

# **Лабораторная работа №6**

**Дисциплина: Архитектура компьютера**

Ли Евгения Олеговна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

4.1	Midnight Commander . . . . .	8
4.2	Окно Midnight Commander. Смена текущего каталога . . . . .	8
4.3	Окно Midnight Commander. Создание каталога . . . . .	9
4.4	Окно Midnight Commander. Создание файла . . . . .	9
4.5	Окно Midnight Commander. Редактор mcedit . . . . .	9
4.6	Листинг 6.1 . . . . .	10
4.7	F3 . . . . .	10
4.8	Создала исполняемый файл . . . . .	10
4.9	Скачала файл in_out.asm . . . . .	11
4.10	Окно Midnight Commander. Копирование файла . . . . .	11
4.11	Окно Midnight Commander. Создание копии файла . . . . .	12
4.12	Создала исполняемый файл . . . . .	12
4.13	Текст программы . . . . .	13
4.14	Копию файла . . . . .	13
4.15	Исполняемый файл . . . . .	14
4.16	Копия файла . . . . .	14
4.17	исполняемый файл . . . . .	14

## Список таблиц

# 1 Цель работы

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

## 2 Задание

Приобрести практические навыки работы в Midnight Commander. Освоить инструкции языка ассемблера `mov` и `int`.

### 3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc - файловый менеджер.

Программа на языке ассемблера NASM состоит из трёх секций:

секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится

память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

## 4 Выполнение лабораторной работы

1. Открыла Midnight Commander(рис. 4.1)

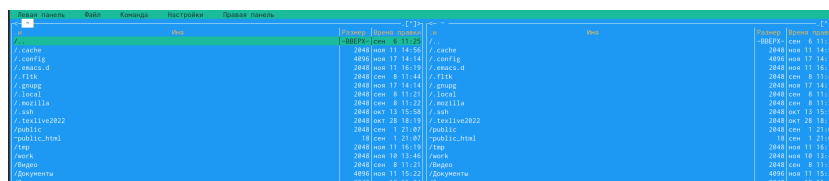


Рис. 4.1: Midnight Commander

2. Пользуясь клавишами перешла в каталог ~/work/arch-рс созданный при выполнении лабораторной работы No5 (рис. 4.2)

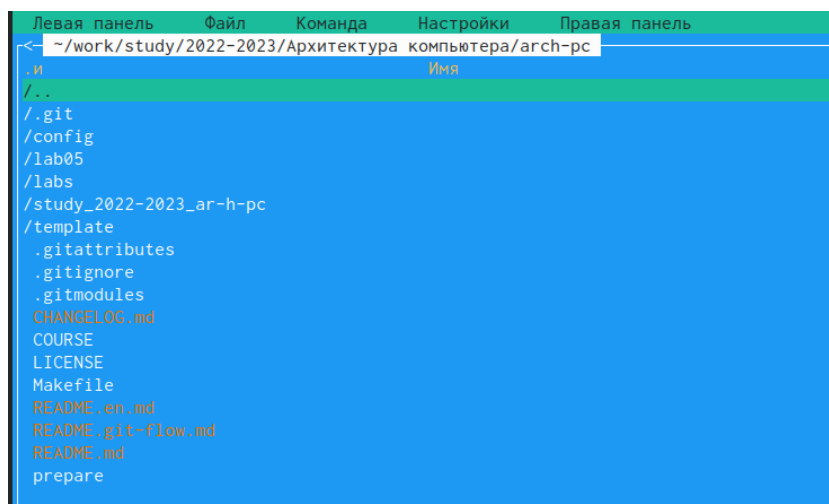


Рис. 4.2: Окно Midnight Commander. Смена текущего каталога

3. С помощью функциональной клавиши F7 создала папку lab06 и перешла в созданный каталог. (рис. 4.3)



