

Лабораторная работа №5

Отчет

Зубов Иван Александрович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задание для самостоятельной работы	12
4	Выводы	15

Список иллюстраций

2.1	Вводим в консоль команду <code>mc</code>	6
2.2	Воспользуемся командой <code>touch</code>	7
2.3	Открываем файл функциональной клавишей, заполняем и сохраняем	8
2.4	Открываем файл и убеждаемся, что файл содержит текст программы	8
2.5	Проверяем, как работает данная программа	8
2.6	Скачиваем файл	9
2.7	Копируем скаченный файл	9
2.8	Создаем копию файла	10
2.9	Открываем и заполняем файл	10
2.10	Смотрим, как сработала программа	11
2.11	Редактируем файл	11
2.12	Смотрим, как сработала программа и сравниваем с прошлой	11
3.1	Создаем копию файла <code>lab5-1.asm</code>	12
3.2	Редактируем файл	12
3.3	Проверяем правильность написания программы	13
3.4	Создаем копию файла <code>lab5-2.asm</code>	13
3.5	Редактируем файл	13
3.6	Проверяем правильность написания программы	14

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Открываем Midnight Commander с помощью команды `mc`

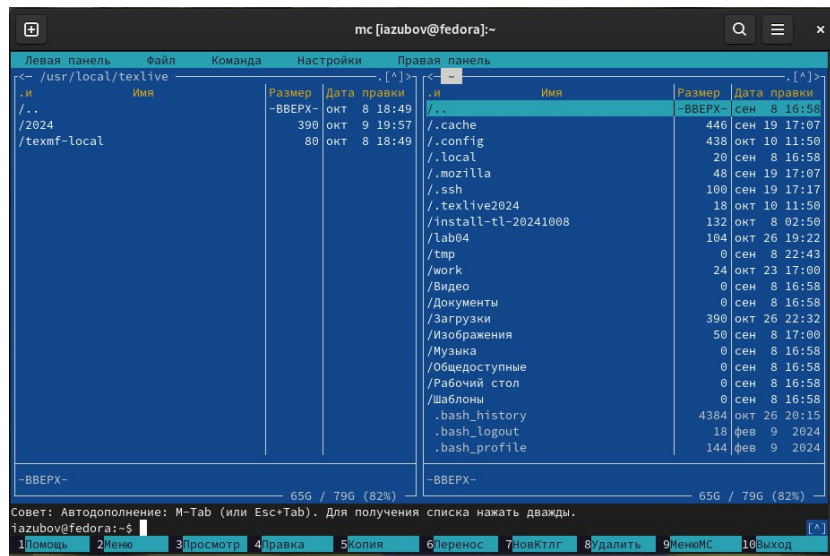
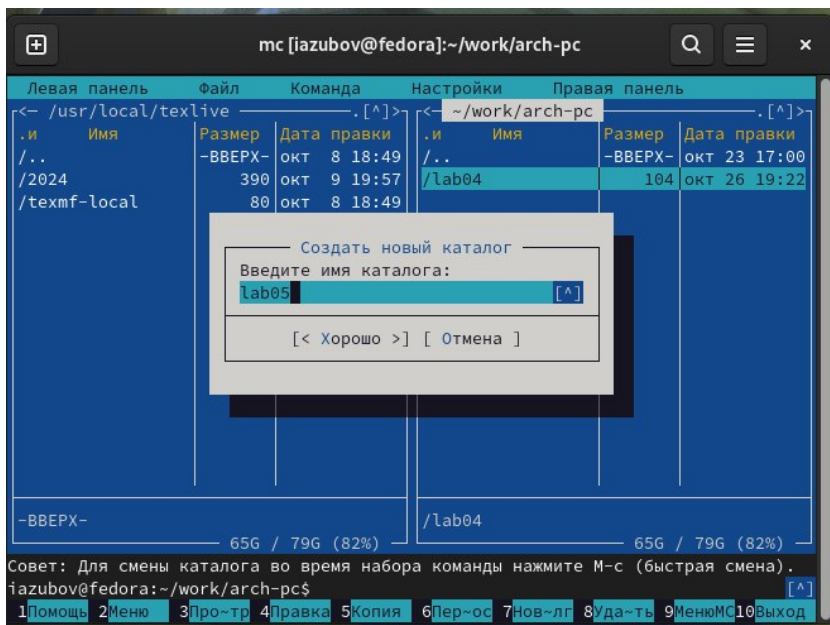


