

# **РУДН. Архитектура компьютеров**

**Отчёт по лабораторной работе №5**

Косинов Никита Андреевич, НПИМбв-02-20

# Содержание

1	Цель работы	5
2	Ход работы	6
3	Навигация в <i>Midnight Commander</i>	7
4	Редактирование файла в <i>mcedit</i>	9
5	Подключение “библиотеки”	13
6	Выводы	22

# Список иллюстраций

3.1	Запуск менеджера . . . . .	7
3.2	Создание каталога в МС . . . . .	7
3.3	Создание файла в МС . . . . .	8
3.4	Изменение файла в МС . . . . .	8
4.1	Код lab5-1 . . . . .	11
4.2	Сохранение файла . . . . .	11
4.3	Просмотр файла . . . . .	12
4.4	Запуск программы . . . . .	12
5.1	Копирование файла . . . . .	13
5.2	Код подключаемого файла . . . . .	14
5.3	Новая структура рабочего каталога . . . . .	14
5.4	Код lab5-2 . . . . .	16
5.5	Результат исполнения lab5-2 . . . . .	16
5.6	Копирование файла . . . . .	17
5.7	Результат исполнения lab5-3 . . . . .	18

## **Список таблиц**

# 1 Цель работы

Цель данной работы заключается в приобретении навыков работы с файловым менеджером *Midnight Commander*: навигация, копирование, перемещение и редактирование объектов. Также, мы продолжаем изучать команды языка программирования Ассемблер такие, как **mov** и **int**.

## 2 Ход работы

Лабораторная работа выполнена с использованием консоли **ОС Linux**, программы **Midnight Commander** и языка программирования ассемблера **NASM**.

1. Навигация в *Midnight Commander*;
2. Редактирование файла в *mcedit*;
3. Подключение “библиотеки”.

В конце выполнена самостоятельная работа.

### 3 Навигация в *Midnight Commander*

С помощью файлового навигатора *Midnight Commander* научимся переходить между каталогами, создавать новые и изменять файлы с помощью встроенного редактора

1. Командой **mc** открываем *Midnight Commander*.

```
nakosinov@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05 $ mc
```

Рис. 3.1: Запуск менеджера

2. Переходим выше по файловой системе в папку *~arch-pc* и создаём в ней каталог *lab05* с помощью клавиши **F7**.

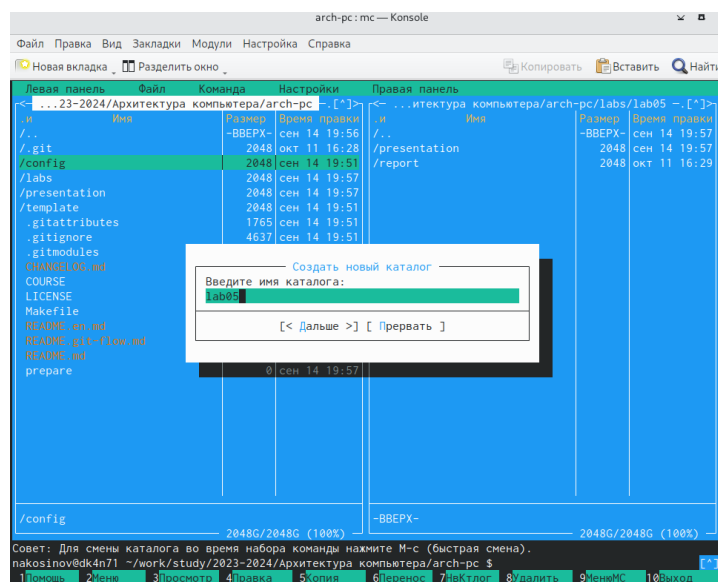


Рис. 3.2: Создание каталога в MC

3. С помощью строки ввода, оставшейся в нижней части консоли, командой **touch** создаём файл программы *lab5-1.asm*.

