Отчёт по лабораторной работе №6

Арифметические операции в NASM

Лань Цяньин

1. Цель работы

Освоение арифметических инструкций языка ассемблера NASM

2. Порядок выполнения лабораторной работы

2.1. Символьные и численные данные в NASM

1.Создан каталог для программ лабораторной работы № 6, выполнен переход в него и создан файл `lab6-1.asm` с помощью команд `mkdir`, `cd` и `touch` (рис. Рисунок 1).

Рисунок 1: Создание каталога lab06 и файла lab6-1.asm

В результате был создан каталог `lab06`, в котором подготовлен файл `lab6-1.asm` для дальнейшей работы над программой.

2.В файл `lab6-1.asm` был введён текст программы из листинга 6.1 (рис. Рисунок 2).

Рисунок 2: Ввод текста программы lab6-1.asm

Дан пример работы с регистрами и подпрограммой `sprintLF`: после вычислений значение сохраняется в `buf1`, затем в `eax` загружается адрес `buf1` и выполняется вывод.

Файл `in_out.asm` был скопирован в каталог `lab06`, после чего выполнены сборка и запуск программы. (рис. Рисунок 3)

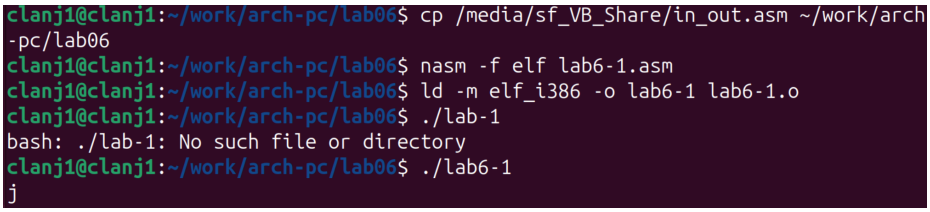

Рисунок 3: Сборка и запуск программы lab6-1

В результате на экране был выведен символ `j`, что связано с тем, что в программе выполняется сложение кодов символов `'6'` и `'4'`, а полученное значение соответствует символу `j` в таблице ASCII.

3.Текст программы `lab6-1.asm` был изменён: вместо символьных констант в регистры записаны числовые значения.

Строки

`mov eax, '6'` и `mov ebx, '4'`

заменены на

`mov eax, 6` и `mov ebx, 4` в соответствии с заданием.(рис. Рисунок 4)

Рисунок 4: Изменение текста программы lab6-1.asm

Программа была скомпилирована, слинкована и запущена командами `nasm`, `ld` и `./lab6-1` после замены символьных значений на числовые в файле `lab6-1.asm`. (рис. Рисунок 5)

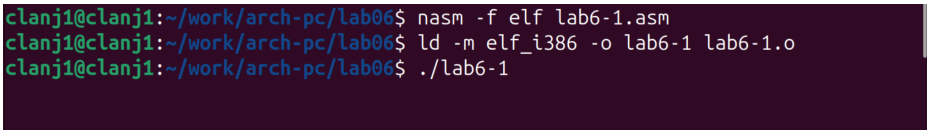

Рисунок 5: Сборка и запуск программы `lab6-1`

После запуска программы на экране не появляется видимого вывода, так как функция `sprintLF` интерпретирует значение в регистре `eax` как ASCII-код символа.

При значении `eax = 10` выводится управляющий символ с кодом 10 (перевод строки), поэтому отображаемый результат визуально отсутствует.

4.По заданию создан файл `lab6-2.asm` (листинг 6.2): в регистры `eax` и `ebx` записываются символы `'6'` и `'4'`, выполняется сложение `add eax, ebx`, после чего результат выводится подпрограммой `iprintLF`. (рис. Рисунок 6)

Рисунок 6: Редактирование файла `lab6-2.asm` (листинг 6.2)

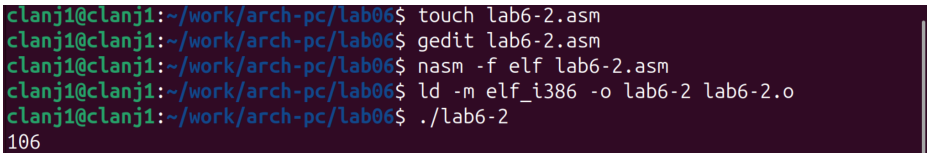

Рисунок 7: Сборка и запуск программы `lab6-2`, вывод результата

После запуска программа выводит `106`, потому что команда `add` складывает ASCII-коды символов `'6'` и `'4'` (54 + 52 = 106). Подпрограмма `iprintLF` выводит содержимое `eax` как десятичное число, поэтому на экране отображается именно `106`. (рис. Рисунок 7)

