

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

дисциплина: Архитектура компьютера

Холопов Илья Алексеевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13

Список иллюстраций

2.1	Создание каталога и файла в нем lab6-1.asm	6
2.2	Содержимое файла lab6-1.asm	7
2.3	Создание и запуск исполняемого файла	7
2.4	Результат работы измененной программы	7
2.5	Использование подпрограмм из in_out.asm	8
2.6	Создание и запуск исполняемого файла	8
2.7	Результат работы программы с числами вместо символов	8
2.8	Работа программы с iprint вместо iprintLF	9
2.9	Содержимое файла lab6-3.asm	9
2.10	Создание и запуск исполняемого файла	9
2.11	Создание и запуск исполняемого файла для вычисления выражения	10
2.12	Содержимое файла variant.asm	10
2.13	Создание и запуск исполняемого файла, вычисляющего номер ва- рианта	10
2.14	Содержимое файла программы для вычисления значения у	12
2.15	Создание и запуск исполняемого файла, вычисляющего значение у	12

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Создадим каталог для лабораторной работы № 6, перейдем в него и создадим файл lab6-1.asm (рис. 2.1).

```
ikhloпов0323@ikhloпов0323:~$ mkdir ~/work/arch-pc/lab06
ikhloпов0323@ikhloпов0323:~$ cd ~/work/arch-pc/lab06
ikhloпов0323@ikhloпов0323:~/work/arch-pc/lab06$ touch lab6-1.asm
ikhloпов0323@ikhloпов0323:~/work/arch-pc/lab06$ ls
lab6-1.asm
ikhloпов0323@ikhloпов0323:~/work/arch-pc/lab06$
```

Рис. 2.1: Создание каталога и файла в нем lab6-1.asm

Введем в файл lab6-1.asm текст программы. В данной программе в регистр `eax` записывается символ 6 (`mov eax, '6'`), в регистр `ebx` символ 4 (`mov ebx, '4'`). Далее к значению в регистре `eax` прибавляем значение регистра `ebx` (`add eax, ebx`, результат сложения запишется в регистр `eax`). Далее выводим результат. Так как для работы функции `sprintf` в регистр `eax` должен быть записан адрес, необходимо использовать дополнительную переменную. Для этого запишем значение регистра `eax` в переменную `buf1` (`mov [buf1], eax`), а затем запишем адрес переменной `buf1` в регистр `eax` (`mov eax, buf1`) и вызовем функцию `sprintf` (рис. 2.2).

```
/home/ikhologov0323/work-arch-pc/lab06/lab6-1.asm 173/173 100%
#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. 2.2: Содержимое файла lab6-1.asm

Создадим исполняемый файл и запустим его (рис. 2.3).

```
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ./lab6-1
j
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$
```

Рис. 2.3: Создание и запуск исполняемого файла

Далее изменим текст программы, вместо символов запишем в регистры числа. Создадим и запустим исполняемый файл. В результате программа выводит символ с кодом 10, т.е символ перевода строки (рис. 2.4).

```
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ./lab6-1

ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$
```

Рис. 2.4: Результат работы измененной программы

Для работы с числами в файле in_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из lab6-1.asm с использованием этих функций. (рис. 2.5).

```
/home/ikhologov0323/work-arch-pc/lab06/lab6-2.asm 118/118 100%
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

1Помощь 2Раз-рн 3Выход 4Чех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход

Рис. 2.5: Использование подпрограмм из in_out.asm

Создадим исполняемый файл и запустим его (рис. 2.6).

```
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ./lab6-2
106
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$
```

Рис. 2.6: Создание и запуск исполняемого файла

Теперь изменим символы на числа. Создадим и запустим исполняемый файл. На рис. 2.7 видно, что в результате выводится на экран число 10.

```
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ./lab6-2
10
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$
```

Рис. 2.7: Результат работы программы с числами вместо символов

Заменяем функцию iprintLF на iprint. Создадим исполняемый файл и запустим его. На рис. 2.8 видно, что iprint в отличие от iprintLF не выводит символ перевода строки.


```

ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-
2.o
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ./lab6-2
10
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$

```

Рис. 2.8: Работа программы с `iprint` вместо `iprintLF`

Создадим файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`. (рис. 2.9).

```

/home/ikholopov0323/work/arch-pc/lab06/lab6-3.asm 1237/1237 100%
%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
1По-шь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Иск-ый 9Формат 10Выход

```

Рис. 2.9: Содержимое файла `lab6-3.asm`

Создадим исполняемый файл и запустим его (рис. 2.10).

```

ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-
3.o
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$

```

Рис. 2.10: Создание и запуск исполняемого файла

Изменим текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$.
Создадим исполняемый файл и проверим его работу (рис. 2.11).

```

ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$

```

Рис. 2.11: Создание и запуск исполняемого файла для вычисления выражения

Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06. Напишем в файле variant.asm код программы, вычисляющей вариант по номеру введенного с клавиатуры (рис. 2.12).

```

/home/ikholopov0323/work/arch-pc/lab06/variant.asm 492/492 100%
#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
1Поиск 2Разрешить 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат 10Выход

```

Рис. 2.12: Содержимое файла variant.asm

Создадим исполняемый файл и запустим его (рис. 2.13).

```

ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ nasm -f elf variant.asm
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
5
Ваш вариант: 6
ikholopov0323@ikholopov0323:~/work/arch-pc/lab06$

```

Рис. 2.13: Создание и запуск исполняемого файла, вычисляющего номер варианта

Следующие строки отвечают за вывод на экран сообщения ‘Ваш вариант:’:

```
mov eax, rem  
call sprint
```

Данные введенные с клавиатуры записываются в x:

```
mov ecx, x  
mov edx, 80  
call sread
```

Инструкция “call atoi” используется для перевода ASCII кода в число.

Следующие строки отвечают за вычисление варианта:

```
xor edx, edx  
mov ebx, 20  
div ebx  
inc edx
```

В регистр edx записывается остаток от деления при выполнении инструкции “div ebx”. Инструкция “inc edx” используется, так как необходимо вычислить остаток от деления на 20 и прибавить 1.

Следующие строки отвечают за вывод на экран результата вычислений:

```
mov eax, edx  
call iprintLF
```

Напишем программу для вычисления выражения $y = f(x) = (9x - 8)/8$ (рис. 2.14).

```

/home/ikhologov0323/work-arch-pc/lab06/task.asm 540/540 100%
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
res: DB 'y = ',0
rem: DB 'Остаток: ',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 ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
mov ebx, 9
mul ebx
sub eax, 8
xor edx, edx
mov ebx, 8
div ebx
mov ebx, eax
mov eax, res
call sprint
mov eax, ebx
call iprintLF
mov eax, rem
call sprint
mov eax, edx
call iprintLF
call quit
1Помощь 2Раз-рн 3Выход 4Hex 5Пер-ти 6 7Поиск 8Исх-ый 9Формат 10Выход

```

Рис. 2.14: Содержимое файла программы для вычисления значения у

Создадим исполняемый файл и запустим его (рис. 2.14).

```

ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ nasm -f elf task.asm
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ld -m elf_i386 -o task task.o
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ./task
Введите x: 8
y = 8
Остаток: 0
ikhologov0323@ikhologov0323:~/work/arch-pc/lab06$ ./task
Введите x: 64
y = 71
Остаток: 0

```

Рис. 2.15: Создание и запуск исполняемого файла, вычисляющего значение у

3 Выводы

В результате выполнения лабораторной работы были освоены арифметические инструкции языка ассемблера NASM, созданы программы для вычисления выражений.