Рис. 2.1: Вводим в консоль команду `mc`

Переходим в каталог, созданный при выполнении 4 ЛБ и создаем каталог `lab05`



Создаем файл lab5-1.asm

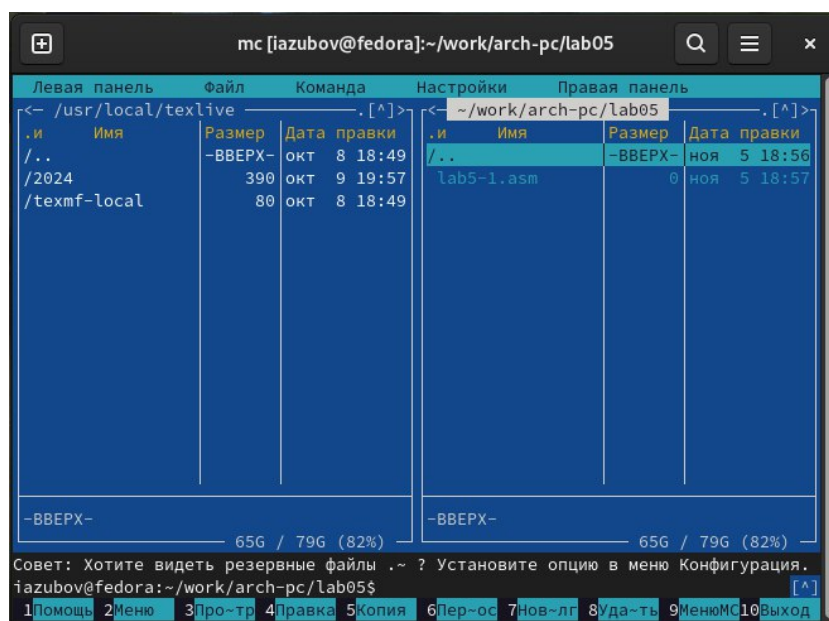


Рис. 2.2: Воспользуемся командой touch

Открываем файл для редактирования и заполняем его по листингу

```

mc [iazubov@fedora]:~/work/arch-pc/lab05
GNU nano 7.2 /home/iazubov/work/arch-pc/lab05/lab5-1.asm
----- Объявление переменных -----
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
^C Справка ^O Записать ^M Поиск ^K Вырезать ^T Выполнить ^C Позиция ^M-U Отмена
^X Выход ^R ЧитФайл ^N Замена ^U Вставить ^D Выводить ^J К строке ^M-E Повтор

```

Рис. 2.3: Открываем файл функциональной клавишей, заполняем и сохраняем

Открывем файл для просмотра

```

mc [iazubov@fedora]:~/work/arch-pc/lab05
GNU nano 7.2 /home/iazubov/work/arch-pc/lab05/lab5-1.asm
----- Объявление переменных -----
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
^C Справка ^O Записать ^M Поиск ^K Вырезать ^T Выполнить ^C Позиция ^M-U Отмена
^X Выход ^R ЧитФайл ^N Замена ^U Вставить ^D Выводить ^J К строке ^M-E Повтор
Прочитано 33 строки

```

Рис. 2.4: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем исполняемый файл

```

iazubov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
iazubov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
iazubov@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Зубов Иван Александрович
iazubov@fedora:~/work/arch-pc/lab05$

