

Отчёта по лабораторной работе 6

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM

ОНВУДИВЕ ВИКТОР ЧИБУИКЕ!

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Окно Midnight Commander Создание каталога	8
4.2	Создание файлов в Midnight Commander	9
4.3	Редактирование файла 1 в Midnight Commander	10
4.4	Проверка программы 2	10
4.5	Файл in_out.asm	11
4.6	Окно Midnight Commander. Копирование файла	11
4.7	создать lab6-2.asm	12
4.8	Редактирование файла 2 в Midnight Commander	13
4.9	Проверка программы 3	13
4.10	Редактирование файла 4 в Midnight Commander	14
4.11	Проверка программы 5	14

Список таблиц

1 Цель работы

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

2 Задание

1. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab6-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:” • ввести строку с клавиатуры; • вывести введённую строку на экран.
4. Загрузите файлы на GitHub.

3 Теоретическое введение

Midnight Commander - это программа, предназначенная для просмотра содержимого каталогов и выполнения основных функций управления файлами в UNIX-подобных операционных системах.

Главное окно программы Midnight Commander состоит из трех полей. Два поля, называемые “панелями”, идентичны по структуре и обычно отображают перечни файлов и подкаталогов каких-то двух каталогов файловой структуры. Эти каталоги в общем случае различны, хотя, в частности, могут и совпасть. Каждая панель состоит из заголовка, списка файлов и информационной строки.

Третье поле экрана, расположенное в нижней части экрана, содержит командную строку текущей оболочки. В этом же поле (самая нижняя строка экрана) содержится подсказка по использованию функциональных клавиш F1 - F10. Самая верхняя строка экрана содержит строку горизонтального меню.

Эта строка может не отображаться на экране; в этом случае доступ к ней можно получить, щелкнув мышью по верхней рамке или нажав клавишу F9.

Панели Midnight Commander обеспечивают просмотр одновременно двух каталогов. Одна из панелей является активной в том смысле, что пользователь может выполнять некоторые операции с отображаемыми в этой панели файлами и каталогами.

4 Выполнение лабораторной работы

1. С помощью функциональной клавиши F7 создайте папку lab06 и перейдите в созданный каталог рис. 4.1)

