Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Пономарева Татьяна Александровна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

## 2.1 Символьные и численные данные в NASM

Создаю каталог для программам лабораторной работы № 6 , далее перехожу в него и создаю файл lab6-1.asm (рис. 1).

Терминал. Создание каталога lab06 и файла lab6-1.asm

Рис. 1: Терминал. Создание каталога lab06 и файла lab6-1.asm

Ввожу в файл lab6-1.asm текст программы из листинга 6.1 (рис. 2).

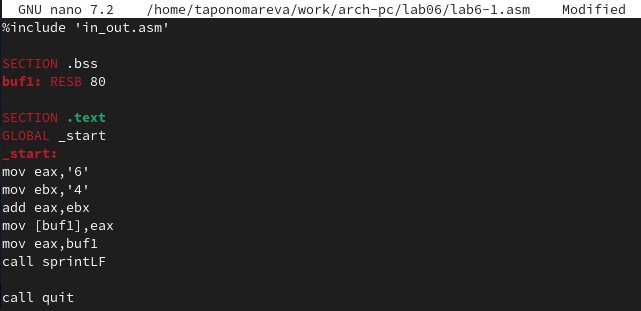


Рис. 2: Окно Midnight Commander. Содержание файла lab6-1.asm

Создаю исполняемый файл lab6-1 и запускаю его (рис. 3).

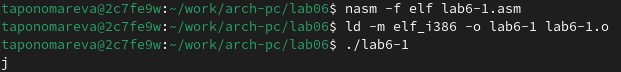


Рис. 3: Терминал. Создание исполняемого файла lab6-1 и его запуск

Далее изменяю текст программы и вместо символов, пишу в регистры числа (рис. 4).

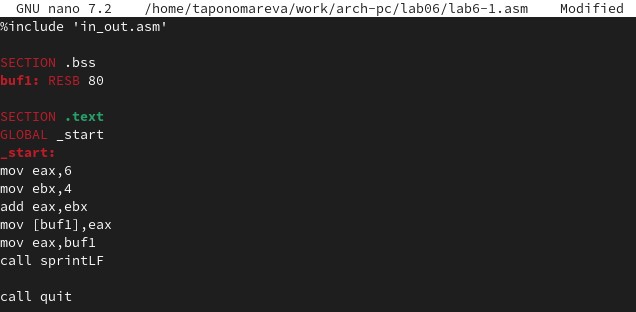


Рис. 4: Окно Midnight Commander. Содержание файла lab6-1.asm

Создаю исполняемый файл lab6-1 и запускаю его (рис. 5).

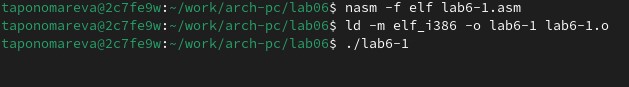


Рис. 5: Терминал. Создание исполняемого файла lab6-1 и его запуск

Согласно таблице ASCII, код 10 соответствует символу перевода строки (newline), который обозначается как LF (Line Feed). Символ с кодом 10 (перевод строки) не отображается визуально при выводе на экран. Вместо этого он вызывает действие — переход на новую строку.

Создаю файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 при помощи команды touch ~/work/arch-pc/lab06/lab6-2.asm (рис. 6).

Терминал. Создание файла lab6-2.asm

Рис. 6: Терминал. Создание файла lab6-2.asm

Ввожу в него текст программы из листинга 6.2 (рис. 7).



Рис. 7: Окно Midnight Commander. Содержание файла lab6-2.asm

Создаю исполняемый файл lab6-2 и запускаю его (рис. 8).

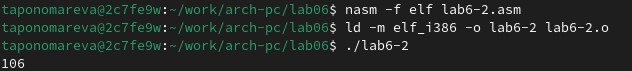


Рис. 8: Терминал. Создание исполняемого файла lab6-2 и его запуск

Аналогично предыдущему примеру изменяю символы на числа (рис. 9).

