

Лабораторная работа №5

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Колонтырский Илья Русланович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	17

Список иллюстраций

2.1	Запуск ms	6
2.2	Создание папки	6
2.3	Создание файла	7
2.4	Редактирование файла	8
2.5	Открытие файла для чтения	9
2.6	Сборка и запуск файла	9
2.7	Копирование файла in_out.asm	10
2.8	Копирование файла	11
2.9	Вставка кода	12
2.10	Проверка и сборка программы	12
2.11	Редактирование программы	13
2.12	Повторная сборка и запуск	13
2.13	изменение скопированного файла	14
2.14	Проверка программы и её сборка	15
2.15	Редактирование копии файла	15
2.16	Проверка работоспособности программы	16

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Откроем mc (рис. 2.1)

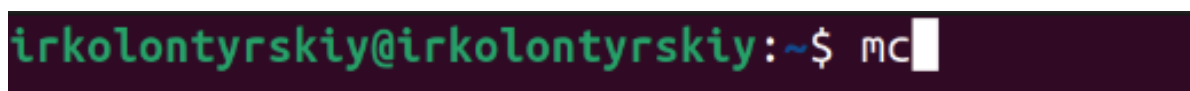


Рис. 2.1: Запуск mc

Перейдём в папку с файлами для лабораторных работ и создадим в ней папку lab05 (рис. 2.2)

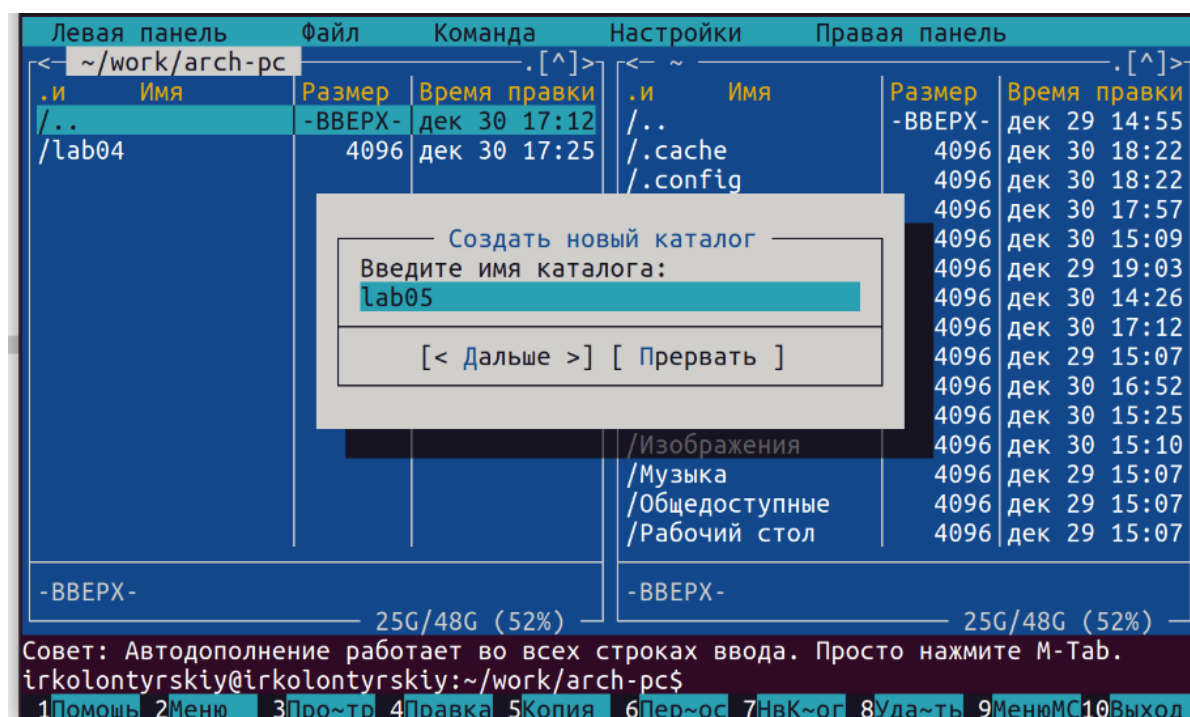


Рис. 2.2: Создание папки

Перейдём в неё и создадим файл lab5-1.asm (рис. 2.3)

Left panel: File Manager (Файл) | Command (Команда) | Settings (Настройки) | Right panel

Left panel: ~ /work/arch-pc/lab05 .[^]>

File Name (Имя)	Size (Размер)	Time (Время)
..	-ВВЕРХ-	дек 30 18:23
lab5-1.asm	0	дек 30 18:23

