

# **Отчет по лабораторной работе №5**

**Основы работы с Midnight Commander (mc)**

Карапетян Мари Рафаеловна

# 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>Выполнение самостоятельной работы</b>	<b>12</b>
<b>6</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

## Список иллюстраций

4.1	Открытие файла и ввод текста . . . . .	8
4.2	Оттранслирование, компоновка и запуск файла, ввод данных . . .	9
4.3	Подключение внешнего файла . . . . .	9
4.4	Создание и редактирование файла . . . . .	10
4.5	Оттранслирование, компоновка и запуск файла, ввод данных . . .	10
4.6	Копирование файла . . . . .	11
5.1	Копирование файла . . . . .	12
5.2	Изменение программы . . . . .	13
5.3	Запуск программы . . . . .	13
5.4	Копирование файла . . . . .	14
5.5	Изменение программы . . . . .	15
5.6	Запуск программы . . . . .	15

## Список таблиц

# 1 Цель работы

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

## 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

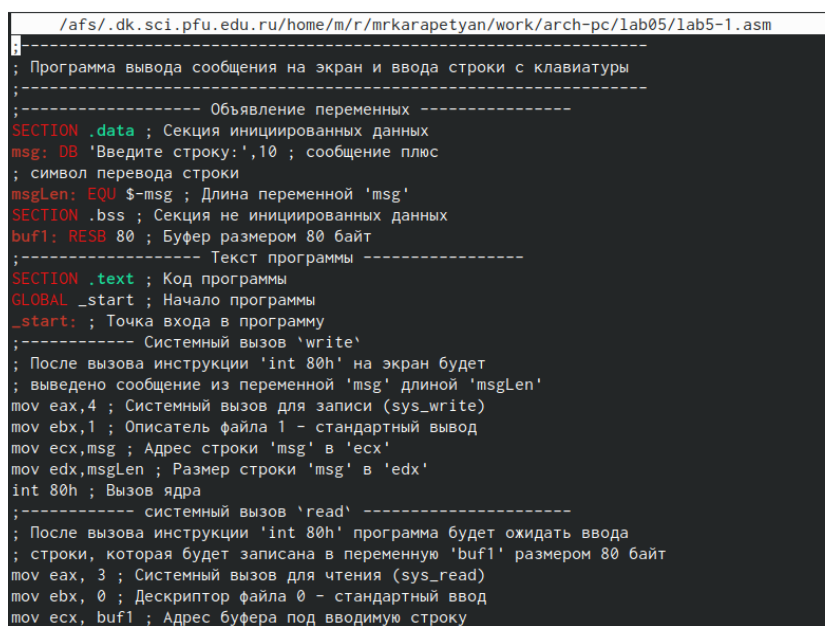
### 3 Теоретическое введение

5.2.1. Основы работы с Midnight Commander Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter (рис. 5.1). В Midnight Commander используются функциональные клавиши F1 — F10, к которым привязаны часто выполняемые операции (табл. 5.1).

5.2.2. Структура программы на языке ассемблера NASM Программа на языке ассемблера NASM, как правило, 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

С помощью команды `mc` открыла Midnight Commander, нашла файл `lab5-1.asm`, зашла в него и ввела нужный текст (Рис. @fig:001)



```
/afs/.dk.sci.pfu.edu.ru/home/m/r/mrkarapetyan/work/arch-pc/lab05/lab5-1.asm
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;----- Объявление переменных -----
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
;----- 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 ; 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
```

Рис. 4.1: Открытие файла и ввод текста

Оттранслируйте текст программы `lab5-1.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл и ввела свои имя и фамилию (Рис. @fig:002)



```
mrkarapetyan@dk5n59 ~ $ mc
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Mari Karapetyan
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $
```

