

**Лабораторная работа №5. Основы
работы с Midnight Commander (mc).
Структура программы на языке
ассемблера NASM.**

Дисциплина: Архитектура ЭВМ

Иван Шевырев

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Midnight Commander	6
2.1.1	Откроем файловый менеджер Midnight Commander	6
2.1.2	Перейдем в каталог ~/work/arch-pc/	6
2.1.3	Создадим папку	6
2.1.4	Создадим файл	7
2.1.5	Откроем файл	8
2.1.6	Введем код в файл	8
2.1.7	Запустим код	10
2.2	Подключение внешнего файла	11
2.2.1	Написание кода для lab5-2	12
2.2.2	Исполнение кода	13
2.2.3	Изменим строчку в коде	13
3	Задания для самостоятельной работы	15
3.1	Программа без использования in_out.asm	16
3.2	Программа с использованием in_out.asm	17
4	Выводы	19

Список иллюстраций

2.1	открытие Midnight Commander	6
2.2	Переход в каталог	6
2.3	Создание папки	7
2.4	Переход в новую папку	7
2.5	Создание файла	7
2.6	Созданный файл	8
2.7	Открытый пустой файл	8
2.8	Файл с введенным кодом	9
2.9	Сохранение файла	9
2.10	Проверка, что файл сохранился	10
2.11	Трансляция, линковка и запуск	10
2.12	Просмотр двух папок, одновременно	11
2.13	Копирование файла	11
2.14	Скопированный файл	12
2.15	Создание копии файла	12
2.16	Введенный код	13
2.17	Трансляция, линковка и запуск	13
2.18	Запуск измененного кода	14
3.1	Код lab5-2.1.asm	16
3.2	Вывод скомпилированной программы	17
3.3	Код lab5-2.1.asm	18
3.4	Вывод скомпилированной программы	18

Список таблиц

1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение

2 Выполнение лабораторной работы

2.1 Midnight Commander

2.1.1 Откроем файловый менеджер Midnight Commander

```
[idshevihryov@pivo ~]$ mc
```

