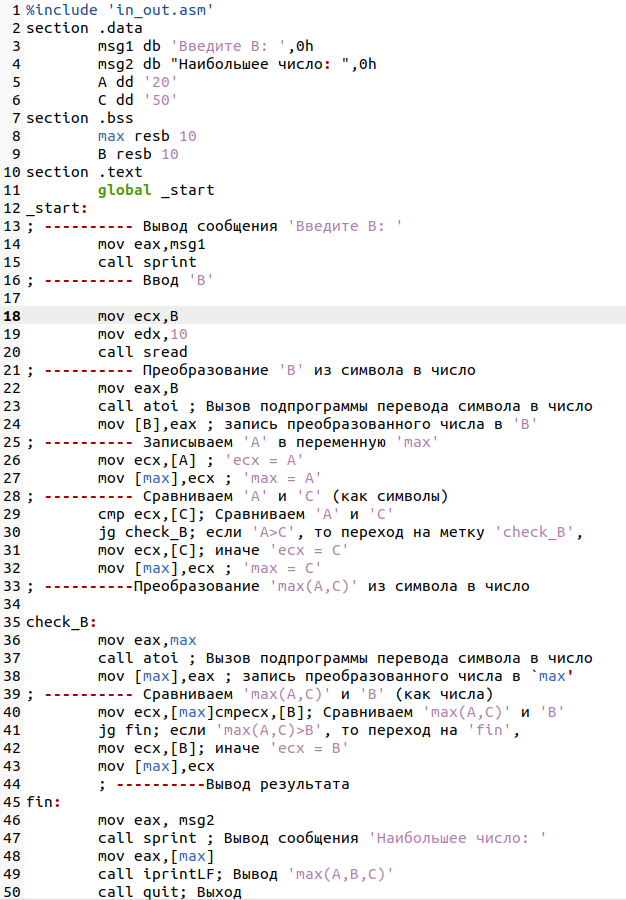


Рис. 8: Листинга 8.3

Создадим измененный исполняемый файл и запустим его(рис. 9)

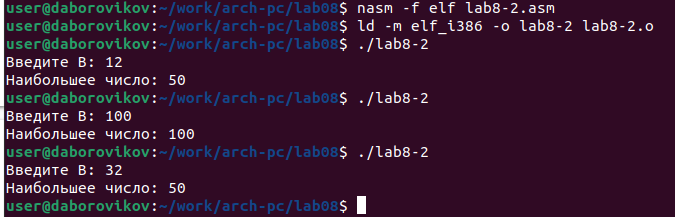


Рис. 9: Запуск исполняемого файла lab8-2.asm

Создадим файл листинга для программы из файла lab8-2.asm и откроем файл при помощи текстового редактора mcedit(рис. 10)

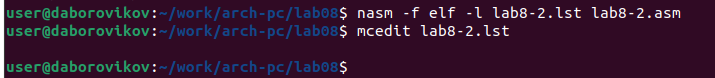


Рис. 10: Создание файла листинга

Разберем три строки листинга программы

21 00000101 B8[0A000000] mov eax,B

Значение строки:

21-номер строки в коде листинга от начала сегмента  
   
 00000101 - адрес  
   
 B8[0A000000] - машинный код(B8[0A000000] - инструкция mov eax,B ; B8 - обозначет что действие производится с регистром eax, а конкретно mov eax(в данной программе) ; [0A000000] - ссылка на переменную B)  
   
 mov eax,B - исходный текст программы

22 00000106 E891FFFFFF call atoi ; Вызов подпрограммы перевода символа в число

Значение строки:

22-номер строки в коде листинга от начала сегмента  
   
 00000106 - адрес  
   
 E891FFFFFF - машинный код(E891FFFFFF - инструкция call atoi ; E8 - обозначает что работает с с регисторм eax)  
   
 call atoi ; Вызов подпрограммы перевода символа в число - исходный текст программы

23 0000010B A3[0A000000] mov [B],eax ; запись преобразованного числа в ‘B’

Значение строки:

23-номер строки в коде листинга от начала сегмента  
   
 0000010B - адрес  
   
 A3[0A000000] - машинный код(A3[0A000000] - инструкция mov [B],eax ; A3 - обозначет что действие производится с регистром eax, а конкретно присваивание значения из eax в другую переменную(в данной программе) ; [0A000000] - ссылка на переменную B, обозначение, что мы работаем с этой переменной)  
   
 mov eax,B - исходный текст программы

(рис. 11)

