

Лабораторная работа №6

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

Татьяна Алексеевна Коннова, НПИбд-01-22

Содержание

1	Цель работы	4
2	Задание	5
2.1	Выполнение лабораторной работы : Программа Hello world! . . .	5
3	Выводы	15

Список иллюстраций

2.1	Переход	5
2.2	lab6-2.asm	6
2.3	Текст с последующим выводом необходимого	6
2.4	Действия с in_out.asm	7
2.5	Копирование	7
2.6	Проверка	9
2.7	Результат	10
2.8	Вводимый и измененный текст	11
2.9	Проверка	11
2.10	Изменение файла	13
2.11	Проверка	14

1 Цель работы

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

2 Задание

Освоение инструкций языка ассемблера mov и int.

2.1 Выполнение лабораторной работы : Программа Hello world!

Перейдем в Midnight Commander.

(рис. 2.1)

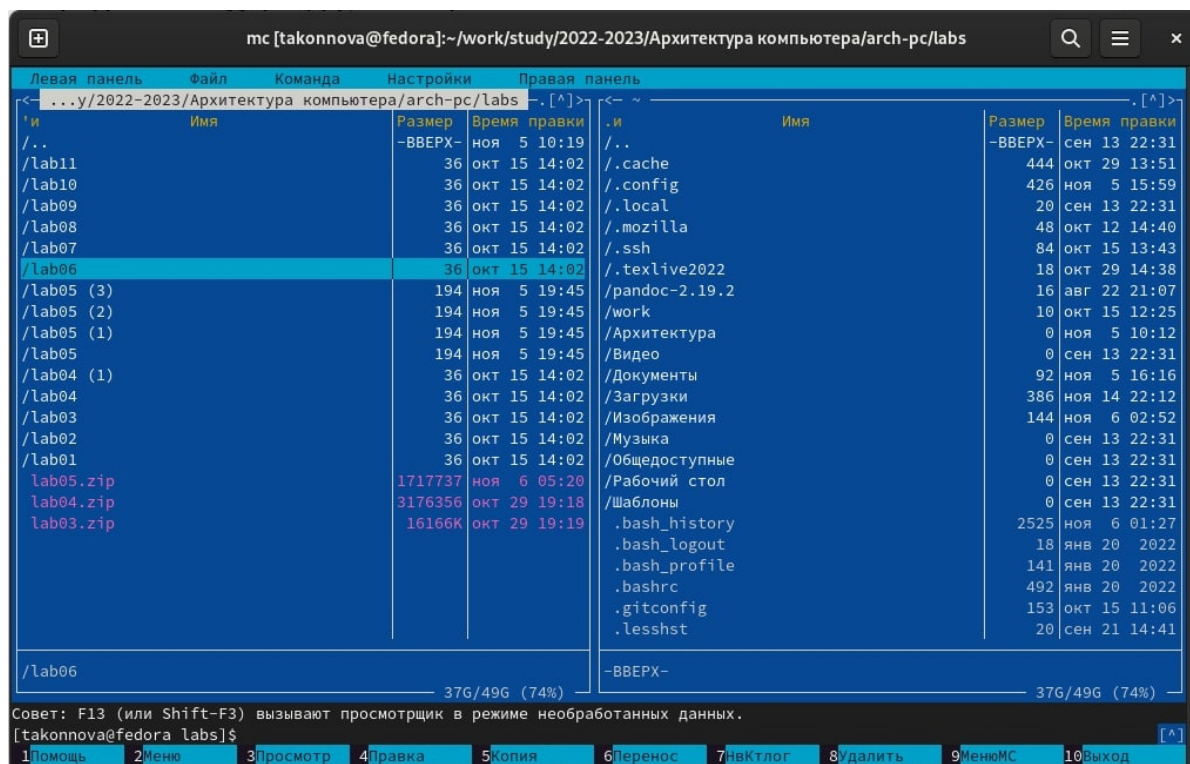


Рис. 2.1: Переход

Создадим с помощью команды touch файл lab6-2.asm (рис. 2.2)

Левая панель	Файл	Команда	Настройки	Правая п
< ...-2023/Архитектура компьютера/arch-pc/labs/lab06 >-.[^]>				
'и	Имя	Размер	Время правки	
/..		-ВВЕРХ-	ноя 6	05:20
/report		62	окт 15	14:02
/presentation		56	окт 15	14:02
lab6-1.asm		0	ноя 14	22:22

