Лабораторная работа №7

Архитектура компьютера

Голованова Мария Константиновна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Создать программы вывода символьных и численных значений, программы вычисления арифметических выражений. Создать программу вычисления выражения y = f(x), выбрав вариант из таблицы 7.3 по номеру, полученному в при выполнении лабораторной работы.

# 3 Теоретическое введение

Арифметические инструкций языка ассемблера NASM - операции, позволяющие производить арифметические вычисления в ассемблере. К ним относятся такие операции как команда целочисленного сложения add, команда целочисленного вычитания sub, команды инкремента и декремента inc (от англ. increment) и dec (от англ. decrement), команда изменения знака операнда neg, команды умножения mul и imu, команды деления div и idiv . # Выполнение лабораторной работы

## 3.1 Символьные и численные данные в NASM

Я создала каталог для программам лабораторной работы №7, перешла в него и создала файл lab7-1.asm (рис. 1).

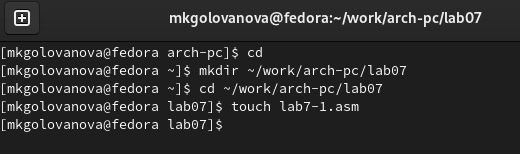


Рис. 1: Создание каталога для программам лабораторной работы №7 и файла lab7-1.asm

Я введла в файл lab7-1.asm текст программы из листинга 7.1 (рис. 2).

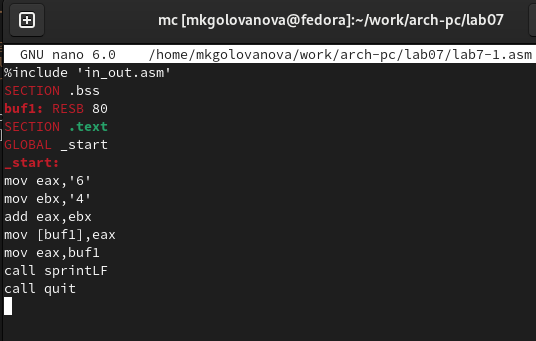


Рис. 2: Введение текста программы из листинга 7.1

Я создала исполняемый файл и запустите его (рис. 3).

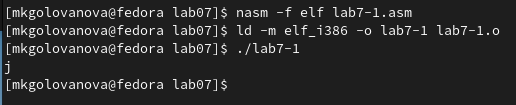


Рис. 3: Создание и запуск исполняемого файла lab7-1

В данном случае при выводе значения регистра eax я ожидала увидеть число 10. Однако результатом будет символ j. Это происходит потому, что кодсимвола 6 равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа 4 – 00110100 (52). Команда add eax,ebx записала в регистр eax сумму кодов – 01101010 (106), что в свою очередь является кодом символа j. Я исправила текст программы и вместо символов записала в регистры числа, заменив строки mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4 (рис. 4).

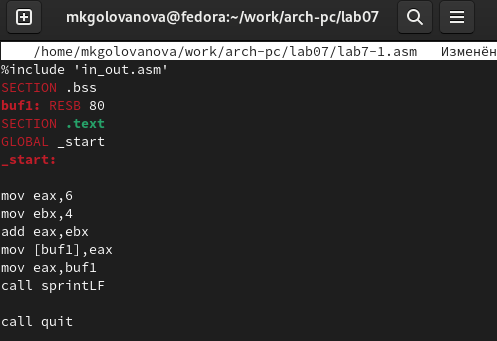


Рис. 4: Исправление текста файла lab7-1.asm

Я создала исполняемый файл и запустила его (рис. 5).

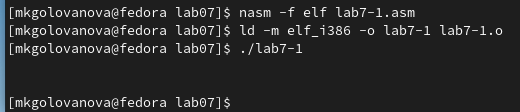


Рис. 5: Создание и запуск исполняемого исправленного файла lab7-1

В данном случае выводится символ с кодом 10. Пользуясь таблицей ASCII, я определа, что код 10 соответствует символу ‘LF, /n’, причём символ не отображается при выводе на экран.

Я создала файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и ввела в него текст программы из листинга 7.2 (рис. 6, рис. 7).