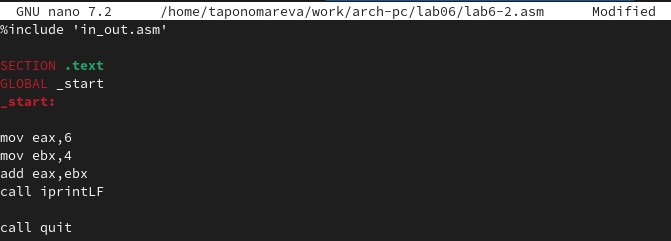


Рис. 9: Окно Midnight Commander. Содержание файла lab6-2.asm

Создаю исполняемый файл lab6-2 и запускаю его. В терминале выводится 10 (рис. 10).

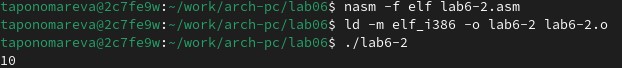


Рис. 10: Терминал. Создание исполняемого файла lab6-2 и его запуск

Заменяю функцию iprintLF на iprint (рис. 11).

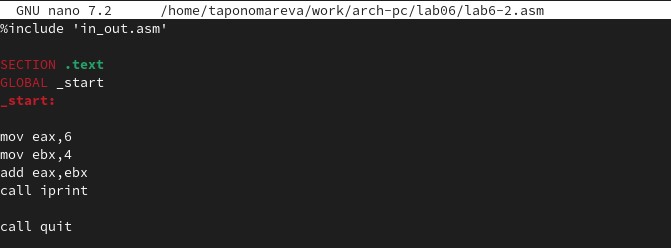


Рис. 11: Окно Midnight Commander. Содержание файла lab6-2.asm

Создаю исполняемый файл lab6-2 и запускаю его. iprintLF добавляет перевод строки после числа, а iprint выводит число без переноса строки (рис. 12).

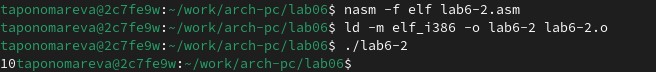


Рис. 12: Терминал. Создание исполняемого файла lab6-2 и его запуск

## 2.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 при помощи команды touch ~/work/arch-pc/lab06/lab6-3.asm (рис. 13).

Терминал. Создание файла lab6-3.asm

Рис. 13: Терминал. Создание файла lab6-3.asm

Ввожу в lab6-3.asm текст программы из листинга 6.3 (рис. 14).

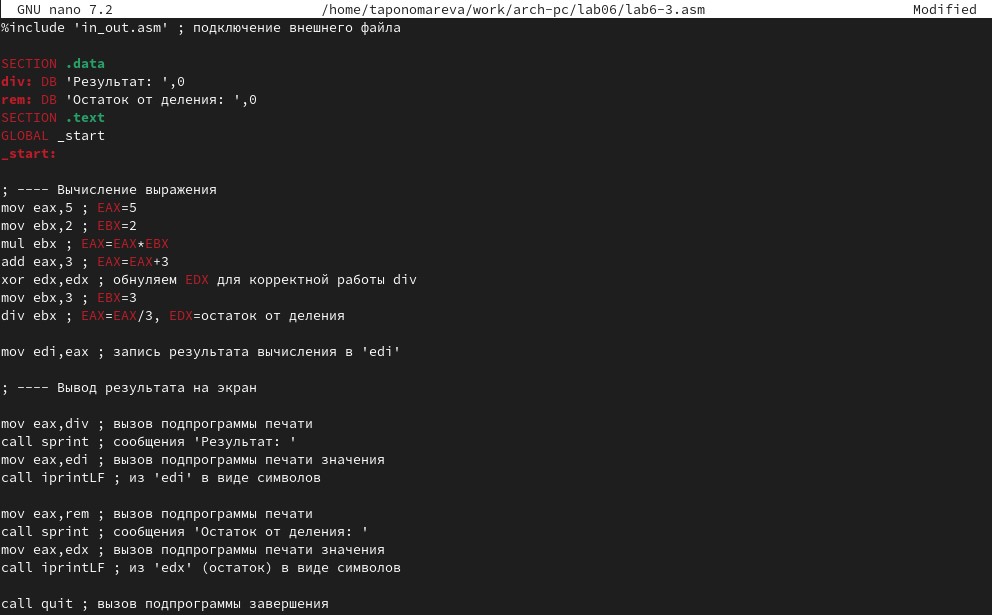


Рис. 14: Окно Midnight Commander. Содержание файла lab6-3.asm

Создаю исполняемый файл lab6-3 и запускаю его (рис. 15).

