

Отчёт по лабораторной работе №6

Цатурьян Лев Вячеславович НММбд-03-23

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Задание для самостоятельной работы	10
4	Выводы	13

Список иллюстраций

2.1	Создание каталога lab06, переход в него, создание в нём файла . .	5
2.2	Введенный текст листинга 6.1 в Midnight commander	5
2.3	Создание объектного и исполняемого файлов	5
2.4	Запуск исполняемого файла	6
2.5	Изменённый текст программы	6
2.6	Создание объектного и исполняемого файлов. Запуск программы	7
2.7	Создание файла lab6-2 с помощью копирования (F5)	7
2.8	Создание объектного и исполняемого файлов. Запуск программы	7
2.9	Создание объектного и исполняемого файлов. Запуск программы	8
2.10	Создание объектного и исполняемого файлов. Запуск программы	8
2.11	Изменение текста программы для вычисления значения функции $f(\boxtimes) = (4 \boxtimes 6 + 2)/5$	8
2.12	Создание объектного и исполняемого файлов. Запуск программы	9
2.13	Создание объектного и исполняемого файлов. Запуск программы	9
3.1	Создание файла u.asm	10
3.2	Создание объектного и исполняемого файлов. Запуск программы	11
3.3	Проверка работы программы подстановкой. Все работает правильно	12

1 Цель работы

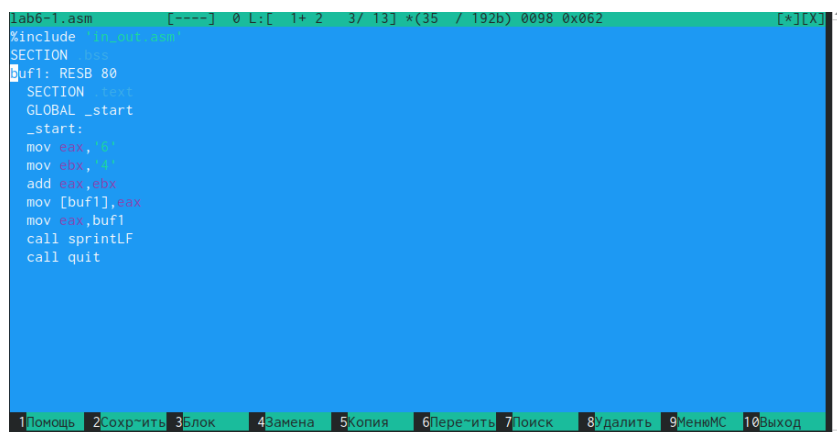
Получение навыков работы с арифметическими инструкциями языка ассемблера NASM.

2 Выполнение лабораторной работы

```
lvcaturnjyan@dk6n52 ~/work/arch-pc $ mkdir lab06
lvcaturnjyan@dk6n52 ~/work/arch-pc $ cd lab06
lvcaturnjyan@dk6n52 ~/work/arch-pc/lab06 $ touch lab6-1.asm
```

Рис. 2.1: Создание каталога lab06, переход в него, создание в нём файла

Далее я ввёл в созданный файл текст листинга 6.1



```
lab6-1.asm  [-----]  0 L: [ 1+ 2 3/ 13] *(35 / 192b) 0098 0x062  [*][X] ~
%include "linout.asm"
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, 0
mov ebx, 4
add eax, ebx
mov [buf1], eax
mov eax, buf1
call sprintf
call quit
```

Рис. 2.2: Введенный текст листинга 6.1 в Midnight commander

```
lvcaturnjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
lvcaturnjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
lvcaturnjyan@dk6n52 ~/work/arch-pc/lab06 $
```

Рис. 2.3: Создание объектного и исполняемого файлов

```
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./lab6-1
j
```

Рис. 2.4: Запуск исполняемого файла

На экран вывелся символ j, так как команда add записала в eax сумму двоичных кодов цифр 6 и 4, что в свою очередь является кодом символа j

```
%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. 2.5: Изменённый текст программы

Вместо символов мы уже записали числа. Теперь создадим объектный и исполняемый файлы и запустим программу

```
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./lab6-1
```

Рис. 2.6: Создание объектного и исполняемого файлов. Запуск программы

На экран ничего не вывелось, это произошло потому, что код элемента 10 соответствует символу перевода строки

Копирование

Копировать файл "lab6-2.asm" с исходным шаблоном:

[x] Метасимволы shell

В:

an/work/arch-pc/lab06/lab6-2.asm

☐ Разыменовывать ссылки
 ☐ Внутрь подкаталога, если есть

☒ Сохранять атрибуты
 ☐ Изменять относительные ссылки

Рис. 2.7: Создание файла lab6-2 с помощью копирования (F5)

В этот файл я вставил текст из листинга 6.2

```
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./lab6-2
106
```

Вывелось число 106

из-за функции `iprintLF`

Далее я изменил символы на числа в тексте этого файла