Рис. 2.2: lab6-2.asm

Оттранслируем текст программы lab6-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введем наше имя. (рис. 2.3)

```
takonnova@fedora:~/work/study/2022-
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (8/8), 50.30 КиБ | 1.93 МиБ/с, готово.
Всего 8 (изменений 7), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
To github.com:KONNOVAT/study_2022-2023_arh-pc.git
 f359a62..1c632eb master -> master
[takonnova@fedora lab05]$ mc
[takonnova@fedora lab06]$ touch lab6-1.asm
[takonnova@fedora lab06]$ nasm -f elf lab6-1.asm
lab6-1.asm:26: warning: label alone on a line without a colon might be in error [-w+label-orphan]
[takonnova@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[takonnova@fedora lab06]$ ./lab6-1
Введите строку:
Коннова Татьяна Алексеевна
[takonnova@fedora lab06]$
```

Рис. 2.3: Текст с последующим выводом необходимого

Скачаем 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. В одной из панелей mc откроем каталог с файлом lab6-1.asm. В другой панели каталог со скаченным файлом in_out.asm (для перемещения между панелями используем Tab). Скопируем файл in_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши (рис. 2.4)

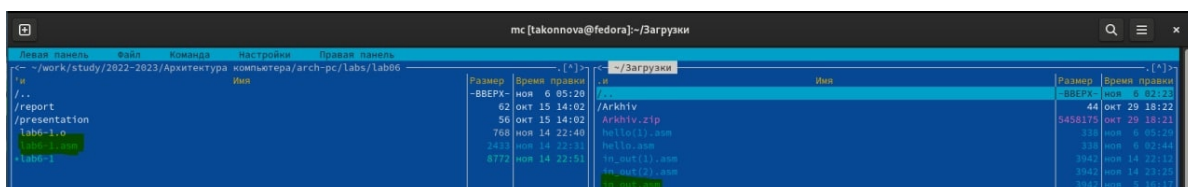


Рис. 2.4: Действия с in_out.asm

С помощью функциональной клавиши F6 создаем копию файла lab6- 1.asm с именем lab6-2.asm. Выделяем файл lab6-1.asm, нажимаем клавишу F6 , введем имя файла lab6-2.asm и нажмем клавишу Enter (рис. 2.5)

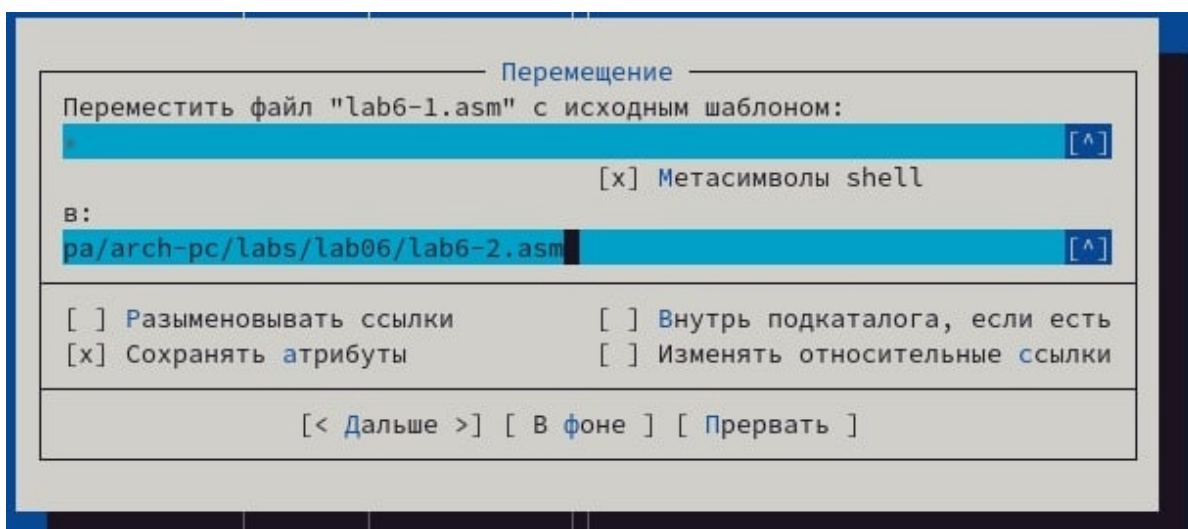


Рис. 2.5: Копирование

Исправим текст программы в файле lab6-2.asm с использованием под- программ из внешнего файла in_out.asm (используем подпрограммы sprintLF, sread и quit)

