

Отчёт по лабораторной работе №6

дисциплина: Архитектура компьютера

Сычев Егор Олегович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Самостоятельная работа	14
4	Вывод	18

Список иллюстраций

2.1	Midnight Commander	6
2.2	Создание каталога	7
2.3	Файл lab6-1.asm	8
2.4	mcedit	9
2.5	Проверка	10
2.6	Преобразования и запуск	11
2.7	Файл in_out.asm	11
2.8	Файл lab6-2.asm	12
2.9	mcedit	12
2.10	Преобразования и запуск	13
2.11	mcedit	13
2.12	Преобразования и запуск	13
3.1	Файл lab6-3.asm	14
3.2	mcedit	15
3.3	Преобразования и запуск	16
3.4	Файл lab6-4.asm	16
3.5	mcedit	17
3.6	Преобразования и запуск	17

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Открываем Midnight Commander.

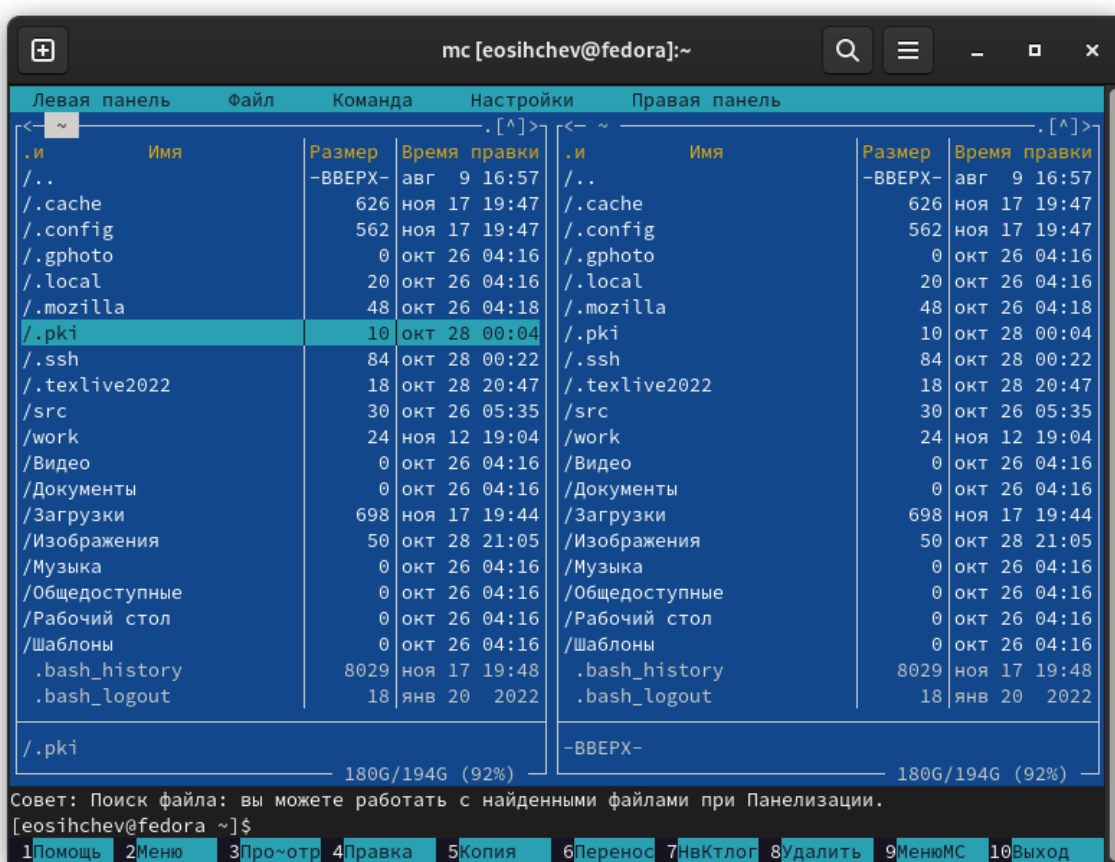


Рис. 2.1: Midnight Commander

2. Переходим в каталог ~/work/arch-pc и создаем каталог lab06.