5.В файле `lab6-2.asm` символы заменены на числа, а `iprintLF` — на `iprint`: `mov eax, '6'` → `mov eax, 6`, `mov ebx, '4'` → `mov ebx, 4`; вызов `call iprint`. (рис. Рисунок 8)

Рисунок 8: Редактирование `lab6-2.asm`: числа и `iprint`

```
clanj1@clanj1:~/work/arch-pc/lab06$ gedit lab6-2.asm
clanj1@clanj1:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
clanj1@clanj1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
clanj1@clanj1:~/work/arch-pc/lab06$ ./lab6-2
10clanj1@clanj1:~/work/arch-pc/lab06$
```

Рисунок 9: Сборка, линковка и запуск `lab6-2`: вывод `10` без переноса строки

При выполнении программы выводится число `10`, так как складываются значения 6 и 4.

Функция `iprint` печатает число без перевода строки, тогда как `iprintLF` добавляет перенос строки. (рис. Рисунок 9)

2.2. Выполнение арифметических операций в NASM

6.Создан файл `lab6-3.asm` и введён текст программы из листинга 6.3 для вычисления выражения (рис. **?@fig-010**)

Open *lab6-3.asm ~/work/arch-pc/lab06 Save

```
1 %include 'in_out.asm'
2
3 SECTION .data
4 div: DB 'Результат: ',0
5 rem: DB 'Остаток от деления: ',0
6
7 SECTION .text
8 GLOBAL _start
9
10 _start:
11 mov eax,5
12 mov ebx,2
13 mul ebx
14 add eax,3
15 xor edx,edx
16 mov ebx,3
17 div ebx
18 mov edi,eax
19
20 mov eax,div
21 call sprint
22 mov eax,edi
23 call iprintLF
24 mov eax,rem
25 call sprint
26 mov eax,edx
27 call iprintLF
28
29 call quit
```

