Лабораторная работа №7

НКАбд-06-23

Улитина Мария Максимовна

Содержание

[1 Цель работы 1](#__RefHeading___Toc319_1968280025)

[2 Задание 1](#__RefHeading___Toc321_1968280025)

[3 Теоретическое введение 2](#__RefHeading___Toc323_1968280025)

[3.1 Команды безусловного перехода 2](#__RefHeading___Toc325_1968280025)

[3.2 Команды условного перехода 2](#__RefHeading___Toc327_1968280025)

[3.2.1 Регистр флагов 2](#__RefHeading___Toc329_1968280025)

[3.2.2 Описание инструкции cmp 2](#__RefHeading___Toc331_1968280025)

[3.2.3 Описание команд условного перехода 3](#__RefHeading___Toc333_1968280025)

[3.3 Файл листинга и его структура 3](#__RefHeading___Toc335_1968280025)

[4 Выполнение лабораторной работы 3](#__RefHeading___Toc337_1968280025)

[4.1 Задания для самостоятельной работы 9](#__RefHeading___Toc339_1968280025)

[5 Выводы 9](#__RefHeading___Toc341_1968280025)

[6 Список литературы 9](#__RefHeading___Toc343_1968280025)

# 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 файла листинга.
3. Написание программы нахождения наименьшей из 3 целочисленных переменных.
4. Написание программы для вычисления заданной функции.

# 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.

## 3.1 Команды безусловного перехода

Безусловный переход выполняется инструкцией jmp (от англ. jump – прыжок), которая включает в себя адрес перехода, куда следует передать управление: jmp < адрес\_перехода > Адрес перехода может быть либо меткой, либо адресом области памяти, в которую предварительно помещен указатель перехода. Кроме того, в качестве операнда можно использовать имя регистра, в таком случае переход будет осуществляться по адресу, хранящемуся в этом регистре.

## 3.2 Команды условного перехода

Как отмечалось выше, для условного перехода необходима проверка какого-либо условия. В ассемблере команды условного перехода вычисляют условие перехода анализируя флаги из регистра флагов.

### 3.2.1 Регистр флагов

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. В следующей таблице указано положение битовых флагов в регистре флагов.

### 3.2.2 Описание инструкции cmp

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: cmp < операнд\_1 >, < операнд\_2 > Команда cmp, так же как и команда вычитания, выполняет вычитание - , но результат вычитания никуда не записывается и единственным результатом команды сравнения является формирование флагов.

### 3.2.3 Описание команд условного перехода

Команда условного перехода имеет вид j < мнемоника перехода > label Мнемоника перехода связана со значением анализируемых флагов или со способом фор- мирования этих флагов. В табл. 7.3. представлены команды условного перехода, которые обычно ставятся после команды сравнения cmp. В их мнемокодах указывается тот результат сравнения, при котором надо делать переход. Мнемоники, идентичные по своему действию, написаны в таблице через дробь (например, ja и jnbe). Программист выбирает, какую из них применить, чтобы получить более простой для понимания текст программы.

## 3.3 Файл листинга и его структура

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.

# 4 Выполнение лабораторной работы

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

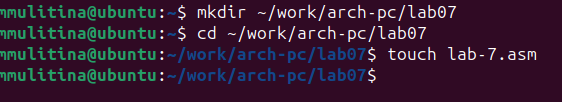


Figure 1: Создание файла

Введём в файл текст программы (рис. [2](#fig:002)).