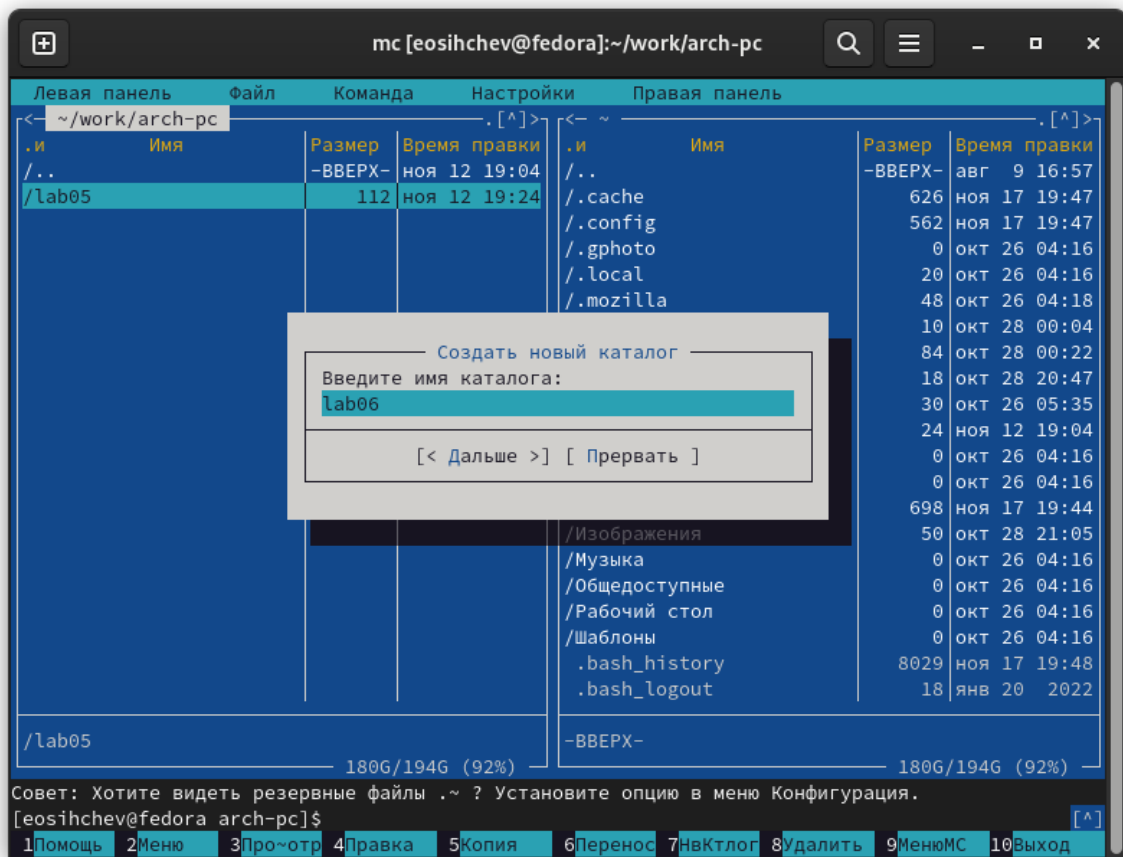


Рис. 2.2: Создание каталога

3. Создаем файл `lab6-1.asm` при помощи команды `touch`.

