

Лабораторная работа №6

Дисциплина: Архитектура компьютера

Комягин Андрей Николаевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа 6.1	6
2.2	Программа 6.2	8
2.3	Программа 6.3	9
2.4	Программа 6.4	10
2.5	Самостоятельная работа	11
3	Выводы	13

Список иллюстраций

2.1	Создание каталога и файла ассемблера	6
2.2	Ввод команлы	7
2.3	Запуск файла	7
2.4	Запуск измененной программы	8
2.5	Создание программы 6.2	8
2.6	Запуск команды со строками	9
2.7	Запуск команды с числами	9
2.8	Сравнение команд вывода	9
2.9	Создание и запуск файла	10
2.10	Изменение и запуск файла	10
2.11	Запуск программы	10
2.12	Запуск программы	12

Список таблиц

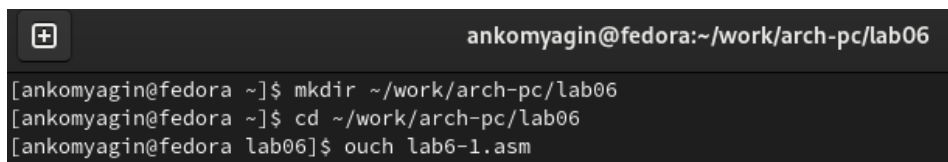
1 Цель работы

Изучить арифметические инструкции языка ассемблера NASM и применить полученные знания на практике.

2 Выполнение лабораторной работы

2.1 Программа 6.1

Создадим каталог для программ лабораторных работ, в нем создадим файл **lab6-1.asm**. (рис. 2.1).

A terminal window with a dark background. The title bar shows a plus icon and the text 'ankomyagin@fedora:~/work/arch-pc/lab06'. The terminal contains three lines of text: the first line shows the user at the home directory (~) with a prompt character (\$) and the command 'mkdir ~/work/arch-pc/lab06'; the second line shows the user at the same directory with a prompt character (\$) and the command 'cd ~/work/arch-pc/lab06'; the third line shows the user at the directory 'lab06' with a prompt character (\$) and the command 'ouch lab6-1.asm'.

```
ankomyagin@fedora:~/work/arch-pc/lab06
[ankomyagin@fedora ~]$ mkdir ~/work/arch-pc/lab06
[ankomyagin@fedora ~]$ cd ~/work/arch-pc/lab06
[ankomyagin@fedora lab06]$ ouch lab6-1.asm
```

Рис. 2.1: Создание каталога и файла ассемблера

Введем в созданный файл текст из листинга **6.1** (рис. 2.2).

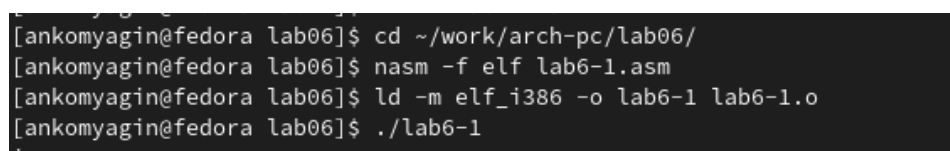


```
Открыть + lab6-1.asm ~/work/arch-pc/lab06
5-1.asm lab5-1 (копия).as lab5-2.asm lab5-2 (копия).as lab6-1.asm x

%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. 2.2: Ввод команды

Создадим исполняемый файл и запустим его (рис. 2.3). Результат его работы - вывод символа **j**.



```
[ankomyagin@fedora lab06]$ cd ~/work/arch-pc/lab06/
[ankomyagin@fedora lab06]$ nasm -f elf lab6-1.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[ankomyagin@fedora lab06]$ ./lab6-1
j
```

Рис. 2.3: Запуск файла

Изменим текст программы (вместо символов запишем в регистры числа) и запустим её (рис. 2.4). Программа выводит пустой символ (переноса строки).

```
[ankomyagin@fedora lab06]$ nasm -f elf lab6-1_2.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o lab6-1_2 lab6-1_2.o
[ankomyagin@fedora lab06]$ ./lab6-1_2

[ankomyagin@fedora lab06]$ ./lab6-1_2

[ankomyagin@fedora lab06]$
```

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

Рис. 2.4: Запуск измененной программы

2.2 Программа 6.2

Создадим файл **lab6-2.asm** и введем в него текст программы из листинга 6.2 (рис. 2.5).

