

# **Архитектура вычислительных машин**

**Лабораторная работа N5**

Норсоян Шушаник Гагиковна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>11</b>
	<b>Список литературы</b>	<b>12</b>

## Список иллюстраций

2.1	рисунок1	6
2.2	рисунок2	7
2.3	рисунок3	7
2.4	рисунок4	8
2.5	рисунок5	8
2.6	рисунок6	8
2.7	рисунок7	8
2.8	рисунок 8	9
2.9	рисунок 9	9
2.10	рисунок 10	10

## Список таблиц

# 1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## 2 Выполнение лабораторной работы

- 1.Открою Midnight Commander
- 2.Перейду в каталог, созданный при выполнении лабораторной работы N5
- 3.С помощью функциональной клавиши создам папку lab5
- 4.Пользуясь строкой ввода и спецтальной командой создам файл lab5
- 5.С помощью функциональной клавиши открою файл lab5 для редактирования во встроенном редакторе.

Левая панель		Файл	Команда	Настройки	Правая панель	
<		...k/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05			.[^]>	
.и		Имя			Размер	Время правки
/..					-ВВЕРХ-	окт 26 13:19
/presentation					2048	окт 26 13:19
/report					2048	окт 26 13:19
in_out(1).asm					3942	ноя 30 14:40
*lab5-1					8744	ноя 30 14:19
lab5-1.asm					3423	ноя 30 14:19
lab5-1.o					752	ноя 30 14:19
lab5-2.asm					3942	ноя 30 14:40

Рис. 2.1: рисунок1

- 6.Введу текст программы из листинга, сохраню изменения и закрою файл.
- 7.С помощью функциональной клавиши открою файл lab5 для просмотра. Убежусь, что файл содержит текст программы

```

;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
call sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 2.2: рисунок2

8.Оттранслирую текст программы lab5 в объектный файл. Выполню компоновку объектного файла и запущу получившийся исполняемый файл.

```

sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -o lab5-2.o -f
elf lab5-2.asm
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 lab5
-2.o -o bir
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./bir
Введите строку: Норсоян Шушаник Гагиковна
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ █

```

Рис. 2.3: рисунок3

9.Скачаю файл in\_out.asm со страницы курса в ТУИС.

10.Скопирую файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши

11.С помощью функциональной клавиши создам копию файла lab5-1.asm с именем lab5-2.asm.

12.Исправлю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm

13.В файле lab6-2.asm заменю подпрограмму sprintf на printf. Создам исполняемый файл и проверю его работу.

```
SECTION .text ; Код программы^M
GLOBAL _start ; Начало программы^M
_start: ; Точка входа в программу^M
mov eax, msg ; запись адреса выводимого сообщения в 'EAX'^M
call sprint ; вызов подпрограммы печати сообщения^M
mov ecx, buf1 ; запись адреса переменной в 'EAX'^M
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'^M
```

Рис. 2.4: рисунок4

```
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -o lab5-2.o -f
elf lab5-2.asm
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 lab5
-2.o -o bir
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./bir
Введите строку: Норсоян Шушаник Гагиковна
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ █
```

Рис. 2.5: рисунок5