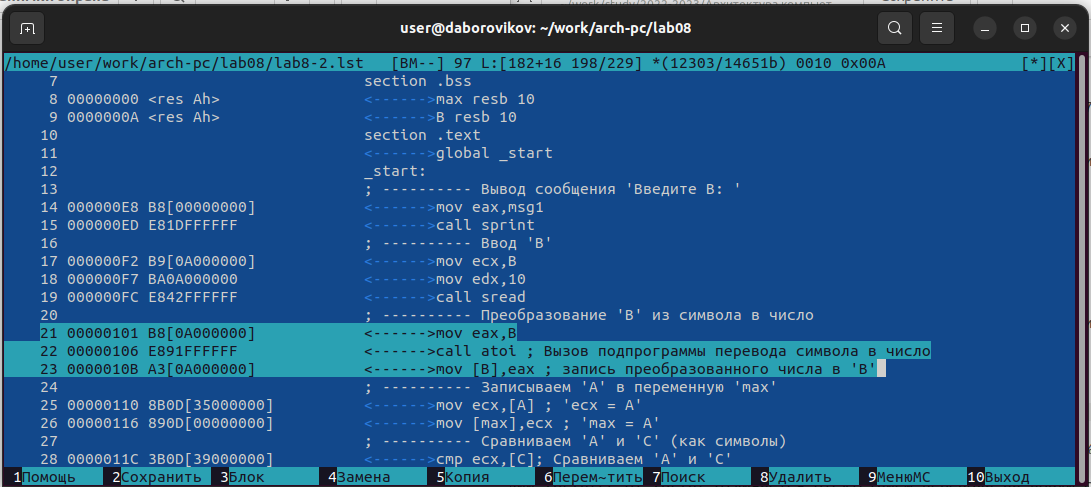


Рис. 11: Листинг программы

Откроем файл с программой lab8-2.asm и в любой инструкции с двумя операндами удалим один операнд. Выполните трансляцию с получением файла листинга:(рис. 12)

Рис. 12: Ошибка трансляции в терминале

Рис. 12: Ошибка трансляции в терминале

(рис. 13)

Рис. 13: Вывод ошибки в листинге

Рис. 13: Вывод ошибки в листинге

На выходе получаем листинг lab8-2.lst с ошибкой 46 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* error: invalid combination of opcode and operands

#Самостоятельная работа

Мой вариант номер 7

Напишем программу нахождения наименьшей из 3 целочисленных переменных A, B, C. Значения переменных выберем из табл. 8.5 в соответствиис вариантом, полученным при выполнении лабораторной работы № 7. Создадим исполняемый файл и проверьте его работу.(рис. 14)

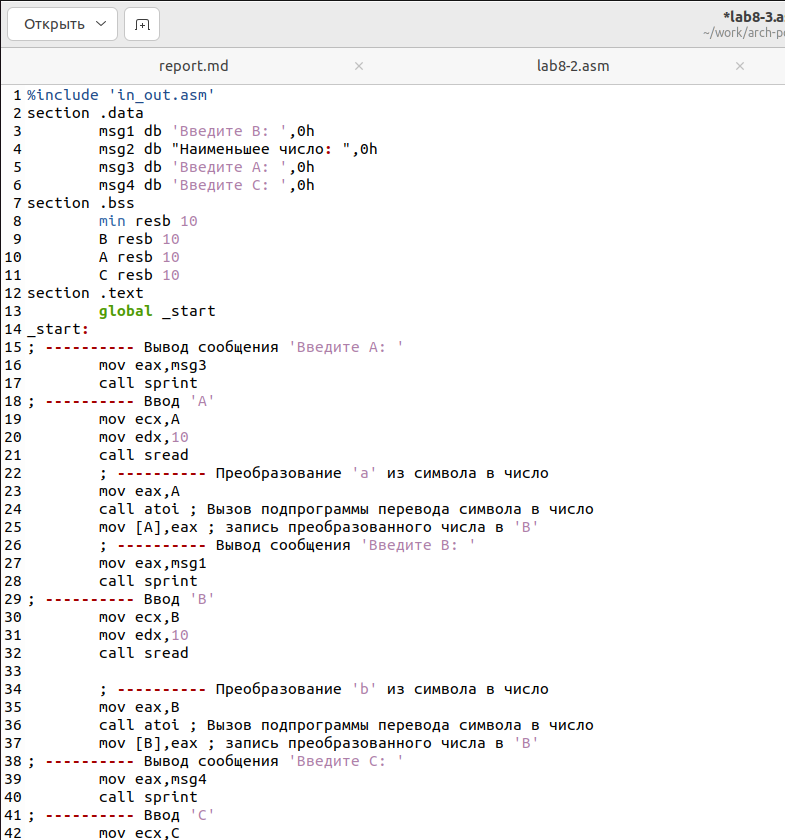


Рис. 14: Текст программы

Текст программы (рис. 15)

