Лабораторная работа №7

Дисциплина: архитектура компьютера

Мусатова Екатерина Викторовна

Содержание

# 1 Цель работы

Изучить команды условного и безусловного переходов и научиться писать программы с использованием этих переходов.

# 2 Выполнение лабораторной работы

**1**

Создаю каталог для лабораторной работы № 7, перехожу в него и создаю файл lab7-1.asm (рис. [1](#fig:001)).

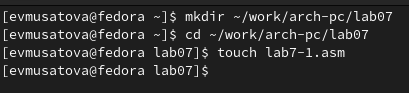


Figure 1: Создание файла

**2**

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

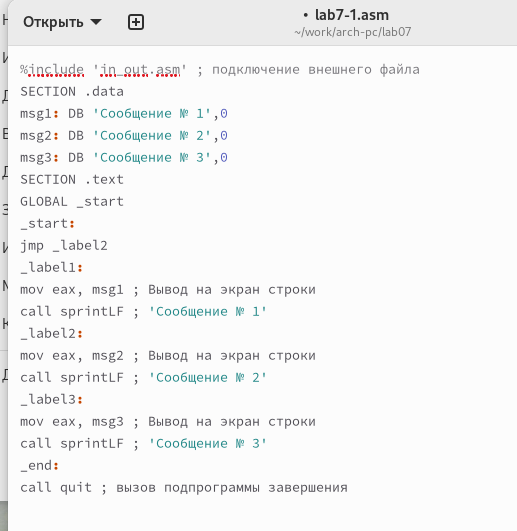


Figure 2: Ввод текста программы

Создаю исполняемый файл и проверяю правильность работы программы (рис. [3](#fig:003)).

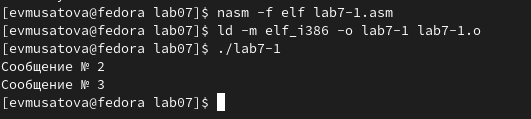


Figure 3: Проверка

Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу в соответствии с листингом 7.2 (рис. [4](#fig:004)).

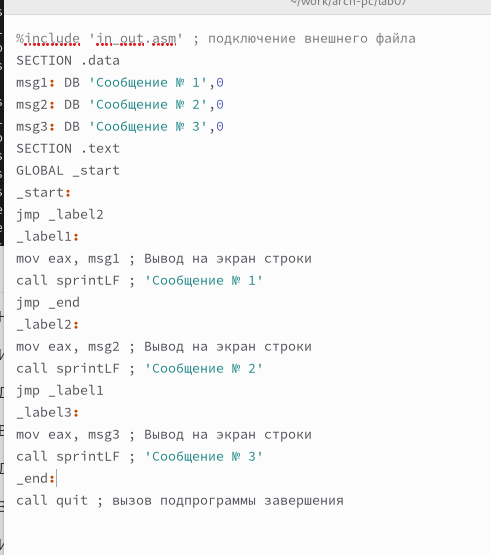


Figure 4: Изменение программы

Создаю исполняемый файл и проверяю работу программы (рис. [5](#fig:005)).

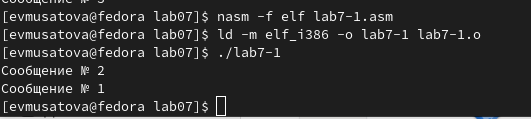


Figure 5: Проверка

Изменяю текст программы добавляя и изменяя инструкции jmp (рис. [6](#fig:006)).

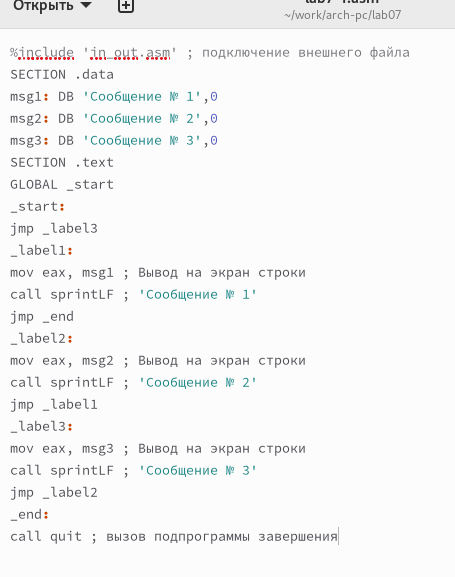


Figure 6: Изменение программы

Создаю исполняемый файл и проверяю работу программы (рис. [7](#fig:007)).

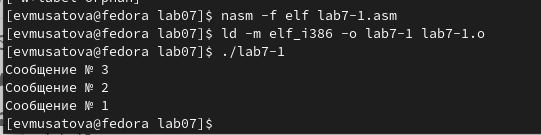


Figure 7: Проверка

**3**

Создаю файл lab7-2.asm (рис. [8](#fig:008)).

