

Шаблон отчёта по лабораторной работе

Простейший вариант

Кудинец Максим Антонович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Задания для самостоятельной работы	20
6	Выводы	22

Список иллюстраций

4.1	Переход в каталог и создание файла	8
4.2	Программа вывода значения регистра еах	9
4.3	Исполнение программы из листинга 6.1	10
4.4	Исправленный текст программы lab6-1.asm	11
4.5	Исполнение программы lab6-1	12
4.6	Исполнение программы lab6-2	12
4.7	Исправленный текст программы	13
4.8	Запуск программы	13
4.9	Внесение изменений в программу	14
4.10	Запуск исполняемого файла	14
4.11	Текст программы lab6-3.asm	15
4.12	Запуск исполняемого файла	15
4.13	Исправление текста программы	16
4.14	Запуск исполняемого файла lab6-3	16
4.15	Текст программы variant.asm	18
4.16	Запуск программы	19
5.1	Текст программы lab6-4.asm	21
5.2	Запуск программы	21

Список таблиц

1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

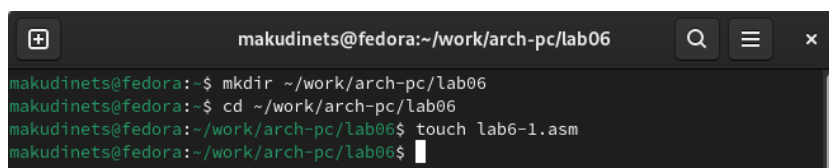
2 Задание

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. При выполнении задания преобразовывать (упрощать) выражения для $f(x)$ нельзя. При выполнении деления в качестве результата можно использовать только целую часть от деления и не учитывать остаток (т.е. $5 \div 2 = 2$).

3 Теоретическое введение

4 Выполнение лабораторной работы

1. Создаю каталог для программ лабораторной работы №6, перехожу в него и создаю файл lab6-1.asm.



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

Рис. 4.1: Переход в каталог и создание файла

2. Ввожу в файл lab6-1.asm текст программы из листинга 6.1. Запускаю исполняемый файл.


```
%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 printf

    call quit
```

Рис. 4.2: Программа вывода значения регистра eax

```
makudinets@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
makudinets@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 lab6-1.o -o lab6-1
makudinets@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o
makudinets@fedora:~/work/arch-pc/lab06$ ./lab6-1
j
makudinets@fedora:~/work/arch-pc/lab06$
```

Рис. 4.3: Исполнение программы из листинга 6.1

3. Изменим текст программы и вместо символов, запишем в регистры числа. Запустим исправленную программу. Теперь на экран вывелся символ с кодом 10. В соответствии с ASCII таблицей это символ перевода строк и он не отображается при выводе на экран.

```
%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 sprintf

call quit
```

Рис. 4.4: Исправленный текст программы lab6-1.asm

```

makudinets@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
makudinets@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 lab6-1.o -o lab6-1
makudinets@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o
makudinets@fedora:~/work/arch-pc/lab06$ ./lab6-1
makudinets@fedora:~/work/arch-pc/lab06$

```

Рис. 4.5: Исполнение программы lab6-1

4. Создадим файл lab6-2.asm. Преобразуем текст программы из Листинга 6.1 с использованием подпрограмм из файла in_out.asm, которые преобразуют ASCII символы в числа и обратно.

```

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

```

Рис. 4.6: Исполнение программы lab6-2

5. Аналогично предыдущему примеру исправим символы на числа и запустим программу. Теперь при запуске программы было получено само число 10, а не символ с соответствующим кодом в системе ASCII.

```

%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

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

call quit

```

Рис. 4.7: Исправленный текст программы

```

makudinets@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
makudinets@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 lab6-2.o -o lab6-2
makudinets@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o
makudinets@fedora:~/work/arch-pc/lab06$ ./lab6-2
10
makudinets@fedora:~/work/arch-pc/lab06$

```

Рис. 4.8: Запуск программы

6. Далее заменяю функцию iprintLF на iprint. Запускаю исполняемый файл. Изменений нет.

```

%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit

```

Рис. 4.9: Внесение изменений в программу

```

makudinets@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
makudinets@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 lab6-2.o -o lab6-2
makudinets@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o
makudinets@fedora:~/work/arch-pc/lab06$ ./lab6-2
10makudinets@fedora:~/work/arch-pc/lab06$

```

Рис. 4.10: Запуск исполняемого файла

7. Создаю файл lab6-3.asm. Ввожу в него текст программы из листинга 6.3.