{#fig-

010-width=80%}

Программа была скомпилирована, слинкована и запущена командами `nasm`, `ld` и `./lab6-3`. (рис. Рисунок 10)

```
clanj1@clanj1:~/work/arch-pc/lab06$ touch lab6-3.asm
clanj1@clanj1:~/work/arch-pc/lab06$ gedit lab6-3.asm
clanj1@clanj1:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
clanj1@clanj1:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
clanj1@clanj1:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Рисунок 10: Сборка и запуск программы `lab6-3`: результат и остаток

В результате работы программы получено: `Результат: 4`, `Остаток от деления: 1`,
что соответствует вычислению (13 / 3 = 4) и остатку (1).

Для задания 6.3.2 был изменён текст программы для вычисления выражения
(f(x) = (4 □6 + 2)/5). В программе изменены константы:
`mov eax, 4`, `mov ebx, 6`, затем `mul ebx` и `add eax, 2`.
Перед делением выполняется `xor edx, edx`, после чего деление выполняется на 5: `mov ebx, 5` и `div ebx`.
После `div` частное находится в `eax`, остаток — в `edx`. (рис. Рисунок 11)

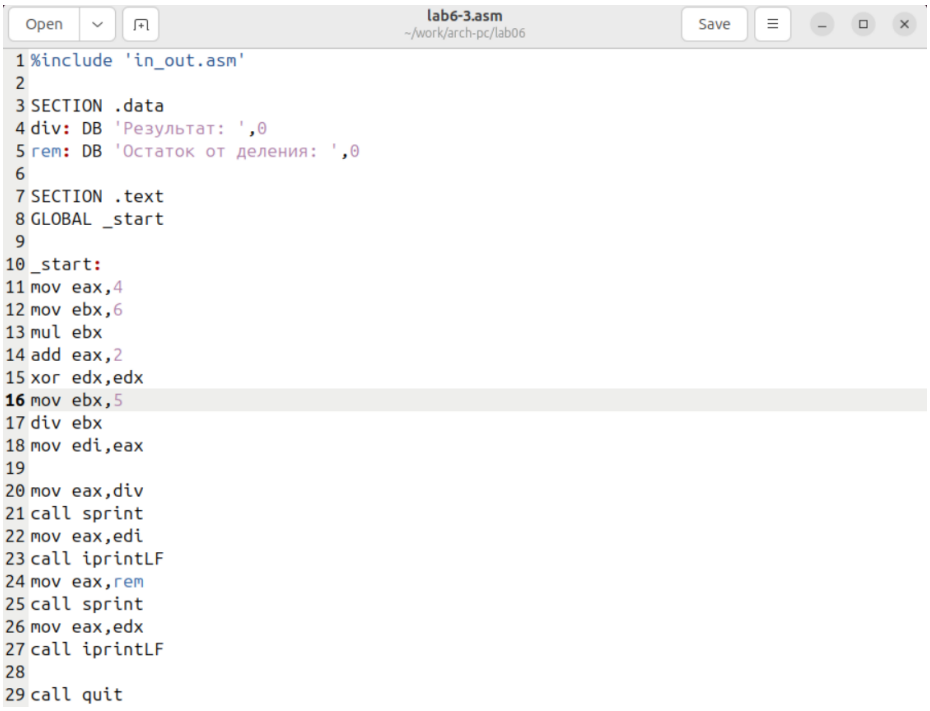

Рисунок 11: Изменённый текст программы `lab6-3.asm` для вычисления ((4□6+2)/5)

Программа была скомпилирована (nasm), слинкована (ld) и запущена.

В результате получено: `Результат: 5`, `Остаток от деления: 1`.

Это соответствует вычислению ((4□6+2)=26), (26/5 = 5) и остаток (1). (рис. Рисунок 12)

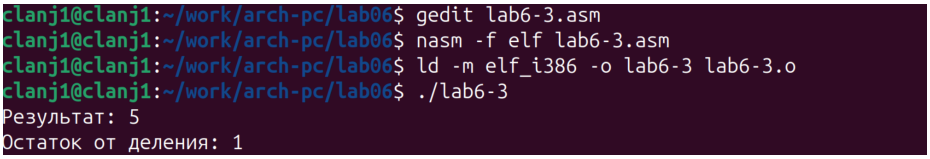

Рисунок 12: Сборка и запуск программы: вывод результата и остатка

По Листингу 6.4 реализована программа `variant.asm` для вычисления номера варианта по номеру студенческого билета по формуле (S_n mod 20) + 1 (рис. Рисунок 13).

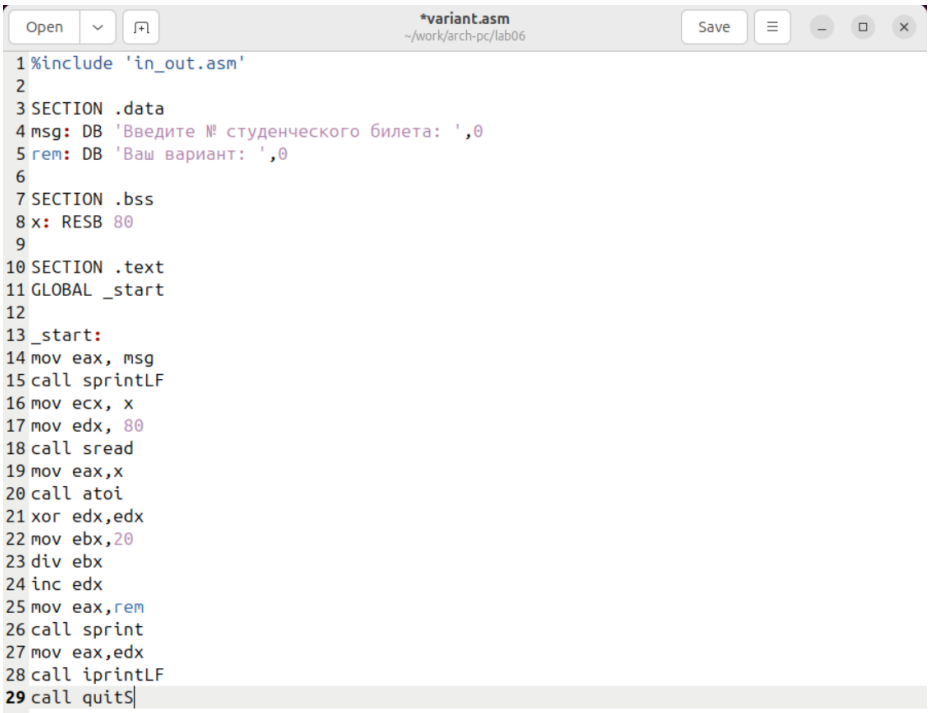

Рисунок 13: Исходный код программы `variant.asm` по Листингу 6.4

Программа была скомпилирована с помощью `nasm`, слинкована с помощью `ld` и запущена. В качестве входных данных был введён номер студенческого билета 1132254528. В результате работы программы на экран был выведен номер варианта: 9. (рис. Рисунок 14)

Рисунок 14: Сборка и запуск программы `variant`

Ответы на вопросы к листингу 6.4

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения «Ваш вариант:»? Ответ За вывод сообщения «Ваш вариант:» отвечают строки,

