Отчёта по лабораторной работе №6

Арифметические операции в NASM

Хохлачева Полина Дмитриевна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

Создаём каталог для программам лабораторной работы №6

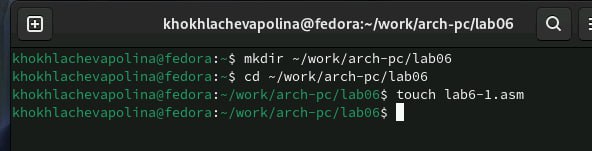


Рис. 1: Создаём с помощью команды mkdir

Открываем файл и заполняем в соответствие с листингом

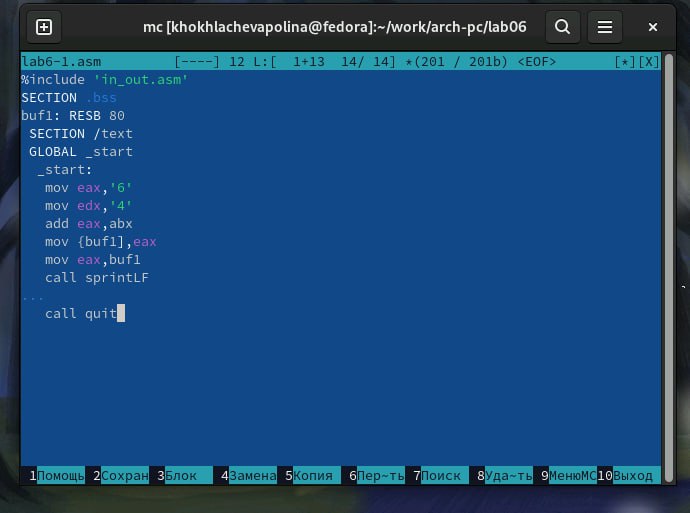


Рис. 2: Заполняем файл

Создаём исполняемый файл и запускаем его

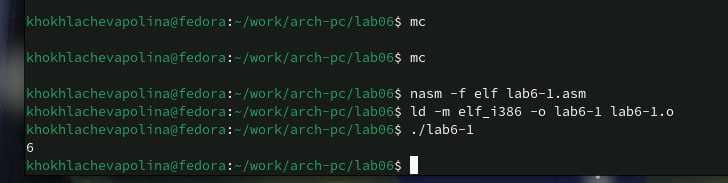


Рис. 3: Запускаем файл и смотрим работу

Открываем файл для редактирования

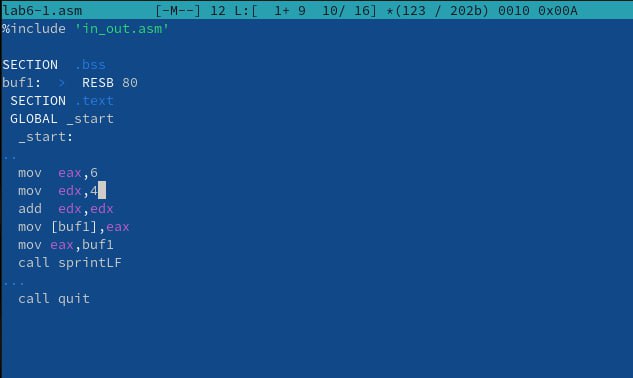


Рис. 4: Убираем кавычки

Создаём файл и запускаем его

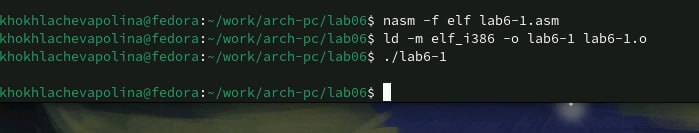


Рис. 5: Запускаем смотрим работу

Создаём новый файл в каталоге

Создаём файл

Рис. 6: Создаём файл

Заполняем файл в соответствие с листингом

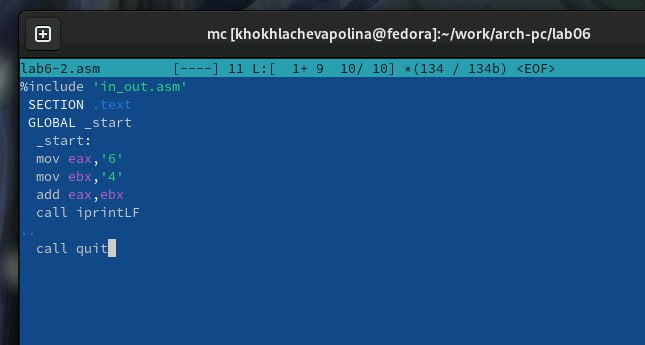


Рис. 7: Заполняем файл

Создаём исполняемый файл и запускаем его