Запускаю исполняемый файл.

```
%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
```

Рис. 4.11: Текст программы lab6-3.asm

```
makudinets@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
makudinets@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 lab6-3.o -o lab6-3
makudinets@fedora:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1.asm  lab6-2      lab6-2.o    lab6-3.asm
lab6-1      lab6-1.o    lab6-2.asm  lab6-3      lab6-3.o
makudinets@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
makudinets@fedora:~/work/arch-pc/lab06$
```

Рис. 4.12: Запуск исполняемого файла

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

```

%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,5 ; 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

```

Рис. 4.13: Исправление текста программы

```

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

```

Рис. 4.14: Запуск исполняемого файла lab6-3

9. 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: $(S_n \bmod 20) + 1$, где S_n – номер студенческого билета (В данном случае $a \bmod b$ – это остаток от деления a на b). • вывести на экран номер 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. 4.15: Текст программы variant.asm

```
makudinets@fedora:~/work/arch-pc/lab06$ nasm -f elf nomervar.asm
makudinets@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 nomervar.o -o nomervar
makudinets@fedora:~/work/arch-pc/lab06$ ./nomervar
Введите № студенческого билета:
1132246729
Ваш вариант: 10
makudinets@fedora:~/work/arch-pc/lab06$
```

Рис. 4.16: Запуск программы

5 Задания для самостоятельной работы

1. Создаю файл lab6-4.asm и ввожу в него текст программы для вычисления значения выражения $5(x+18)-28$. Выражение было в варианте 10. Вычисляю значение выражения для заданных x .

```

%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0

SECTION .bss ; секция неиницированных данных
x: RESB 80 ; Переменная, вводится с клавиатуры, выделенный размер - 80 байт

SECTION .text
GLOBAL _start
_start:

    ; Вычисление выражения
    mov eax,msg ; EAX=выводимое сообщение
    call sprint
    mov ecx, x ; запись адреса переменной x в ecx
    mov edx, 80 ; запись длины вводимого значения в edx
    call sread
    mov eax,x ; вызов подпрограммы преобразования
    call atoi ; ASCII кода в число, eax = x
    mov ebx,18 ;ebx=18
    add eax,ebx ;eax = x+18
    mov ebx,5
    mul ebx ;eax=eax*5=5*(x+18), edx: содержит остаток от деления
    mov ebx, 28 ;запись значения 28 в регистр ebx
    sub eax,ebx ;eax=eax-ebx = 5(x+18)-28 - вычитаем 28 от остатка деления
    mov edi, eax ; запись результата вычисления в edi

    ; Вывод результата на экран
    mov eax,rem
    call sprint
    mov eax,edi
    call iprintLF ; из 'edi' в виде символов
    call quit ; вызов подпрограммы завершения

```

Рис. 5.1: Текст программы lab6-4.asm

```

makudinets@fedora:~/work/arch-pc/lab06$ nasm -f elf samrab.asm
makudinets@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 samrab.o -o samrab
makudinets@fedora:~/work/arch-pc/lab06$ ./samrab
Введите значение переменной x: 2
Результат: 72
makudinets@fedora:~/work/arch-pc/lab06$ ./samrab
Введите значение переменной x: 3
Результат: 77
makudinets@fedora:~/work/arch-pc/lab06$

```

Рис. 5.2: Запуск программы

6 Выводы

В результате выполнения лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.