Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Глущенко Евгений Игоревич

# 1 Цель работы

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

# 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Символьные и численные данные в NASM

С помощью утилиты mkdir создаю директорию, в которой буду работать (рис. 1). Перехожу в созданный каталог с помощью cd.

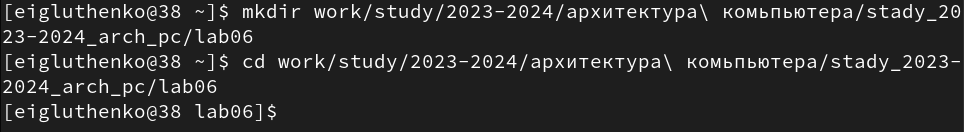


Рис. 1: Создание директории

С помощью touch создаю файл lab6-1.asm (рис. 2).

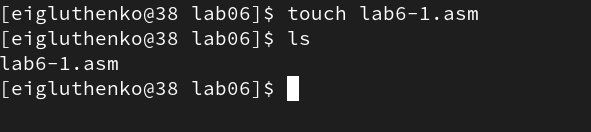


Рис. 2: Создание файла

Копирую в текущий каталог файл in\_out.asm (рис. 3).

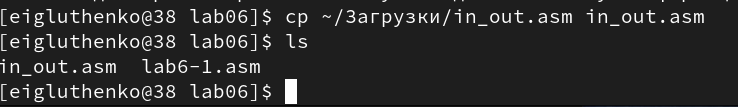


Рис. 3: Создание копии файла

Открываю созданный файл lab6-1.asm, вставляю в него программу вывода(рис. 4).



Рис. 4: Редактирование файла

Создаю исполняемый файл программы и запускаю его (рис. 5). Вывод программы: символ j, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.

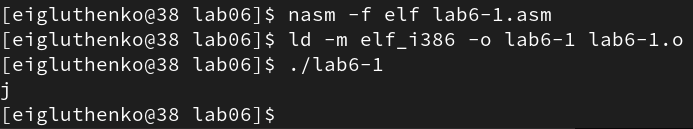


Рис. 5: Запуск исполняемого файла

Изменяю в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 6).

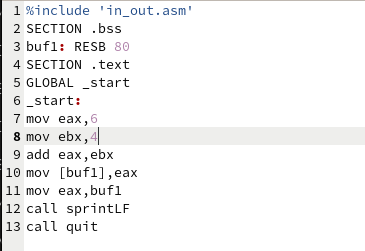


Рис. 6: Редактирование файла

Создаю новый исполняемый файл программы и запускаю его (рис. 7). Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

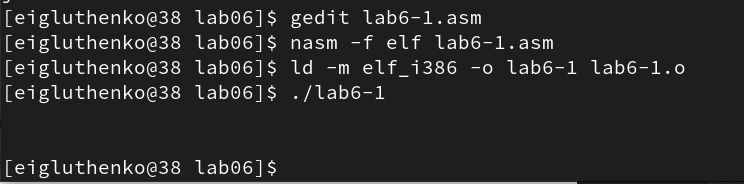


Рис. 7: Запуск исполняемого файла

Создаю новый файл lab6-2.asm с помощью touch (рис. 8).

C:\Users\jecka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана от 2023-11-15 11-58-42.png

Рис. 8: Создание файла

Ввожу в файл текст другой программы для вывода значения регистра eax (рис. 9).

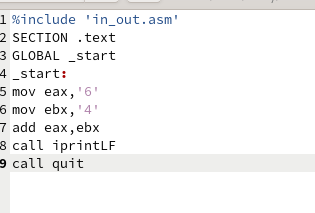


Рис. 9: Редактирование файла

Создаю и запускаю исполняемый файл lab6-2 (рис. 10). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4”.

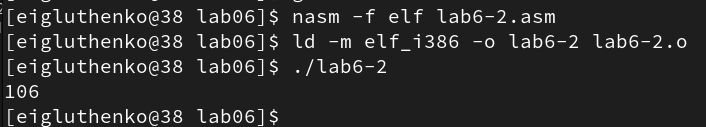


Рис. 10: Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 11).

