

Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Астраханцева Анастасия Александровна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Подключение внешнего файла in_out.asm	9
2.2	Задания для самостоятельной работы	12
3	Выводы	18

Список иллюстраций

2.1	Открытие Midnight Commander	6
2.2	Создание папки lab06	6
2.3	Создание lab06-1.asm	7
2.4	Запись текста программы	7
2.5	Комановка и запуск lab06-1.asm	9
2.6	Перенос in_out.asm	10
2.7	Создание lab06-2	10
2.8	Правка текста программы	11
2.9	Запуск lab06-2.asm	12
2.10	Запуск lab06-2.asm	12
2.11	Изменение lad06-1.asm	13
2.12	Изменение lad06-1.asm	14
2.13	Создание lab06-2-1.asm	15
2.14	Изменения в lab06-2-1.asm	15
2.15	Изменения в lab06-2-1.asm	17

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 открою Midnight Commander с помощью команды `mc` (рис. 2.1):

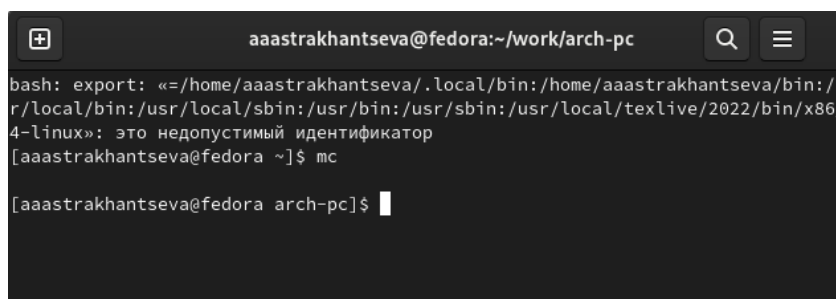


Рис. 2.1: Открытие Midnight Commander

Далее с помощью клавиши F7 создаю папку `lab06` и перехожу в нее (рис. 2.2):

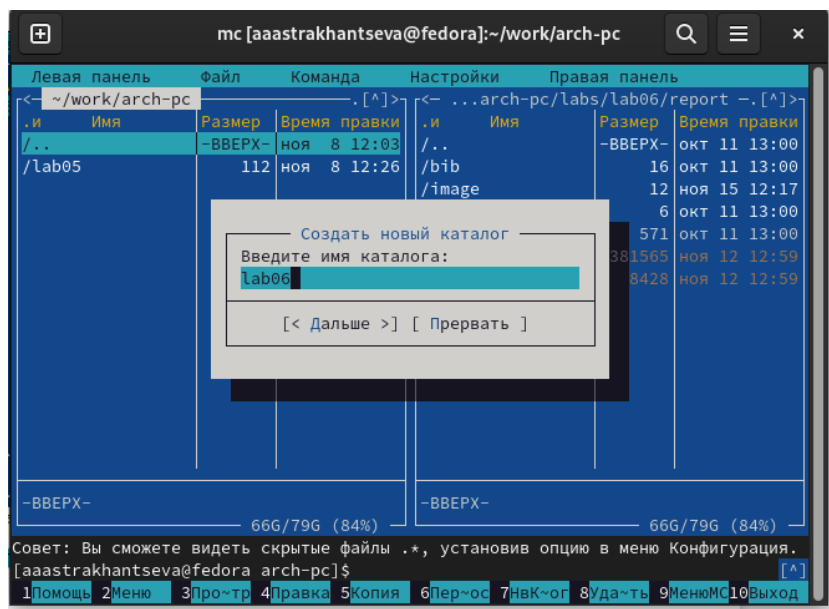


Рис. 2.2: Создание папки `lab06`

После этого, пользуясь строкой ввода и командой touch создаю файл lab06-1.asm (рис. 2.3):