Рис. 2.1: открытие Midnight Commander

2.1.2 Перейдем в каталог ~/work/arch-pc/

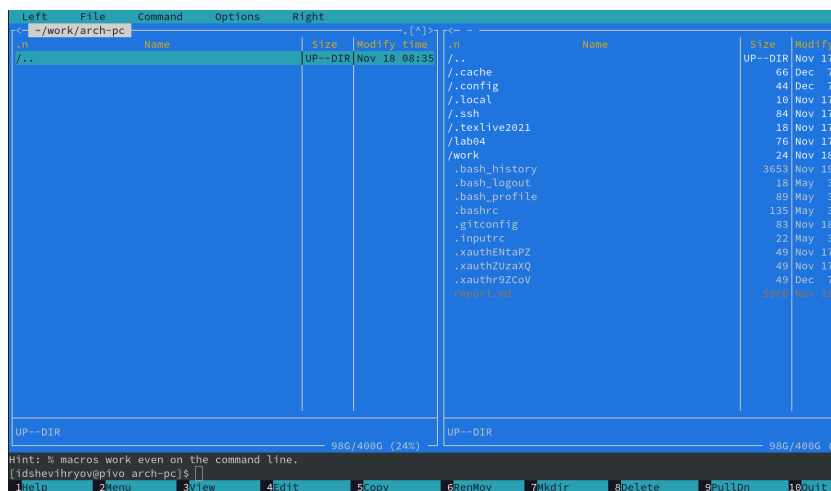


Рис. 2.2: Переход в каталог

2.1.3 Создадим папку

Нажмем клавишу F7

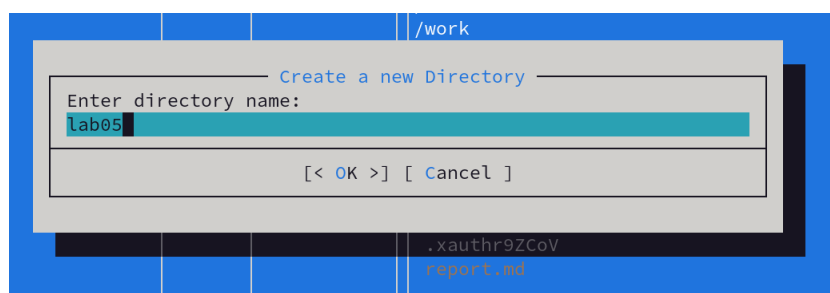


Рис. 2.3: Создание папки

Перейдем в созданную папку

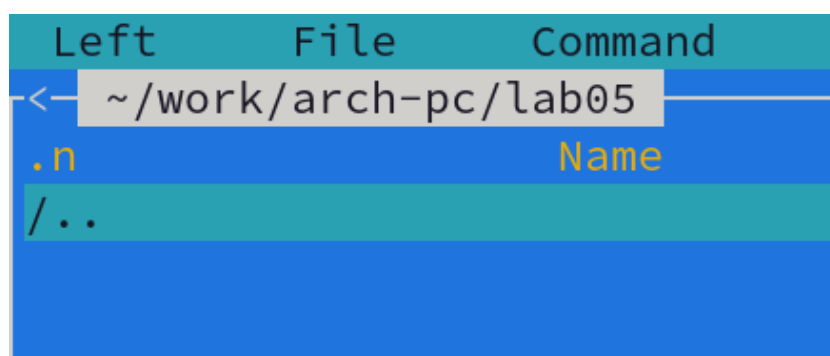


Рис. 2.4: Переход в новую папку

2.1.4 Создадим файл

Перейдя в папку, наберем `touch lab5-1.asm`, воспользовавшись строкой ввод.

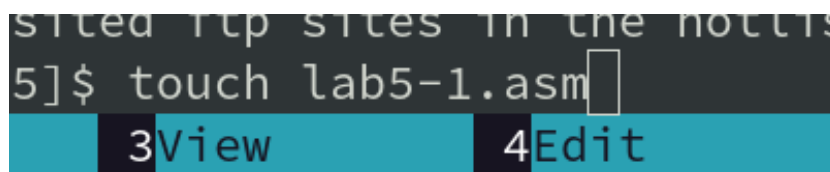


Рис. 2.5: Создание файла

Как видим в МС, файл создан

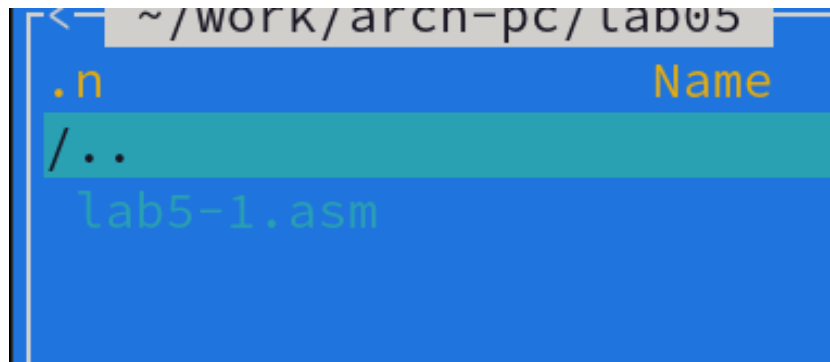


Рис. 2.6: Созданный файл

2.1.5 Откроем файл

Нажмем F4, что бы открыть файл

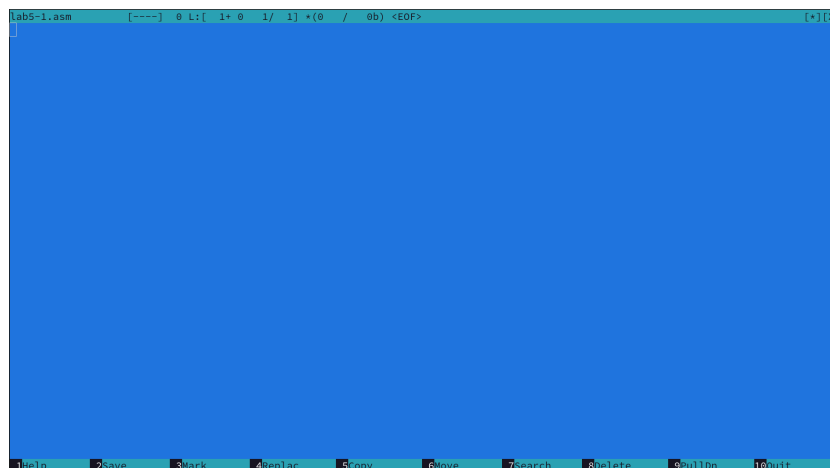


Рис. 2.7: Открытый пустой файл

2.1.6 Введем код в файл

Введем текст программы из листинга 6.1


```

lab5-1.asm      [-M--]  4  L:[  1+27  28/ 30]  *(772 / 809b) 0105 0x069
SECTION
msg:          DB 'Введите строку:', 10
msgLen: EQU $-msg

SECTION
buf1:        RESB 80

SECTION
GLOBAL _start
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, msg
    mov edx, msgLen
    int 80h

    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h

    mov eax, 1
    mov ebx, 0
    int 80h