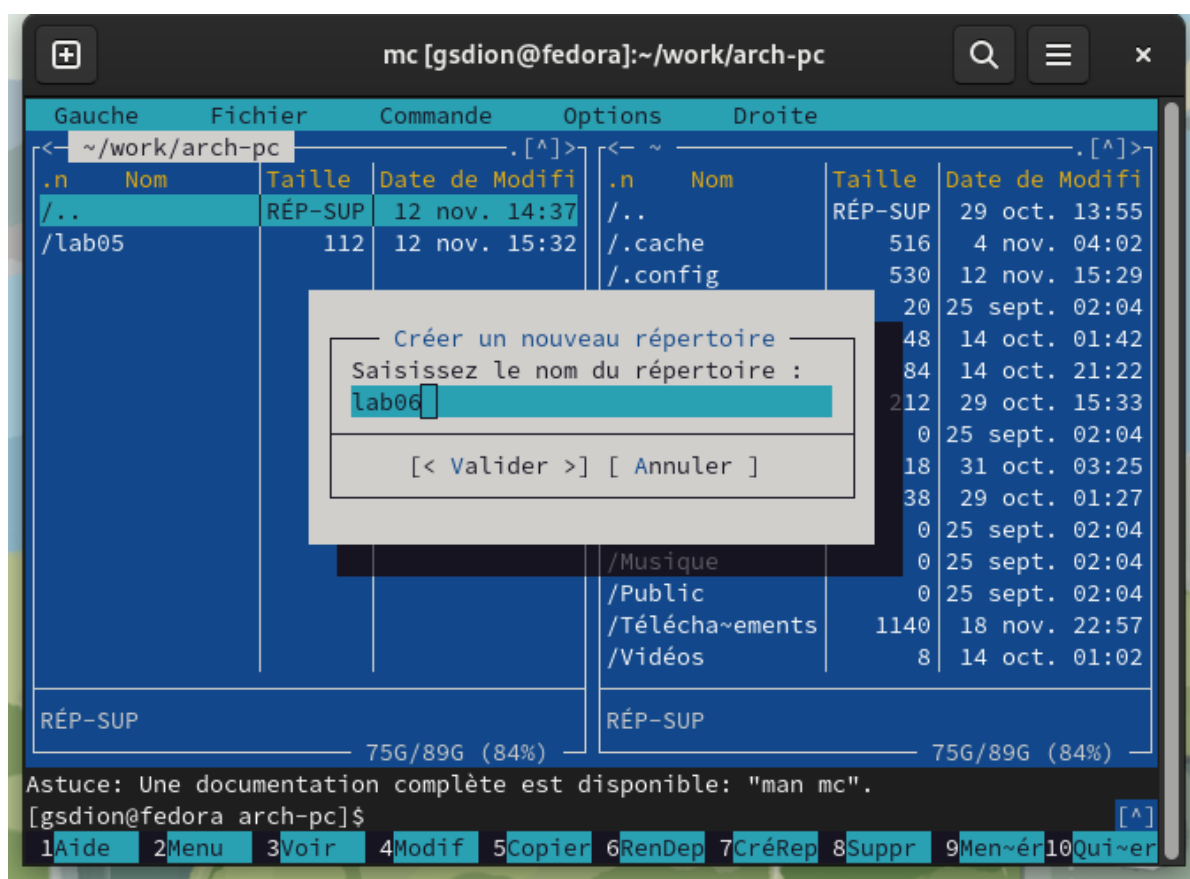


Рис. 4.1: Окно Midnight Commander Создание каталога

2. Создадим новый подкаталог с именем lab06 и в нем файл lab6-1.asm. рис. 4.2)