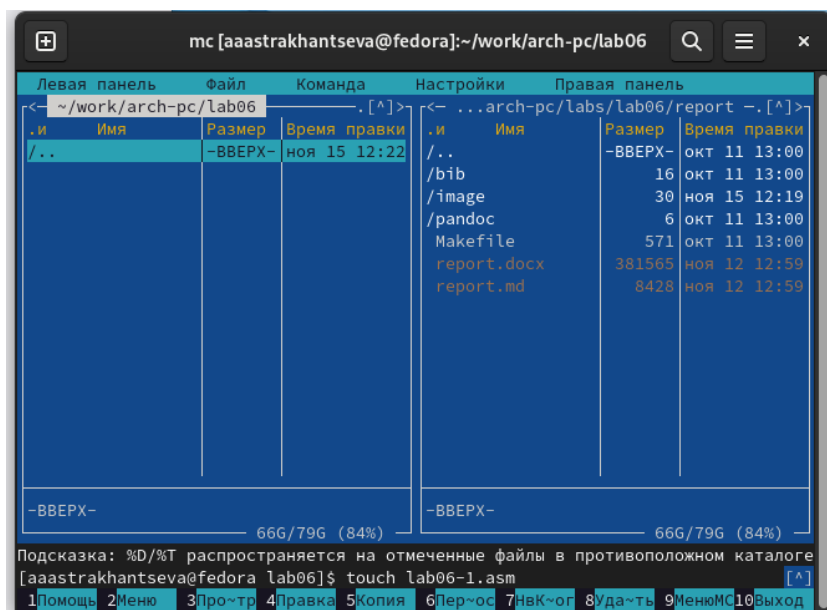


Рис. 2.3: Создание lab06-1.asm

Далее с помощью клавиши F4 открываю созданный файл для редактирование во встроенном редакторе, записываю текст программы, сохраняю его (рис. 2.4):