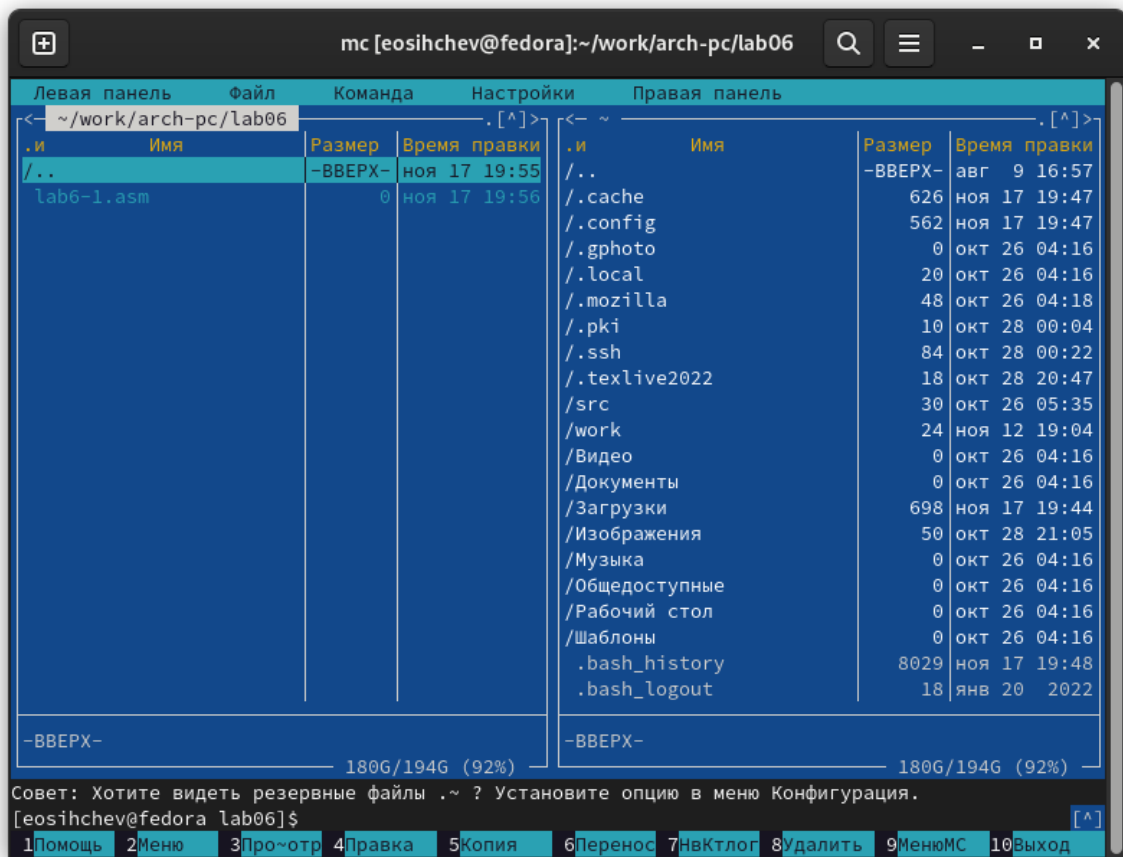


Рис. 2.3: Файл lab6-1.asm

4. Вводим текст программы в созданный файл с помощью mcedit.


```
mc [eosihchev@fedora]:~/work/arch-pc/lab06
lab6-1.asm [----] 0 L: [ 1+35 36/ 36] *(2432/2432b) <EOF>
;
; 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
```

Рис. 2.4: mcedit

```
mc [eiosihchev@fedora]:~/work/arch-pc/lab06
/home/eiosihchev/work/arch-pc/lab06/lab6-1.asm 2432/2432
;-----
; 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
```

Рис. 2.5: Проверка

5. Оттранслируем текст программы lab6-1.asm в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```