Right panel: ~ .[^]>

File Name (Имя)	Size (Размер)	Time (Время)
..	-ВВЕРХ-	дек 29 14:55
/.cache	4096	дек 30 18:22
/.config	4096	дек 30 18:22
/.gnupg	4096	дек 30 17:57
/.local	4096	дек 30 15:09
/.ssh	4096	дек 29 19:03
/snap	4096	дек 30 14:26
/work	4096	дек 30 17:12
/Видео	4096	дек 29 15:07
/Документы	4096	дек 30 16:52
/Загрузки	4096	дек 30 15:25
/Изображения	4096	дек 30 15:10
/Музыка	4096	дек 29 15:07
/Общедоступные	4096	дек 29 15:07
/Рабочий стол	4096	дек 29 15:07

Bottom status bar: -ВВЕРХ- 25G/48G (52%) -ВВЕРХ- 25G/48G (52%) -

Footer: Совет: Автодополнение работает во всех строках ввода. Просто нажмите M-Tab. irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05\$

Рис. 2.3: Создание файла

Откроем его для редактирования и вставим следующий код (рис. 2.4)

```
GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab05/lab5-1.asm *
;-----
; 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'

^G Справка    ^O Записать   ^W Поиск      ^K Вырезать   ^T Выполнить  ^C Позиция
^X Выход      ^R ЧитФайл   ^\ Замена     ^U Вставить   ^J Выводить   ^/ К строке
```

Рис. 2.4: Редактирование файла

Сохраним его и откроем файл для просмотра, чтобы убедиться, что всё было сохранено (рис. 2.5)


```
/home/irkolontyrskiy/wo~ch-pc/lab05/lab5-1.asm 1448/2432 59%
;-----
; 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
1Помощь 2Раз~рн 3Выход 4Нех 5Пер~ти 6 7Поиск 8Исх~ый 9Формат10Выход
```

Рис. 2.5: Открытие файла для чтения

Соберём его и проверим на работоспособность (рис. 2.6)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab
5-1.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Колонтырский Илья Русланович
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$
```

Рис. 2.6: Сборка и запуск файла

Скопируем файл in_out.asm в нашу папку (рис. 2.7)

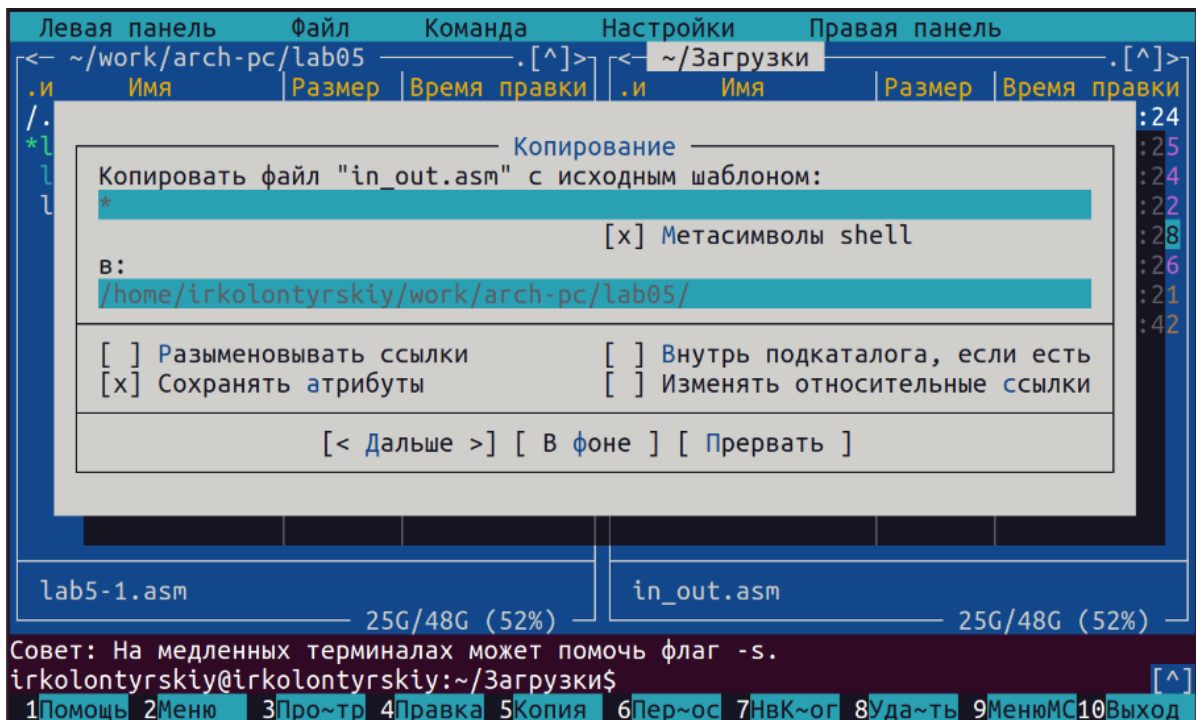


Рис. 2.7: Копирование файла in_out.asm

Создадим копию файла lab5-1.asm (рис. 2.8)

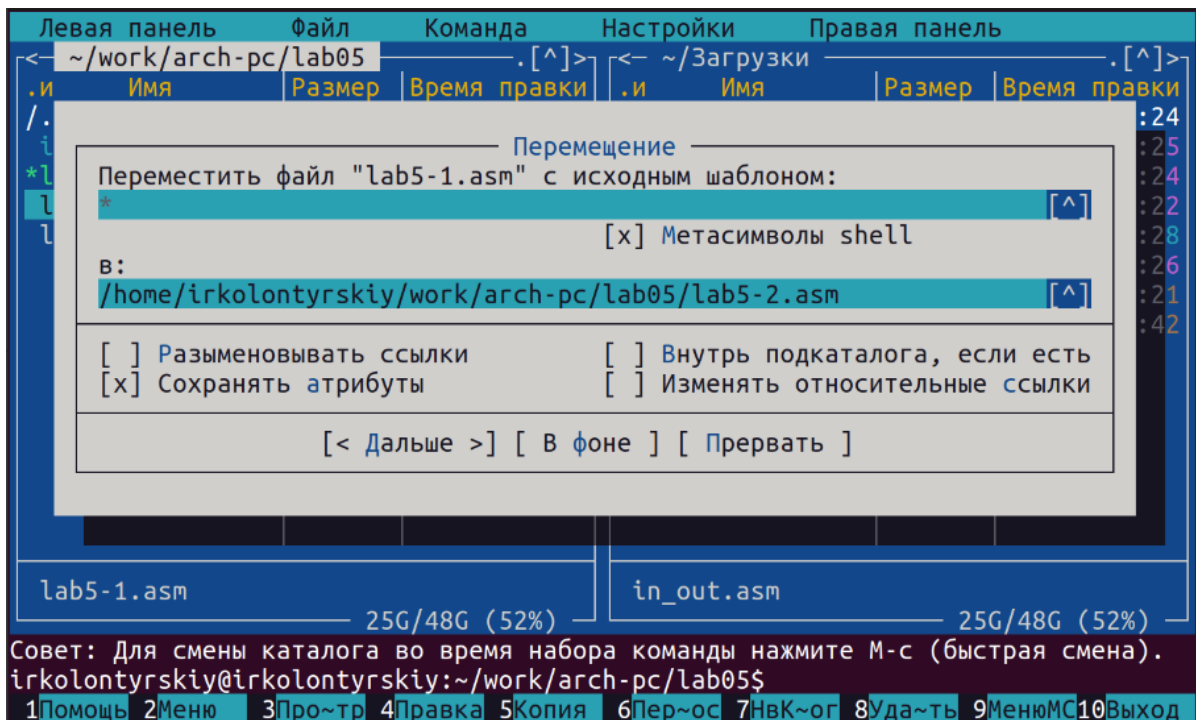


Рис. 2.8: Копирование файла

И вставим в него следующий код (рис. 2.9)