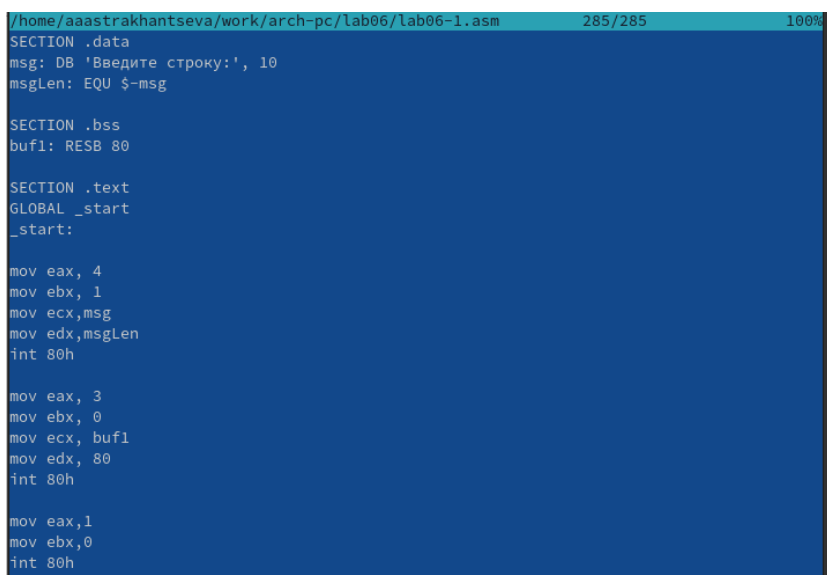


Рис. 2.4: Запись текста программы

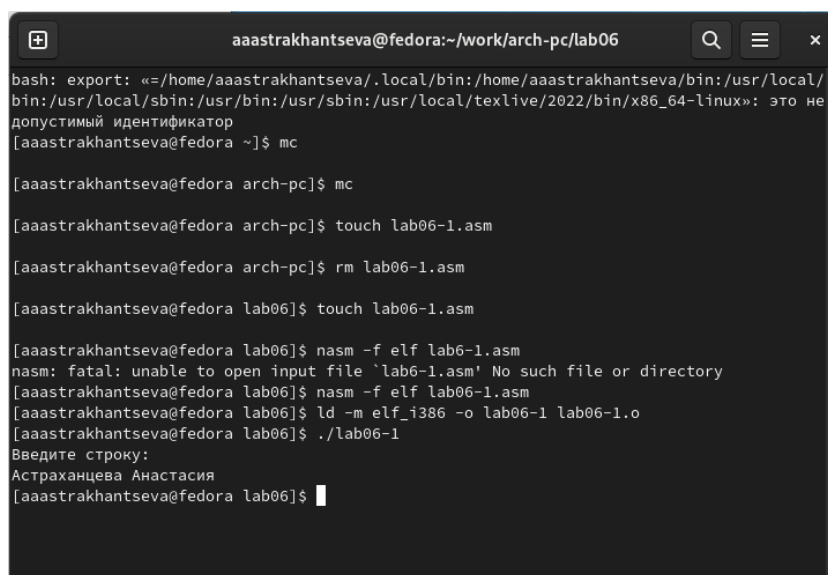
Код программы 1:

```
;-----  
; 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
```

После этого нужно создать объектный файл lab06-1.o. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл с помощью команд (рис. 2.5):



```
bash: export: «=/home/aaastrakhantseva/.local/bin:/home/aaastrakhantseva/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/usr/local/textlive/2022/bin/x86_64-linux»: это не допустимый идентификатор
[aaastrakhantseva@fedora ~]$ mc

[aaastrakhantseva@fedora arch-pc]$ mc

[aaastrakhantseva@fedora arch-pc]$ touch lab06-1.asm

[aaastrakhantseva@fedora arch-pc]$ rm lab06-1.asm

[aaastrakhantseva@fedora lab06]$ touch lab06-1.asm

[aaastrakhantseva@fedora lab06]$ nasm -f elf lab6-1.asm
nasm: fatal: unable to open input file `lab6-1.asm' No such file or directory
[aaastrakhantseva@fedora lab06]$ nasm -f elf lab06-1.asm
[aaastrakhantseva@fedora lab06]$ ld -m elf_i386 -o lab06-1 lab06-1.o
[aaastrakhantseva@fedora lab06]$ ./lab06-1
Введите строку:
Астраханцева Анастасия
[aaastrakhantseva@fedora lab06]$
```

Рис. 2.5: Комановка и запуск lab06-1.asm

2.1 Подключение внешнего файла in_out.asm

Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход из программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную программу более удобной для написания и чтения.

Далее необходимо скачать файл in_out.asm со страницы курса в ТУИС. Подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется, поэтому с помощью клавиши F5 я копирую скачанный

файл в папку lab06 (рис. 2.6):

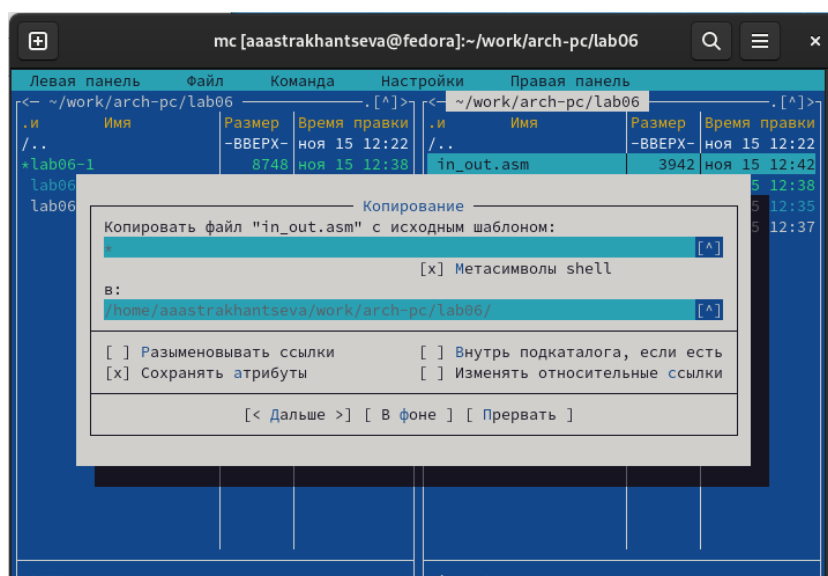


Рис. 2.6: Перенос in_out.asm

С помощью клавиши F6 создаю копию файла lab06-1 с именем lab06-2 (рис. 2.7):