[eosihchev@fedora lab06]$ nasm -f elf lab6-1.asm
[eosihchev@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[eosihchev@fedora lab06]$ ls
lab6-1  lab6-1.asm  lab6-1.o
[eosihchev@fedora lab06]$ ./lab6-1
Введите строку:
Сычев Егор Олегович
[eosihchev@fedora lab06]$

```

Рис. 2.6: Преобразования и запуск

6. Скачиваем файл in_out.asm и перемещаем его в нужный каталог.

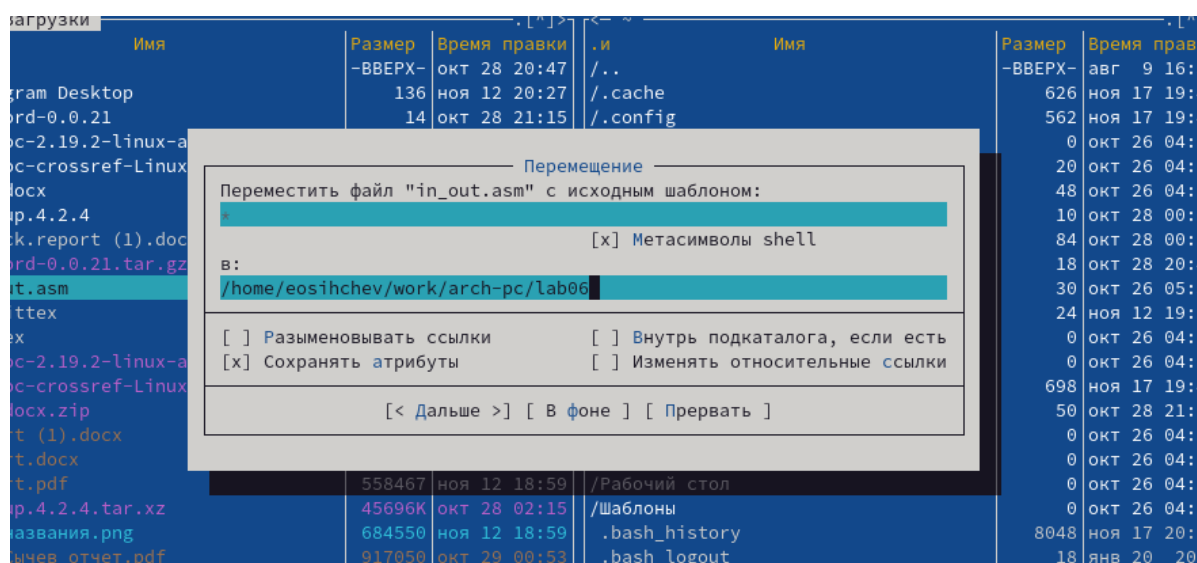


Рис. 2.7: Файл in_out.asm

7. Создаем копию файлы lab6-1.asm с именем lab6-2.asm.

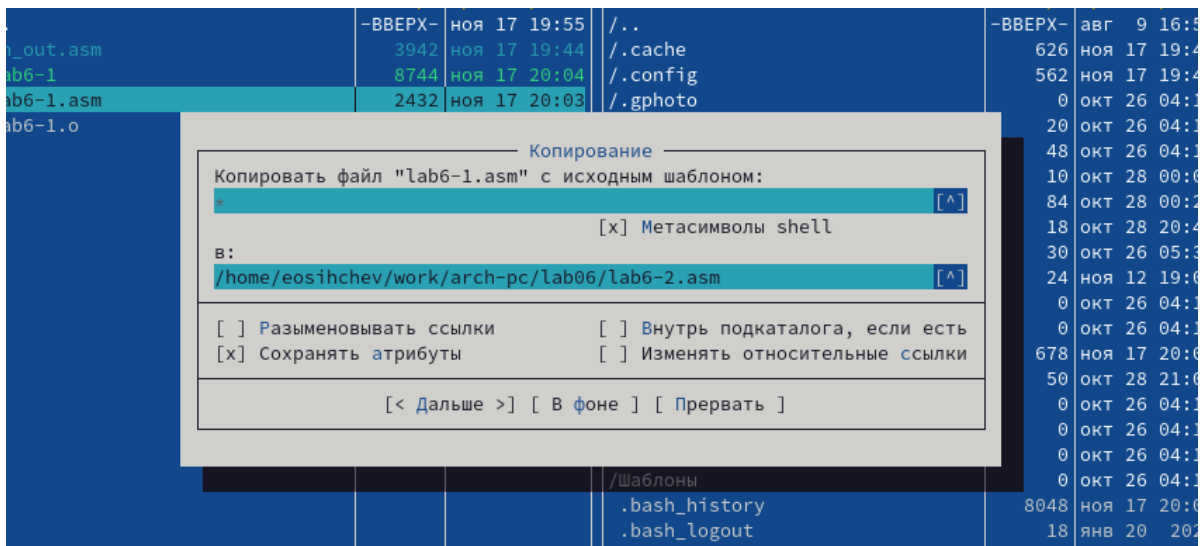


