

Отчёт по лабораторной работе 6

Архитектура вычислительных систем

Колосов Даниил Дмитриевич

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Написать программу вычисления выражения $y = f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x , вычислять заданное выражение в зависимости от введенного x , выводить результат вычислений. Вид функции $f(x)$ 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 x_1 и x_2 из 6.3.

3 Выполнение лабораторной работы

1. Создаем каталог для программ лабораторной работы No 6, перейдем в него и создадим файл lab6-1.asm: Вводим код и выводим ответ:

```
ddkolosov@dk3n54 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab06
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch lab6-1.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc

ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
lab6-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1
j
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.1: Создание

2. Далее заходим в код и убираем кавычки

```

GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintf
call quit

```

Рис. 3.2: Ковычки

3. Выводим на экран

```

ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $

```

Рис. 3.3: Вывод

4. Создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы из листинга 7.2. Выводим ответ:

```

ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
106
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $

```

Рис. 3.4: Вывод

5. Далее заходим в код и убираем ковычки

```
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.5: Вывод

- Далее заходим в код и смотрим на различие `iprintLF` и `iprint`

```
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.6: Вывод

- Создайте файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`: И выводим:

```
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-3.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 3.7: Вывод

- Изменяем и выводим:

```
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-3.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 3.8: Вывод

- Далее создаем файл `variant.asm` Заходим, вставляем и выводим:

```
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch variant.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc

ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf variant.asm
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o variant variant.o
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant.o
bash: ./variant.o: Отказано в доступе
ddkolosov@dk3n54 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant
Введите No студенческого билета:
1132229061
Ваш вариант: 2
```

Рис. 3.9: Вывод

10. Ответы на вопросы

11. Какие строки листинга 7.4 отвечают за вывод на экран сообщения 'Ваш вариант:'?

Ответ: `mov eax,rem call sprint`

2. Для чего используются следующие инструкции? `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` `call sread` - для считывания в переменную какого-то числа

3. Для чего используется инструкция 'call atoi'?

Ответ: Конвертирует строку, на которую указывает параметр `str`, 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: Остаток запишется 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`

11. Вариант 2 Решение:

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Вариант 2: (12x + 3)5 ',0
gsm: DB 'Введите значение для x: ',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov eax, gsm
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi
mov ebx, 12
mul ebx
add eax, 3
mov ebx, 5
mul ebx
mov edi, eax

mov eax, rem
call sprintf
mov eax, edi
call sprintf

call quit
```

Рис. 3.10: Код

```
ddkolosov@dk3n66 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf variant.asm
ddkolosov@dk3n66 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o variant variant.o
ddkolosov@dk3n66 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant
Вариант 2: (10 + 2x)/3
Введите значение для x:
1
Результат: 4
```

Рис. 3.11: Вывод

```
ddkolosov@dk3n66 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf variant.asm
ddkolosov@dk3n66 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o variant variant.o
ddkolosov@dk3n66 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant
Вариант 2: (10 + 2x)/3
Введите значение для x:
6
Результат: 7
```

Рис. 3.12: Вывод

4 Выводы

Я освоил арифметических инструкций языка ассемблера NASM.