Figure 8: Создание файла

Figure 8: Создание файла

В созданный файл ввожу текст из листинга 7.3 (рис. [9](#fig:009)).

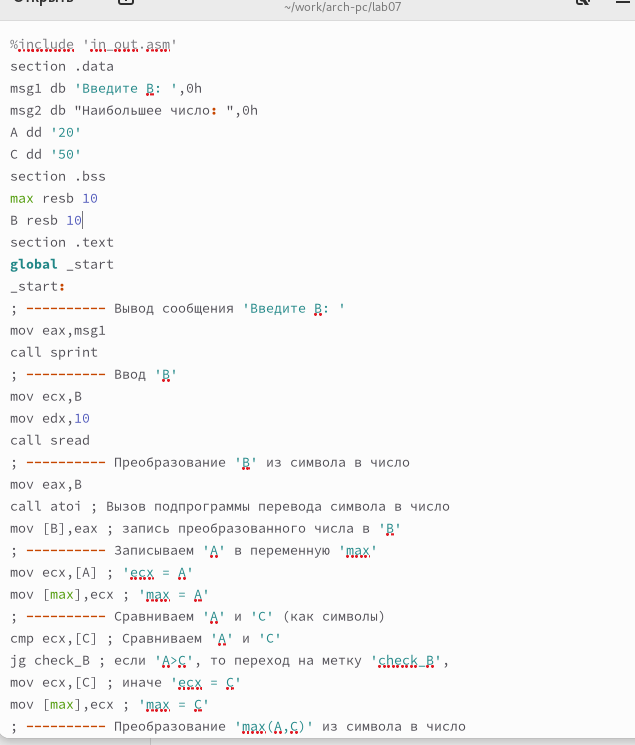


Figure 9: Ввод текста

Создаю исполняемый файл и проверяю его работу для разных значений B (рис. [10](#fig:010)).

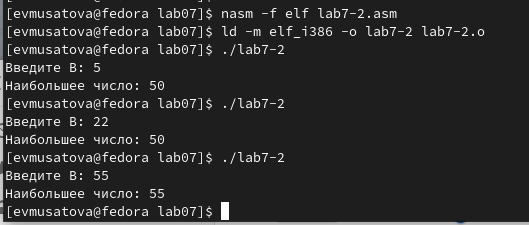


Figure 10: Проверка

**4**

Создаю файл листинга для программы из файла lab7-2.asm (рис. [11](#fig:011)).

Figure 11: Создание файла листинга

Figure 11: Создание файла листинга

Открываю файл листинга с помощью текстового редактора mcedit (рис. [12](#fig:012)). В строке 9 содержится номер сторки [8], адресс [00000003], машинный код [803800] и содержимое строки кода [cmp byte [eax], 0] в строке 11 содержится номер сторки [10], адресс [00000008], машинный код [40] и содержимое строки кода [inc eax] в строке 24 содержится номер сторки [23], адрес [0000000F], машинный код [52] и содержимое строки кода [push edx].

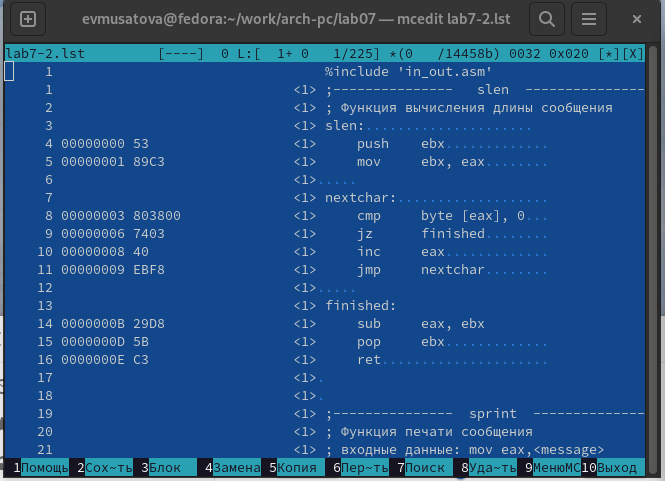


Figure 12: Открытие файла листинга

Открываю файл с программой lab7-2.asm и в инструкции с двумя операндами удаляю операнд B (рис. [13](#fig:013)).



Figure 13: Удаление операнда

Выполняю трансляцию с получением файла листинга (рис. [14](#fig:014)).

Figure 14: Трансляция

Figure 14: Трансляция

Проверяю файл листинга и вижу, что если в коде появляется ошибка, то ее описание появится в файле листинга (рис. [15](#fig:015)).