```
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -o lab5-2.o -f
elf lab5-2.asm
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 lab5
-2.o -o bir
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./bir
Введите строку: Норсоян Шушаник Гагиковна
sgnorsoyan@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ █
```

Рис. 2.6: рисунок6

## Задание для самостоятельной работы

### 1.Создам копию файла lab5-1.asm.

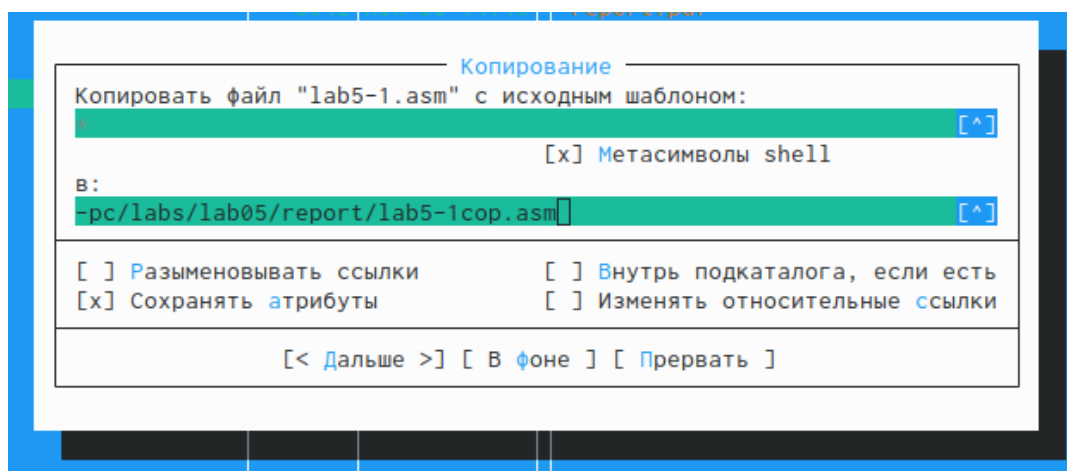


Рис. 2.7: рисунок7

### 2.Создам копию файла lab5-2.asm.



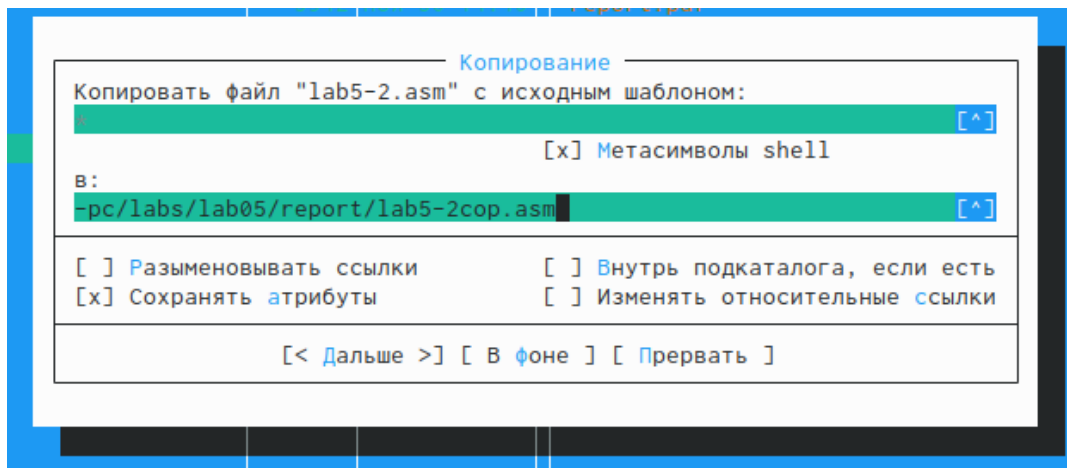


Рис. 2.8: рисунок 8

3.Получим исполняемый файл и проверим его работу. На приглашение ввести строку вводим свою фамилию

```

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- Объявление переменных -----
SECTION .data ;
msg: DB 'Введите строку:',10 ; сообщение плюс
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт

;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h

;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h

mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h

;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу

```

Рис. 2.9: рисунок 9

#### 4. Создаем копию файла lab6-2.asm. Исправляем текст программы

```
-----  
; Программа вывода сообщения на экран и ввода строки с клавиатуры  
-----  
%include 'in_out.asm' ; подключение внешнего файла  
  
SECTION .data ; Секция иницированных данных  
msg: DB 'Введите строку: ',0h ; сообщение  
  
SECTION .bss ; Секция не иницированных данных  
buf1: RESB 80 ; Буфер размером 80 байт  
  
SECTION .text ; Код программы  
GLOBAL _start ; Начало программы  
_start: ; Точка входа в программу  
  
mov eax, msg ; запись адреса выводимого сообщения в 'EAX'  
call sprint ; вызов подпрограммы печати сообщения  
  
mov ecx, buf1 ; запись адреса переменной в 'ECX'  
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'  
call sread ; вызов подпрограммы ввода сообщения  
  
mov eax, buf1 ;  
call sprint ;  
  
call quit ; вызов подпрограммы завершения
```

Рис. 2.10: рисунок 10

## 3 Выводы

Я приобрела практические навыки работы в Midnight commander. Освоили инструкции языка ассемблера mov и int.

## **Список литературы**