

Лабораторная работа №5

**Основы работы с МС (Midnight Commnader), структура программы
на языке ассемблера NASM, системные вызовы в ОС GNU Linux**

Приходько Иван Иванович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение задания для самостоятельной работы	13
4	Выводы	17

Список иллюстраций

2.1	Запуск Midnight Commander	6
2.2	Интерфейс Midnight Commander	6
2.3	Переход в каталог arch-rc	7
2.4	Создание файла lab5-1.asm	7
2.5	Редактирование файла lab5-1.asm	8
2.6	Компиляция lab5-1.asm	8
2.7	Сборка lab5-1.asm	8
2.8	Запуск lab5-1.asm	8
2.9	Результат lab5-1.asm	9
2.10	Открытие папки с файлом in_out.asm в правой панели	9
2.11	Копирование файла in_out.asm	9
2.12	Создание копии lab5-1.asm	10
2.13	Вид рабочей папки	10
2.14	Редактирование файла lab5-2.asm	11
2.15	Компиляция и сборка lab5-2.asm	11
2.16	Запуск lab5-2.asm	11
2.17	Изменение файла lab5-2.asm	12
2.18	Повторный запуск lab5-2.asm	12
3.1	Создание копии файла lab5-1.asm	13
3.2	Изменение файла lab5-1-1.asm	14
3.3	Создание и запуск lab5-1-1.asm	14
3.4	Создание копии файла lab5-2.asm	15
3.5	Изменение файла lab5-2-1.asm	15
3.6	Запуск lab5-2-1.asm	16

Список таблиц

1 Цель работы

Ознакомиться с программой Midnight commander и освоить написание программ на языке ассемблера с помощью инструкций `mov` и `int`

2 Выполнение лабораторной работы

Для начала запустим Midnight Commander (рис. 2.1).

```
ivanprihodko@fedora:~$ mc
```

Рис. 2.1: Запуск Midnight Commander

Мы увидим такой интерфейс (рис. 2.2).

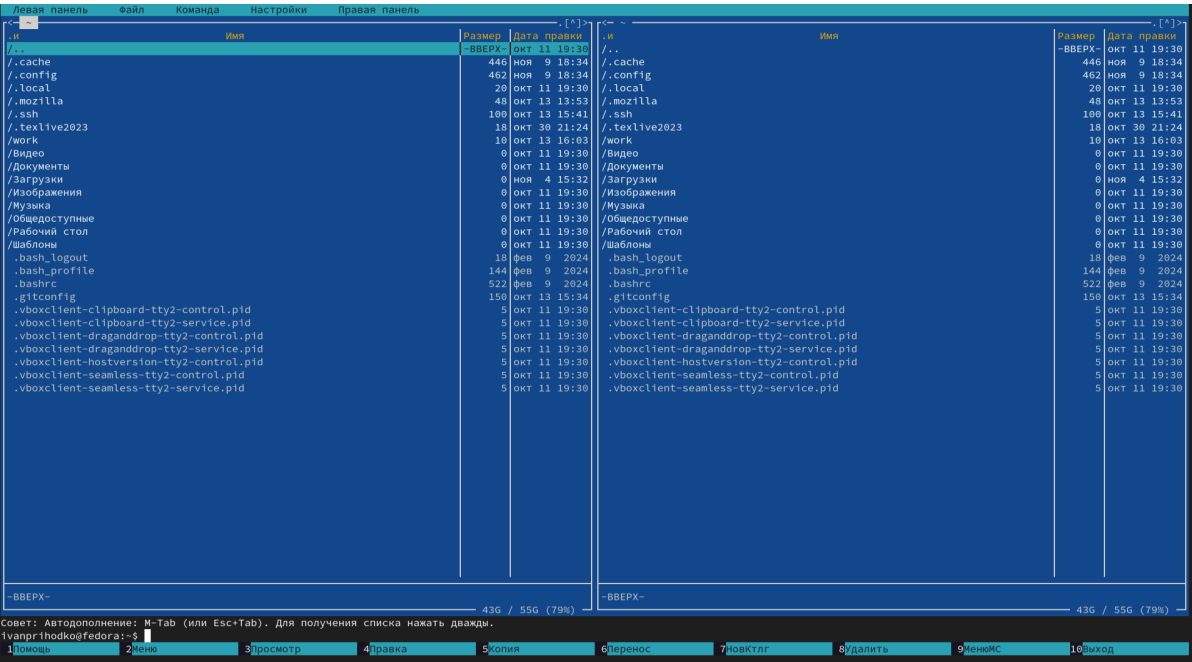


Рис. 2.2: Интерфейс Midnight Commander

Перейдем в каталог arch-rc (рис. 2.3).

