Лабораторная работа № 6

Дисциплина: Архитектура компьютера

Жибицкая Евгения Дмитриевна

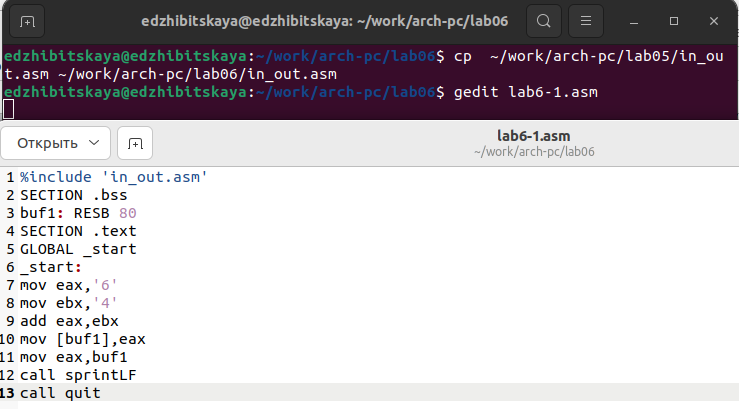
Содержание

# 1 Цель работы

Продолжение знакомства с языком ассемблера, освоение арифметических операций NASM.

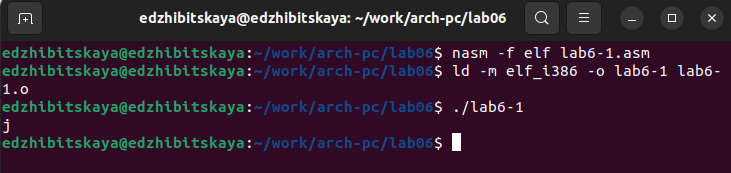
# 2 Выполнение лабораторной работы

Начнем работу с того, что cоздадим каталог lab06 и файл lab6-1.asm(mkdir ~/work/arch-pc/lab06; cd ~/work/arch-pc/lab06; touch lab6-1.asm) Скопируем файл in\_out.asm в lab06 и вставим код из Листинг 6.1 в lab6-1.asm(рис. ??).



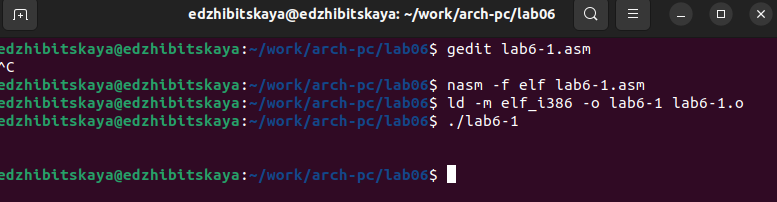
Листинг 6.1

Создадим исполняемый файл и запустим его(рис. ??).



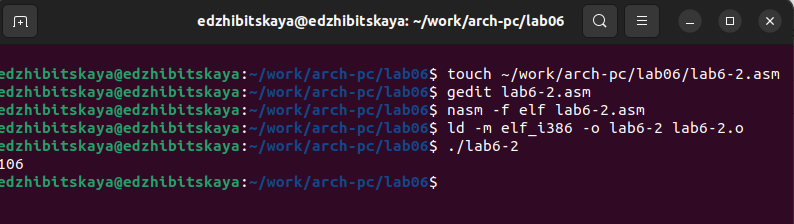
Запуск программы

Затем изменим текст программы и вместо символов, запишем в регистры числа, так как в предыдущий раз программа не вывела ожидаемый результат, также создадим исполняемый файл и запустим программу (рис. ??).



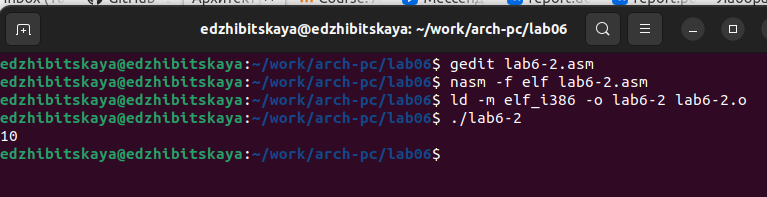
Запуск измененной программы

Далее создадим файл lab6-2.asm, введем в него текст из Листинг 6.2, создадим исполняемый файл и запустим(рис. ??).



