Шаблон отчёта по лабораторной работе №7

Дисциплина: архитектура компьютера

Пронякова Ольга Максимовна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Ответы на вопросы по программе
4. Выполнение заданий для самостоятельной работы

# 3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обра- ботке. Это могут быть данные хранящиеся в регистре или в ячейке памяти. Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации. Существует три основных способа адресации: • Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: mov ax,bx. • Непосредственная адресация – значение операнда задается непосред- ственно в команде, Например: mov ax,2. • Адресация памяти – операнд задает адрес в памяти. В команде указывает- ся символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию. Ввод информации с клавиатуры и вывод её на экран осуществляется в сим- вольном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информа- цией). Согласно стандарту ASCII каждый символ кодируется одним байтом. Расширенная таблица ASCII состоит из двух частей. Первая (символы с кода- ми 0-127) является универсальной (см. Приложение.), а вторая (коды 128-255) предназначена для специальных символов и букв национальных алфавитов и на компьютерах разных типов может меняться. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). Поэтому, например, чтобы вывести число, надо предварительно преоб- разовать его цифры в ASCII-коды этих цифр и выводить на экран эти коды, а не само число. Если же выводить число на экран непосредственно, то экран воспримет его не как число, а как последовательность ASCII-символов – каждый байт числа будет воспринят как один ASCII-символ – и выведет на экран эти символы. Аналогичная ситуация происходит и при вводе данных с клавиатуры. Вве- денные данные будут представлять собой символы, что сделает невозможным получение корректного результата при выполнении над ними арифметических операций. Для решения этой проблемы необходимо проводить преобразование ASCII символов в числа и обратно. Для выполнения лабораторных работ в файле in\_out.asm реализованы под- программы для преобразования ASCII символов в числа и обратно. Это: • iprint – вывод на экран чисел в формате ASCII, перед вызовом iprint в регистр eax необходимо записать выводимое число (mov eax,). • iprintLF – работает аналогично iprint, но при выводе на экран после числа добавляет к символ перевода строки. • atoi – функция преобразует ascii-код символа в целое число и записает результат в регистр eax, перед вызовом atoi в регистр eax необходимо записать число (mov eax,).

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM

С помощью mkdir создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №7 (рис. 1).

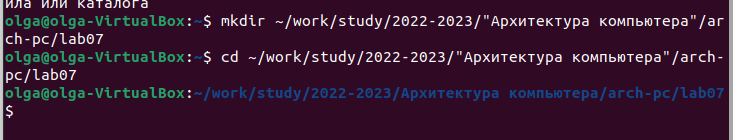


Рис. 1: Создание директории

С помощью touch создаю файл lab7-1.asm (рис. 2).

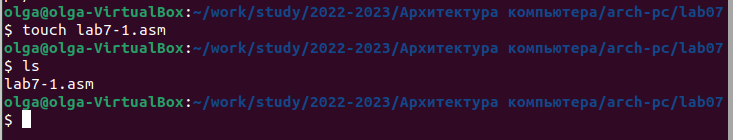


Рис. 2: Создание файла

С помощью cp копирую в текущий каталог файл in\_out.asm, так как он будет использоваться в других программах (рис. 3).

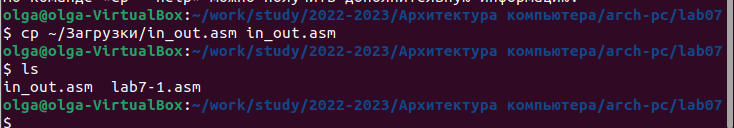


Рис. 3: Копирование файла

Открываю файл lab7-1.asm и вставляю в него программу вывода значения регистра eax (рис. 4).

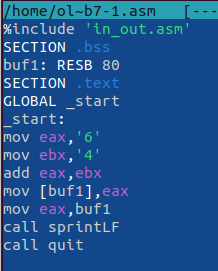


Рис. 4: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла. Программа выводит j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6 (рис. 5).

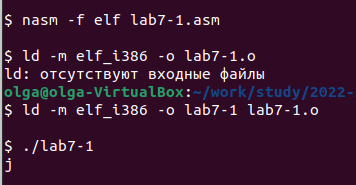


Рис. 5: Исполнение файла

Изменяю в программе символы “6” и “4” на цифры 6 и 4 (рис. 6).

