Отчёта по лабораторной работе №7

Архитектура компьютера

Еремина Оксана Андреевна НКАбд-02-23

Содержание

# Цель работы

Целью данной лабораторной работы является освоение арифметческих инструкций языка ассемблера NASM.

# Задание

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга
3. Выполнение заданий для самостоятельной работы

# 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.

Флаг – это бит, принимающий значение 1 («флаг установлен»), если выполнено некоторое условие, и значение 0 («флаг сброшен») в противном случае. Флаги работают независимо друг от друга, и лишь для удобства они помещены в единый регистр — регистр флагов, отражающий текущее состояние процессора

Листинг (в рамках понятийного аппарата NASM) — 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.

# Выполнение лабораторной работы

1. Реализация переходов в NASM

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

Figure 1: Создание папки и файла

Figure 1: Создание папки и файла

Ввожу в файл lab7-1.asm текст программы. (рис.2)

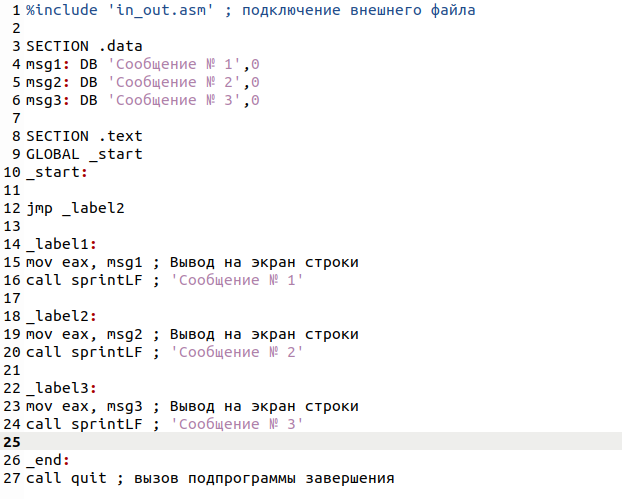


Figure 2: Содержание файла lab7-1.asm

Создаю исполняемый файл и проверяю работу программы. (рис.3)

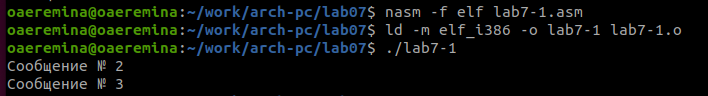


Figure 3: Выподнение программы

Изменяю код программы так, чтобы она сначала выводила ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. (рис.4)

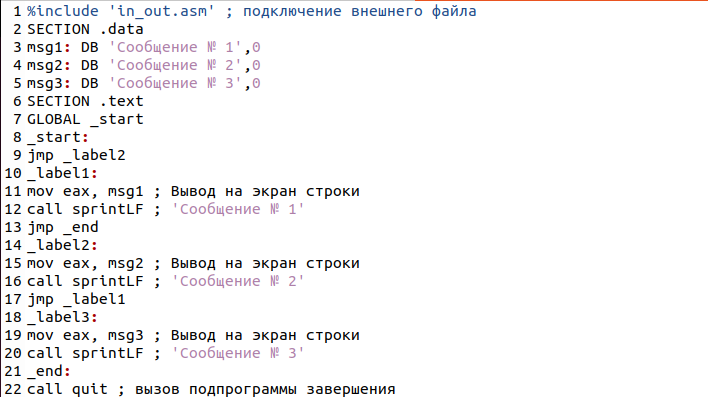


Figure 4: Изменение программы

Создаю исполняемый файл и проверяю работу программы. (рис.5)

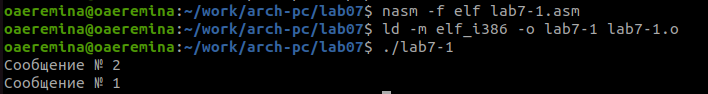


Figure 5: Выподнение программы

Изменяю код программы так, чтобы она сначала выводила ‘Сообщение № 3’, потом ‘Сообщение № 2’, ‘Сообщение № 1’ и завершала работу. (рис.6)

