

# **Отчет по Лабораторной работе №5**

**Архитектура компьютеров и операционные системы**

Бекауов Артур Тимурович НКАбд-01-23

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Ход лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Ход самостоятельной работы</b>	<b>16</b>
<b>4</b>	<b>Выводы</b>	<b>19</b>

# Список иллюстраций

2.1	Вызов Midnight Commander . . . . .	6
2.2	Файловый менеджер Midnight Commander . . . . .	7
2.3	Создание каталога с помощью МС . . . . .	8
2.4	Создание текстового файла “lab5-1.asm” . . . . .	9
2.5	Сохранение текстовой программы “lab5-1.asm” в mcedit .	10
2.6	Проверка файла “lab5-1.asm” . . . . .	11
2.7	Создание исполняемого файла, его запуск и результаты. .	11
2.8	Копирование подключаемого файла в папку с “lab5-1.asm”	12
2.9	Создание копии файла “lab5-1.asm” с именем “lab5-2.asm”	13
2.10	Новый текст программы “lab5-2.asm” . . . . .	14
2.11	Создание исполняемого файла “lab5-2”, его запуск и результаты. . . . .	14
2.12	Замена sprintf на sprin в файле “lab5-2.asm” . . . . .	15
2.13	Сравнение старого и нового вывода программы “lab5-2” .	15
3.1	Добавление в текст программы “lab5-1-2.asm” . . . . .	16
3.2	Новый текст программы “lab5-2-2.asm” . . . . .	17
3.3	Вывод исполняемых файлов “lab5-1-2” и “lab5-2-2” . . . .	18

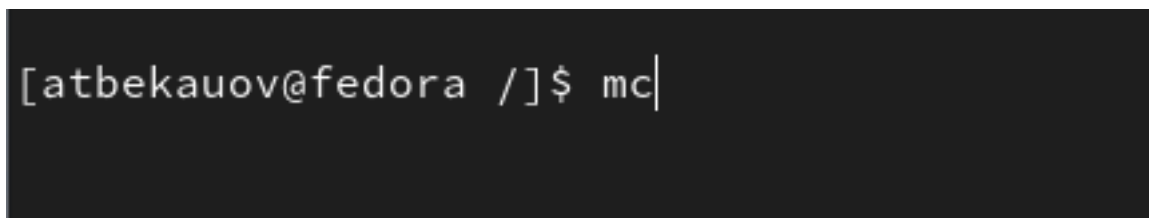
# Список таблиц

# 1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander. Также освоение инструкций языка ассемблера `mov` и `int`.

## 2 Ход лабораторной работы

Открываю терминал и ввожу команду для вызова файлового менеджера Midnight Commander(mc) (Рис. 2.1 и 2.2). Перехожу в каталог `work/arch-rc` созданный при выполнении ЛРН<sup>о</sup>4 и создаю в нём папку “lab05” (Рис. 2.3). С помощью команды `touch` создаю в этой папке текстовый файл “lab5-1.asm” (Рис. 2.4). Пользуясь функциональной клавишей `f4`, открываю “lab5-1.asm” для редактирования в встроенном редакторе `mcedit`. Ввожу текст предложенного в методичке листинга 5.1, сохраняю изменения и закрываю файл. (Рис. 2.5). Оставил комментарии в программе для удобства редактирования.



```
[atbekauov@fedora /]$ mc|
```

