Отчёт по лабораторной работе №7

Дисциплина: Архитектура Компьютера

Иванов Сергей Владимирович

Содержание

# 1 Цель работы

Целью лабораторной работы является изучение команд условного и безусловного переходов, приобретение навыков написания программ с использованием переходов и знакомство с назначением и структурой файла листинга.

# 2 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 7, перейдём в него и создадим файл lab7-1.asm.(Рис. [1](#fig:001))

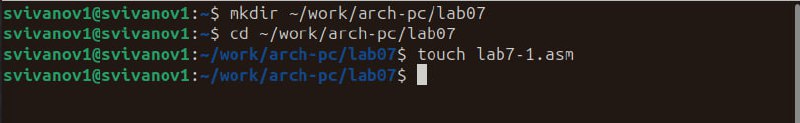


Figure 1: Создание lab7-1.asm

Введем в файл lab7-1.asm текст программы из листинга 7.1 (Рис. [2](#fig:002))

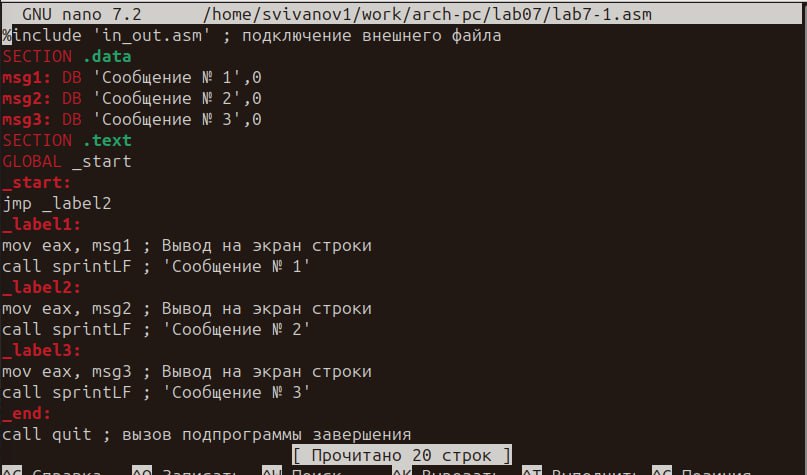


Figure 2: Текст из листинга 7.1

Создадим исполняемый файл и запустим его. (Рис. [3](#fig:003))

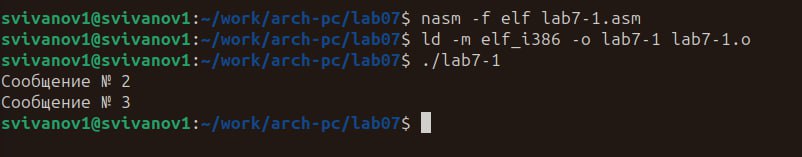


Figure 3: Исполняемый файл

Изменим текст программы в соответствии с листингом 7.2, создадим исполняемый файл и проверим его работу. (Рис. [4](#fig:004))

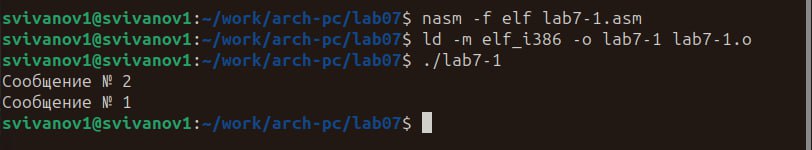


Figure 4: Программа из листинга 7.2

Изменим текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим: Сообщение №3, Сообщение №2, Сообщение №1. (Рис. [5](#fig:005))

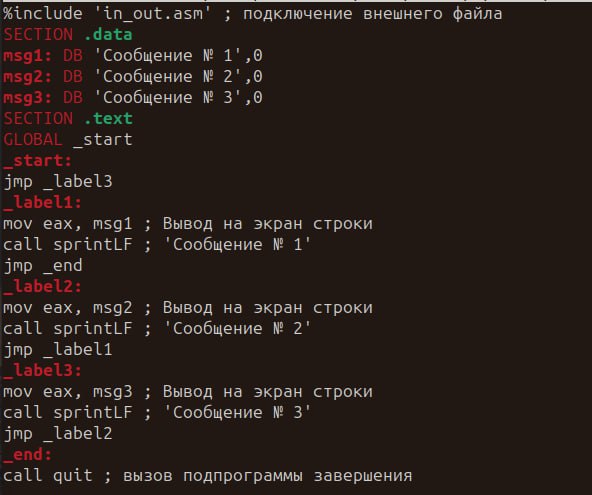


Figure 5: Изменение программы

Создадим исполняемый файл и запустим его. (Рис. [6](#fig:006))

