

Шаблон отчёта по лабораторной работе

5

Талебу тенке франк устонб НКАбд-05-23

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы :	6
2.1	Выводы по результатам выполнения заданий :	18
3	Задание для самостоятельной работы :	19
3.1	Создание программы без использования внешнего файла	19
3.1.1	создание программы с использованием внешнего файла .	19
4	Выводы	21
	Список литературы	22

Список иллюстраций

2.1	Ресунок 1	7
2.2	Ресунок 2	8
2.3	Ресунок 3	9
2.4	Ресунок 4	11
2.5	Ресунок 10	12
2.6	Ресунок 6	13
2.7	Ресунок 6	14
2.8	Ресунок 8	15
2.9	Ресунок 9	16
3.1	Ресунок 12	19
3.2	Ресунок 11	20

Список таблиц

1 Цель работы

- На шестой лабораторной работе мы научимся использовать “Midnight commander” и освоим инструкции **mov** и **int** языка ассемблера.

2 Выполнение лабораторной работы :

- На этом этапе мы запустили тс.(рис. [2.1])

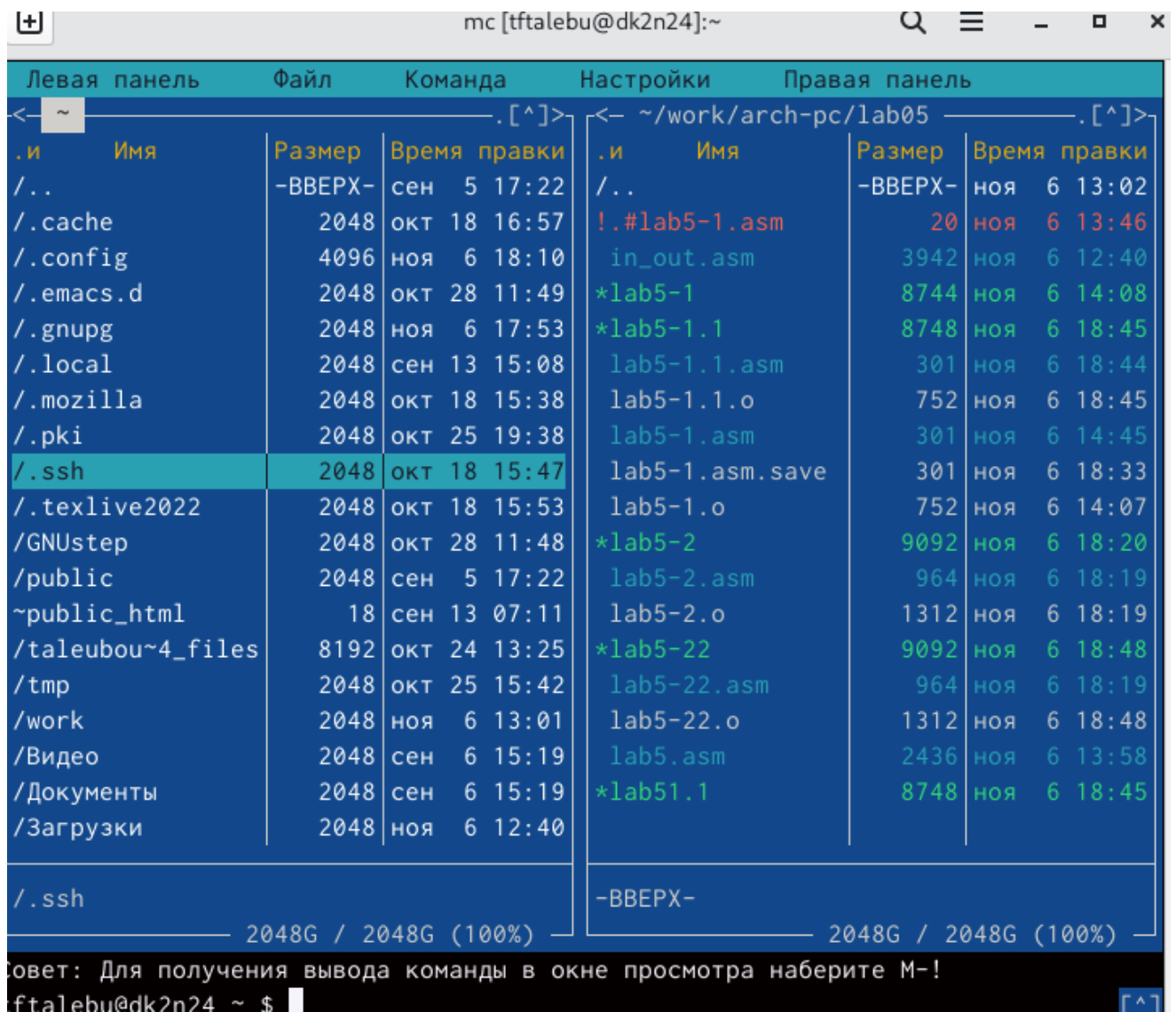


Рис. 2.1: Ресунок 1

- После этого мы переместились в каталог ~/work/arch-pc.(рис. [2.2])

