

Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Мишина Анастасия Алексеевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий самостоятельной работы	14
4	Выводы	20

Список иллюстраций

2.1	Переход в каталог ~/work/arch-pc	6
2.2	Создание каталога lab06	7
2.3	Создание файла lab6-1.asm	7
2.4	Программа в lab6-1.asm	8
2.5	Работа с файлом lab6-1.asm	10
2.6	Копирование файла in_out.asm в папку lab06	10
2.7	Создание копии файла lab6-1.asm	11
2.8	Программа с кодом 6.2	11
2.9	Работа с файлом lab6-2.asm	12
2.10	Замена подпрограммы spinrLF на sprint	13
2.11	Работа с измененным файлом lab6-2.asm	13
3.1	Создание копии файла lab6-1.asm	15
3.2	Добавление вывода	15
3.3	Работа с файлом lab6-1-my.asm	16
3.4	Создание копии файла lab6-2.asm	18
3.5	Добавление вывода	18
3.6	Работа с файлом lab6-2-my.asm	19

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Для начала открываем Midnight Commander с помощью команды `mc`, переходим в каталог `~/work/arch-pc` (рис. 2.1). Создаем каталог для шестой работы, используя команду `F7` (рис. 2.2).

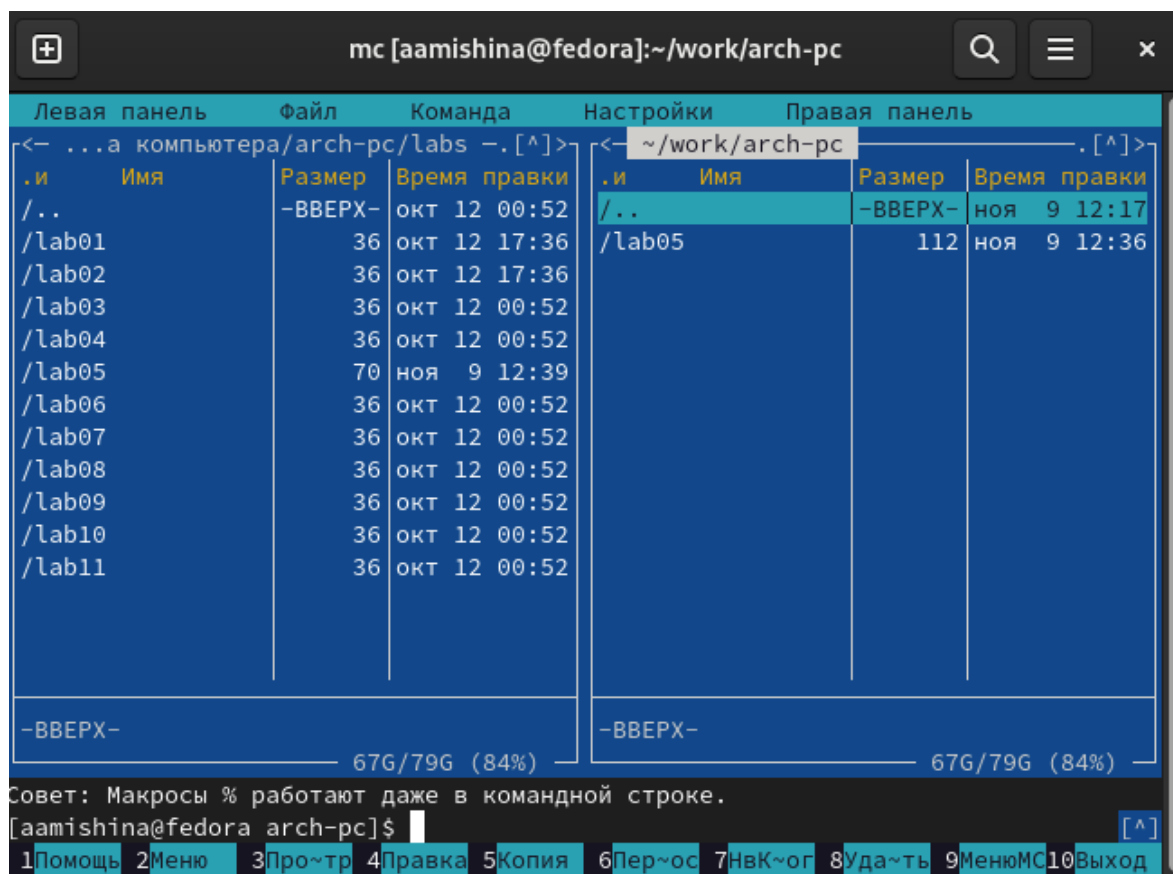


Рис. 2.1: Переход в каталог `~/work/arch-pc`

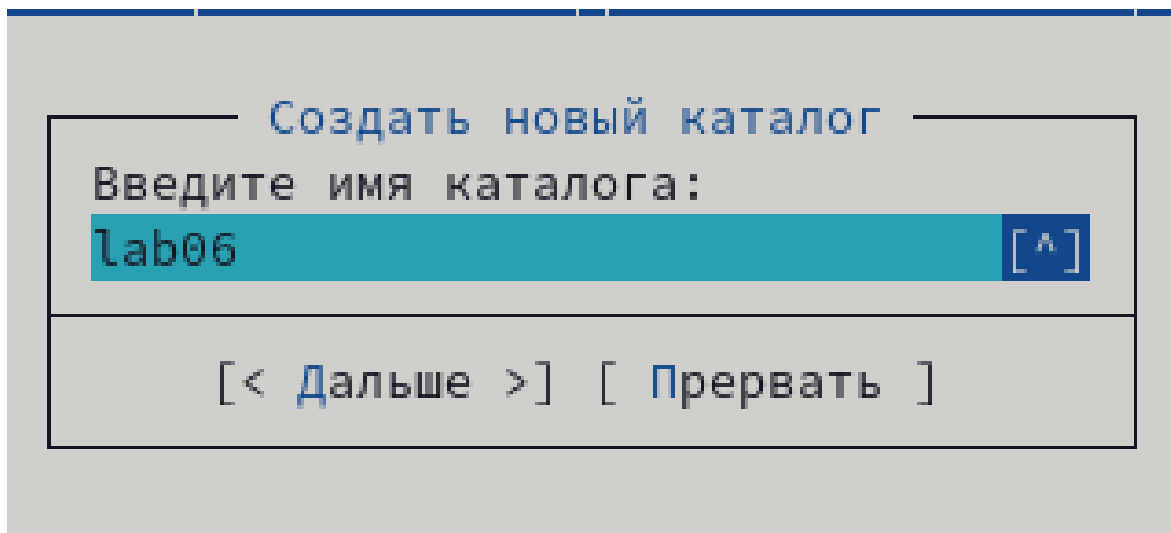


Рис. 2.2: Создание каталога lab06

В новом каталоге создаем файл lab6-1.asm (рис. 2.3) и открываем его с помощью клавиши F4. Теперь введем текст 6.1 из мануала в файл (рис. 2.4).

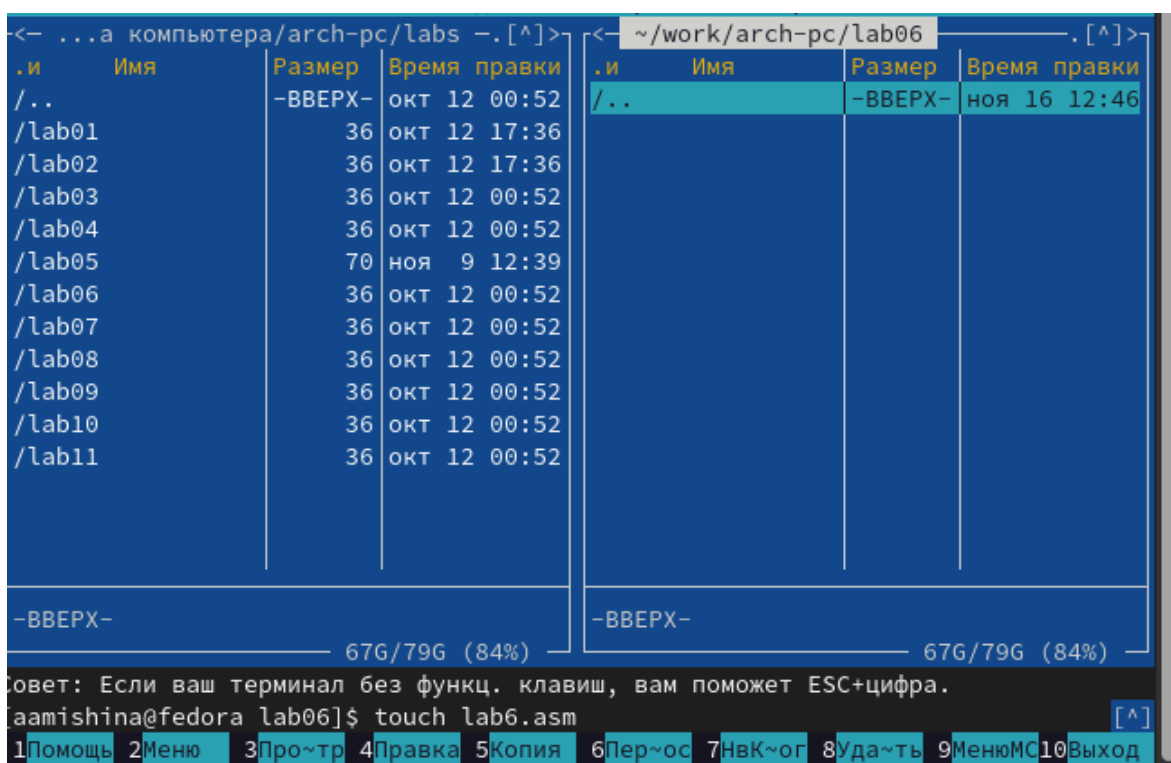


Рис. 2.3: Создание файла lab6-1.asm

```
mc [aamishina@fedora]:~/work/arch-pc/lab06
/home/aamishina/work/arch-pc/lab06/lab6-1.asm 1448/2431 59%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
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
;----- 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
1Помощь 2Раз-рн 3Выход 4Нех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход
```

