Лабораторная работа №6

Арифметические операции в NASM.

Бочаров Андрей

Содержание

# 1 Цель работы

Целью работы является изучение арифметические инструкции языка ассемблера NASM # Выполнение лабораторной работы создал и першел в каталог для 6 лабораторной работы и командой touch сделал файл lab6-1.asm (рис. 1).

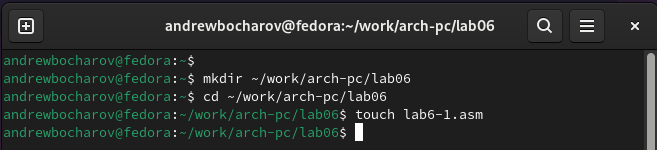


Рис. 1: Каталог lab06

Переписал код из листинга 6.1 (рис. 2).

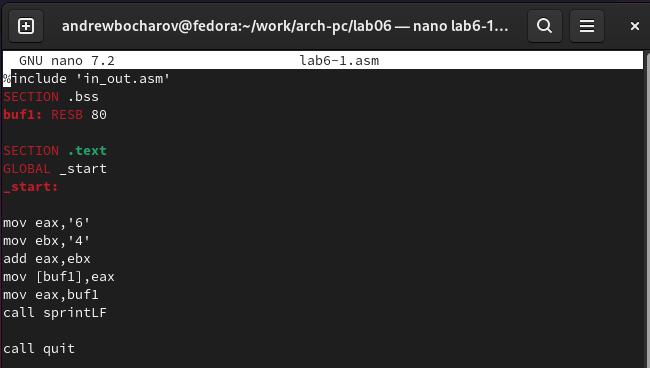


Рис. 2: Листинг кода

Листинг кода 6.1:

%include 'in\_out.asm'  
SECTION .bss  
buf1: RESB 80  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax, '6'  
mov ebx, '4'  
add eax,ebx  
mov [buf1],eax  
mov eax,buf1  
call sprintLF  
  
call quit

Создал исполняемый файл и запустил его. (рис. 3).

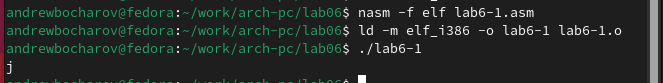


Рис. 3: Работает

Заменил строки с числами в ковычках, на числа без ковычек (рис. 4).

Рис. 4: Обновленный код

Рис. 4: Обновленный код

Создал исполняемый файл и запустил его. На экран выведен символ переноса строки, он соответствует коду 10 кодировки ASCII (рис. 5).

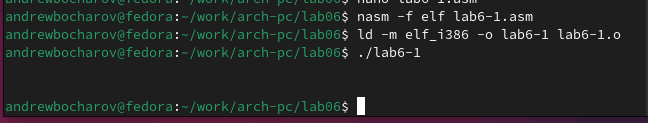


Рис. 5: Результат работы файла

Создал файл lab6-1.asm и переписал в него листинг кода 6.2 (рис. 6).



Рис. 6: Листинг кода

Листинг кода 6.2:

%include 'in\_out.asm'  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax, 6  
mov ebx, 4  
add eax,ebx  
call iprintLF  
  
call quit

Создал исполняемый файл и запустил его. (рис. 7).

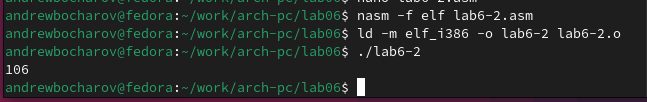


Рис. 7: Работает

Вместо символов, записал в регистры числа. Создал исполняемый файл и запустил его. (рис. 8).



Рис. 8: Обновленные строчки кода

Создал исполняемый файл и запустил его. Результат работы число 10.(рис. 9).

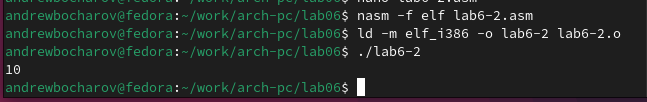


Рис. 9: Результат работы

Заменил команду iprintLF на iprint. (рис. 10).