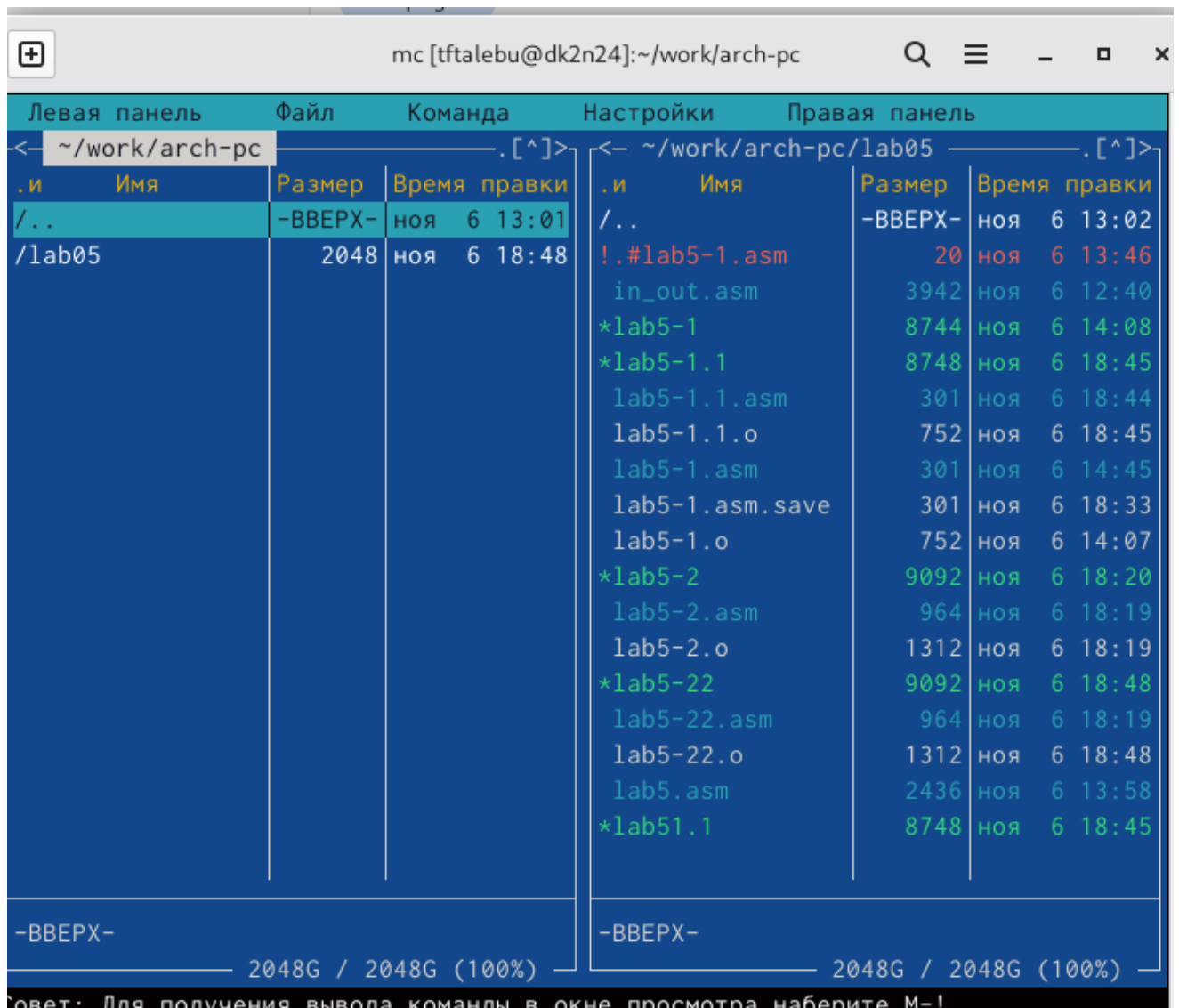


Рис. 2.2: Ресунок 2

- Мы скопировали текст программы из листинга 6.1 в файл `asm`, затем сохранили изменения и закрыли файл. (рис. [2.3])


```
mc [tftalebu@dk3n54]:~/work/arch-pc/lab05 x tftalebu@dk3n54:~/work/arch-pc/lab05
/afs/.dk.sci.pfu.edu.ru/home/t/f/tftalebu/work/arch-pc/lab05/lab5-1.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 80h

mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
```

[Прочитано 25 строк]

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	M-U Отмена
^X Выход	^R ЧитФайл	^N Замена	^U Вставить	^C Позиция	M-E Повтор

Рис. 2.3: Ресунок 3

- Используя функциональную клавишу F3, мы открыли файл lab6-1.asm для

просмотра. и мы проверили, что файл содержит текст программы. (рис. [2.4])

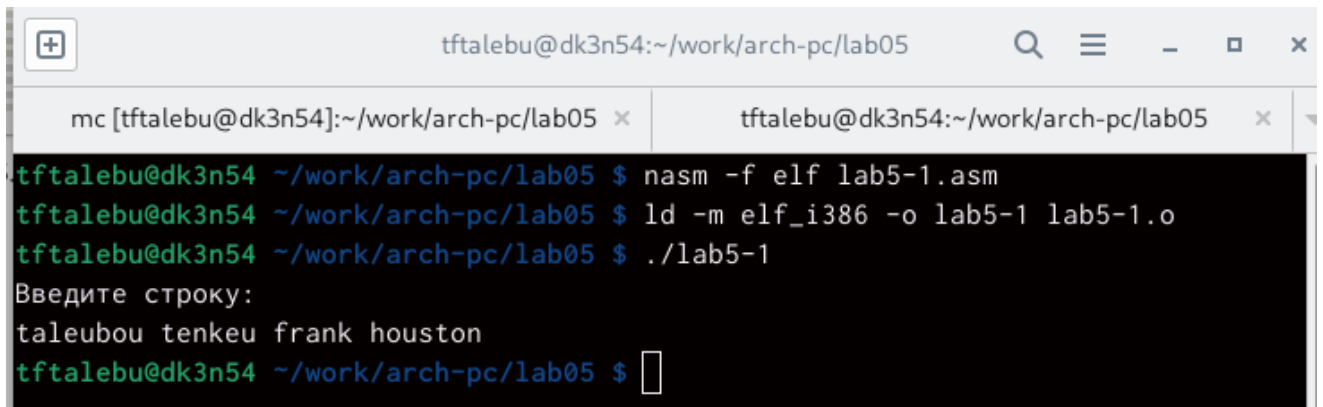
The image shows a terminal window titled "Места Терминал" with a tab labeled "mc [tftalebu@dk3n54]:~/work/arch-pc/lab05". The terminal displays the assembly code for a file named "lab5.asm". The code is written in MASM syntax and includes comments in Russian. It defines a data section with a message string and its length, a bss section for a buffer, and a text section for the program code. The code uses system calls (int 80h) for writing to the screen, reading from the keyboard, and exiting the program. The terminal interface includes a search bar, a menu icon, and a status bar at the bottom with navigation shortcuts.

```
Места Терминал
mc [tftalebu@dk3n54]:~/work/arch-pc/lab05
/afs/.dk.sci.pfu.edu.ru/home~/work/arch-pc/lab05/lab5.asm 2435/2435 100%
;-----
; 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
;----- 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

1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исханный 9Формат 10Выход
```