```
GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab05/lab5-2.asm *
;-----
; 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

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция
^X Выход        ^R ЧитФайл     ^\ Замена       ^U Вставить     ^J Выворнять    ^/_ К строке
```

Рис. 2.9: Вставка кода

Соберём программу и проверим её работу (рис. 2.10)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Колонтырский Илья Русланович
```

Рис. 2.10: Проверка и сборка программы

Заменим в коде программы sprintf на sprint (рис. 2.11)

```

;-----
; 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

```

Рис. 2.11: Редактирование программы

Соберём программу заново и запустим (рис. 2.12)

```

irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Колонтырский Илья Русланович

```

Рис. 2.12: Повторная сборка и запуск

Теперь мы вводим данные в той же строке, в которой они выводятся. Теперь сделаем копию первого файла и назовём её lab5-3.asm и изменим её так, чтобы выводилась введённая нами строка (рис. 2.13)

```

GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab05/lab5-3.asm *
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных 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 `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 'msg' в 'ecx'
mov edx,80 ; Размер строки 'msg' 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

```

Рис. 2.13: изменение скопированного файла

И запустим собранную программу для проверки (рис. 2.14)

```

irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Колонтырский Илья Русланович
Колонтырский Илья Русланович

```

Рис. 2.14: Проверка программы и её сборка

Теперь создадим копию второго файла (lab5-2.asm) и назовём её lab5-4.asm. Сделаем так, чтобы она также выводила введённую там информацию (рис. 2.15)

```

GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab05/lab5-4.asm *
;-----
; 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
mov eax, buf1
call sprintf
call quit ; вызов подпрограммы завершения

```

Рис. 2.15: Редактирование копии файла

Теперь запустим её, предварительно собрав (рис. 2.16)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ nasm -f elf lab5-4.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-4 lab
5-4.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab05$ ./lab5-4
Введите строку:
Колонтырский Илья Русланович
Колонтырский Илья Русланович
```

Рис. 2.16: Проверка работоспособности программы

3 Выводы

Был получен навык работы с тс, а также были получены навыки ввода и вывода информации в `asm`