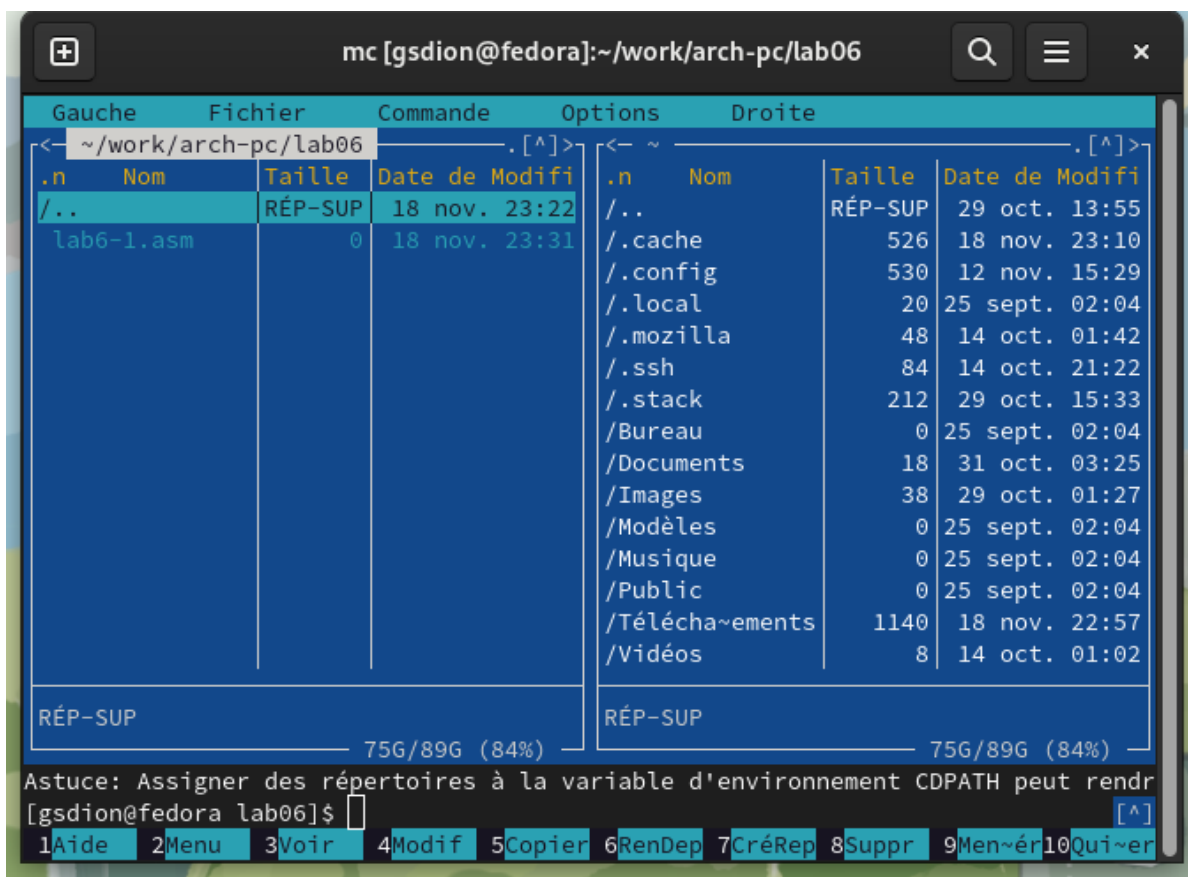


Рис. 4.2: Создание файлов в Midnight Commander

- Введем в файл lab6-1.asm текст программы вывода сообщения на экран и ввода строки с клавиатуры (Листинг 1.). Создадим исполняемый файл и проверим его работу. (рис. 4.3, 4.4)

```
mc [gsdion@fedora]:~/work/arch-pc/lab06
/home/gsdion/work/arch-pc/lab06/lab6-1.asm 2706/2706 100%
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
1Aide 2San~Lg 3Qui~er 4Hex 5All~ à 6 7Rech 8Brut 9Fmter 10Qui~er
```

Рис. 4.3: Редактирование файла 1 в Midnight Commander

```
[gsdion@fedora lab06]$ ./lab6-1
Введите строку:
gsdion
[gsdion@fedora lab06]$
```

Рис. 4.4: Проверка программы 2

4. Скачали с туис доп файл, скопировали программу. (рис. 4.5)

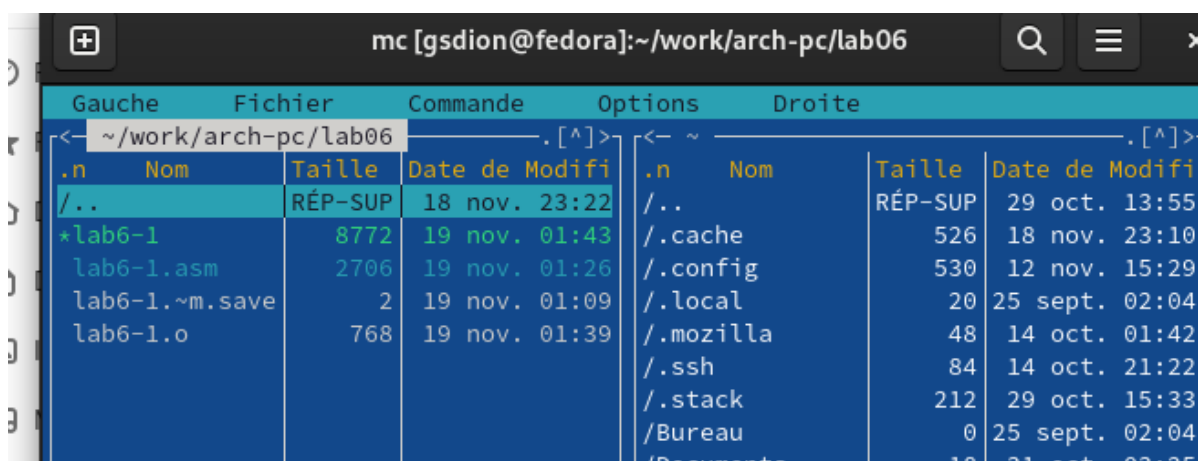


Рис. 4.5: Файл in_out.asm

- копия файла in_out.asm, который должен находиться в том же каталоге, что и файл программы, в котором он используется. (рис. 4.6)