Рис. 2.1: Вызов Midnight Commander



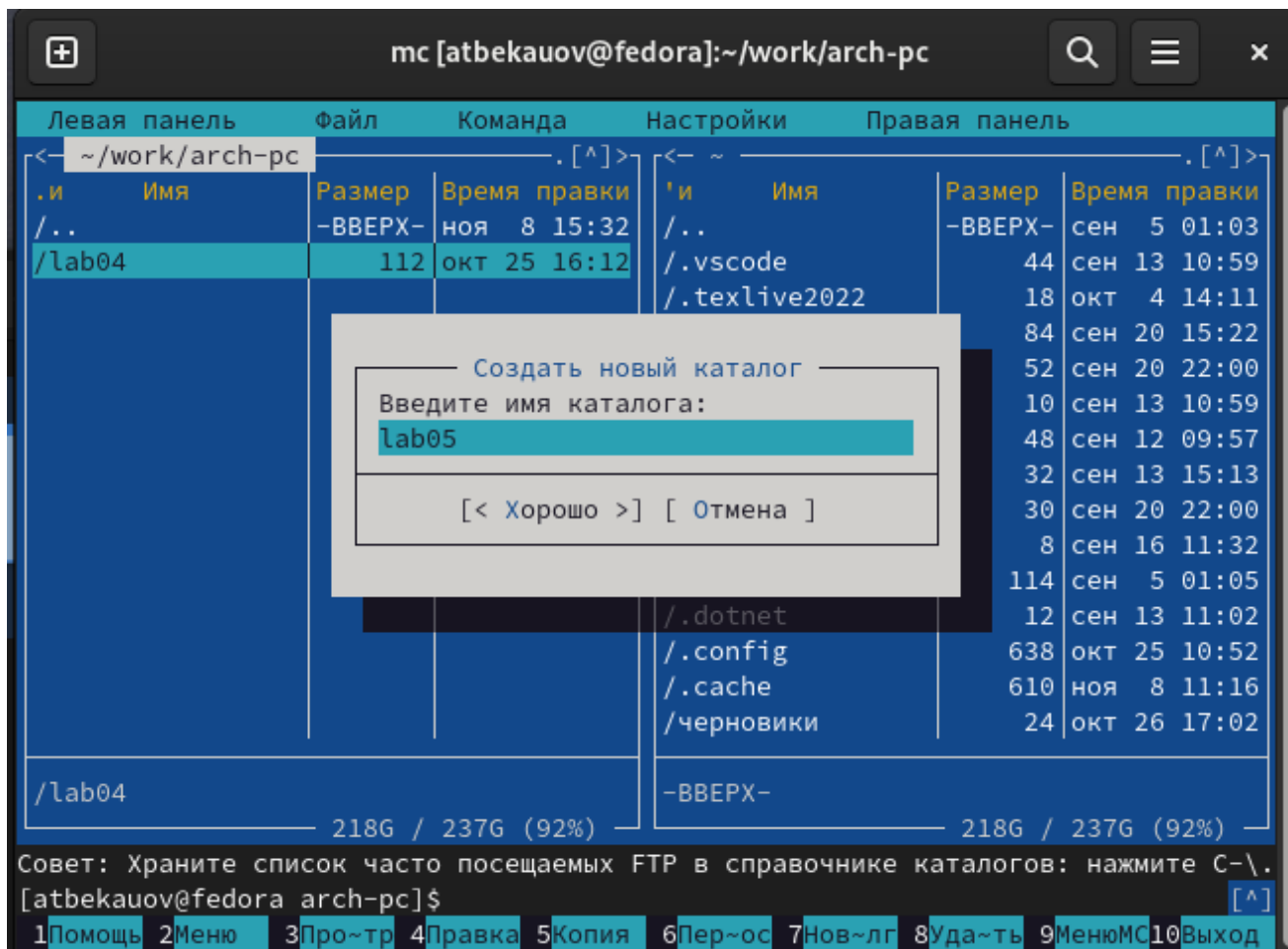


Рис. 2.3: Создание каталога с помощью MC



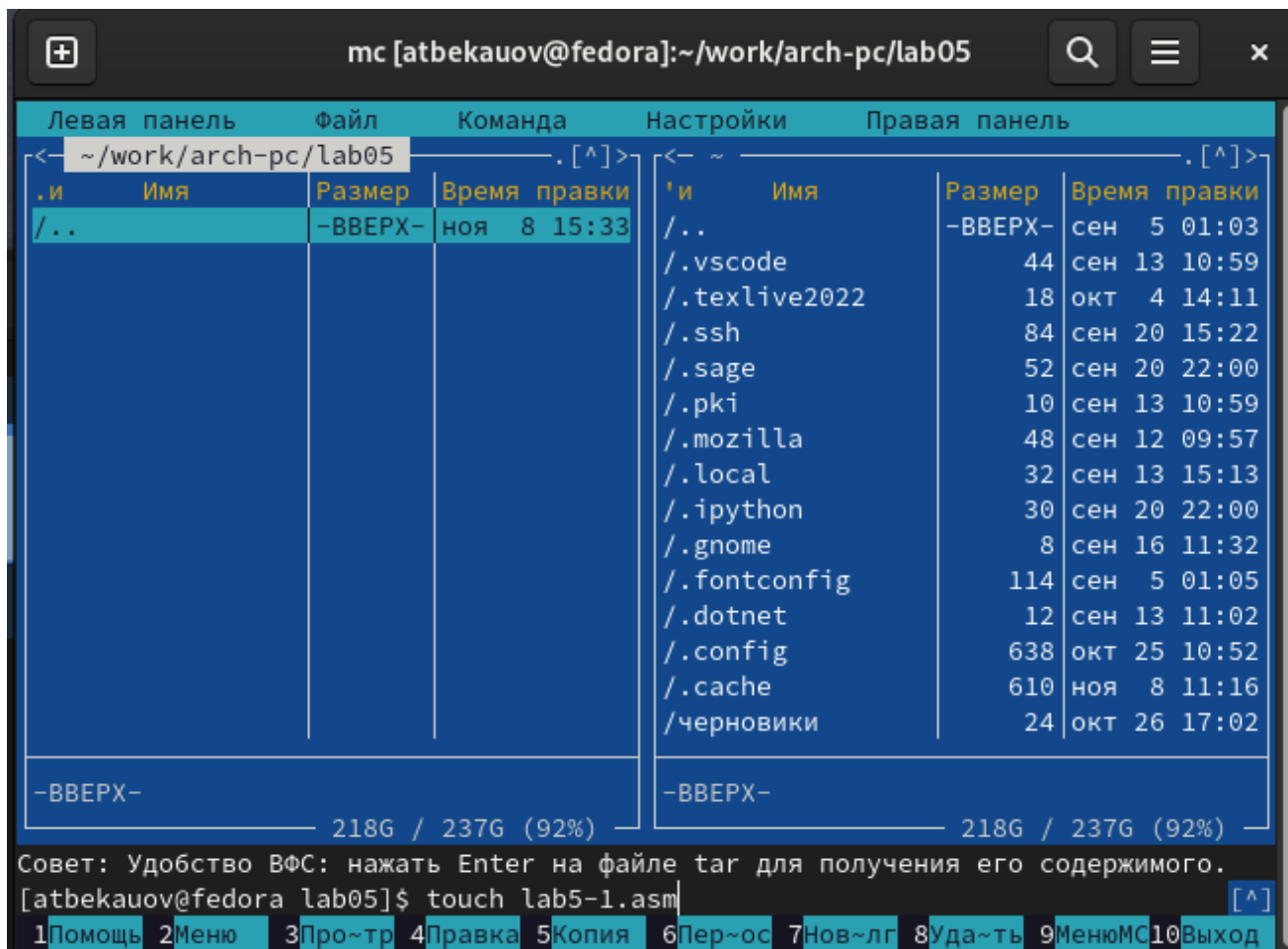


Рис. 2.4: Создание текстового файла “lab5-1.asm”

