Отчет по лабораторной работа №7

Группа НПИбд-02-22

Стариков Данила Андреевич

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Основная часть

## 2.1 Выполнение лабораторной работы

Для выполнения лабораторной работы создали каталог ~/work/study/arch-pc/lab07. В нем создали файл lab7-1.asm и ввели текст из Листинга 1. Также положили в каталог файл in\_out.asm, использованный в лабораторной работе №6.

Листинг 1: Программа для вывода значения регистра eax

%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

Создали исполняемый файл и запустили его (Рис. 1). На выводе получили символ ‘j’, так как в двоичном представлении код символа ‘6’ - 00110110 (54 в десятичном), а код символа ‘4’ - 00110100 (52). Команда add eax, ebx записала в регистр eax сумму кодов - 01101010 (106), что соответствует символу ‘j’ в ASCII.

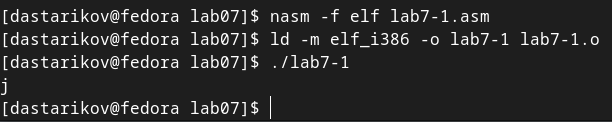


Рис. 1: Результат запуска файла lab7-1.

Заменили текст программы и вместо символов записали в регистры числа:

mov eax, '6'  
mov ebx, '4'

на строки

mov eax, 6  
mov ebx, 4

Создали исполняемый файл и запустили (Рис. 2). В результате выполнения на экран выводится число с кодом 10, что соответствует символу переноса строки.

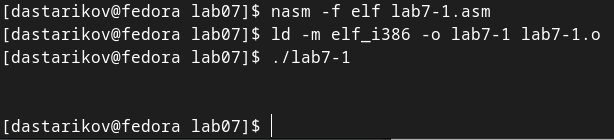


Рис. 2: Результат запуска файла lab7-1 после изменения текста программы.

Чтобы напечатать числа вместо ASCII символов изменили текст программы, создали файл lab7-2.asm, воспользовавшись подпрограммамми из файла in\_out.asm (Листинг 2).

Листинг 2: Программа вывода значения регистра eax

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

Создали исполняемый файл и запустили (Рис. 3). В результате выполнения на экран выводится число 106, так как в регистры были записаны символы ‘6’ и ‘4’. Подпрограмма iprintLF позволила вывести число, а не символ, кодом которого является это число.

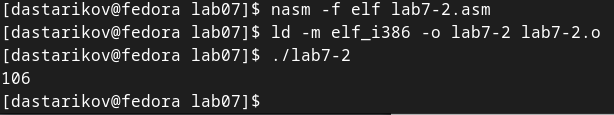


Рис. 3: Результат запуска файла lab7-2

Заменили текст программы и вместо символов записали в регистры числа:

mov eax, '6'  
mov ebx, '4'

на строки

mov eax, 6  
mov ebx, 4

Создали исполняемый файл и запустили (Рис. 4). В результате выполнения на экран выводится число 10.

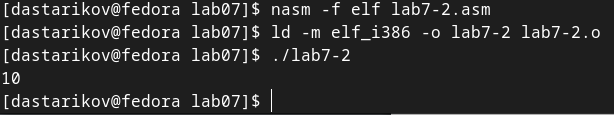


Рис. 4: Результат запуска файла lab7-2 после изменения текста программы.

Функция iprint, в отличии от iprintLF, не печатает символ переноса строки (Рис. 5).

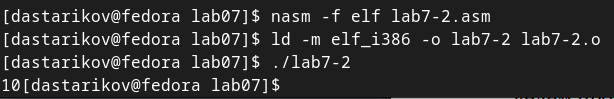


Рис. 5: Результат запуска файла lab7-2 с функцией iprint

Для примера выполнения арифметических операций в NASM создали программу вычисления арифметического выражения . Создали файл lab7-3.asm и ввели текст из Листинга 3.

Листинг 3: Программа вычисления выражения f(x) = (5\*2+3)/3

;--------------------------------  
; Программа вычисления выражения  
;--------------------------------  
%include 'in\_out.asm' ; подключение внешнего файла  
  
SECTION .data  
  
div: DB 'Результат: ', 0  
rem: DB 'Остаток от деления: ', 0  
  
SECTION .text  
GLOBAL \_start  
  
\_start:  
; ---- Вычисление выражения  
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'  
  
; ---- Вывод результата на экран  
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

Cоздали исполняемый файл и запустили (Рис. 6). Заметим изменили программу в соответствии с Листингом 4, чтобы она вычисляла выражение (Рис. 7).

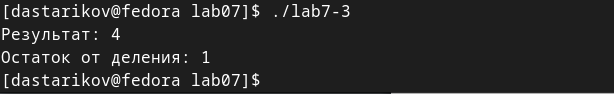


Рис. 6: Результат запуска файла lab7-2 после изменения текста программы.

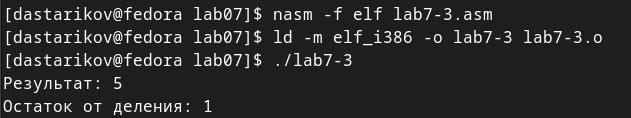


Рис. 7: Результат запуска файла lab7-2 после изменения текста программы.