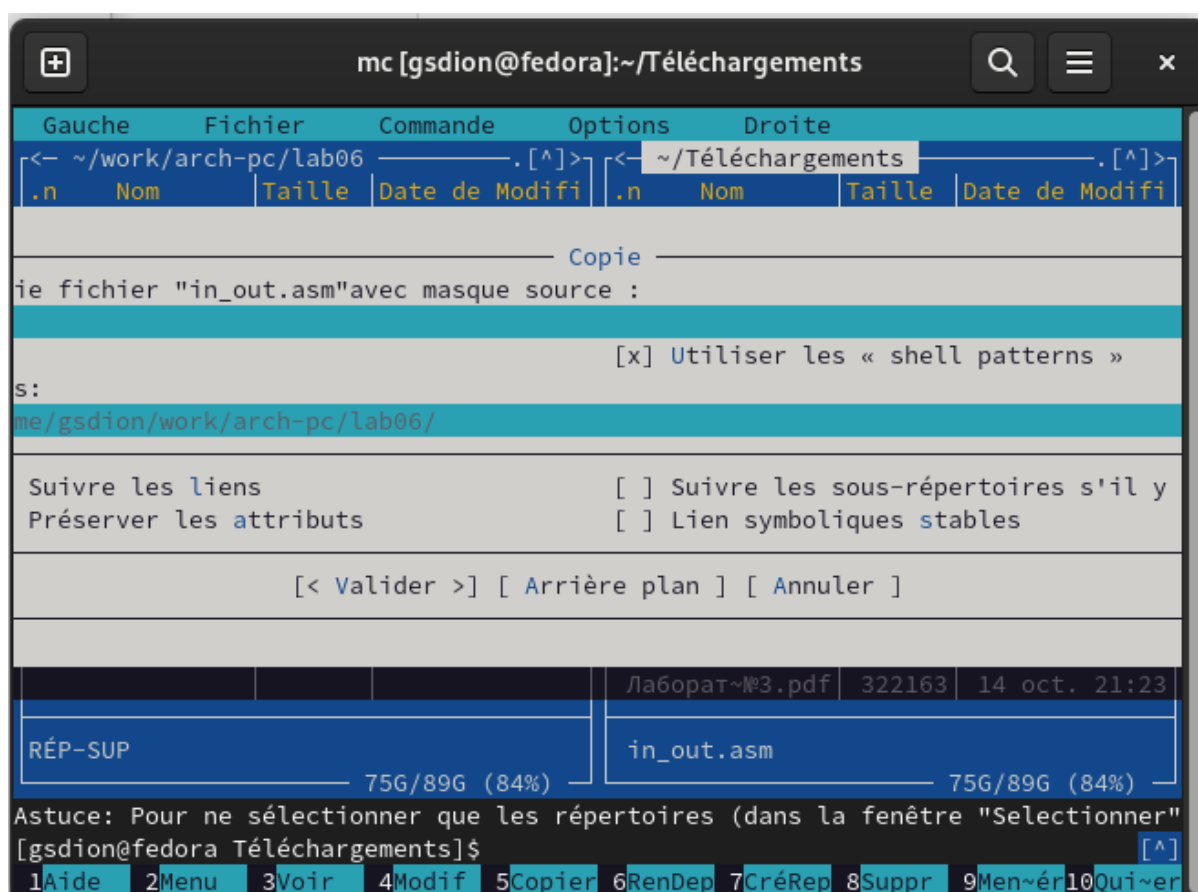


Рис. 4.6: Окно Midnight Commander. Копирование файла

6. Используйте функциональную клавишу F6, чтобы создать копию lab6-1.asm с именем lab6-2.asm. Выделите файл lab6-1.asm, нажмите кнопку F6, введите имя файла lab6-2.asm и нажмите кнопку. (рис. 4.7)

