

Лабораторная работа №5. Основы работы с Midnight Commander. Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

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

Ход работы.

Откроем консоль, перейдем в домашний каталог пользователя и запустим Midnight Commander командой mc (рисунок 1).

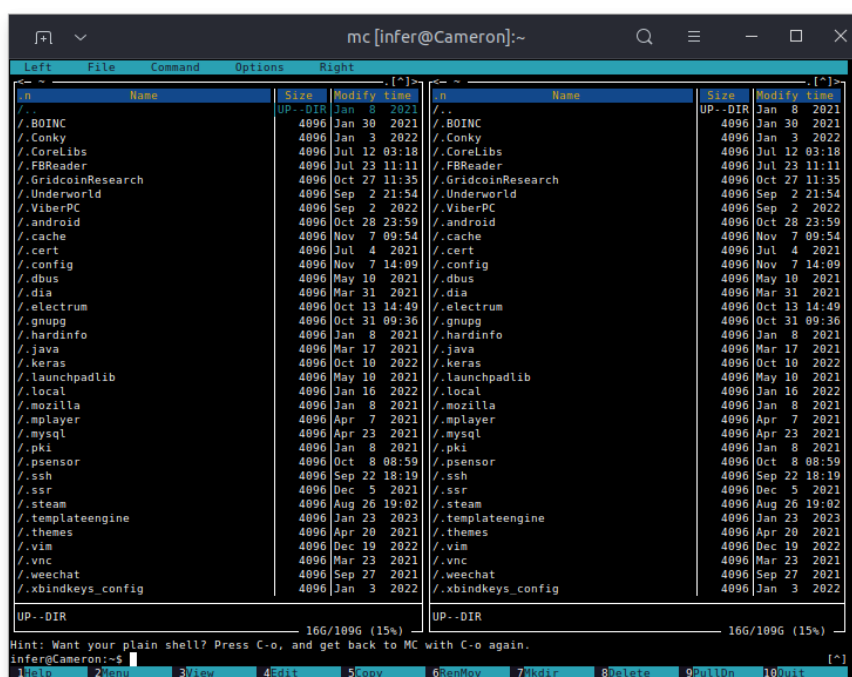


Рисунок 1 — Внешний вид программы Midnight Commander

Нажав кнопку F7 создадим директорию с лабораторной работы (рисунок 2).