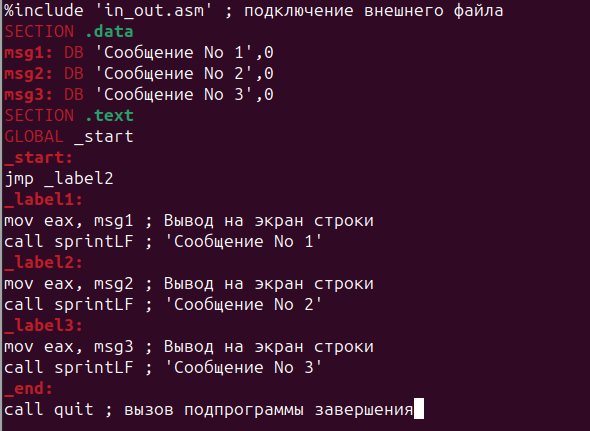


Figure 2: Текст программы

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

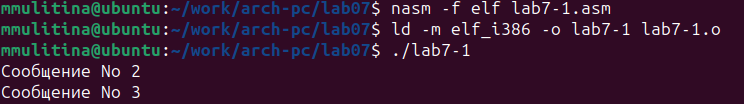


Figure 3: Работа программы

Изменим текст программы (рис. [4](#fig:004)).

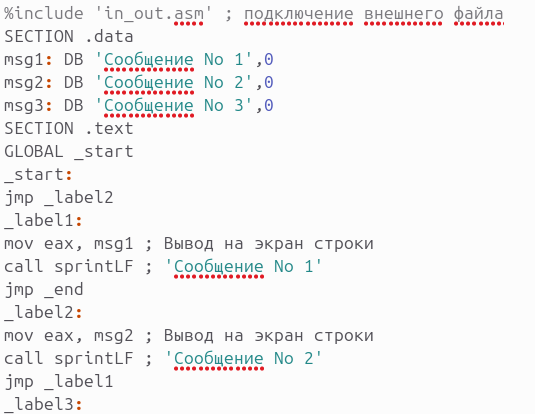


Figure 4: Измененный текст программы

Запустим программу и проверим его работу (рис. [5](#fig:005)).

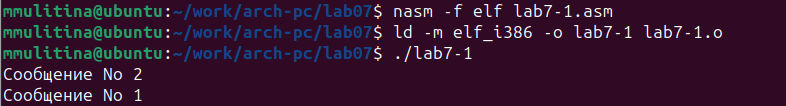


Figure 5: Работа программы

Изменим текст программы (рис. [6](#fig:006)).

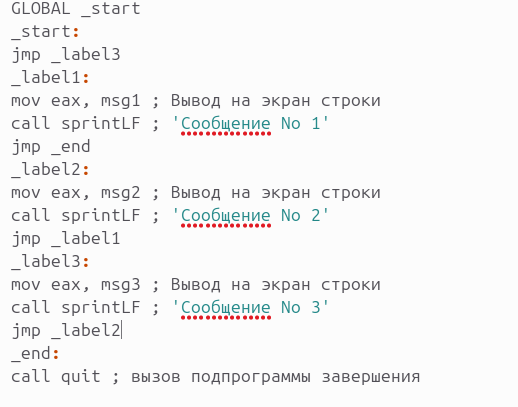


Figure 6: Изменение текста программы

Запустим программу (рис. [7](#fig:007)).

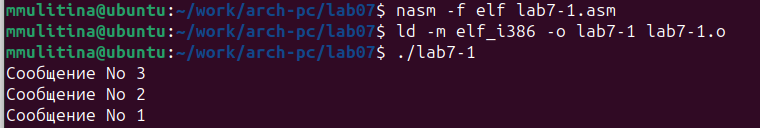


Figure 7: Работа программы

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

Figure 8: Создание файла

Figure 8: Создание файла

и введём в неё предложенный текст (рис. [9](#fig:009)).

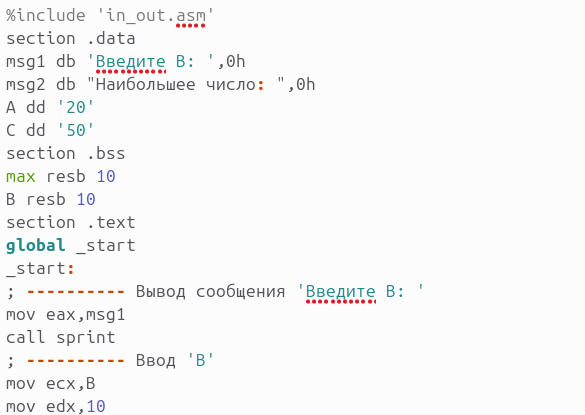


Figure 9: Ввод текста

Проверим работу программы с разными значениями (рис. [10](#fig:010)).

