

Лабораторная работа №6

Архитектура вычислительных систем

Басманова Дарья Кириллова

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Самостоятельная работа	14
4	Выводы	16

Список иллюстраций

2.1	1	5
2.2	2	5
2.3	3	6
2.4	4	6
2.5	5	7
2.6	6	7
2.7	7	7
2.8	8	8
2.9	9	8
2.10	10	9
2.11	11	9
2.12	12	9
2.13	13	10
2.14	14	10
2.15	15	11
2.16	16	11
2.17	18	12
2.18	19	13
3.1	20	15
3.2	21	15

1 Цель работы

Приобретение практических навыков по разработке небольших командных файлов. Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.1)

- 1) Создадим каталог для программ лабораторной работы №6, и перейдем в него и создадим файл lab6-1.asm:

```
dkbasmanova@dk2n25 ~ $ mkdir ~/work/arch-pc/lab06
dkbasmanova@dk2n25 ~ $ cd ~/work/arch-pc/lab06
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ touch lab6-1.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ls
lab6-1.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.1: 1

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.2)

- 2) Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. Введем в файл lab6-1.asm текст программы из листинга 7.1.

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 2.2: 2

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.3) 3) Создайте исполняемый файл и запустите его.

```
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-1
j
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.3: 3

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.4)

- 3) Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (Листинг 1) следующим образом.

```
%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: 4

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.5)

```
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-1

dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.5: 5

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.6) 4) Создадим файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введем в него текст программы из листинга 7.

<- ~/work/arch-pc/lab06			. [^]>			<- ~/work/arch-pc/lab06		
Имя	Размер	Время правки	Имя	Размер	Время правки	Имя	Размер	Время правки
./..	-ВВЕРХ-	дек 19 10:59	./..	-ВВЕРХ-	дек 19 10:59	./..	-ВВЕРХ-	дек 19 10:59
in_out.asm	3942	дек 19 10:59	in_out.asm	3942	дек 19 10:59	in_out.asm	3942	дек 19 10:59
lab6-1	2100	дек 19 11:00	lab6-1	2100	дек 19 11:00	lab6-1	2100	дек 19 11:00
lab6-1.asm	178	дек 19 11:00	lab6-1.asm	178	дек 19 11:00	lab6-1.asm	178	дек 19 11:00
lab6-1.o	1200	дек 19 11:06	lab6-1.o	1200	дек 19 11:06	lab6-1.o	1200	дек 19 11:06
lab6-2.asm	118	дек 19 11:43	lab6-2.asm	118	дек 19 11:43	lab6-2.asm	118	дек 19 11:43

Рис. 2.6: 6

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.7) 5) Создадим исполняемый файл и запустим его

```
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-2
106
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.7: 7

В результате работы программы мы получим число 106. В данном случае, как и в первом, команда add складывает коды символов '6' и '4' ($54+52=106$).

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.8)

- 6) Аналогично предыдущему примеру изменим символы на числа. Заменяем строки аналогично предыдущему.

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 2.8: 8

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.9)

```
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-2
10
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.9: 9

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.10)

- 7) Заменяем функцию `iprintLF` на `iprint`. Создадим исполняемый файл и запустим его. `iprintLF` и `iprint` выводят ответ на разных строках.


```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 2.10: 10

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.11)

```
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-2
10dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.11: 11

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.12)

8) Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06:

```
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ cd
dkbasmanova@dk2n25 ~ $ touch ~/work/arch-pc/lab06/lab6-3.asm
dkbasmanova@dk2n25 ~ $ cd ~/work/arch-pc/lab06
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.12: 12

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.13)

9) Введем в lab6-3.asm. текст из листинга и запустим его.

```

; Програма 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. 2.13: 13

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.14)

```

dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $

```

Рис. 2.14: 14

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.15)

- 10) Изменим текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$.
Создадим исполняемый файл и проверьте его работу.

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/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 iprintf ; 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 iprintf ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 2.15: 15

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.16)

```

dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $

```

Рис. 2.16: 16

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.17)

- 11) В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:
 - вывести запрос на введение № студенческого билета
 - вычислить номер варианта по формуле: $(S_n \bmod 20) + 1$, где S_n – номер студенческого билета (В данном случае $a \bmod b$ – это остаток от деления a на b).
 - вывести

на экран номер варианта. В данном случае число, над которым необходимо проводить арифметические операции, вводится с клавиатуры. Как отмечалось выше ввод с клавиатуры осуществляется в символьном виде и для корректной работы арифметических операций в NASM символы необходимо преобразовать в числа. Для этого может быть использована функция `atoi` из файла `in_out.asm`. Создадим файл `variant.asm` в каталоге `~/work/arch-pc/lab06`: `touch ~/work/arch-pc/lab06/variant.asm`

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/work/arch-pc/lab06/variant.asm
;-----
; 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 sprintf
mov eax, edx
call iprintf
call quit

```

Прочитано 28 строк

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

Рис. 2.17: 18

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 2.18)

```
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1032220040
Ваш вариант: 1
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $
```

Рис. 2.18: 19

3 Самостоятельная работа

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.1)

- 1) 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. Вид функции $f(x)$ выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создадим исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3.

1 вариант: $(10 + 2x)/3$ $x_1=1$ $x_2=10$

```

; Программа вычисления варианта
;-----
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите x ',0
div: DB 'Результат ',0
SECTION .bss
x: RESB 80
y: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call read
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
mov ebx, 2
mul ebx
add eax, 10
xor edx, edx
mov ebx, 3
div ebx

mov [y], eax

mov eax, div
call sprintf
mov eax, [y]
call iprintf
call quit

```

Рис. 3.1: 20

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.2)

```

dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ nasm -f elf variant1.asm
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant1 variant1.o
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./variant1
Введите x
1
Результат 4
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $ ./variant1
Введите x
10
Результат 10
dkbasmanova@dk2n25 ~/work/arch-pc/lab06 $

```

Рис. 3.2: 21

4 Выводы

В ходе выполнения лабораторной работы №4 я приобрела практические навыки по разработке командных файлов и освоила арифметические инструкции ассемблера.