Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Мусатова Екатерина Викторовна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

**1**

Создайте каталог для программ лабораторной работы № 6, перехожув него и создаю файл lab6-1.asm (рис. [1](#fig:001)).

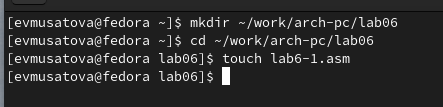


Figure 1: Создание необходимого каталога и файла

**2**

Ввожу в файл lab6-1.asm текст программы из листинга 6.1 (рис. [2](#fig:002)).

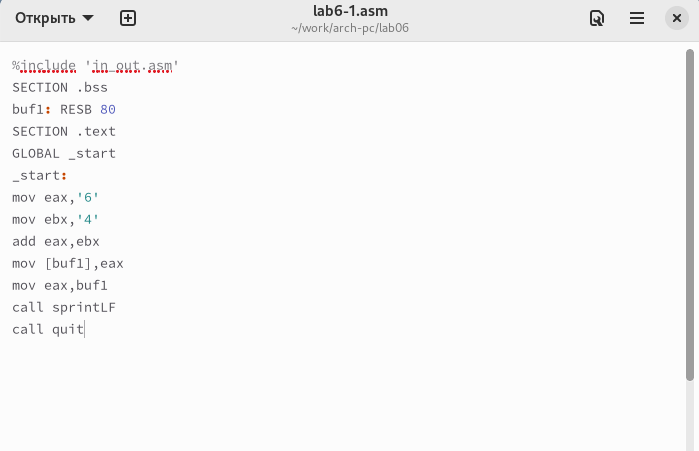


Figure 2: Ввод программы

Создаю исполняемы файл и запускаю его для проверки программы и вижу, что выводится символ j (рис. [3](#fig:003)).

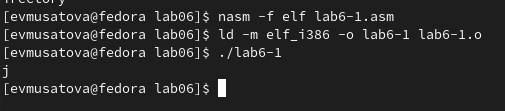


Figure 3: Проверка

**3**

Исправляю текст программы из листинга 6.1, заменяя символы “6” и “4” на цифры 6 и 4 (рис. [4](#fig:004)).

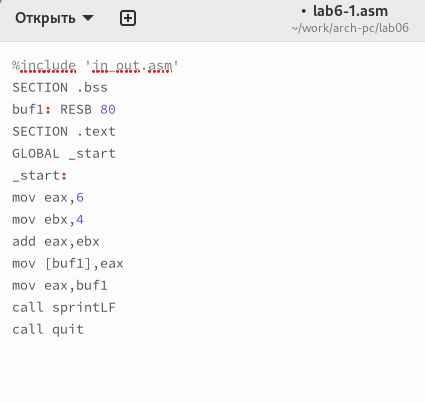


Figure 4: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. [5](#fig:005)). Теперь вывелся символ с кодом 10, это символ перевода строки, он не отображается при выводе на экран.

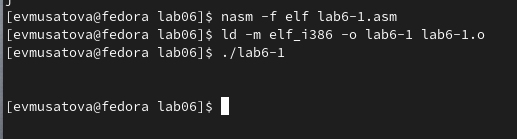


Figure 5: Проверка программы

**4**

Создаю файл lab6-2.asm в каталоге lab06 (рис. [6](#fig:006)).

Figure 6: Создание файла

Figure 6: Создание файла

Ввожу в созданный файл текст программы из листинга 6.2 (рис. [7](#fig:007)).



Figure 7: Программа вывода значения регистра eax

Создаю исполняемый файл и запускаю его (рис. [8](#fig:008)). Теперь выводится число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”.

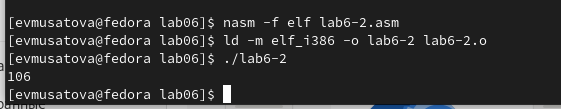


Figure 8: Проверка программы

**5**

Аналогично предыдущему примеру изменяю символы на числа (рис. [9](#fig:009)).

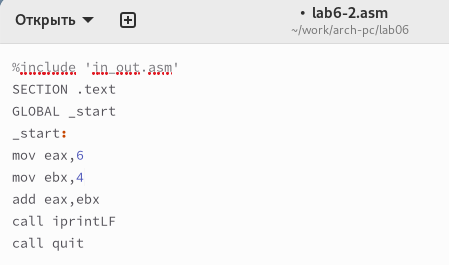


Figure 9: Изменение программы

Создаю исполняемый файл и запускаю его (рис. [10](#fig:010)). Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10.

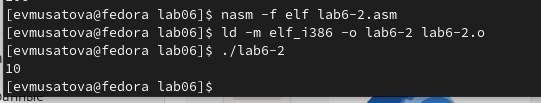


Figure 10: Проверка

Заменяю функцию iprintLF на iprint (рис. [11](#fig:011)).