.и	Имя	Размер	Дата правки	.и	Имя	Размер	Дата правки
./..	-BBEPX-	Окт 13 16:03		./..	-BBEPX-	Окт 11 19:30	
./git	218	ноя 4 16:04		./cache	446	ноя 9 18:34	
./config	24	Окт 13 16:03		./config	462	ноя 9 18:34	
./labs	152	Окт 30 21:00		./local	20	Окт 11 19:30	
./presentation	78	Окт 30 21:00		./mozilla	48	Окт 13 13:53	
./template	36	Окт 13 16:03		./ssh	100	Окт 13 15:41	
*.gitattributes	1765	Окт 13 15:58		./texlive2023	18	Окт 30 21:24	
*.gitignore	4637	Окт 13 15:58		./work	10	Окт 13 16:03	
*.gitmodules	278	Окт 13 15:58		./video	0	Окт 11 19:30	
*CHANGELOG.md	4786	Окт 13 15:58		./документы	0	Окт 11 19:30	
*COURSE	8	Окт 13 15:58		./загрузки	0	ноя 4 15:32	
*LICENSE	18657	Окт 13 15:58		./изображения	0	Окт 11 19:30	
*Makefile	988	Окт 30 21:00		./Музыка	0	Окт 11 19:30	
*Makefile (Копия)	988	Окт 13 15:58		./Общедоступные	0	Окт 11 19:30	
*README.en.md	152	Окт 13 15:58		./Рабочий стол	0	Окт 11 19:30	
*README.git-flow.md	5653	Окт 13 15:58		./Шаблоны	0	Окт 11 19:30	
*README.md	4304	Окт 13 15:58		./bash_logout	18	Фев 9 2024	
*prepare	0	Окт 30 21:00		./bash_profile	144	Фев 9 2024	
				./bashrc	522	Фев 9 2024	
				./gitconfig	150	Окт 13 15:34	
				./vboxclient-clipboard-tty2-control.pid	5	Окт 11 19:30	
				./vboxclient-clipboard-tty2-service.pid	5	Окт 11 19:30	
				./vboxclient-draganddrop-tty2-control.pid	5	Окт 11 19:30	
				./vboxclient-draganddrop-tty2-service.pid	5	Окт 11 19:30	
				./vboxclient-hostversion-tty2-control.pid	5	Окт 11 19:30	
				./vboxclient-seamless-tty2-control.pid	5	Окт 11 19:30	
				./vboxclient-seamless-tty2-service.pid	5	Окт 11 19:30	

Рис. 2.3: Переход в каталог arch-pc

Создадим с помощью touch файл lab5-1.asm (рис. 2.4).

.и	Имя	Размер	Дата правки	.и
./..	-BBEPX-	Окт 30 21:00		./
./presentation	100	Окт 30 21:00		./
./report	62	Окт 30 21:00		./
lab5-1.asm	0	ноя 9 18:39		./

Рис. 2.4: Создание файла lab5-1.asm

Вставим в него следующий код (рис. 2.5).

```
GNU nano 7.2 /home/ivanprihodko/work/study/2024-2025/Архите
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
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 ;Descriptor файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой 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. 2.5: Редактирование файла lab5-1.asm

Теперь скомпилируем и соберем его (рис. 2.6 и 2.7).

```
Совет: Требуется вставить литерал? Нажмите Control-q и литерал.
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f elf lab5-1.asm
```

Рис. 2.6: Компиляция lab5-1.asm

```
Совет: Требуется вставить литерал? Нажмите Control-q и литерал.
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

Рис. 2.7: Сборка lab5-1.asm

Запустим его (рис. 2.8 и 2.9).

```
Совет: Требуется вставить литерал? Нажмите Control-q и литерал.
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ./lab5-1
```

Рис. 2.8: Запуск lab5-1.asm


```
ivanprihodko@fedora:~/work/st
Введите строку:
Приходько Иван Иванович
```

Рис. 2.9: Результат lab5-1.asm

Теперь скачаем файл in_out.asm и откроем папку с ним в правой панели (рис. 2.10).