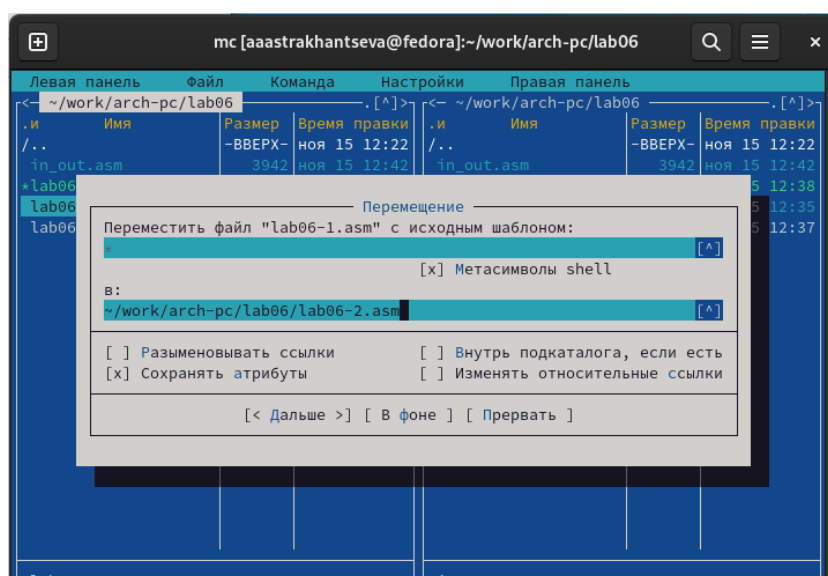
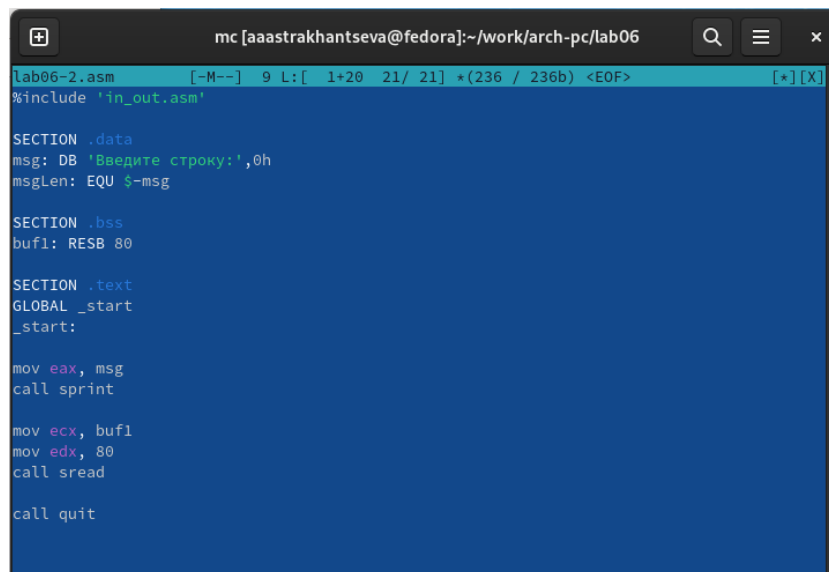


Рис. 2.7: Создание lab06-2

После этого я исправляю текст программы в файле lab06-2(рис. 2.8):



```
mc [aastrakhantseva@fedora]:~/work/arch-pc/lab06
lab06-2.asm  [-M--]  9 L: [ 1+20 21/ 21] *(236 / 236b) <EOF>  [*][X]
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку:',0h
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprint

mov ecx, buf1
mov edx, 80
call sread

call quit
```