Рис. 2.8: Файл lab6-2.asm

8. Редактируем файл lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm.

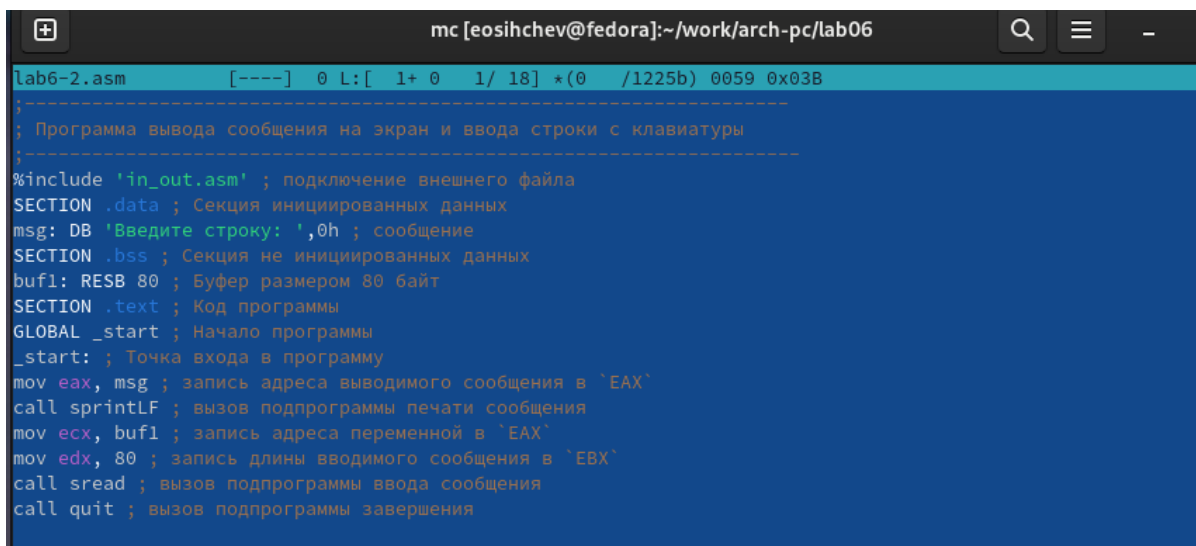


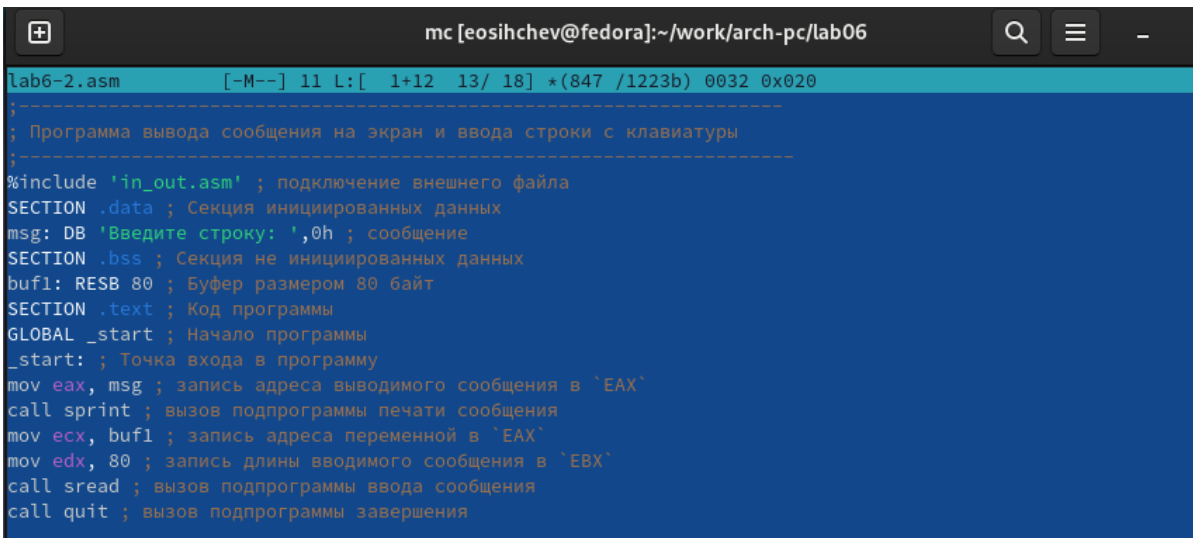
Рис. 2.9: mcedit

9. Оттранслируем текст программы lab6-2.asm в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```
[eosihchev@fedora lab06]$ nasm -f elf lab6-2.asm
[eosihchev@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[eosihchev@fedora lab06]$ ./lab6-2
Введите строку:
Сычев Егор Олегович
[eosihchev@fedora lab06]$
```