Рис. 2.4: Ресунок 4

- Затем мы перевели текст программы lab6-1.asm в объектный файл . Выполнил разметку объектного файла и запустил полученный исполняемый файл, где программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. По запросу, в этот момент мы ввели наше имя и фамилию.(рис. [2.5])



```
tftalebu@dk3n54:~/work/arch-pc/lab05
mc [tftalebu@dk3n54]:~/work/arch-pc/lab05 x tftalebu@dk3n54:~/work/arch-pc/lab05 x
tftalebu@dk3n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
tftalebu@dk3n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
tftalebu@dk3n54 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
taleubou tenkeu frank houston
tftalebu@dk3n54 ~/work/arch-pc/lab05 $
```

Рис. 2.5: Ресунок 10

- После этого мы загрузили файл in_out.asm из ТУИСА и с помощью mc мы смогли переместить файл в правильный каталог. (рис. [2.7])

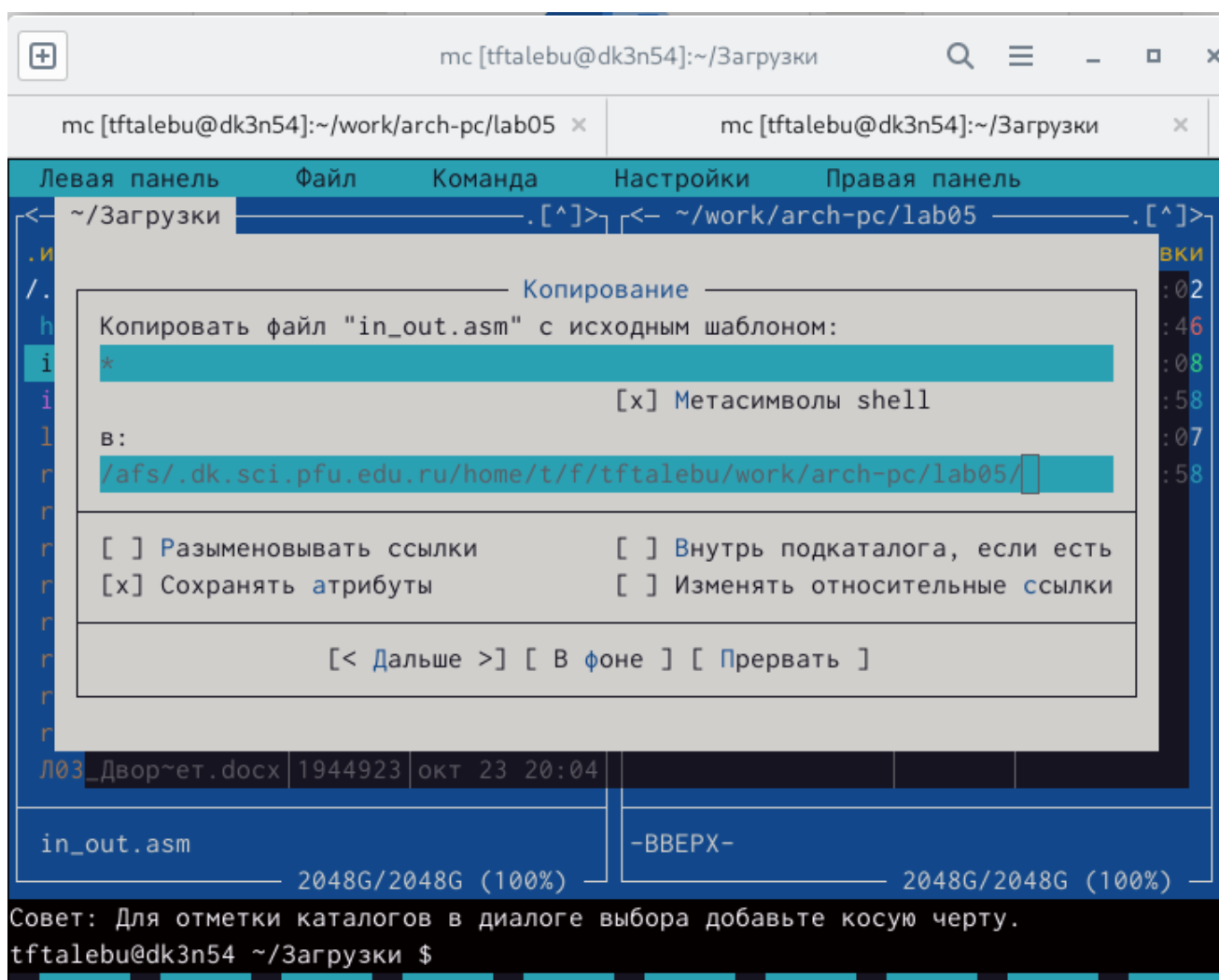


Рис. 2.6: Ресунок 6

- Используя функциональную клавишу f5, мы создали копию файла lab6-1.asm с именем lab6-2.asm.(рис. [2.7])