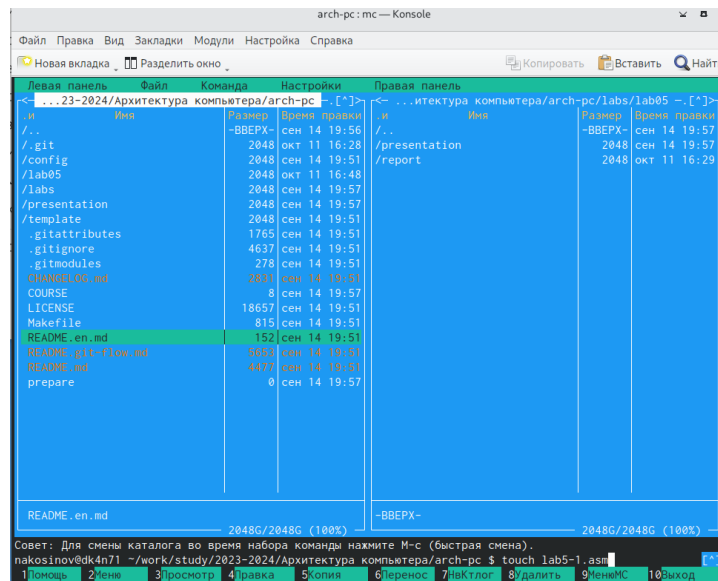


Рис. 3.3: Создание файла в МС

4. Выделив стрелками созданный файл и нажав клавишу **F4**, открываем файл во встроенном текстовом редакторе *mcedit*.

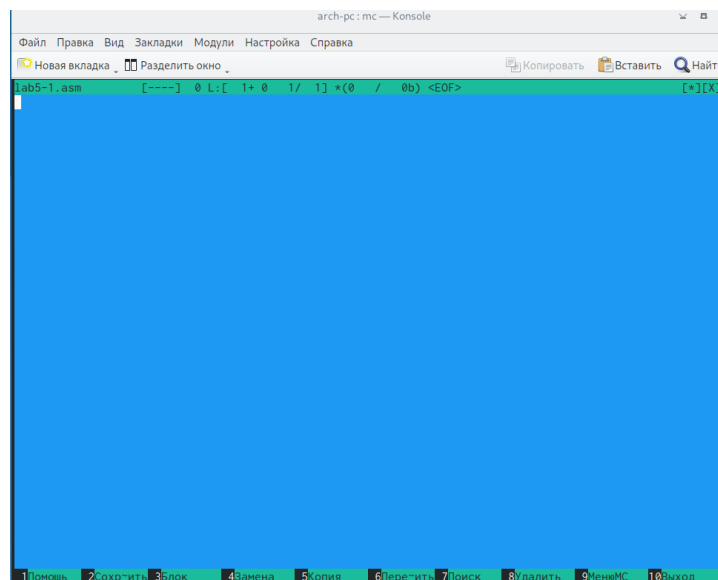


Рис. 3.4: Изменение файла в МС



## 4 Редактирование файла в *mcedit*

Открыв файл программы, напишем сам набор команд.

1. В блоке *.data* вводим переменные **msg** и **msgLen**, т.к. их значение нам известно.
2. В блоке *.bss* вводим переменную **buf1**, т.к. в ней будет храниться значение, заданное пользователем.
3. В блоке *.text* записываем последовательность групп команд. Каждая группа заканчивается вызовом ядра: **int 80h**.
  1. Первая группа отвечает за вывод на экран значения переменной **msg**.
  2. Вторая группа отвечает за сохранение значения, введённого пользователем, в переменную **buf1**.
  3. Третья группа отвечает за выход из программы без ошибок.

```
''' 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
```

```
    ,,,
```

```
arch-pc:mc — Console
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно Копировать Вставить Найти
lab5-1.asm [----] 11 L: [ 1+27 28/ 28] *(373 / 373b) <EOF> [*][X]
SECTION .data
msg: DB "Message: 123456789",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 ecx,3
mov ebx,0
mov esi,buf1
mov edi,80
int 80h
mov ecx,1
mov ebx,0
int 80h
1 Помощь 2 Сохранить 3 Блок 4 Замена 5 Копия 6 Перечитать 7 Поиск 8 Удалить 9 Меню MS 10 Выход
```

Рис. 4.1: Код lab5-1

4. Клавишей F2 сохраняем данные изменения.

```
arch-pc:mc — Console
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно Копировать Вставить Найти
lab5-1.asm [----] 11 L: [ 1+27 28/ 28] *(373 / 373b) <EOF> [*][X]
SECTION .data
msg: DB "Message: 123456789",10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
Сохранить файл
запись файла "/afs/.dk.sci.pfu.edu.ru/home/n/a/nakosinov/work/study/2023-2024/Архитектура компьютера/arch-
Сохранить [ Прервать ]
mov ecx,msg
mov edx,msgLen
int 80h
mov ecx,3
mov ebx,0
mov esi,buf1
mov edi,80
int 80h
mov ecx,1
mov ebx,0
int 80h
1 Помощь 2 Сохранить 3 Блок 4 Замена 5 Копия 6 Перечитать 7 Поиск 8 Удалить 9 Меню MS 10 Выход
```

Рис. 4.2: Сохранение файла

5. Клавишей F3 откроем написанный код только для просмотра.