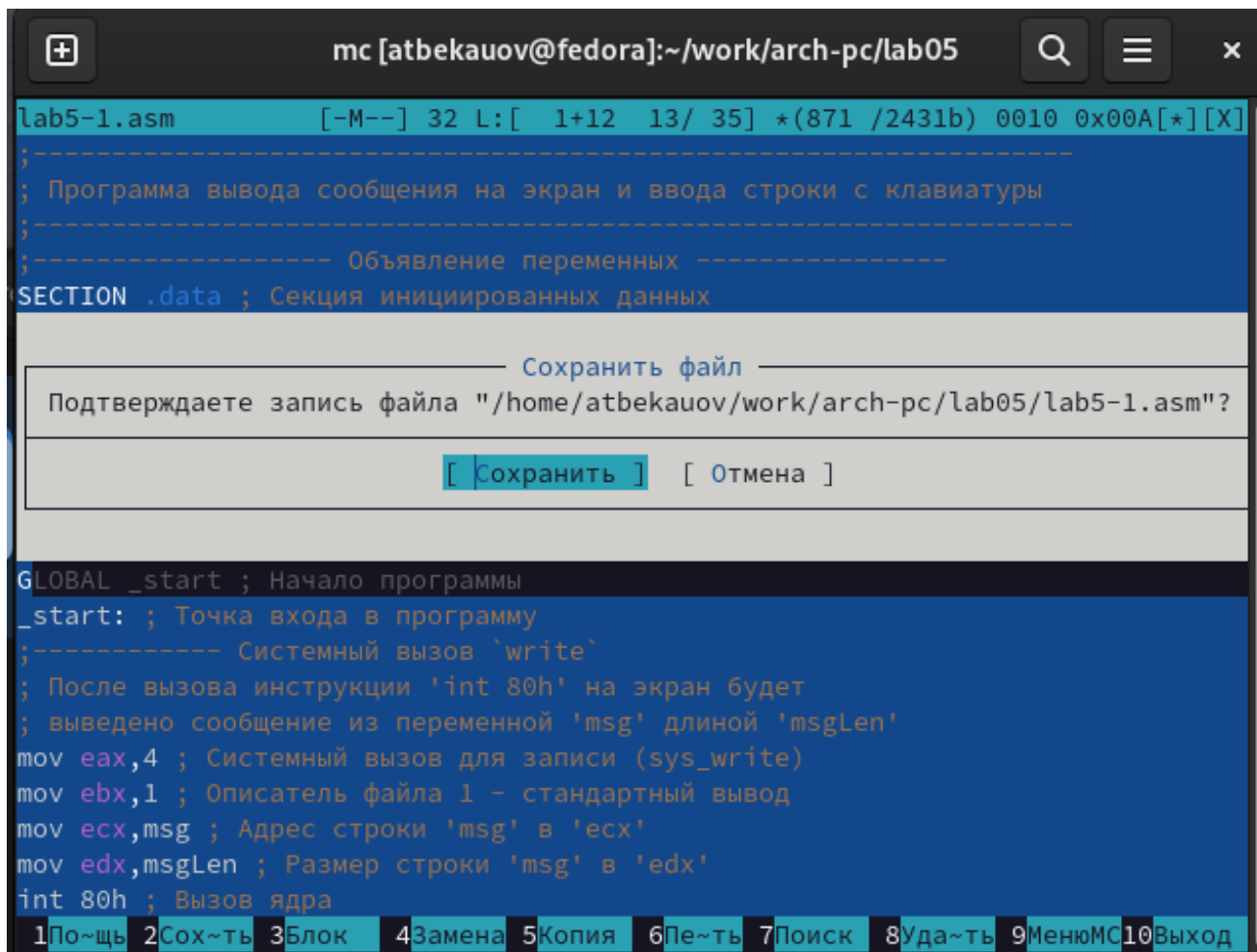
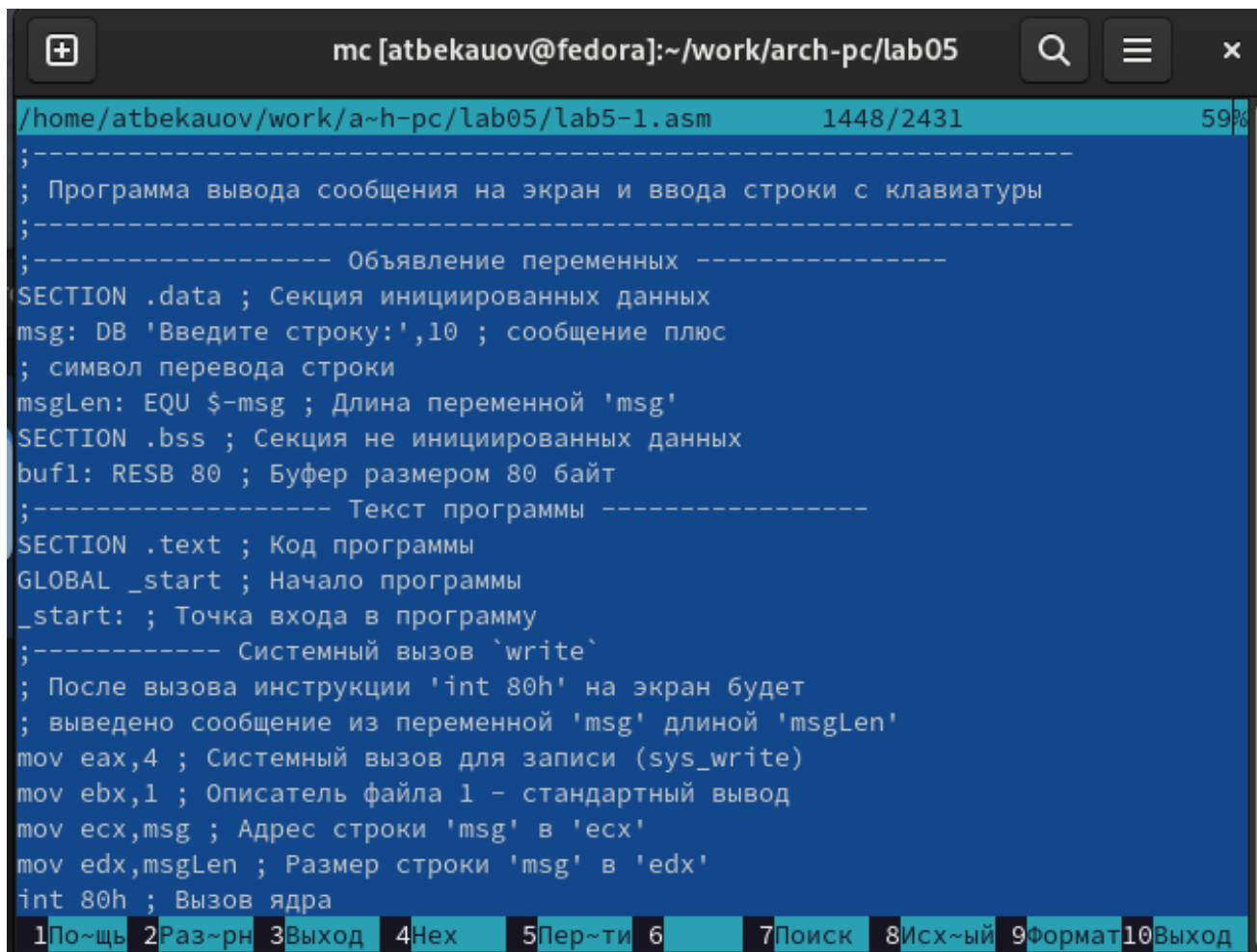