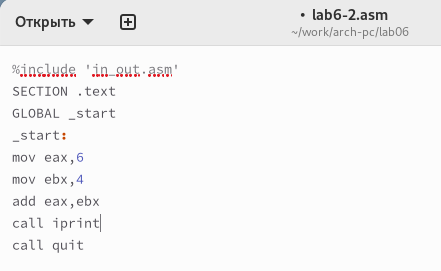


Figure 11: Замена функции

Создаю исполняемый файл и запускаю его (рис. [12](#fig:012)).Теперь функция iprint добавляет к выводу символ переноса строки, в отличие от iprintLF.

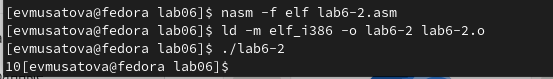


Figure 12: Проверка программы

**6**

Создаю файл lab6-3 (рис. [13](#fig:013)).

Figure 13: Создание нового файла

Figure 13: Создание нового файла

Ввожу в текст программы из листинга 6.3 в созданный файл. Затем создаю исполняемый файл и проверяю работу программы (рис. [14](#fig:014)).

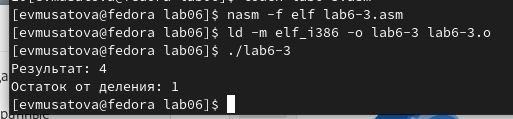


Figure 14: Проверка программы

Изменяю текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5 (рис. [15](#fig:015)).

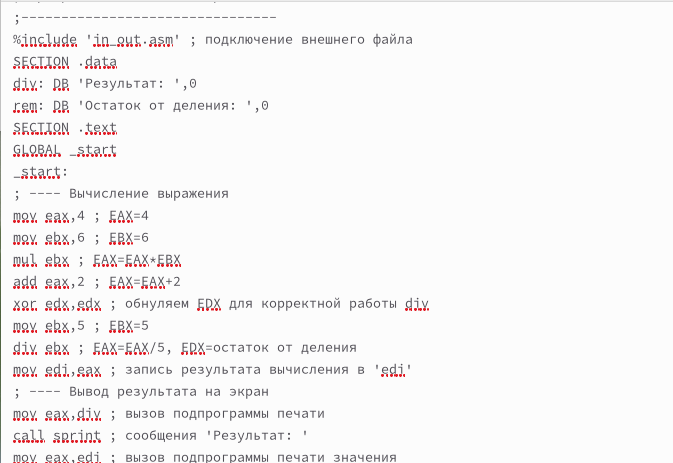


Figure 15: Изменение программы

Создаю исполняемый файл и проверяю его работу (рис. [16](#fig:016)).

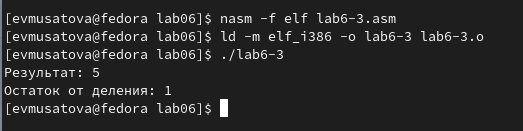


Figure 16: Проверка

**7**

Создаю файл variant.asm в каталоге lab06 (рис. [17](#fig:017)).

Figure 17: Создание нового файла

Figure 17: Создание нового файла

Ввожу текст программы из листинга 6.4 (рис. [18](#fig:018)).

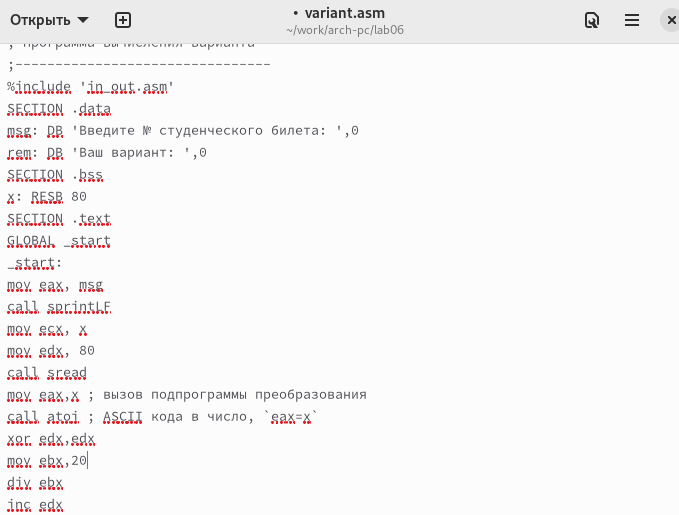


Figure 18: Создание программы

Создаю исполняемый файл и запускаю его, затем ввожу номер студенческого билета и узнаю номер своего варианта (рис. [19](#fig:019)).