```
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./lab6-2
10
```

Рис. 2.8: Создание объектного и исполняемого файлов. Запуск программы

Наконец вывелось число 10, это произошло потому, что программа теперь считает цифры 6 и 4 за числа, а не за символы

Далее я изменил команду `iprintLF` на `iprint`, теперь результат не должен выводиться на следующей строке

```

lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./lab6-2
10lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $

```

Рис. 2.9: Создание объектного и исполняемого файлов. Запуск программы

Ожидания подтвердились, ответ выводится на той же строке

Далее я создал файл lab6-3 и вставил в него текст из листинга 6.3

```

lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 2.10: Создание объектного и исполняемого файлов. Запуск программы

Ответ верен, программа работает корректно

После этого я изменил текст программы для вычисления выражения $f(\boxtimes) = (4 \boxtimes 6 + 2)/5$.

```

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'

```

Рис. 2.11: Изменение текста программы для вычисления значения функции $f(\boxtimes)$

$$= (4 \boxtimes 6 + 2)/5$$


```

lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 2.12: Создание объектного и исполняемого файлов. Запуск программы

Результат верный, программа работает правильно

Далее я создал файл variant.asm, чтобы выполнить следующее задание, и вставил в него текст из листинга 6.4

```

lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ touch variant.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ mc

lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1132236086
Ваш вариант: 7

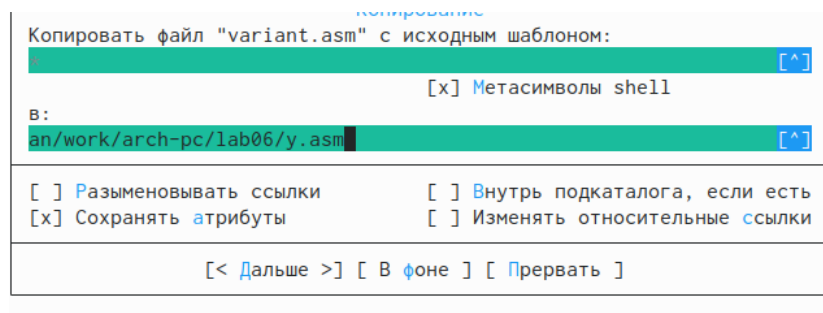
```

Рис. 2.13: Создание объектного и исполняемого файлов. Запуск программы

На запрос ввёл номер своего студенческого билета, после этого вычислил номер вручную, мой ответ совпал с результатом программы. Мой вариант: 7

Ответы на вопросы: 1) За вывод строки Ваш вариант отвечают строки: mov eax,rem call sprint 2) mov ecx,x используется для записи адреса вводимой строки x в регистр ecx, mov edx,80 запись в регистр edx длины вводимой строки call spread вызов подпрограммы для ввода значений с клавиатуры 3) call atoi - вызов подпрограммы из внешнего файла, преобразующей ascii код символа в число, и записывает его в eax 4) xor edx,edx mov ebx,20 div ebx inc edx эти строки отвечают за вычисление варианта 5) При выполнении div ebx остаток от деления записывается в edx 6) Инструкция inc edx увеличивает значение в регистре edx на 1 7) За вывод на экран результата вычисления отвечают строки mov eax,edx call iprintLF

3 Задание для самостоятельной работы



Копировать файл "variant.asm" с исходным шаблоном: [^]

[x] Метасимволы shell

В: an/work/arch-pc/lab06/y.asm [^]

<input type="checkbox"/> Разыменовывать ссылки	<input type="checkbox"/> Внутрь подкаталога, если есть
<input checked="" type="checkbox"/> Сохранять атрибуты	<input type="checkbox"/> Изменять относительные ссылки

[< Дальше >] [В фоне] [Прервать]

Рис. 3.1: Создание файла y.asm

Мой вариант 7, поэтому я пишу программу для вычисления функции $y=5(x-1)^2$

```

y.asm [-----] 6 L: [ 1+ 4 5/ 28
#include 'in_out.asm'.
SECTION .data
msg: DB 'y=5(x-1)^2. Введите x: ',0
rem: DB 'Ответ: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi

dec eax
mul eax
mov ebx, 5
mul ebx
mov edi, eax

mov eax, rem
call sprint
mov eax, edi
call iprintLF
call quit

```

dec eax - уменьшение

значения в eax(регистр с переменной) на 1 mul eax - умножение значения в eax на себя же (возведение в квадрат) mov ebx,5 - запись числа 5 в ebx mul ebx - умножение на 5 mov edi,eax - запись результата в edi

```

lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ nasm -f elf y.asm
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o y y.o
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./y

```

Рис. 3.2: Создание объектного и исполняемого файлов. Запуск программы

```
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./y
y=5(x-1)^2. Введите x: 3
Ответ: 20
lvcaturjyan@dk6n52 ~/work/arch-pc/lab06 $ ./y
y=5(x-1)^2. Введите x: 5
Ответ: 80
```

Рис. 3.3: Проверка работы программы подстановкой. Все работает правильно

4 Выводы

Я освоил арифметические инструкции языка ассемблера NASM