

Отчёт по лабораторной работе №5

Номер студенческого билета: 1032253642

Пашутина Анна Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Символьные и численные данные в NASM	7
3.2	Выполнение арифметических операций в NASM	10
3.3	Ответы на вопросы по программе	13
3.4	Задание для самостоятельной работы	13
4	Выводы	16

Список иллюстраций

3.1	Создаем каталог lab06 с помощью команды mkdir и файл lab6-1.asm с помощью команды touch	7
3.2	Заполняем файл lab6-1.asm	7
3.3	Запускаем файл и смотрим на его работу	8
3.4	Изменяем файл в соответствии с заданием	8
3.5	Создаем исполняемый файл, запускаем этот файл и смотрим на его работу	8
3.6	Создаем файл	8
3.7	Открываем и заполняем файл lab6-2.asm	9
3.8	Смотрим на работу программы	9
3.9	Изменяем файл в соответствии с заданием	9
3.10	Смотрим на работу программы	9
3.11	Изменяем файл в соответствии с заданием	10
3.12	Смотрим на работу программы	10
3.13	Создаем файл lab6-3.asm в каталоге ~/work/arch-pc/lab06	10
3.14	Заполняем файл в соответствии с листингом 6.3	11
3.15	Смотрим на результат работы программы	11
3.16	Редактируем файл в соответствии с заданием	11
3.17	Смотрим на результат работы программы	12
3.18	Создаем файл	12
3.19	Заполняем созданный файл variant.asm	12
3.20	Проверяем результат работы программы	12
3.21	Создаем файл lab6-4.asm в нужном каталоге	14
3.22	Заполняем файл в соответствии с задачей	14
3.23	Проверяем работу программы	14
3.24	Проверяем работу программы	15

Список таблиц

1 Цель работы

Освоить арифметических инструкций языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

2 Задание

Написать программы для решения выражений.

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

Создаем каталог для программ ЛБ6, и в нем создаем файл lab6-1.asm (рис.1).

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

Рисунок 3.1: Создаем каталог lab06 с помощью команды mkdir и файл lab6-1.asm с помощью команды touch

Открываем файл в Midnight Commander с помощью команды mc и заполняем его в соответствии с листингом 6.1 (рис.2).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
/home/aapashutina/work/arch-pc/lab06/lab6-1.asm *
GNU nano 7.2
%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
```

Рисунок 3.2: Заполняем файл lab6-1.asm

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

```

aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf64 lab6-1.asm -o lab6-1.o
lab6-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ./lab6-1
)
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$

```

Рисунок 3.3: Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис.4).

```

GNU nano 7.2 mc [aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06
/home/aapashutina/work/arch-pc/lab06/lab6-1.asm
#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

```

Рисунок 3.4: Изменяем файл в соответствии с заданием

Создаем исполняемый файл и запускаем его (рис.5).

```

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

aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$

```

Рисунок 3.5: Создаем исполняемый файл, запускаем этот файл и смотрим на его работу

Создаем новый файл lab6-2.asm в каталоге lab06(рис.6).

```

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

aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$

```

Рисунок 3.6: Создаем файл

Открываем файл lab6-2.asm и заполняем его в соответствии с листингом 6.2 (рис.7).


```
mc[aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06
GNU nano 7.2 /home/aapashutina/work/arch-pc/lab06/lab6-2.asm
#include "in_out.asm"
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рисунок 3.7: Открываем и заполняем файл lab6-2.asm

Создаем исполняемый файл и запускаем его (рис.8).

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

Рисунок 3.8: Смотрим на работу программы

Снова открываем файл lab6-2.asm для редактирования и убираем кавычки с числовых значений (рис.9).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
GNU nano 7.2 /home/aapashutina/work/arch-pc/lab06/lab6-2.asm *
#include "in_out.asm"
SECTION .text
GLOBAL _start
_start:
mov eax, 6
mov ebx, 4
add eax, ebx
call iprintLF
call quit
```

Рисунок 3.9: Изменяем файл в соответствии с заданием

Создаем исполняемый файл и запускаем его (рис.10).

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

Рисунок 3.10: Смотрим на работу программы

Снова открываем файл lab6-2.asm для редактирования и меняем iprintLF на iprint (рис.11).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
GNU nano 7.2 /home/aapashutina/work/arch-pc/lab06/lab6-2.asm
#include "in_out.asm"
section .text
global _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рисунок 3.11: Изменяем файл в соответствии с заданием

Создаем исполняемый файл и запускаем его (рис.12).

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

Рисунок 3.12: Смотрим на работу программы

Вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.(ответ на вопрос задания)

3.2 Выполнение арифметических операций в NASM

Создаем новый файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`(рис.13).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$
```

Рисунок 3.13: Создаем файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`

Открываем файл и редактируем в соответствии с листингом 6.3(программа вычисления арифметического выражения $f(x)=(5*2+3)/3$)(рис.14).

```

mc [aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06
GNU nano 7.2 /home/aapashutina/work/arch-pc/lab06/lab6-3.asm *
#include "in_out.asm"
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintf

mov eax,rem
call sprint
mov eax,edx
call iprintf

call quit

```

Рисунок 3.14: Заполняем файл в соответствии с листингом 6.3

Создаем исполняемый файл и запускаем его (рис.15).

```

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

```

Рисунок 3.15: Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения $f(\boxtimes) = (4 * 6 + 2)/5$ (рис.16).

```

aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
GNU nano 7.2 /home/aapashutina/work/arch-pc/lab06/lab6-3.asm *
#include "in_out.asm"
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintf

mov eax,rem
call sprint
mov eax,edx
call iprintf

call quit

```

Рисунок 3.16: Редактируем файл в соответствии с заданием

Компилируем файл и запускаем программу (рис.17).