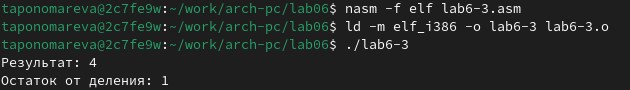


Рис. 15: Терминал. Создание исполняемого файла lab6-3 и его запуск

Изменяю текст программы для вычисления выражения f(x) = (4 \* 6 + 2)/5 (рис. 16).

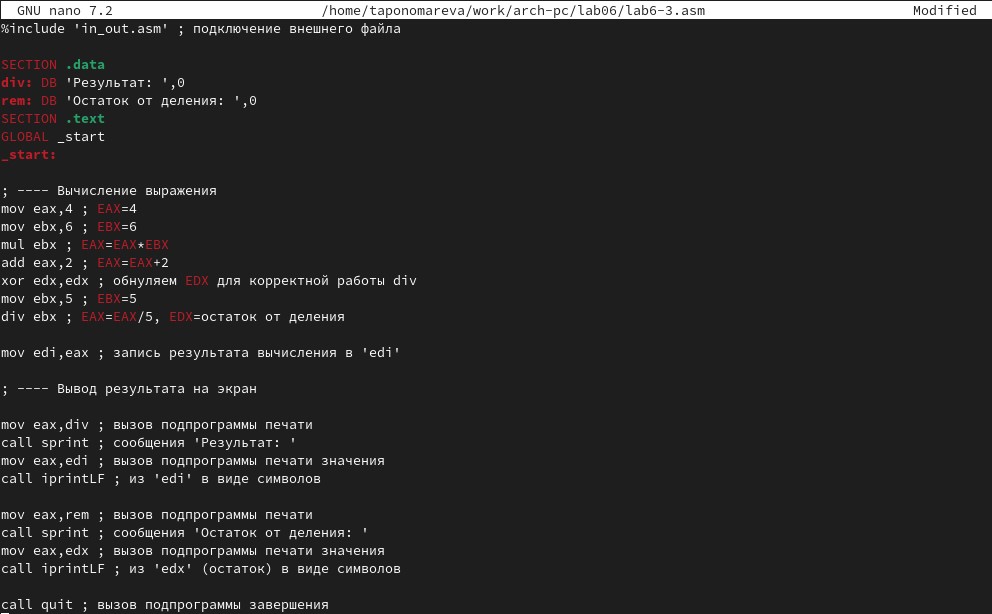


Рис. 16: Окно Midnight Commander. Содержание файла lab6-3.asm

Создаю исполняемый файл lab6-3 и проверяю его работу (рис. 17).

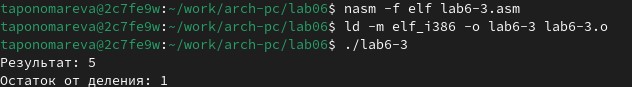


Рис. 17: Терминал. Создание исполняемого файла lab6-3 и его запуск

Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 18).

Терминал. Создание файла variant.asm

Рис. 18: Терминал. Создание файла variant.asm

Ввожу в variant.asm текст программы из листинга 6.4 (рис. 19).