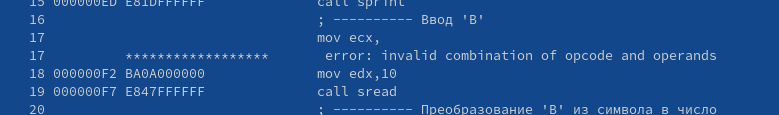


Figure 15: Проверка листинга

# 3 Самостоятельная работа

Создаю файл lab7-3 и пишу программу из 3 целочисленных переменных A,B и С в соответствии с 8 вариантом (рис. ??).

![Написание программы(image/16.png){#fig:016 width=70%}

Создаю исполняемый файл и проверяю работу программы (рис. [16](#fig:017)).

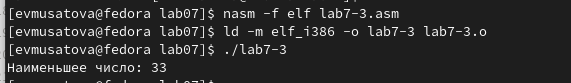


Figure 16: Проверка

Создаю файл lab7-4 и пишу программу, которая для введенных с клавиатуры значений 𝑥 и 𝑎 вычисляет значение заданной функции из варианта 8 (рис. [17](#fig:018)).

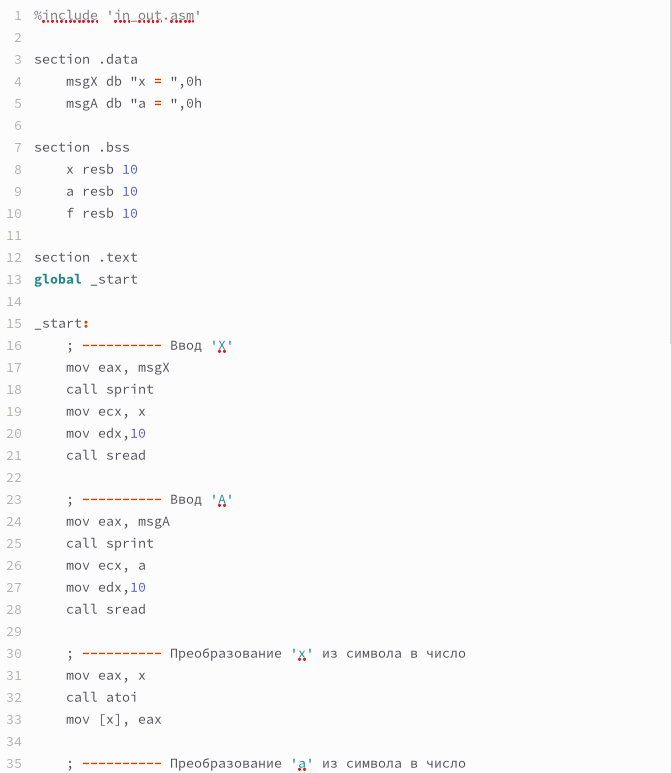


Figure 17: Написание программы

Создаю исполняемый файл и проверяю работу программы (рис. [18](#fig:019)).

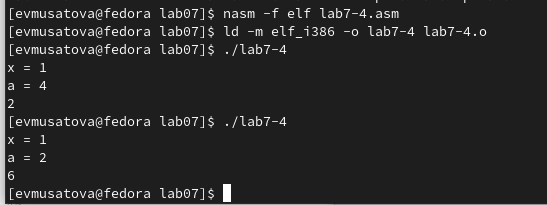


Figure 18: Проверка

**Листинг к заданию 1**

%include ‘in\_out.asm’ section .data msg1 db “Наименьшее число:”,0h a dd 33 b dd 40 c dd 52 section .bss min resb 10 section .text global \_start \_start: mov ecx, [a] mov [min], ecx ; ‘min = A’ ; ———- Сравниваем ‘A’ и ‘С’ (как числа) 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: ; ———- 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, msg1 call sprint mov eax,[min] call iprintLF ; Вывод ‘min(A,B,C)’ call quit ; Выход

\*\*Листинг к заданию 2\*\*

%include ‘in\_out.asm’

section .data msgX db “x =”,0h msgA db “a =”,0h

section .bss x resb 10 a resb 10 f resb 10

section .text global \_start

\_start: ; ———- Ввод ‘X’ mov eax, msgX call sprint mov ecx, x mov edx,10 call sread

; ---------- Ввод 'A'  
mov eax, msgA  
call sprint  
mov ecx, a  
mov edx,10  
call sread  
  
; ---------- Преобразование 'x' из символа в число  
mov eax, x  
call atoi  
mov [x], eax  
  
; ---------- Преобразование 'a' из символа в число  
mov eax, a  
call atoi  
mov [a], eax  
  
  
mov ecx, [a]  
cmp ecx, 3 ;сравниваем а и цифру 3  
  
ja newfunc ;если а больше то идем по метке  
  
mov eax, [a] ;иначе а умножаем на 3  
mov ebx, 3  
mul ebx  
jmp fin

newfunc: mov eax, [x] mov ebx, 1 add eax, ebx

fin: call iprintLF call quit

# 4 Выводы

Я изучила команды условного и безусловного переходов и научилась писать программы с использованием этих переходов.