

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6**

*дисциплина: Архитектура вычислительных систем*

Студент: Чистякова Дарья Владимировна

Группа: НБИбд-03-22

**МОСКВА**

2022 г.

## Цель работы:

Освоить арифметические инструкции языка ассемблера NASM.

## Ход работы:

1) Создаем каталог для программ лабораторной работы № 7, переходим в него и создаем файл lab7-1.asm. (Рис. 1)

```
dvchistyakova@dk3n31 ~ $ mkdir ~/work/arch-pc/lab07
dvchistyakova@dk3n31 ~ $ cd ~/work/arch-pc/lab07
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ touch lab7-1.asm
```

Рис. 1. Создание каталога с файлом

2) Рассмотрим примеры программ вывода символьных и численных значений.

Программы будут выводить значения записанные в регистр eax. (Рис. 2)

Создаем исполняемый файл и запускаем его. (Рис. 3)

```
lab7-1.asm [-M--3 9 L: 1+12 13/ 13] *(172 / 172b) <EOF>
#include <stdio.h>
SECTION .text
buf1: RESB 80
SECTION .start
GLOBAL _start
_start:
mov eax, 5
mov ebx, 4
add eax, ebx
mov [buf1], eax
mov ebx, buf1
call sprintf
call quit
```

Сохранить файл

Подтверждаете запись файла "/afs/.dk.sci.pfu.edu.ru/home/d/v/dvchistyakova/work/arch-pc/lab07/lab7-1.asm"?

ОХРАНИТЬ [ Прервать ]

Рис. 2. Текст программы

```
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ./lab7-1
j
```

Рис. 3. Запуск файла

3) Далее изменяем текст программы и вместо символов, записываем в регистры числа. Исправляем текст программы. Создаем исполняемый файл и запускаем его. (Рис. 4)

```
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ./lab7-1
```

Рис. 4. Запуск файла

Этот символ не отображается при выводе на экран.

4) Создаем файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и вводим в него текст программы. (Рис. 5)

Создаем исполняемый файл и запускаем его. (Рис. 6)

```
bininclude "bin\win32\user32.dll"
SECTION .text
GLOBAL _start
_start:
mov eax, 0
mov ebx, 1
add ebx, 2
call iprintLF
call quit
```

Сохранить файл  
Подтверждаете запись файла "/afs/.dk.sci.pfu.edu.ru/home/d/v/dvchistyakova/work/arch-pc/lab07/lab7-2.asm"?  
[ Сохранить ] [ Прервать ]

Рис. 5. Текст программы

```
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ./lab7-2
10
```

Рис. 6. Запуск файла

5) Аналогично предыдущему примеру изменяем символы на числа.

Создаем исполняемый файл и запускаем его. (Рис. 7)

```
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ./lab7-2
10
```

Рис. 7. Запуск файла

6) Создаем файл lab7-3.asm в каталоге ~/work/arch-pc/lab07 и вводим в него текст программы. (Рис. 8)

Создаем исполняемый файл и запускаем его. (Рис. 9)

Изменяем текст программы для вычисления выражения  $f(x)=(4*6+2)/5$ .

Создаем исполняемый файл и проверяем его работу. (Рис. 10)

```
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
```

Рис. 8. Текст программы

```

dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
dvchistyakova@dk3n31 ~/work/arch-pc/lab07 $ ./lab7-3
Результат: 4
Остаток от деления: 1

```

Рис. 9. Запуск файла

```

Результат: 5
Остаток от деления: 1

```

Рис. 10. Запуск файла

7) Создаем файл `variant.asm` в каталоге `~/work/arch-pc/lab07` и вводим в него текст программы. (Рис. 11)

Создаем исполняемый файл и запускаем его.

```

%include 'in_out.asm'
SECTION .data
msg: DB 'Введите Но 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`

```

Рис. 11. Текст программы

Программа вывела 4 вариант.

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

```
mov eax,rem
```

```
call sprint
```

2. Для чего используются следующие инструкции `nasm`?

```
mov ecx,x – запись входной переменной в регистр ecx;
```

```
mov edx,80 – запись переменной в регистр edx;
```

```
call sread – вызов процедуры чтения данных.
```

3. Для чего используется инструкция “`call atoi`”?

Для вызова подпрограммы преобразования ASCII кода в число, ``eax=x``.

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

```
xor edx,edx
```

```
mov ebx,20
```

```
div ebx
```

```
inc edx.
```

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

В регистр ebx.

6. Для чего используется инструкция “inc edx”?

Для увеличения на 1.

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

```
mov eax,rem
```

```
call sprint
```

```
mov eax,edx
```

```
call iprintLF.
```

### Задание для самостоятельной работы.

1) Напишем программу вычисления выражения  $\frac{4}{3}(x - 1) + 5$  (Вариант 4).

(Рис. 12)

Получаем исполняемый файл и проверяем его работу для значений  $x_1 = 4$  и  $x_2 = 10$  (Вариант 4). (Рис. 13)

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
rem: DB 'Ответ: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
xor edx, edx
sub eax, 1
mov ebx, 4
mul ebx
mov ecx, 3
div ecx
add eax, 5
mov edi, eax
mov eax, rem
call sprint
mov eax, edi
call iprintLF
call quit
```

Рис. 12. Текст программы

```
dvchistyakova@nbibd-03-22:~/work/arch-pc/lab07$ nasm -f elf lab7-sr.asm
dvchistyakova@nbibd-03-22:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-sr lab7-sr.o
dvchistyakova@nbibd-03-22:~/work/arch-pc/lab07$ ./lab7-sr
Введите x: 4
Ответ: 9
dvchistyakova@nbibd-03-22:~/work/arch-pc/lab07$ ./lab7-sr
Введите x: 10
Ответ: 17
```

Рис. 13. Запуск файла

**Вывод:**

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