

**Отчет Лабораторная работа No5. Основы
работы с Midnight Commander (mc).
Структура программы на языке
ассемблера NASM. Системные вызовы в
ОС GNU Linux**

Простейший вариант

Матвеева Анастасия Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Ход работы	7
4	Выводы	22
	Список литературы	23

Список иллюстраций

3.1	...	7
3.2	МС	8
3.3	МС	9
3.4	МС	10
3.5	МС	11
3.6	МС	11
3.7	МС	12
3.8	МС	13
3.9	МС	13
3.10	МС	14
3.11	МС	14
3.12	МС	15
3.13	МС	16
3.14	МС	18
3.15	МС	20
3.16	МС	21

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Написать программу вычисления выражения $y = f(x)$. Программа должно 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 Ход работы

Лабораторная работа

1. Создайте каталог для программ лабораторной работы No 7, перейдите в него и создайте файл lab7-1.asm:

```
asmatveeva1@dk5n51 ~ $ cd ~/work/arch-pc/lab06
bash: cd: /afs/.dk.sci.pfu.edu.ru/home/a/s/asmatveeva1/work/arch-pc/lab06: Нет такого файла или каталога
asmatveeva1@dk5n51 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab06
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch lab6-1.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc

asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1
.asm
lab6-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
```

Рис. 3.1: .

2. Рассмотрим примеры программ вывода символьных и численных значений.
Программы будут выводить значения записанные в регистры еах.

```
%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 sprintfLF

    call quit
```

Рис. 3.2: MC

3. Создайте исполняемый файл и запустите его

```
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab7-1
bash: ./lab7-1: Нет такого файла или каталога
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1
j
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.3: МС

4. Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправьте текст программы (Листинг 1) замените строки

```
%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. 3.4: MC

5. Создайте исполняемый файл и запустите его.

```
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-1.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-1
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.5: МС

6. Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и введите в него текст программы из листинга 7.2.

```
asmatveeva1@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06/report $ touch lab6-2.asm
```

Рис. 3.6: МС

7. Переделайте его

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax, '6'

mov ebx, '4'
add eax, ebx
call iprintLF

call quit
```

Рис. 3.7: MC

8. Создайте исполняемый файл и запустите его.

```

asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab
6-2.o
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
106
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ █

```

Рис. 3.8: МС

9. Так же убираем как и в прошлый раз

```

%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax, 6

mov ebx, 4
add eax, ebx
call iprintLF

call quit

```

Рис. 3.9: МС

10. Создайте исполняемый файл и запустите его.

```
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.10: MC

11. Создаем файл lab6-3

```
asmatveeva1@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06/report
$
touch lab6-3.asm
```

Рис. 3.11: MC

12. В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения $\boxtimes(\boxtimes) = (5 \boxtimes 2 + 3)/3$

```

;-----
; 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
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. 3.12: MC

13. Создайте исполняемый файл и запустите его. Результат работы программы должен быть следующим:

```

asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf lab6-2.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab
6-2.o
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./lab6-2
10asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $

```

Рис. 3.13: МС

14. Создайте файл variant.asm в каталоге ~/work/arch-pc/lab07:

```

asmatveeva1@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06/report
$
touch variant.asm

```



```

%include 'in_out.asm'
SECTION .data
msg: DB 'Вариант 2: (10 + 2x)/3 ',0
gsm: DB 'Введите значение для x:',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov eax, gsm
call sprintLF

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi
mov ebx, 2
mul ebx
add eax, 10
mov ebx, 3
div ebx
mov edi, eax

mov eax, rem
call sprint
mov eax, edi
call iprintLF

call quit

```

15. Мой вариант

```

asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ touch variant.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ mc

asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ nasm -f elf variant.asm
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o

asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ld -m elf_i386 -o variant variant.o
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant
Введите No студенческого билета:
1132222000
Ваш вариант: 1
asmatveeva1@dk5n51 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $

```

Рис. 3.14: MC

Ответы на вопросы

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения 'Ваш вариант:'?

Ответ: `mov eax,rem call sprint`

2. Для чего используются следующие инструкции? `nasm mov ecx, x mov edx, 80 call sread`

Ответ: `nasm` - переход к языку ассемблера `mov ecx, x` - присвоение значения `x` переменной `ecx` `mov edx, 80` - присвоение значение `80` переменной `edx` `call sread` - для считывания в переменную какого то числа

3. Для чего используется инструкция "call atoi"?

Ответ: Конвертирует строку, на которую указывает параметр `str`, в величину типа `int`

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

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

5. В какой регистр записывается остаток от деления при выполнении инструкции "div ebx"?

Ответ: Остаток запишется в регистр dx

6. Для чего используется инструкция “inc edx”?

Ответ: Это инкремент для прибавления единицы к переменной

7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений?

Ответ: mov eax,rem call sprint mov eax,edx call iprintLF

Самостоятельная работа

1. Написать программу вычисления выражения $x = x(x)$. 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 $x(x)$ 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

```

%include 'in_out.asm'
SECTION .data
msg: DB 'Вариант 2: (10 + 2x)/3 ',0
gsm: DB 'Введите значение для x:',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov eax, gsm
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

```

Рис. 3.15: МС

Мы добавили несколько строк для вывода результата нашей программы

2. Выводим результат программы

```
asmatveeva1@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant
Вариант 2: (10 + 2x)/3
Введите значение для x:
1
Результат: 4
asmatveeva1@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $ ./variant
Вариант 2: (10 + 2x)/3
Введите значение для x:
10
Результат: 10
asmatveeva1@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06 $
```

Рис. 3.16: МС

4 Выводы

Мы обучились работать с NASM.

Список литературы