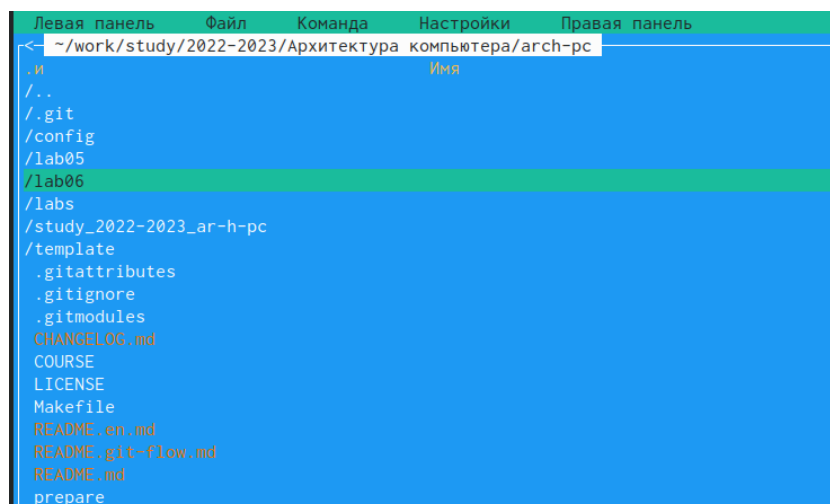


Рис. 4.3: Окно Midnight Commander. Создание каталога

4. Пользуясь строкой ввода и командой touch создала файл lab6-1.asm (рис. 4.4)

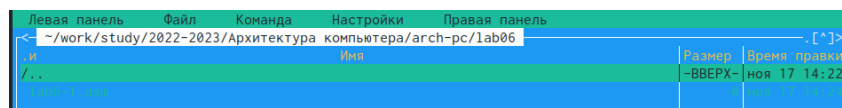


Рис. 4.4: Окно Midnight Commander. Создание файла

5. С помощью функциональной клавиши F4 открыла файл lab6-1.asm для редактирования во встроенном редакторе. (рис. 4.5)

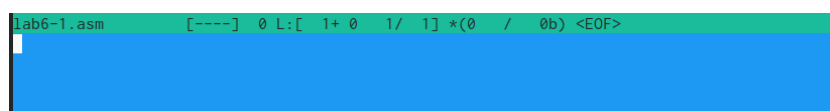


Рис. 4.5: Окно Midnight Commander. Редактор mcedit

6. Ввела текст программы из листинга 6.1 , сохранила изменения и закрыла файл.(рис. 4.6)

```

lab6-1.asm [M-2 0 KiB 1+36 37/37] (2449/2449) <edit>
-----
Программа выводит сообщение на экран и ждет строку с клавиатуры.
-----
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку:", 10, " сообщение лмс
; Символ перевода строки
orglen: EQU $-msg ; Длина переменных '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 переменных 'msg' длиной 'orglen'
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Файловый дескриптор stdout - стандартный вывод
mov ecx, msg ; Адрес строки 'msg' в 'ecx'
mov edx, orglen ; 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 вводную строку
mov edx, 80 ; Длина вводной 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. 4.6: Листинг 6.1

7. С помощью функциональной клавиши F3 открыла файл lab6-1.asm для просмотра. Убедилась, что файл содержит текст программы.(рис. 4.7)

```

~/fsf/dk.scl.pfu.edu.ru/home/e/o/eol/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06/lab6-1.asm
-----
Программа выводит сообщение на экран и ждет строку с клавиатуры.
-----
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку:", 10, " сообщение лмс
; Символ перевода строки
orglen: EQU $-msg ; Длина переменных '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 переменных 'msg' длиной 'orglen'
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Файловый дескриптор stdout - стандартный вывод
mov ecx, msg ; Адрес строки 'msg' в 'ecx'
mov edx, orglen ; 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 вводную строку
mov edx, 80 ; Длина вводной 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. 4.7: F3

8. Оттранслировала текст программы lab6-1.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл. На запрос 'Введите строку:' ввела ФИО. (рис. 4.8)

```

eol1@dk6n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ touch lab6-1.asm
eol1@dk6n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-1.asm
lab6-1.asm:26: warning: label alone on a line without a colon might be in error [-w+label-orphan]
eol1@dk6n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
eol1@dk6n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-1
Введите строку:
Li Evgenia Olegovna

```

Рис. 4.8: Создала исполняемый файл

### 6.3.1. Подключение внешнего файла in\_out.asm

9. Скачай файл in\_out.asm со страницы курса в ТУИС.(рис. 4.9)



Рис. 4.9: Скачала файл in\_out.asm

10. Подключаемый файл in\_out.asm переместила в каталог, что и файл с программой, в которой он используется (рис. 4.10)

