

# **Отчёт по лабораторной работе №5**

**Дисциплина: Архитектура компьютера**

**Машков Илья Евгеньевич**

# Содержание

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Цель работы</b>                                      | <b>4</b>  |
| <b>2</b> | <b>Задание</b>  | <b>5</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>                   | <b>6</b>  |
| 3.1      | Основы работы с тс . . . . .                            | 6         |
| 3.2      | Структура программы на языке ассемблера NASM . . . . .  | 7         |
| 3.3      | Подключение внешнего файла . . . . .                    | 9         |
| 3.4      | Выполнение заданий для самостоятельной работы . . . . . | 12        |
| <b>4</b> | <b>Выводы</b>   | <b>14</b> |
| <b>5</b> | <b>Список литературы</b>                                | <b>15</b> |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 3.1  | Непосредственно ms. . . . .                                  | 6  |
| 3.2  | Рабочая папка и каталог lab05 в ней. . . . .                 | 7  |
| 3.3  | Файл lab5-1.asm. . . . .                                     | 7  |
| 3.4  | Код программы. . . . .                                       | 8  |
| 3.5  | Проверка содержимого файла. . . . .                          | 9  |
| 3.6  | Трансляция, компоновка и запуск программы. . . . .           | 9  |
| 3.7  | Файл in_out.asm в директории lab05. . . . .                  | 10 |
| 3.8  | Копирование содержимого одного файла в другой. . . . .       | 10 |
| 3.9  | Файл lab5-2.asm. . . . .                                     | 10 |
| 3.10 | Редактирование кода программы. . . . .                       | 11 |
| 3.11 | Запуск программы со строкой 'sprintf'. . . . .               | 11 |
| 3.12 | Запуск программы со строкой 'sprintf'. . . . .               | 11 |
| 3.13 | Изменённый код программы. . . . .                            | 12 |
| 3.14 | Получение и запуск исполняемого файла. . . . .               | 12 |
| 3.15 | Изменённый код программы, использующей внешний файл. . . . . | 13 |
| 3.16 | Получение и запуск исполняемого файла. . . . .               | 13 |

# 1 Цель работы

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

## 2 Задание

1. Основы работы с тс
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

## 3 Выполнение лабораторной работы

### 3.1 Основы работы с mc

Открываю Midnight Commander с помощью команды mc (рис. [3.1]).