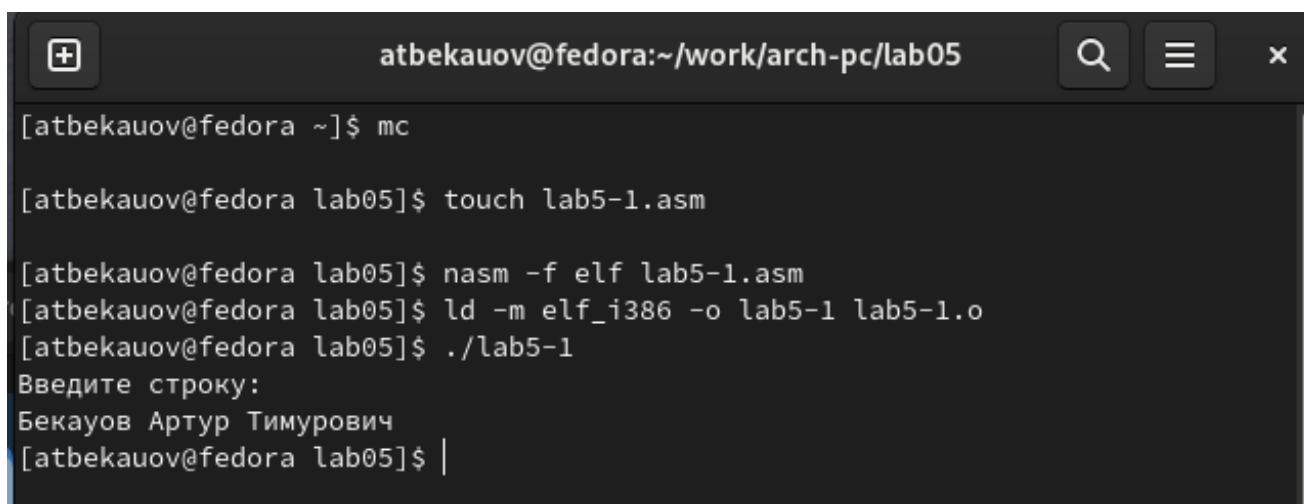
Рис. 2.5: Сохранение текстовой программы “lab5-1.asm” в mcedit

Удостоверюсь, что файл “lab5-1.asm” содержит текст программы, открыв его для просмотра функциональной клавишей f3 (Рис. 2.6). Оттранслирую текст программы в объектный файл (с помощью команды `nasm`), затем выполню компоновку объектного файла (с помощью команды `ld`) и запущу получившийся исполняемый файл. Программа вывела строку “Введите строку:” и оживает ввода с клавиатуры. Ввожу свои ФИО. (Рис. 2.7)



```
mc [atbekauov@fedora]:~/work/arch-pc/lab05
/home/atbekauov/work/arch-pc/lab05/lab5-1.asm 1448/2431 59%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных 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
1Поиск 2Разрешить 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат10Выход
```

Рис. 2.6: Проверка файла “lab5-1.asm”



```
atbekauov@fedora:~/work/arch-pc/lab05
[atbekauov@fedora ~]$ mc
[atbekauov@fedora lab05]$ touch lab5-1.asm
[atbekauov@fedora lab05]$ nasm -f elf lab5-1.asm
[atbekauov@fedora lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[atbekauov@fedora lab05]$ ./lab5-1
Введите строку:
Бекавов Артур Тимурович
[atbekauov@fedora lab05]$ |
```

Рис. 2.7: Создание исполняемого файла, его запуск и результаты.

## Подключение внешнего файла in\_out.asm

Скачаю со страницы курса в ТУИС файл “in\_out.asm”. Подключаемый файл должен лежать в том же каталоге, что и файл с программой, в которой он используется. Поэтому копирую файл “in\_out.asm” из папки загрузки домашнего католога, в папку с “lab5-1.asm” (С помощью МС и функциональной клавиши f5) (Рис. 2.8).