```

Рис. 2.5: Проверяем, как работает данная программа

Скачиваем файл со страницы курса

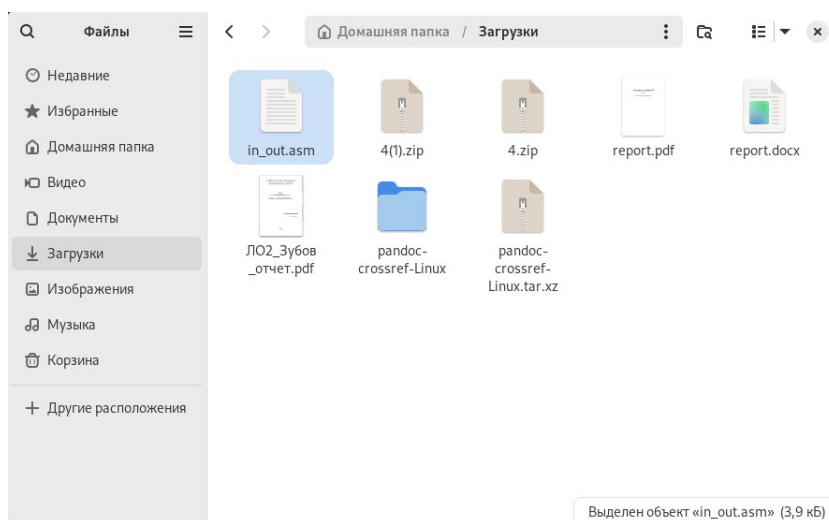


Рис. 2.6: Скачиваем файл

Копируем файл в нужную директорию

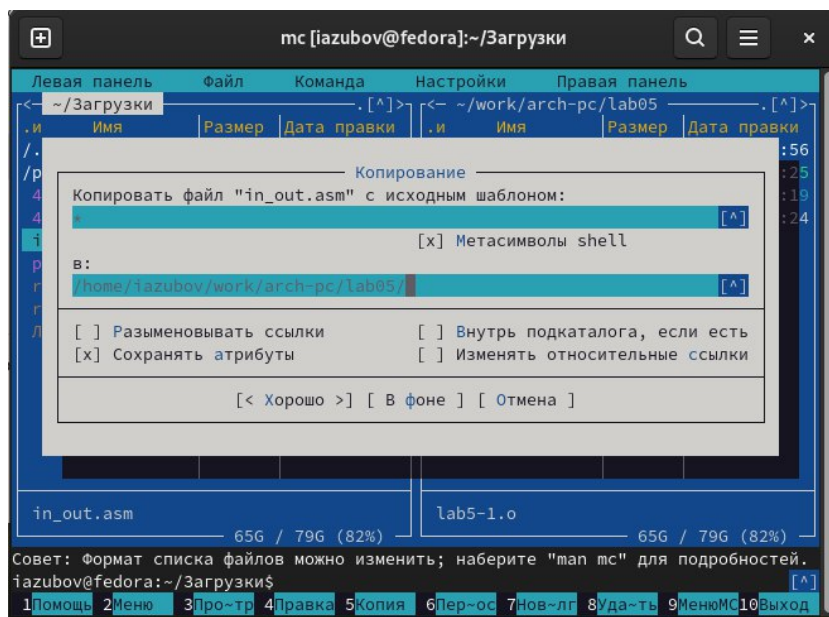


Рис. 2.7: Копируем скаченный файл

Создаем копию файла lab5-1.asm клавишей F6 и проверяем созданный файл



Рис. 2.8: Создаем копию файла

Открываем новый файл и заполняем его в соответствии с листингом

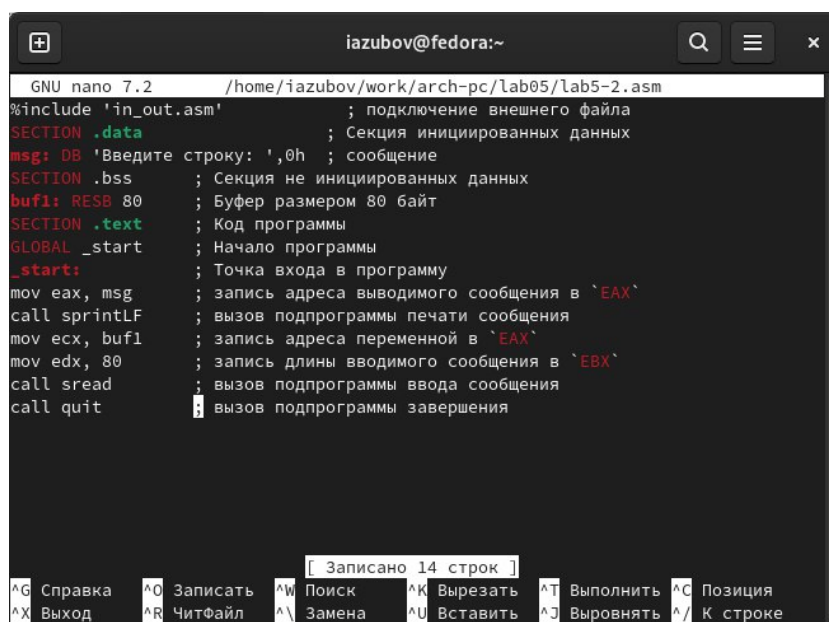


Рис. 2.9: Открываем и заполняем файл

Транслируем и запускаем новый файл

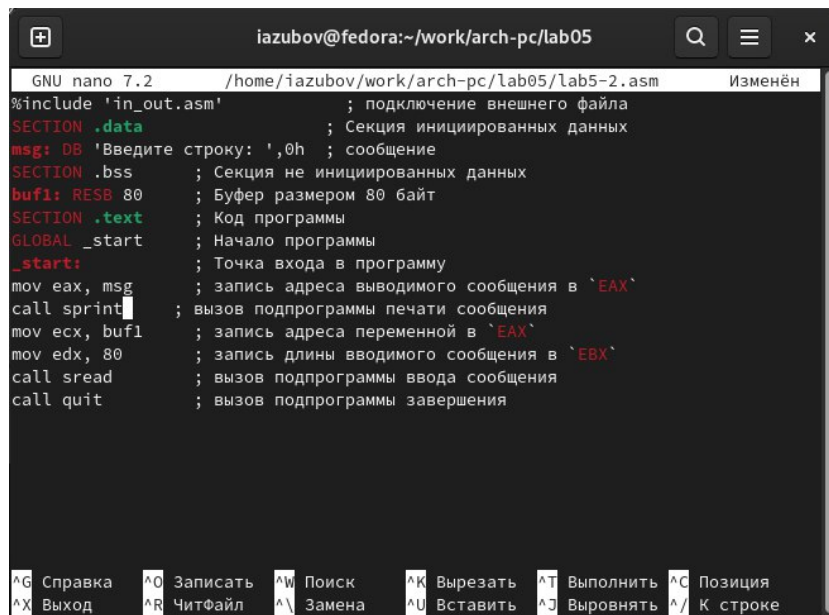
```

iazubov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
iazubov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
iazubov@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Зубов Иван Александрович
iazubov@fedora:~/work/arch-pc/lab05$

