Отчёт по лабораторной работе №6

Дициплина : архитектура компьютера

Зарина Исмайилбековна Исаева

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. Задание для самостоятельной 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 программы без каких-либо условий.

Безусловный переход выполняется инструкцией jmp. Инструкция cmp является одной из инструкций, которая позволяет сравнить операнды и выставляет флаги в зависимости от результата сравнения. Инструкция cmp является командой сравнения двух операндов и имеет такой же формат, как и команда вычитания.

Листинг — это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

# 4 Выполнение лабораторной работы

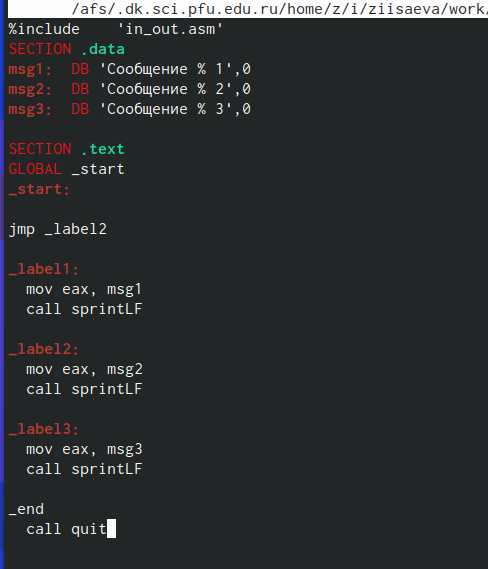
## 4.1 Реализация переходов в NASM

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

Создание файлов для лабораторной работы

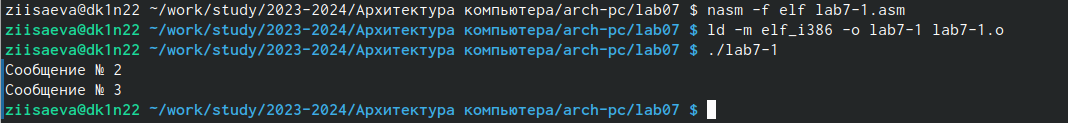
Создание файлов для лабораторной работы

Ввожу в файл lab7-1.asm текст программы из листинга 7.1 (рис. [??]).



Ввод текста программы из листинга 7.1

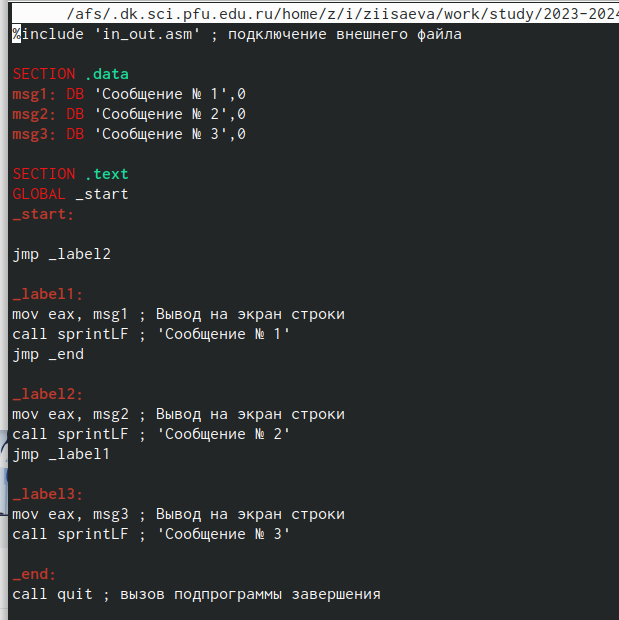
Создаю исполняемый файл и запускаю его (рис. [??]).



Запуск программного кода

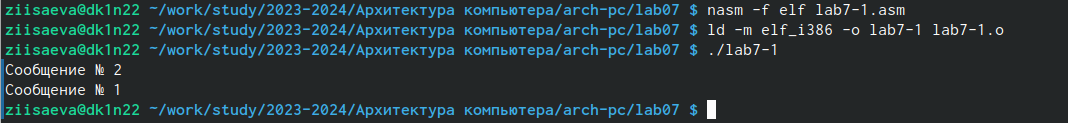
Таким образом, использование инструкции jmp \_label2 меняет порядок исполнения инструкций и позволяет выполнить инструкции начиная с метки \_label2, пропустив вывод первого сообщения.

Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого изменим текст программы в соответствии с листингом 7.2 (рис. [??]).



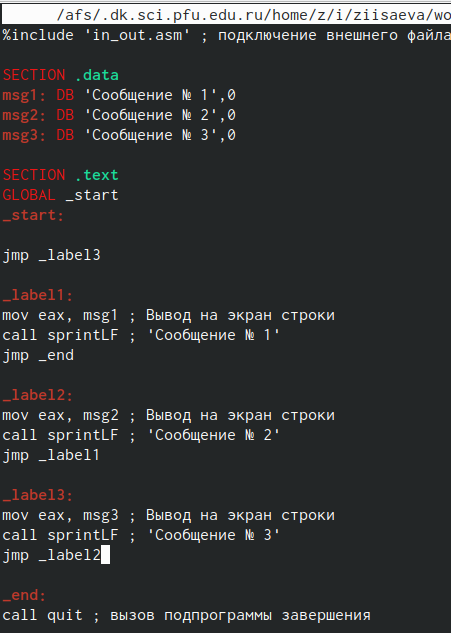
Редактирование текста программы

Создаю исполняемый файл и проверяю его работу (рис. [??])



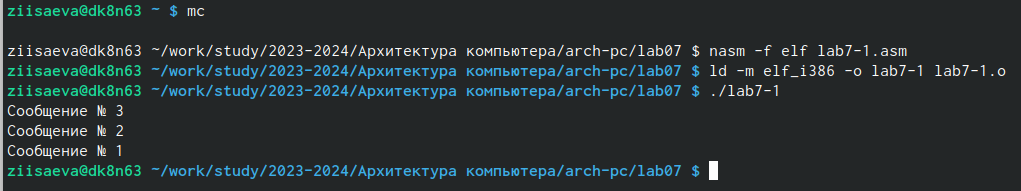
Создание исполняемого файла

Затем изменим текст программы, добавив в начале программы jmp \_label3, jmp \_label2 в конце метки jmp \_label3, jmp \_label1 добавлю в конце метки jmp \_label2, и добавлю jmp \_end в конце метки jmp \_label1 (рис. [??]).



Редактирование текста программы

Для того, чтобы вывод программы был следующим: (рис. [??])



Вывод программы

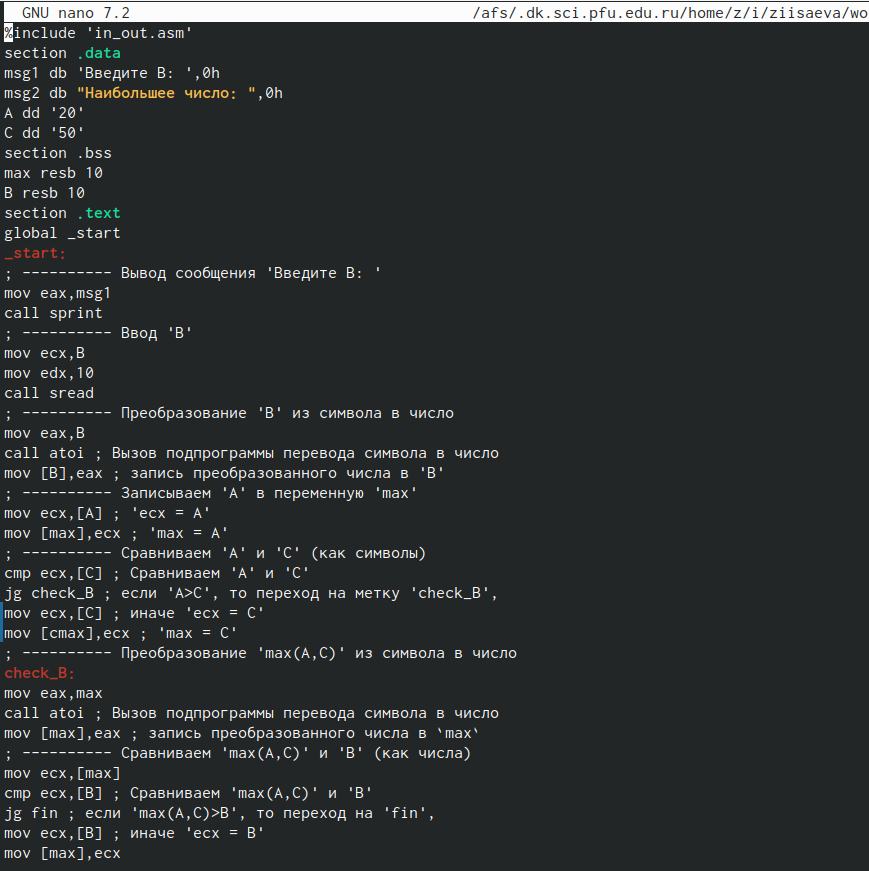
Рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C. Значения для A и C задаются в программе, значение B вводиться с клавиатуры.

Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 (рис. [??]).

Создание файла

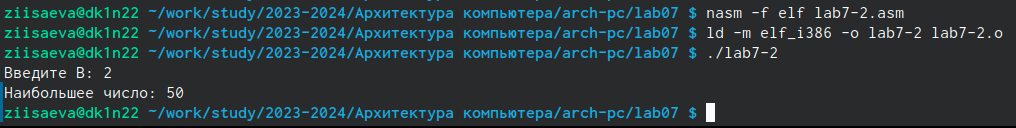
Создание файла

Текст программы из листинга 7.3 ввожу в lab7-2.asm (рис. [??]).



Ввод текста программы из листинга 7.3

Создаю исполняемый файл и проверьте его работу (рис. [??])

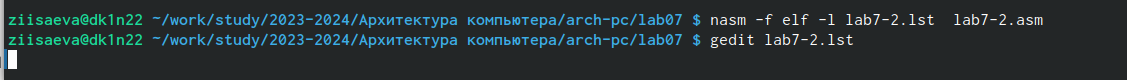


Проверка работы файла

Файл работает корректно.

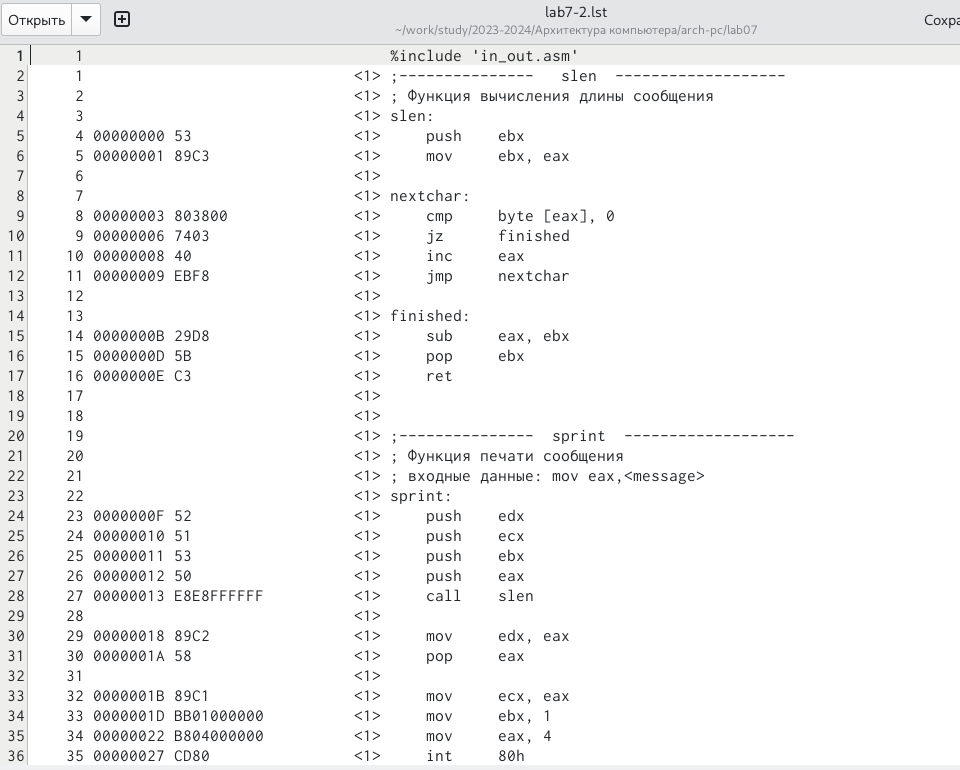
## 4.2 Изучение структуры файлов листинга

Создаю файл листинга для программы из файла lab7-2.asm (рис. [??]).