в которых в регистр eax загружается адрес строки с текстом сообщения, после чего вызывается подпрограмма вывода sprint (или sprintLF

2. Для чего используются следующие инструкции? `mov ecx, x` `mov edx, 80` `call sread` Ответ Данные инструкции используются для чтения строки с клавиатуры: ecx содержит адрес буфера x, edx задаёт максимальный размер вводимой строки, подпрограмма sread выполняет ввод данных.

3. Для чего используется инструкция call atoi? Ответ Инструкция call atoi используется для преобразования введённой строки, содержащей число в текстовом виде, в целочисленное значение.

4. Какие строки листинга 6.4 отвечают за вычисление варианта? Ответ За вычисление номера варианта отвечают строки, содержащие арифметические операции, включая инструкцию деления div, а также подготовку регистров перед выполнением деления

5. В какой регистр записывается остаток от деления при выполнении инструкции div ebx? Ответ Остаток от деления при выполнении инструкции div ebx записывается в регистр edx.

6. Для чего используется инструкция inc edx? Ответ Инструкция inc edx используется для увеличения значения регистра edx на единицу, что позволяет скорректировать номер варианта в соответствии с заданным алгоритмом.

7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений? Ответ За вывод результата вычислений отвечают строки, в которых значение варианта преобразуется в строку и выводится на экран с помощью подпрограммы sprint (или sprintLF).

3. Задание для самостоятельной работы

По образцу из Листингов 6.3–6.4 (ввод x через `sread` + `atoi`) написана программа `lab6-ind.asm`, которая вычисляет
вариант 9 из табл. 6.3: y = 10 + (31x – 5), и выводит результат.

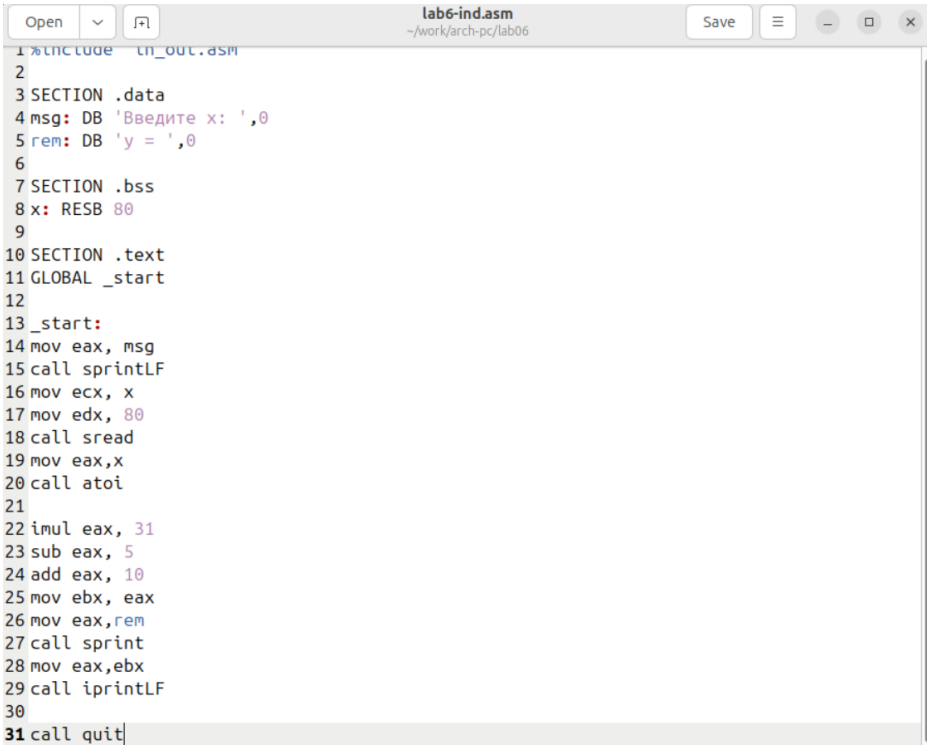

Рисунок 1: Код программы `lab6-ind.asm` (ввод x, вычисление y)

Рисунок 2: Запуск программы и проверка для x=3 и x=1

Проверка: при x=3 получено y=98, при x=1 получено y=36 — соответствует формуле варианта 9.

Выводы

В ходе выполнения лабораторной работы №6 были изучены арифметические инструкции языка ассемблера NASM и принципы работы с числовыми данными. Была реализована программа вычисления номера варианта и программа вычисления заданного арифметического выражения. Цель лабораторной работы достигнута.