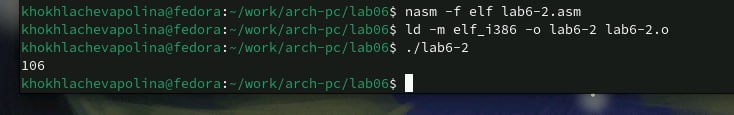


Рис. 8: Смотрим на работу команды

Открываем файл и редактируем его

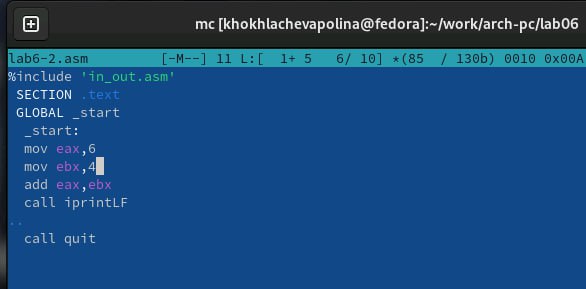


Рис. 9: Убираем кавычки

Создаём исполняемый файл и запускаем его

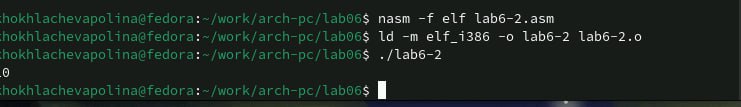


Рис. 10: Смотрим на работу программы

Открываем файл и редактируем его

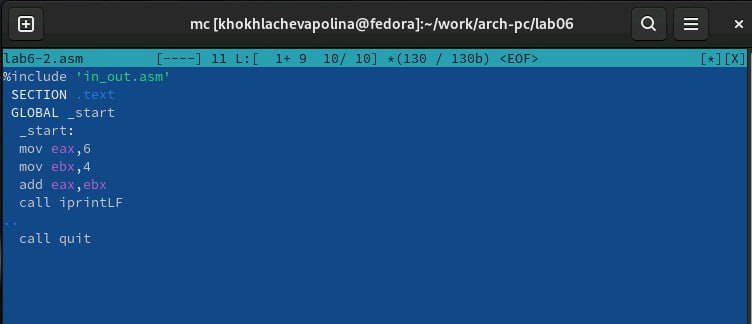


Рис. 11: Изменяем файл

Создаём исполняемый файл и запускаем его

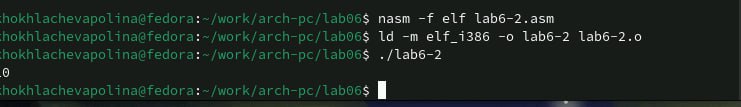


Рис. 12: Смотрим на работу программы

#Выполнение арифметических операций NASM

Создаём новый файл в каталоге

Создаём файл

Рис. 13: Создаём файл

Открываем файл и редактируем в соответствие с листингом



Рис. 14: Заполняем файл

Создаём исполняемый файл и запускаем его

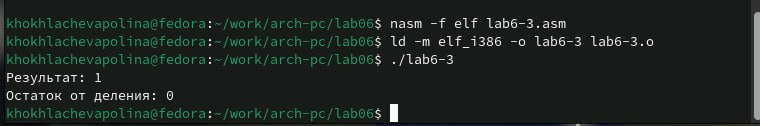


Рис. 15: Смотрим на результат работы программы

Создаём файл для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5.