Рис. 2.10: Преобразования и запуск

10. В файле lab6-2.asm заменяем подпрограмму sprintLF на sprint.



```
lab6-2.asm      [-M--] 11 L: [ 1+12 13/ 18] *(847 /1223b) 0032 0x020
;-----
; 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 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. 2.11: mcedit

11. Оттранслируем текст программы lab6-2.asm в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```
[eosihchev@fedora lab06]$ nasm -f elf lab6-2.asm
[eosihchev@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[eosihchev@fedora lab06]$ ./lab6-2
Введите строку: Сычев Егор Олегович
[eosihchev@fedora lab06]$
```

Рис. 2.12: Преобразования и запуск

3 Самостоятельная работа

1. Создаем копию файла lab6-1.asm с именем lab6-3.asm.

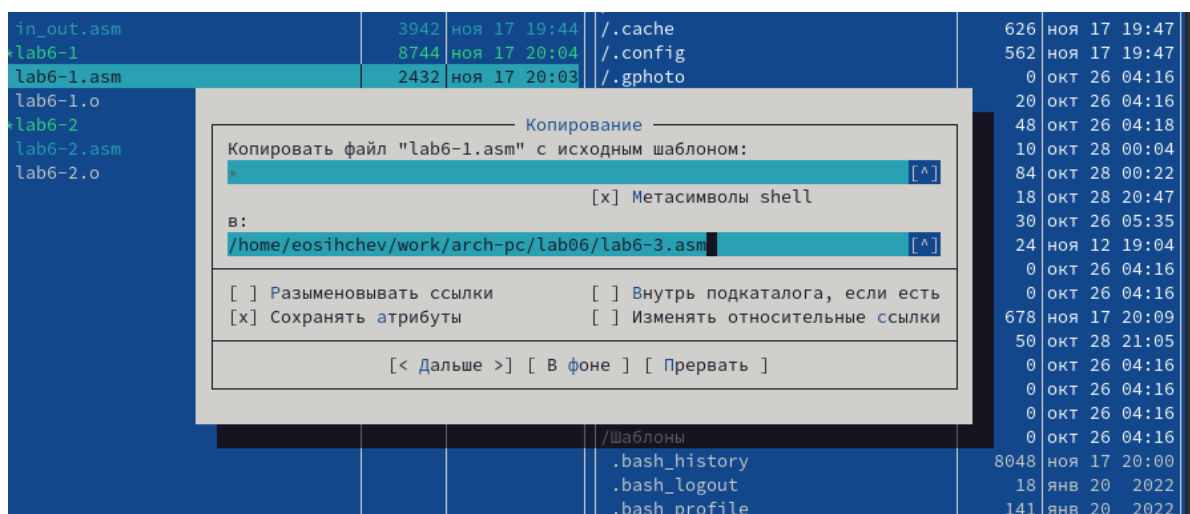
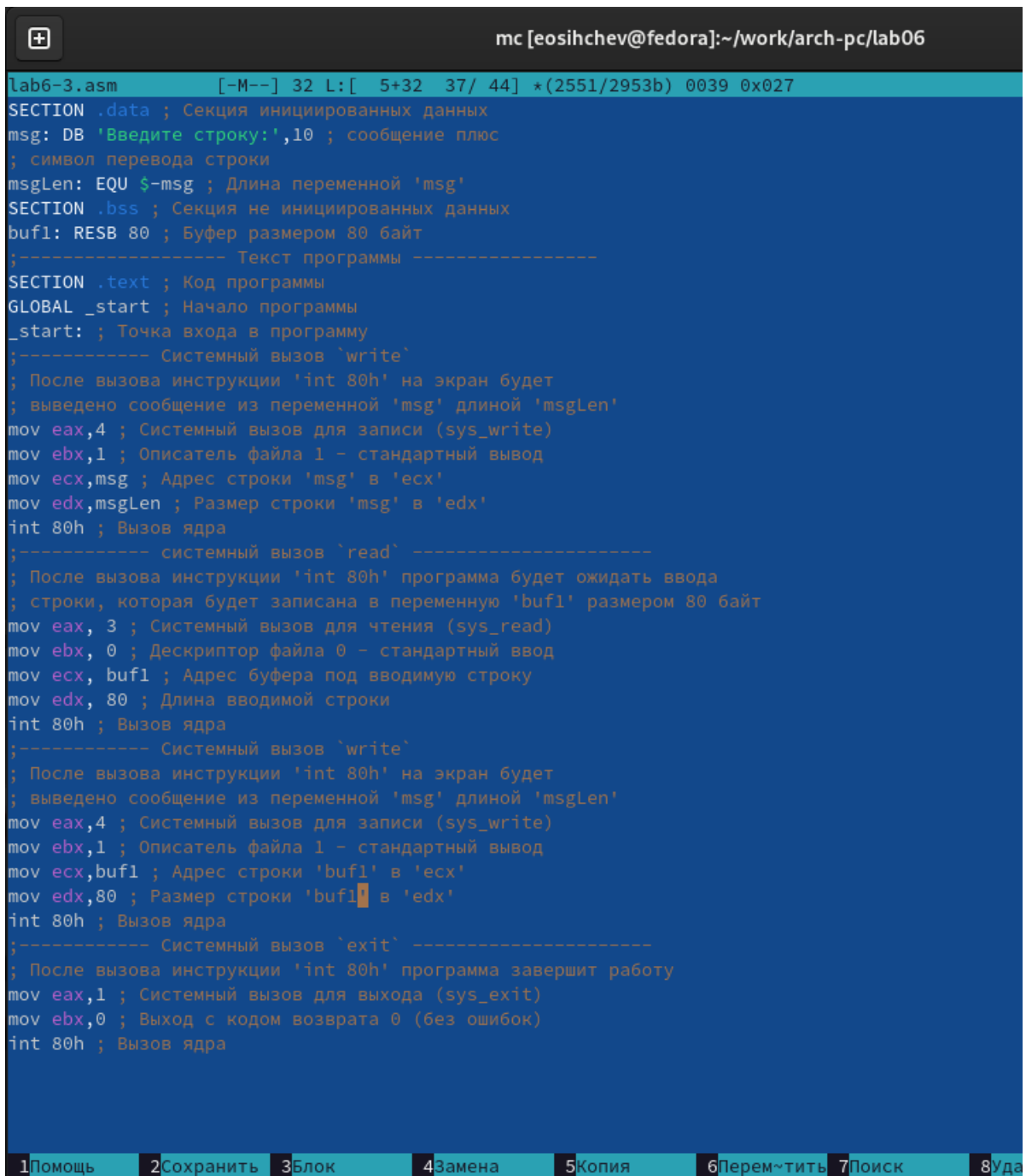