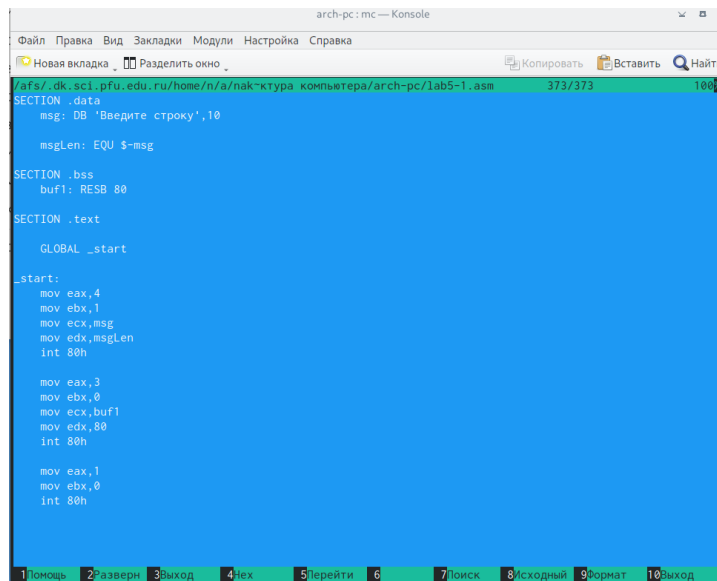


Рис. 4.3: Просмотр файла

6. Транслируем написанный код в объектный файл и компоуем последний. Запустив исполняемый файл, убеждаемся, что он работает исправно!

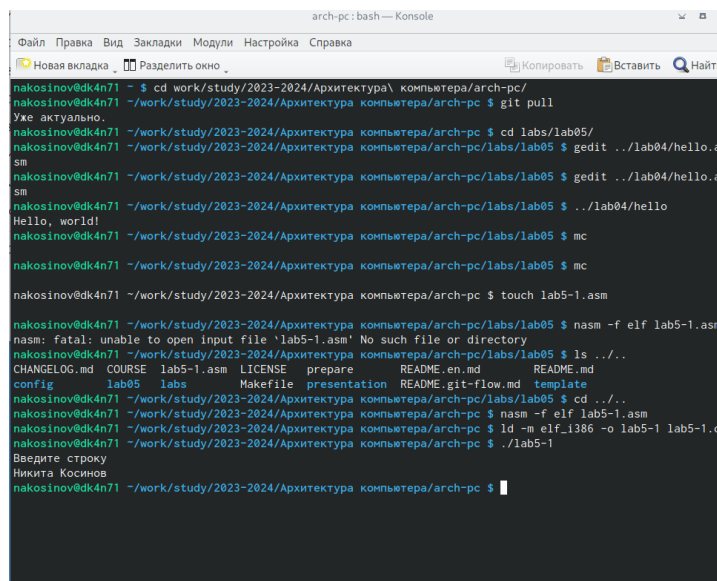


Рис. 4.4: Запуск программы

## 5 Подключение “библиотеки”

Для удобства работы и ускорения процесса, для возможности командной работы в ассемблере предусмотрена работа с подключаемыми файлами, в которых находятся прописанные заранее команды. При подключении такого файла ассемблер, видя незнакомую ему команду, ищет её в подключенном внешнем файле и подставляет написанный там набор команд.

1. Скачиваем файл *in\_out.asm*, открываем папку *Загрузки* во второй вкладке *МС* и копируем файл в наш рабочий каталог.