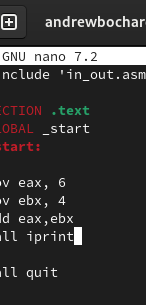


Рис. 10: Обновленные строчки кода

Запустил исполняемый файл, результат вывелся без переноса строки после себя. (рис. 11).

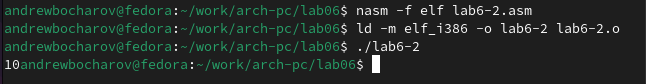


Рис. 11: Результат работы

Командой touch сделал файл lab6-3.asm (рис. 12).

Рис. 12: Создание файла

Рис. 12: Создание файла

Переписал код из листинга 6.3 (рис. 13).

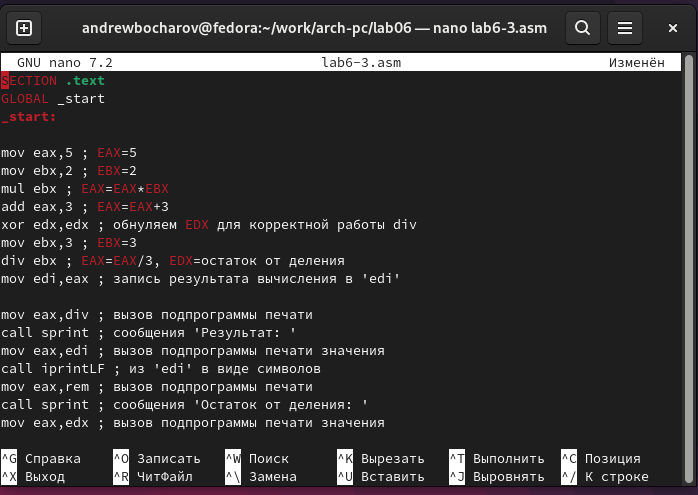


Рис. 13: Копирование файла

Листинг кода 6.3:

%include 'in\_out.asm' ; подключение внешнего файла  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax,5 ; EAX=5  
mov ebx,2 ; EBX=2  
mul ebx ; EAX=EAX\*EBX  
add eax,3 ; EAX=EAX+3  
xor edx,edx ; обнуляем EDX для корректной работы div  
mov ebx, 3 ; EBX=3  
div ebx ; EAX=EAX/3, EDX=остаток от деления  
  
mov edi,eax ; запись результата вычисления в 'edi'  
  
mov eax,div ; вызов подпрограммы печати  
call sprint ; сообщения 'Результат: '  
mov eax,edi ; вызов подпрограммы печати значения  
call iprintLF ; из 'edi' в виде символов  
  
mov eax,rem ; вызов подпрограммы печати  
call sprint ; сообщения 'Остаток от деления: '  
mov eax,edx ; вызов подпрограммы печати значения  
call iprintLF ; из 'edx' (остаток) в виде символов  
  
call quit ; вызов подпрограммы завершения

Создал исполняемый файл и запустил его. Результат совпадает с ожидаемым. (рис. 14).

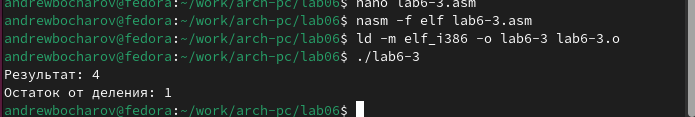


Рис. 14: Код рабоает

Переписал код в файле lab6-3.asm для вычисления выражения f(𝑥) = (4 ∗ 6 + 2)/5. (рис. 15).