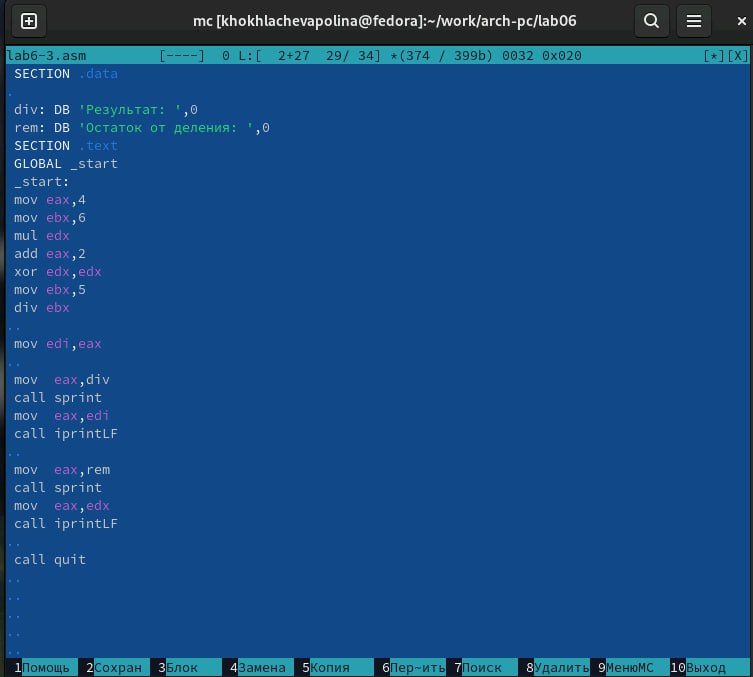


Рис. 16: Редактируем файл

Компилируем файл и запускаем программу

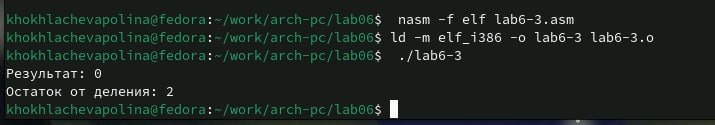


Рис. 17: Смотрим на результат работы

Создаём новый файл в каталоге

Создаём файл

Рис. 18: Создаём файл

Открываем файл и редактируем в соответствие с листингом

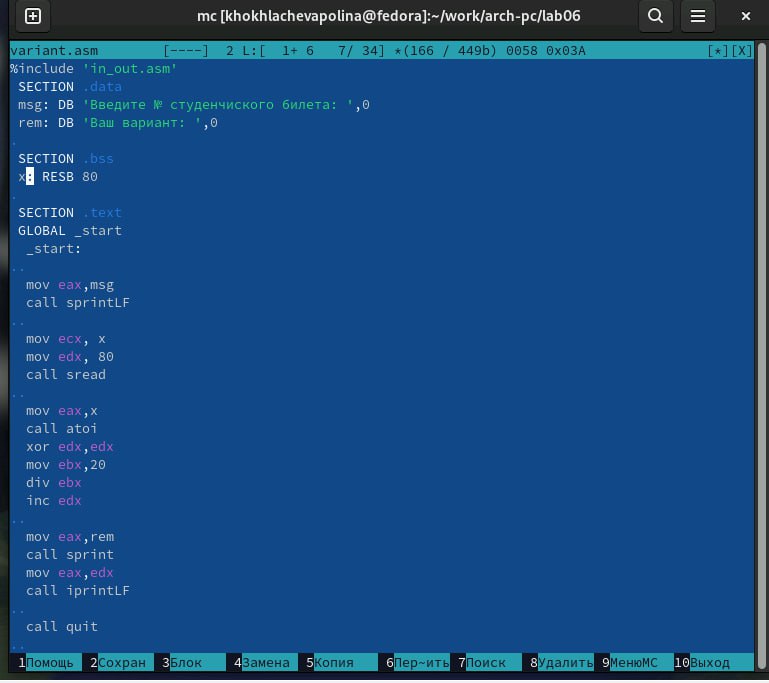


Рис. 19: Смотрим на результат работы

Компилируем файл и запускаем его

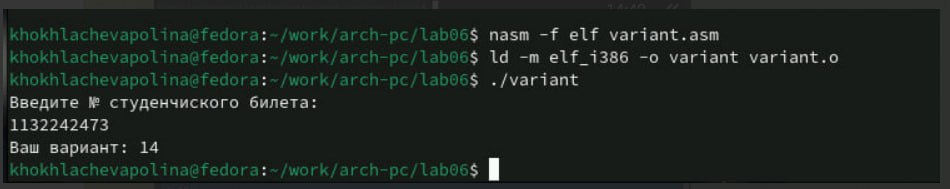


Рис. 20: Проверяем работу программы

# 3 Ответы на вопросы по программе

1. Строка ‘mov eax,msg’ и ‘call sprintLF’
2. Эти инструкции используются для считывания строки, введённой пользователем, и её сохранения в памяти по адресу x. • mov ecx, x — задаёт адрес для сохранения строки. • mov edx, 80 — задаёт максимальную длину считываемой строки. • call sread — вызывает процедуру чтения строки из стандартного ввода.
