

Отчёт по лабораторной работе Архитектура компьютера

Лабораторная работа №6

Дмитрий Алексеевич Митяков

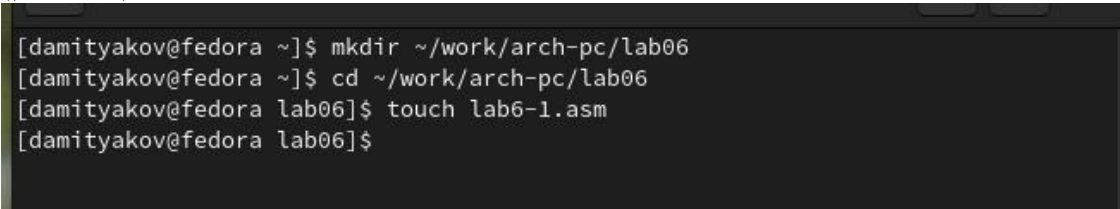
Содержание

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Я создал каталог для 6 лабораторной работы и в нем нужный файл (рис. ??).

A terminal window with a dark background and light gray text. It shows a series of commands being executed in a shell. The prompt is [damityakov@fedora ~]. The commands are: mkdir ~/work/arch-pc/lab06, cd ~/work/arch-pc/lab06, touch lab6-1.asm, and a final prompt [damityakov@fedora lab06]\$.

```
[damityakov@fedora ~]$ mkdir ~/work/arch-pc/lab06
[damityakov@fedora ~]$ cd ~/work/arch-pc/lab06
[damityakov@fedora lab06]$ touch lab6-1.asm
[damityakov@fedora lab06]$
```

В этот файл скопировал код и в директорию скопировал файл in_out

```

lab6-1.asm      [-----]  0 L:[  1+13  14/ 1
%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. ??).

(рис. ??).

Левая панель			Файл	Команда	Настройки	Правая панель			
< ~ /work/arch-pc/lab06					< ~ /work/arch-pc/lab05				
.м	Имя	Размер	Время правки		.и	Имя	Размер	Время правки	
/..			-ВВЕРХ-	дек 15 19:54	/..			-ВВЕРХ-	дек 15 19:54
in_out.asm			3942	дек 15 01:13	in_out.asm			3942	дек 15 01:13
lab6-1.asm			173	дек 15 19:56	*lab5-1			8744	дек 15 01:44
					lab5-1.asm			2431	дек 15 01:11
					lab5-1.o			752	дек 15 01:12
					*lab5-2			9092	дек 15 01:27
					*lab5-2-2			9092	дек 15 01:48
					lab5-2-2.asm			1223	дек 15 01:43
					lab5-2-2.o			1312	дек 15 01:48
					lab5-2.asm			1223	дек 15 01:43
					lab5-2.o			1312	дек 15 01:43
					*lab5-3			8744	дек 15 02:39
					lab5-3.asm			2731	дек 15 02:39
					lab5-3.o			784	дек 15 02:39
					*lab5-4			9092	дек 15 02:57
in_out.asm					in_out.asm				
948М / 14G (6%)					948М / 14G (6%)				

Совет: Для получения вывода команды в окне просмотра наберите M-!

Совет: Для получения вывода команды в окне просмотра наберите M-!

Создал исполняемый файл и получил такой результат (рис. ??).

```

[damityakov@fedora lab06]$ nasm -f elf lab6-1.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[damityakov@fedora lab06]$ ./lab6-1
j

```

Затем изменил в исходном файле вместо символов записал в

```
_start:
mov  eax,6
mov  ebx,4
add  eax,ebx
```

регистры числа (рис. ??). И получил такой результат - символ не отображается на экране (рис. ??).

```
[damityakov@fedora lab06]$ nasm -f elf lab6-1.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[damityakov@fedora lab06]$ ./lab6-1
```

```
[damityakov@fedora lab06]$
```

Создал следующий файл (рис. ??).

```
[damityakov@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-2.asm
```

И скопировал туда уже новый текст (рис. ??).

```
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
```

Исполняемый файл возвращает следующее значение (рис. ??).

```
[damityakov@fedora lab06]$ nasm -f elf lab6-2.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[damityakov@fedora lab06]$ ./lab6-2
106
[damityakov@fedora lab06]$
```

```
_start:
mov  eax,6
mov  ebx,4
add  eax,ebx
```

Заменяю символы на числа (рис. ??).

И

получил такой результат (рис. ??).

```
[damityakov@fedora lab06]$ nasm -f elf lab6-2.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[damityakov@fedora lab06]$ ./lab6-2
10
[damityakov@fedora lab06]$
```

Создал еще один файл и вставил в него такой код, и после запуска получил результат: (рис. ??).

```
lab6-3.asm      [----] 13 L:[ 1+ 1  2/ 26] *(88 /1236b) 0010 0x
#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
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Мен
```

(рис. ??).

```
[damityakov@fedora lab06]$ touch ~/work/arch-pc/lab06/lab6-3.asm
[damityakov@fedora lab06]$ nasm -f elf lab6-3.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[damityakov@fedora lab06]$ ./lab6-3
Результат: 4
Остаток от деления: 1
[damityakov@fedora lab06]$
```

Изменил следующие строки и получил нужный результат (рис. ??).

```

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

```

(рис. ??).

```

[damityakov@fedora lab06]$ nasm -f elf lab6-3.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[damityakov@fedora lab06]$ ./lab6-3
Результат: 5
Остаток от деления: 1
[damityakov@fedora lab06]$

```

Получил номер варианта с помощью программы: (рис. ??).