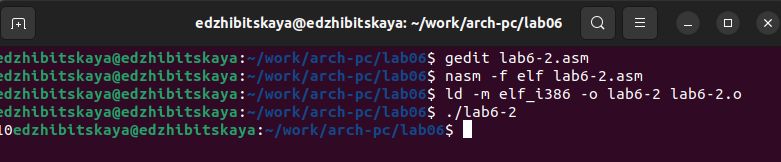
Запуск lab6-2

Опять заменим символы на числа и запустим программу.(рис. ??).



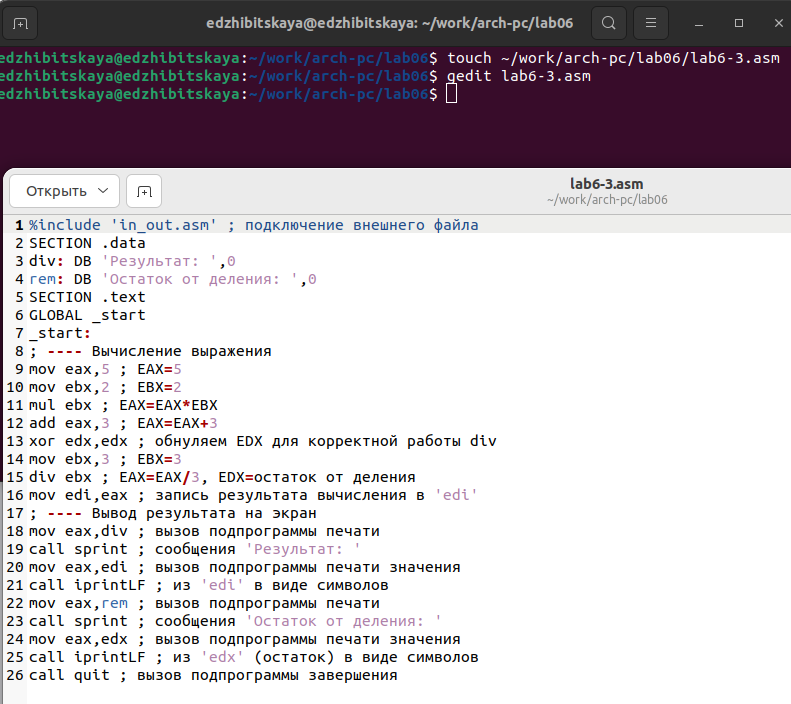
Повторный запуск lab6-2

В конце запустим программу еще раз, заменив iprintLF на iprint(отличаются наличием/отсутсвием переноса строки)(рис. ??).



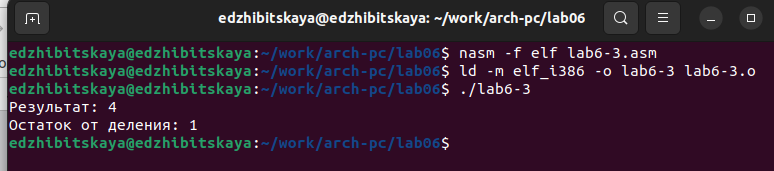
Запуск с iprint

Еще создадим файл lab6-3.asm и заполним его текстом из Листинг 6.3(рис. ??).



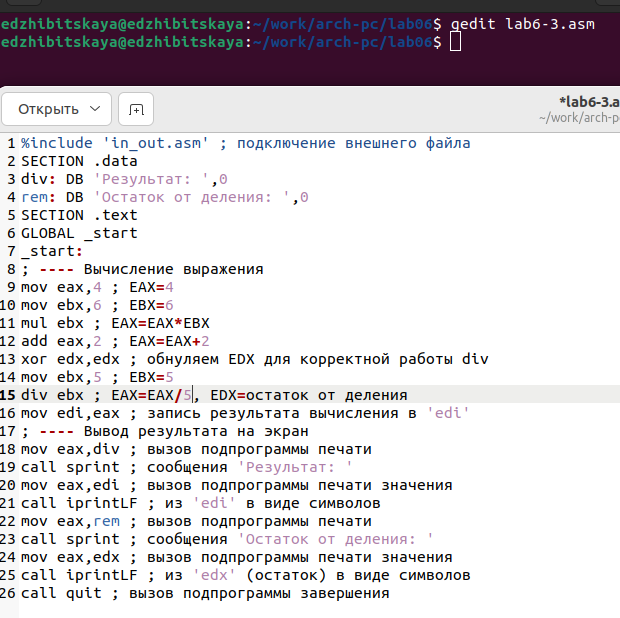
Листинг 6.3

Создадим исполняемый файл и запустим его.(рис. ??).