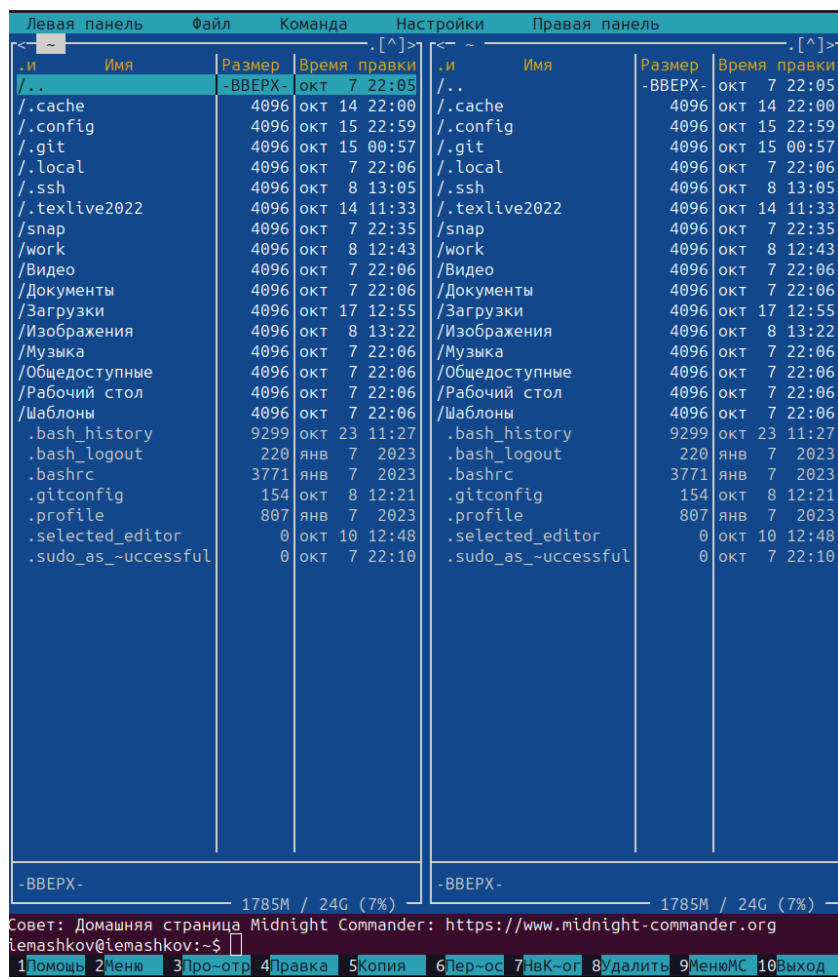


Рис. 3.1: Непосредственно mc.

Потом перехожу в рабочий каталог, используя `mc`, и создаю каталог `lab05`. Создать каталог можно, как через терминал (именно так я его и создал), так и с помощью кнопки `F7` (Рис. [3.2]).

| Левая панель                                   | Файл | Команда | Настройка    |
|--|------|---------|--------------|
| [< ...24/Архитектура компьютера/arch-rc -.[^]> |      |         |              |
| .и   | Имя  | Размер  | Время правки |
| /..  |      | -ВВЕРХ- | окт 10 13:00 |
| /.git  |      | 4096    | окт 23 10:22 |
| /config  |      | 4096    | окт 10 13:00 |
| /lab04   |      | 4096    | окт 22 19:25 |
| /lab05   |      | 4096    | ноя 7 12:25  |

Рис. 3.2: Рабочая папка и каталог `lab05` в ней.

Далее перехожу в созданный каталог и с помощью команды `'touch'` создаю файл `lab5-1.asm` (Рис. [3.3]).

| Левая панель   | Файл | Команда | Настройка    |
|--|------|---------|--------------|
| [< ...24/Архитектура компьютера/arch-rc/lab05 -.[^]> |      |         |              |
| .и   | Имя  | Размер  | Время правки |
| /..  |      | -ВВЕРХ- | ноя 7 12:25  |
| lab5-1.asm   |      | 0       | ноя 7 12:27  |

Рис. 3.3: Файл `lab5-1.asm`.

## 3.2 Структура программы на языке ассемблера NASM

Открываю файл с помощью клавиши `F4` и ввожу код программы для запроса строки у пользователя, сохраняю изменения с помощью комбинации (`Ctrl+O`) (Рис. [3.4]).

```

/home/iemashkov/work/study/2023-2024/Архитектура компьютера/arch-pc/lab06/lab6.asm *
SECTION .data ; Секция инициализированных данных
msg:  DB 'Введите строку:',10 ; сообщение плюс
                                ; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
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
;----- 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 ; Вызов ядра

;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт

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 ; Вызов ядра

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

mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.4: Код программы.

А затем просматриваю содержимое файла с помощью клавиши F3 (Рис. [3.5]).



```

/home/temashkov/work/study/2-ютепа/arch-pc/lab06/lab6.asm 2070/2070
SECTION .data ; Секция инициализированных данных
msg:  DB 'Введите строку:',10 ; сообщение плюс
                                ; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
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
;----- 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 ; Вызов ядра

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

;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.5: Проверка содержимого файла.

Далее транслирую текст программы в объектный файл, компаную полученный файл и запускаю исполняемый файл (Рис. [3.6]).

```

temashkov@temashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-1.asm
temashkov@temashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
temashkov@temashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ls
lab5-1  lab5-1.asm  lab5-1.o
temashkov@temashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-1
Введите строку:
Машков Илья Евгеньевич
temashkov@temashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ 

```

Рис. 3.6: Трансляция, компоновка и запуск программы.