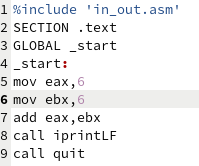


Рис. 11: Редактирование файла

Создаю и запускаю новый исполняемый файл (рис. 12).. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 12.

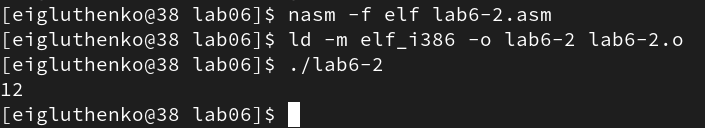


Рис. 12: Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис. 13).

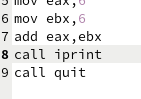


Рис. 13: Редактирование файла

Создаю и запускаю новый исполняемый файл (рис. 14). Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией iprintLF, а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF.

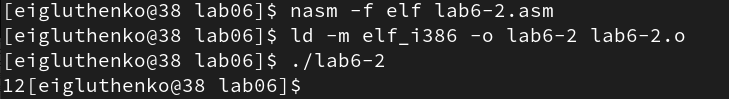


Рис. 14: Запуск исполняемого файла

## 3.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm с помощью утилиты touch (рис. 15).

C:\Users\jecka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана от 2023-11-15 12-04-02.png

Рис. 15: Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 16).

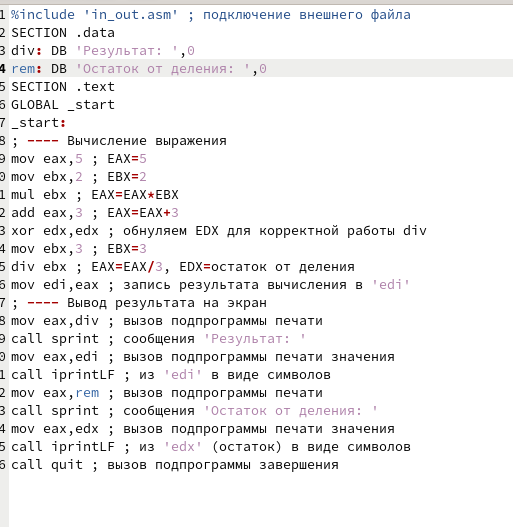


Рис. 16: Редактирование файла

Создаю исполняемый файл и запускаю его (рис. 17).

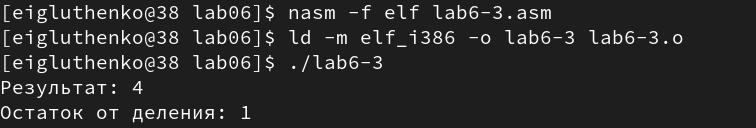


Рис. 17: Запуск исполняемого файла

Изменяю программу, чтобы она вычисляла значение f(x) = (4 \* 6 + 2)/5 (рис. 18).

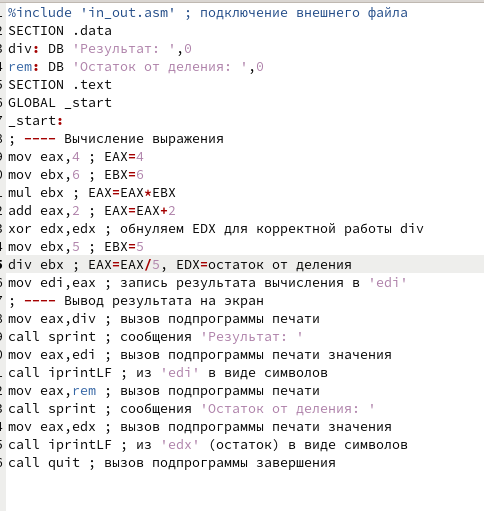


Рис. 18: Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 19). Как видим, все верно.

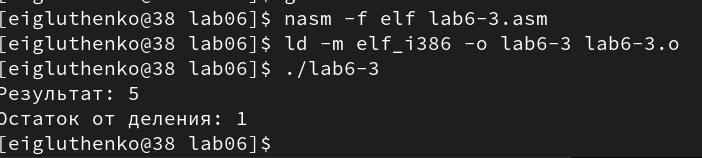


Рис. 19: Запуск исполняемого файла

Создаю файл variant.asm (рис. 20).

