

Лабораторная работа №6

Архитектура компьютера

Кучмар София Игоревна

Содержание

1	Цель работы	3
2	Задание	4
3	Выполнение лабораторной работы	5
4	Выводы	11

1 Цель работы

Эта работа направлена на освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Данная работа посвящена практическому освоению ассемблера NASM. Будут изучены основы работы адресацией в NASM, освоены арифметические операции в NASM, целочисленное сложение `add`, целочисленное вычитание `sub`, команды инкремента и декремента, команда изменения знака операнда `neg`, основными директивами ассемблера, команды умножения `mul` и `imul` и будет написана программа для вычисления выражений. Будет подключен внешний файл `in_out.asm` с функциями ввода и вывода данных.

3 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 6, перейдём в него и создадим файл lab6-1.asm(рис. 3.1).

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

Рис. 3.1: Создание каталога и файла в нём

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax (рис. 3.2).

```

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

```

Рис. 3.2: Вводим программу

Создадим исполняемый файл и запустим его (рис. 3.3).

```

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

```

Рис. 3.3: Запуск файла

Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (рис. 3.4).

```

lab6-1.asm      [-M--]  9 L:[ 1+
#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. 3.4: Изменение текст программы

Создадим исполняемый файл и запустим его (рис. 3.5)

```

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

```

Рис. 3.5: Запуск файла

Создадим файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы. Создадим исполняемый файл и запустим его (рис. 3.6).

```

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

```

Рис. 3.6: Запуск файла

Аналогично предыдущему примеру изменим символы на числа. Создадим исполняемый файл и запустим его (рис. 3.7).

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

Рис. 3.7: Запуск файла

Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы (рис. 3.8).

```
lab6-3.asm      [----] 19 L:[ 5+10  15/ 26] *(451
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
```

Рис. 3.8: Изменение файла lab5-3.asm

Создадим исполняемый файл и запустим его (рис. 3.9).

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

Рис. 3.9: Запуск файла

Изменим текст программы для вычисления выражения $f(x)=(4 \cdot 6 + 2)/5$. Создадим исполняемый файл и запустим его (рис. 3.10).


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

Рис. 3.10: Запуск файла

Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы/ Создадим исполняемый файл и запустим его (рис. 3.11).

```
sikuchmar@vbox:~/work/arch-pc/lab06$ nasm -f elf variant.asm
sikuchmar@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
sikuchmar@vbox:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132246765
Ваш вариант: 6
```

Рис. 3.11: Запуск файла

Заметим, что 1. Строки отвечающие за вывод на экран сообщения 'Ваш вариант: mov eax, rem call sprintLF 2. Инструкции mov ecx, x, mov edx, 80, call sread используются для чтения строки (в данном случае номера студенческого билета) из стандартного ввода. ecx указывает на буфер x, edx — на количество байтов для чтения, а sread считывает строку из стандартного ввода и сохраняет ее в буфере. 3. Инструкция "call atoi" используется для преобразования строки, хранящейся в x, в целое число. Преобразованное число сохраняется в регистре eax. 4. Строки, отвечающие за вычисления варианта: xor edx,edx mov ebx,20 div ebx inc edx 5. В регистр edx записывается остаток от деления при выполнении инструкции "div ebx" 6. Инструкция "inc edx" используется для увеличения остатка от деления на 1. Это необходимо, так как номера вариантов нумеруются с 1, а деление на 20 дает остаток от 0 до 19. 7. Строки, отвечающие за вывод на экран результата вычислений: mov eax,rem call sprint mov eax,edx call iprintLF

Создадим файл lab5-4.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы (рис. 3.12).

```

lab6-4.asm      [----] 41 L: [ 1+30 31/ 32] *(1258/1259b) 0010
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите значение переменной x для выражения x^3/2+1 : ',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
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'
mov ebx, eax ; ebx = x
mul ebx ; eax = x*x
mul ebx ; eax = x*x*x
mov edx, 0 ; Обнуляем EDX для div
mov ebx, 2
div ebx ; eax = x^3 / 2, edx = x^3 % 2
add eax, 1
mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, rem ; вызов подпрограммы печати
call sprintf ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprintf ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 3.12: Изменение файла lab5-4.asm

Создадим исполняемый файл и запустим его (рис. 3.13).

```

sikuchmar@vbox:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
sikuchmar@vbox:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
sikuchmar@vbox:~/work/arch-pc/lab06$ ./lab6-4
Введите значение переменной x для выражения x^3/2+1 :
2
Результат:
5
sikuchmar@vbox:~/work/arch-pc/lab06$ ./lab6-4
Введите значение переменной x для выражения x^3/2+1 :
5
Результат:
63

```

Рис. 3.13: Запуск файла

4 Выводы

В рамках данной работы были успешно освоены основы работы с ассемблером NASM. Были освоены арифметические операции в NASM и создание программ для вычисления выражений.