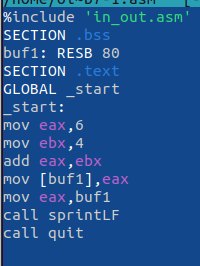


Рис. 6: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла. Теперь программа выводит символ с кодом 10, символ перевода строки, он не отображается при выводе на экран (рис. 7).

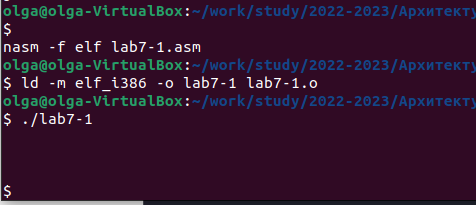


Рис. 7: Исполнение файла

С помощью touch создаю файл lab7-2.asm (рис. 8).

Рис. 8: Создание файла

Рис. 8: Создание файла

Ввожу в файл текст другой программы для вывода значения регистра eax (рис. 9).

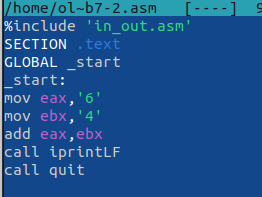


Рис. 9: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла. Теперь выводит число 106, потому что программа позволяет вывести именно число, а не символ (рис. 10).

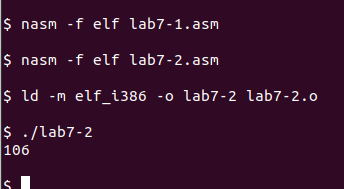


Рис. 10: Исполнение файла

Изменяю в программе символы “6” и “4” на цифры 6 и 4 (рис. 11).

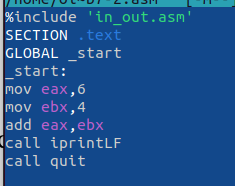


Рис. 11: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому выводит 10 (рис. 12).

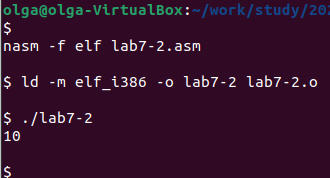


Рис. 12: Исполнение файла

Заменяю в тексте программы функцию iprintLF на iprint (рис. 13).

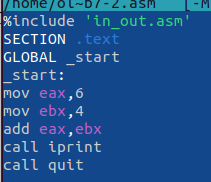


Рис. 13: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла. Вывод не изменился, так как символ переноса строки не отобразился, когда программа исполнялась с функцией iprintLF, а iprint не доюавляет к выводу символ переноса строки (рис. 14).

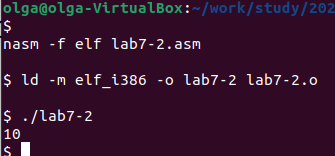


Рис. 14: Исполнение файла

## 4.2 Выполнение арифметических операций в NASM

С помощью touch создаю файл lab7-3.asm (рис. 15).

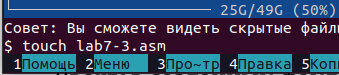


Рис. 15: Создание файла

Ввожу в файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 16).

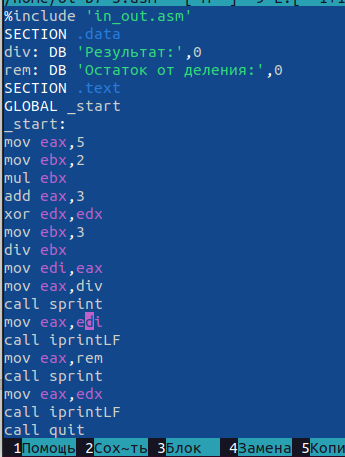


Рис. 16: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла (рис. 17).

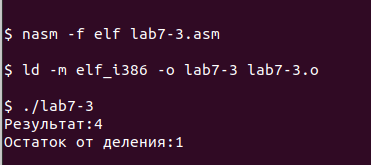


Рис. 17: Исполнение файла

Изменяю программу так, чтобы она вычисляла выражение f(x) = (4 \* 6 + 2)/5 (рис. 18).

