Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Игнатенкова В. Н

Содержание

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM

# 2 Выполнение лабораторной работы

Создала директорию lab07, переместилась в нее, создала файл lab7-1.asm, проверила, что файл был создан (рис. 1)

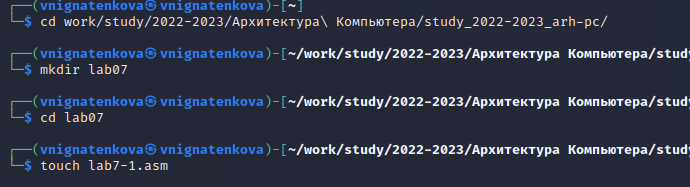


Рис. 1: Создание файла

Записала в файл lab7-1.asm программу (рис. 2).

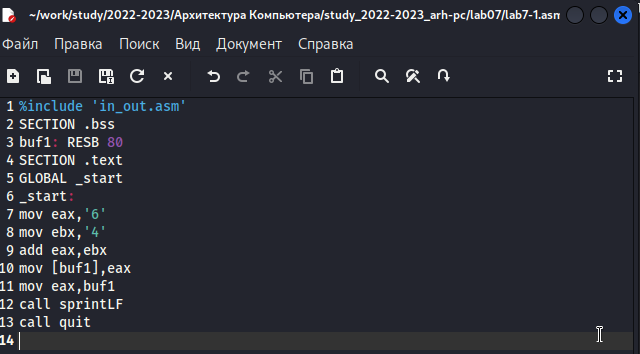


Рис. 2: Запись программы в файл

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

Рис. 3: Копирование файла

Рис. 3: Копирование файла

Создаю исполняемый файл программы и запускаю его (рис. 4). Программа вывела символ j, потому что он соответствует сумме двоичных кодов символов 4 и 6 по системе ASCII.

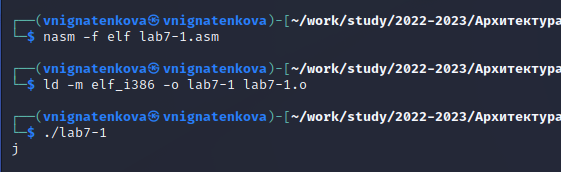


Рис. 4: Запуск исполняемого файла

Изменаю программу (рис. 5).

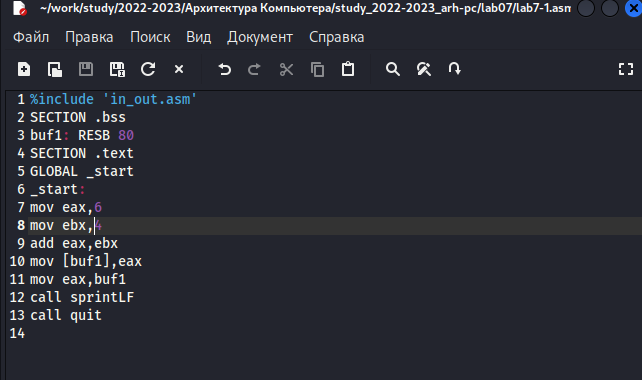


Рис. 5: Изменение программы

Создаю исполняемый файл программы и запускаю его (рис. 6). Программа вывела символ перевода строки, потому что он соответствует коду 10=6+4 по системе ASCII.

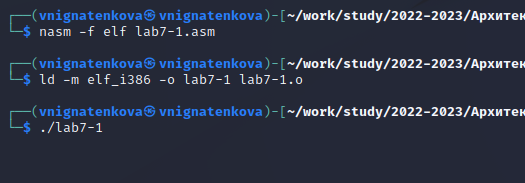


Рис. 6: Запуск исполняемого файла

Создаю файл lab7-2.asm (рис. 7).

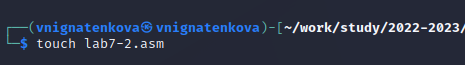


Рис. 7: Создание файла

Ввожу текст программы в созданный файл (рис. 8).

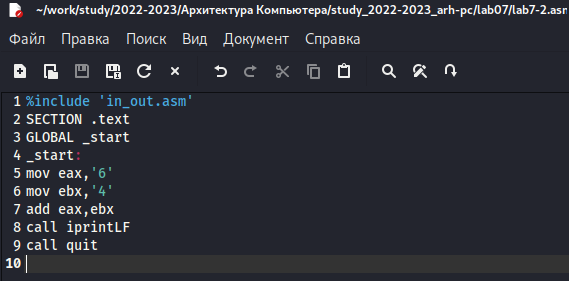


Рис. 8: Запись программы в файл

Создаю и запускаю исполняемый файл lab7-2 (рис. 9). Теперь программа выводит число, а не символ, но это число - сложение кодов символов “6” и “4”.

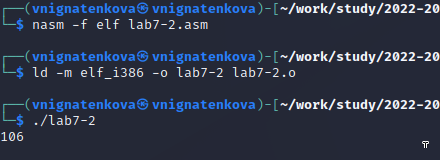


Рис. 9: Запуск исполняемого файла

Изменяю программу (рис. 10).