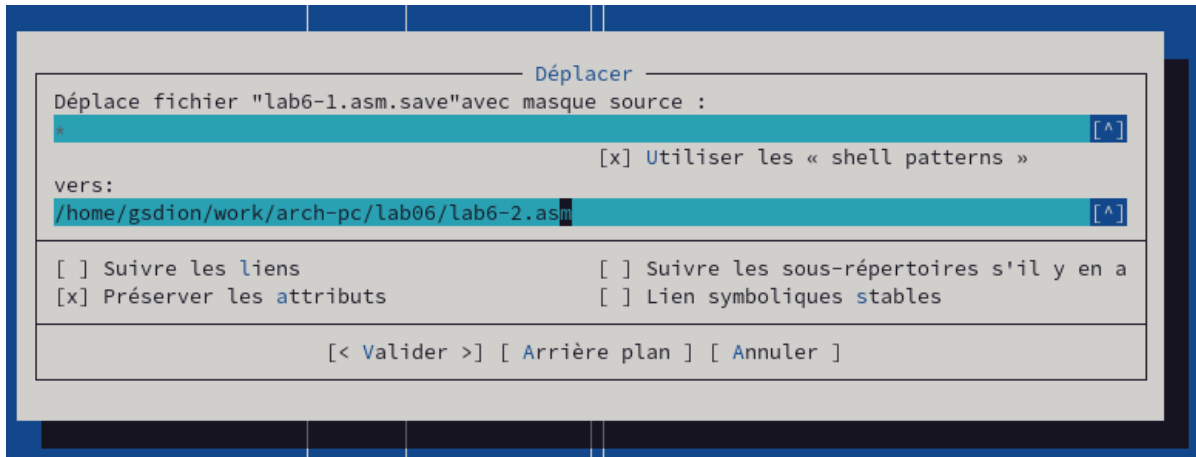
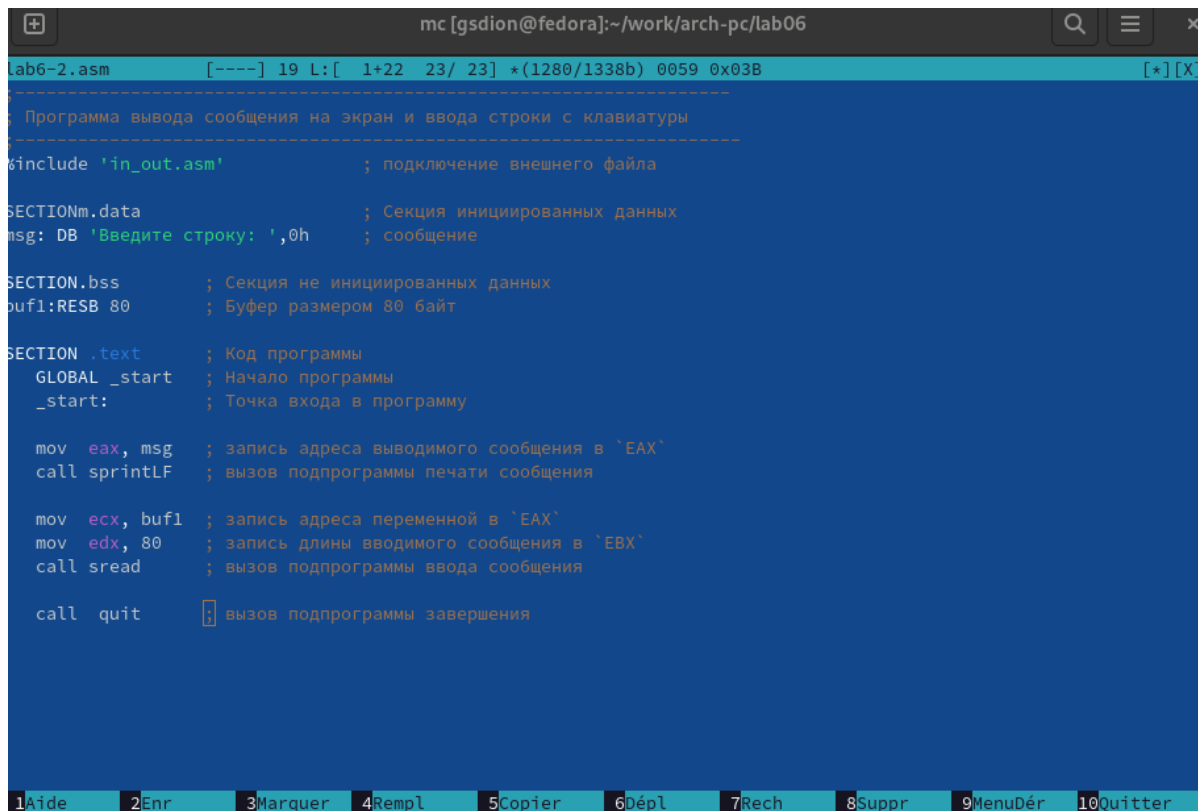


Рис. 4.7: создать lab6-2.asm

7. Введем в файл lab6-1.asm текст программы вывода сообщения на экран и ввода строки с клавиатуры (Листинг 2.). Создадим исполняемый файл и проверим его работу. (рис. 4.8, 4.9)



```
mc [gsdion@fedora]:~/work/arch-pc/lab06
lab6-2.asm [----] 19 L:[ 1+22 23/ 23] *(1280/1338b) 0059 0x03B [*][X]
-----
; 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 printf ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call read ; вызов подпрограммы ввода сообщения

call quit ; вызов подпрограммы завершения
```