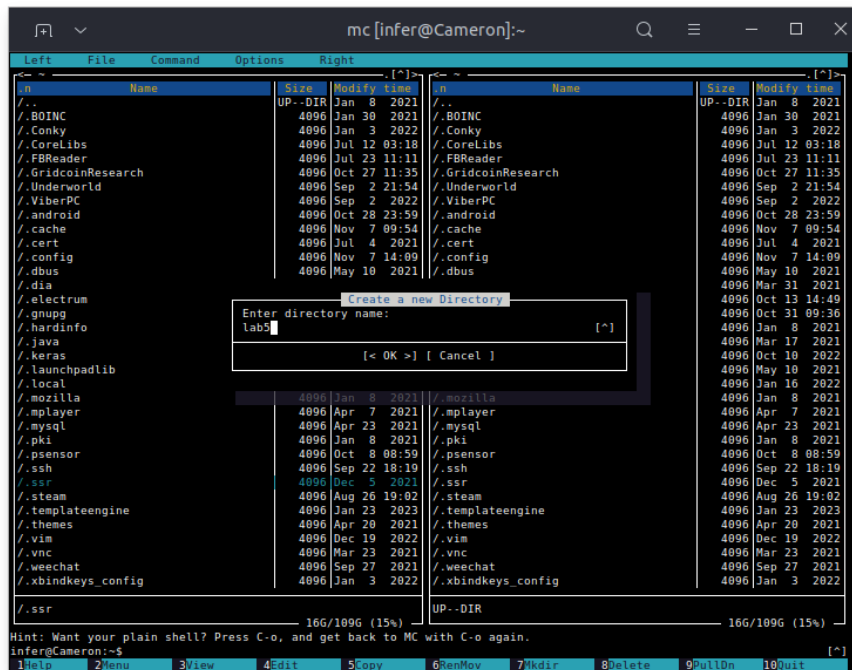


Рисунок 2 — Создание директории lab5

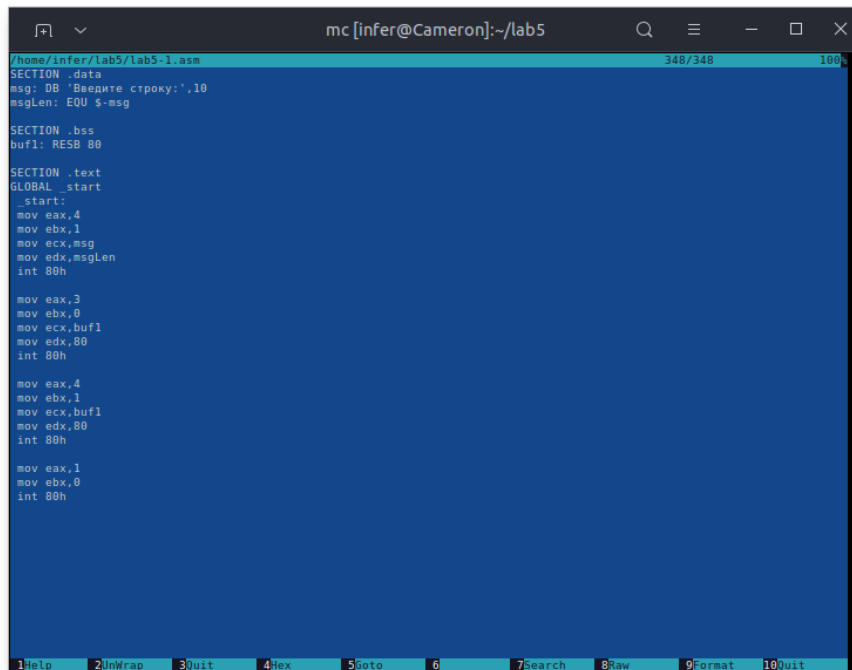
Задание 1.

Создайте копию файла lab5-1.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 введенную строку на экран.

Решение.

Исходный код программы показан на рисунке 3.



```
mc [infer@Cameron]:~/lab5
/home/infer/lab5/lab5-1.asm 348/348 100%
SECTION .data
msg: DB "Введите строку:",10
msgLen: EQU $-msg

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
mov edx,80
int 80h

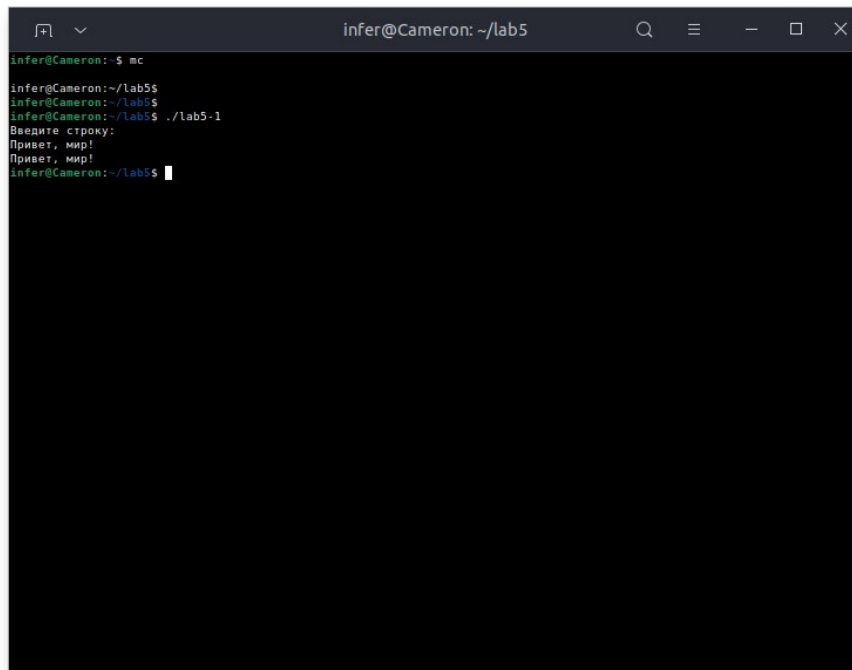
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h

mov eax,1
mov ebx,0
int 80h
```

Рисунок 3 — Исходный код программы

Как видно из исходного кода, для вывода строки на экран, использован тот же фрагмент кода, который использовался для вывода приглашения, за исключением записи в регистр `ecx` содержимого буфера.