```
[ankomyagin@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
[ankomyagin@fedora lab06]$
```

```
Открыть ▾ + lab6-2.asm
~/work/arch-pc/lab06

%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 2.5: Создание программы 6.2

Создадим исполняемый файл и запустим его (рис. 2.6). Результат работы - число 106


```
[ankomyagin@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
[ankomyagin@fedora lab06]$ nasm -f elf lab6-2.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[ankomyagin@fedora lab06]$ ./lab6-2
106
[ankomyagin@fedora lab06]$
```

Рис. 2.6: Запуск команды со строками

Заменяем строки на числа, создадим исполняемый файл и запустим его (рис. 2.7). Результат работы программы - 10.

```
106
[ankomyagin@fedora lab06]$ nasm -f elf lab6-2_2.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o lab6-2_2 lab6-2_2.o
[ankomyagin@fedora lab06]$ ./lab6-2_2
10
[ankomyagin@fedora lab06]$
```

```
-----
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
```

Рис. 2.7: Запуск команды с числами

Заменяем функцию **iprintLF** на **iprint**. Запустим файл. Разница работы программ и наличия и отсутствия переноса строки (рис. 2.8).

```
106
[ankomyagin@fedora lab06]$ nasm -f elf lab6-2_2.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o lab6-2_2 lab6-2_2.o
[ankomyagin@fedora lab06]$ ./lab6-2_2
10
[ankomyagin@fedora lab06]$ nasm -f elf lab6-2_2.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o lab6-2_2 lab6-2_2.o
[ankomyagin@fedora lab06]$ ./lab6-2_2
10[ankomyagin@fedora lab06]$
```

Рис. 2.8: Сравнение команд вывода

2.3 Программа 6.3

Создадим файл **lab6-3.asm** и заполним его в соответствии с листингом **6.3**. Запустим файл, результат работы изображен на (рис. 2.9).

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

Рис. 2.9: Создание и запуск файла

Изменим текст программы для вычисления выражения $(4 \times 6 + 2)/5$ и проверим работоспособность (рис. 2.10).

```
[ankomyagin@fedora lab06]$ nasm -f elf lab6-3.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[ankomyagin@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[ankomyagin@fedora lab06]$
```

```
%include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
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
mov edi,eax ; запись результата вычисления в 'edi'
```

Рис. 2.10: Изменение и запуск файла

2.4 Программа 6.4

Создадим файл **variant.asm**, заполним его в соответствии с листингом 6.4 (рис. 2.11).

```
[ankomyagin@fedora lab06]$ touch ~/work/arch-pc/lab06/variant.asm
[ankomyagin@fedora lab06]$ nasm -f elf variant.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[ankomyagin@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236126
Ваш вариант: 7
[ankomyagin@fedora lab06]$
```

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
```

Рис. 2.11: Запуск программы

Результат программы – 7. Проверим его аналитически. Для определения остатка

деления на 20 достаточно смотреть на 2 последние цифры номера студенческого билета. В моем случае это 26. Остаток деления 26 на 20 = 6, добавляем единицу, которая нужна, чтобы не получился вариант 0, $6+1 = 7$.

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? **Строки `mov eax,rem call sprint`**
2. Для чего используются следующие инструкции? `mov ecx, x mov edx, 80 call sread`. **Для создания переменной x**
3. Для чего используется инструкция “`call atoi`”? **Для преобразования кода символа в число**
4. Какие строки листинга 6.4 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 `ax`**
6. Для чего используется инструкция “`inc edx`”? **Для добавления единицы**
7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычисления? **`mov eax,edx call iprintLF`**

2.5 Самостоятельная работа

Написать программу для вычисления значения функции.

Вариант 7. $f(x) = 5(x - 1)^2$. $x_1 = 3$, $x_2 = 5$

Программа (рис. 2.12).

The image shows a terminal window and a code editor window. The terminal window, titled 'ankomyagin@fedora:~/work/arch-pc/lab06', displays the following commands and output:

```
[ankomyagin@fedora lab06]$ nasm -f elf ans_7.asm
[ankomyagin@fedora lab06]$ ld -m elf_i386 -o ans_7 ans_7.o
[ankomyagin@fedora lab06]$ ./ans_7
Введите x:
3
Ответ: 20
[ankomyagin@fedora lab06]$ ./ans_7
Введите x:
5
Ответ: 80
[ankomyagin@fedora lab06]$
```

The code editor window, titled 'ans_7.asm' and located at '~/work/arch-pc/lab06', shows the assembly code for 'variant.asm'. The code includes comments in Russian and uses standard assembly instructions:

```
%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 sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`

sub eax,1
mov ebx,eax
mul ebx
mov ebx,5
mul ebx
mov edi,eax ; запись результата вычисления в 'edi'
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF
call quit
```

Рис. 2.12: Запуск программы

3 Выводы

Я ходе работы я освоил арифметические конструкции языка ассемблера NASM. Изучил несколько программ и написал собственную.