Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Аветисян Алина Эдуардовна НММбд-01-23

Содержание

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

С помощью утилиты mkdir создаю директорию, в которой буду создавать файлы с программами для лабораторной работы №6.

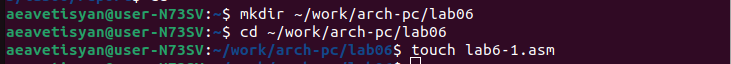


Figure 1: Перехожу в созданный каталог с помощью утилиты cd. С помощью утилиты touch создаю файл lab6-1.asm.

Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp, т.к. он будет использоваться в других программах.

Figure 2: Копирование in_out.asm.

Figure 2: Копирование in\_out.asm.

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода значения регистра eax.

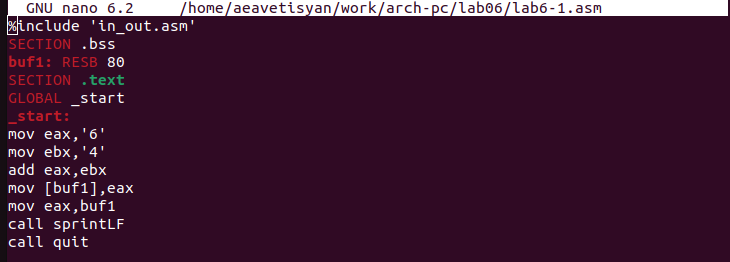


Figure 3: Открываю файл lab6-1.asm с помощью текстового редактора nano.

Создаю исполняемый файл программы и запускаю его. Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.

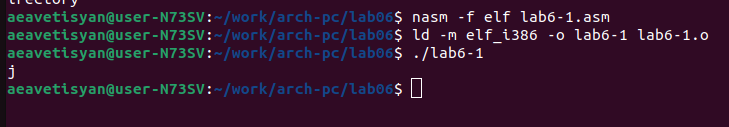


Figure 4: Запуск исполняемого файла.

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4.

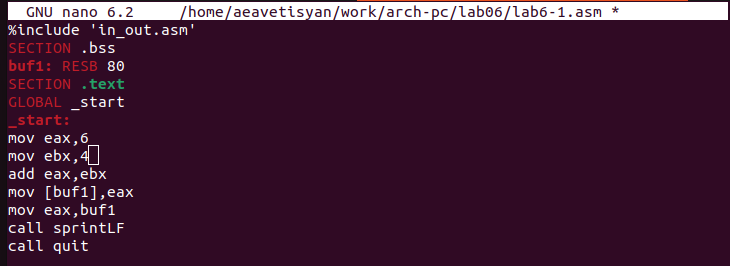
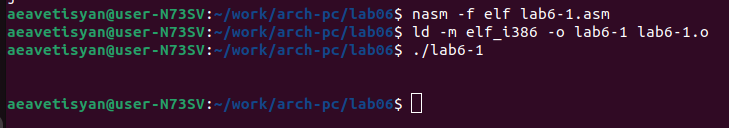


Figure 5: Редактирование файла.

Создаю новый исполняемый файл программы и запускаю его. Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

{#fig:001 width=70%

Создаю новый файл lab6-2.asm с помощью утилиты touch.

Figure 6: Создание файла.

Figure 6: Создание файла.

Ввожу в файл текст другой программы для вывода значения регистра eax.

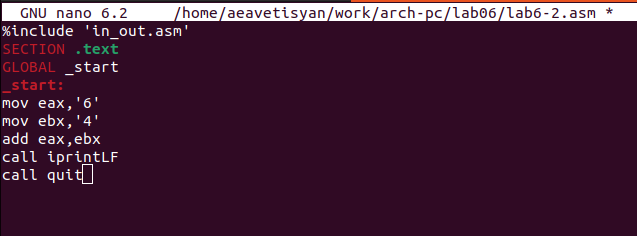


Figure 7: Редактирование файла.

Создаю и запускаю исполняемый файл lab6-2.asm . Теперь выводится число 106, пото- му что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”.

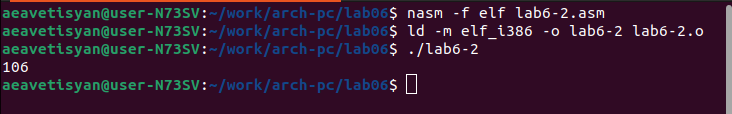


Figure 8: Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4.