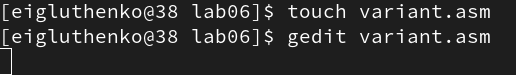


Рис. 20: Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 21).

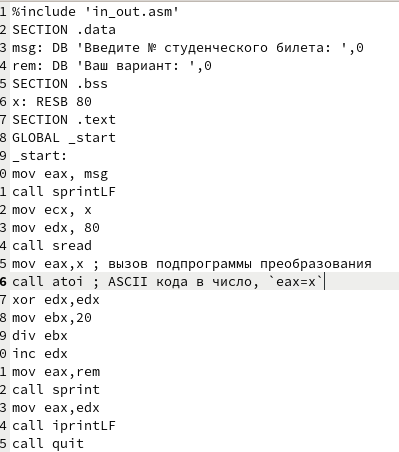


Рис. 21: Редактирование файла

Создаю и запускаю исполняемый файл (рис. 22). Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 11.

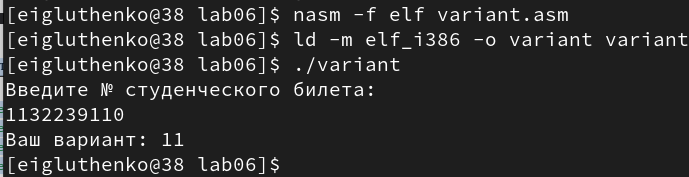


Рис. 22: Запуск исполняемого файла

### 3.2.1 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. 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

## 3.3 Выполнение заданий для самостоятельной работы

Создаю файл lab7-4.asm (рис. 23).

C:\Users\jecka\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана от 2023-11-15 12-13-51.png

Рис. 23: Создание файла

Открываю созданный файл для редактирования, ввожу в него текст программы для вычисления значения выражения (11 + x) \* 2 - 6 (рис. 24).

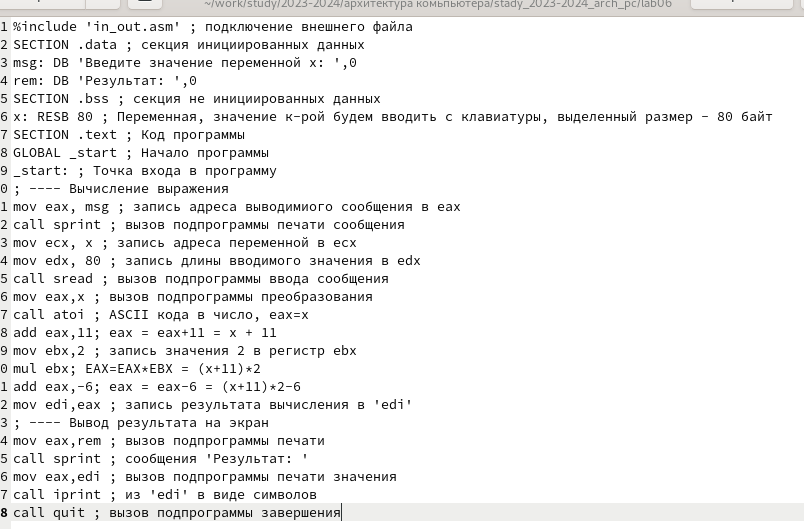


Рис. 24: Написание программы

Создаю и запускаю исполняемый файл (рис. 25). При вводе значения 3, вывод - 22.

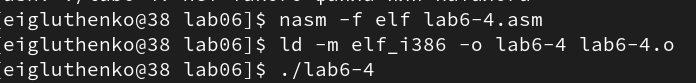


Рис. 25: Запуск исполняемого файла

Провожу еще один запуск исполняемого файла для проверки работы программы с другим значением на входе (рис. 26). Программа отработала верно.

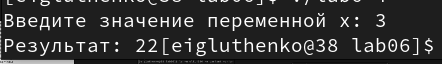


Рис. 26: Запуск исполняемого файла

**Программа для вычисления 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 iprint ; из 'edi' в виде символов  
call quit ; вызов подпрограммы завершения

# 4 Выводы

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