```

Рис. 2.8: Файл с введенным кодом

Сохраним файл

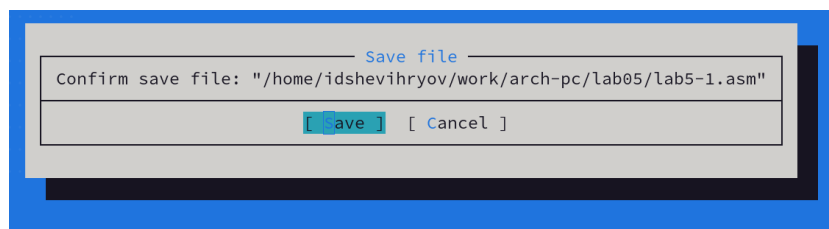


Рис. 2.9: Сохранение файла

Откроем заново файл, что б убедиться, что данные сохранились.

```

lab5-1.asm [----] 0 L: [ 1+29 30/ 30] *(809 / 80
SECTION .data
msg: DB 'Введите строку:', 10
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
mov eax, 4
mov ebx, 1
mov ecx, msg
mov edx, msgLen
int 80h

mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h

mov eax, 1
mov ebx, 0
int 80h

```

Рис. 2.10: Проверка, что файл сохранился

2.1.7 Запустим код

Создадим исполняемый файл и запустим его.

```

[idshevihryov@pivo lab05]$ nasm -f elf lab5-1.asm
[idshevihryov@pivo lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[idshevihryov@pivo lab05]$ ./lab5-1
Введите строку:
Иван Шевырев
[idshevihryov@pivo lab05]$ 

```

Рис. 2.11: Трансляция, линковка и запуск

Программа вывела наше имя.

