

Лабораторная работа №7

Архитектура компьютера

Кирилюк Светлана Алексеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Ввод текста программы	6
2.3	Запуск программы	6
2.4	Изменение текста программы	7
2.5	Запуск программы	7
2.6	Создание второго файла	8
2.7	Ввод текста программы	8
2.8	Запуск программы	8
2.9	Изменение текста программы	8
2.10	Запуск программы	9
2.11	Запуск программы с изменённой функцией	9
2.12	Создание третьего файла	9
2.13	Ввод текста программы	10
2.14	Запуск третьего файла	10
2.15	Изменение текста программы	11
2.16	Запуск изменённого файла	11
2.17	Создание нового файла	11
2.18	Ввод текста программы	12
2.19	Создание исполняемого файла, его запуск	12
2.20	Ввод текста программы	13
2.21	Ввод текста программы	14
2.22	Ввод текста программы	14

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Сначала я создала каталог для программ лабораторной работы №7, перешла в него и создала файл lab7-1.asm (рис. 2.1).

```
sakirilyuk@dk5n55 ~ $ mkdir ~/work/arch-pc/lab07
sakirilyuk@dk5n55 ~ $ cd ~/work/arch-pc/lab07
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ touch lab7-1.asm
```

Рис. 2.1: Создание каталога и файла

Затем я ввела текст программы из листинга 7.1 (рис. 2.2) и запустила файл (рис. 2.3).

```
%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. 2.2: Ввод текста программы

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./lab7-1
j
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.3: Запуск программы

Далее я поменяла текст программы, записав место символов в регистры числа (рис. 2.4), создала исполняемый файл и запустила его (рис. 2.5). 10 - символ перевода строки, поэтому при выводе на экран он не отображается.

```
%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. 2.4: Изменение текста программы

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./lab7-1

sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.5: Запуск программы

Создала новый файл lab7-2.asm (рис. 2.6), ввела в него текст программы из второго листинга (рис. 2.7) и запустила программу (рис. 2.8), создав исполняемый файл.

```
sakirilyuk@dk5n55 ~ $ cd ~/work/arch-pc/lab07
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07/lab7-2.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.6: Создание второго файла

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

    mov eax, '6'
    mov ebx, '4'
    add eax, ebx
    call iprintLF

    call quit
```

Рис. 2.7: Ввод текста программы

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./lab7-2
106
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.8: Запуск программы

Аналогично я изменила текст второго файла (рис. 2.9) и запустила его (рис. 2.10), создав исполняемый файл.

```
%include 'in_out.asm'

SECTION .text
GLOBAL _start
_start:

    mov eax, 6
    mov ebx, 4
    add eax, ebx
    call iprintLF

    call quit
```

Рис. 2.9: Изменение текста программы


```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./lab7-2
10
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.10: Запуск программы

В том же файле, в тексте программы, я заменила функцию `iprintLF` на `iprint`, создала исполняемый файл и запустила его (рис. 2.11).

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./lab7-2
10sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.11: Запуск программы с изменённой функцией

Далее я создала третий файл (рис. 2.12) и ввела туда текст программы из листинга 7.3 (рис. 2.13).

```
sakirilyuk@dk5n55 ~ $ cd ~/work/arch-pc/lab07
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07/lab7-3.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.12: Создание третьего файла

```
%include 'in_out.asm'

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax
```

Рис. 2.13: Ввод текста программы

Я вновь создала исполняемый файл и запустила его (рис. 2.14), затем изменила текст программы для вычисления выражения $f(x)=(4*6+2)/5$ (рис. 2.15) запустила исполняемый файл (рис. fig. 2.16).

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./lab7-3
Результат: 4
Остаток от деления: 1
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.14: Запуск третьего файла

```
%include 'in_out.asm'

SECTION .data

div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0

SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax
```

[^]G Справка [^]O Записать [^]W Поиск [^]K Вырезать [^]T Выполнить M-U Отмена
[^]X Выход [^]R ЧитФайл [^]\ Замена [^]U Вставить [^]C Позиция M-E Повтор

Рис. 2.15: Изменение текста программы

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./lab7-3
Результат: 5
Остаток от деления: 1
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.16: Запуск изменённого файла

Я создала новый файл variant.asm (рис. 2.17), ввела в него текст программы из листинга 7.4 (рис. 2.18), создала исполняемый файл и запустила его (рис. 2.19).

```
sakirilyuk@dk5n55 ~ $ cd ~/work/arch-pc/lab07
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ touch ~/work/arch-pc/lab07/variant.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.17: Создание нового файла

```
%include 'in_out.asm

SECTION .data
msg: DB 'Введите No студенческого билета: ',0
rem: DB 'Ваш вариант: ',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
```

Рис. 2.18: Ввод текста программы

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf variant.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o variant variant.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./variant
Введите No студенческого билета:
1132226471
Ваш вариант: 12
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

Рис. 2.19: Создание исполняемого файла, его запуск

#Ответы на вопросы

- 1)За вывод на экран сообщений отвечают такие строки листинга, как: `mov, eax, rem, call sprint`.
- 2)Инструкции `nasm mov ecx, x mov edx, 80 call sread` используются для ввода переменной X с клавиатуры и сохранения введенных данных.
- 3)Инструкция “`call atoi`” используется для преобразования кода переменной ASCII в число.
- 4)За вычисления варианта отвечают `xor edx, edx; mov ebx, 20 div ebx inc edx`.
- 5)Остаток от деления при выполнении инструкции “`div ebx`” записывается в регистр `edx`.

6) Инструкция “inc edx” используется для увеличения значения edx на 1.

7) За вывод на экран результата вычислений отвечают mov eax, edx call iprintLF.

#Выполнение задания для самостоятельной работы

Мне достался пример из варианта №12. Для начала я создала новый файл 12.asm, затем ввела текст программы (рис. 2.20), (рис. 2.21), (рис. 2.22), создала исполняемый файл и запустила его (рис. ??).



```
%include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
f: DB '(8*x-6)/2',0
x1: DB 'Введите значение x: ',
otv1: DB 'Ответ при x= ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,f
call sprintLF

mov eax,x1
```

[Прочитано 49 строк]

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	M-U Отмена
^X Выход	^R ЧитФайл	^V Замена	^U Вставить	^C Позиция	M-E Повтор

Рис. 2.20: Ввод текста программы

```

mov ecx,x
mov edx,80
call sread

mov eax,x
call atoi
mov ebx,8
mul ebx
add eax,-6
xor edx, edx
mov ebx,2
div ebx

mov edi, eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

```

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	M-U Отмена
^X Выход	^R ЧитФайл	^V Замена	^U Вставить	^C Позиция	M-E Повтор

Рис. 2.21: Ввод текста программы

```

add eax,-6
xor edx, edx
mov ebx,2
div ebx

mov edi, eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit

```

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	M-U Отмена
^X Выход	^R ЧитФайл	^V Замена	^U Вставить	^C Позиция	M-E Повтор

Рис. 2.22: Ввод текста программы

```
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf 12.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o 12 12.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./12
(8*x-6)/2
Введите значение x: Ответ при x=
1
Результат: 1
Остаток от деления: 0
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ nasm -f elf 12.asm
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o 12 12.o
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $ ./12
(8*x-6)/2
Введите значение x: Ответ при x=
5
Результат: 17
Остаток от деления: 0
sakirilyuk@dk5n55 ~/work/arch-pc/lab07 $
```

3 Выводы

В ходе лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.