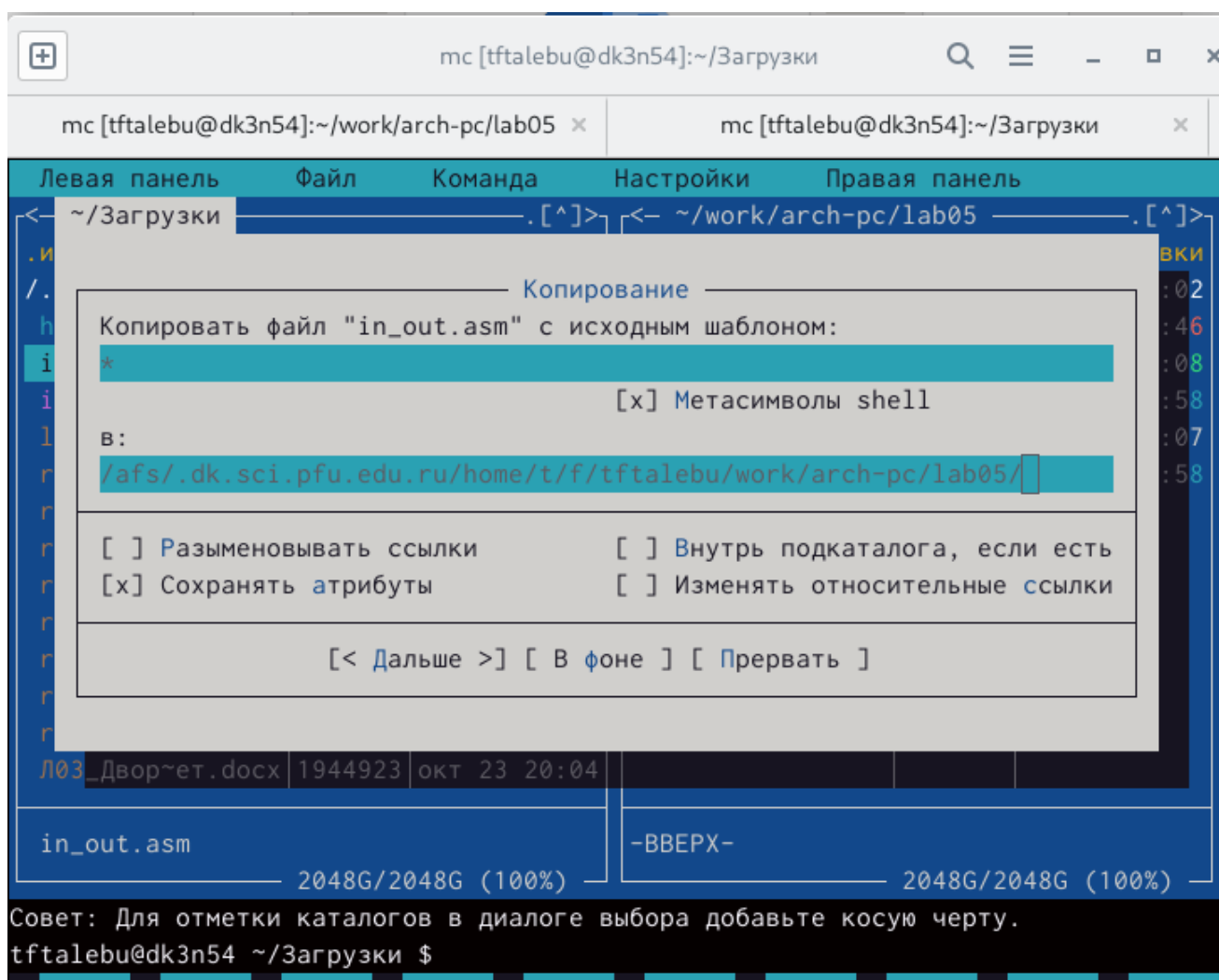


Рис. 2.7: Ресунок 6

- После этого мы исправляем текст программы в файле lab6-2.asm, используя подпрограммы из внешнего файла in_out.asm.(рис. [2.8])

