

Отчёт по лабораторной работе №5

Основы работы с Midnight Commander (mc), структура программы на языке ассемблера NASM и системные вызовы в ОС GNU Linux

Степан Михайлович Токаев

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	14

Список иллюстраций

2.1	Открытие Midnight Commander	6
2.2	Создание папки lab05	6
2.3	Создание файла lab5-1.asm	7
2.4	Ввод текст программы	7
2.5	Проверка	8
2.6	Запуск файла	8
2.7	Загрузка и копирование файла	9
2.8	Копирование файла	9
2.9	Текст программы	9
2.10	Запуск файла	10
2.11	Изменение	10
2.12	Запуск файла	10
2.13	Копия	11
2.14	Изменения	11
2.15	Запуск файла	12
2.16	Копия	12
2.17	Изменение	12
2.18	Запуск файла	13

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Открываем Midnight Commander.

```
smtokaev@dk3n40 ~ $ mc
```

Рис. 2.1: Открытие Midnight Commander

2. Переходим в каталог ~/work/arch-pc и создаем папку lab05.

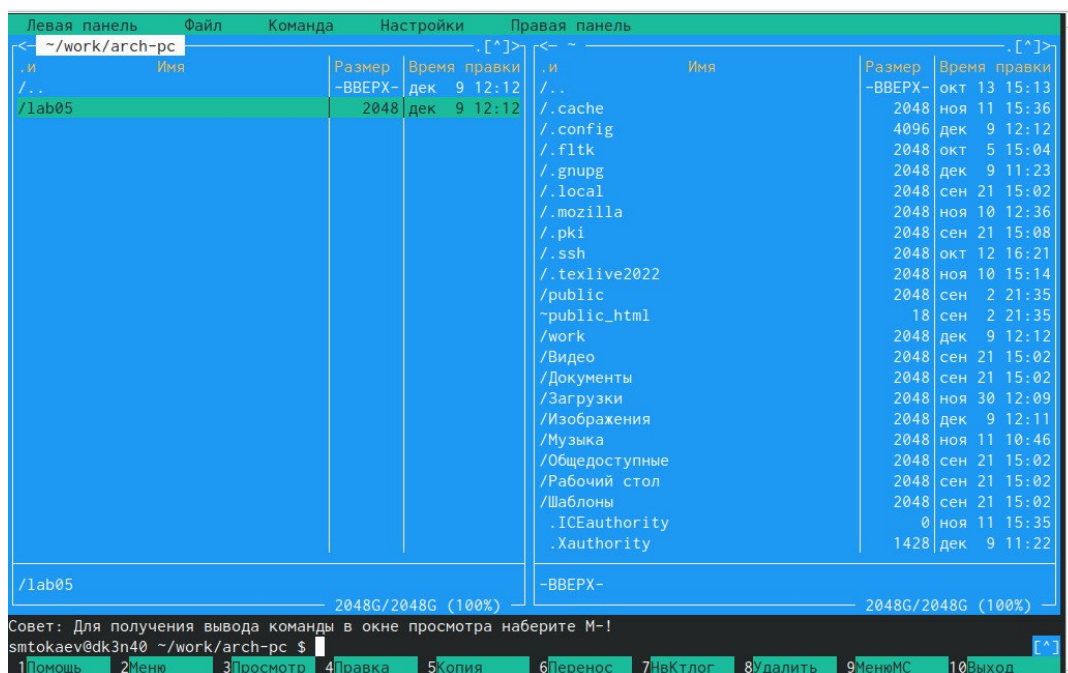


Рис. 2.2: Создание папки lab05

3. Создаем файл lab5-1.asm.



Рис. 2.3: Создание файла lab5-1.asm

4. Открываем файл lab5-1.asm для редактирования и вводим текст программы, сохраняем изменения и закрываем файл.

