Отчёт по лабораторной работе №6. Арифметические операции в NASM.

Арихитектура вычеслительных систем

Гандич Дарья Владимировна. НБИбд-02-22.

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

2 Задание

1. Написать программу вычисления выражения ☒ = ☒(☒). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения ☒, вычислять заданное выражение в зависимости от введенного ☒, выводить результат вычислений. Вид функции ☒(☒) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений ☒1 и ☒2 из 6.3.

3 Выполнение лабораторной работы

1. Создаем файл lab6-1.asm в каталоге лаборотрной работы и переносим текст из листинга 7.1.

```
dvgandich@dk3n52 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab06
dvgandich@dk3n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch lab6-1.asm
dvgandich@dk3n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
```

```
%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
```

2. Создаем исполняемый файл и запускам его, получаем символ j.

```
dvgandich@dk3n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
dvgandich@dk3n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
dvgandich@dk3n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1
j
```

Рис. 3.1: Исполняемый файл

3. Изменим текст листинга, убрав кавычки у цифр, снова запустим исполнемый файл, получим пустое поле.

```
%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
```

```
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1


dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

4. Создаем файл lab6-2.asm, переписываем текст листинга 7.2, запускаем, далее меняем листинг,убрав кавычки. В первом случае строка выводит 106, во втором - 10.

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF

call quit
```

```
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
106
```

```
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10
```

5. Теперь заменим iprintLF на iprint в листинге и сделаем вывод, что LF отвечает за перенос строки.

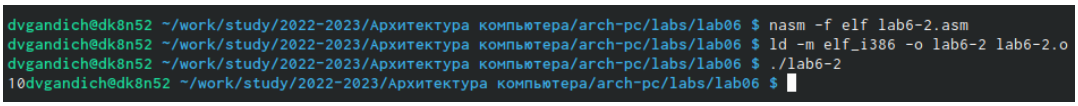


Рис. 3.2: Замена подпрограммы

6. Создаем файл lab6-3.asm, переписываем текст листинга 7.3., запускаем исполняемый файл.

```
;---------------------------------
; 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
```

```
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-3.asm
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-3
bash: ./lab6-3: Нет такого файла или каталога
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

7. Заменим в тексте листинга числа для вычисления выражения ⊠(⊠) = (4 ⊠ 6 +

2)/5.

```
...ci.pfu.edu.ru/home/d/v/dvgandich/work/study/2022-202
;-----------------------------
; 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=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX

add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; 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
```

```
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-3.asm
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
dvgandich@dk8n52 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

8. Создаем файл variant.asm, переписываем текст листинга 7.4, запускаем

исполняемый файл, узнаем свой вариант.

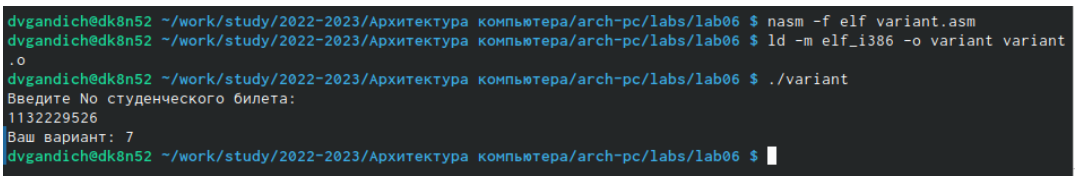


Рис. 3.3: Запуск файла 5

Вопросы: 1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

Ответ: mov eax,rem call sprint

2. Для чего используется следующие инструкции? nasm mov ecx, x mov edx, 80 call sread

Ответ: mov ecx, x - присвоение значения x переменной ecx mov edx, 80 - присвоение значения 80 переменной edx nasm - переход к языку ассемблера call sread - для считывания в переменную какого то числа

3. Для чего используется инструкция “call atoi”?

Ответ: конвертирует строку в величину типа int

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

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

5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

Ответ: в регистр dx

6. Для чего используется инструкция “inc edx”?