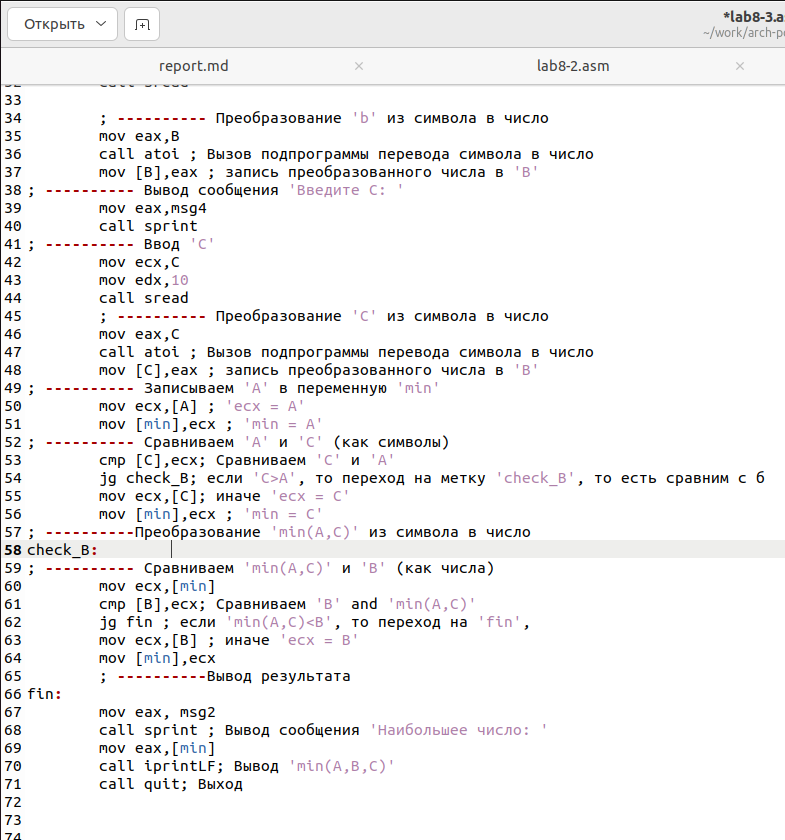


Рис. 15: Текст программы

Проверка работы исполняемого файла(рис. 16)

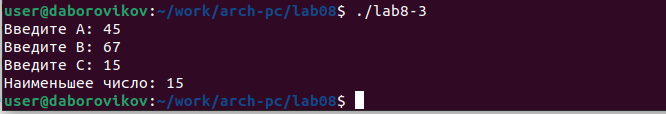


Рис. 16: Проверка работы исполняемого файла

Листинг программы:

%include ‘in\_out.asm’

section .data

msg1 db 'Введите B: ',0h  
  
msg2 db "Наименьшее число: ",0h  
  
msg3 db 'Введите A: ',0h  
  
msg4 db 'Введите C: ',0h

section .bss

min resb 10  
  
B resb 10  
  
A resb 10  
  
C resb 10

section .text

global \_start

\_start:

; ———- Вывод сообщения ‘Введите A:’

mov eax,msg3  
  
call sprint

; ———- Ввод ‘A’

mov ecx,A  
  
mov edx,10  
  
call sread  
  
; ---------- Преобразование 'a' из символа в число  
  
mov eax,A  
  
call atoi ; Вызов подпрограммы перевода символа в число  
  
mov [A],eax ; запись преобразованного числа в 'B'  
  
; ---------- Вывод сообщения 'Введите B: '  
  
mov eax,msg1  
  
call sprint

; ———- Ввод ‘B’

mov ecx,B  
  
mov edx,10  
  
call sread  
  
; ---------- Преобразование 'b' из символа в число  
  
mov eax,B  
  
call atoi ; Вызов подпрограммы перевода символа в число  
  
mov [B],eax ; запись преобразованного числа в 'B'

; ———- Вывод сообщения ‘Введите C:’

mov eax,msg4  
  
call sprint

; ———- Ввод ‘C’

mov ecx,C  
  
mov edx,10  
  
call sread  
  
; ---------- Преобразование 'C' из символа в число  
  
mov eax,C  
  
call atoi ; Вызов подпрограммы перевода символа в число  
  
mov [C],eax ; запись преобразованного числа в 'B'