Результат выполнения программы показан на рисунке 4. Для переключения между консолью и Midnight Commander используем сочетание клавиш `Ctrl+O`.

A terminal window titled 'infer@Cameron: ~/lab5' with standard window controls. The terminal shows a sequence of commands and their outputs: 'infer@Cameron: \$ mc' followed by a blank line; 'infer@Cameron: ~/lab5\$' followed by a blank line; 'infer@Cameron: ~/lab5\$' followed by a blank line; 'infer@Cameron: ~/lab5\$./lab5-1' followed by the prompt 'Введите строку:'; then the input 'Привет, мир!' followed by the output 'Привет, мир!'; and finally 'infer@Cameron: ~/lab5\$' followed by a blank line and a cursor.

```
infer@Cameron: $ mc
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$ ./lab5-1
Введите строку:
Привет, мир!
Привет, мир!
infer@Cameron: ~/lab5$
```

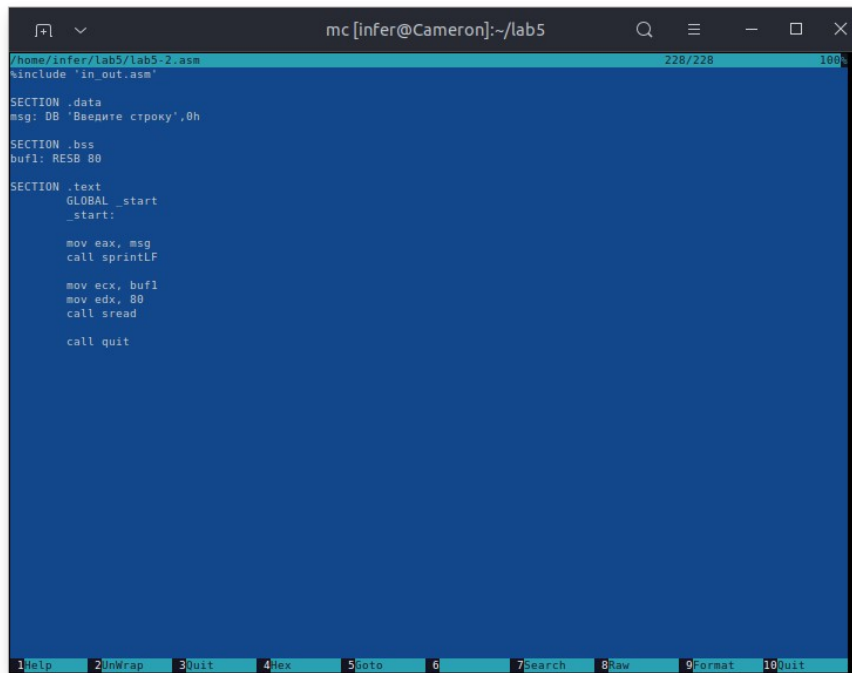
Рисунок 4 — Результат работы программы

Задание 2.

Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.

Решение.

Исходный код программы показан на рисунке 5.



```
mc [infer@Cameron]:~/lab5
/home/infer/lab5/lab5-2.asm 228/228 100%
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку',0h

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax, msg
    call sprintf

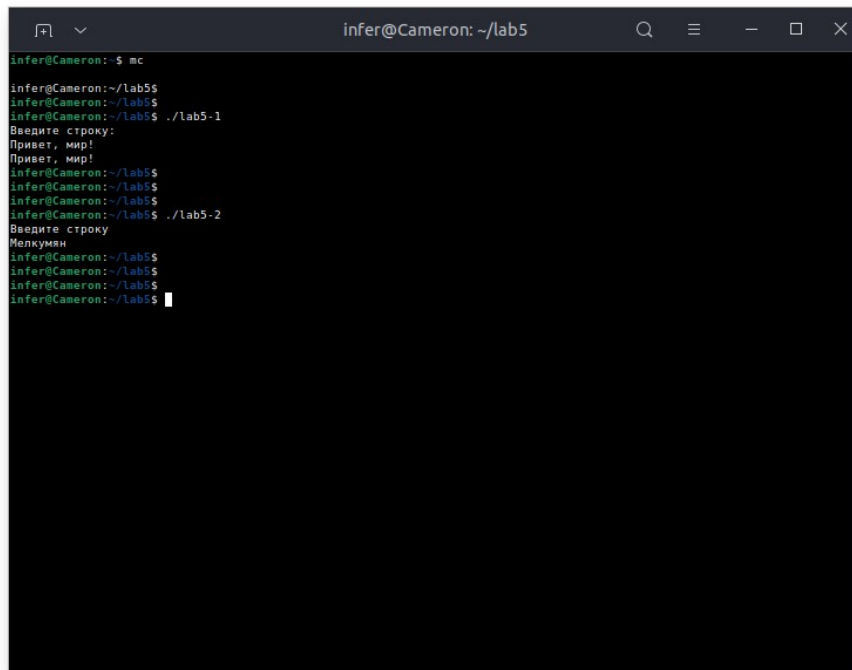
    mov ecx, buf1
    mov edx, 80
    call sread

    call quit
```

