

Отчет по лабораторной работе №6

Архитектура компьютера

Дмитрий Константинович Кобзев

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	14
5	Выводы	16
	Список литературы	17

Список иллюстраций

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

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. Создайте исполняемый файл и проверьте его работу для значений x_1 и x_2 из 6.3

3 Выполнение лабораторной работы I

[1–6]

Создаем каталог для программ лабораторной работы № 6, переходим в него

```
dkkobzev@dk6n54:~$ mkdir ~/work/arch-pc/lab06  
dkkobzev@dk6n54:~$ cd ~/work/arch-pc/lab06  
dkkobzev@dk6n54:~/work/arch-pc/lab06$ touch lab6-1.asm
```

и создаем файл lab6-1.asm:..

Вводим в файл lab6-1.asm текст программы из листинга 6.1 (рис. 1.2).

```
%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
```

Создаем исполняемый файл и запускаем его (рис. 1.3).

```
dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1.o lab6-1.o
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./lab6-1
```

Изменяем текст программы и вместо символов, записываем в регистры числа

```
%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. 1.4).

```
dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1.o lab6-1.o
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./lab6-1
```

Создаем исполняемый файл и запускаем его (рис. 1.5).

Создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы из листинга 6.2 (рис. 1.6), (рис. 1.7).

```
dkkobzev@dk6n54:~$ touch ~/work/arch-pc/lab06/lab6-2.asm
```



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

```
dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf_i
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./lab6-2
106
```

Создаем исполняемый файл и запускаем его (рис. 1.8).

Аналогично предыдущему примеру изменяем символы на числа. (рис. 1.9).

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

```

dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./lab6-2
10

```

Создаем исполняемый файл и запускаем его (рис. 1.10).

Заменяем функцию `iprintLF` на `iprint`. Создаем исполняемый файл и запускаем его (рис. 1.11).

```

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

```

```

dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./lab6-2
10dkkobzev@dk6n54:~/work/arch-pc/lab06$

```

Создаем файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06` (рис. 1.13).

```

dkkobzev@dk6n54:~$ touch ~/work/arch-pc/lab06/lab6-3.asm

```

Внимательно изучаем текст программы из листинга 6.3 и вводим в `lab6-3.asm`

```
%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. 1.14).

```
dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
```

Создаем исполняемый файл и запускаем его (рис. 1.15).

Изменяем текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$.

Создаем исполняемый файл и проверяем его работу (рис. 1.16), (рис. 1.17).

```

#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 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

```

```

dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Создаем файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 1.18).

```
dkkobzev@dk6n54:~$ touch ~/work/arch-pc/lab06/variant.asm
```

Внимательно изучаем текст программы из листинга 6.4 и вводим в variant.asm

```

#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 iprintLF
call quit

```

(рис. 1.19).

```

dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm -f elf
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -m elf_
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132231936
Ваш вариант: 17

```

Создаем исполняемый файл и запускаем его (рис. 1.20).

Ответы на вопросы: 1. `mov eax, rem` `call sprintf` 2. Используются для чтения строки ввода. 3. Инструкция `call atoi` используется для преобразования строки в число. 4. `mov eax, x` `call atoi` `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 результата деления на 20 на 1. 7. `mov eax, edx` `call iprintLF`

4 Самостоятельная работа

Задание 1. Пишем программу вычисления выражения $y = f(x)$. Программа выводит выражение для вычисления, выводит запрос на ввод значения x , вычисляет заданное выражение в зависимости от введенного x , выводит результат вычислений. Вид функции $f(x) = 18(x+1)/6$. Создаем исполняемый файл и проверьте его работу для значений 3 и 1 (рис. 2.1).

```

#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'x: ',0
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .bss
x: RESB 10
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,msg
call sprintLF
mov ecx,x
mov edx, 10
call sread
mov eax,x ; EAX=x
call atoi
mov ebx,18 ; EBX=18
add eax,1 ; EAX=EAX+1
mul ebx ; EAX=EAX*EBX
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,6 ; EBX=6
div ebx ; EAX=EAX/6, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div
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
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

```

dkkobzev@dk6n54:~/work/arch-pc/lab06$ nasm
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ld -
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./sr
x:
3
Результат: 12
Остаток от деления: 0
dkkobzev@dk6n54:~/work/arch-pc/lab06$ ./sr
x:
1
Результат: 6
Остаток от деления: 0

```

5 Выводы

В ходе выполнения лабораторной работы мною были освоены арифметические инструкции языка ассемблера NASM.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.