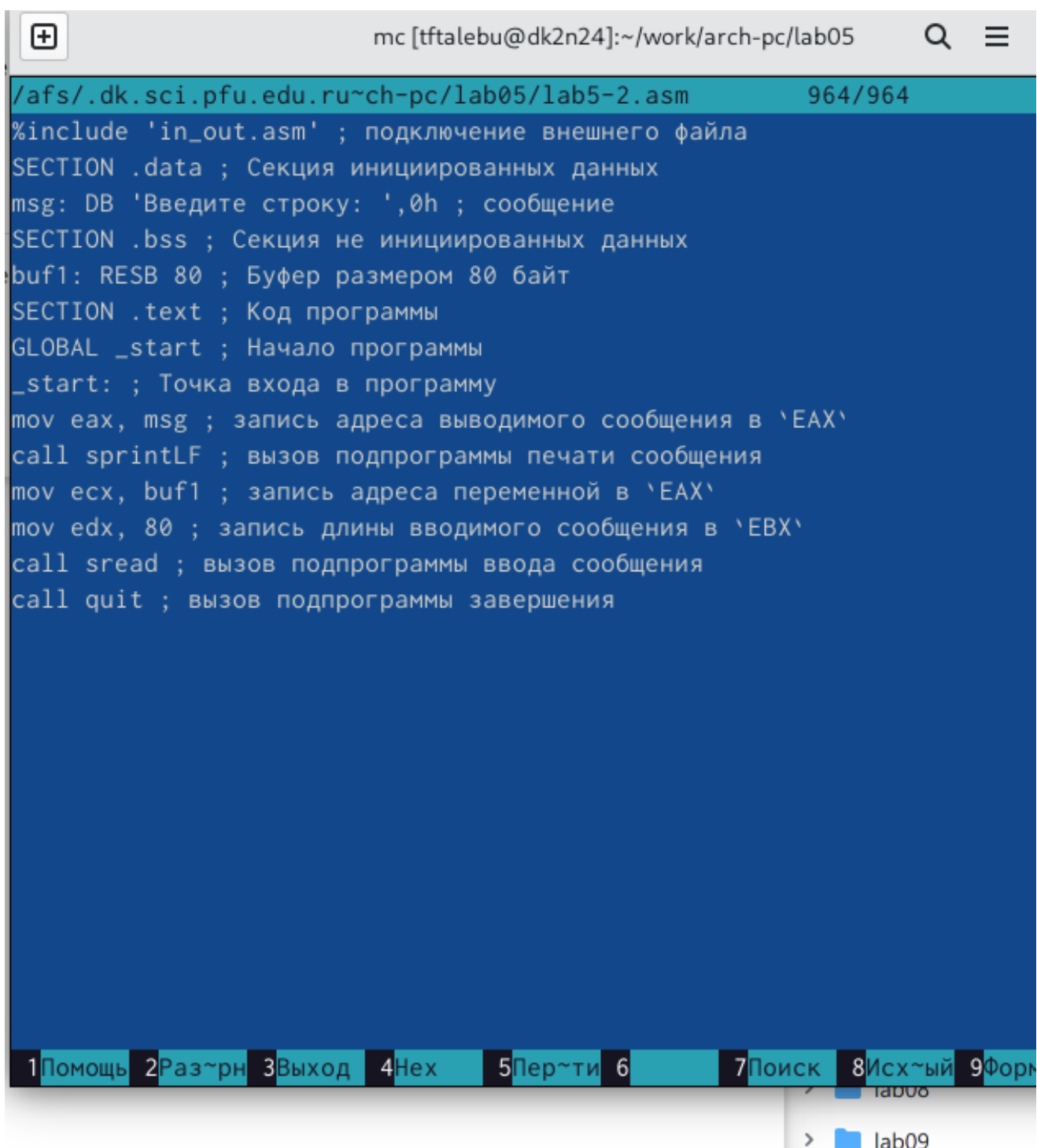
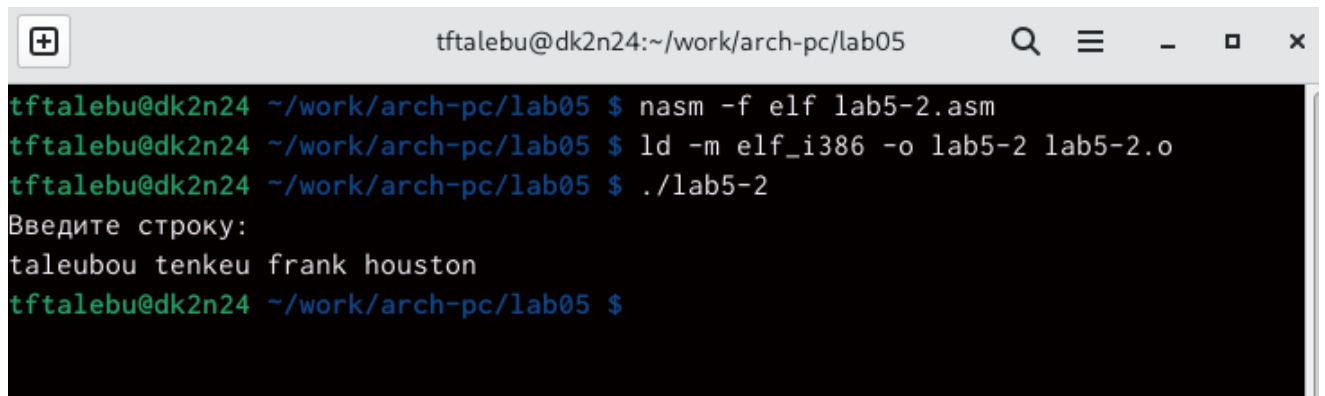


Рис. 2.8: Ресунок 8

- Затем мы перевели текст программы lab6-2.asm в объектный файл . Вы-

полнил разметку объектного файла и запустил полученный исполняемый файл.(рис. [2.9])



```
tftalebu@dk2n24:~/work/arch-pc/lab05
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
taleubou tenkeu frank houston
tftalebu@dk2n24 ~/work/arch-pc/lab05 $
```

Рис. 2.9: Ресунок 9

- На этом шаге мы меняем функцию sprintLF на функцию sprint Создал исполняемый файл, и разница заключалась в том, что эта функция изменяет входные данные на новую строку.(рис. [??])(рис. [??])


```
mc [tftalebu@dk3n54]:~/work/arch-pc/lab05
/afs/.dk.sci.pfu.edu.ru/home/t/f/tftalebu/work/arch-pc/lab05/lab5-1.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 80h

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

[ Прочитано 25 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить M-U Отмена
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция M-E Повтор

tftalebu@dk2n24:~/work/arch-pc/lab05
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
taleubou tenkeu frank houston
tftalebu@dk2n24 ~/work/arch-pc/lab05 $
```

