

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Новичков Максим Алексеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Порядок выполнения лабораторной работы	6
3.2	Ответы на вопросы	11
3.3	Задание для самостоятельной работы	12
4	Выводы	15

Список иллюстраций

3.1	Создаём каталог	6
3.2	заполняем программу	7
3.3	результат	7
3.4	заполняем	8
3.5	результат	8
3.6	подмена	8
3.7	результат	8
3.8	пишем программу	9
3.9	результат	9
3.10	меняем выражение	10
3.11	результат	10
3.12	пишем программу	11
3.13	результат	11
3.14	пишем программу	13
3.15	результат	13
3.16	пишем программу	14
3.17	результат	14

1 Цель работы

Научиться писать и анализировать ассемблерный код с арифметическими операциями и понять синтаксис. Работа поможет развить навыки низкоуровневого программирования и понимания работы процессора.

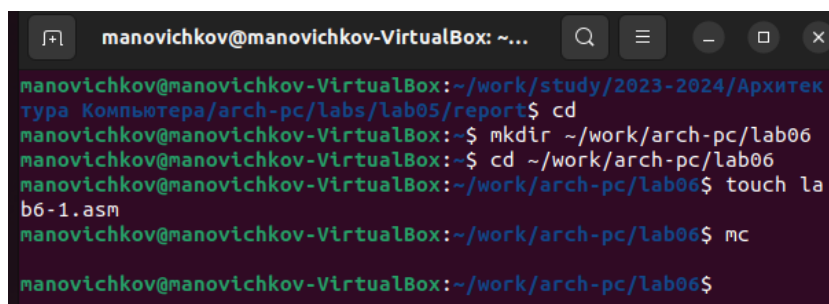
2 Задание

Написать несколько программ для вычислений.

3 Выполнение лабораторной работы

3.1 Порядок выполнения лабораторной работы

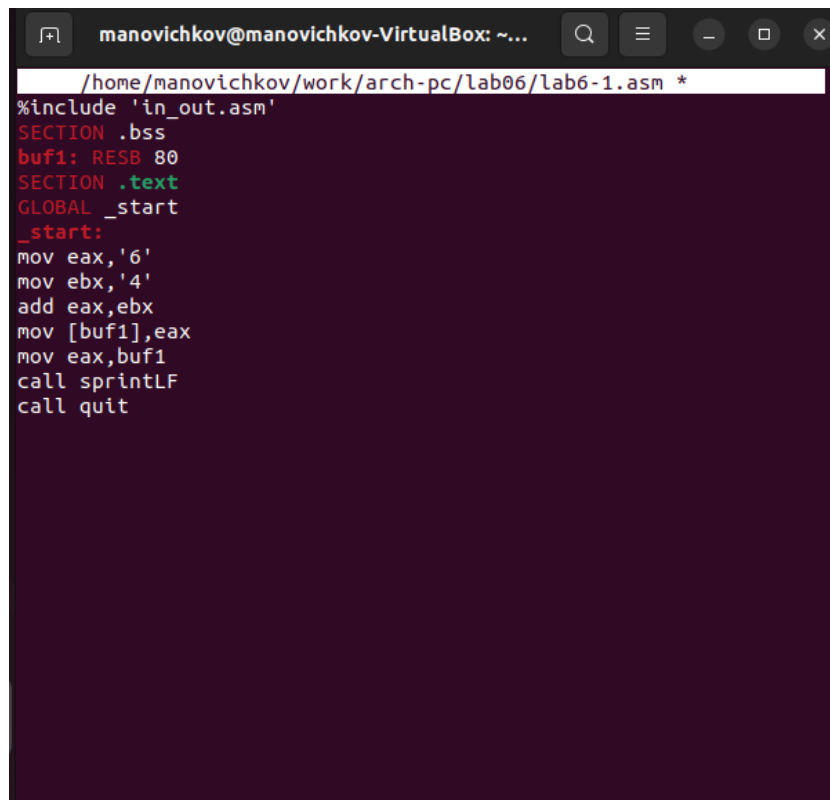
Создайте каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm (рис. 3.1)



```
manovichkov@manovichkov-VirtualBox: ~/...  
manovichkov@manovichkov-VirtualBox:~/work/study/2023-2024/Архитек  
тура Компьютера/arch-pc/labs/lab05/report$ cd  
manovichkov@manovichkov-VirtualBox:~$ mkdir ~/work/arch-pc/lab06  
manovichkov@manovichkov-VirtualBox:~$ cd ~/work/arch-pc/lab06  
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ touch la  
b6-1.asm  
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ mc  
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$
```

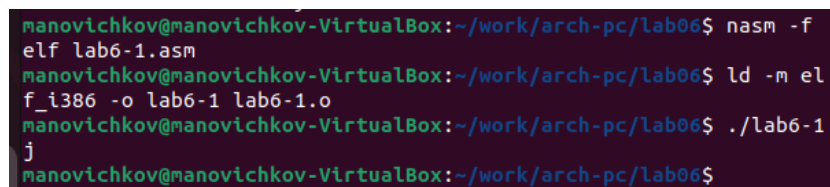
Рис. 3.1: Создаём каталог

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax (рис. 3.3)



```
manovichkov@manovichkov-VirtualBox: ~...
/home/manovichkov/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 sprintLF
call quit
```

Рис. 3.2: заполняем программу



```
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ nasm -f
elf lab6-1.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ld -m el
f_i386 -o lab6-1 lab6-1.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-1
j
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.3: результат

Преобразуем текст программы из Листинга 6.1 с использованием этих функций.
(рис. 3.5)