Рис. 2.4: Программа в lab6-1.asm

Программа lab6-1.asm:

```
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
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
;----- 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
;----- 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

```

Пользуясь клавишами F2, F10, F3 сохраняем программу, выходим из редактора и открываем файл для просмотра, убеждаемся, что все на месте.

Далее преобразовываем текст программы в объектный файл, выполняем компоновку и запускаем исполняемый файл. Вводим ФИО, программа отработывает (рис. 2.5).

```
[aamishina@fedora lab06]$ nasm -f elf lab6-1.asm
[aamishina@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[aamishina@fedora lab06]$ ./lab6-1
Введите строку:
Мишина Анастасия Алексеевна
[aamishina@fedora lab06]$
```

Рис. 2.5: Работа с файлом lab6-1.asm

Скачиваем файл in_out.asm из ТУИСа, копируем его из каталога загрузок в каталог lab06 (рис. 2.6).

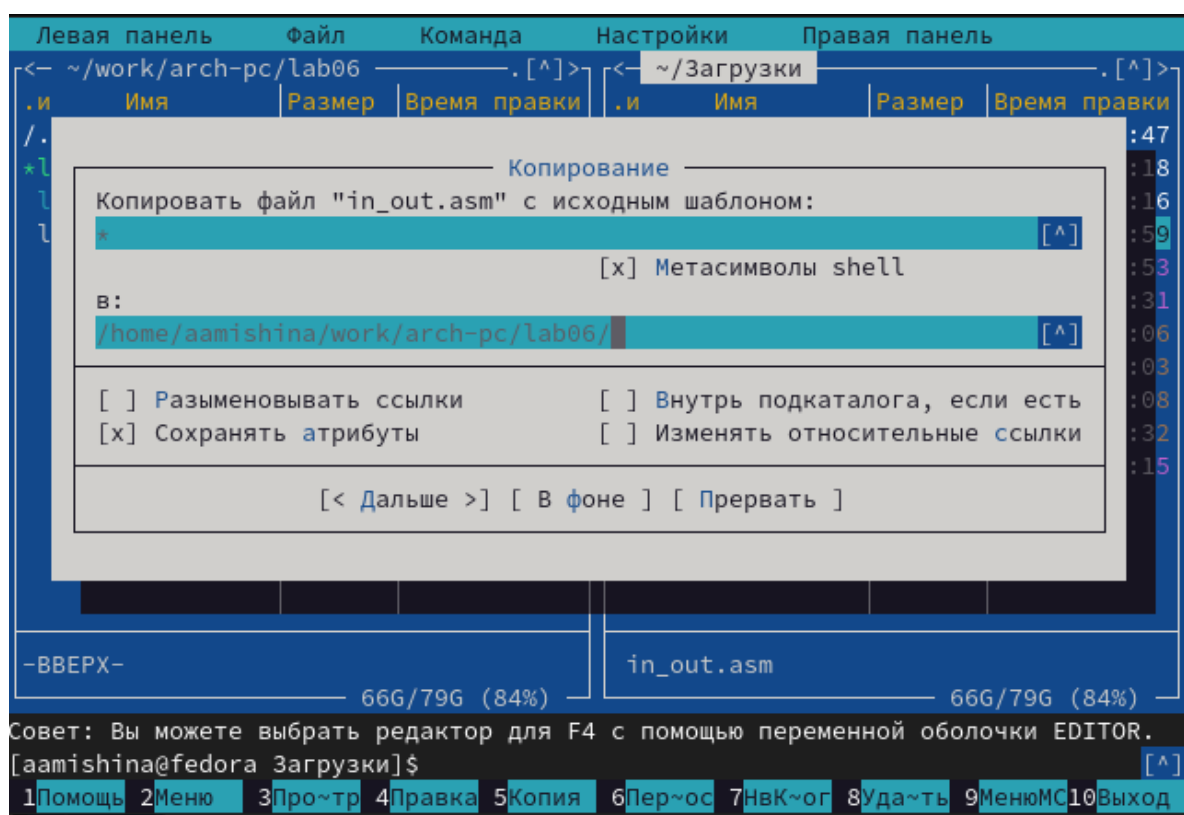


Рис. 2.6: Копирование файла in_out.asm в папку lab06

Создаем копию файла lab6-1.asm с помощью клавиши F5, называем его lab6-2.asm (рис. 2.7). Вставляем код 6.2 в новый файл (рис. 2.8).