Создание файла листинга

Открываю файл листинга lab7-2.lst с помощью текстового редактора и внимательно изучаю его формат и содержимое (рис. [??])



Изучение файла листинга

В представленных трех строчках содержаться следующие данные: (рис. [??]).

Выбранные строки файла

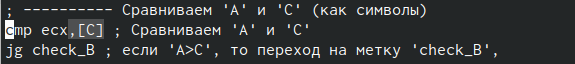
Выбранные строки файла

“2” - номер строки кода, “; Функция вычисления длинны сообщения” - комментарий к коду, не имеет адреса и машинного кода.

“3” - номер строки кода, “slen” - название функции, не имеет адреса и машинного кода.

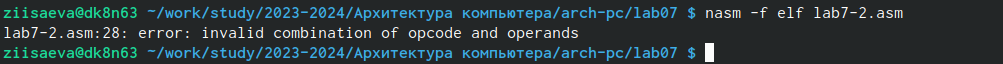
“4” - номер строки кода, “00000000” - адрес строки, “53” - машинный код, “push ebx” - исходный текст программы, инструкция “push” помещает операнд “ebx” в стек.

Открываю файл с программой lab7-2.asm и в выбранной мной инструкции с двумя операндами удаляю выделенный операнд (рис. [??])



Удаление выделенного операнда

Выполняю трансляцию с получением файла листинга (рис. [??]).

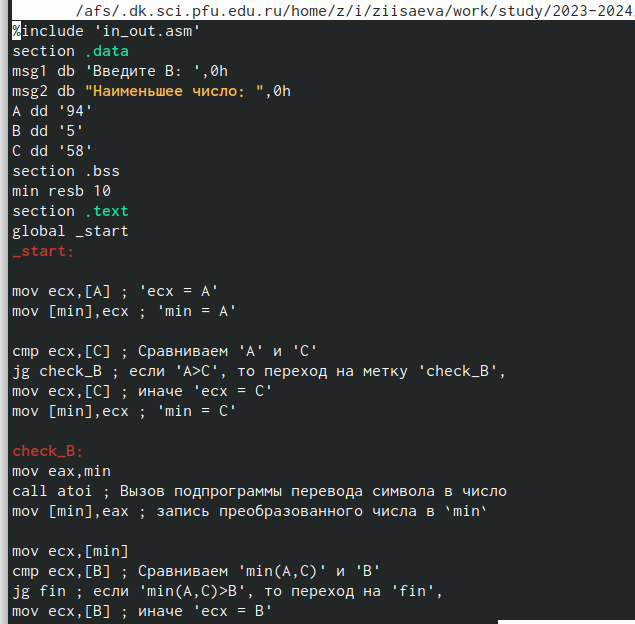


Получение файла листинга

На выходе я не получаю ни одного файла из-за ошибки:инструкция mov (единственная в коде содержит два операнда) не может работать, имея только один операнд, из-за чего нарушается работа кода.

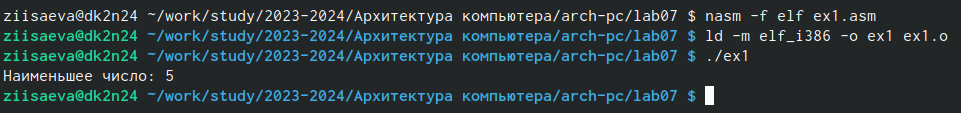
## 4.3 Задания для самостоятельной работы

1. Пишу программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбираю из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 6. Мой вариант под номером 3, поэтому мои значения - 94, 5 и 58 (рис. [??]).



Написание программы

Создаю исполняемый файл и проверяю его работу, подставляя необходимые значение (рис. [??]).



Запуск файла и проверка его работы

Программа работает корректно.