; ———- Записываем ‘A’ в переменную ‘min’

mov ecx,[A] ; 'ecx = A'  
  
mov [min],ecx ; 'min = A'

; ———- Сравниваем ‘A’ и ‘С’ (как символы)

cmp [C],ecx; Сравниваем 'C' и 'A'  
  
jg check\_B; если 'C>A', то переход на метку 'check\_B', то есть сравним с б  
  
mov ecx,[C]; иначе 'ecx = C'  
  
mov [min],ecx ; 'min = C'

; ———-Преобразование ‘min(A,C)’ из символа в число

check\_B:

; ———- Сравниваем ‘min(A,C)’ и ‘B’ (как числа)

mov ecx,[min]  
  
cmp [B],ecx; Сравниваем 'B' and 'min(A,C)'  
  
jg fin   
  
mov ecx,[B] ; иначе 'ecx = B'  
  
mov [min],ecx  
  
; ----------Вывод результата

fin:

mov eax, msg2  
  
call sprint ; Вывод сообщения 'Наибольшее число: '  
  
mov eax,[min]  
  
call iprintLF; Вывод 'min(A,B,C)'  
  
call quit; Выход

Напишем программу, которая для введенных с клавиатуры значений 𝑥 и 𝑎 вычисляет значение заданной функции 𝑓(𝑥) и выводит результат вычислений. Вид функции 𝑓(𝑥) выбрать из таблицы 8.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений 𝑥 и 𝑎 из 8.6.(рис. 17)

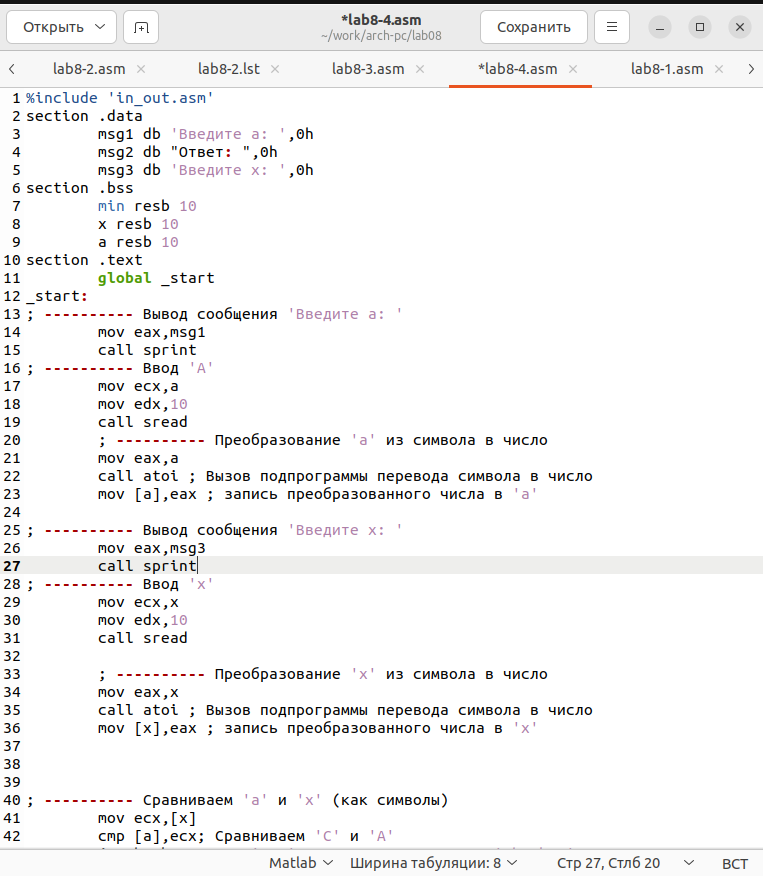
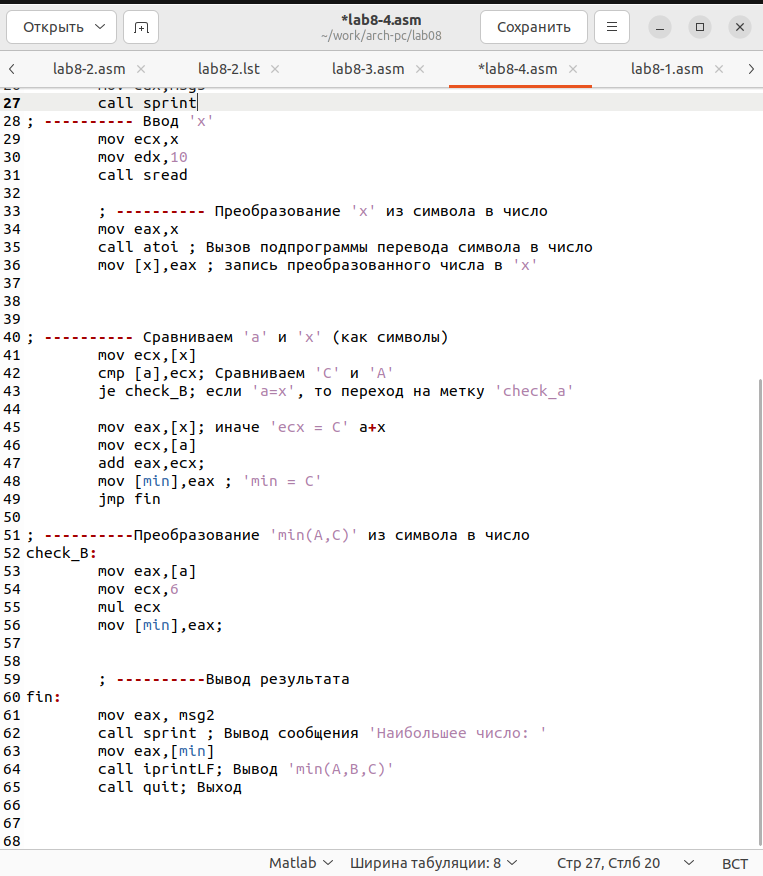