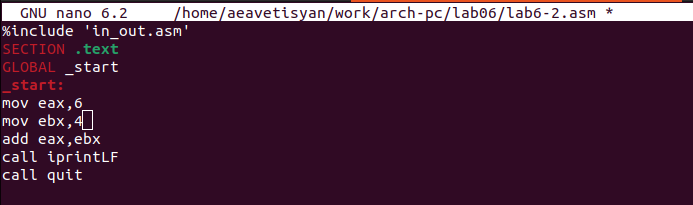


Figure 9: Редактирование файла.

Создаю и запускаю новый исполняемый файл. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

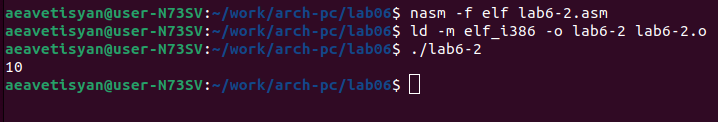


Figure 10: Запуск исполняемого файла.

Заменяю в тексте программы функцию iprintLF на iprint.

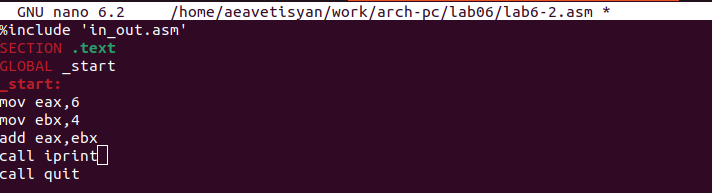


Figure 11: Редактирование файла.

Создаю и запускаю новый исполняемый файл. Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией iprintLF, а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF.

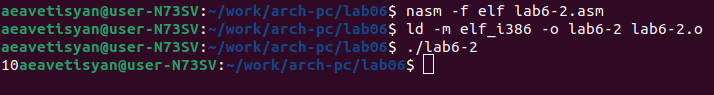


Figure 12: Запуск исполняемого файла.

Создаю файл lab6-3.asm с помощью утилиты touch.

Figure 13: Создание файла.

Figure 13: Создание файла.

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

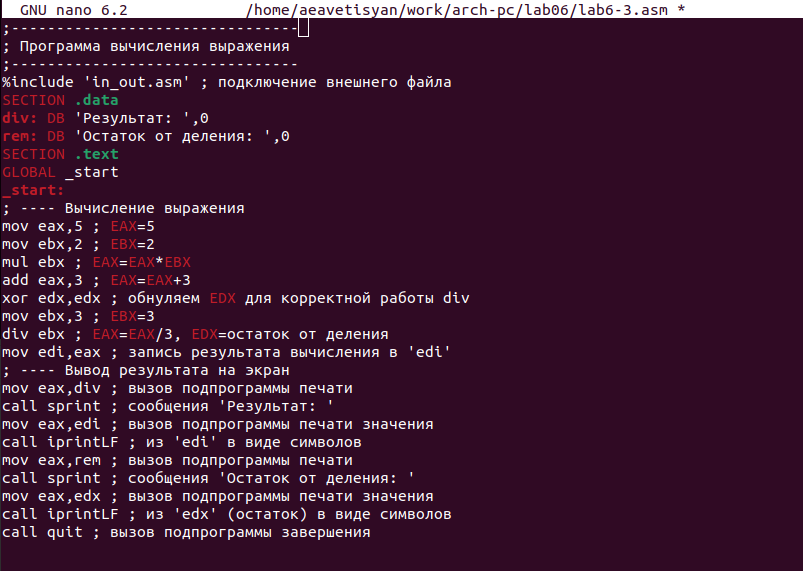


Figure 14: Редактирование файла.

Создаю исполняемый файл и запускаю его.

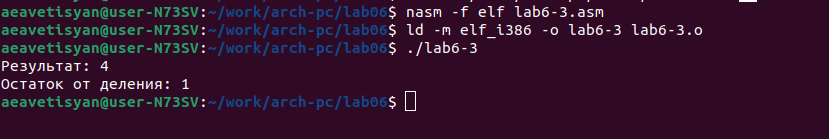


Figure 15: Запуск исполняемого файла.