Рис. 2.8: Правка текста программы

Код программы 2:

```
;-----
; 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`

После этого нужно оттранслировать текст программы lab06-2.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл с помощью команд: (рис. 2.9):

```
[aaastrakhantseva@fedora lab06]$ nasm -f elf lab06-2.asm
[aaastrakhantseva@fedora lab06]$ ld -m elf_i386 -o lan06-2 lab06-2.o
[aaastrakhantseva@fedora lab06]$ ./lab06-2
bash: ./lab06-2: Нет такого файла или каталога
[aaastrakhantseva@fedora lab06]$ ls
in_out.asm  lab06-1  lab06-1.o  lab06-2.asm  lab06-2.o  lan06-2
[aaastrakhantseva@fedora lab06]$ ./lan06-2
Введите строку:Astrakhantseva Anastasiia
[aaastrakhantseva@fedora lab06]$
```

Рис. 2.9: Запуск lab06-2.asm

2.2 Задания для самостоятельной работы

Для начала создаю копию файла lab06-1.asm с именем lab06-1-1.asm (рис. 2.10):

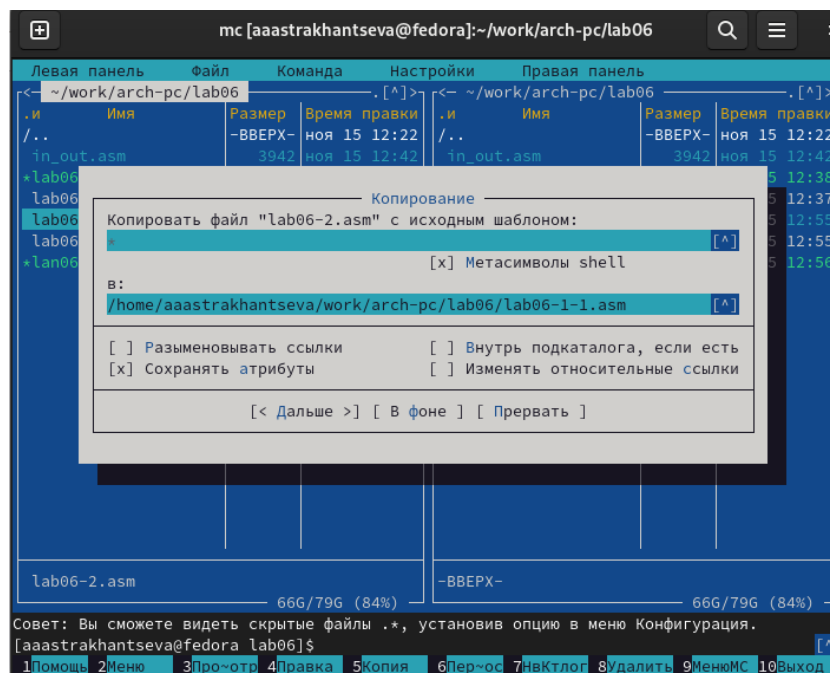
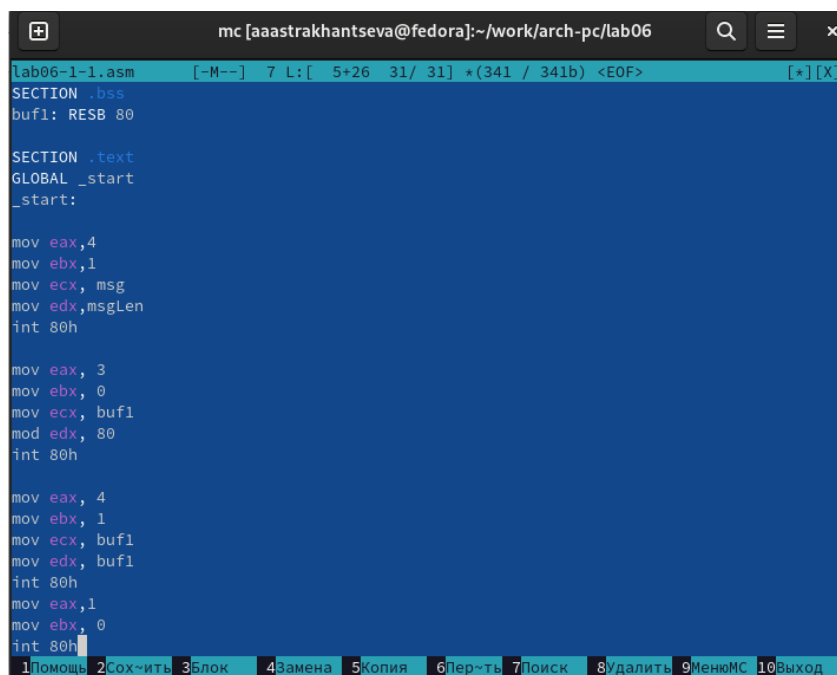


Рис. 2.10: Запуск lab06-2.asm

После этого вношу изменения в программу (рис. 2.11):



