

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура вычислительных систем

Студент: Чистякова Дарья Владимировна

Группа: НБИбд-03-22

МОСКВА

2022 г.

Цель работы:

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

Ход работы:

1) Открываем Midnight Commander. (Рис. 1-2)



Рис. 1. Вызов Midnight Commander

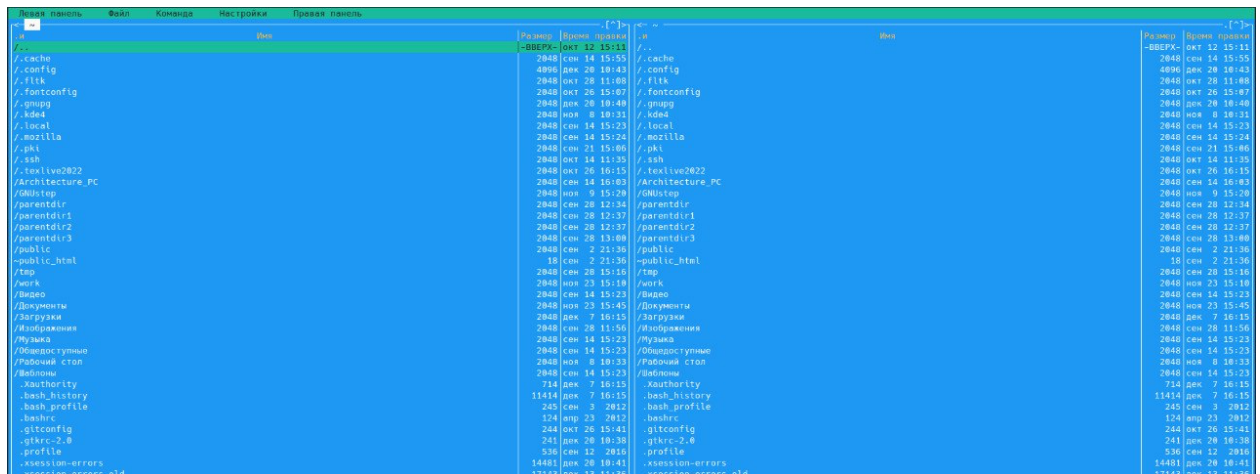


Рис. 2. Midnight Commander

2) Переходим в каталог ~/work/arch-pc. (Рис. 3)

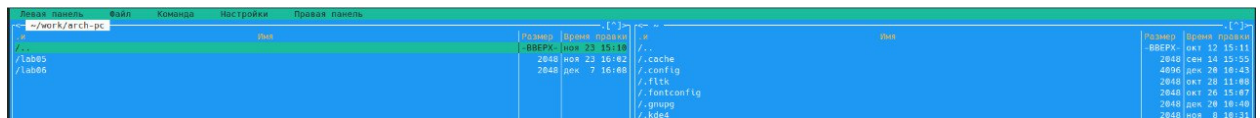


Рис. 3. Каталог

3) Создаем папку lab06. (Рис. 4)

И переходим в созданный каталог. (Рис. 5)

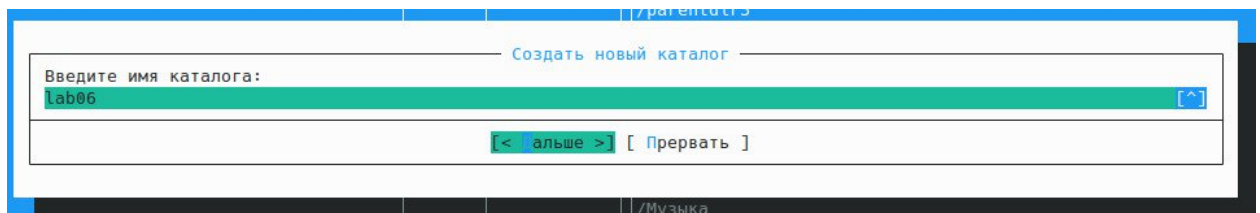


Рис. 4. Создание каталога

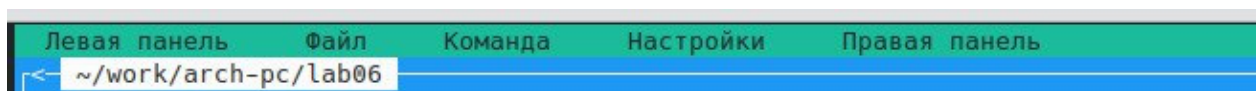
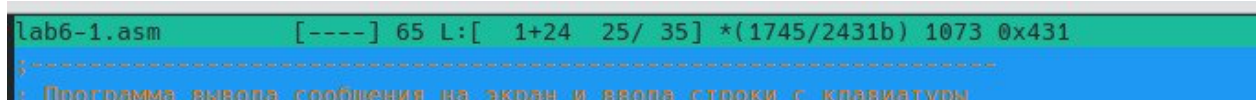


