

Отчет по лабораторной работе №6

Мухин Тимофей Владимирович

Содержание

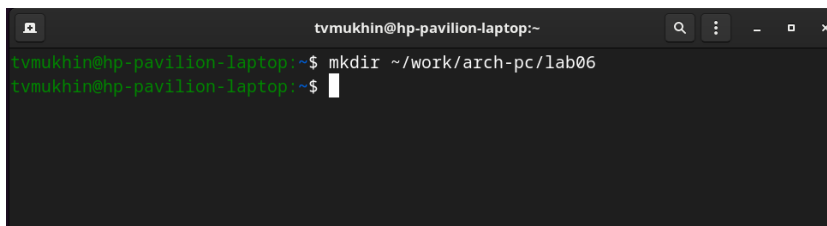
1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выводы	10

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

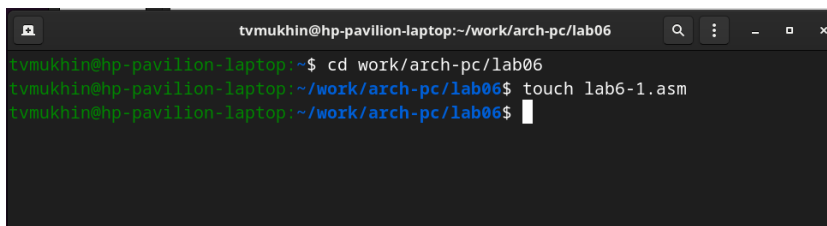
2 Выполнение лабораторной работы

1. Создаем каталог для программ лабораторной работы №6

A terminal window titled 'tvmukhin@hp-pavilion-laptop:~' showing the command 'mkdir ~/work/arch-pc/lab06' being executed. The prompt changes to 'tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06\$' after the command is run.

```
tvmukhin@hp-pavilion-laptop:~$ mkdir ~/work/arch-pc/lab06
tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$
```

2. Переходим в каталог и создаем файл lab6-1.asm

A terminal window titled 'tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06' showing two commands: 'cd work/arch-pc/lab06' and 'touch lab6-1.asm'. The prompt changes to 'tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06\$' after each command.

```
tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ cd work/arch-pc/lab06
tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ touch lab6-1.asm
tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$
```

3. Программа вывода значения регистра eax. Вводим текст программы из листинга 7.1.

```
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06 — nano lab6-1.asm
GNU nano 6.4 lab6-1.asm Modified
%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 printf
call quit

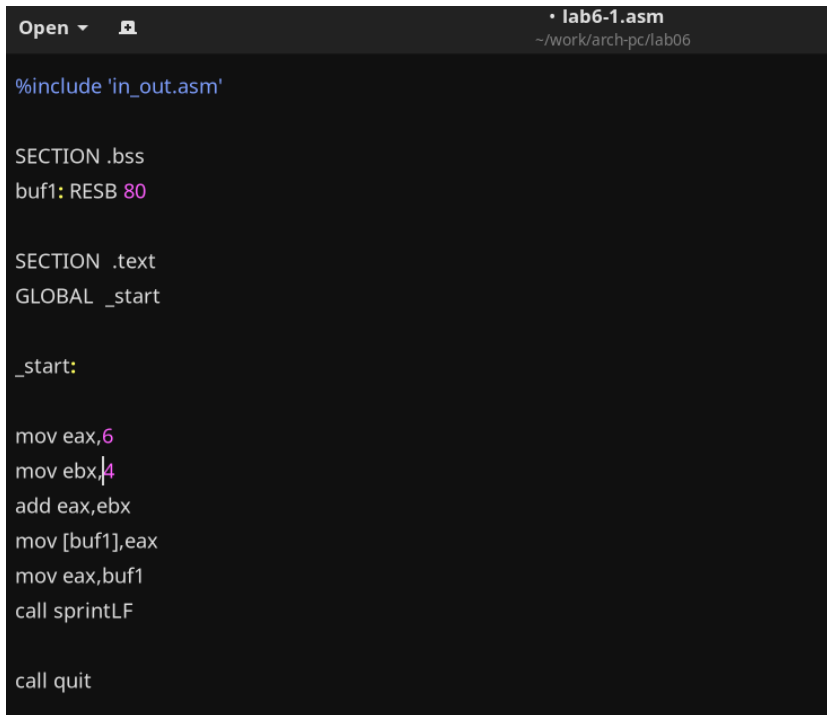
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

4. Создаем исполняемый файл и запускаем его. В данном случае при выводе значения регистра `eax` мы ожидаем увидеть число 10. Однако результатом будет символ `j`. Это происходит потому, что код символа 6 равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа 4 – 00110100 (52). Команда `add eax, ebx` запишет в регистр `eax` сумму кодов – 01101010 (106), что в свою очередь является кодом символа `j`.

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

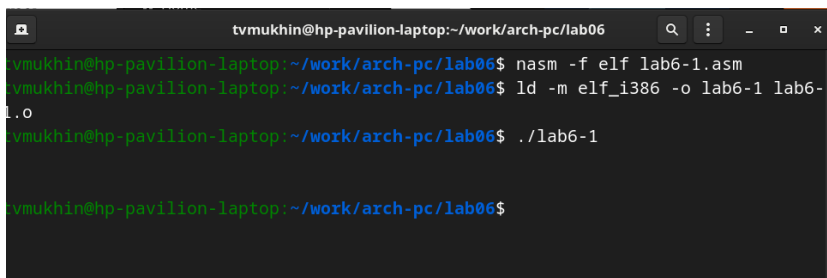
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ ./lab6-1
j
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$
```

5. Изменим текст программы и вместо символов, запишем в регистры числа.



```
Open ▾  
• lab6-1.asm  
~/work/arch-pc/lab06  
  
%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
```

6. Создаем исполняемый файл и запускаем его



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

7. Для работы с числами в файле in_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 7.1 с использованием этих функций. Создаем файл 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
```

8. В результате работы программы мы получим число 106. В данном случае, как и в первом, команда `add` складывает коды символов '6' и '4' ($54+52=106$). Однако, в отличие от программы из листинга 7.1, функция `iprintLF` позволяет вывести число, а не символ, кодом которого является это число.

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

9. Создаём исполняемый файл и проверяем его работу

```
tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ ./lab6-2
106
tvmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$
```

10. Аналогично предыдущему примеру изменим символы на числа. Проверяем работу программы.

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

11. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения. Создаем файл 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
mov ebx,3
div ebx
mov edi,eax
; ----

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

12. Создаем исполняемый файл и запускаем его.

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


13. Программа вычисления варианта задания по номеру студенческого билета. Программа работает по следующему алгоритму:
- 1) вывести запрос на введение № студенческого билета.
 - 2) вычислить номер варианта по формуле
 - 3) вывести на экран номер варианта.

```
%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
mov ebx,3
div ebx
mov edi,eax
; ----

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

14. Создаем исполняемый файл и проверяем правильность работы программы

```
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ nasm -f elf variant.asm
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant vari
ant.o
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132227119
Ваш вариант: 20
tvmmukhin@hp-pavilion-laptop:~/work/arch-pc/lab06$
```

3 Выводы

В ходе выполнения лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.