

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Дисциплина: Архитектура компьютера

Студент: Диденко Герман Максимович

Студ. Билет: № 1032253558

Группа: НКАбд-02-25

МОСКВА

2025 г.

Оглавление

1. Цель работы.....	3
2. Задание.....	3
3. Выполнение лабораторной работы.....	3
3.1. Символьные и численные данные в NASM.....	3
3.2. Символьные и численные данные в NASM.....	6
4. Задание для самостоятельной работы.....	8
Задание 1.....	9
5. Вывод.....	9

1. Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM

2. Задание

На основе методических указаний ознакомится с арифметическими инструкциями языка ассемблера NASM

3. Выполнение лабораторной работы

3.1. Символьные и численные данные в NASM

Создаем каталог для программ лабораторной работы № 6, переходим в него и создаем файл lab6-1.asm

```
gmdidenko@rudn:~$ mkdir ~/work/arch-pc/lab06
gmdidenko@rudn:~$ cd ~/work/arch-pc/lab06
gmdidenko@rudn:~/work/arch-pc/lab06$ touch lab6-1.asm
```

Рис. 1 Созданный файл lab6-1.asm

Вводим в файл lab6-1.asm код из Листинга 6.1. Проверяем наличие кода в файле

```
gmdidenko@rudn:~/work/arch-pc/lab06$ cat lab6-1.asm
%include 'in_out.asm'
SECTION .bss
buf1: RESB 80
```

Рис. 2 Содержимое файла lab6-1.asm

Создаем исполняемый файл и запускаем его

```
gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-1
j
```

Рис. 3 Создание и запуск программы lab6-1

Затем изменим текст программы и вместо символов, запишем в регистры числа

```
mov eax,6
mov ebx,4
```

Рис. 4 Измененные регистры числа в lab6-1.asm

Создаем новый исполняемый файл и запускаем его

```
gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-1

gmdidenko@rudn:~/work/arch-pc/lab06$ █
```

Рис. 5 Вывод измененной программы lab6-1

Символ 10 - это \n (перенос строки), поэтому при выводе мы видим два переноса (две пустые строки)

Далее создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы из Листинга № 6.2

```

gmdidenko@rudn:~/work/arch-pc/lab06$ cat lab6-2.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'

```

Рис. 6 Содержимое файла lab6-2.asm

Создаю исполняемый файл и запускаю его.

```

gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-2
106

```

Рис. 7 Компиляция и запуск программы lab6-2

Аналогично предыдущему примеру изменим символы на числа. Создаю обновленный исполняемый файл и запускаю его.

```

gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-2
10

```

Рис. 8 Компиляция и запуск измененной программы lab6-2

В результате выполнение программы получаем результат 10.

Теперь заменяем функцию `iprintlnf` на `iprint`. Создаем исполняемый файл и запускаем его.

```

gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-2
10gmdidenko@rudn:~/work/arch-pc/lab06$

```

Рис. 9 Компиляция и запуск измененной программы lab6-2 (`iprint`)

Программы `ciprintLF` и `iprint` отличаются переносом строки.

3.2. Символьные и численные данные в NASM

Создаем файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`. Вводим в файл код из Листинга № 6.3.

```
gmdidenko@rudn:~/work/arch-pc/lab06$ cat lab6-3.asm
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
```

Рис. 10 Файл `lab6-3.asm`

Создаем исполняемый файл и запускаем его.

```
gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рис. 11 Компиляция и запуск программы `lab6-3`

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

```
; ---- 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
```

Рис. 13 Измененный файл `lab6-3.asm`

Создаем исполняемый файл и проверяем его работу.

```

gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 14 Результат программы lab6-3

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

- вывести запрос на введение № студенческого билета
- вычислить номер варианта по формуле: $(Sn \bmod 20) + 1$, где Sn – номер студенческого билета (В данном случае $a \bmod b$ – это остаток от деления a на b).
- вывести на экран номер варианта

Создайте файл variant.asm в каталоге ~/work/arch-pc/lab06 и записываем в него код из Листинга № 6.4. Проверяем содержимое файла.

```

gmdidenko@rudn:~/work/arch-pc/lab06$ cat variant.asm
%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:

```

Рис. 15 Содержимое файла variant.asm

Создаем исполняемый файл и запускаем его

```

gmdidenko@rudn:~/work/arch-pc/lab06$ nasm -f elf variant.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
gmdidenko@rudn:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032253558
Ваш вариант: 19

```

Рис. 15 Вывод программы variant

Ответы на вопросы:

1.

```
mov eax,rem  
call sprint
```

2.

```
mov ecx, x - загружает адрес переменной x в регистр ECX (буфер для ввода)  
mov edx, 80 - устанавливает максимальную длину ввода (80 символов)  
call sread - вызывает функцию чтения строки с клавиатуры
```

3.

Функция `atoi` преобразует ASCII-строку в целое число. После вызова в регистре EAX будет находиться числовое значение введенного номера студенческого билета.

4.

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

5.

При выполнении инструкции `div ebx` остаток от деления записывается в регистр EDX.

6.

Команда `inc edx` увеличивает значение в регистре EDX на 1. В данном контексте она преобразует остаток от деления (0-19) в номер варианта (1-20).

7.

```
mov eax,rem  
call sprint  
mov eax,edx  
call iprintLF
```

4. Задание для самостоятельной работы

Перехожу к заданиям для самостоятельного выполнения.

Задание 1.

Мне попался вариант №19, поэтому беру выражение $(1/3 x + 5) \cdot 7$

Создаю файл lab6-4.asm и вношу в него код моей программы. Содержимое файла проверяю.

```
gmdidenko@rudn:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-4.asm
gmdidenko@rudn:~/work/arch-pc/lab06$ cat lab6-4.asm
%include 'in_out.asm'

SECTION .data
msg db "Вычисление функции f(x) = ((1/3)x + 5) * 7",0
inmsg db "Введите x: ",0
resmsg db "Результат: ",0
```

Рис. 16 Содержимое файла lab6-4.asm

Создаю исполняемый файл и запускаю его. Значение для X (3 и 9) беру из таблицы. Запускаю и проверяю вывод программы.

```
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-4
Вычисление функции f(x) = ((1/3)x + 5) * 7
Введите x: 3
Результат: 42
gmdidenko@rudn:~/work/arch-pc/lab06$ ./lab6-4
Вычисление функции f(x) = ((1/3)x + 5) * 7
Введите x: 9
Результат: 56
```

Рис. 15 Запуск программы lab6-4 с выводом результата

5. Вывод

В ходе выполнения лабораторной работы была достигнута цель по освоению арифметических инструкций языка ассемблера NASM.