Рис. 5. Каталог

4) Создаем файл lab5-1.asm.

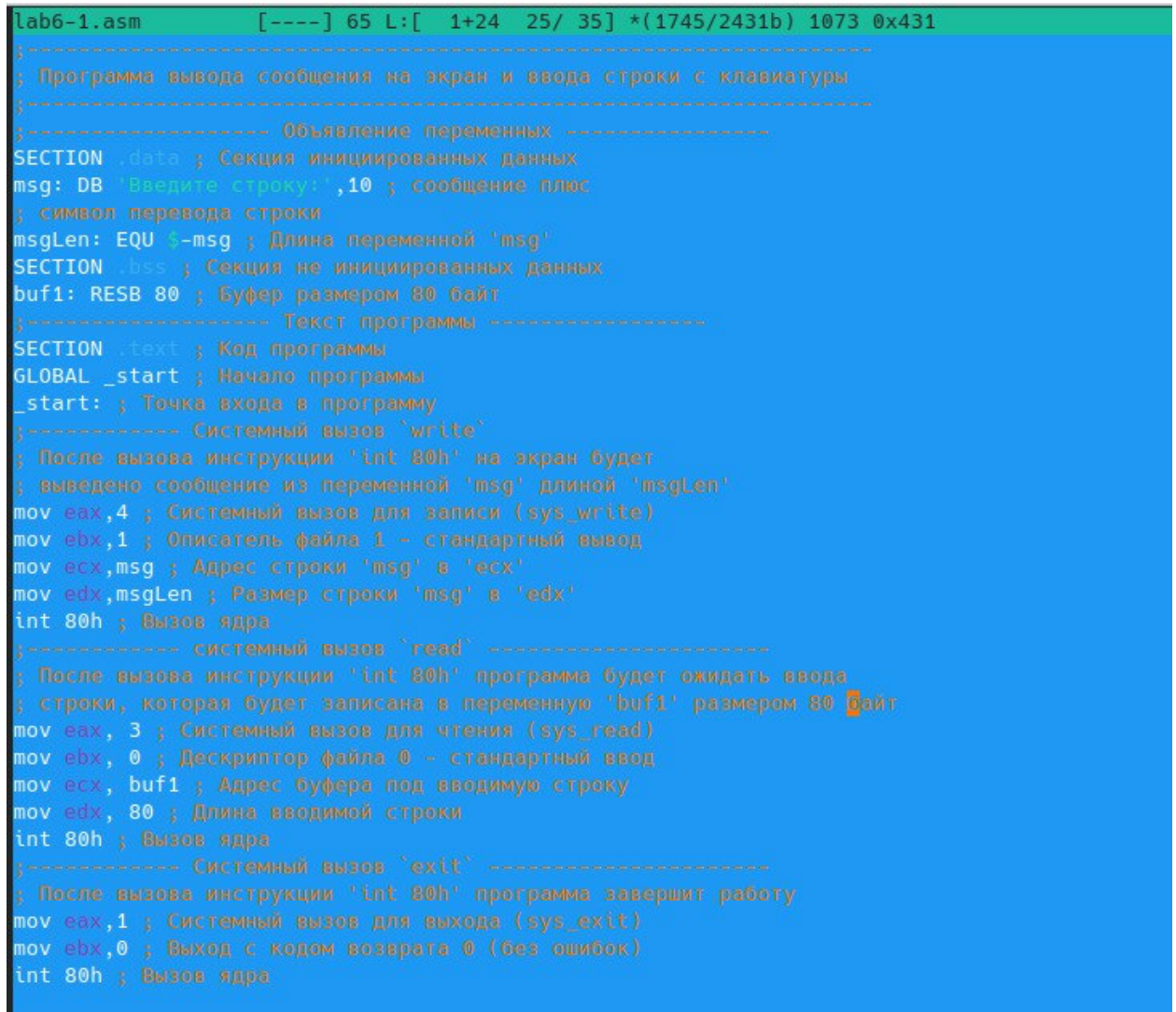
5) Открываем файл lab6-1.asm для редактирования. (Рис. 6)



```
lab6-1.asm [----] 65 L:[ 1+24 25/ 35] *(1745/2431b) 1073 0x431
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
```