Листинг 4: Программа вычисления выражения f(x) = (4\*6+2)/5

;--------------------------------  
; Программа вычисления выражения  
;--------------------------------  
%include 'in\_out.asm' ; подключение внешнего файла  
  
SECTION .data  
div: DB 'Результат: ', 0  
rem: DB 'Остаток от деления: ', 0  
SECTION .text  
GLOBAL \_start  
  
\_start:  
; ---- Вычисление выражения  
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'  
  
; ---- Вывод результата на экран  
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

В качестве другого примера рассмотрели задачу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

* вывести запрос на введение № студенческого билета
* вычислить номер варианта по формуле: , где – номер студенческого билета (В данном случае mod – это остаток от деления на ).
* вывести на экран номер варианта.

Номер студенческого билета, т. е. число, надо которым будут проводиться арифметические вычисления, вводится с клавиатуры, необходимо преобразовать введенные символы в число, так как ввод с клавиатуры осуществляется в символьном виде. В файле in\_out.asm для этого есть функция atoi.

Создали файл variant.asm и ввели текст из Листинга 5. При выполнении программы ввели номер студенческого билета (1132226531) и получили Вариант 12 (Рис. 8), что соответуствует подсчетам в уме.

Листинг 5: Программа вычисления варианта задания по номеру студенческого билета

;--------------------------------  
; 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

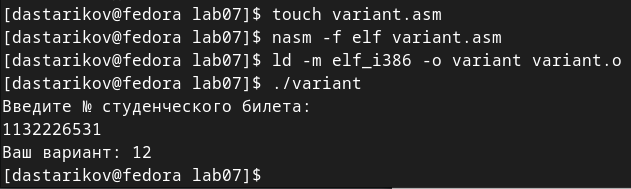


Рис. 8: Результат запуска файла lab7-2 после изменения текста программы.

Ответы на вопросы:

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

mov eax,rem  
call sprint

1. Эти строки используются для чтения ввода с клавиатуры. В регистр ECX передает адрес буфера x, а в EDX - длина вводимой строки.
2. Инструкция “call atoi” используется для вызова подпрограммы atoi, преобразующей символы в числа.
3. За вычисление варианты отвечают строки:

xor edx,edx  
mov ebx,20  
div ebx  
inc edx

1. Остаток от деления записывается в регистр EDX.
2. Инструкция увеличивает значение регистра EDX на 1.
3. За вывод на экран результата вычислений отвечают строки:

mov eax,edx  
call iprintLF

## 2.2 Выполнение заданий для самостоятельной работы.

Варианту 12 соответствует выражение . Для проверки корректности работы программы ввели 2 числа . Программа lav7-var12 принимает ввод с клавиатура в качестве значения x и выводит резульата вычисления выражения , под делением принимается целочисленное деление с отбрасываем остатка (Рис. 9). Текст файла lab7-var12.asm соответствует Листингу 6.

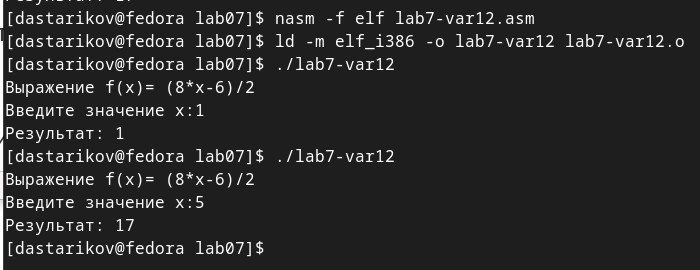


Рис. 9: Результат работы программы (Вариант 12)

Листинг 6: Программа вычисления выражения f(x) = (8x-6)/2

%include "in\_out.asm"  
  
SECTION .data  
 msg: DB "Выражение f(x)= (8\*x-6)/2", 10, "Введите значение x:", 0  
 div: DB 'Результат: ', 0  
  
SECTION .bss  
 x: RESB 80 ; буфер размером 80 байт  
  
SECTION .text  
 GLOBAL \_start  
  
 \_start:  
;Вывод строки с выражением f(x)  
 mov eax, msg  
 call sprint  
  
;Ввод значения x  
 mov ecx, x  
 mov edx, 80  
 call sread  
  
;Преобразование ввода в числовой вид  
 mov eax, x  
 call atoi; EAX=x как число  
  
;Вычисление выражения  
 mov ebx, 8  
 mul ebx; EAX=EAX\*EBX  
 sub eax, 6; EAX= EAX-6  
   
 xor edx,edx; обнуляем значение регистра EDX для корректной работы команды div  
 mov ebx, 2  
 div ebx; EAX=EAX/EBX - частное, EDX= EAX % EBX - остаток от деления  
 mov [x], eax; Сохраняю результат вычислений в адрес x  
  
;Вывод результата вычислений  
 mov eax, div  
 call sprint  
 mov eax, [x]  
 call iprintLF  
   
 call quit

Созданные файлы \*.asm скопировали в каталог ~/work/study/2022-2023/“Архитектура компьютера”/archpc/labs/lab07/ и загрузили на Github.

# 3 Выводы

В рамках лабораторной работы получили практические навыки использования арифметических инструкций языка ассемблера NASM.