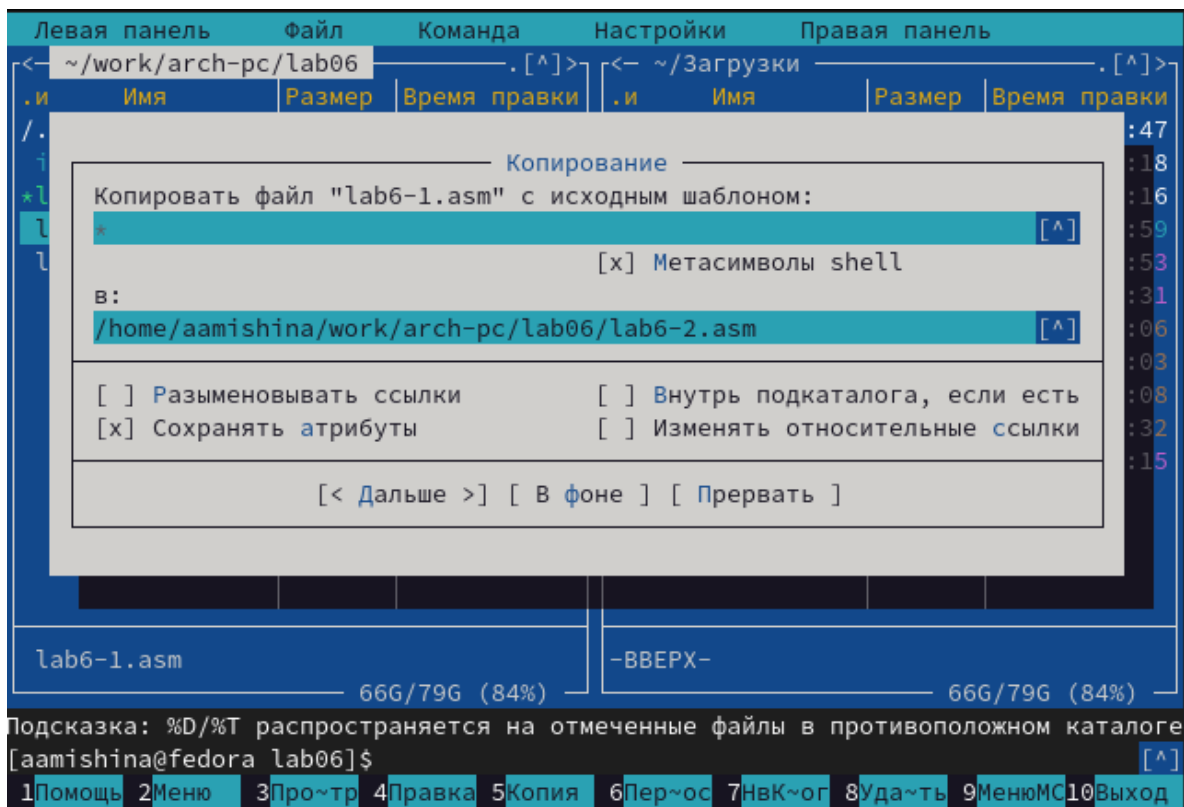


Рис. 2.7: Создание копии файла lab6-1.asm

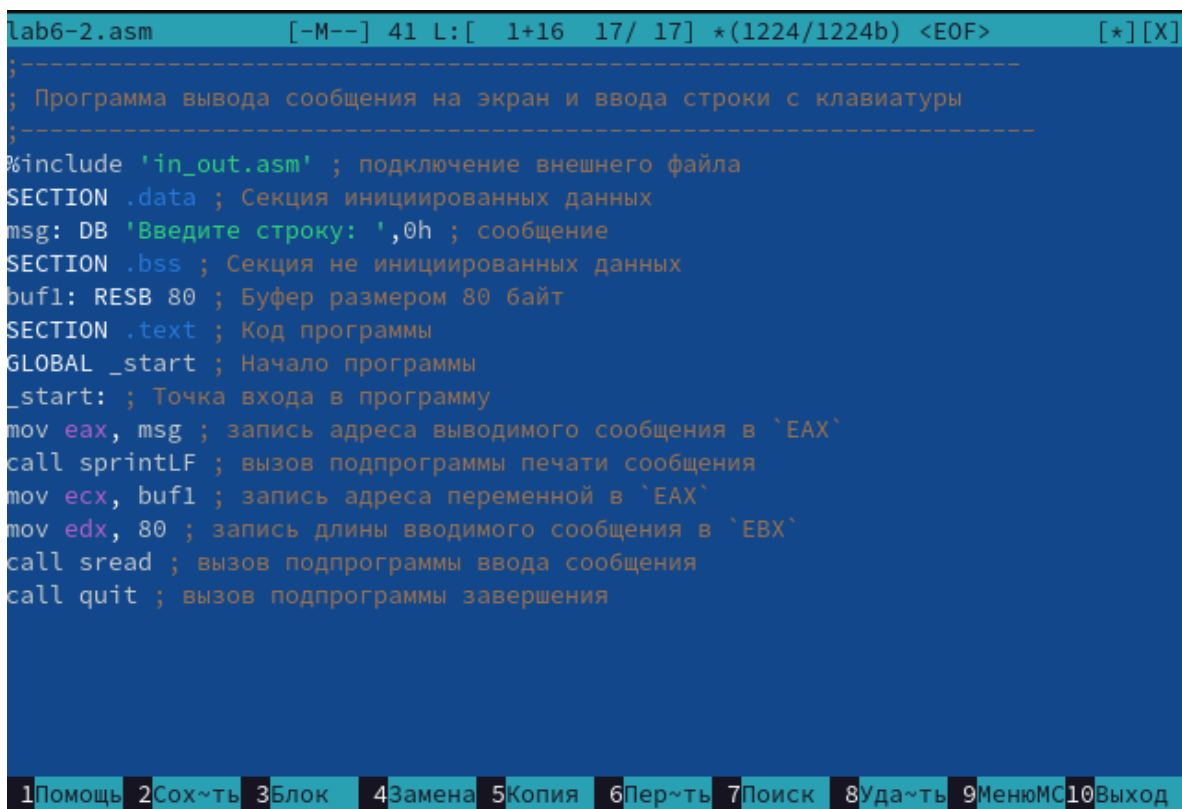
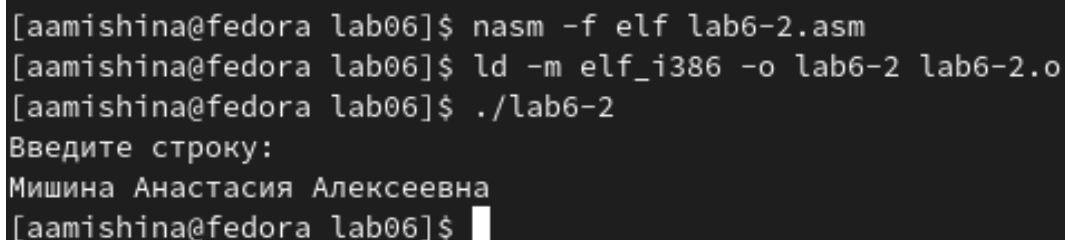


Рис. 2.8: Программа с кодом 6.2

Программа lab6-2.asm:

```
;-----  
; Программа вывода сообщения на экран и ввода строки с клавиатуры  
;-----  
  
%include 'in_out.asm' ; подключение внешнего файла  
  
SECTION .data ; Секция инициированных данных  
msg: DB 'Введите строку: ',0h ; сообщение  
  
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  
  
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 ; вызов подпрограммы ввода сообщения  
call quit ; вызов подпрограммы завершения
```

Создаем исполняемый файл, убеждаемся в его работоспособности (рис. 2.9).



```
[aamishina@fedora lab06]$ nasm -f elf lab6-2.asm  
[aamishina@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o  
[aamishina@fedora lab06]$ ./lab6-2  
Введите строку:  
Мишина Анастасия Алексеевна  
[aamishina@fedora lab06]$
```

Рис. 2.9: Работа с файлом lab6-2.asm

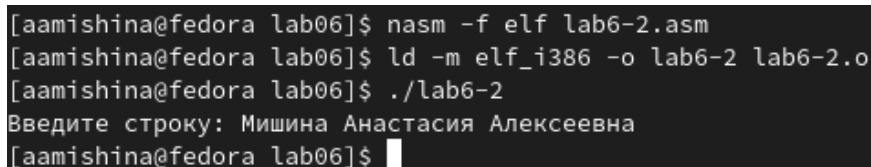
Заменяем подпрограмму spinrLF на sprint (рис. 2.10). Пересоздаем испол-

няемый файл, заметим следующую разницу: теперь нет символа перевода строки при выводе надписи “Введите строку”, следовательно ввод текста будет происходить без перехода на новую строку (рис. 2.11).



```
call sprint; вызов подпрограммы печати сообщения
```

Рис. 2.10: Замена подпрограммы spinrLF на sprint



```
[aamishina@fedora lab06]$ nasm -f elf lab6-2.asm
[aamishina@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[aamishina@fedora lab06]$ ./lab6-2
Введите строку: Мишина Анастасия Алексеевна
[aamishina@fedora lab06]$
```

Рис. 2.11: Работа с измененным файлом lab6-2.asm

3 Выполнение заданий

самостоятельной работы

Для начала создадим копию файла lab6-1.asm (рис. 3.1). Как мы знаем, эта программа уже умеет выводить приглашение типа “Введите строку”, также она считывает нашу строку и записывает ее в buf1. Ориентируясь на вывод сообщения, создаем кусочек кода, который выводит строку, записанную в buf1 (рис. 3.2). Убеждаемся в работоспособности программы (рис. 3.3).