Рис. 6. Открытие файла

6) Вводим текст программы, сохраняем изменения и закрываем файл. (Рис. 7)



```
lab6-1.asm [----] 65 L:[ 1+24 25/ 35] *(1745/2431b) 1073 0x431
;-----
; 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 ; Файловый дескриптор 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 ; Файловый дескриптор 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. 7. Программа

7) Открываем файл lab6-1.asm для просмотра. Убеждаемся, что файл содержит текст программы. (Рис. 8)

```
lab6-1s.asm      [----]  0 L:[ 1+ 0  1/ 35] *(0  /2431b) 0059 0x03B
;-----
; 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 ; Описание 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
;----- 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. 8. Программа

8) Оттранслируем текст программы lab6-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. (Рис. 9)

```
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ./lab6-1
Введите строку:
Чистякова Дарья Владимировна
```

Рис. 9. Запуск файла

5.3.1. Подключение внешнего файла in_out.asm

9) Скачаем файл in_out.asm со страницы курса в ТУИС. (Рис. 10)



Рис. 10. Загрузка файла

10) Скопируем файл in_out.asm в каталог с файлом lab6-1.asm. (Рис. 11)

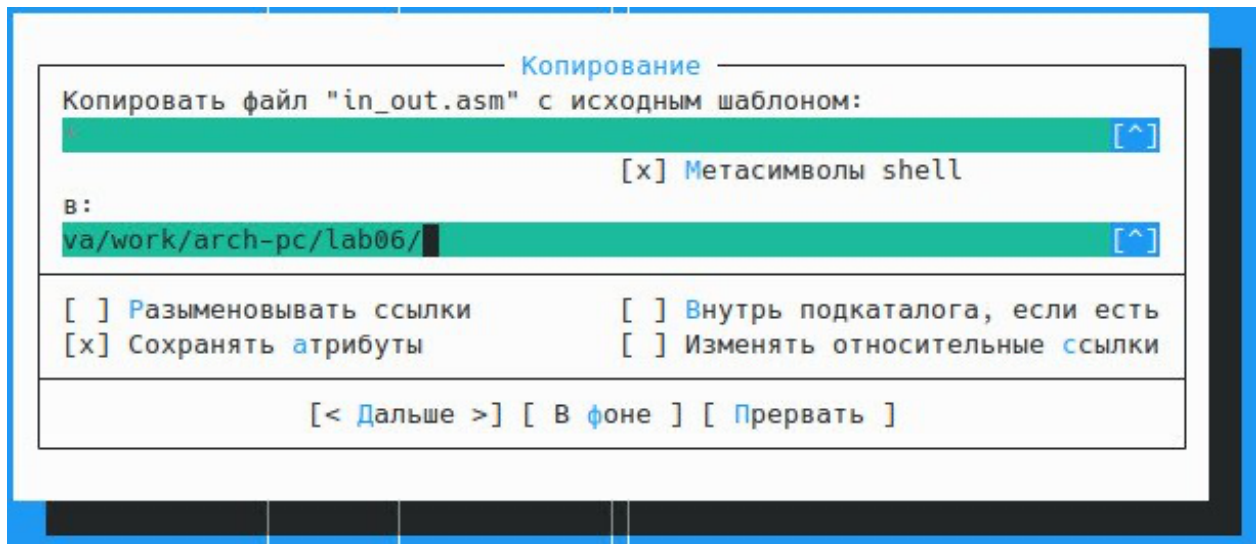


Рис. 11. Копирование файла

11) Создаем копию файла lab6-1.asm с именем lab6-2.asm. (Рис. 12)