```

variant.asm      [-----]  9 L:[  1+ 0   1/ 28] *(9   / 6.
;-----
; 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 sprintLF
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

```

Мой вариант - 8 (рис. ??).


```
[damityakov@fedora lab06]$ touch ~/work/arch-pc/lab06/variant.asm
[damityakov@fedora lab06]$ nasm -f elf variant.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o variant variant.o
[damityakov@fedora lab06]$ ./variant
Введите № студенческого билета:
1132236067
Ваш вариант: 8
[damityakov@fedora lab06]$
```

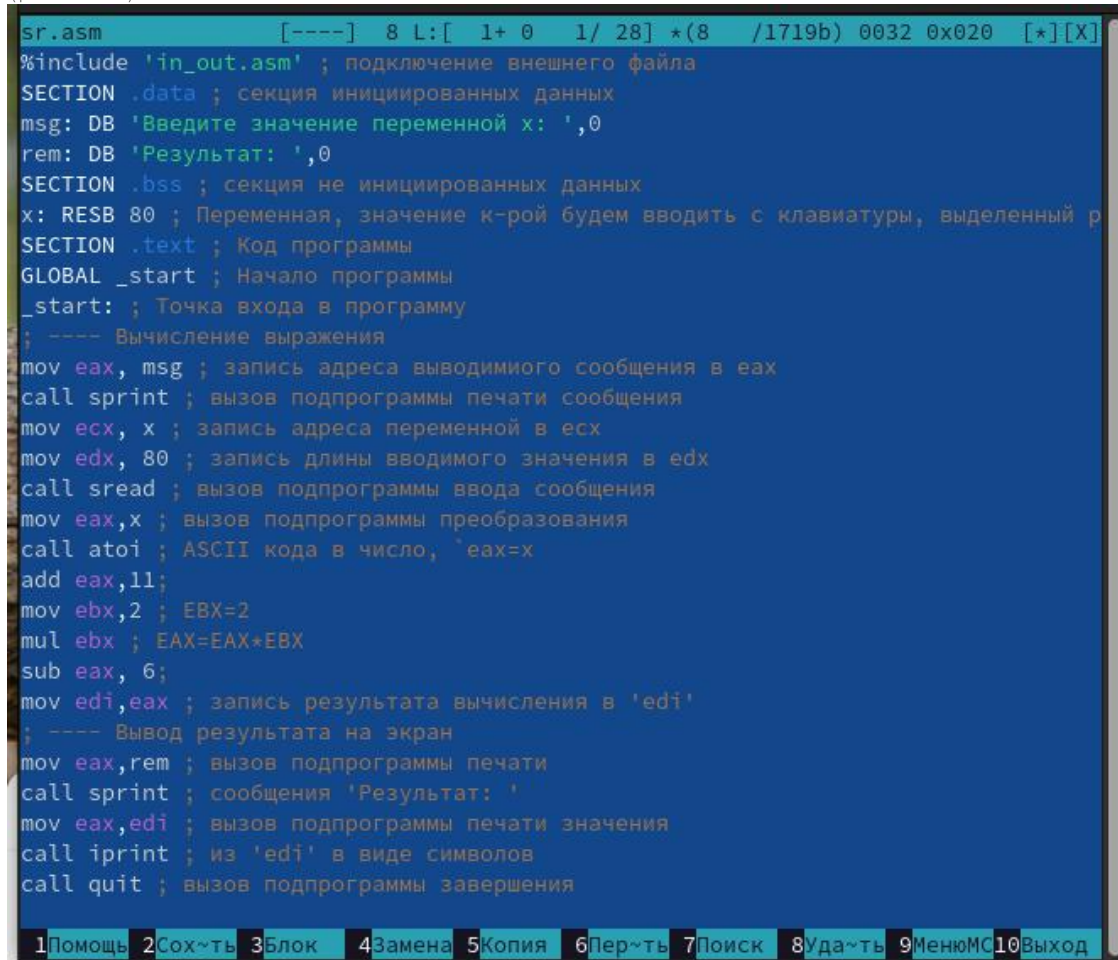
(рис. ??). - 1. За вывод сообщения “Ваш вариант” отвечают строки кода: `mov eax,rem` `call sprint` 2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры 3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax` 4. За вычисления варианта отвечают строки: `xor edx,edx` ; обнуление `edx` для корректной работы `div mov ebx,20` ; `ebx = 20` `div ebx` ; `eax = eax/20`, `edx` - остаток от деления `inc edx` ; `edx = edx + 1` 5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx` 6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1 7. За вывод на экран результатов вычислений отвечают строки:

3 Задания для самостоятельной работы

Написал код программы, описывающий вычисление результата функции 8 варианта

```
[damityakov@fedora lab06]$ touch ~/work/arch-pc/lab06/sr.asm
[damityakov@fedora lab06]$ nasm -f elf sr.asm
[damityakov@fedora lab06]$ ld -m elf_i386 -o sr sr.o
[damityakov@fedora lab06]$ ./sr
Введите значение переменной x: 1
Результат: 18[damityakov@fedora lab06]$ ./sr
Введите значение переменной x: 9
Результат: 34[damityakov@fedora lab06]$
```

(рис. ??).



```
sr.asm      [----]  8 L:[ 1+ 0  1/ 28] *(8  /1719b) 0032 0x020  [*][X]
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; секция инициализированных данных
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss ; секция не инициализированных данных
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный р
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
; ---- Вычисление выражения
mov eax, msg ; запись адреса выводимого сообщения в eax
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x
add eax,11;
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
sub eax, 6;
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprint ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения

1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС10Выход
```

4 Выводы

Я освоил арифметические инструкции языка ассемблера NASM.

Список литературы