```
lab06-1-1.asm [-M--] 7 L: [ 5+26 31/ 31] *(341 / 341b) <EOF> [*] [X]
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h

mov eax,3
mov ebx,0
mov ecx,buf1
mod edx,80
int 80h

mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,buf1
int 80h
mov eax,1
mov ebx,0
int 80h
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Меню 10Выход

Рис. 2.11: Изменение lab06-1.asm

Код программы 3:

```
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10
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

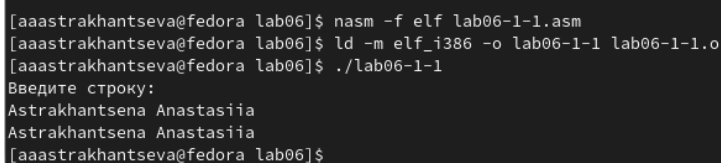
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
```

```

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
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, buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax, 1 ; Системный вызов для выхода (sys_exit)
mov ebx, 0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

После этого нужно создать объектный файл lab06-1-1.o. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл с помощью команд (рис. 2.12):



```

[aaastrakhantseva@fedora lab06]$ nasm -f elf lab06-1-1.asm
[aaastrakhantseva@fedora lab06]$ ld -m elf_i386 -o lab06-1-1 lab06-1-1.o
[aaastrakhantseva@fedora lab06]$ ./lab06-1-1
Введите строку:
Astrakhantsena Anastasiia
Astrakhantsena Anastasiia
[aaastrakhantseva@fedora lab06]$

```

Рис. 2.12: Изменение lab06-1.asm