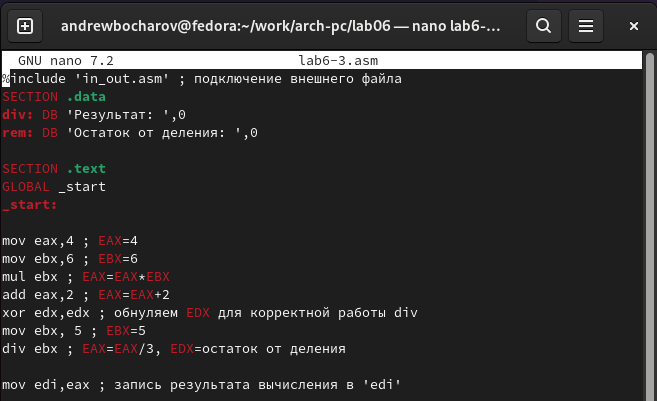


Рис. 15: Обновленый код

Обновленный листинг кода:

%include 'in\_out.asm' ; подключение внешнего файла  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax,4 ; EAX=4  
mov ebx,6 ; EBX=6  
mul ebx ; EAX=EAX\*EBX  
add eax,2 ; EAX=EAX+2  
xor edx,edx ; обнуляем EDX для корректной работы div  
mov ebx, 5 ; EBX=5  
div ebx ; EAX=EAX/5, EDX=остаток от деления  
  
mov edi,eax ; запись результата вычисления в 'edi'  
  
mov eax,div ; вызов подпрограммы печати  
call sprint ; сообщения 'Результат: '  
mov eax,edi ; вызов подпрограммы печати значения  
call iprintLF ; из 'edi' в виде символов  
  
mov eax,rem ; вызов подпрограммы печати  
call sprint ; сообщения 'Остаток от деления: '  
mov eax,edx ; вызов подпрограммы печати значения  
call iprintLF ; из 'edx' (остаток) в виде символов  
  
call quit ; вызов подпрограммы завершения

Создал исполняемый файл и запустил его. Результат совпадает с ожидаемым. (рис. 16).

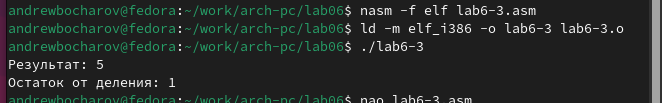


Рис. 16: Результат выполнения

Создал файл variant.asm и переписал в него листинг кода 6.4 (рис. 17).

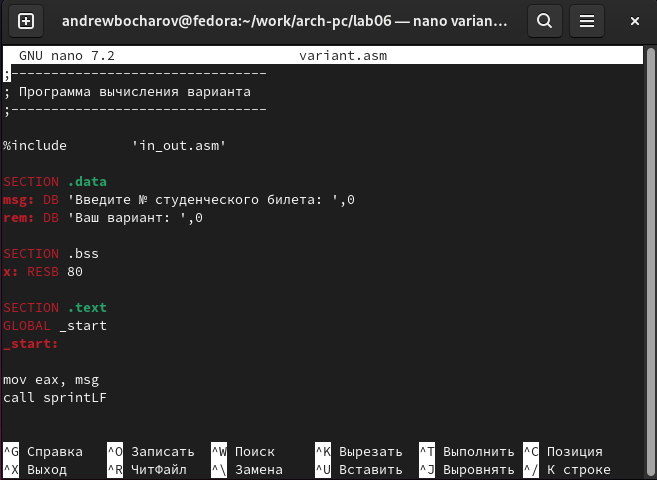


Рис. 17: Работает файла

Листинг 6.4:

;--------------------------------  
; Программа вычисления варианта  
;--------------------------------  
  
%include 'in\_out.asm'  
  
SECTION .data  
msg: DB 'Введите № студенческого билета: ',0  
rem: DB 'Ваш вариант: ',0  
  
SECTION .bss  
x: RESB 80  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax, msg  
call sprintLF  
  
mov ecx, x  
mov edx, 80  
call sread  
  
mov eax,x ; вызов подпрограммы преобразования  
call atoi ; ASCII кода в число, `eax=x`  
xor edx,edx  
mov ebx,20  
div ebx  
inc edx  
  
mov eax,rem  
call sprint  
mov eax,edx  
call iprintLF  
  
call quit

Создал исполняемый файл и запустил его, ввел номер своего студенческого билета, в ответ выведен номер моего варианта. (рис. 18).