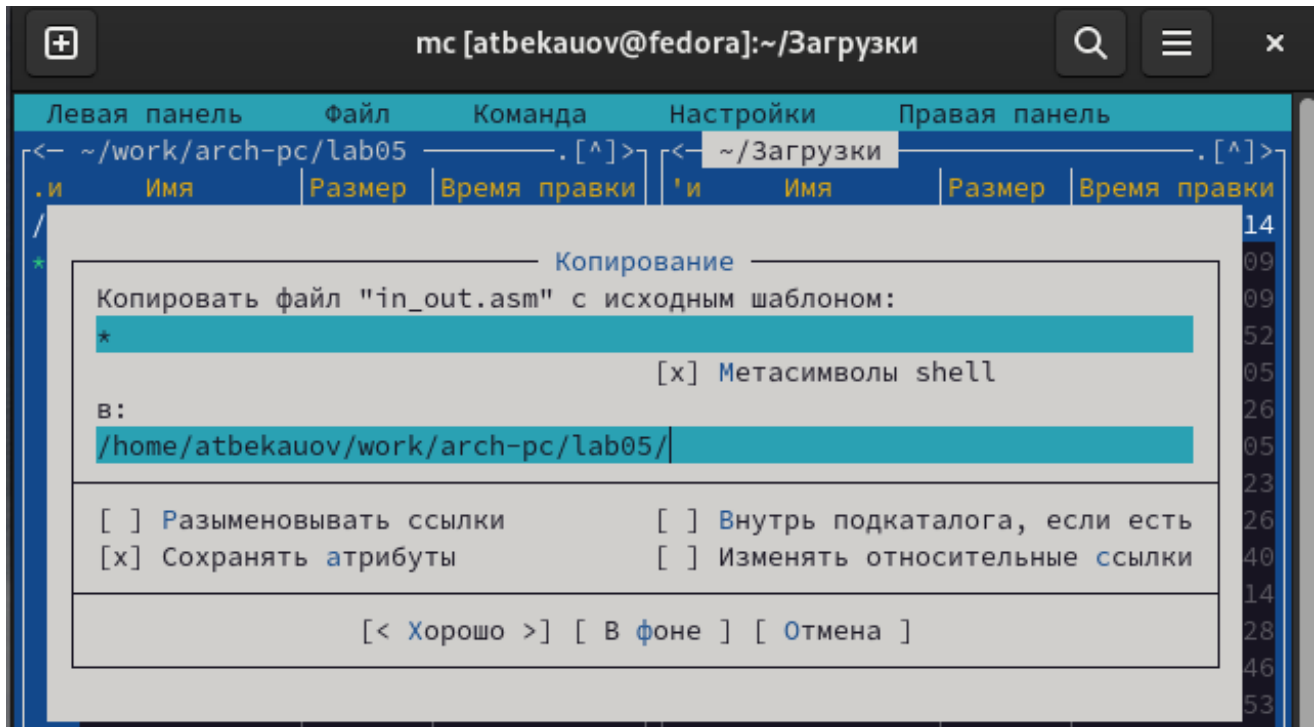


Рис. 2.8: Копирование подключаемого файла в папку с “lab5-1.asm”

Скопирую файл “lab5-1.asm” с новым именем “lab5-2” (Рис. 2.9). Открою созданный в результате копирования файл в редакторе mcedit. Отередктирую содержимое файла в соответствии с листингом 5.2, что позволит мне исползовать подпрограммы из внешнего файла (Рис. 2.10). Создам исполняемый файл и проверю его работу. (Рис. 2.11).

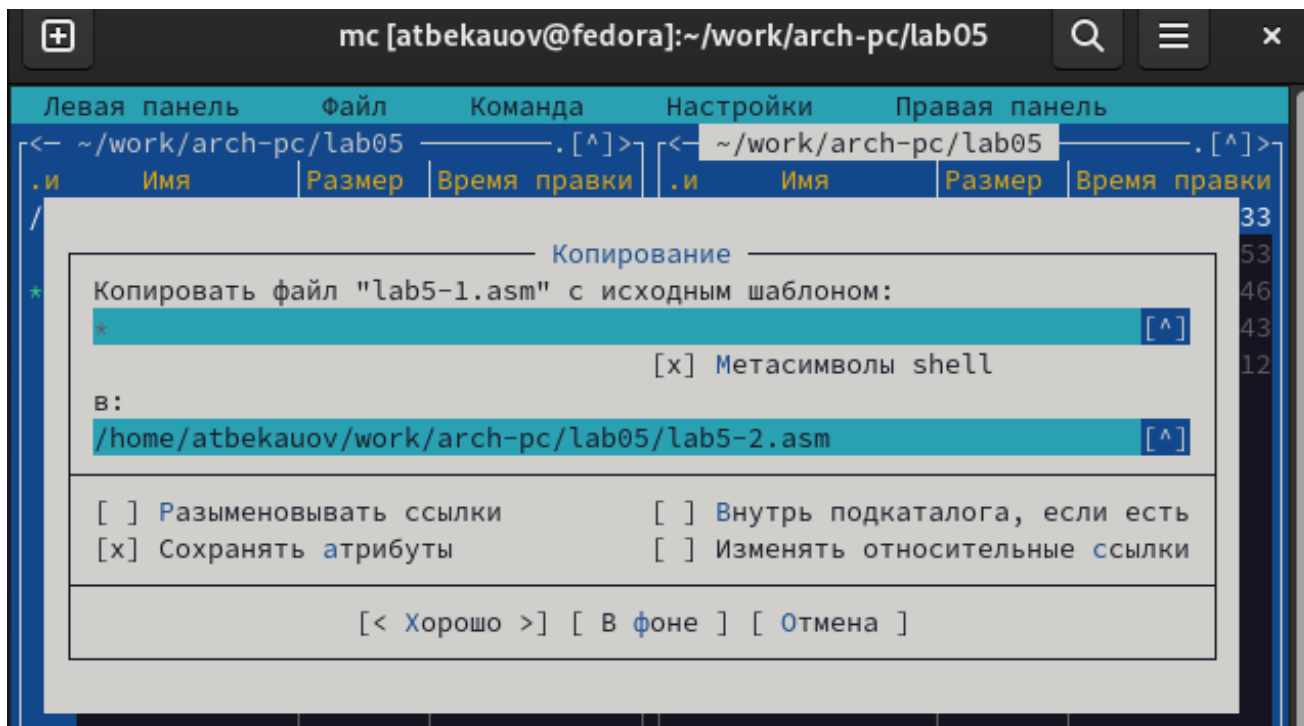
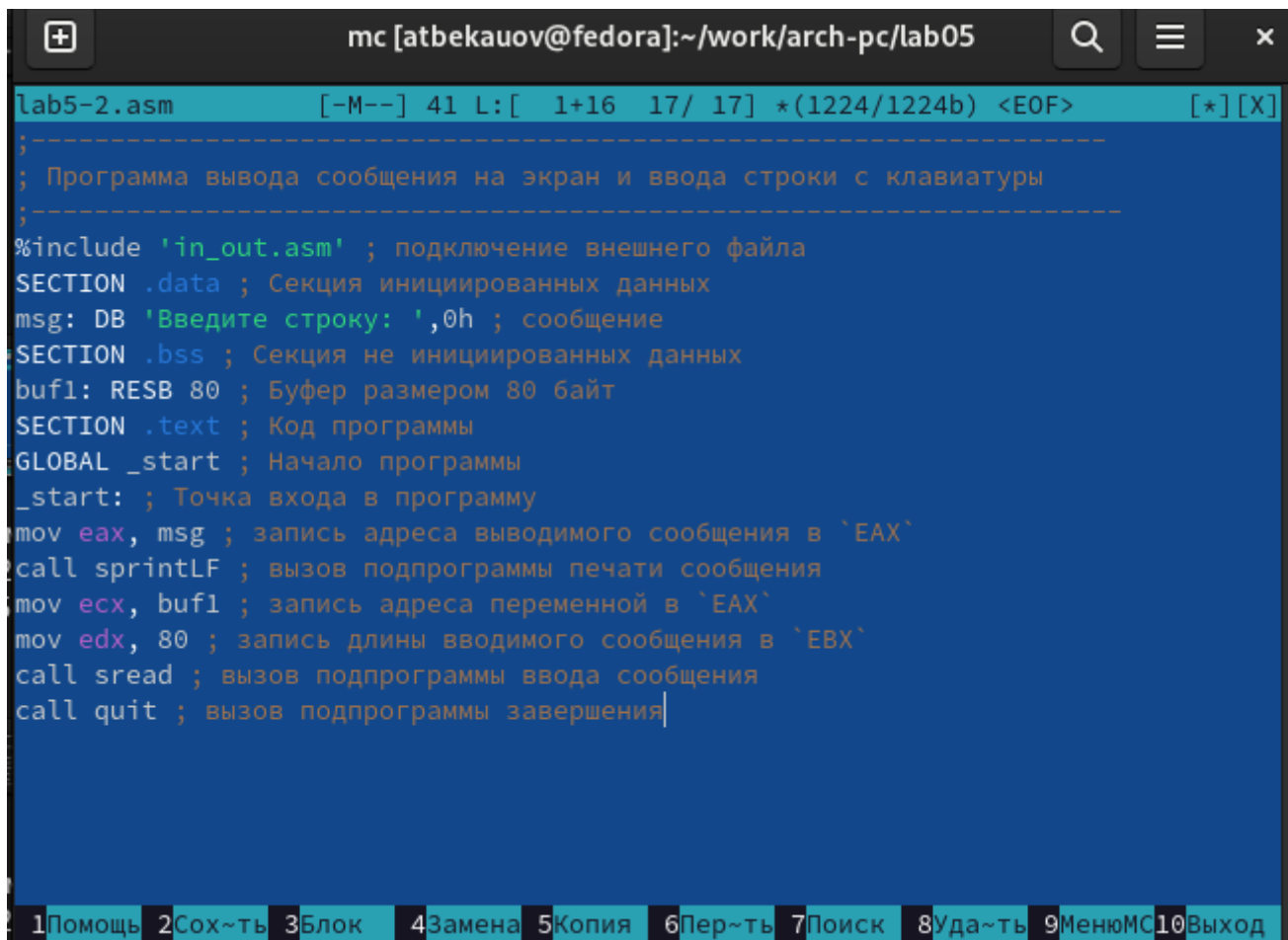