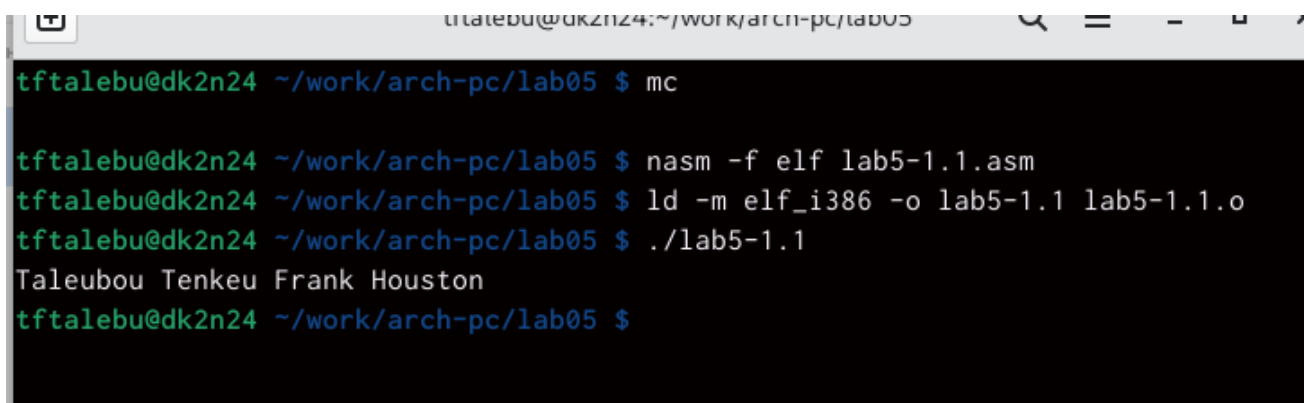
2.1 Выводы по результатам выполнения заданий :

- В ходе лабораторных работ мы узнали, как использовать `midnight commander`, и мы овладели навыками использования инструмента `nasm`.

3 Задание для самостоятельной работы :

3.1 Создание программы без использования внешнего файла

- В этой части мы должны были сделать копию файла lab6-1.asm, а затем мы должны были создать программу, которая запрашивает ввод строки, затем позволяет выполнить ввод с клавиатуры и, наконец, отобразить введенную строку, но без использования внешнего файла in_out.asm.(рис. [3.1]) |



```
tftalebu@dk2n24: ~/work/arch-pc/lab05
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ mc
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.1.asm
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1.1 lab5-1.1.o
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ./lab5-1.1
Taleubou Tenkeu Frank Houston
tftalebu@dk2n24 ~/work/arch-pc/lab05 $
```

Рис. 3.1: Ресунок 12

3.1.1 создание программы с использованием внешнего файла

- в этой части мы попытались выполнить ту же программу, но с использованием внешнего файла.(рис. [3.2])

```
nasm: fatal: unable to open input file 'lab5-22.asm': No such file or directory
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ mc

tftalebu@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5-22.asm
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-22 lab5-22.o
tftalebu@dk2n24 ~/work/arch-pc/lab05 $ ./lab5-22
Введите строку:
Taleubou Tenkeu Frank Houston
tftalebu@dk2n24 ~/work/arch-pc/lab05 $
```

Рис. 3.2: Ресунок 11

3.1.1.1 Выводы по результатам выполнения заданий :

- В этой части мы узнали, как создавать и редактировать программы с помощью подпрограмм и как управлять с помощью языка ассемблера.

Более подробно об Unix см. в [1–6].

4 Выводы

- На шестой лабораторной работе мы научимся использовать “Midnight commander” и освоим инструкции mov и int языка ассемблера и мы узнали, как создавать и редактировать программы с помощью подпрограмм и как управлять с помощью языка ассемблера.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.