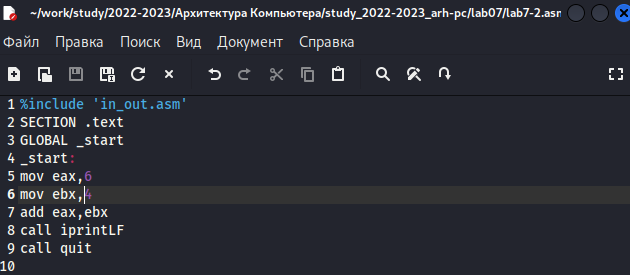


Рис. 10: Изменение программы

Запускаю новый исполняемый файл, теперь складываются именно цифры (рис. 11).

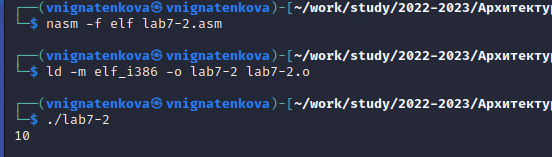


Рис. 11: Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис. 12).

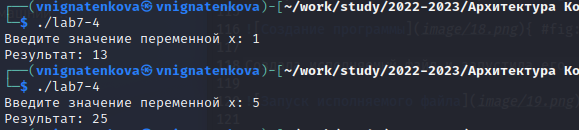


Рис. 12: Изменение программы

Создаю и запускаю новый исполняемый файл (рис. 13). 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.

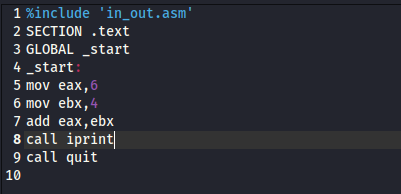


Рис. 13: Создание программы

Создаю файл lab7-3.asm и в нем записываю программу (рис. 14).

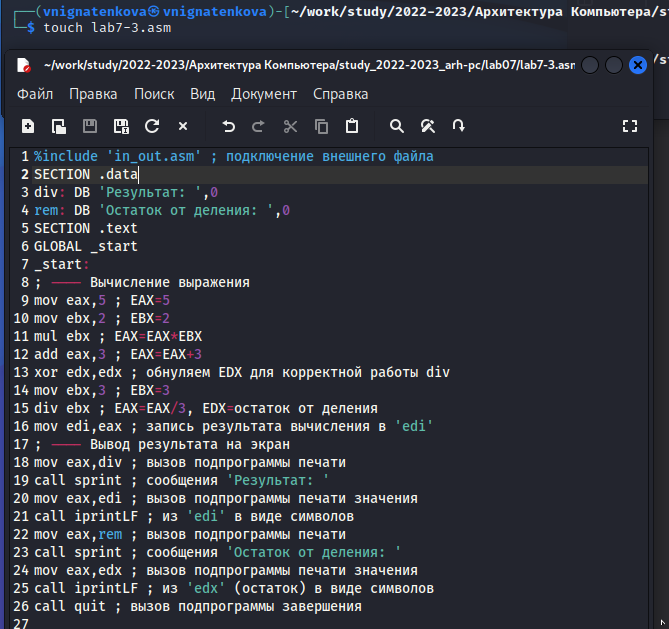


Рис. 14: Создание программы

Создаю исполняемый файл и запускаю его (рис. 15).

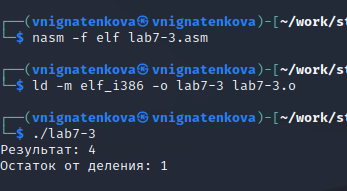


Рис. 15: Запуск исполняемого файла

Изменяю текст программы, чтобы посчитать значение выражения (4\*6+2)/5 (рис. 16).

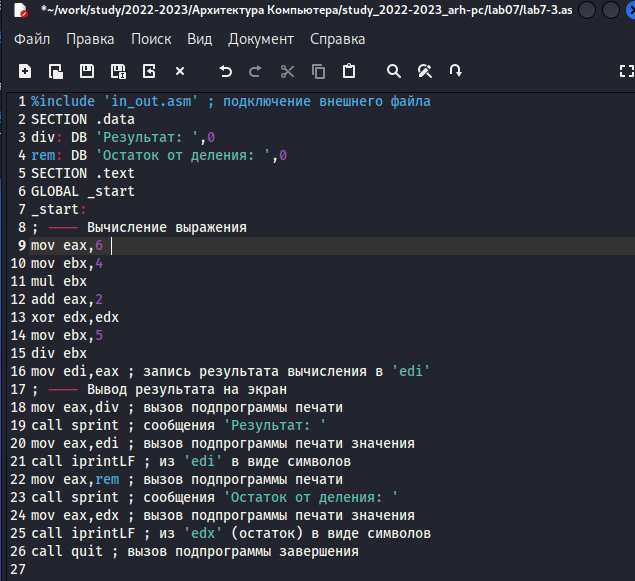


Рис. 16: Изменение программы

Создаю исполняемый файл, запускаю его, программа работает корректно (рис. 17).