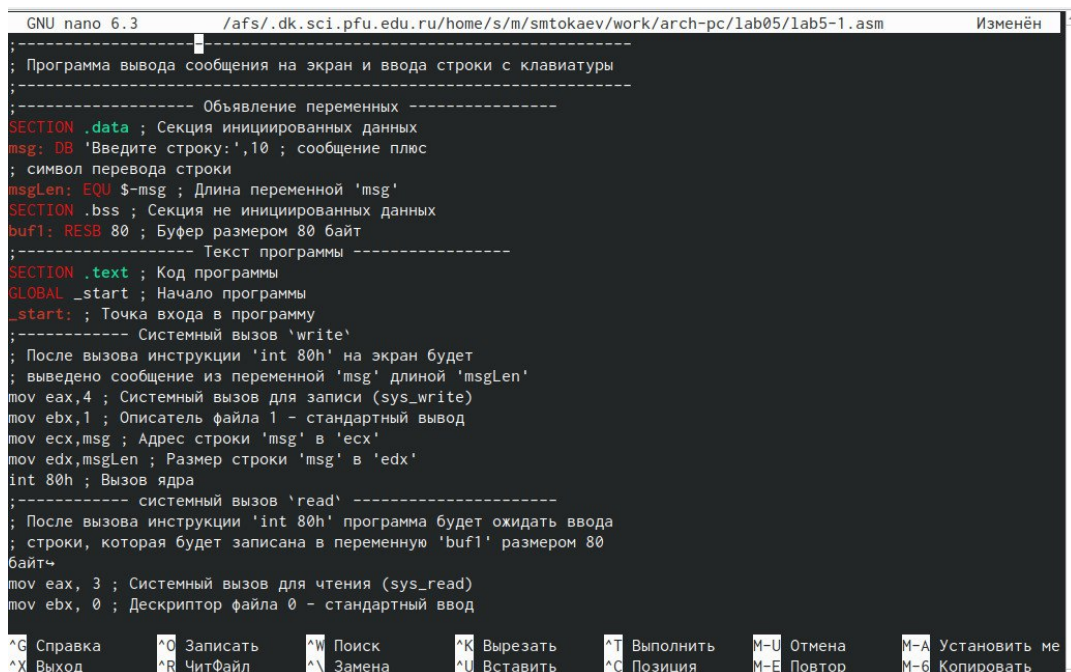


Рис. 2.4: Ввод текст программы

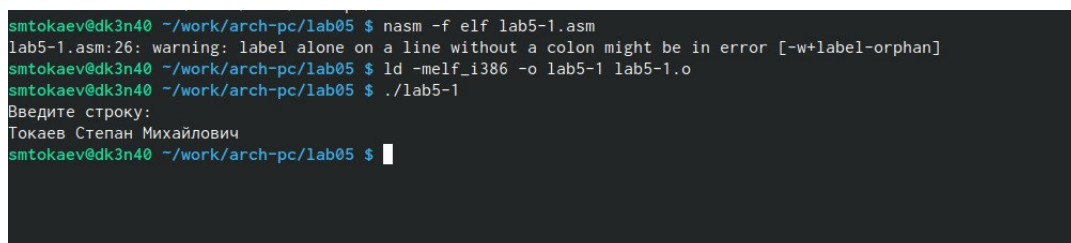
5. Открываем файл lab5-1.asm для просмотра. Убеждаемся, что файл содержит текст программы.



```
lab5-1.asm [----] 0 L: [ 1+ 0 1/ 37] *(0 /2435b) 0059 0x03B [*][X]
; 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
mov edx, 80 ; Длина вводимой строки
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пере-тить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 2.5: Проверка

6. Оттранслируем текст программы lab5-1.asm в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл.