Код программы:

%include ‘in\_out.asm’ section .data msg db “Наименьшее число:”,0h A dd ‘94’ B dd ‘5’ C dd ‘58’ section .bss min resb 10 section .text global \_start \_start:

mov ecx,[A] ; ‘ecx = A’ mov [min],ecx ; ‘min = A’

cmp ecx,[C] ; Сравниваем ‘A’ и ‘С’ jg check\_B ; если ‘A>C’, то переход на метку ‘check\_A’, mov ecx,[C] ; иначе ‘ecx = C’ mov [min],ecx ; ‘min = C’

check\_B:

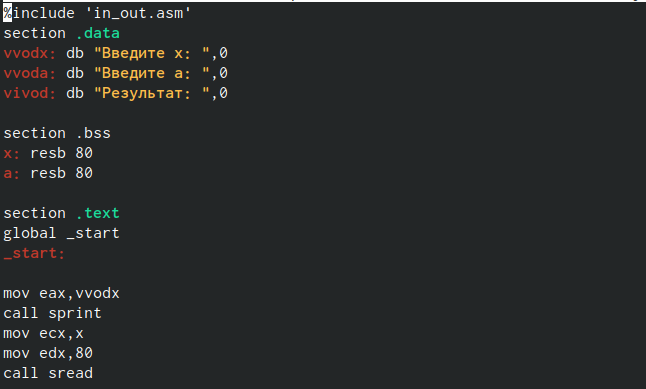
mov ecx,[min] cmp ecx,[B] ; Сравниваем ‘min(B,C)’ и ‘A’ jle fin ; если ‘min(B,C)>A’, то переход на ‘fin’, mov ecx,[B] ; иначе ‘ecx = A’ mov [min],ecx

fin: mov eax, msg call sprint ; Вывод сообщения ‘Наибольшее число:’ mov eax,min call sprintLF ; Вывод ‘min(A,B,C)’ call quit ; Выход

1. Пишу программу (выражения которого взяты из таблицы 7.6),которая для введенных с клавиатуры значений х и а вычисляет значение и выводит результат вычислений заданной для моего варианта функции f(x):

3\*x, если x = 3 a + 1, если x ≠ 3

(рис. [??])



Написание программы

Создаю исполняемый файл и проверяю его работу для значений х и а соответственно: (3;4), (1;4) (рис. [??]).

|  |
| --- |
| Запуск файла и проверка его работы |

Запуск файла и проверка его работы

Программа работает корректно.

Код программы:

%include ‘in\_out.asm’ section .data vvodx: db “Введите x:”,0 vvoda: db “Введите а:”,0 vivod: db “Результат:”,0

section .bss x: resb 80 a: resb 80

section .text global \_start \_start:

mov eax,vvodx call sprint mov ecx,x mov edx,80 call sread

mov eax,x call atoi cmp eax,2 jg \_functionx mov eax,vvoda call sprint

mov ecx,a mov edx,80 call sread

mov eax,a call atoi jmp \_functiona

\_functiona: mov edx,3 mul edx jmp \_end

\_functionx: add eax,+1 jmp \_end

\_end: mov ecx,eax mov eax,vivod call sprint mov eax,ecx call iprintLF call quit

# 5 Выводы

Я изучила команды условного и безусловного переходов, приобрела навыки написания программ с использованием переходов и ознакомилась с назначением и структурой файла листинга, что поможет мне при выполнении последующих лабораторных работ.

# 6 Список литературы

1. [Лабораторная работа №7](%5Bhttps://esystem.rudn.ru/pluginfile.php/2089086/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%966.%20%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8%20%D0%B2%20NASM..pdf%5D(https://esystem.rudn.ru/pluginfile.php/2089087/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%967.%20%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B%20%D0%B1%D0%B5%D0%B7%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B8%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D0%BE%D0%B2%20%D0%B2%20Nasm.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%82%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9..pdf)https://esystem.rudn.ru/pluginfile.php/2089087/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%967.%20%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B%20%D0%B1%D0%B5%D0%B7%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B8%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BF%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%B4%D0%BE%D0%B2%20%D0%B2%20Nasm.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%B2%D0%B5%D1%82%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9..pdf)