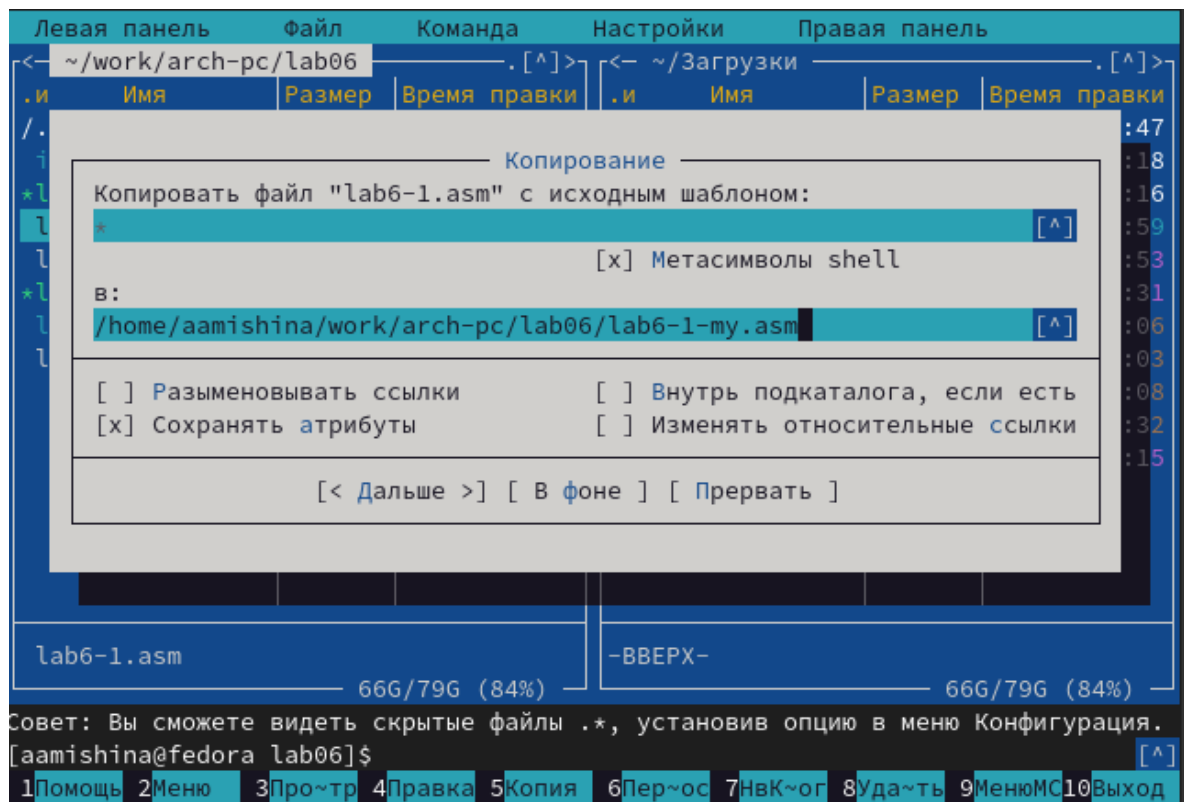


Рис. 3.1: Создание копии файла lab6-1.asm

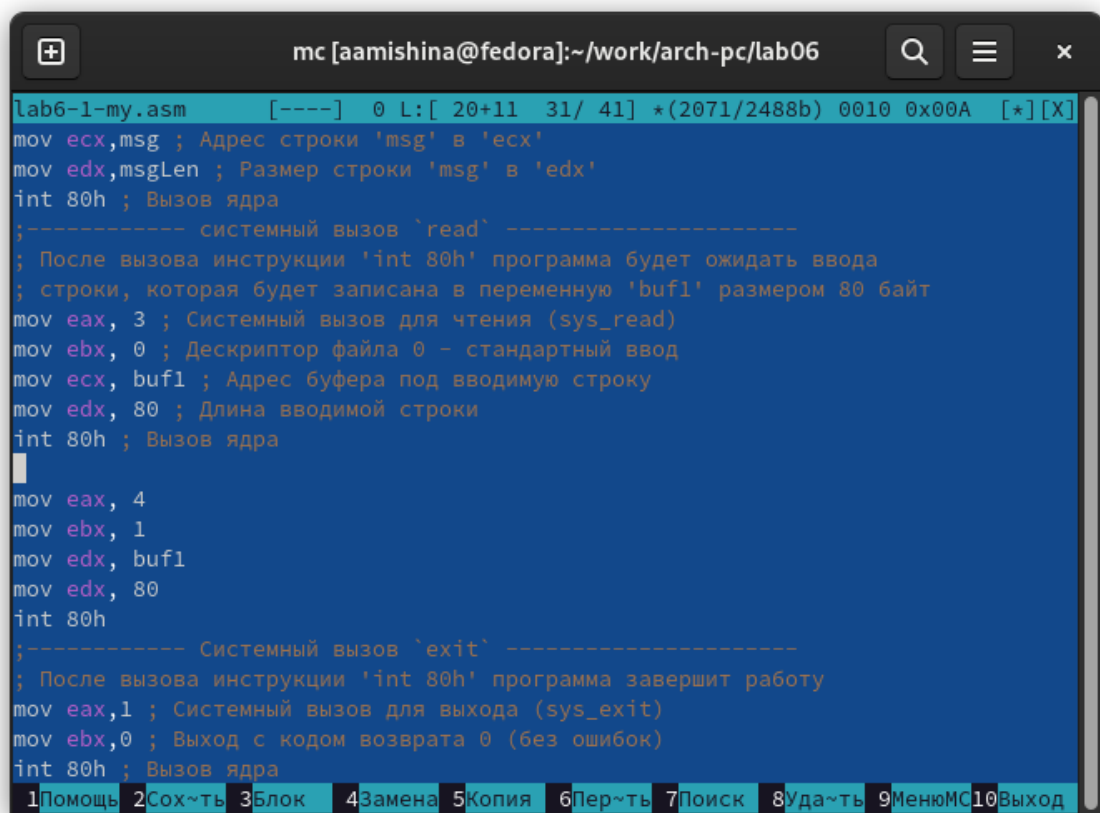


Рис. 3.2: Добавление вывода

```

[aamishina@fedora lab06]$ nasm -f elf lab6-1-my.asm
[aamishina@fedora lab06]$ mc

[aamishina@fedora lab06]$ ld -m elf_i386 -o lab6-1-my lab6-1-my.o
[aamishina@fedora lab06]$ ./lab-1-my
bash: ./lab-1-my: Нет такого файла или каталога
[aamishina@fedora lab06]$ ./lab6-1-my
Введите строку:
Мишина Анастасия Алексеевна
Мишина Анастасия Алексеевна
[aamishina@fedora lab06]$

```

Рис. 3.3: Работа с файлом lab6-1-my.asm

Программа lab6-1-my.asm:

```

;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
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
;----- 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 edx, buf1
mov edx, 80
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

```

Теперь делаем копию файла lab6-2.asm (рис. 3.4), там также добавляем вывод нашей строки (рис. 3.5). Создаем исполняемый файл, убеждаемся, что все работает (рис. 3.6).