Рис. 6: Создание файла lab7-2.asm

Рис. 6: Создание файла lab7-2.asm



Рис. 7: Введение текста программы из листинга 7.2

Я создала исполняемый файл и запустила его (рис. 8).

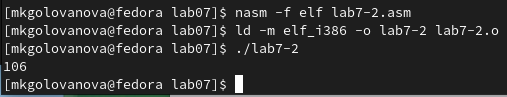


Рис. 8: Создание и запуск исполняемого файла lab7-2

В результате работы программы я получила число 106. В данном случае, как и в первом, команда add складывает коды символов ‘6’ и ‘4’ (54+52=106). Однако, в отличии от программы из листинга 7.1, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число. Я изменила символы на числа, заменив строки mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4 (рис. 9).

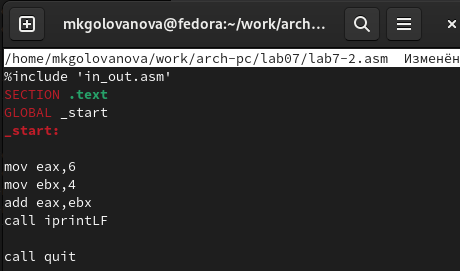


Рис. 9: Исправление текста файла lab7-2.asm

Я создала исполняемый файл и запустила его (рис. 10).

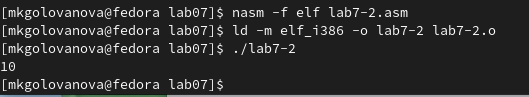


Рис. 10: Создание и запуск исполняемого исправленного файла lab7-2

В результате работы программы я получила число 10. Я заменила функцию iprintLF на iprint, создала исполняемый файл и запустила его (рис. 11, рис. 12). После вывода функция iprint в отличии от функции iprintLF не переводит строку.



Рис. 11: Замена функции iprintLF на iprint в тексте файла lab7-2.asm

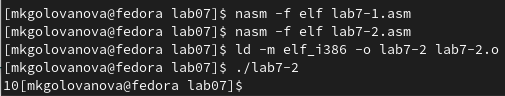


Рис. 12: Создание и запуск исполняемого исправленного файла lab7-2

## 3.2 Выполнение арифметических операций в NASM

В качестве примера выполнения арифметических операций в NASM я рассмотрела программу вычисления арифметического выражения f(x) = (5 \* 2 + 3)/3. Я создала файл lab7-3.asm в каталоге ~/work/arch-pc/lab07 (рис. 13).

Рис. 13: Создание файла lab7-3.asm

Рис. 13: Создание файла lab7-3.asm

Я внимательно изучила текст программы из листинга 7.3 и ввела в lab7-3.asm (рис. 14)

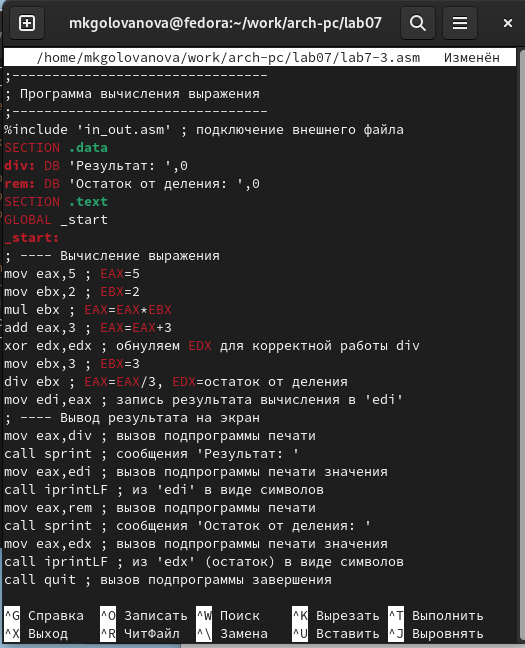


Рис. 14: Введение текста программы из листинга 7.3

Я создала исполняемый файл и запустила его (рис. 15).