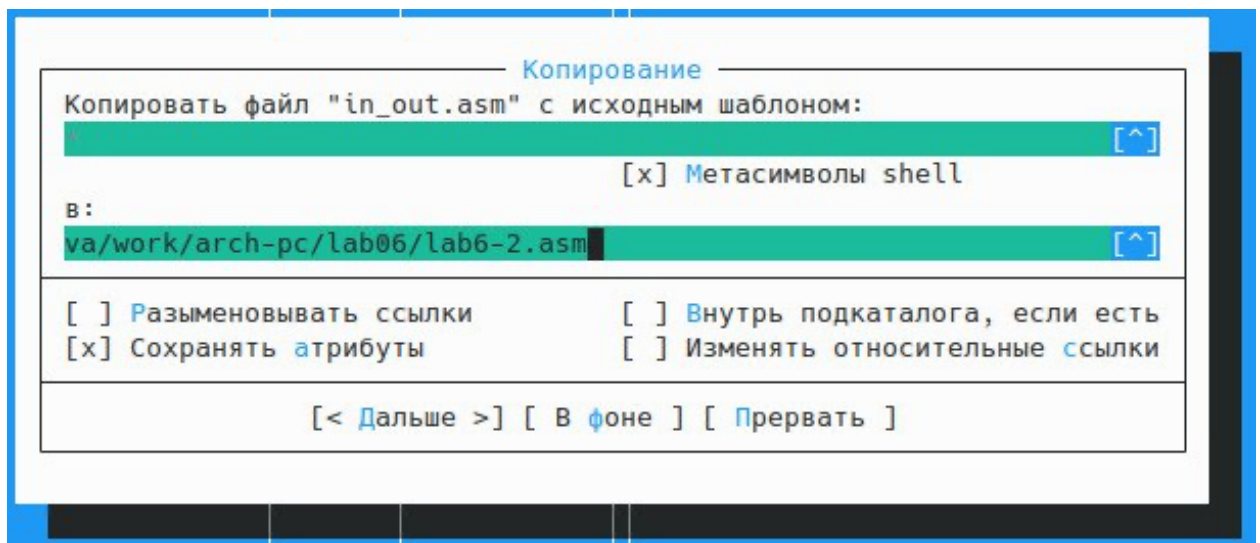


Рис. 12. Копия файла

12) Исправим текст программы в файле lab6-2.asm. (Рис. 13)

```
lab6-2.asm      [-M--] 41 L:[ 1+16 17/ 17] *(1224/1224b) <EOF>
;-----
; 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 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 13. Текст программы

13) Создаем исполняемый файл и проверяем его работу. (Рис. 14)

```
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:
Чистякова Дарья Владимировна
```

Рис. 14. Запуск файла

14) Заменяем подпрограмму sprintf на sprintf в файле lab6-2.asm. (Рис. 15)

```
lab6-2.asm      [----] 11 L:[ 1+12 13/ 17] *(847 /1222b) 0032 0x020
;-----
; 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 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 15. Изменения

15) Создаем исполняемый файл и проверяем его работу. (Рис. 16)

```
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку: Чистякова Дарья Владимировна
```

Рис. 16. Запуск файла

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

Задание для самостоятельной работы.

1) Создаем копию файла `lab6-1.asm`. (Рис. 17)

Вносим изменения в программу. (Рис. 18)