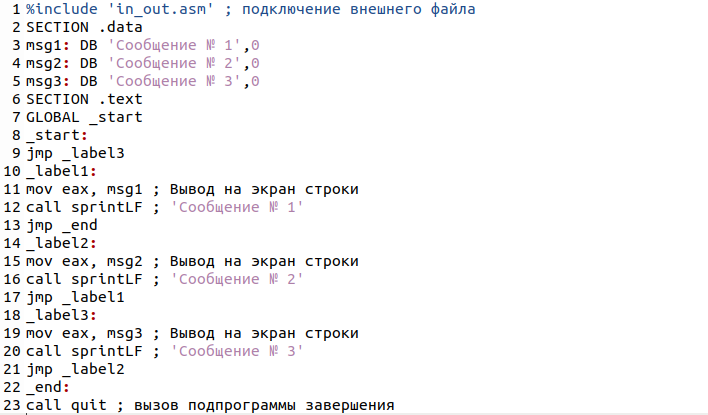


Figure 6: Изменение программы

Создаю исполняемый файл и проверяю работу программы. (рис.7)

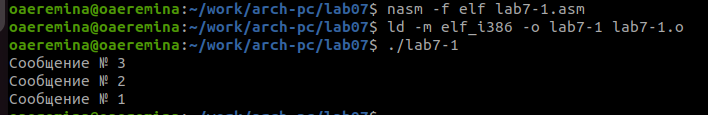


Figure 7: Выподнение программы

Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Ввожу код программы в созданный файл. (рис.8)

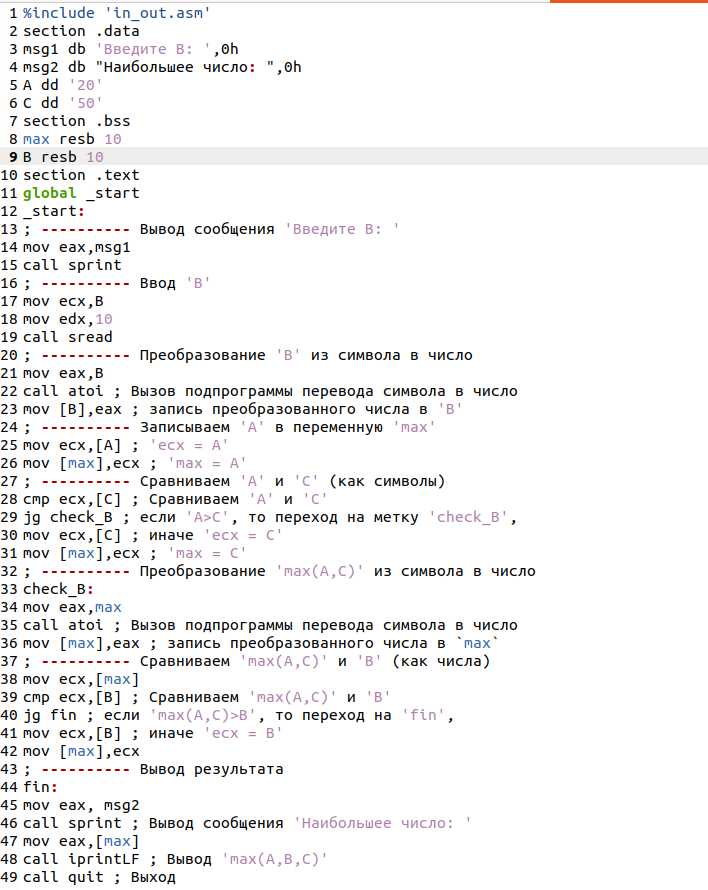


Figure 8: Содержание файла lab7-2.asm

Создаю исполняемый файл и провяю его работу для разных значений B (1; 5). (рис.9)

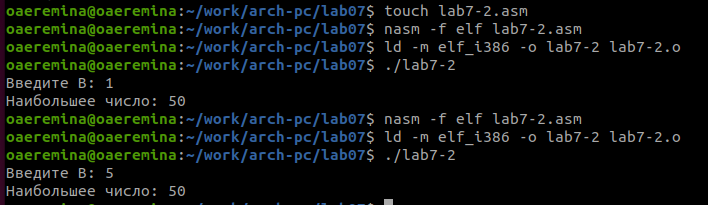


Figure 9: Выподнение программы

1. Изучение структуры файлы листинга

Создаю файл листинга для программы из файла lab7-2.asm, знакамлюсь с его форматом и содержимым файла листинга, она показан ниже. (рис.10)