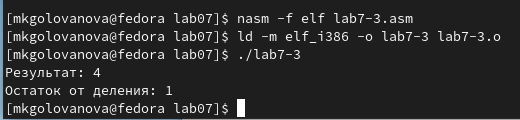


Рис. 15: Создание и запуск исполняемого файла lab7-3

Я изменила текст программы для вычисления выражения f(x) = (4 \* 6 + 2)/5, создала исполняемый файл и проверила его работу (рис. 16, рис. 17).

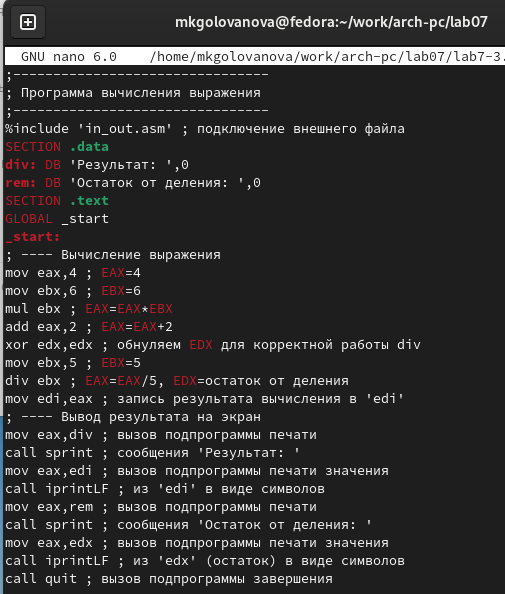


Рис. 16: Исправление текста файла lab7-3.asm

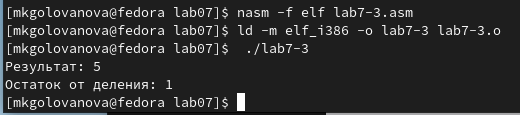


Рис. 17: Создание и запуск исполняемого изменённого файла lab7-3

В качестве другого примера я рассмотрела программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму: • вывести запрос на введение No студенческого билета; • вычислить номер варианта по формуле: (Sn mod 20) + 1, где Sn – номер студенческого билета (В данном случае a mod b – это остаток от деления a на b); • вывести на экран номер варианта. Я создала файл variant.asm в каталоге ~/work/arch-pc/lab07 (рис. 18).

Рис. 18: Создание файла variant.asm

Рис. 18: Создание файла variant.asm

Внимательно изучите текст программы из листинга 7.4 и введите в файл variant.asm (рис. 19).

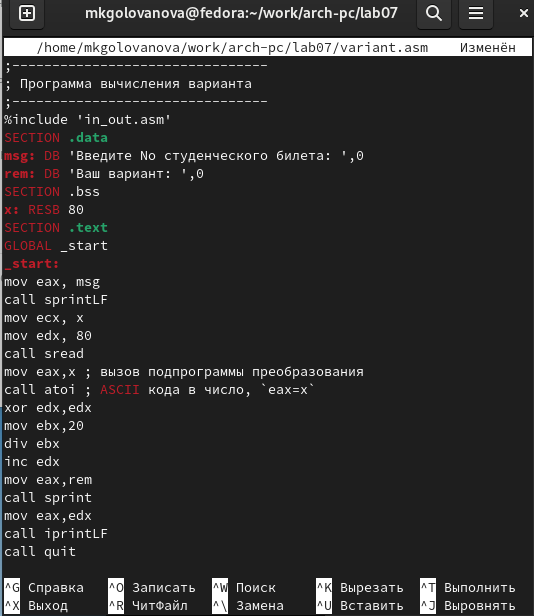


Рис. 19: Введение текста программы из листинга 7.4

Я создала исполняемый файл и запустила его (рис. 20).