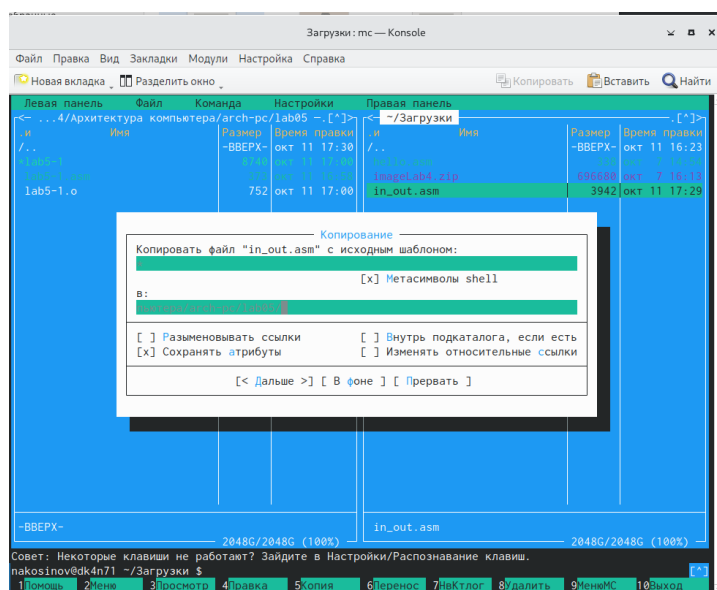


Рис. 5.1: Копирование файла

2. Клавишей **F3** откроем файл *in\_out.asm*, чтобы ознакомиться с его структурой.

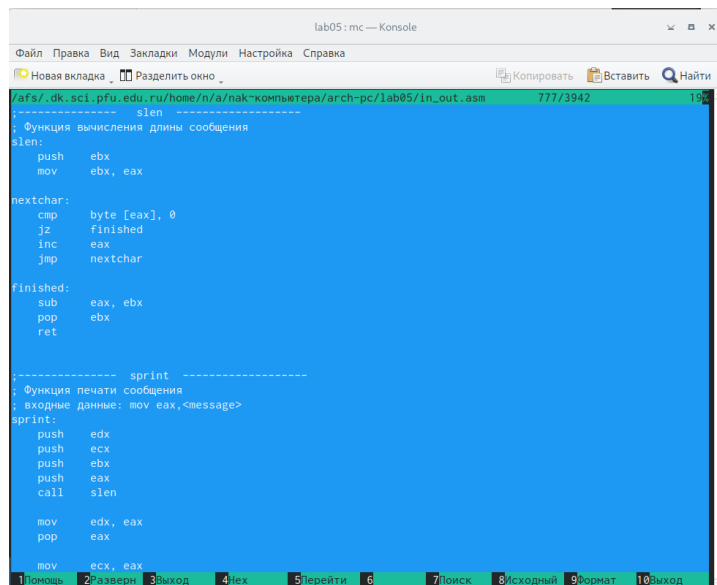


Рис. 5.2: Код подключаемого файла

3. Также скопируем написанную нами ранее программу, дав ей новое имя и оставив в той же папке.

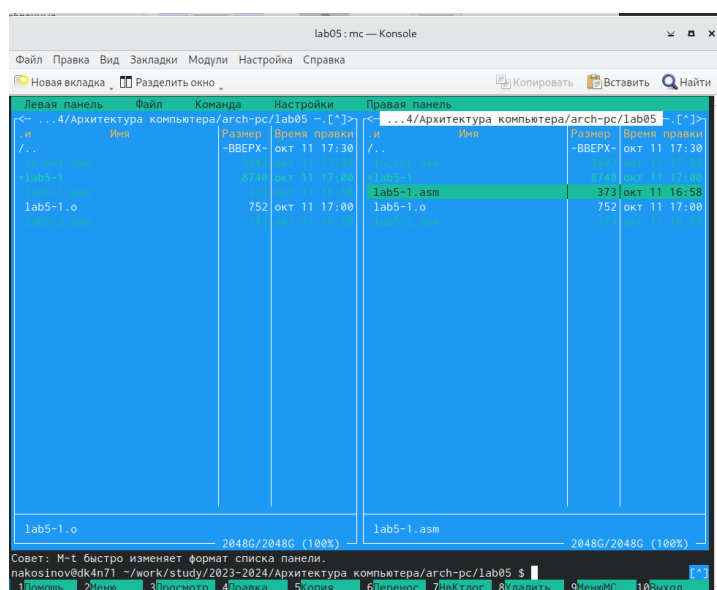


Рис. 5.3: Новая структура рабочего каталога

4. Изменяем код, подключая внешний файл и меняя стандартные команды

на команды из “библиотеки”.

```
''' %include 'in_out.asm'
SECTION .data

msg: DB 'Введите строку',0h

SECTION .bss

buf1: RESB 80

SECTION .text

GLOBAL _start

_start:

mov eax, msg
call sprintfLF

mov ecx, buf1
mov edx, 80
call sread

call quit

'''
```





