

Отчет по выполнению лабораторной работы №6

Дисциплина: архитектура компьютеров

Гром Мария Леонидовна

Содержание

Цель работы	1
Выполнение лабораторной работы	1
Символьные и численные данные в NASM	1
Выполнение фрифметических операций в NASM	5
Выводы	8
Список литературы.....	8

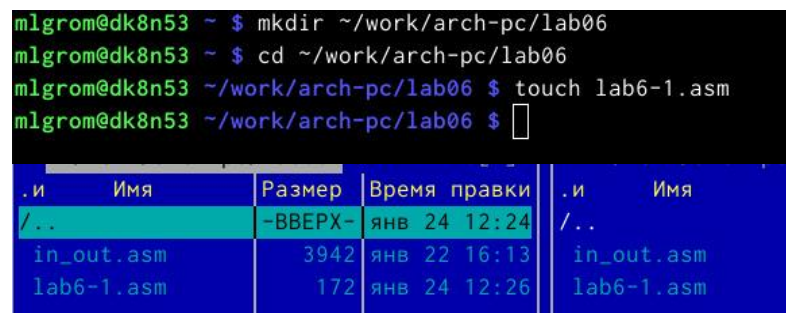
Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

Выполнение лабораторной работы

Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы №6, перехожу в него и создаю файл lab6-1.asm (рис. [fig:001]),(рис. [fig:002])



```
mlgrom@dk8n53 ~ $ mkdir ~/work/arch-pc/lab06
mlgrom@dk8n53 ~ $ cd ~/work/arch-pc/lab06
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ touch lab6-1.asm
mlgrom@dk8n53 ~/work/arch-pc/lab06 $
```

.и	Имя	Размер	Время правки	.и	Имя
/..		-ВВЕРХ-	янв 24 12:24	/..	
	in_out.asm	3942	янв 22 16:13		in_out.asm
	lab6-1.asm	172	янв 24 12:26		lab6-1.asm

Ввожу в файл lab6-1.asm текст программы из листинга 6.1.(рис. [fig:003]).

```
%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
```

Ввод программы из листинга 6.1

Создаю исполняемый файл и запускаю его.(рис. [fig:004]).

```
m1grom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
m1grom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
m1grom@dk8n53 ~/work/arch-pc/lab06 $ ./lab6-1
j
m1grom@dk8n53 ~/work/arch-pc/lab06 $
```

Запуск файла

Изменяю текст программы и вместо символов записываю в регистры числа.(рис. [fig:005]).

```
%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
```

Исправление программы

Далее создаю исполняемый файл и запускаю его.(рис. [fig:006]).

```
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ./lab6-2

mlgrom@dk8n53 ~/work/arch-pc/lab06 $
```

Запуск исправленной программы

Пользуясь таблицей ASCII можно определить, что код 10 соответствует символу переносу строки. Этот символ не отображается на экране.

Создаю новый файл lab6-3 в том же каталоге и ввожу в него текст программы из листинга 6.2.(рис. [fig:007]).

```
/afs/.dk.sci.pfu.edu.ru/home/m/1/mlgrom/work/arch-pc/lab06/lab6-3.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Создание нового файла

Создаю исполняемый файл и запускаю его.(рис. [fig:008]).

```
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ./lab6-3
106
mlgrom@dk8n53 ~/work/arch-pc/lab06 $
```

Запуск новой программы

Аналогично предыдущему примеру изменяю символы на числа.(рис. [fig:009]).

```
/afs/.dk.sci.pfu.edu.ru/home/m/l/mlgrom/work/arch-pc/lab06/lab6-3.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Изменение программы

Создаю исполняемый файл и запускаю его.(рис. [fig:010]).

```
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3-2.asm
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3-2 lab6-3-2.o
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ./lab6-3-2
10
mlgrom@dk8n53 ~/work/arch-pc/lab06 $
```

Запуск исправленной программы

При исполнении программы было получено число 10.

Заменяю функцию iprintLF на iprint. Создаю исполняемый файл и запускаю его.(рис. [fig:011]).(рис. [fig:012]).

```
/afs/.dk.sci.pfu.edu.ru/home/m/l/mlgrom/work/arch-pc/lab06/lab6-3-3.asm
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Изменение файла

```
m1grom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3-3.asm
m1grom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3-3 lab6-3-3.o
m1grom@dk8n53 ~/work/arch-pc/lab06 $ ./lab6-3-3
10m1grom@dk8n53 ~/work/arch-pc/lab06 $
```

Запуск программы

Вывод функции `iprint` отличается от `iprintLF` тем, что выведенное сообщение не переносится на следующую строку.

Выполнение фрифметических операций в NASM

Создаю файл `lab6-4.asm` в каталоге `~/work/arch-pc/lab06`. (рис. [fig:013]).