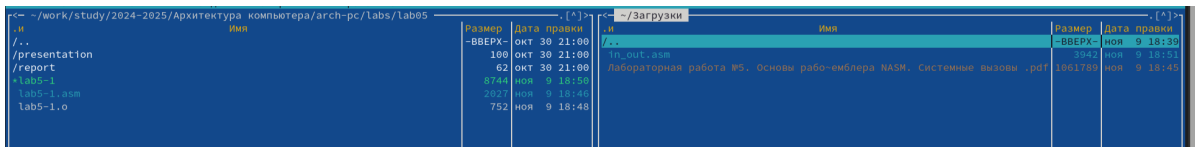


Рис. 2.10: Открытие папки с файлом in_out.asm в правой панели

Скопируем его в рабочую область (рис. 2.11).

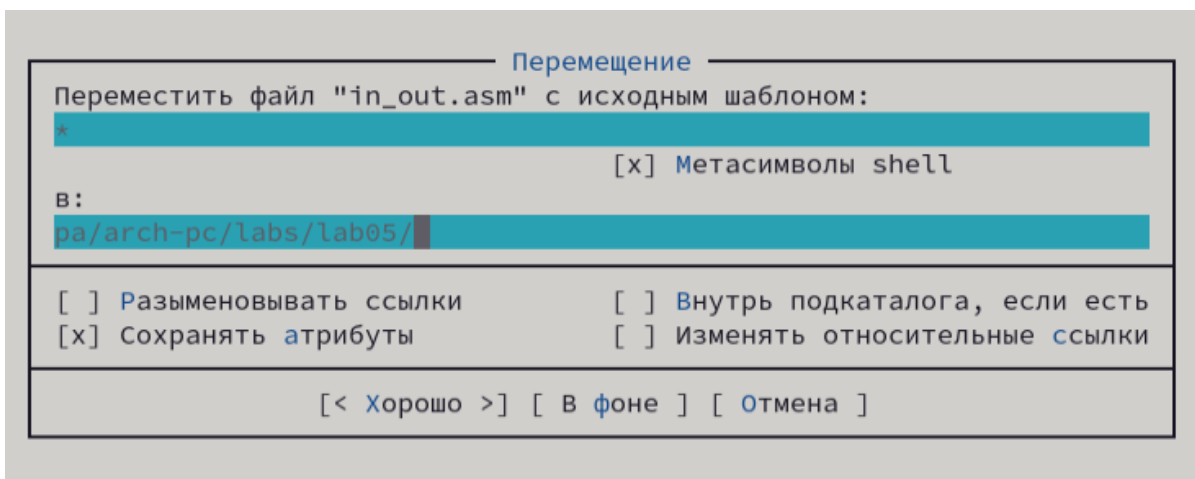


Рис. 2.11: Копирование файла in_out.asm

Теперь сделаем копию файла lab5-1.asm и назовём копию lab5-2.asm (рис. 2.12).