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

Рисунок 3.17: Смотрим на результат работы программы

Создаем новый файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис.18).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$
```

Рисунок 3.18: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.4 (рис.19).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
GNU nano 7.2 /home/aapashutina/work/arch-pc/lab06/variant.asm
#include "in.out.asm"
/*-----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 sprintf
mov ecx, x
mov edx, 80
call read
mov eax, 1
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprintf
mov eax, edx
call printf
call quit
```

Рисунок 3.19: Заполняем созданный файл variant.asm

Компилируем файл и запускаем его. Программа выдала ответ 3, проверим аналитически верно ли это. Номер моего студенческого = 1032253642, остаток деления его на 20=2, далее прибавляем 1 и получаем ответ=3. Программа верно вычислила вариант.(рис.20).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf variant.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032253642
Ваш вариант: 3
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$
```

Рисунок 3.20: Проверяем результат работы программы

3.3 Ответы на вопросы по программе

1. Строка «mov eax,ret» и строка «call sprint» отвечают за вывод на экран сообщения „Ваш вариант:“.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре esx, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.
3. Инструкция «call atoi» используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в регистре eax.
4. Строка «xor edx,edx» обнуляет регистр edx перед выполнением деления. Строка «mov ebx,20» загружает значение 20 в регистр ebx. Строка «div ebx» выполняет деление регистра eax на значение регистра ebx с сохранением частного в регистре eax и остатка в регистре edx.
5. Остаток от деления записывается в регистр edx.
6. Инструкция «inc edx» используется для увеличения значения в регистре edx на 1. В данном случае, она увеличивает остаток от деления на 1.
7. Строка «mov eax,edx» передает значение остатка от деления в регистр eax. Строка «call iprintLF» вызывает процедуру iprintLF для вывода значения на экран вместе с переводом строки.

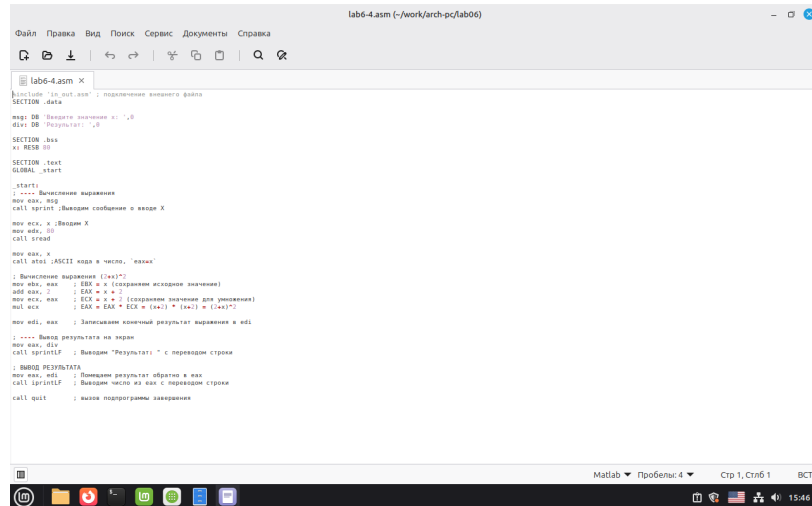
3.4 Задание для самостоятельной работы

Создаем новый файл lab6-4.asm в каталоге ~/work/arch-pc/lab06(рис.21).

```
aapashutina@aapashutina-VirtualBox: ~/work/arch-pc/lab06
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-4.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$
```

Рисунок 3.21: Создаем файл lab6-4.asm в нужном каталоге

Открываем его и заполняем, чтобы программа вычисляла значение выражения $f(x)=(2+x)^2$ в зависимости от вводимых x . (рис.22).



```
lab6-4.asm X
include "is_out.asm" ; подключение внешнего файла
SECTION .data
msg: DB "Введите значение x: ",0
fmt DB "%d\n",0

SECTION .bss
x: RESB 10

SECTION .text
GLOBAL _start

_start:
; --- Вывесление сообщения
mov eax, msg
call printf ; Выводим сообщение о вводе X

mov ecx, x ; Вводим X
mov edi, 0
call read

mov ebx, x
call atoi ; ASCII код в число, 'ebx'

; Вычисление выражения (2+x)^2
mov ebx, ebx ; EBX = x (сохранены исходные значения)
add ebx, 2 ; EBX = x + 2
mov ecx, ebx ; ECX = x + 2 (сохранены значения для умножения)
mul ecx ; EDI = EBX * ECX = (x+2) * (x+2) = (2+x)^2
mov edi, ebx ; Записываем конечный результат выражения в edi

; --- Вывод результата на экран
mov ebx, edi
call printf ; Выводим "Результат:" с переводом строки

; Вывод результата
mov ecx, edi ; Помещаем результат обратно в ecx
call printf ; Выводим число из ecx с переводом строки

call quit ; вынос программы завершения
```

Рисунок 3.22: Заполняем файл в соответствии с задачей

Компилируем программу и проверяем ответ программы для $x=2$. Программа вывела ответ 16, что является верным решением. (рис.23).

```
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ./lb6-4
Введите значение x: 2
Результат:
16
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$
```

Рисунок 3.23: Проверяем работу программы

Компилируем программу и проверяем для $x=8$. Программа вывела ответ 100, что является верным решением. (рис.24).

```
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$ ./lb6-4
Введите значение x: 8
Результат:
100
aapashutina@aapashutina-VirtualBox:~/work/arch-pc/lab06$
```

Рисунок 3.24: Проверяем работу программы

4 Выводы

Мы приобрели навыки создания исполняемых файлов для решения выражений и освоили арифметические инструкции в NASM.