```
m1grom@dk8n53 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/lab6-4.asm
m1grom@dk8n53 ~/work/arch-pc/lab06 $
```

Создание файла

После внимательного прочтения текста программы из листинга 6.3 ввожу его в `lab6-4.asm`. (рис. [fig:014]).

```
/afs/.dk.sci.pfu.edu.ru/home/m/1/m1grom/work/arch-pc/lab06/lab6-4.asm  Изм
;-----
; 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,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
```

Ввод программы из листинга 6.3

Создаю исполняемый файл и запускаю его. (рис. [fig:013]).

```
m1grom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4.asm
m1grom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
m1grom@dk8n53 ~/work/arch-pc/lab06 $ ./lab6-4
Результат: 4
Остаток от деления: 1
m1grom@dk8n53 ~/work/arch-pc/lab06 $
```

Запуск программы

Изменяю тест программы для вычисления выражения $f(x)=(4*6+2)/5$. (рис. [fig:014]).

Создаю исполняемый файл и запускаю его.(рис. [fig:015]).

```
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf lab6-4-2.asm
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-4-2 lab6-4-2.o
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ./lab6-4-2
Результат: 5
Остаток от деления: 1
mlgrom@dk8n53 ~/work/arch-pc/lab06 $
```

Запуск программы

Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06.(рис. [fig:016]).

```
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
```

Создание файла variant.asm

Внимательно изучаю текст программы из листинга 6.4 и ввожу в файл variant.asm.(рис. [fig:017]).

```
/afs/.dk.sci.pfu.edu.ru/home/m/1/mlgrom/work/arch-pc/lab06/variant.asm
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 sprint
mov eax, edx
call iprintf
call quit
```

Создание файла variant.asm

Создаю исполняемый файл и запускаю его. Мой вариант - 5.(рис. [fig:018]).

```
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ./variant
Введите No студенческого билета:
1132236024
Ваш вариант: 5
```

Запуск файла

Ответы на вопросы

1. За вывод сообщение “Ваш вариант” отвечают строки: `mov eax,rem call sprint`
2. Эти строки используются чтобы считать x.

3. Call atoi преобразовывает код ASCII в целое число.
4. За вычисление варианта отсчитывают строки: xor edx,edx mov ebx,20 div ebx inc edx
5. Остаток от деления записывается в регистр edx.
6. Инструкция inc edx используется для того, чтобы увеличить значение регистра edx на 1.
7. Для вывода на экран результата вычислений используются строки: mov eax,edx call iprintLF

#Задание для самостоятельной работы

Создаю файл task.asm в каталоге ~/work/arch-pc/lab06.

Открываю созданную файл и начинаю печатать в него текст программы для вычисления $(9 \cdot x - 8) / 8$ (вариант 5)(рис. [fig:19]).

```

/afs/.dk.sci.pfu.edu.ru/home/m/l/mlgrom/work/arch-pc/lab06/task.asm
; Программа вычисления выражения
;-----
%include 'in_out.asm' ; подключение внешнего файла

SECTION .data
msg: DB 'Введите x: ',0
res: DB 'Результат: ',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
call atoi
;
; ---- Вычисление выражения

mov ebx,9
mul ebx
sub eax,8
xor edx,edx
mov ecx,8
div ecx

```

Ввод программы

Далее сохраню файл, создаю исполняемый файл и запускаю его. (рис. [fig:020]).

```
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ nasm -f elf task.asm
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o task task.o
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ./task
Введите x:
8
Результат: 8
mlgrom@dk8n53 ~/work/arch-pc/lab06 $ ./task
Введите x:
64
Результат: 71
mlgrom@dk8n53 ~/work/arch-pc/lab06 $
```

Проверка для значения x1, x2

Всё верно работает.

Код:

```
%include 'in_out.asm'
```

```
SECTION .data msg: DB 'Введите x:',0 rem: DB 'Результат:',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 call atoi
```

```
mov ebx,9 mul ebx sub eax,8 xor edx,edx mov ecx,8 div ecx mov edi,eax
```

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

Выводы

После выполнения данной работы я освоила фрифметические инструкции языка ассемблера NASM

Список литературы