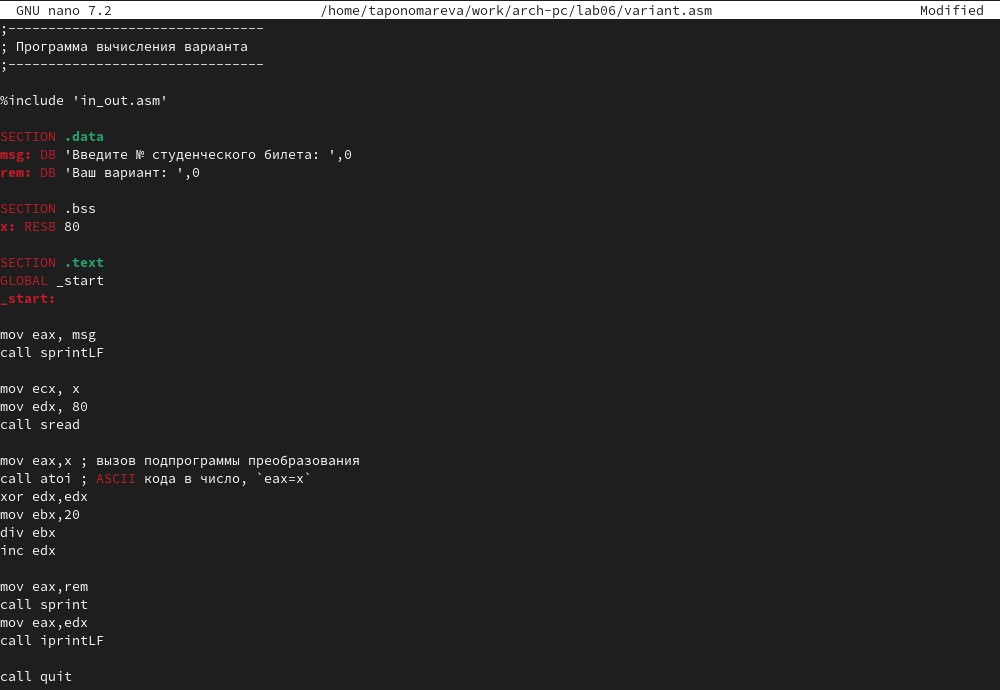


Рис. 19: Окно Midnight Commander. Содержание файла variant.asm

Создаю исполняемый файл variant и запускаю его. Ввожу свой номер студенческого билета - 1132246742. Получаю вариант 3 (рис. 20).

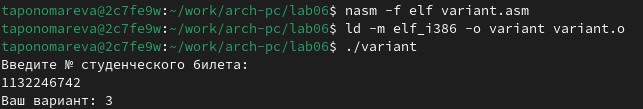


Рис. 20: Терминал. Создание исполняемого файла variant и его запуск

Программа делит номер студенческого билета на 20, сохраняет остаток в регистре edx, а затем увеличивает его на 1 с помощью инструкции inc edx, что приводит к тому, что результат деления с остатком дает вариант 3.

1. За вывод на экран сообщения “Ваш вариант:” отвечают следующие строки: mov eax,rem call sprint Эти строки загружают строку “Ваш вариант:” в регистр eax и затем вызывают подпрограмму sprint, которая отвечает за вывод строки на экран.
2. mov ecx, x mov edx, 80 call sread Эти инструкции загружают адрес переменной для ввода данных и максимальную длину строки, затем вызывают подпрограмму sread для чтения ввода с клавиатуры.
3. Инструкция call atoi вызывается для преобразования строки, содержащей ASCII-коды, в целое число, которое сохраняется в регистре eax.
4. За вычисления варианта отвечают следующие строки: mov eax,x ; Загружаем число из переменной x в регистр eax call atoi ; Преобразуем строку в число (номер студенческого билета) xor edx,edx ; Обнуляем регистр edx, чтобы он использовался для остатка от деления mov ebx,20 ; Загружаем делитель (20) в регистр ebx div ebx ; Делим eax на ebx, результат в eax, остаток в edx inc edx ; Увеличиваем остаток на 1, получаем вариант
5. Остаток от деления при выполнении инструкции div ebx записывается в регистр edx.
6. Инструкция inc edx используется для увеличения остатка от деления в регистре edx на 1.
7. За вывод на экран результата вычислений отвечают следующие строки: mov eax,rem ; Загружаем адрес строки ‘Ваш вариант:’ в регистр eax call sprint ; Выводим строку “Ваш вариант:” mov eax,edx ; Загружаем результат (вариант) из регистра edx в eax call iprintLF ; Выводим значение из eax (номер варианта)

# 3 Задание для самостоятельной работы

Создаю файл lab6-4.asm в каталоге ~/work/arch-pc/lab06 при помощи touch ~/work/arch-pc/lab06/lab6-4.asm (рис. 21).