Рис. 4.8: Редактирование файла 2 в Midnight Commander

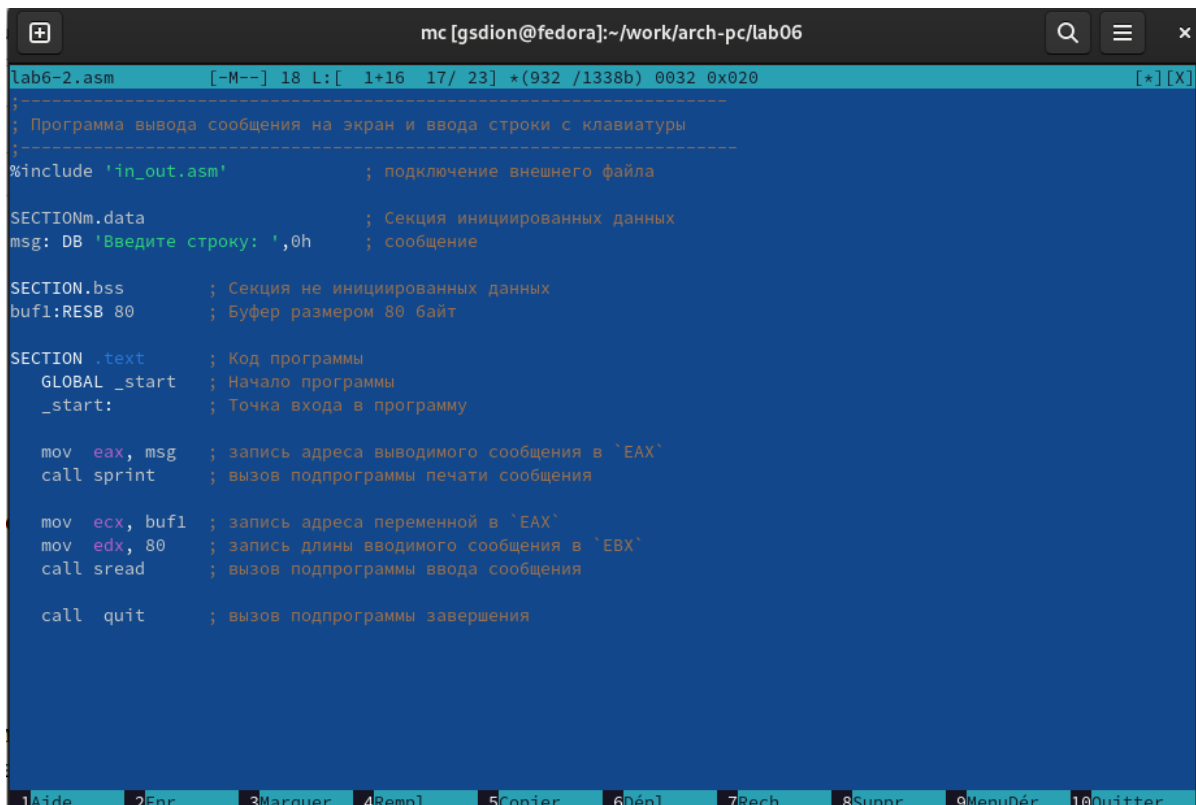


```
[gsdion@fedora lab06]$ ./lab6-2
Введите строку:
Дион Гонсан Седрик Мишель
[gsdion@fedora lab06]$
```

Рис. 4.9: Проверка программы 3

8. Внесем изменения в программу (без использования внешнего файла `in_out.asm`), так чтобы она работала по следующему алгоритму: (рис. 4.10, 4.11)

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.



```
mc [gsdion@fedora]:~/work/arch-pc/lab06
lab6-2.asm [-M--] 18 L: [ 1+16 17/ 23] *(932 /1338b) 0032 0x020 [*] [X]
-----
; 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

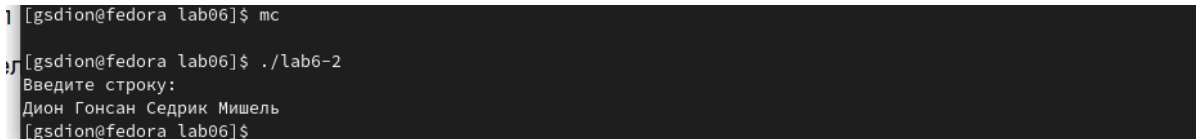
    mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
    call sprint ; вызов подпрограммы печати сообщения

    mov ecx, buf1 ; запись адреса переменной в 'EAX'
    mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
    call sread ; вызов подпрограммы ввода сообщения

    call quit ; вызов подпрограммы завершения

1 Aide 2 Fnc 3 Marquer 4 Rempl 5 Copier 6 Dépl 7 Rech 8 Suppr 9 MenuDéf 10 Quitter
```

Рис. 4.10: Редактирование файла 4 в Midnight Commander



```
[gsdion@fedora lab06]$ mc
[gsdion@fedora lab06]$ ./lab6-2
Введите строку:
Дион Гонсан Седрик Мишель
[gsdion@fedora lab06]$
```

Рис. 4.11: Проверка программы 5

5 Выводы

В заключение можно сказать, что данная лабораторная позволила нам научиться писать основные программы на ассемблере. Владение инструкциями по сборке `mov` и `int`.

Список литературы

1. Справочная система по языку Assembler
2. Midnight Commander