Рис. 3.1: Файл lab6-3.asm

2. Вносим изменения в программу файла lab6-3.asm, так чтобы она работала по нужному алгоритму.



```
mc [eosihchev@fedora]:~/work/arch-pc/lab06
lab6-3.asm [-M--] 32 L:[ 5+32 37/ 44] *(2551/2953b) 0039 0x027
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
;----- 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 'buf1' в 'ecx'
mov edx,80 ; Размер строки 'buf1' 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
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перем~тить 7Поиск 8Уда

Рис. 3.2: mcedit

3. Оттранслируем текст программы lab6-3.asm в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```
[eosihchev@fedora lab06]$ nasm -f elf lab6-3.asm
[eosihchev@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[eosihchev@fedora lab06]$ ./lab6-3
Введите строку:
Сычев Егор Олегович
Сычев Егор Олегович
[eosihchev@fedora lab06]$
```

Рис. 3.3: Преобразования и запуск

4. Создаем копию файла lab6-2.asm с именем lab6-4.asm.

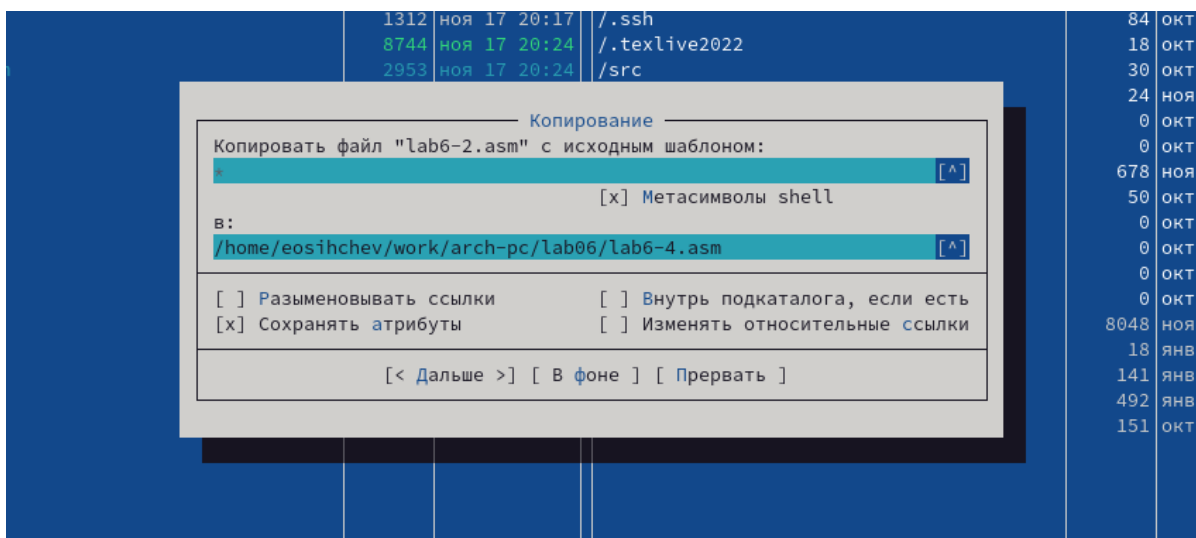
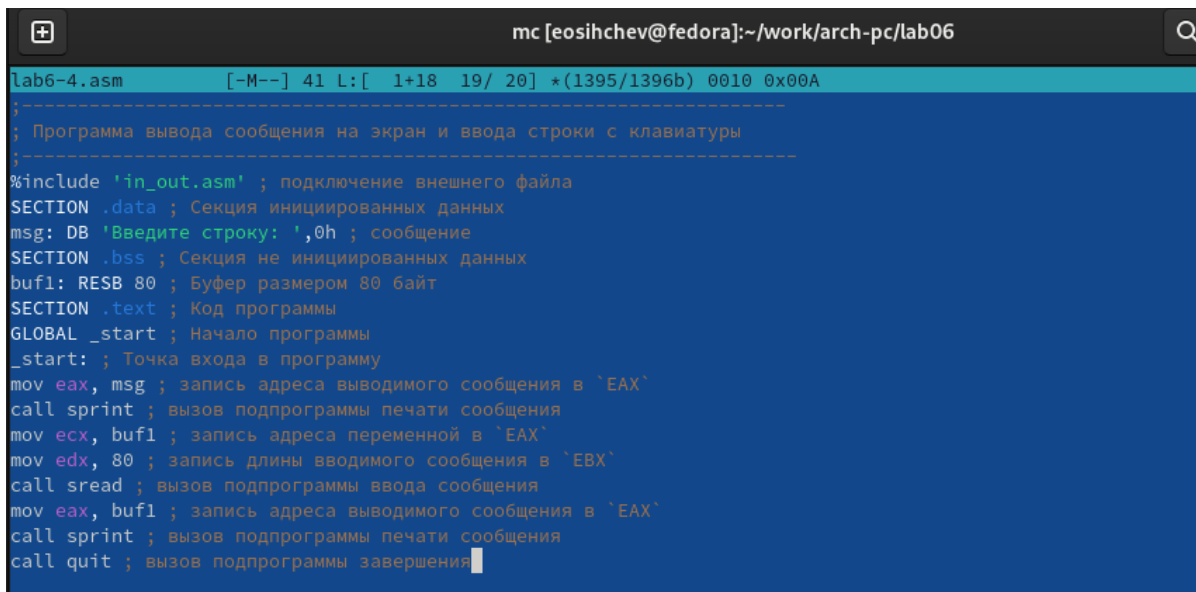


Рис. 3.4: Файл lab6-4.asm

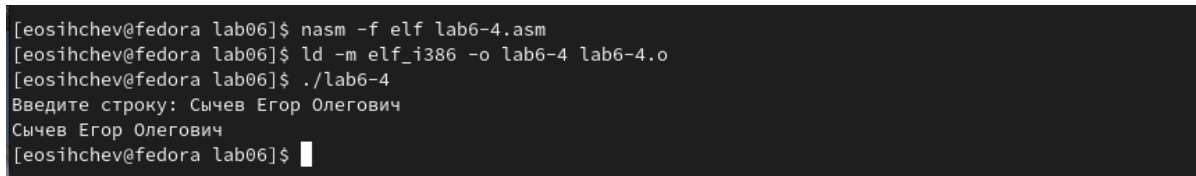
5. Вносим изменения в программу файла lab6-4.asm, так чтобы она работала по нужному алгоритму.



```
mc [eosihchev@fedora]:~/work/arch-pc/lab06
lab6-4.asm [-M--] 41 L: [ 1+18 19/ 20] *(1395/1396b) 0010 0x00A
;-----
; 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
mov eax, buf1 ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 3.5: mcedit

6. Оттранслируем текст программы lab6-4.asm в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл.



```
[eosihchev@fedora lab06]$ nasm -f elf lab6-4.asm
[eosihchev@fedora lab06]$ ld -m elf_i386 -o lab6-4 lab6-4.o
[eosihchev@fedora lab06]$ ./lab6-4
Введите строку: Сычев Егор Олегович
Сычев Егор Олегович
[eosihchev@fedora lab06]$
```

Рис. 3.6: Преобразования и запуск

4 Вывод

Я приобрел практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера `mov` и `int`.