Рис. 17: Листинг программы lab8-4.asm

Листинг программы lab8-4.asm(рис. **¿fig:018?**)

 (рис. 18)

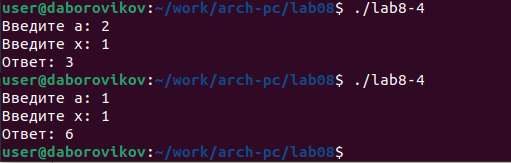


Рис. 18: Проверка работы программы

Листинг программы:

%include ‘in\_out.asm’

section .data

msg1 db 'Введите a: ',0h  
  
msg2 db "Ответ: ",0h  
  
msg3 db 'Введите x: ',0h

section .bss

min resb 10  
  
x resb 10  
  
a resb 10

section .text

global \_start

\_start:

; ———- Вывод сообщения ‘Введите a:’

mov eax,msg1  
  
call sprint

; ———- Ввод ‘A’

mov ecx,a  
  
mov edx,10  
  
call sread  
  
; ---------- Преобразование 'a' из символа в число  
  
mov eax,a  
  
call atoi ; Вызов подпрограммы перевода символа в число  
  
mov [a],eax ; запись преобразованного числа в 'a'

; ———- Вывод сообщения ‘Введите x:’

mov eax,msg3  
  
call sprint

; ———- Ввод ‘x’

mov ecx,x  
  
mov edx,10  
  
call sread  
  
; ---------- Преобразование 'x' из символа в число  
  
mov eax,x  
  
call atoi ; Вызов подпрограммы перевода символа в число  
  
mov [x],eax ; запись преобразованного числа в 'x'

; ———- Сравниваем ‘a’ и ‘x’ (как символы)

mov ecx,[x]  
  
cmp [a],ecx; Сравниваем 'C' и 'A'  
  
je check\_B; если 'a=x', то переход на метку 'check\_a'  
  
mov eax,[x]; иначе 'ecx = C' a+x  
  
mov ecx,[a]  
  
add eax,ecx;  
  
mov [min],eax ; 'min = C'  
  
jmp fin

; ———-Преобразование ‘min(A,C)’ из символа в число

check\_B:

mov eax,[a]  
  
mov ecx,6  
  
mul ecx  
  
mov [min],eax;  
   
  
; ----------Вывод результата

fin: mov eax, msg2

call sprint ; Вывод сообщения 'Наибольшее число: '  
  
mov eax,[min]  
  
call iprintLF; Вывод 'min(A,B,C)'  
  
call quit; Выход

# 3 Выводы

В ходе лабораторной работы мы изучили команды условного и безусловного переходов, приобрели навыки написания программ с использованием переходов, познакомились с назначением и структурой файла листинга.

https://github.com/daBorovikov/study\_2022-2023\_arh-pc-

:::