```
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
lab5-1.asm:26: warning: label alone on a line without a colon might be in error [-w+label-orphan]
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ld -melf_i386 -o lab5-1 lab5-1.o
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Токаев Степан Михайлович
smtokaev@dk3n40 ~/work/arch-pc/lab05 $
```

Рис. 2.6: Запуск файла

7. Скачаем файл in_out.asm со страницы курса в ТУИС и скопируем файл in_out.asm в каталог с файлом lab5-1.asm.

.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
/..	-ВВЕРХ-	дек	9 12:12	/..	-ВВЕРХ-	дек	9 12:12
in_out.asm	1940	ноя	23 11:43	in_out.asm	1940	ноя	23 11:43
*lab5-1	8772	дек	9 12:18	*lab5-1	8772	дек	9 12:18
lab5-1.asm	2435	дек	9 12:18	lab5-1.asm	2435	дек	9 12:18
lab5-1.o	768	дек	9 12:17	lab5-1.o	768	дек	9 12:17

Рис. 2.7: Загрузка и копирование файла

8. Создаем копию файла lab5-1.asm с именем lab5-2.asm.

.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
/..	-ВВЕРХ-	дек	9 12:12	/..	-ВВЕРХ-	дек	9 12:12
in_out.asm	1940	ноя	23 11:43	in_out.asm	1940	ноя	23 11:43
*lab5-1	8772	дек	9 12:18	*lab5-1	8772	дек	9 12:18
lab5-1.asm	2435	дек	9 12:16	lab5-1.asm	2435	дек	9 12:18
lab5-1.o	768	дек	9 12:17	lab5-1.o	768	дек	9 12:17

Копирование

Копировать файл "lab5-1.asm" с исходным шаблоном:

[x] Метасимволы shell

В:

work/arch-pc/lab05/lab5-2.asm

☐ Разыменовывать ссылки
☒ Сохранять атрибуты

☐ Внутрь подкаталога, если есть
☐ Изменять относительные ссылки

Рис. 2.8: Копирование файла

9. Исправим текст программы в файле 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 sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 2.9: Текст программы

10. Создаем исполняемый файл и проверяем его работу.

```

smtokaev@dk3n40 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ./lab5-2
bash: ./lab5-2: команда не найдена
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Токаев Степан Михайлович
smtokaev@dk3n40 ~/work/arch-pc/lab05 $

```

Рис. 2.10: Запуск файла

11. Заменяем подпрограмму sprintLF на sprint в файле lab5-2.asm.

```

lab5-2.asm      [----] 11 L: [ 1+12 13/ 17] *(847 /1221b) 0059 0x03B      [*]
; 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 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call read ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 2.11: Изменение

12. Создаем исполняемый файл и проверяем его работу.

```

smtokaev@dk3n40 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2c.asm
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2c lab5-2c.o
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ./lab5-2c
Введите строку: Токаев

```

Рис. 2.12: Запуск файла

Разница в том, что подпрограмма sprintLF переводит сообщение на следующую строку, а sprint данного действия не делает.

13. Создаем копию файла lab5-1.asm и вносим изменения в программу.