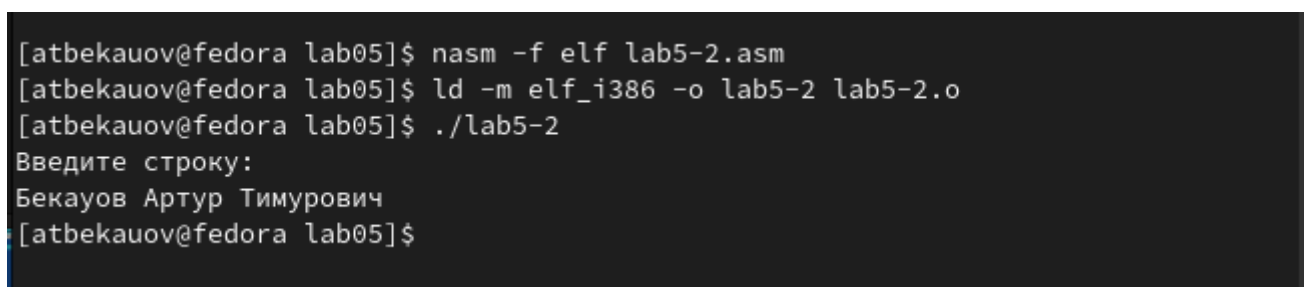
Рис. 2.9: Создание копии файла “lab5-1.asm” с именем “lab5-2.asm”



```
lab5-2.asm      [-M--] 41 L: [ 1+16 17/ 17] *(1224/1224b) <EOF>  [*] [X]
;-----
; 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 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
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС10Выход

Рис. 2.10: Новый текст программы “lab5-2.asm”

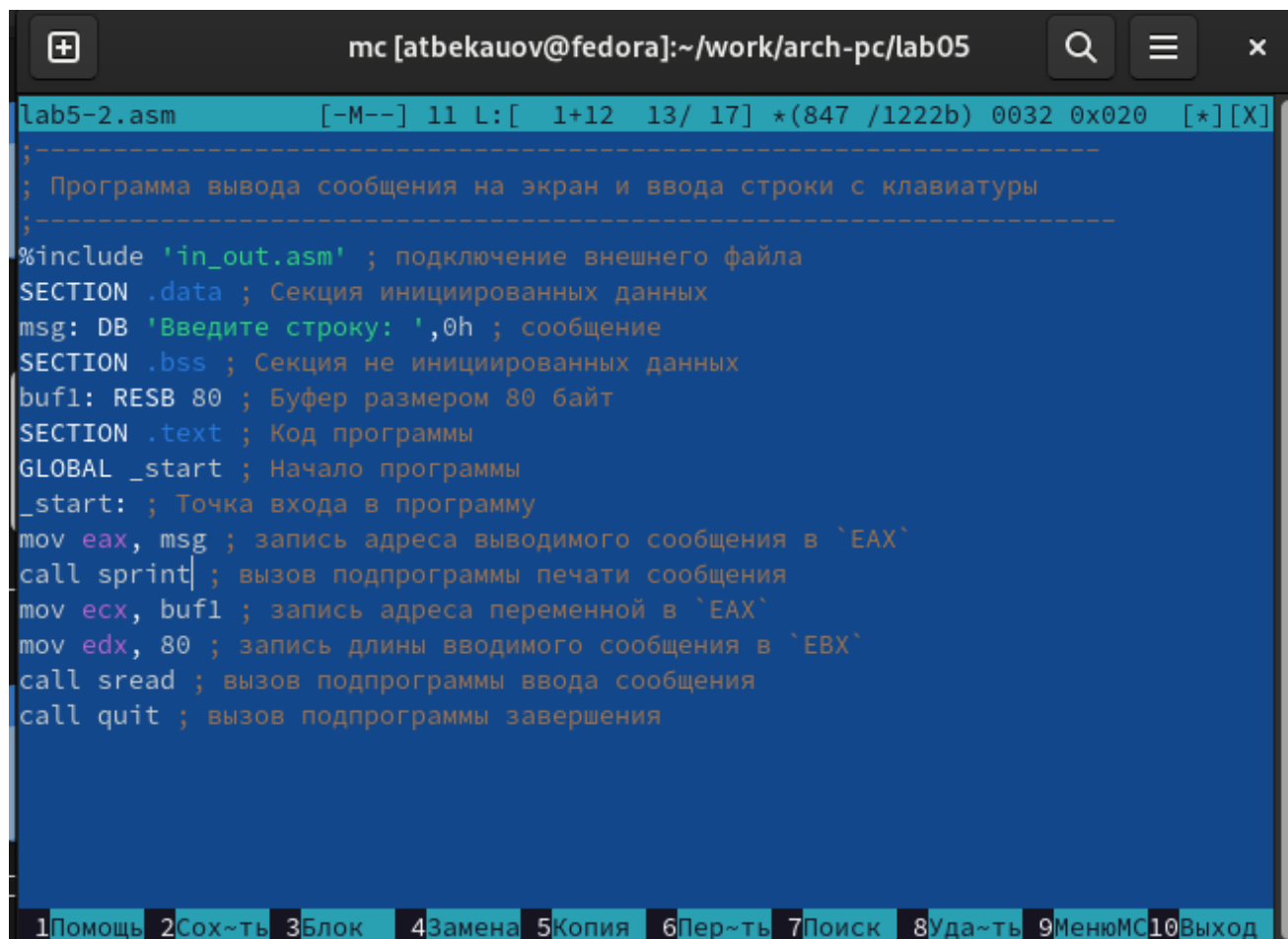