Рис. 4.2: Оттранслирование, компоновка и запуск файла, ввод данных

### Подключение внешнего файла in\_out.asm(Рис.@fig:003)

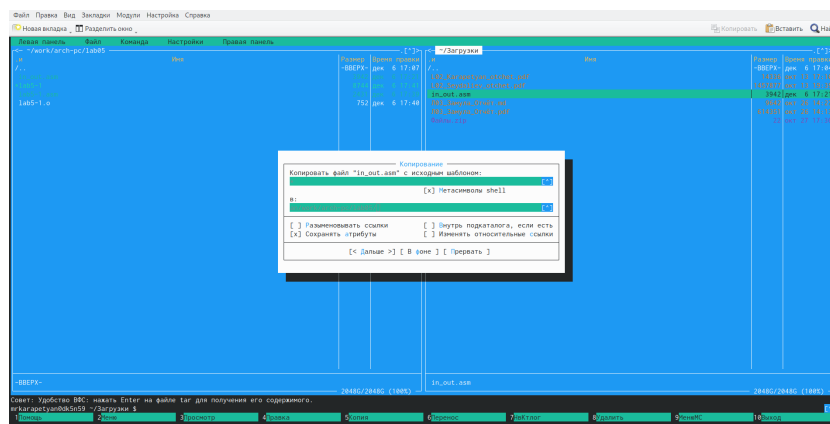


Рис. 4.3: Подключение внешнего файла

Создала копию файла lab5-1.asm с именем lab5-2.asm и редоктировала в нем текст(Рис.@fig:004)

```
/afs/.dk.sci.pfu.edu.ru/home/m/r/mrkarapetyan/work/arch-pc/lab05/lab5-2.asm
;-----
; 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 sprint ; 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. 4.4: Создание и редактирование файла

Оттранслируйте текст программы lab5-2.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл и ввела свои имя и фамилию(Рис.@fig:005)

```
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Karapetyan Mari
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $
```

Рис. 4.5: Оттранслирование,компоновка и запуск файла,ввод данных

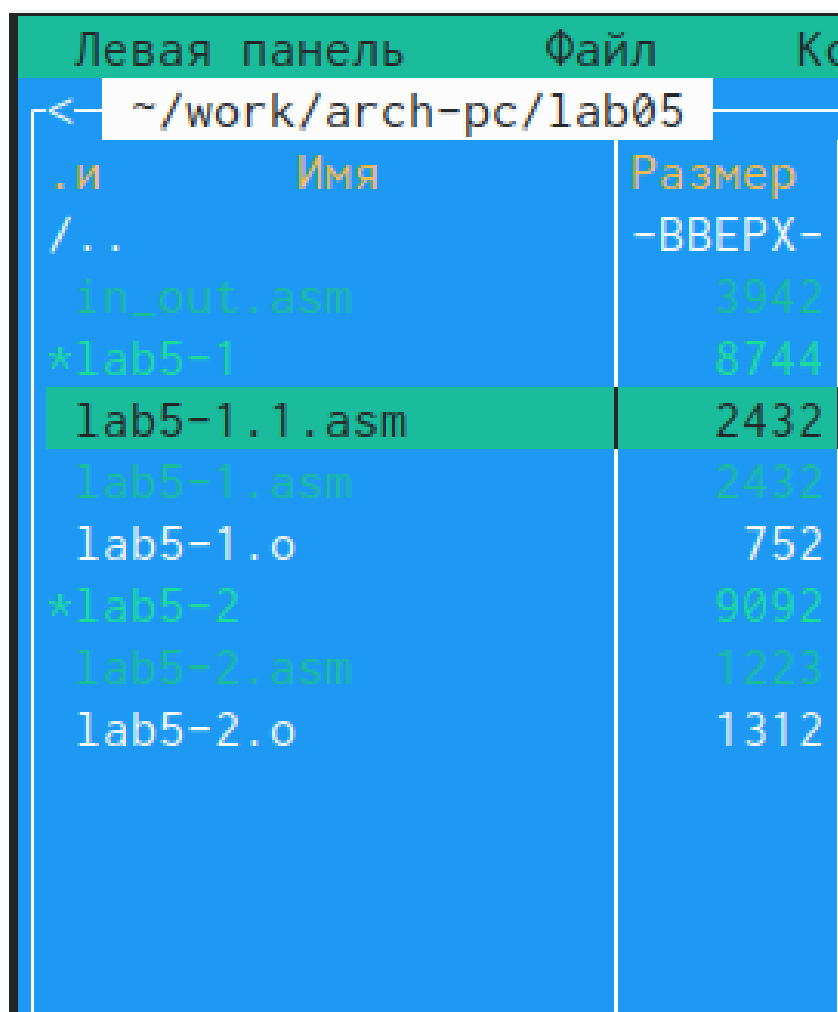
Копирую файл lab5-1.asm с именем lab5-1.1.asm(Рис.@fig:006)

Левая панель		Файл	Код
<		~/work/arch-pc/lab05	
.и	Имя	Размер	
/..		-ВВЕРХ-	
in_out.asm		3942	
*lab5-1		8744	
lab5-1.1.asm		2432	
lab5-1.asm		2432	
lab5-1.o		752	
*lab5-2		9092	
lab5-2.asm		1223	
lab5-2.o		1312	

Рис. 4.6: Копирование файла

## 5 Выполнение самостоятельной работы

Копирую файл lab5-1.asm с именем lab5-1.1.asm(Рис.@fig:006)



Левая панель		Файл	Код
<		~/work/arch-pc/lab05	
.и	Имя	Размер	
/..		-ВВЕРХ-	
	in_out.asm	3942	
	*lab5-1	8744	
	lab5-1.1.asm	2432	
	lab5-1.asm	2432	
	lab5-1.o	752	
	*lab5-2	9092	
	lab5-2.asm	1223	
	lab5-2.o	1312	

Рис. 5.1: Копирование файла

Изменяю код программы,добавляя вывод введенной строки(Рис.@fig:007)