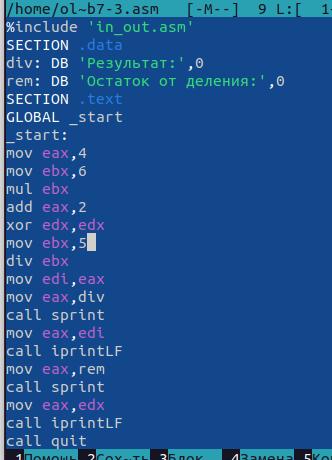


Рис. 18: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла (рис. 19).

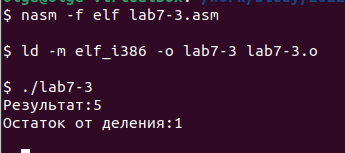


Рис. 19: Исполнение файла

С помощью touch создаю файл variant.asm (рис. 20).

Рис. 20: Создание файла

Рис. 20: Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 21).

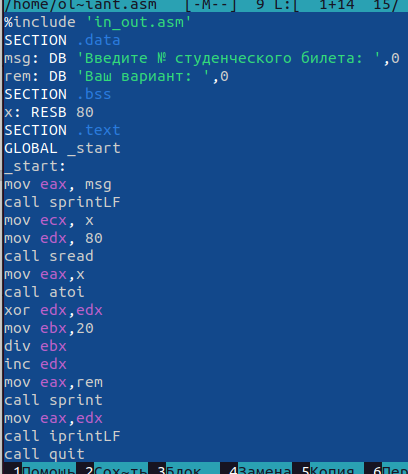


Рис. 21: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла. Ввожу номер студенческого билета, программа вывела мой вариант-14 (рис. 22).

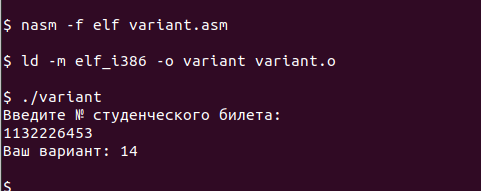


Рис. 22: Исполнение файла

## 4.3 Ответы на вопросы по программе

1. За вывод сообщения “Ваш вариант” отвечают строки: mov eax,rem call sprint
2. mov ecx - положить адрес вводимой строки x в регистр ecx mov edx,80 - запись а регистр edx длины вводимой строки call sread - вызов подпрограммы из внешнего файла, обеспечивающий ввод сообщения с клавиатуры
3. call atoi - вызов подпрограммы тз внешнего файла, которая преобразует ASCII-код символа в целое число и записывает результат в регистр eax
4. За вычисления варианта отвечают строки: xor edx,edx mov ebx,20 div ebx inc edx
5. При выполнении инструкции div ebx остаток от деления записывется в регистр edx
6. Инструкция inc edx увеличивает значение регистра edx на 1
7. За вывод результатов вычислений отвечают строки: mov eax,edx call iprintLF

## 4.4 Выполнение заданий для самостоятельной работы

С помощью touch создаю файл lab7-4.asm (рис. 23).

Рис. 23: Создание файла

Рис. 23: Создание файла

Открываю файл и ввожу в него текст программы для вычисления значения выражения 14 варианта (x/2 + 8) \* 3 (рис. 24).

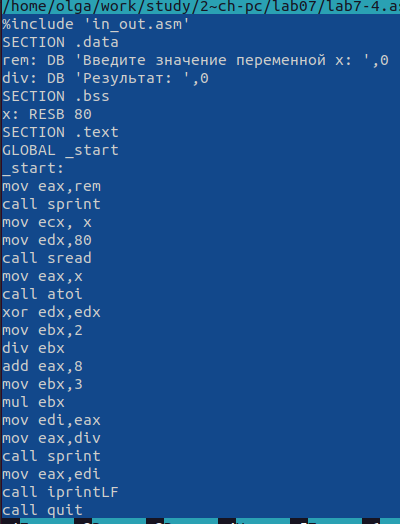


Рис. 24: Редактирование файла

Транслирую текст программы файла в объектный файл. Выполняю компоновку объектного файла (рис. 25).

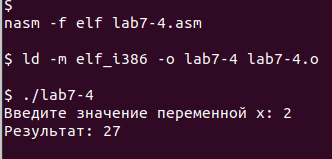


Рис. 25: Исполнение файла

# 5 Выводы

Я освоила арифметические инструкции языка ассемблера NASM.

# Список литературы

[1. Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/1584622/mod_resource/content/1/Лабораторная%20работа%20№3.pdf)