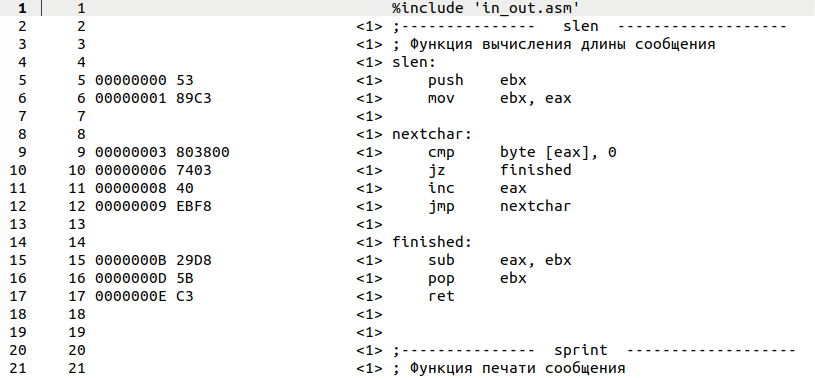


Figure 10: Структура и содержимое файла листинга

Открываю файл с программой lab7-2.asm и в выбранной мной инструкции с двумя операндами удаляю выделенный операнд. (рис.11)

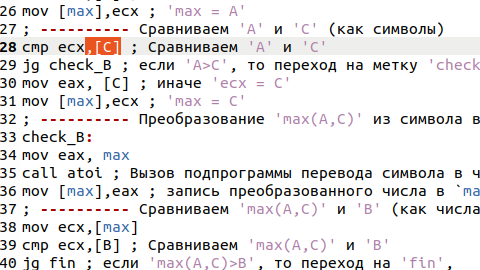


Figure 11: Изменение файла

Выполняю трансляцию с получением 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.(рис.12)

Figure 12: Ошибка в работе кода

Figure 12: Ошибка в работе кода

1. Выполнение заданий для самостоятельной работы

Пишу программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбираю из табл. 7.5. Мой вариант под номером 17, поэтому мои значения - 26, 12 и 68.

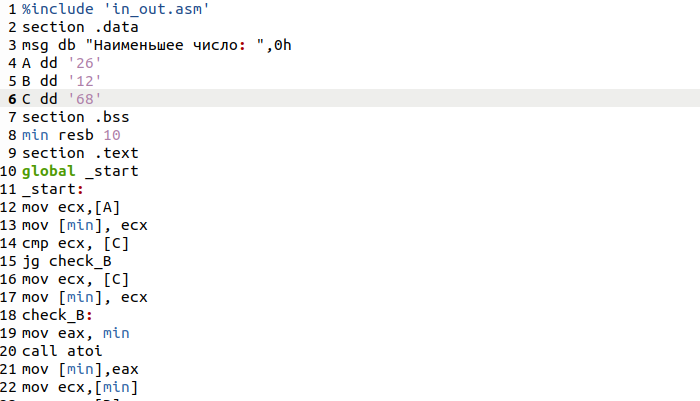


Figure 13: Код прогрраммы

Создаю исполняемый файл и проверяю работу программы. (рис.14)

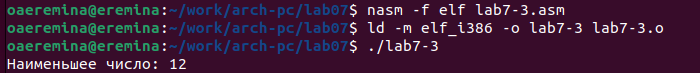


Figure 14: Выподнение программы

Код программы:

%include ‘in\_out.asm’

section .data

msg db “Наименьшее число:”,0h

A dd ‘26’

B dd ‘12’

C dd ‘68’

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

mov ecx,[C] ; иначе ‘ecx = C’

mov [min],ecx ; ‘min = C’

check\_B:

mov eax,min

call atoi ; Вызов подпрограммы перевода символа в число

mov [min],eax ; запись преобразованного числа в min

mov ecx,[min]

cmp ecx,[B] ; Сравниваем ‘min(A,C)’ и ‘B’

jl fin

mov ecx,[B]

mov [min],ecx

fin:

mov eax, msg

call sprint ; Вывод сообщения ‘Наименьшее число:’

mov eax,[min]

call iprintLF ; Вывод ‘min(A,B,C)’

call quit ; Выход

# Вывод

При выполнении данной лабораторной работы я изучила команды условного и безусловного перехода, приобрела навык написания программ с использованием переходов и познакомилась с назначением и структурой файла листинга.

# Список литературы

Архитектура ЭВМ