3. Инструкция call atoi используется для преобразования строки (ASCII-кодов), введённой пользователем, в число, которое будет сохранено в регистре eax. 4.xor edx, edx mov ebx, 20 div ebx inc edx
4. Остаток от деления записывается в регистр edx
5. Инструкция inc edx увеличивает значение в регистре edx на 1. В данном случае это используется для прибавления 1 к остатку от деления, чтобы получить номер варианта.
6. mov eax, rem call sprint mov eax, edx call iprintLF

#Задание для самостоятельной работы

Создаём новый файл в каталоге

Создаём файл

Рис. 21: Создаём файл

Открываем его и заполняем, чтобы решалось выражение (x/2+8)\*3

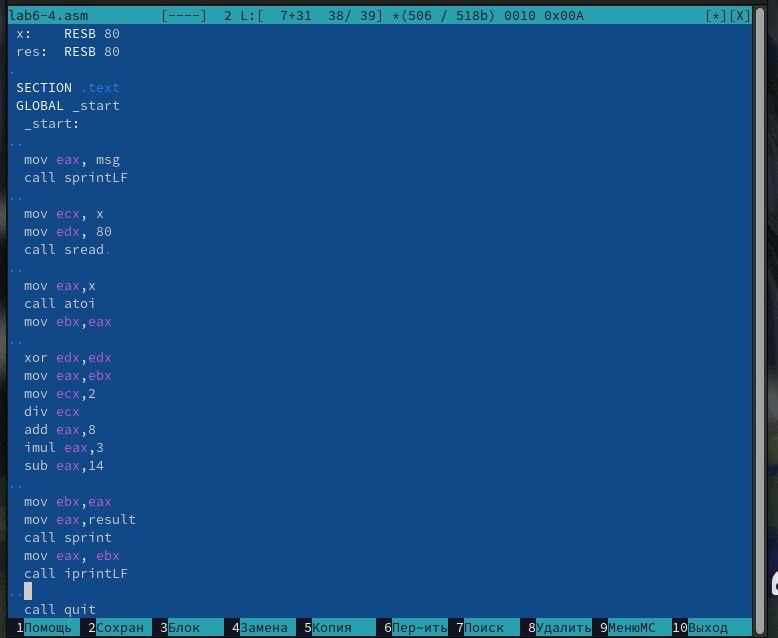


Рис. 22: Создаём файл

Компилируем программу и проверяем для х=1

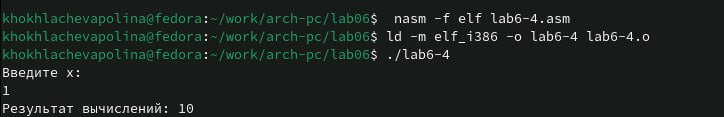


Рис. 23: Создаём файл

Компилируем программу и проверяем для х=4

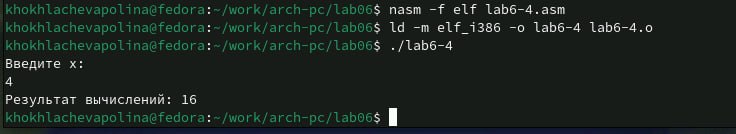


Рис. 24: Создаём файл

# 4 Выводы

Мы приобрели навыки создания исполнительный файлов для решения выражений и освоили арифметический инструкции

# Список литературы