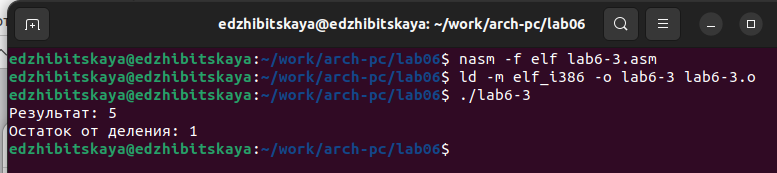
Запуск lab6-3

Изменим имеющуюся программу, для вычисления f(x) = (4 \* 6 + 2)/5 (рис. ??].



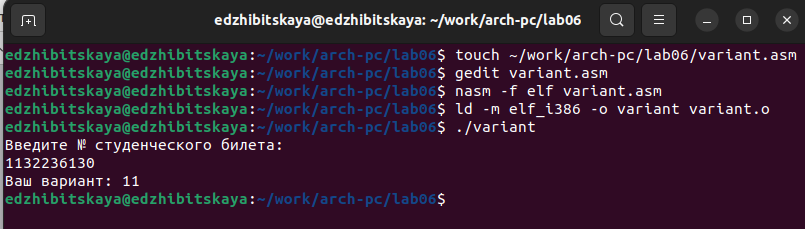
Изменение текста программы lab6-3

Затем запустим и проверим корректность работы (рис. ??).



Запуск измененной программы lab6-3

В конце создадим файл variant.asm, заполним его текстом из Листинг 6.4, создадим исполняемый файл и запустим(рис. ??).



Запуск variant

Далее ответим на следующие вопросы:

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

mov eax,rem call sprint

1. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread

Для считывания переменной x.

1. Для чего используется инструкция “call atoi”?

Для преобразования ascii-код символа в целое число.

1. Какие строки листинга 6.4 отвечают за вычисления варианта?

xor edx,edx mov ebx,20 div ebx inc edx

1. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

AH

1. Для чего используется инструкция “inc edx”?

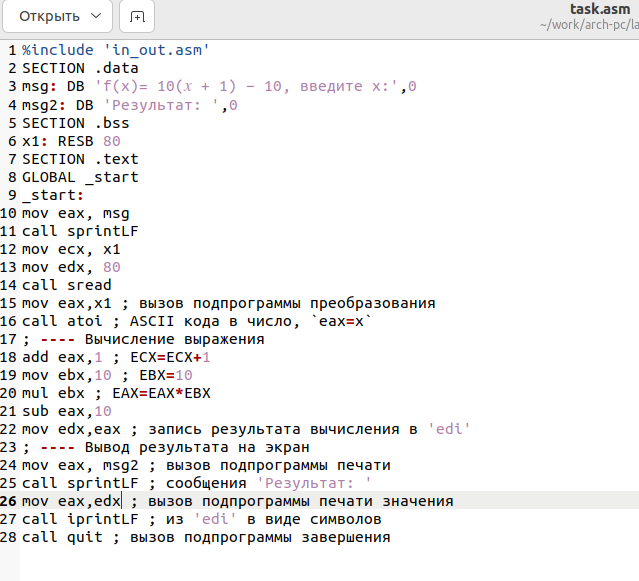
Для прибавления единицы к значению регистра ebx.

1. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

mov eax,edx call iprintLF

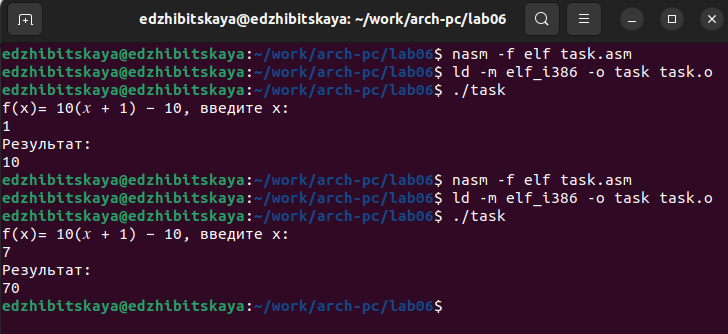
# 3 Задание для самостоятельной работы

Создадим файл task.asm для выполнения задания, напишем туда программу для вычисления выражения(10(x + 1) − 10), x = 1, x = 7(рис. ??).



Написание программы

Запустим файл и убедимся в корректности работы(рис. ??).



Запуск кода

# 4 Выводы

В ходе работы было произведено знакомство с арифметическими инструкциями языка ассемблера NASM, изучены некоторые его особенности, а также написана небольщая программа вычисления выражения.