Далее копирую файл lab06-2.asm с именем lab06-2-1.asm (рис. 2.13):

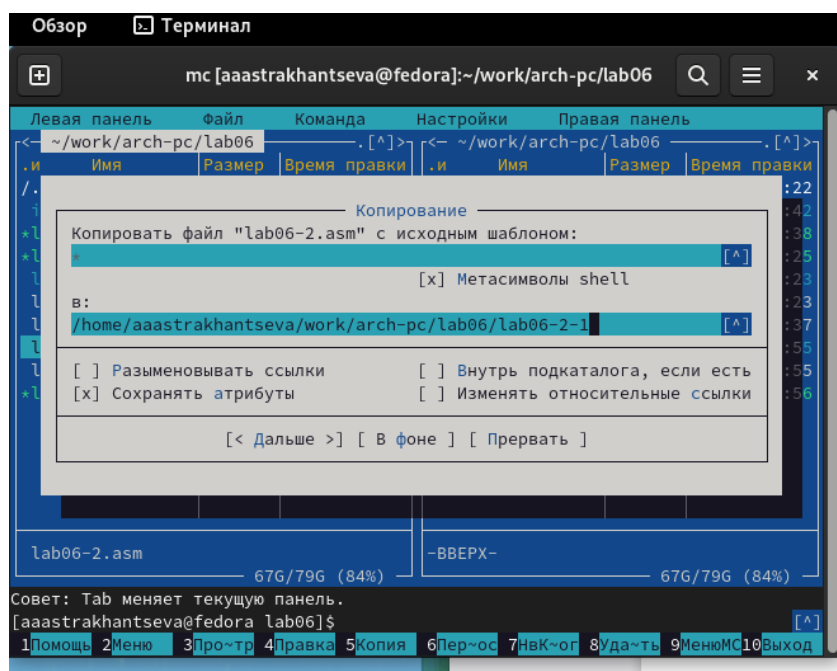


Рис. 2.13: Создание lab06-2-1.asm

Вношу изменения в программу, чтобы она выводила сторчку на экран (рис. 2.14):

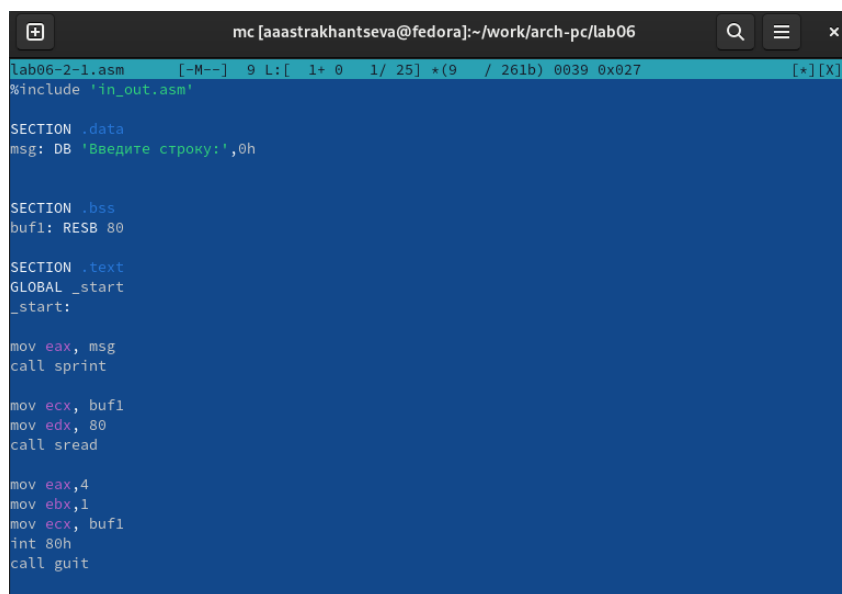


Рис. 2.14: Изменения в lab06-2-1.asm

Код программы 4:

```

%include 'in_out.asm'

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,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения

```

После этого нужно создать объектный файл lab06-2-1.o. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл с помощью команд (рис. 2.15):


```
[aaastrakhantseva@fedora lab06]$ nasm -f elf lab06-2-1.asm
[aaastrakhantseva@fedora lab06]$ d -m elf_i386 -o lab06-2-1 lab06-2-1.o
bash: d: команда не найдена...
[aaastrakhantseva@fedora lab06]$ -m elf_i386 -o lab06-2-1 lab06-2-1.o
bash: -m: команда не найдена...
[aaastrakhantseva@fedora lab06]$ ld -m elf_i386 -o lab06-2-1 lab06-2-1.o
[aaastrakhantseva@fedora lab06]$ ./lab06-2-1
Введите строку:
Astrakhantseva Abastasiia
Astrakhantseva Abastasiia
[aaastrakhantseva@fedora lab06]$ ./lab06-2-1
Введите строку:
Astrakhantseva Anastasiia
Astrakhantseva Anastasiia
[aaastrakhantseva@fedora lab06]$
```

Рис. 2.15: Изменения в lab06-2-1.asm

3 Выводы

Я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера `mov` и `int`.