```
[atbekauov@fedora lab05]$ nasm -f elf lab5-2.asm
[atbekauov@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[atbekauov@fedora lab05]$ ./lab5-2
Введите строку:
Бекавов Артур Тимурович
[atbekauov@fedora lab05]$
```

Рис. 2.11: Создание исполняемого файла “lab5-2”, его запуск и результаты.

Заменю в файле “lab5-2.asm” подпрограмму sprintf на printf.(Рис. 2.12) Создам исполняемый файл и проверю его работу (Рис. 2.13). Как видно разница заключается в том, что подпрограмма sprintf сначала совершает переход на новую строку, а затем запрашивает ввод с клавиатуры. Под-

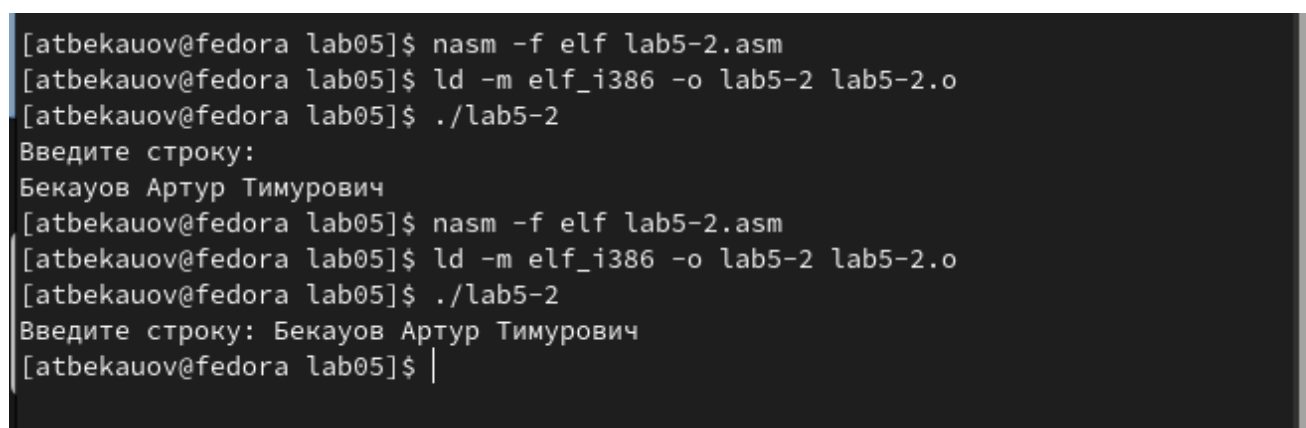
программа `sprint` же совершает запрос на ввод с клавиатуры без перехода на новую строку.



```
lab5-2.asm [-M--] 11 L:[ 1+12 13/ 17] *(847 /1222b) 0032 0x020 [*] [X]
;-----
; 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
call quit ; вызов подпрограммы завершения

1Помощь 2Сох~ть 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС10Выход
```

Рис. 2.12: Замена `sprintf` на `sprin` в файле “lab5-2.asm”