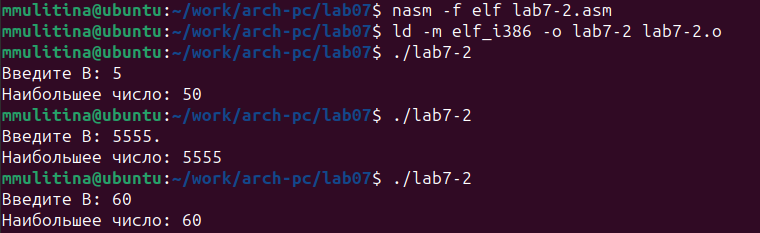


Figure 10: Проверка

Получим объектный файл, указав ключ -l (рис. [11](#fig:011)).

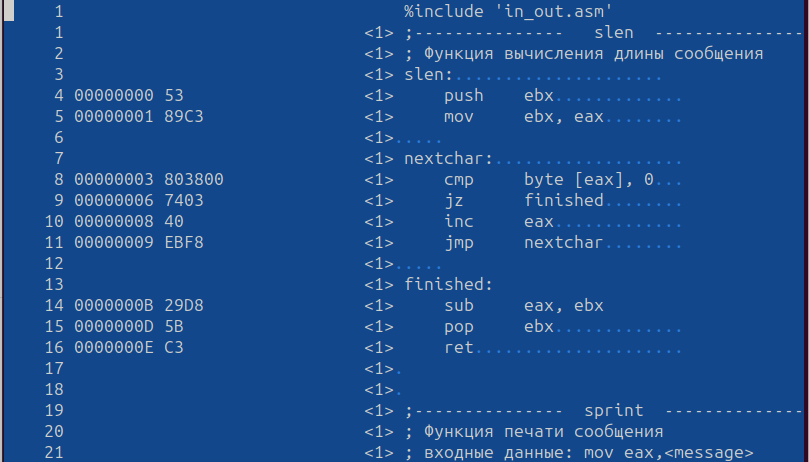


Figure 11: Листинг

Откроем файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалим один операнд. (рис. [12](#fig:012)).

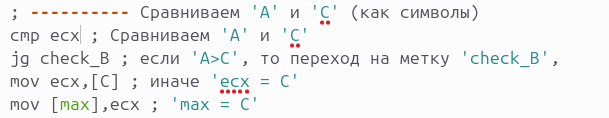


Figure 12: Изменение программы

Откроем файл листинга (рис. [13](#fig:013)).



Figure 13: Листинг

(рис. [14](#fig:014)).

Figure 14: Программа

Figure 14: Программа

## 4.1 Задания для самостоятельной работы

Напишем программу для нахождения наименьшей из 3 целочисленных переменных (рис. [15](#fig:015)).

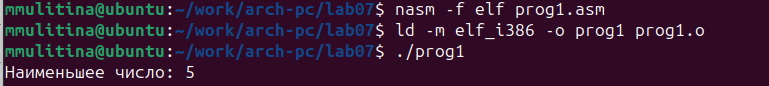


Figure 15: Программа

Напишем программу для вычисления функции для введённых с клавиатуры значений (рис. [16](#fig:016)).

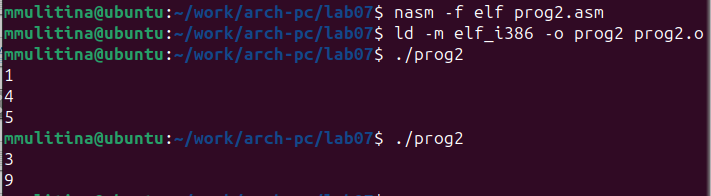


Figure 16: Программа

# 5 Выводы

В процессе выполнения работы были изучены команды условного и безусловного переходов. Приобретены навыки написания программ с использованием переходов.

# 6 Список литературы

Архитектура ЭВМ. Лабораторная работа №7.