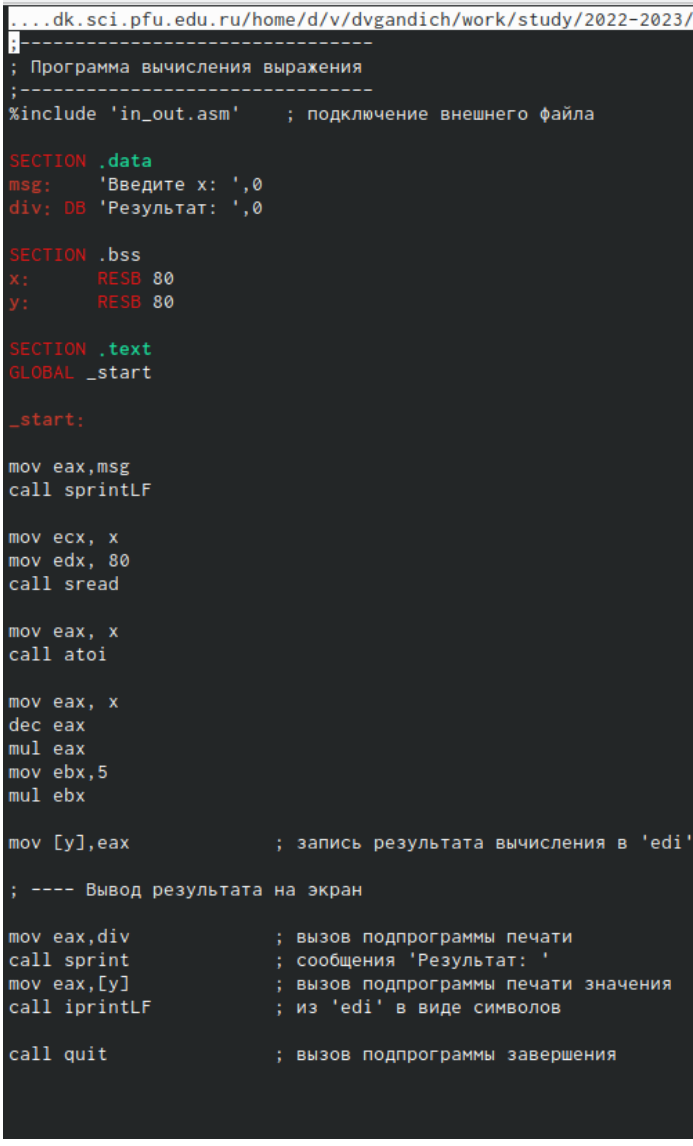
Ответ: это инкремент для прибавления единицы к переменной.

7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений?

Ответ: mov eax,rem call sprint mov eax,edx call iprintLF

Задание для самостоятельной работы:

1. Я попыталась составить программу для вычисления 5(⊠ – 1)^2. Не знаю в чем проблема, но исходный файл не создается.



```
....dk.sci.pfu.edu.ru/home/d/v/dvgandich/work/study/2022-2023/
;-----------------------------
; Программа вычисления выражения
;-----------------------------
%include 'in_out.asm'    ; подключение внешнего файла

SECTION .data
msg:    'Введите x: ',0
div: DB 'Результат: ',0

SECTION .bss
x:      RESB 80
y:      RESB 80

SECTION .text
GLOBAL _start

_start:

mov eax,msg
call sprintLF

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

mov eax, x
dec eax
mul eax
mov ebx,5
mul ebx

mov [y],eax             ; запись результата вычисления в 'edi'

; ---- Вывод результата на экран

mov eax,div             ; вызов подпрограммы печати
call sprint             ; сообщения 'Результат: '
mov eax,[y]             ; вызов подпрограммы печати значения
call iprintLF           ; из 'edi' в виде символов

call quit               ; вызов подпрограммы завершения
```

4 Вывод

Мы освоили арифметические функции языка ассемблера nasm. . ::: {#refs} :::