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

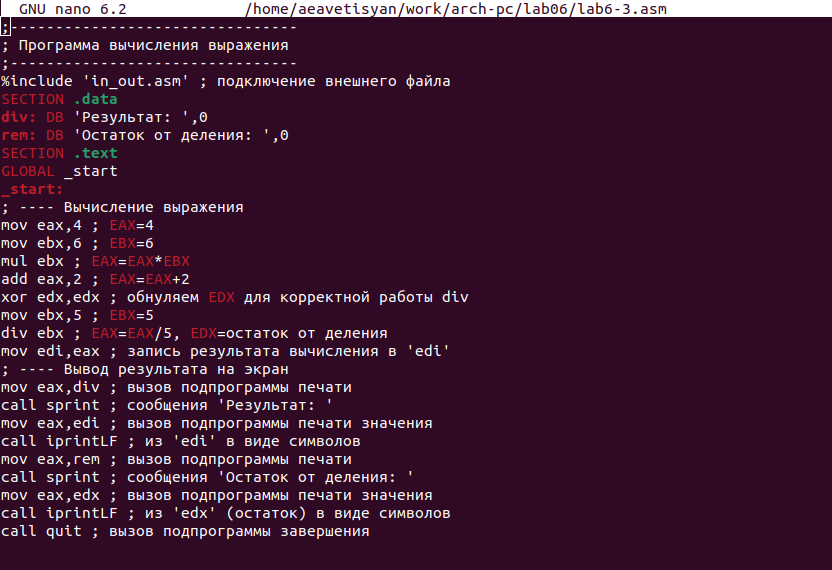


Figure 16: Изменение программы.

Создаю и запускаю новый исполняемый файл.

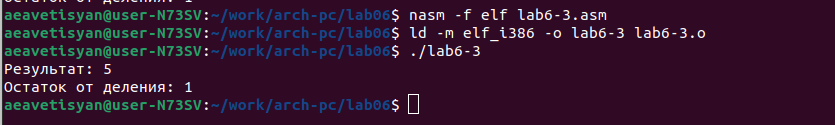


Figure 17: Запуск исполняемого файла.

Создаю файл variant.asm с помощью утилиты touch.

Figure 18: Создание файла.

Figure 18: Создание файла.

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета.

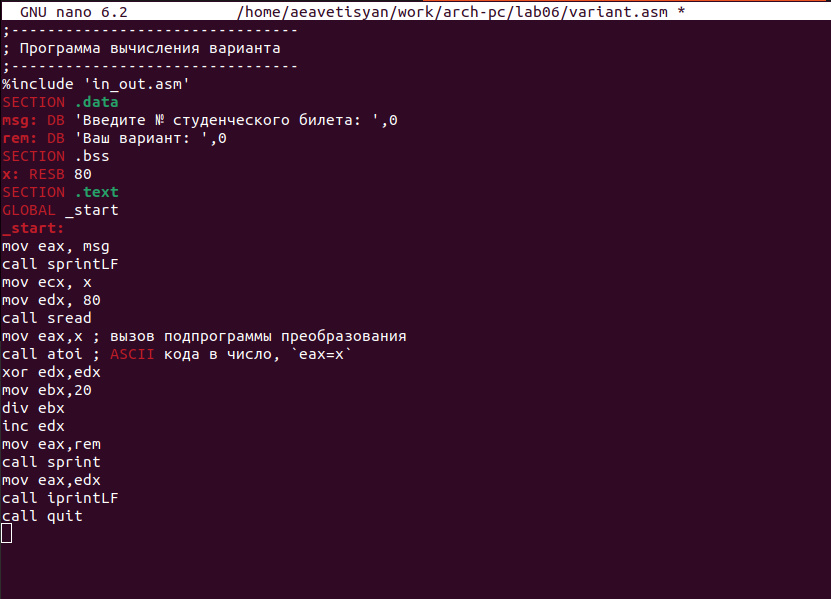


Figure 19: Редактирование файла.

Создаю и запускаю исполняемый файл. Ввожу номер своего студенческого билета с клавиатуры, программа вывела, что мой вариант 15.

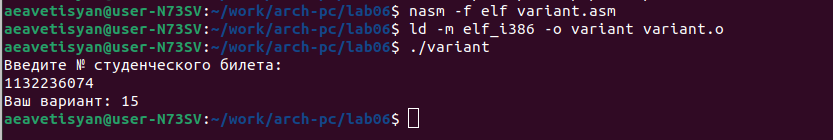


Figure 20: Запуск исполняемого файла.