```

/home/manovichkov/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

```

Рис. 3.4: заполняем

```

manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ nasm -f
elf lab6-2.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ld -m el
f_i386 -o lab6-2 lab6-2.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-2
106
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 3.5: результат

Изменим символы на числа (рис. 3.7)

```

/home/manovichkov/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

```

Рис. 3.6: подмена

результат

Рис. 3.7: результат

В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $f(x) = (5 * 2 + 3)/3$ (рис. 3.9)


```
mc [manovichkov@manovichkov-VirtualBo... /home/manovichkov/work/arch-pc/lab06/lab6-3.asm *
;-----
; Программа вычисления выражения
;-----
%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. 3.8: пишем программу

```
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ nasm -f
elf lab6-3.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ld -m el
f_i386 -o lab6-3 lab6-3.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$
```

Рис. 3.9: результат

Измените текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$.
Создайте исполняемый файл и проверьте его работу. (рис. 3.13)

```

/home/manovichkov/work/arch-pc/lab06/lab6-3.asm *
;-----
; Программа вычисления выражения
;-----
%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 ; 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. 3.10: меняем выражение

```

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

```

Рис. 3.11: результат

рассмотрим программу вычисления варианта задания по номеру студенческого билета (рис. 3.13)

```

;-----
; Программа вычисления 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 sprintf
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. 3.12: пишем программу

```

manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ nasm -f
elf variant.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ld -m el
f_i386 -o variant variant.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ./varian
t
Введите № студенческого билета:
1132232888
Ваш вариант: 9
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ █

```

Рис. 3.13: результат

3.2 Ответы на вопросы

1. Строка “moveax,rem” 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 ecx, 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`. Строка `"xoredx.edx"` обнуляет регистр `edx` перед выполнением деления. Строка `"movebx,20"` загружает значение 20 в регистр `ebx`. Строка `"divebx"` выполняет деление регистра `eax` на значение регистра `ebx` с сохранением частного в регистре `eax` и остатка в регистре `edx`,
 4. Остаток от деления записывается в регистр `edx`.
 5. Инструкция `"inc edx"` используется для увеличения значения в регистре `edx` на
 6. В данном случае, она увеличивает остаток от деления на 1. 13
 7. Строка `"mov eax,edx"` передает значение остатка от деления в регистр `eax`.
- 36 Строка `"call iprintLF"` вызывает процедуру `iprintLF` для вывода значения на экран вместе с переводом строки.

3.3 Задание для самостоятельной работы

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. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 (рис. 3.17)

```

GNU nano 6.2 /home/manovtchikov/work/arch-pc/lab06/samzadanie.asm *
; -----
; Программа вычисления выражения
; -----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
result_msg: DB 'Результат: ', 0
remainder_msg: DB 'Остаток от деления: ', 0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax, 3 ; x = 3
mov ebx, 31 ; умножение на 31
mul ebx ; EAX = EAX * EBX
sub eax, 5 ; вычитание 5
add eax, 10 ; прибавление 10
mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, result_msg ; вызов подпрограммы печати
call sprint ; сообщение 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprintf ; из 'edi' в виде символов
; ---- Вывод остатка на экран (в данном случае нет деления, остаток всегда 0)
mov eax, remainder_msg ; вызов подпрограммы печати
call sprint ; сообщение 'Остаток от деления: 0'
call quit ; вызов подпрограммы завершения

```

Рис. 3.14: пишем программу

```

manovtchikov@manovtchikov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf samzadanie.asm
manovtchikov@manovtchikov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o samzadanie samzadanie.o
manovtchikov@manovtchikov-VirtualBox:~/work/arch-pc/lab06$ ./samzadanie
Результат: 98
Остаток от деления: manovtchikov@manovtchikov-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 3.15: результат

```

1 ; -----
2 ; Программа вычисления выражения
3 ; -----
4 %include 'in_out.asm' ; подключение внешнего файла
5 SECTION .data
6 result_msg: DB 'Результат: ', 0
7 remainder_msg: DB 'Остаток от деления: ', 0
8 SECTION .text
9 GLOBAL _start
10 _start:
11 ; ---- Вычисление выражения
12 mov eax, 1 ; x = 1
13 mov ebx, 31 ; умножение на 31
14 mul ebx ; EAX = EAX * EBX
15 sub eax, 5 ; вычитание 5
16 add eax, 10 ; прибавление 10
17 mov edi, eax ; запись результата вычисления в 'edi'
18 ; ---- Вывод результата на экран
19 mov eax, result_msg ; вызов подпрограммы печати
20 call sprint ; сообщение 'Результат: '
21 mov eax, edi ; вызов подпрограммы печати значения
22 call iprintLF ; из 'edi' в виде символов
23 ; ---- Вывод остатка на экран (в данном случае нет деления, остаток
    всегда 0)
24 mov eax, remainder_msg ; вызов подпрограммы печати
25 call sprint ; сообщение 'Остаток от деления: 0'
26 call quit ; вызов подпрограммы завершения

```

Рис. 3.16: пишем программу

```

Остаток от деления: manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ nasm -f elf sanzadanie.asm
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ld -m elf_i386 -o sanzadanie sanzadanie.o
manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$ ./sanzadanie
Результат: 36
Остаток от деления: manovichkov@manovichkov-VirtualBox:~/work/arch-pc/lab06$

```

Рис. 3.17: результат

4 Выводы

В работе были изучены арифметические операции в языке ассемблера NASM. Был рассмотрен синтаксис и были написаны и проанализированы программы на ассемблере, которые используют арифметические операции для решения различных задач.