```
[atbekauov@fedora lab05]$ nasm -f elf lab5-2.asm
[atbekauov@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[atbekauov@fedora lab05]$ ./lab5-2
Введите строку:
Бекаов Артур Тимурович
[atbekauov@fedora lab05]$ nasm -f elf lab5-2.asm
[atbekauov@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[atbekauov@fedora lab05]$ ./lab5-2
Введите строку: Бекаов Артур Тимурович
[atbekauov@fedora lab05]$
```

Рис. 2.13: Сравнение старого и нового вывода программы “lab5-2”

### 3 Ход самостоятельной работы

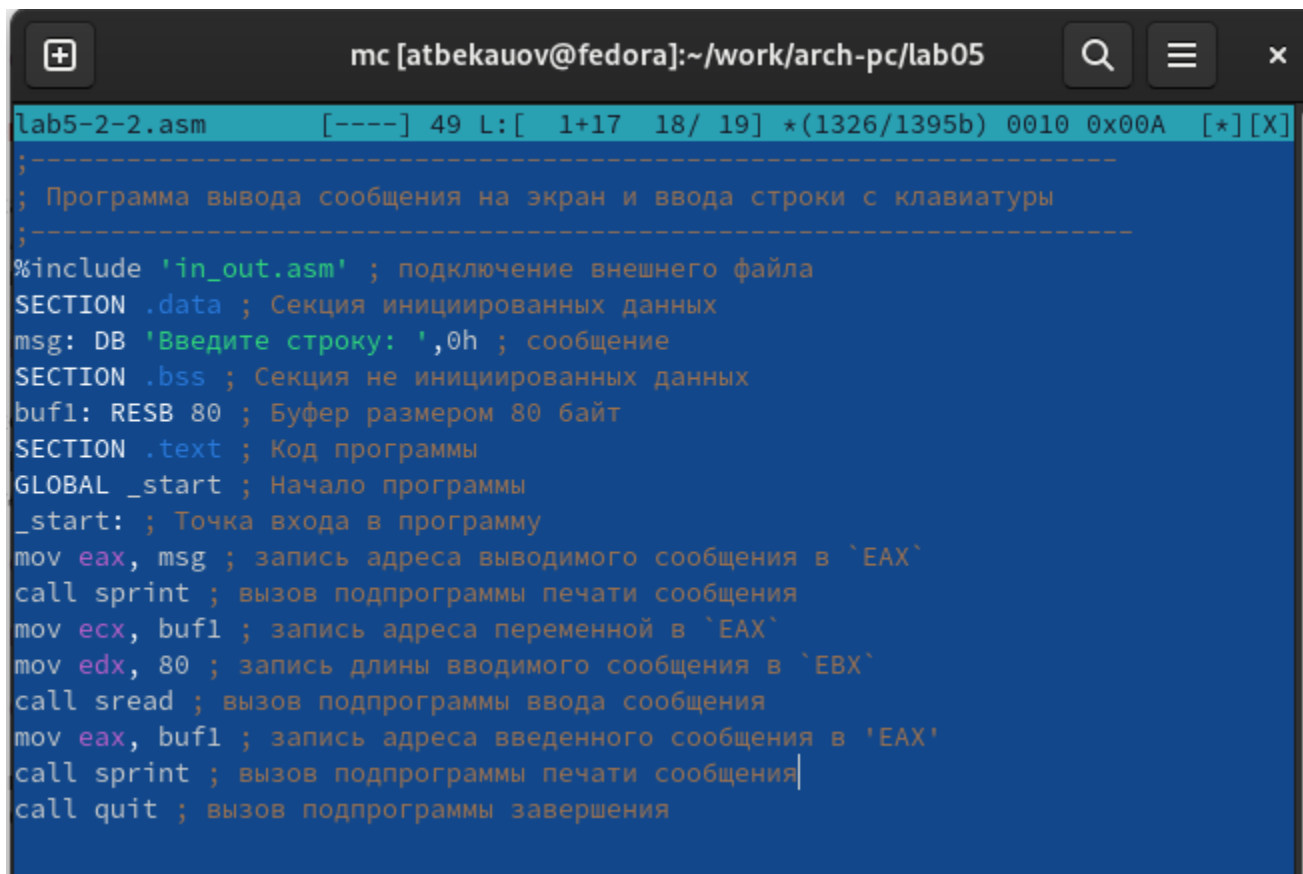
Создам копию файла “lab5-1.asm” с названием “lab5-1-2.asm”. Внесу в программу изменения, так чтобы она выводила внесённую с клавиатуры строку на экран. А именно: после системного вызова ‘read’, перед ‘exit’ добавлю ещё один вызов ‘write’, выглядящий следующим образом (Рис. 3.1). Сохраню изменения и закрою файл.

```
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'buf1' размером 80 байт
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 – стандартный вывод
mov ecx, buf1 ; Адрес строки 'buf1' в 'ecx'
mov edx, 80 ; Размер строки 'buf1' в 'edx'
int 80h ; Вызов ядра
```

Рис. 3.1: Добавление в текст программы “lab5-1-2.asm”

Также создам копию файла “lab5-2.asm” с названием “lab5-2-2.asm”. Внесу в программу изменения, так чтобы она выводила внесённую с клавиатуры строку на экран, но при этом использовала подпрограммы из внешнего файла. А именно: После вызова подпрограммы для ввода сообщения добавлю две строчки - адрес выводимого сообщения в EAX и вызов подпрограммы для вывода сообщения (Рис. 3.2). Сохраню изменения и закрою файл.





```
lab5-2-2.asm [----] 49 L: [ 1+17 18/ 19] *(1326/1395b) 0010 0x00A [*] [X]
;-----
; 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 ; запись адреса введенного сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 3.2: Новый текст программы “lab5-2-2.asm”

Создам исполняемые файлы “lab5-1-2” и “lab5-2-2” и проверю их работу (Рис. 3.3). В качестве ввода с клавиатуры использовал свою фамилию. Как видно программы написаны верно.

```
[atbekauov@fedora lab05]$ nasm -f elf lab5-1-2.asm
[atbekauov@fedora lab05]$ ld -m elf_i386 -o lab5-1-2 lab5-1-2.o
[atbekauov@fedora lab05]$ ./lab5-1-2
Введите строку:
Бекауов
Бекауов
[atbekauov@fedora lab05]$ nasm -f elf lab5-2-2.asm
[atbekauov@fedora lab05]$ ld -m elf_i386 -o lab5-2-2 lab5-2-2.o
[atbekauov@fedora lab05]$ ./lab5-2-2
Введите строку: Бекауов
Бекауов
[atbekauov@fedora lab05]$ |
```

Рис. 3.3: Вывод исполняемых файлов “lab5-1-2” и “lab5-2-2”

## 4 Выводы

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