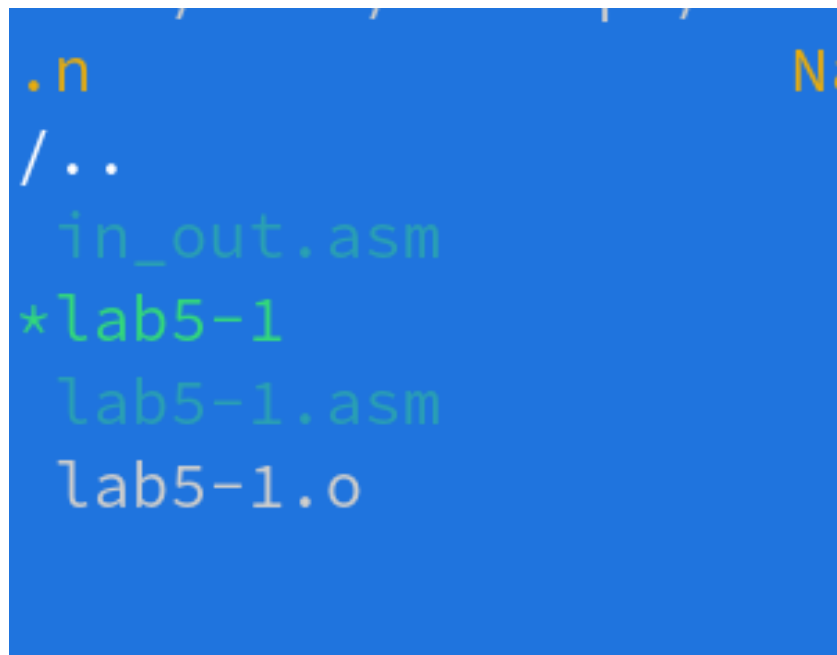


Рис. 2.14: Скопированный файл

Скопируем файл lab5-1.asm в lab5-2.asm, нажав F5

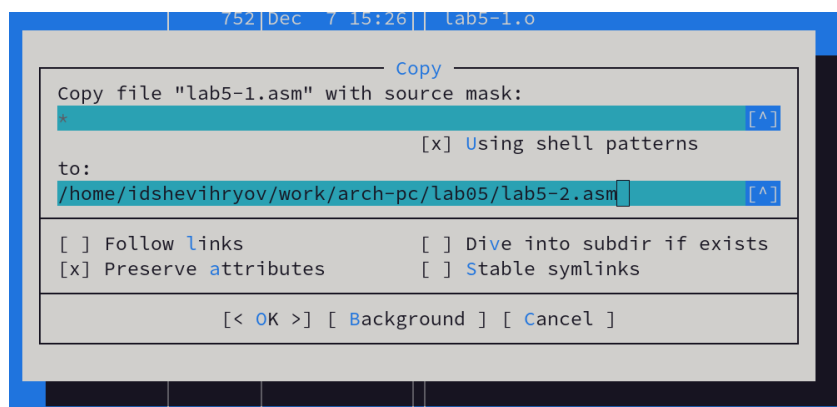


Рис. 2.15: Создание копии файла

2.2.1 Написание кода для lab5-2

Введем текст в файл

```

lab5-2.asm      [----]  0 L:[  1+ 0  1/ 2
%include '../in_out.asm' .....

SECTION .bss .....
buf1:  RESB 80 .....

SECTION .text .....
GLOBAL _start .....
_start: .....

    mov     eax, msg .....
    call    sprintLF .....

    mov     ecx, buf1 .....
    mov     edx, 80 .....
    call    sread .....

    call    quit .....

```

Рис. 2.16: Введенный код

2.2.2 Исполнение кода

```

[idshevihryov@pivo lab05]$ nasm -f elf lab5-2.asm
[idshevihryov@pivo lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[idshevihryov@pivo lab05]$ ./lab5-2
Введите строку:
Иван Шевырев
[idshevihryov@pivo lab05]$ 

```

Рис. 2.17: Трансляция, линковка и запуск

2.2.3 Изменим строчку в коде

Заменяем `sprintLF` на `sprit` и также запустим код.

```
[idshevihryov@pivo lab05]$ nasm -f elf lab5-2.asm
[idshevihryov@pivo lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[idshevihryov@pivo lab05]$ ./lab5-2
Введите строку:
Иван Шевырев
[idshevihryov@pivo lab05]$
```

Рис. 2.18: Запуск измененного кода

3 Задания для самостоятельной работы

3.1 Программа без использования in_out.asm

```
SECTION .data
msg:    DB 'Введите строку:', 10
msgLen: EQU $-msg

SECTION .bss
buf1:   RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, 4
    mov ebx, 1
    mov ecx, msg
    mov edx, msgLen
    int 80h

    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h

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

    mov eax, 1
    mov ebx, 0
    int 80h
```

Рис. 3.1: Код lab5-2.1.asm

Запустим и выведем результат

```
[idshevihryov@pivo lab05]$ nasm -f elf lab5-1.1.asm
[idshevihryov@pivo lab05]$ ld -m elf_i386 -o lab5-1.1 lab5-1.1.o
[idshevihryov@pivo lab05]$ ./lab5-1.1
Введите строку:
Шевырев
Шевырев
[idshevihryov@pivo lab05]$
```

Рис. 3.2: Вывод скомпилированной программы

Программа выводит введенную строку.

3.2 Программа с использованием `in_out.asm`

Напишем ту же программу, используя функции из `in_out.asm`

```

%include 'in_out.asm'

SECTION .data
msg:    DB 'Введите строку:', 10

SECTION .bss
buf1:   RESB 80

SECTION .text
GLOBAL _start
_start:

    mov     eax, msg
    call    sprint

    mov     ecx, buf1
    mov     edx, 80
    call    sread

    mov     eax, buf1
    call    sprint

    call    quit

```

Рис. 3.3: Код lab5-2.1.asm

Странслируем, слинкуем и запустим

```

[idshevihryov@pivo lab05]$ nasm -f elf lab5-2.1.asm
[idshevihryov@pivo lab05]$ ld -m elf_i386 -o lab5-2.1 lab5-2.1.o
[idshevihryov@pivo lab05]$ ./lab5-2.1
Введите строку:
Шевырев
Шевырев

```

Рис. 3.4: Вывод скомпилированной программы

Как видим, эта программа тоже выводит введенную строку.

4 Выводы

Мы приобрели практические навыки работы в Midnight Commander и освоили некоторые инструкции языка ассемблера (а именно `mov` и `int`)