Рисунок 5 — Исходный код программы

По сравнению с предыдущей программой, в данной программе типовые задачи вывод сообщения, чтения текста в буфер из консоли и выхода из программы реализуются с помощью функций `sprintf`, `sread` и `quit` соответственно.

Результат выполнения данной программы показан на рисунке 6



```
infer@Cameron: ~/lab5
infer@Cameron: ~/lab5$ mc
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$ ./lab5-1
Введите строку:
Привет, мир!
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$ ./lab5-2
Введите строку:
Мелкуман
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
```

Рисунок 6 – Результат выполнения программы

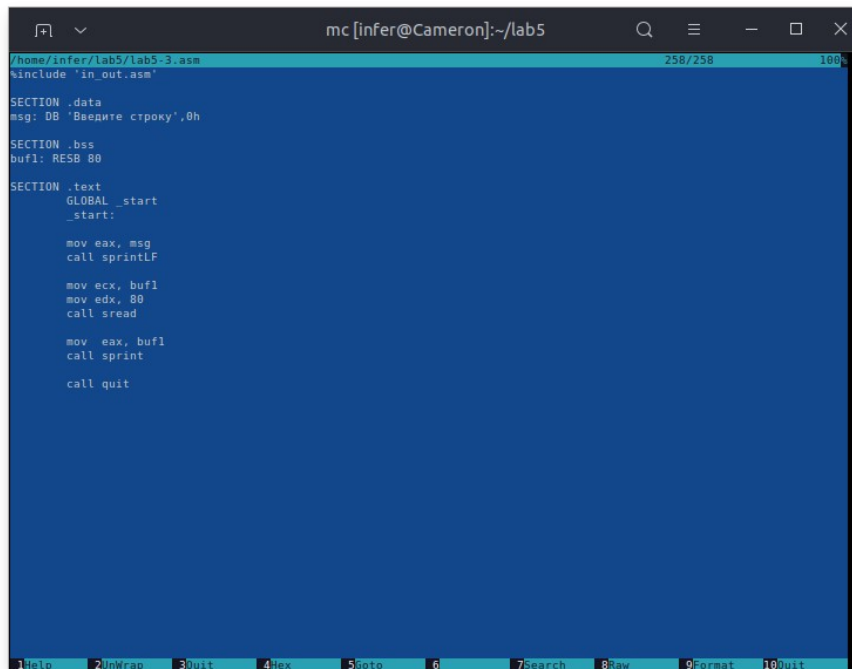
Задание 3.

Создайте копию файла lab5-2.asm. Исправьте текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму:

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

Решение.