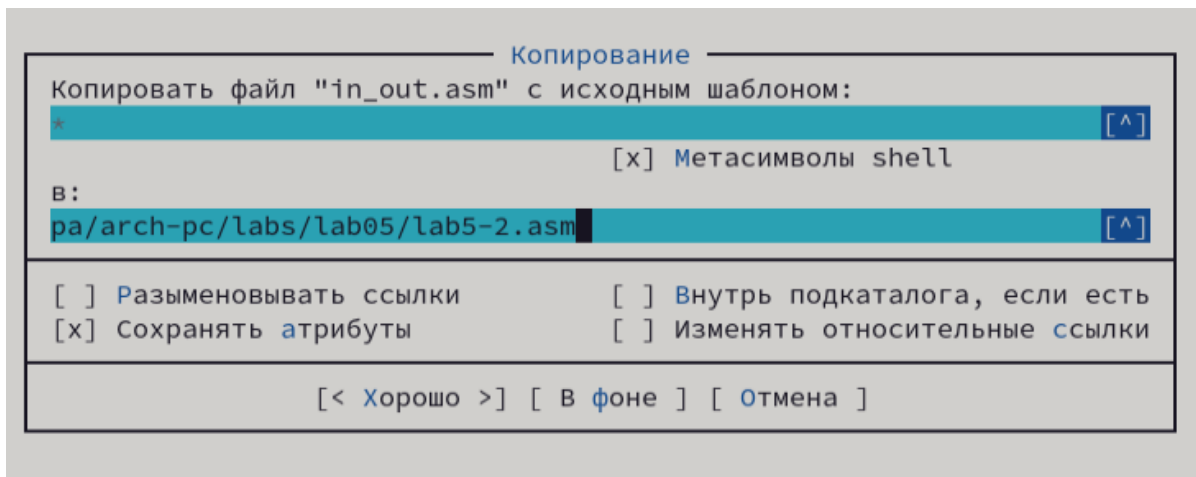


Рис. 2.12: Создание копии lab5-1.asm

Теперь наша папка выглядит так (рис. 2.13).

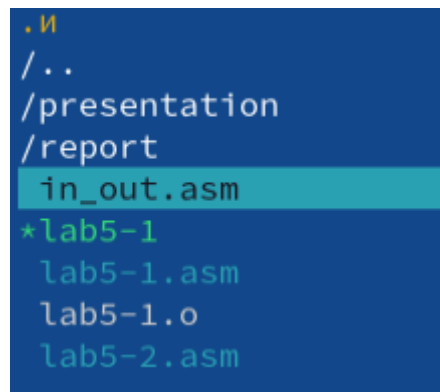


Рис. 2.13: Вид рабочей папки

Напишем в файл lab5-2.asm следующий код(рис. 2.14).

```
GNU nano 7.2 /home/ivanprihodko/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/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
buf: 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, buf ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины выводимого сообщения в 'EDX'
call read ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 2.14: Редактирование файла lab5-2.asm

Теперь скомпилируем, соберем и запустим его (рис. 2.15).

```
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f elf lab5-2.asm
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ./lab5-2
```

Рис. 2.15: Компиляция и сборка lab5-2.asm

```
Введите строку:
Приодько Иван Иванович
```

Рис. 2.16: Запуск lab5-2.asm

Попробуем теперь вместо команды sprintf использовать просто команду printf (рис. 2.17).

```
GNU nano 7.2 /home/ivai
;-----
; 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. 2.17: Изменение файла lab5-2.asm

Теперь запустим его (рис. 2.18).

```
ivanprikhodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ./lab5-2
Введите строку:
Приходько Иван Иванович
ivanprikhodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$
```

Рис. 2.18: Повторный запуск lab5-2.asm

3 Выполнение задания для самостоятельной работы

Создадим копию файла lab5-1.asm (рис. 3.1).