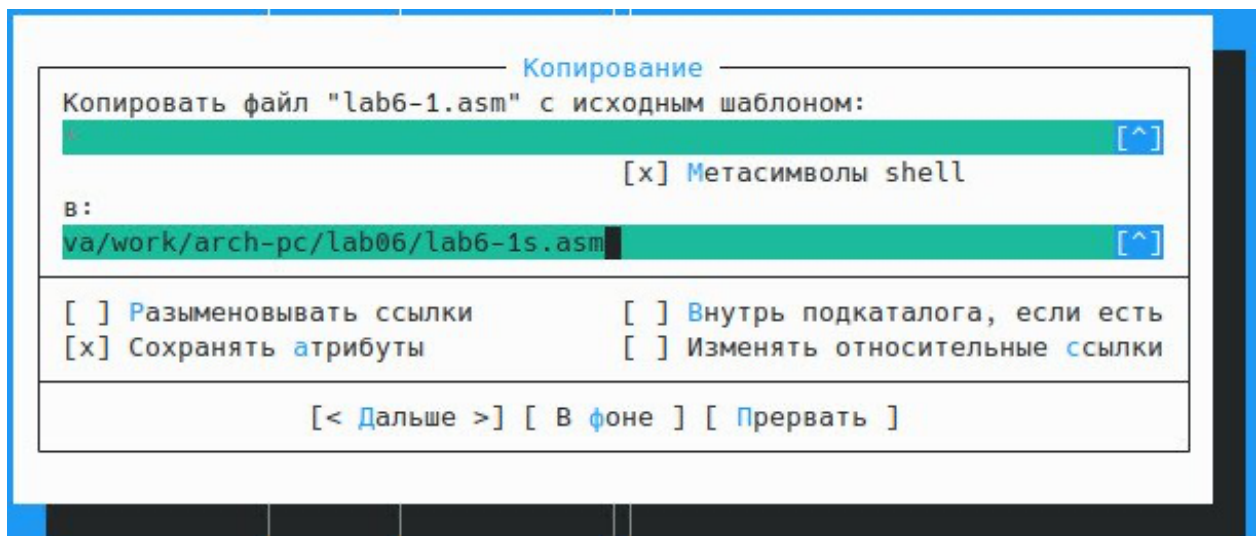


Рис. 17. Создание копии


```

lab6-1s.asm [----] 0 L:[ 1+ 0 1/ 35] *(0 /2431b) 0059 0x03B
; 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 ; Описание 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
;----- 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. 18. Изменения

2) Получаем исполняемый файл и проверяем его работу. (Рис. 19)

```

dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2s.asm
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2s lab6-2s.o
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ./lab6-2s
Введите строку:
Чистякова

```

Рис. 19. Запуск файла

3) Создаем копию файла lab6-2.asm. (Рис. 20)

Вносим изменения в программу. (Рис. 21)

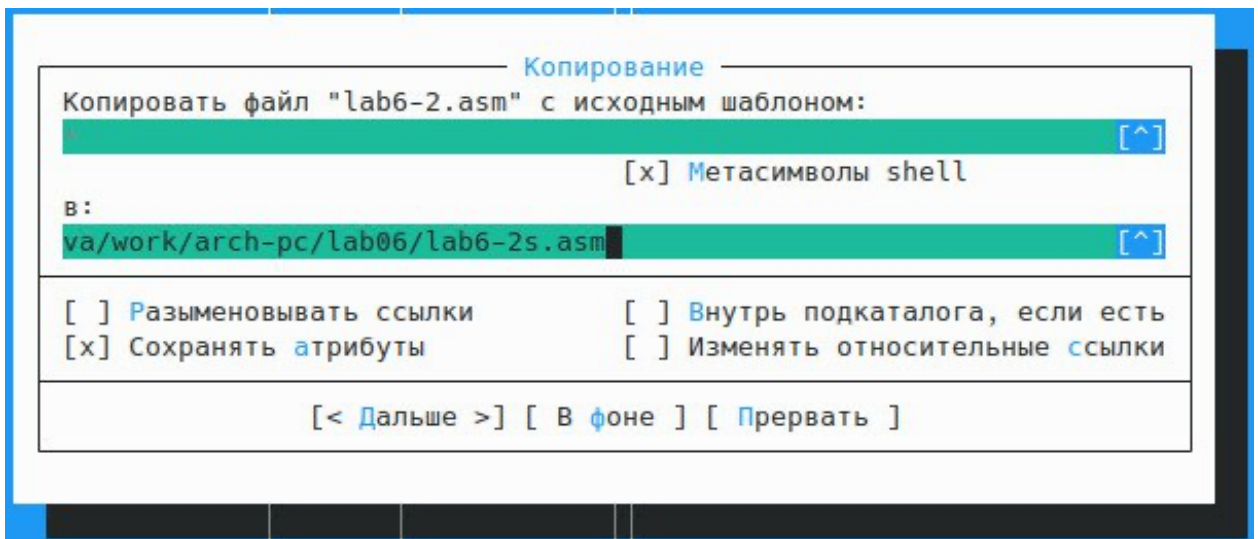


Рис. 20. Создание копии

```
lab6-2s.asm [----] 41 L:[ 1+16 17/ 17] *(1224/1224b) <EOF>
;-----
; 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 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 21. Изменения

4) Получаем исполняемый файл и проверяем его работу. (Рис. 22)

```
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1s.asm
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1s lab6-1s.o
dvchistyakova@dk2n24 ~/work/arch-pc/lab06 $ ./lab6-1s
Введите строку:
Чистякова
```

Рис. 22. Запуск файла

Вывод:

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