Исходный код показан на рисунке. 7. По структуре данная программа идентична программе lab5-1, за исключением того, что типовые задачи ввода, вывода данных и выхода из программы реализованы с помощью функций из файла in_out.asm. Также необходимо отметить, возможность сокращения объема кода, за счет его переиспользования в виде функций.



```
mc [infer@Cameron]:~/lab5
/home/infer/lab5/lab5-3.asm 258/258 100%
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку',0h

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

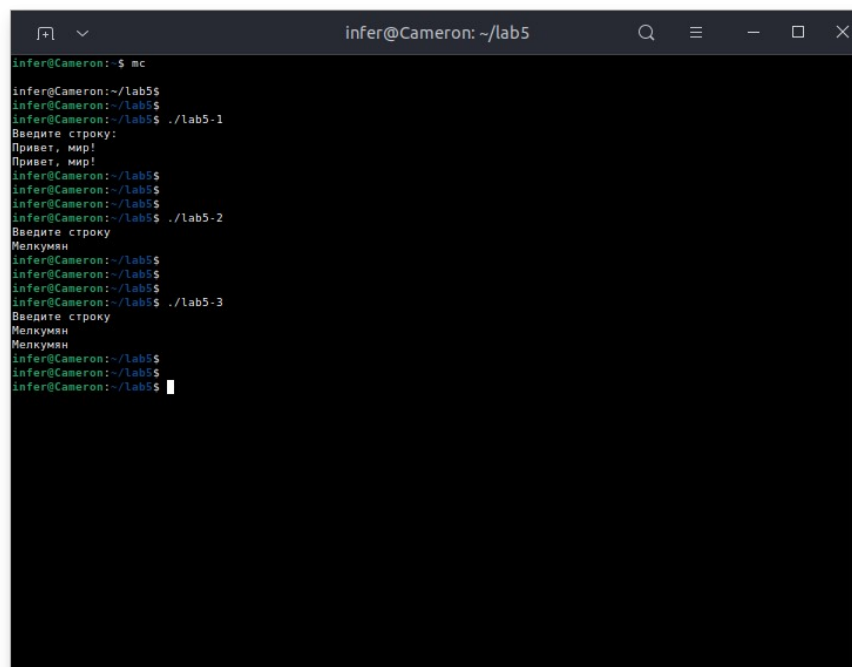
    mov eax, msg
    call sprintf

    mov ecx, buf1
    mov edx, 80
    call sread

    mov eax, buf1
    call sprint
    call quit
```

Рисунок 7 — Исходный код программы

На рисунке 8 показан результат работы программы.



```
infer@Cameron: ~/lab5
infer@Cameron: $ mc
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$ ./lab5-1
Введите строку:
Привет, мир!
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$ ./lab5-2
Введите строку
Мелкумян
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$ ./lab5-3
Введите строку
Мелкумян
infer@Cameron: ~/lab5$
infer@Cameron: ~/lab5$
```

Рисунок 8 — Результат работы программы

После выполнения всех заданий в Midnight Commander можно видеть все созданные файлы (рисунок 9).

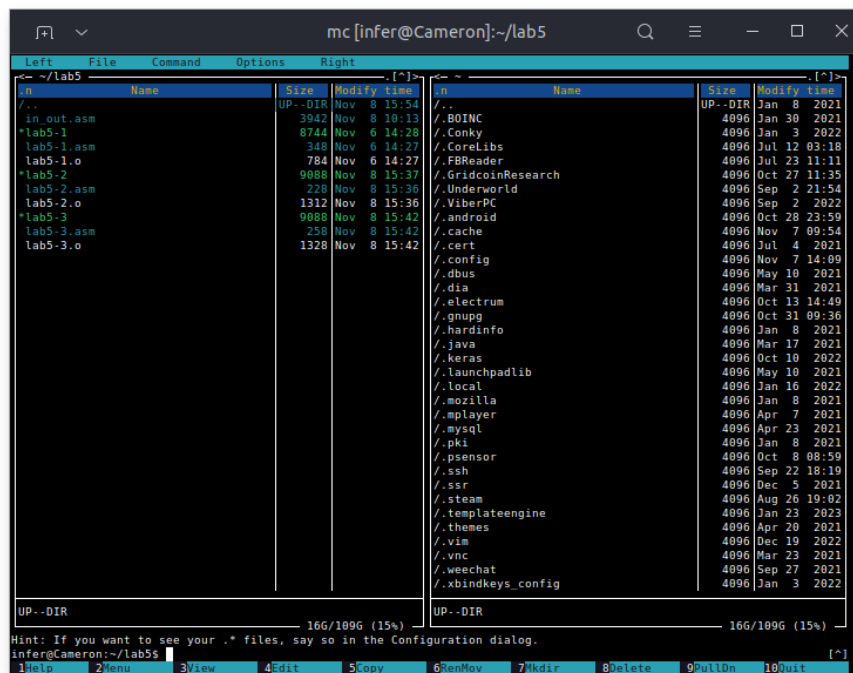


Рисунок 9 — Содержимое каталога lab5

Вывод: В ходе выполнения лабораторной работы я научилась работать с программой Midnight Commander, использовать инструкции `mov` и `int` языка Assembler, а также использовать функции внешнего файла.