```

/afs/.dk.sci.pfu.edu.ru/home/m/r/mrkarapetyan/work/arch-pc/lab05/lab5-1.1.asm
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 ; 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
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,buf1
int 80h
;----- 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. 5.2: Изменение программы

Создаю объектный файл lab5-1.1.o,компоную его в исполняемый файл,запускаю исполняемый файл(Рис.@fig:008)

```

mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ mc
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.1.asm
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1.1 lab5-1.1.o
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ./lab5-1.1
Введите строку:
Karapetyan Mari
Karapetyan Mari
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ █

```

Рис. 5.3: Запуск программы

Копирую файл lab5-2.asm с именем lab5-2.2.asm(Рис.@fig:009)

Левая панель		Файл	Код
< ~/work/arch-pc/lab05			
.и	Имя	Размер	
/..		-BBEPX-	
in_out.asm		3942	
*lab5-1		8744	
*lab5-1.1		8748	
lab5-1.1.asm		2488	
lab5-1.1.o		784	
lab5-1.asm		2432	
lab5-1.o		752	
*lab5-2		9092	
lab5-2.2.asm		1223	
lab5-2.asm		1223	
lab5-2.o		1312	

Рис. 5.4: Копирование файла

Изменяю код программы, добавляя вывод введенной строки (Рис. @fig:0010)

```
/afs/.dk.sci.pfu.edu.ru/home/m/r/mrkarapetyan/work/arch-pc/lab05/lab5-2.2.asm
;-----
; 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 sprint ; 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 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. 5.5: Изменение программы

Создаю объектный файл lab5-2.2.o, компоную его в исполняемый файл, запускаю исполняемый файл (Рис. @fig:0011)

```
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ mc
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.1.asm
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1.1 lab5-1.1.o
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $ ./lab5-1.1
Введите строку:
Karapetyan Mari
Karapetyan Mari
mrkarapetyan@dk5n59 ~/work/arch-pc/lab05 $
```

Рис. 5.6: Запуск программы

## 6 Выводы

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



## Список литературы

GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix). Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science). Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).