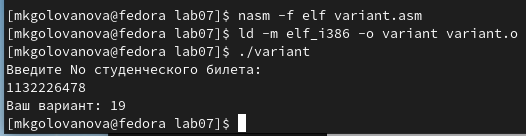


Рис. 20: Создание и запуск исполняемого файла variant

Я проверила результат работы программы, вычислив номер варианта аналитически. Ответы на вопросы: 1. За вывод на экран сообщения ‘Ваш вариант:’ отвечают строки: mov eax,rem call sprint 2. Инструкции nasm: mov ecx, x означает запись адреса переменной в ‘EAX’, mov edx, 80 - запись длины вводимого сообщения в ‘EBX’, call sread - вызов подпрограммы ввода сообщения. 3. Инструкция “call atoi” используется для преобразования ASCII кода в число. 4. За вычисления варианта отвечают строки: xor edx,edx mov ebx,20 div ebx inc edx 5. Остаток от деления при выполнении инструкции “div ebx” записывается в регистр edx. 6. Инструкция “inc edx” используется для прибавления к числу единицы. 7. За вывод на экран результата вычислений отвечают строки: mov eax,edx call iprintLF

# 4 Выполнение самостоятельной работы

Я создала файл var19.asm и написала в нём программу вычисления выражения y = f(x) так, чтобы она выводила выражение для вычисления, выводила запрос на ввод значения x, вычисляла заданное выражение в зависимости от введенного x, выводила результат вычислений (рис. 21, рис. 22). Вид функции f(x) я выбрала из таблицы 7.3 вариантов заданий в соответствии с номером, полученным мной при выполнении лабораторной работы (Вариант №19: ( 1/3 \* x + 5) \* 7, x1=3, x2=9)(рис. 23).

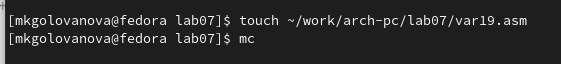


Рис. 21: Создание файла var19.asm

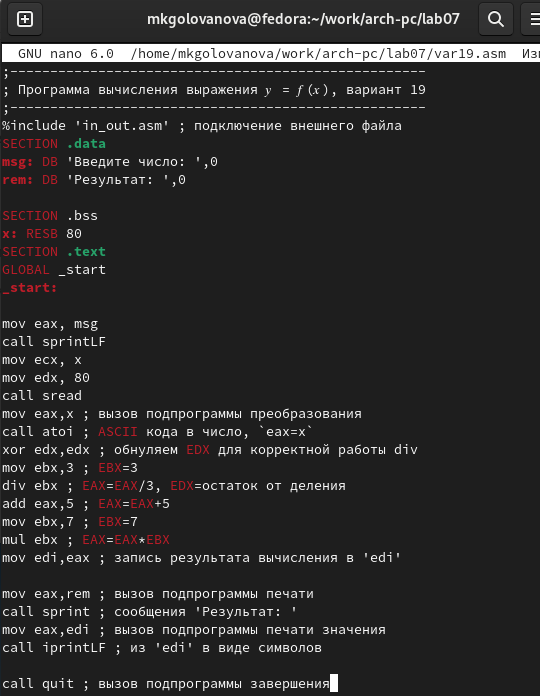


Рис. 22: Введение текста программы для функции из таблицы 7.3

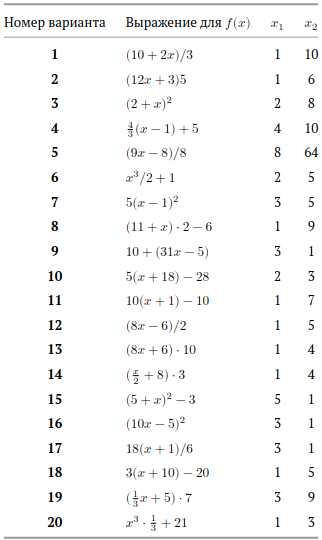


Рис. 23: Таблица 7.3

Я создала исполняемый файл и проверила его работу для значений x1 и x2 из таблицы 7.3 (рис. 24).

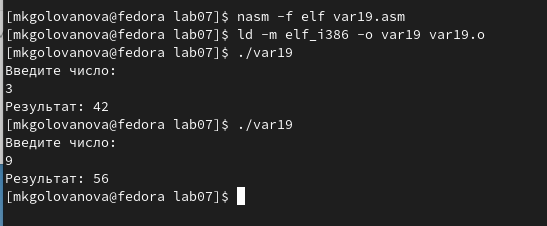


Рис. 24: Создание и проверка исполняемого файла var19

# 5 Выводы

Я освоила арифметические инструкций языка ассемблера NASM.