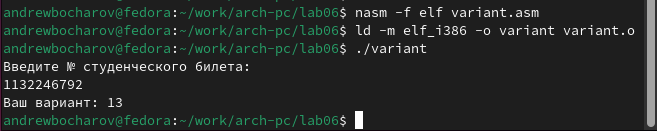


Рис. 18: Номер варианта

Ответы на вопросы: 1. За вывод на экран сообщения ‘Ваш вариант:’ отвечают строчки mov eax,rem call sprint 2. Инструкции mov ecx, x mov edx, 80 call sread отвечают за сохранение вввода в консоль в х

3 Инструкция “call atoi” переводит строку из консоли в число

4 За вычисления варианта отвечают строчки mov ebx,20 div ebx inc edx 5 Остаток от деления при выполнении инструкции div ebx записывается в регистр edx

6 Инструкция inc edx увеличивает значения числа в регистре на 1

7 За вывод на экран результата вычислений отвечают строки mov eax,edx call iprintLF

# 2 Выполнение самостоятельной работы

Создал новый файл и написал в него код отвечающий за вычисление формулы соответсвующей моему варианту. (рис. 19).

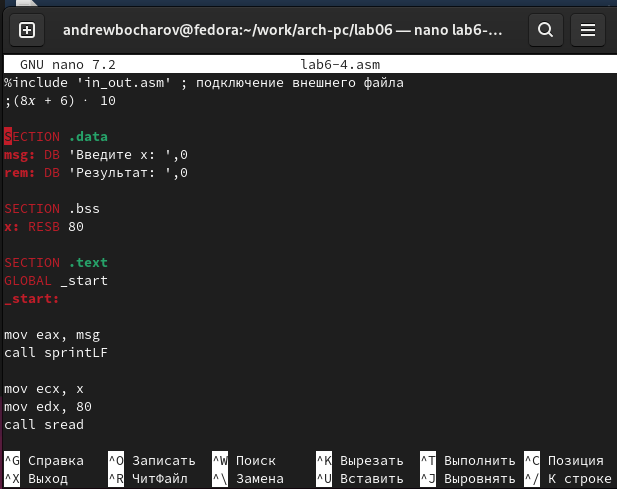


Рис. 19: Листинг кода

Листинг кода:

%include 'in\_out.asm' ; подключение внешнего файла  
;(8𝑥 + 6) ⋅ 10  
  
SECTION .data  
msg: DB 'Введите x: ',0  
rem: DB 'Результат: ',0  
  
SECTION .bss  
x: RESB 80  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
mov eax, msg  
call sprintLF  
  
mov ecx, x  
mov edx, 80  
call sread  
  
mov eax,x ; вызов подпрограммы преобразования  
call atoi ; ASCII кода в число, `eax=x`  
  
mov ebx,8 ; EBX=8  
mul ebx; EAX=EAX\*EBX  
  
add eax, 6 ; EAX=EAX+6  
mov ebx,10 ; EBX=10  
mul ebx ; EAX=EAX\*EBX, EBX=Произвидение  
mov edx, eax  
  
mov eax, rem ; вызов подпрограммы печати  
call sprint ; сообщения 'Результат: '  
mov eax, edx ; вызов подпрограммы печати значения  
call iprintLF ; из 'edi' в виде символов  
  
call quit ; вызов подпрограммы завершения

Сделал трансляцию, компоновку и запустил и проверил код. Код выводит правльные ответы, полученные аналитическим методом. (рис. 20).

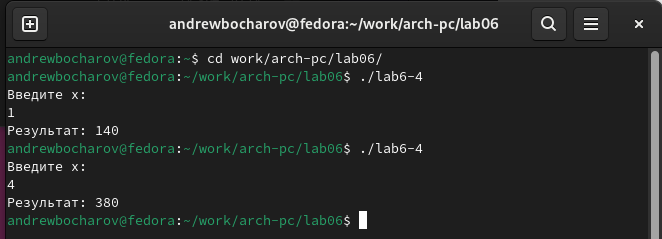


Рис. 20: Результат выполнения

Создал копию файла lab5-2.asm и внес в него изменения, что бы после ввода данных, они выводилилсь на экран с использованием команд из in\_out.asm. Листинг кода:

# 3 Выводы

Выполнив данную лабараторную работу, я oсвоил арифметические инструкции языка ассемблера NASM