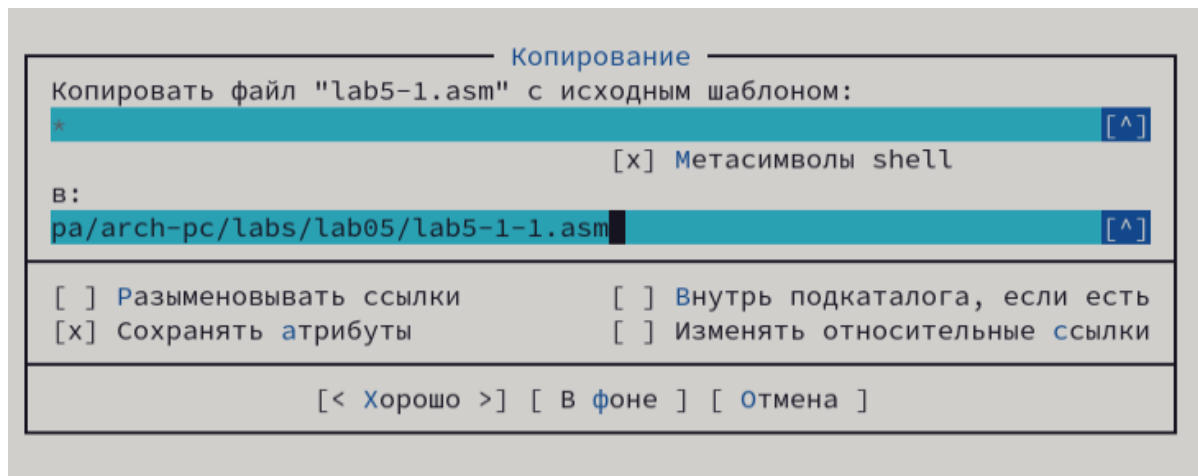


Рис. 3.1: Создание копии файла lab5-1.asm

Изменим копию так, чтобы она выводила тот текст, который получила на ввод (рис. 3.2).

```

GNU nano 7.2 /home/ivanprihodko/v
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- 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 ;Descriptor файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- 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,buf1 ; Адрес строки 'msg' в 'ecx'
mov edx,80 ; Размер строки 'msg' в 'edx'
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. 3.2: Изменение файла lab5-1-1.asm

Теперь создадим исполняем файл и и запустим его (рис. 3.3).

```

ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f elf lab5-1-1.asm
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ./lab5-1-1
Приходько Иван Иванович
Приходько Иван Иванович
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$

```

Рис. 3.3: Создание и запуск lab5-1-1.asm

Создадим копию файла lab5-2.asm (рис. 3.4).

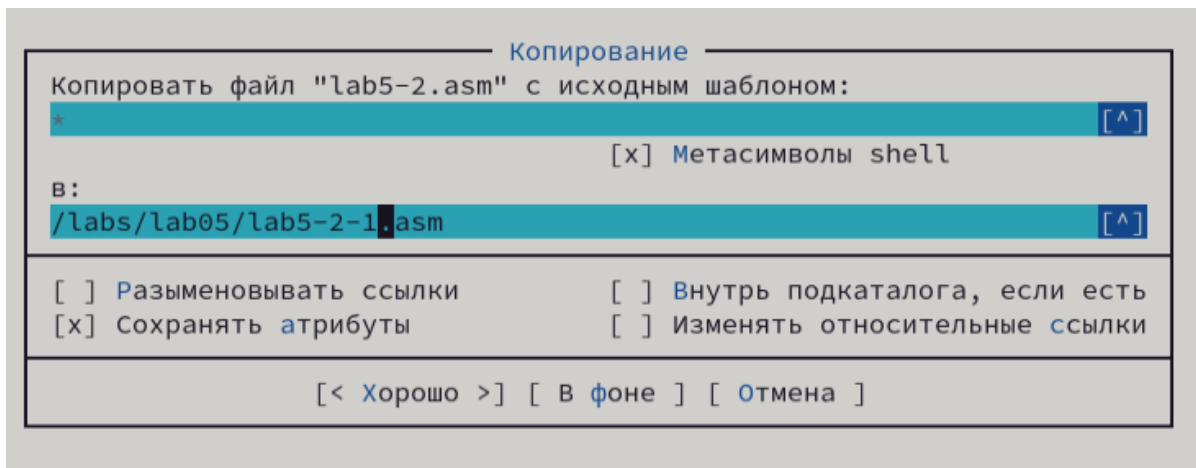


Рис. 3.4: Создание копии файла lab5-2.asm

Теперь сделаем так, чтобы этот код также выводил тот текст, что получит на ввод. Для этого перед последней строкой добавим строчку, которая записывает в `eax` адрес `buf1`, а также строчку, которая вызывает подпрограмму `sprintLF` (рис. 3.5).

```

;-----
; 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 sprintLF ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ;
mov eax, buf1 ;
call sprintLF ; вызов подпрограммы печати сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.5: Изменение файла lab5-2-1.asm

Теперь запустим программу и убедимся, что она работает (рис. 3.6).

```
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f elf lab5-2-1.asm
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$ ./lab5-2-1
Введите строку:
Приходько Иван Иванович
Приходько Иван Иванович
ivanprihodko@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab05$
```

Рис. 3.6: Запуск lab5-2-1.asm

4 Выводы

В результате выполнения работы были получены навыки работы с Midnight commander, а также навыки написания простых программ ввода-вывода на языке ассемблера