Терминал. Создание файла lab6-4.asm

Рис. 21: Терминал. Создание файла lab6-4.asm

Вписываю текст программы в файл lab6-4.asm для вычисления выражения y=f(x). У меня вариант 3, т.е. y=(2 + x)^2, x1=2, x2=8 (рис. 22).

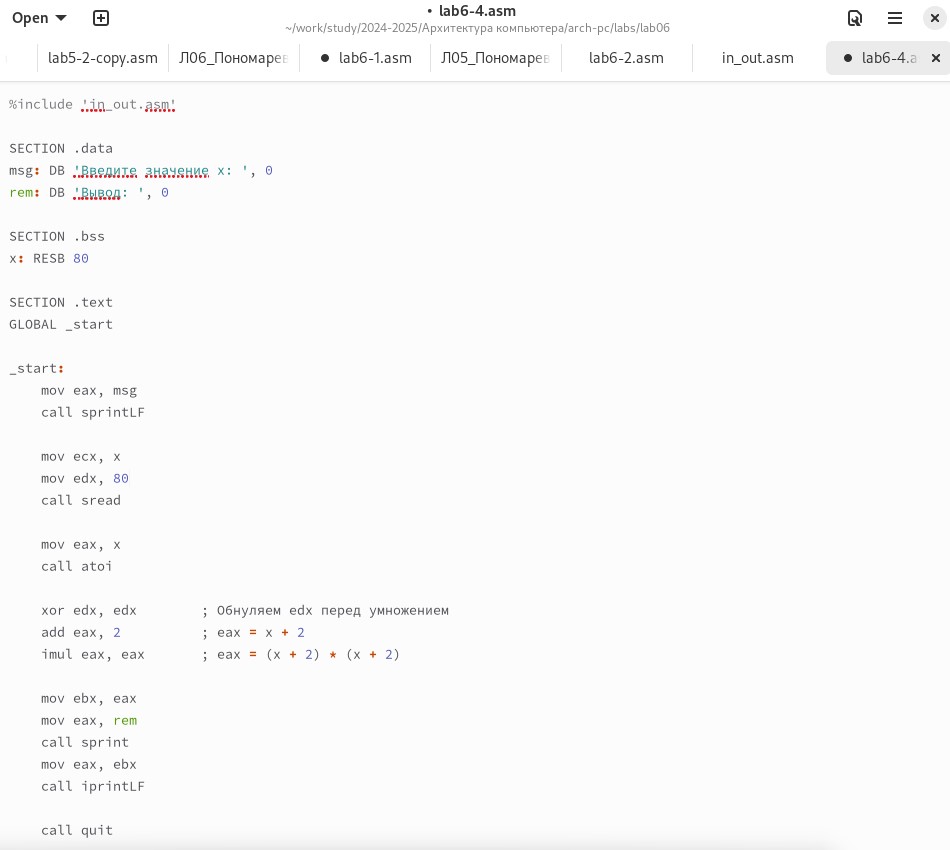


Рис. 22: Окно Midnight Commander. Содержание файла lab6-4.asm

Ответ. При x1=2, имеем 16, при x2=8, имеем 100 (рис. 23).

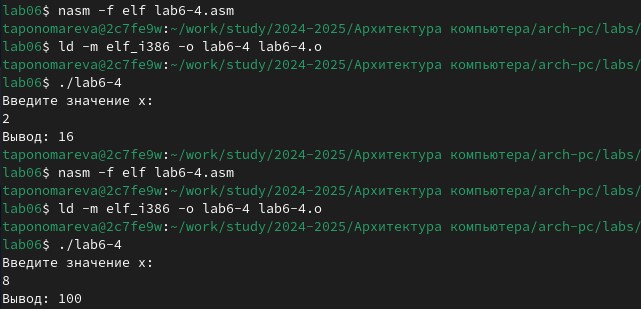


Рис. 23: Терминал. Проверка вычисления программы варианта 3.

Загружаю на GitHub.

# 4 Выводы

Были освоены арифметические инструкции языка ассемблера NASM.

# Список литературы

1. [Курс на ТУИС](https://esystem.rudn.ru/course/view.php?id=112)
2. [Лабораторная работа №6](https://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)