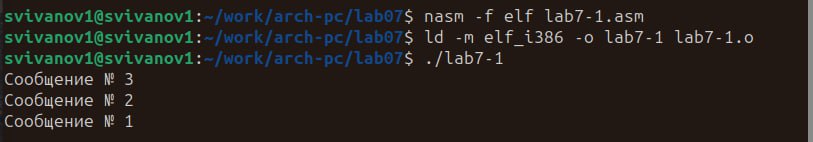


Figure 6: Запуск исполняемого файла

Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Введём в lab7-2.asm текст из листинга 7.3. Создадим исполняемый файл и проверим его работу для разных значений B. (Рис. [7](#fig:007))

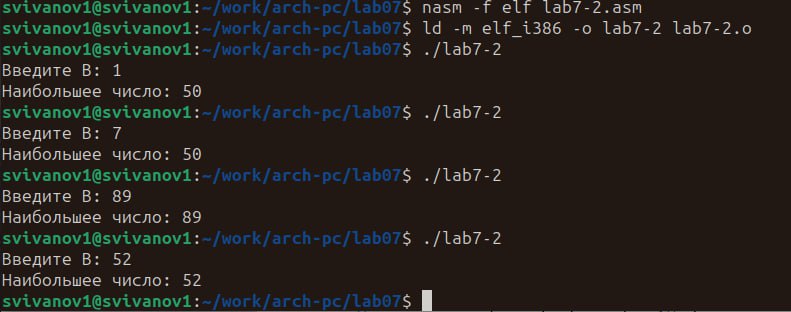


Figure 7: Программа из листинга 7.3

Создадим файл листинга для программы из файла lab7-2.asm и откроем его с помощью mcedit. (Рис. [8](#fig:008))

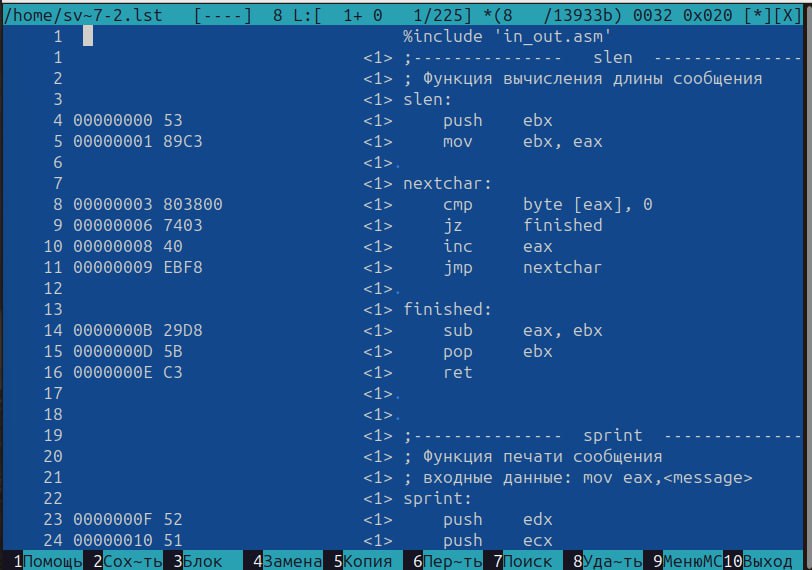


Figure 8: Файл листинга

Объясним содержание строк 4,5 и 8:

Строка 4: 4-номер строки, 00000000-адрес строки, 53-машинный код, push ebx-исходный текст программы.

Строка 5: 5-номер строки, 00000001-адрес строки, 89С3-машинный код, mov ebx,eax-исходный текст программы

Строка 8: 8-номер строки, 00000003-адрес строки, 803800-машинный код, cmp byte[eax],0-исходный текст программы

Откроем файл с программой lab7-2.asm и в строке mov edx,10 удалим 10 (Рис. [9](#fig:009))

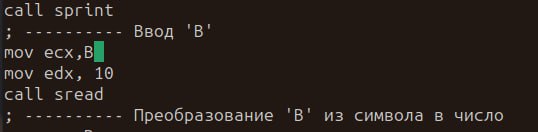


Figure 9: Удаление операнда

Создадим файл листинга и откроем его. Как мы видим, в строке в которой мы удалили операнд, появилось сообщение об ошибке, а в первой части листинга появились \* .(Рис. [10](#fig:010))