### 3.3 Подключение внешнего файла

Скачиваю файл in\_out.asm со страницы курса на ТУИС. Так как он сохранился в папке “Загрузки”, я с помощью клавиши F5 переношу его в каталог lab05 (Рис. [3.7]).

| Левая панель | Файл                            | Команда | Наст         |
|--------------|---------------------------------|---------|--------------|
| <            | ...ура компьютера/arch-pc/lab05 | -.[^]>  |              |
| .и           | Имя                             | Размер  | Время правки |
| /..          |                                 | -ВВЕРХ- | ноя 7 12:29  |
| in_out.asm   |                                 | 3942    | ноя 7 12:46  |
| *lab5-1      |                                 | 8744    | ноя 11 18:34 |
| lab5-1.asm   |                                 | 2069    | ноя 11 18:32 |
| lab5-1.o     |                                 | 752     | ноя 11 18:33 |

Рис. 3.7: Файл in\_out.asm в директории lab05.

Затем с помощью всё той же клавиши F5 копирую содержимое файла lab5-1.asm в файл lab5-2.asm (Рис. [3.8]).

Копирование

Копировать файл "in\_out.asm" с исходным шаблоном:  

☐ Метасимволы shell

в:  

☐

☐ Разыменовывать ссылки      ☐ Внутрь подкаталога, если есть  
☒ Сохранять атрибуты      ☐ Изменять относительные ссылки

Рис. 3.8: Копирование содержимого одного файла в другой.

Этим действием я создал новый файл с содержимым старого (Рис. [3.9]).

|            |      |              |
|------------|------|--------------|
| lab5-2.asm | 6011 | ноя 11 18:44 |
|------------|------|--------------|

Рис. 3.9: Файл lab5-2.asm.

Теперь изменяю содержимое этого файла так, чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm (Рис. [3.10]).

```

#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. 3.10: Редактирование кода программы.

После этого я транслирую, компоную и запускаю программу сначала с параметром 'sprintf' (Рис. [3.11]).

```

lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2.asm
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ls
in_out.asm lab5-1 lab5-1.o lab5-2 lab5-2.o
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку:
Машков Илья Евгеньевич
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ 

```

Рис. 3.11: Запуск программы со строкой 'sprintf'.

А потом меняю этот параметр на 'sprintf' и произвожу те же действия (Рис. [3.12]).

```

lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2.asm
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2
Введите строку: Машков Илья Евгеньевич
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ 

```

Рис. 3.12: Запуск программы со строкой 'sprintf'.

Разница состоит в том, что с параметром 'sprintf' пользователь вводит свой текст с новой строки, а со 'sprintf' ввод происходит на той же строке, где написано "Введите строку:".

### 3.4 Выполнение заданий для самостоятельной работы

1. Создаю копию файла lab5-1.asm, используя клавишу F5. Затем изменяю код программы так, чтобы после вывода запроса о введении строки она также выводила и вводимую пользователем строку (Рис. [3.13]).

```
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10
msgLen: EQU $-msg ; Длина переменной 'msg'
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,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 ; Вызов ядра
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 ; Вызов ядра
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рис. 3.13: Изменённый код программы.

2. Теперь получаю исполняемый файл и запускаю его (Рис. [3.14]).

```
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ nasm -f elf lab5-1-1.asm
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ ls
in_out.asm lab5-1 lab5-1-1 lab5-1-1.asm lab5-1-1.o lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$ ./lab5-1-1
Введите строку:
Машков Илья Евгеньевич
tenashkov@tenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab0$
```

Рис. 3.14: Получение и запуск исполняемого файла.

3. Создаю копию файла lab5-2.asm и также меняю код программы, чтобы он выводил вводимую строку (Рис. [3.15]).

```
%include 'in_out.asm'
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициированных 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
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения
```

Рис. 3.15: Изменённый код программы, использующей внешний файл.

4. Получаю исполняемый файл и проверяю корректность его работы (Рис. [3.16]).

```
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ld -n elf_i386 -o lab5-2-1 lab5-2-1.o
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$ ./lab5-2-1
Введите строку: Машков Илья Евгеньевич
Машков Илья Евгеньевич
lenashkov@lenashkov:~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05$
```

Рис. 3.16: Получение и запуск исполняемого файла.

## 4 Выводы

В ходе выполнения лабораторной работы я приобрел практические навыки работы в Midnight Commander, а также освоил инструкции языка ассемблера `mov` и `int`.

## **5 Список литературы**

Архитектура ЭВМ