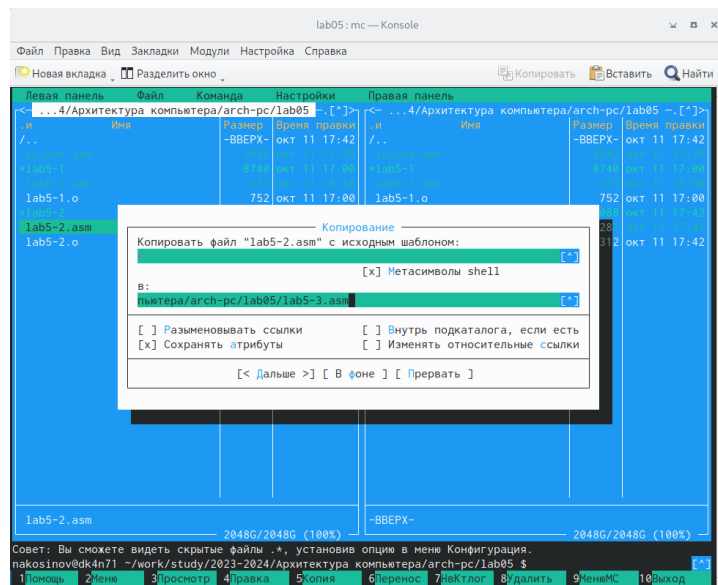


Рис. 5.6: Копирование файла

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

SECTION .data

msg: DB 'Введите строку',0h

SECTION .bss

buf1: RESB 80

SECTION .text

GLOBAL _start

_start:

mov eax, msg
call sprint

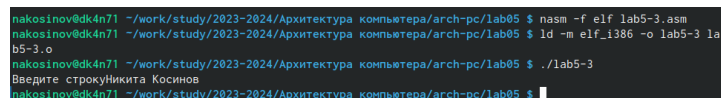
mov ecx, buf1
mov edx, 80
```

```
call sread
```

```
call quit
```

```
'''
```

7. Компилируем её и видим изменения: как и ожидалось, после вывода переменной **msg** не происходит переноса каретки на новую строку.



```
nakosinov@dk4n71 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5-3.asm
nakosinov@dk4n71 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
nakosinov@dk4n71 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05 $ ./lab5-3
Введите строку
nakosinov@dk4n71 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab05 $
```

Рис. 5.7: Результат исполнения lab5-3

### #Самостоятельная работа

1. Копируем файл *lab5-1.asm* (без подключаемого файла).

#### Копирование исходного файла

2. Изменяем код, добавив двоеточие после слов *Введите строку*, а также убрав символ переноса строки, чтобы потренироваться и чтобы результат выглядел логичнее. Также, добавляем группу команд по выводу значения, записанного в **buf1**.

```
''' SECTION .data
```

```
msg: DB 'Введите строку: '
```

```
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
```

””

Сохранение кода lab5-4.asm

3. Компилируем и запускаем соответствующий исполняемый файл

Создание исполняемого файла lab5-4 и её запуск

Результат работы программы

4. Копируем файл *lab5-2.asm* (с подключенным файлом).

Копирование исходного файла

5. Изменяем код, заменив команду **sprintLF** на **sprint** первый раз и добавив вывод значения, введенного пользователем.

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

SECTION .data

msg: DB 'Введите строку: ',0h

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
mov edx, 80
call sprintLF

call quit
```

””

Сохранение кода lab5-5.asm

6. Проверяем полученный результат.

Создание исполняемого файла lab5-5 и её запуск

Результат работы программы

7. В менеджере файлов *Midnight Commander* переносим папку lab05 в папку рабочей директории согласно файловой структуре наших лабораторных работ

Перемещение папки

8. После создания данного отчёта все изменения перемещаются в глобальный репозиторий на *Github*.

## 6 Выводы

В ходе данной лабораторной работы мы научились перемещаться по файловой системе с помощью программы *Midnight Commander*, создавать папки и файлы, изменять и сохранять документы встроенным редактором *mcedit*, а также изучили работу инструкций **mov** и **int** языка Ассемблер.