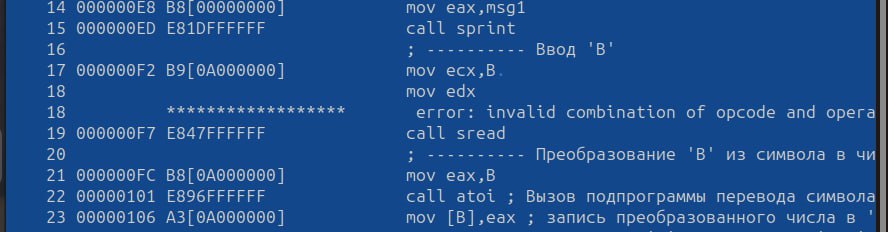


Figure 10: Листинг с ошибкой

Напишем программу нахождения наименьшей из 3 целочисленных переменных a,b и c в соответствии со своим вариантом(8). Создадим исполняемый файл и проверим его работу. Программа работает правильно. (Рис. [11](#fig:011))

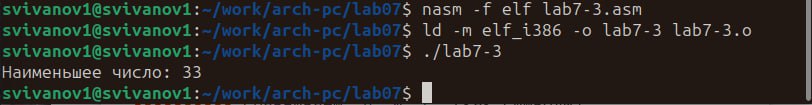


Figure 11: Программа lab7-3.asm

**Листинг программы для самостоятельной работы 1(lab7-3.asm):**

%include 'in\_out.asm'  
section .data  
msg2 db "Наименьшее число: ",0h  
A dd '52'  
B dd '33'  
C dd '40'  
section .bss  
min resb 10  
section .text  
global \_start  
\_start:  
mov eax,B  
call atoi  
mov [B],eax  
; ---------- Записываем 'A' в переменную 'min'  
mov ecx,[A] ; 'ecx = A'  
mov [min],ecx ; 'min = A'  
; ---------- Сравниваем 'A' и 'С' (как символы)  
cmp ecx,[C] ; Сравниваем 'A' и 'С'  
jl check\_B ; если 'A<C', то переход на метку 'check\_B',  
mov ecx,[C] ; иначе 'ecx = C'  
mov [min],ecx ; 'min = C'  
; ---------- Преобразование 'min(A,C)' из символа в число  
check\_B:  
mov eax,min  
call atoi ; Вызов подпрограммы перевода символа в число  
mov [min],eax ; запись преобразованного числа в `min`  
; ---------- Сравниваем 'min(A,C)' и 'B' (как числа)  
mov ecx,[min]  
cmp ecx,[B] ; Сравниваем 'min(A,C)' и 'B'  
jl fin ; если 'min(A,C)<B', то переход на 'fin',  
mov ecx,[B] ; иначе 'ecx = B'  
mov [min],ecx  
; ---------- Вывод результата  
fin:  
mov eax, msg2  
call sprint ;   
mov eax,[min]  
call iprintLF ;  
call quit ; Выход

Напишем программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Создадим исполняемый файл и проверим его работу для значений x и a из 7.6.(Вариант 8). (Рис. [12](#fig:012))

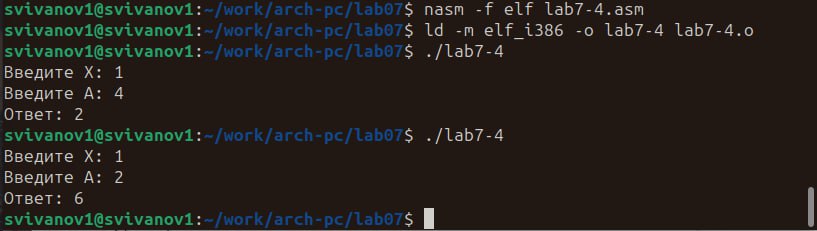


Figure 12: Программа lab7-4.asm

**Листинг программы для самостоятельной работы 2(lab7-4.asm):**

%include 'in\_out.asm'  
section .data  
msg1 DB "Введите X: ",0h  
msg2 DB "Введите A: ",0h  
msg3 DB "Ответ: ",0h  
section .bss  
x: RESB 80  
a: RESB 80  
ans: RESB 80  
section .text  
global \_start  
\_start:  
mov eax,msg1  
call sprint  
mov ecx,x  
mov edx,80  
call sread  
mov eax,x  
call atoi  
mov [x],eax  
mov eax,msg2  
call sprint  
mov ecx,a  
mov edx, 80  
call sread  
mov eax,a  
call atoi  
mov [a],eax  
  
mov eax,[a]  
cmp eax,3  
jl xsa  
  
mov eax,[x]  
add eax,1  
jmp ansv  
  
xsa:  
mov ebx,3  
mov eax,[a]  
mul ebx  
ansv:  
mov [ans],eax  
mov eax,msg3  
call sprint  
mov eax,[ans]  
call iprintLF  
call quit

# 3 Выводы

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