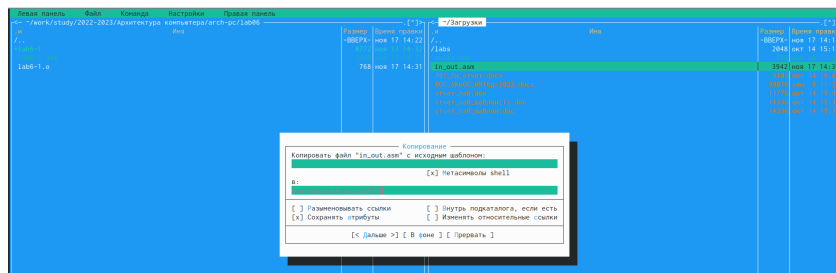


Рис. 4.10: Окно Midnight Commander. Копирование файла

11. С помощью функциональной клавиши F6 создала копию файла lab6-1.asm с именем lab6-2.asm. Выделила файл lab6-1.asm, нажала клавишу F6 , ввела имя файла lab6-2.asm (рис. 4.11)

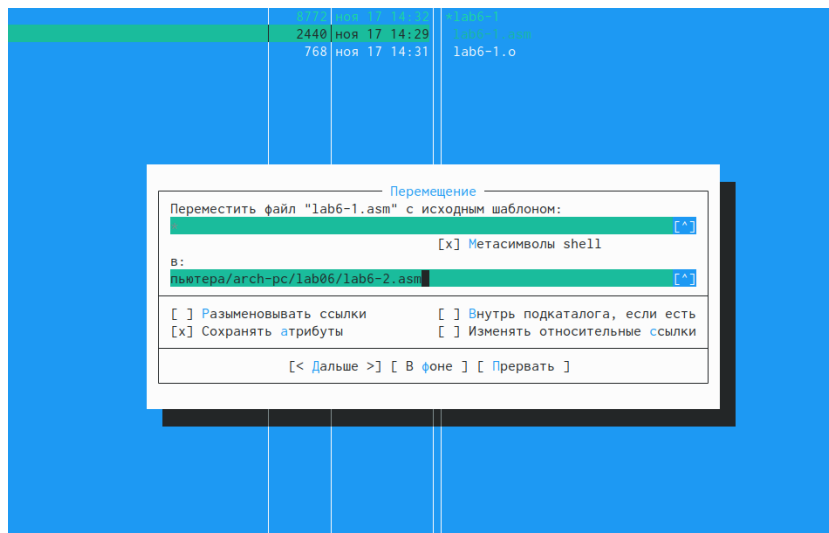


Рис. 4.11: Окно Midnight Commander. Создание копии файла

12. Исправила текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 6.2. Создала исполняемый файл и проверила его работу.(рис. 4.12)

```
eol1@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-2.asm
eol1@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
eol1@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-2
Введите строку: L1 Evgenia Olegovna
```

Рис. 4.12: Создала исполняемый файл

13. В файле lab6-2.asm заменила подпрограмму sprintLF на sprint. Создала исполняемый файл и проверила его работу. (рис. 4.13)



```
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-1.1.asm
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1.1 lab6-1.1.o
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-1.1
Введите строку:
LI
```

Рис. 4.15: Исполняемый файл

3. Создала копию файла lab6-2.asm. Исправила текст программы с использованием подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по требуемому алгоритму(рис. 4.16

Копирование

Копировать файл "lab6-2.asm" с исходным шаблоном:

☒ [x] Метасимволы shell

В:

пьютера/arch-pc/lab06/lab6-2.1.asm

☐ Разыменовывать ссылки
☐ Внутрь подкаталога, если есть

☒ Сохранять атрибуты
☐ Изменять относительные ссылки

[< Дальше >] [ В фоне ] [ Прервать ]

Рис. 4.16: Копия файла

4. Создала исполняемый файл и проверила его работу (рис. 4.17

```
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ nasm -f elf lab6-2.1.asm
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2.1 lab6-2.1.o
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab06 $ ./lab6-2.1
Введите строку:
LI
```

Рис. 4.17: исполняемый файл

## 5 Выводы

Я приобрела практические навыки работы в Midnight Commander. Освоила инструкции языка ассемблера mov и int.

## **Список литературы**