### 2.0.1 Ответы на вопросы.

1. За вывод сообщения “Ваш вариант” отвечают строки кода: mov eax, rem call sprint.
2. Инструкция mov ecx, x используется, чтобы положить адрес вводимой стро- ки x в регистр ecx mov edx, 80 - запись в регистр edx длины вводимой строки call sread - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры.
3. call atoi используется для вызова подпрограммы из внешнего файла, кото- рая преобразует ascii-код символа в целое число и записывает результат в регистр eax.
4. За вычисления варианта отвечают строки: xor edx,edx ; обнуление edx для корректной работы div mov ebx,20 ; ebx = 20 div ebx ; eax = eax/20, edx - остаток от деления inc edx ; edx = edx + 1.
5. При выполнении инструкции div ebx остаток от деления записывается в регистр edx.
6. Инструкция inc edx увеличивает значение регистра edx на 1.
7. За вывод на экран результатов вычислений отвечают строки: mov eax,edx call iprintLF.

# 3 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm с помощью утилиты touch.

Figure 21: Создание файла.

Figure 21: Создание файла.

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения f(x) = (5 + 𝑥)^2 − 3. Это выражение было под вариантом 15.

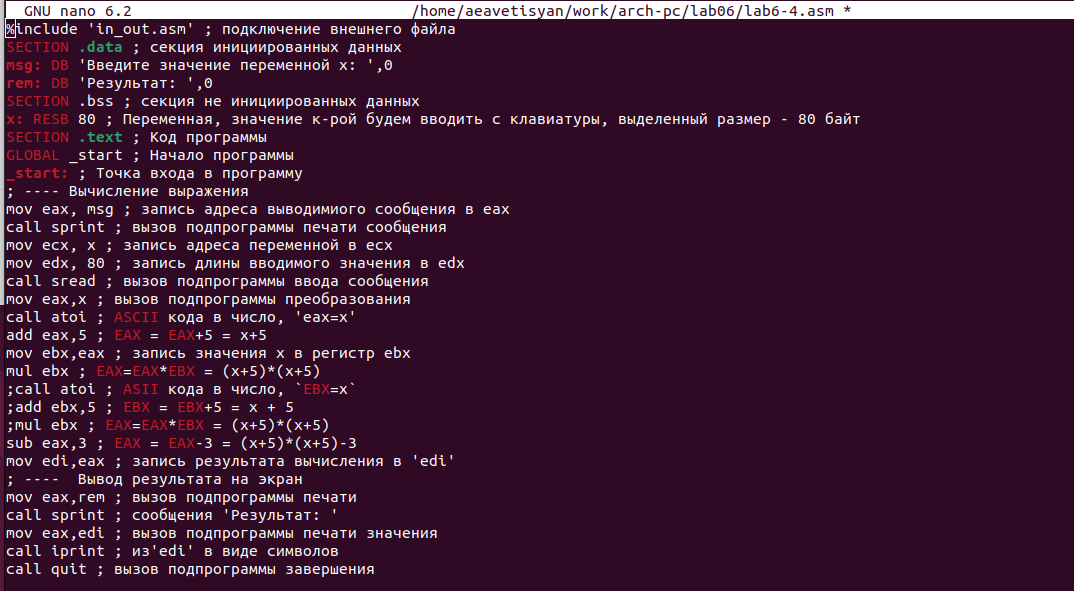


Figure 22: Написание программы.

Создаю и запускаю исполняемый файл. При вводе значения 5, вывод 97.

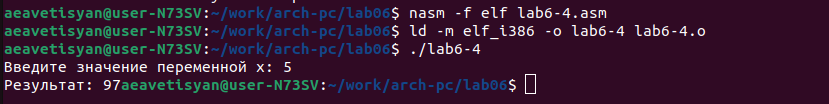


Figure 23: Запуск исполняемого файла.

Провожу еще один запуск исполняемого файла для проверки работы програм- мы с другим значением на входе. Программа отработала верно. При вводе значения 1, вывод 33.

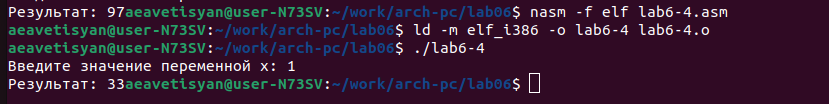


Figure 24: Запуск исполняемого файла.

# 4 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.