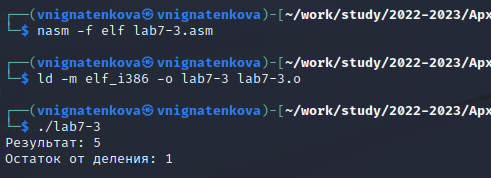


Рис. 17: Запуск исполняемого файла

Создаю файл variant.asm и записываю в него текст программы для вычисления варианта (рис. 18).

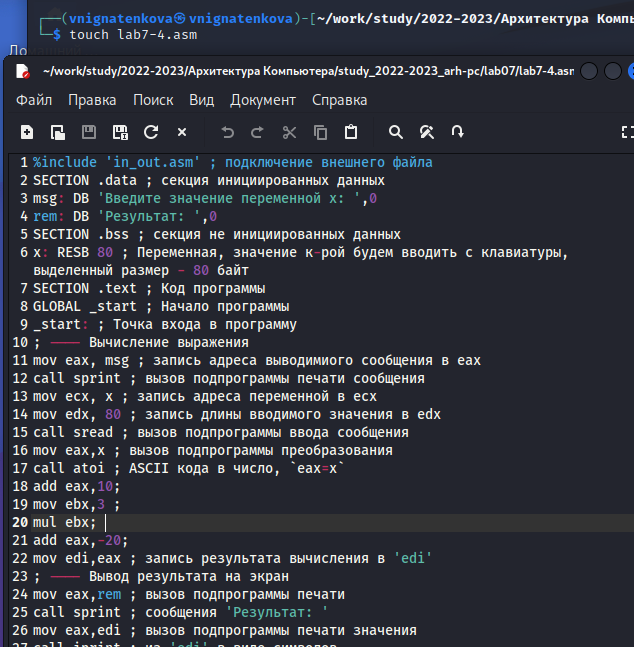


Рис. 18: Создание программы

Создаю исполняемый файл и запускаю его. Мой номер варианта для выполнения заданий для самостоятельной работы - 18 (рис. 19).

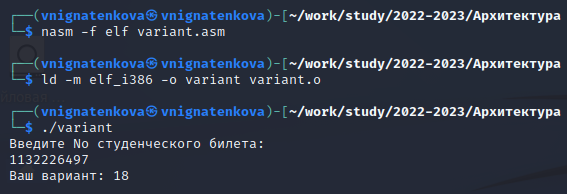


Рис. 19: Запуск исполняемого файла

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

mov eax,rem  
call sprint

1. Инструкция mov ecx, x используется, чтобы положить адрес вводимой стро- ки 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, кото- рая 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 Выполнение заданий для самостоятельной работы

Создала файл lab7-4.asm, в нем написала программу для вычисления выражения из варианта 18: 3\*(х+10)-20 (рис. 20).

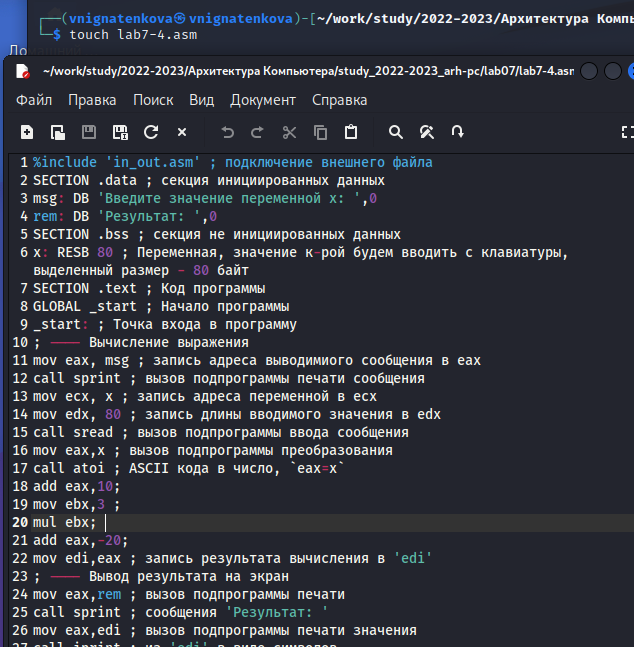


Рис. 20: Создание программы

Создала исполняемый файл и запустила его. Программа работает корректно(рис. 21).

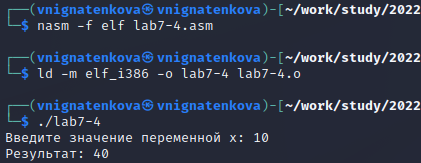


Рис. 21: Запуск исполняемого файла

Проверила работу программы для значений х1 и х2 (рис. 22).

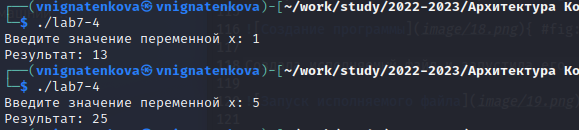


Рис. 22: Проверка работы программы

**Код написанной программы:**

%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,10;   
mov ebx,3 ;   
mul ebx;   
add eax,-20;   
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

В ходе выполнения лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.