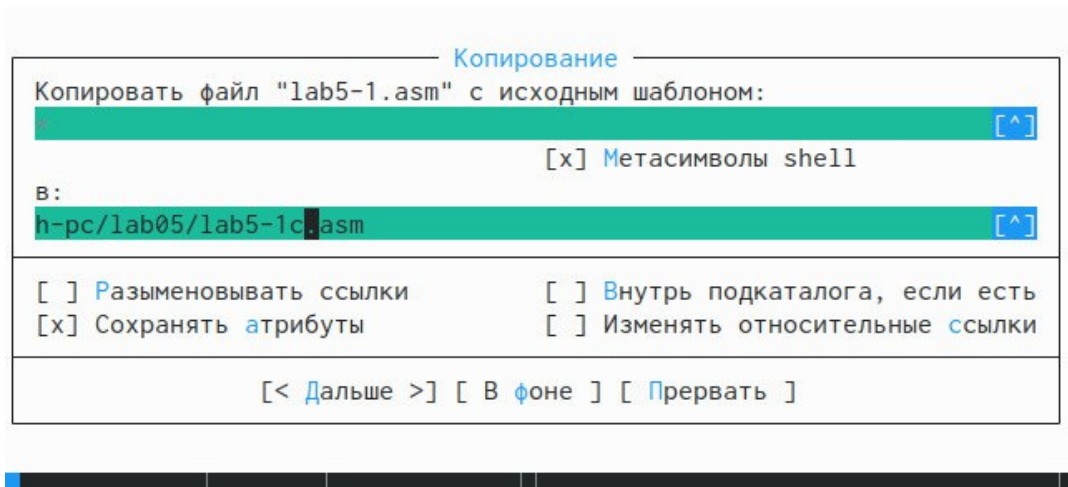


Рис. 2.13: Копия

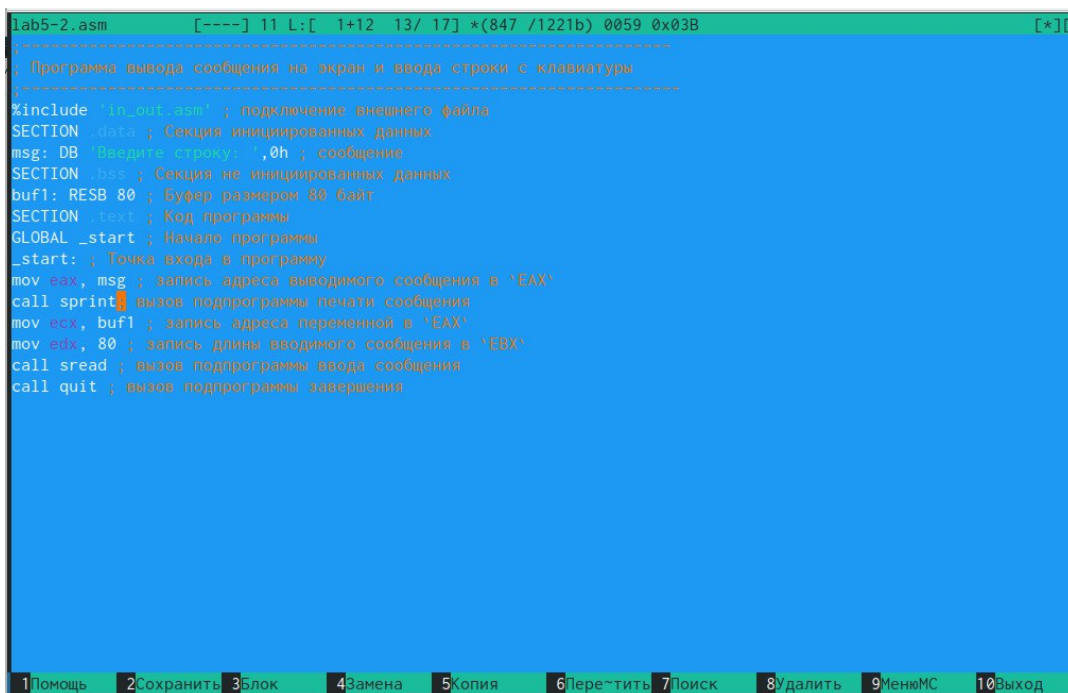


Рис. 2.14: Изменения

14. Получаем исполняемый файл и проверяем его работу.

```
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1c lab5-1c.o
smtokaev@dk3n40 ~/work/arch-pc/lab05 $ ./lab5-1c
Введите строку:
Токаев
```

Рис. 2.15: Запуск файла

15. Создаем копию файла lab5-2.asm и вносим изменения в программу.

Копирование

Копировать файл "lab5-2.asm" с исходным шаблоном:

h-pc/lab05/lab5-2c.asm

[^]

[x] Метасимволы shell

В:

h-pc/lab05/lab5-2c.asm

[^]

<input type="checkbox"/> Разыменовывать ссылки	<input type="checkbox"/> Внутри подкаталога, если есть
<input checked="" type="checkbox"/> Сохранять атрибуты	<input type="checkbox"/> Изменять относительные ссылки

[< Дальше >] [В фоне] [Прервать]

Рис. 2.16: Копия

Файл Правка Вид Закладки Модули Настройка Справка

Новая вкладка Разделить окно

lab5-2c.asm [----] 11 L:[1+12 13/ 17] *(847 /1221b) 0059 0x03B

```

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных 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. 2.17: Изменение

16. Получаем исполняемый файл и проверяем его работу.

```
smtokeev@dk3n40 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2c.asm
smtokeev@dk3n40 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2c lab5-2c.o
smtokeev@dk3n40 ~/work/arch-pc/lab05 $ ./lab5-2c
Введите строку: Токаев
```

Рис. 2.18: Запуск файла

3 Выводы

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