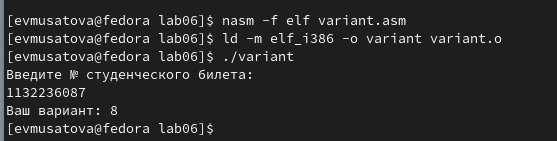


Figure 19: Проверка работы программы

# 3 Ответы на вопросы по программе

1. За вывод сообщения “Ваш вариант” отвечают строки кода:

mov eax,rem call sprint

1. Инструкция mov ecx, x используется, чтобы положить адрес вводимой строки 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.
2. Инструкция call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
3. 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

1. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
2. Инструкция inc edx увеличивает значение регистра edx на 1
3. За вывод на экран результатов вычислений отвечают строки:

mov eax,edx call iprintLF

# 4 Самостоятельная работа

Создаю файл lab6-4 (рис. [20](#fig:020)).

Figure 20: Создание файла

Figure 20: Создание файла

Открываю созданный файл, ввожу в него текст программы для вычисления значения выражения (11 + x) \* 2 - 6 (рис. [21](#fig:021)). Это выражение было под вариантом 8, который мне достался.

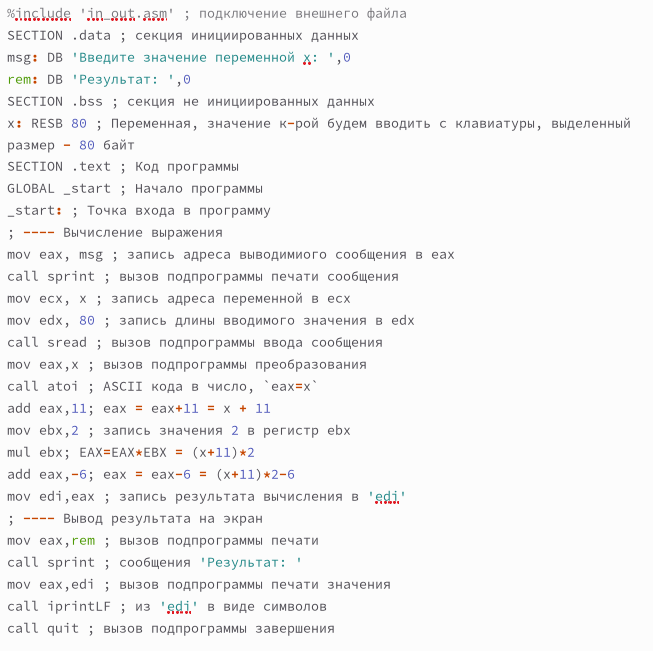


Figure 21: Создание программы

Создаю исполняемый файл (рис. [22](#fig:022)).

Figure 22: Создание исполняемого файла

Figure 22: Создание исполняемого файла

Запускаю исполняемый файл, ввожу значение - 1, при выводе получаю - 18 (рис. [23](#fig:023)).

Figure 23: Проверка работы программы

Figure 23: Проверка работы программы

Запускаю исполняемый файл, теперь ввожу значение - 9, при выводе получаю - 34 (рис. [24](#fig:024)).

Figure 24: Проверка 2

Figure 24: Проверка 2

# 5 Листинг 6.5. Программа для вычисления значения выражения (11 + x) \* 2 − 6.

%include ‘in\_out.asm’ ; подключение внешнего файла SECTION .data ; секция инициированных данных msg: DB ‘Введите значение переменной х:’,0 rem: DB ‘Результат:’,0 SECTION .bss ; секция не инициированных данных x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный размер - 80 байт SECTION .text ; Код программы GLOBAL \_start ; Начало программы \_start: ; Точка входа в программу ; —- Вычисление выражения mov eax, msg ; запись адреса выводимиого сообщения в eax call sprint ; вызов подпрограммы печати сообщения mov ecx, x ; запись адреса переменной в ecx mov edx, 80 ; запись длины вводимого значения в edx call sread ; вызов подпрограммы ввода сообщения mov eax,x ; вызов подпрограммы преобразования call atoi ; ASCII кода в число, eax=x add eax,11; eax = eax+11 = x + 11 mov ebx,2 ; запись значения 2 в регистр ebx mul ebx; EAX=EAX*EBX = (x+11)*2 add eax,-6; eax = eax-6 = (x+11)\*2-6 mov edi,eax ; запись результата вычисления в ‘edi’ ; —- Вывод результата на экран mov eax,rem ; вызов подпрограммы печати call sprint ; сообщения ‘Результат:’ mov eax,edi ; вызов подпрограммы печати значения call iprintLF ; из ‘edi’ в виде символов call quit ; вызов подпрограммы завершения

# 6 Выводы

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