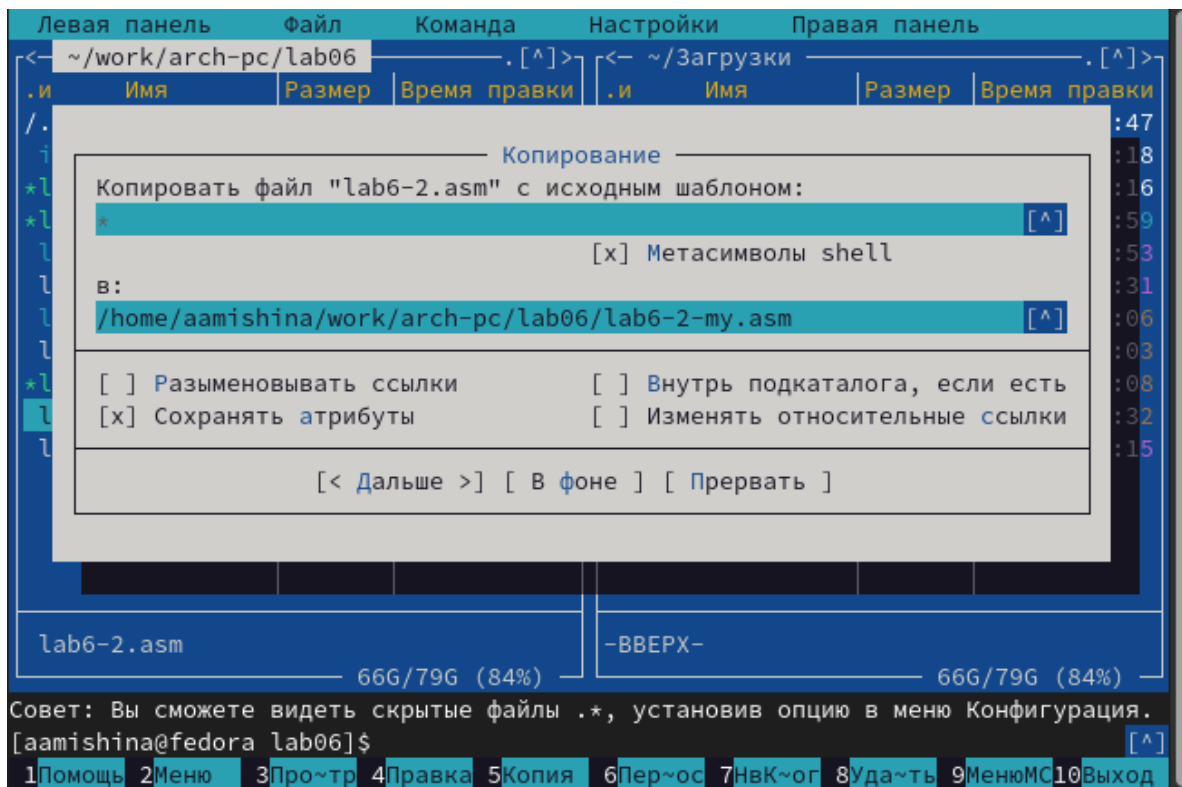


Рис. 3.4: Создание копии файла lab6-2.asm

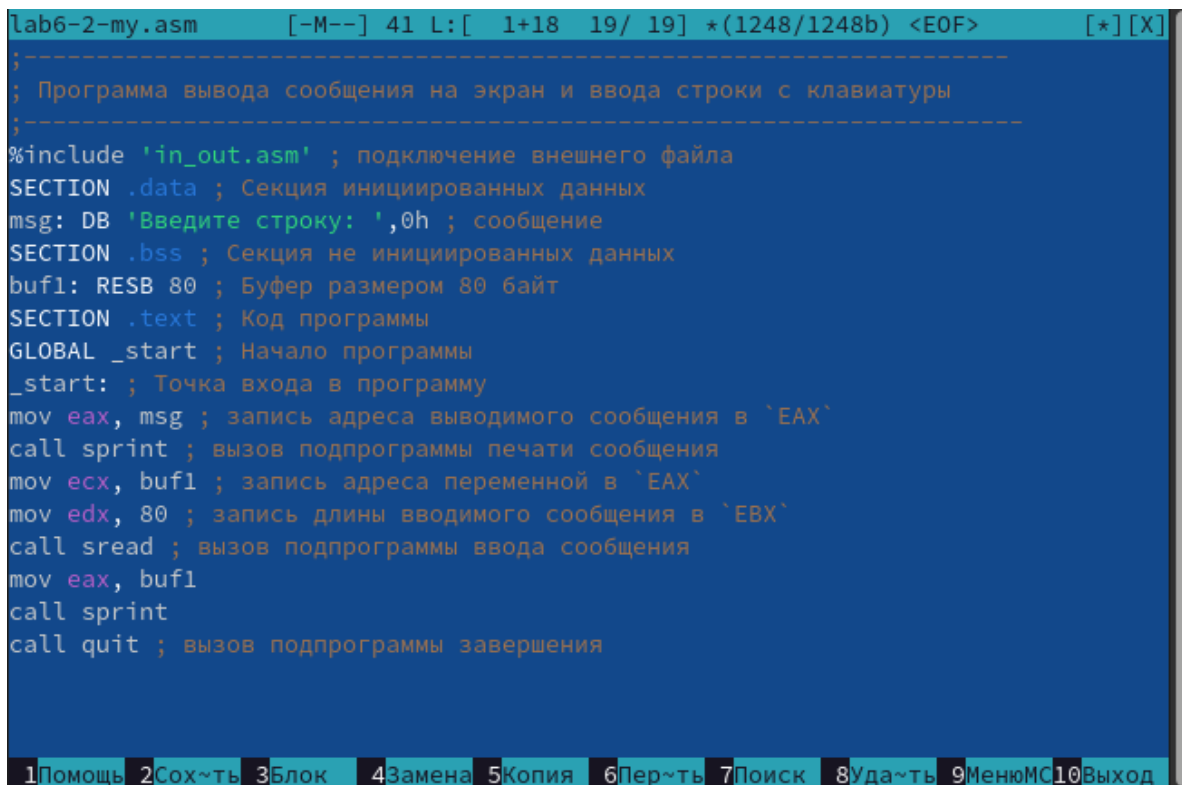


Рис. 3.5: Добавление вывода

```
[aamishina@fedora lab06]$ nasm -f elf lab6-2-my.asm
[aamishina@fedora lab06]$ ld -m elf_i386 -o lab6-2-my lab6-2-my.o
[aamishina@fedora lab06]$ ./lab6-2-my
Введите строку: Мишина Анастасия Алексеевна
Мишина Анастасия Алексеевна
[aamishina@fedora lab06]$
```

Рис. 3.6: Работа с файлом lab6-2-my.asm

Программа lab6-2-my.asm:

```
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

%include 'in_out.asm' ; подключение внешнего файла

SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение

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

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, buf1
call sprint
call quit ; вызов подпрограммы завершения
```

4 Выводы

В ходе выполнения данной лабораторной работы я приобрела практические навыки работы в Midnight Commander. Также я освоила инструкции языка ассемблера mov и int. Вся моя работа была записана и показана в данной лабораторной.