в соответствии с листингом 6.2. Создаем испол-
няемый файл и проверяем его
работу (рис. 2.6)


```
%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 sprintLF

    mov ecx, buf1
    mov edx, 80
    call sread
    mov eax, buf1
    call sprint
    call quit
```

Рис. 2.6: Проверка

Проверяем результат работы программы lab6-2.asm (рис. 2.7)

```
[takonnova@fedora lab06]$ nasm -f elf lab6-2.asm
[takonnova@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[takonnova@fedora lab06]$ ./lab6-2
Введите строку:
Коннова
```

Рис. 2.7: Результат

При включении LF в Sprint вводимые данные отображаются для ввода на следующей строке. ## Задание для самостоятельной работы

1. Создаем копию файла lab6-1.asm. Внесем 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.

Пусть копией lab6-1.asm будет являться lab6-3.asm (рис. 2.8)

```
mc [takonnova@fedora]~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab06
lab6-3.asm [----] 20 L:[ 19+11 30/ 40] *(2061/2452b) 0010 0x00A
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
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 `exit` -----
; После вызова инструкции 'int 80h' программа завершит работу

mov eax,4 ; Системный вызов для выхода (sys_exit)
mov ebx,1 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

mov eax,1
mov ebx,0
int 80h
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пере~тить 7Поиск
```

Рис. 2.8: Вводимый и измененный текст

2. Получаем исполняемый файл и проверяем его работу. На приглашение ввести строку введем свою фамилию. (рис. 2.9)

```
[takonnova@fedora lab06]$ nasm -f elf lab6-3.asm
[takonnova@fedora lab06]$ ld -m elf_i386 -o lab6-3 lab6-3.o
[takonnova@fedora lab06]$ ./lab6-3
Введите строку:
g
g
[takonnova@fedora lab06]$ ./lab6-3
Введите строку:
Коннова
Коннова
```

Рис. 2.9: Проверка

3. Создаем копию файла lab6-2.asm. Исправляем текст программы с исполь-

зование 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.

Подключаемый файл `in_out.asm` должен лежать в том же каталоге, что и файл с программой, в которой он используется. Пусть имя данного файла будет `lab6-4.asm` (рис. 2.10)

```
%include 'in_out.asm'
SECTION .data
msg:DB 'Введите строку: ',0h

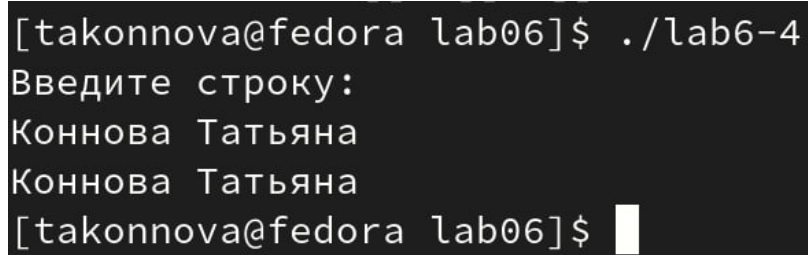
SECTION .bss
buf1: RESB 80

SECTION .text
    GLOBAL _start
    _start:
    ....
    mov eax, msg
    call sprint
    ....
    mov ecx, buf1
    mov edx, 80
    call sread
    mov eax, buf1
    call sprint
    call quit
```

Рис. 2.10: Изменение файла

4. Создаем исполняемый файл и проверяем его работу

(рис. 2.11)

A terminal window with a black background and white text. The prompt is [takonnova@fedora lab06]\$. The user enters ./lab6-4. The output is Введите строку: followed by two lines of Konnova Татьяна. The prompt returns to [takonnova@fedora lab06]\$.

```
[takonnova@fedora lab06]$ ./lab6-4
Введите строку:
Коннова Татьяна
Коннова Татьяна
[takonnova@fedora lab06]$
```

Рис. 2.11: Проверка

3 Выводы

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