```

Рис. 2.10: Смотрим, как сработала программа

Снова открываем файл для редактирования и меняем `sprintLF` на `sprint`



```

GNU nano 7.2 /home/iazubov/work/arch-pc/lab05/lab5-2.asm
%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: Редактируем файл

Транслируем и запускаем файл

```

iazubov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
iazubov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
iazubov@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Зубов Иван Александрович
iazubov@fedora:~/work/arch-pc/lab05$

```

Рис. 2.12: Смотрим, как сработала программа и сравниваем с прошлой

Таким образом можем понять, что команда `sprint` выводит текст в той же строке, а `sprintLF` переносит на новую строку.

3 Задание для самостоятельной работы

Создаем копию файла lab5-1.asm и называем lab5-11

Левая панель				Правая панель			
~/work/arch-pc/lab05				~/work/arch-pc/lab05			
Имя	Размер	Дата	Правки	Имя	Размер	Дата	Правки
./	-ВВЕРХ-	ноя 5 18:56		./	-ВВЕРХ-	ноя 5 18:56	
in_out.asm	3942	ноя 5 19:49		in_out.asm	3942	ноя 5 19:49	
lab5-1.asm	2381	ноя 9 17:42		lab5-1.asm	2381	ноя 9 17:42	
lab5-1.o	752	ноя 9 17:43		lab5-1.o	752	ноя 9 17:43	
*lab5-11	8748	ноя 9 18:10		*lab5-11	8748	ноя 9 18:10	
lab5-11.asm	433	ноя 9 18:09		lab5-11.asm	433	ноя 9 18:09	
lab5-11.o	784	ноя 9 18:10		lab5-11.o	784	ноя 9 18:10	

Рис. 3.1: Создаем копию файла lab5-1.asm

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль

```
GNU nano 7.2 /home/iazubov/work/arch-pc/lab05/lab5-11.asm
;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 0x80

    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 0x80

    mov eax, 4
    mov ebx, 1
    mov ecx, buf1
    mov edx, 80
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рис. 3.2: Редактируем файл

Транслируем файл и запускаем программу

```
iazubov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
iazubov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-22 lab5-22.o
iazubov@fedora:~/work/arch-pc/lab05$ ./lab5-22
Введите строку:
Зубов Иван Александрович
Зубов Иван Александрович
```

Рис. 3.3: Проверяем правильность написания программы

Создаем копию файла lab5-2.asm и называем его lab5-22

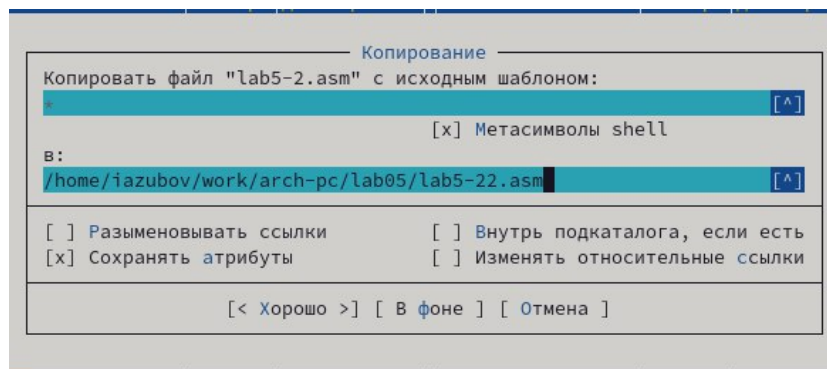


Рис. 3.4: Создаем копию файла lab5-2.asm

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль

```
mc [iazubov@fedora]:~/work/arch-pc/lab05
GNU nano 7.2 /home/iazubov/work/arch-pc/lab05/lab5-22.asm
#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. 3.5: Редактируем файл

Транслируем файл и запускаем программу

```
iazubov@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
iazubov@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-22 lab5-22.o
iazubov@fedora:~/work/arch-pc/lab05$ ./lab5-22
Введите строку: Зубов Иван Александрович
Зубов Иван Александрович
iazubov@fedora:~/work/arch-pc/lab05$
```

Рис. 3.6: Проверяем правильность написания программы

4 Выводы

Мы приобрели навыки работы с Midnight Commander и освоили инструкцию mov.