

Отчёт по лабораторной работе №5

Дисциплина: Архитектура компьютера

Студент: Верниковская Екатерина Андреевна

Содержание

Цель работы	6
Задание	7
Выполнение лабораторной работы	8
Изучение Midnight Commander	8
Подключение внешнего файла in_out.asm	20
Задание для самостоятельной работы	25
Выводы	32

Список таблиц

Список иллюстраций

1	Открытие Midnight Commander	8
2	Открытый Midnight Commander	9
3	Переход в нужный каталог	10
4	Создание папки	10
5	Проверка	11
6	Переход в созданную папку	12
7	Создание файла	12
8	Проверка	13
9	Открытие файла	13
10	Ввод текста программы	14
11	Сохранение файла	15
12	Проверка	15
13	Транслирование	15
14	Проверка 1	16
15	Компиляция	16
16	Проверка 2	17
17	Компоновка	17
18	Проверка 3	18
19	Компоновка 2	18
20	Проверка 4	19
21	Запуск файла	19
22	Ввод с клавиатуры	19
23	Скачивание файла	20
24	Открытие двух панелей	20
25	Копирование файла в нужный каталог	21
26	Проверка	21
27	Создание копии файла с другим именем	21
28	Проверка	22
29	Исправленная программа	23
30	Транслирование	23
31	Проверка	23
32	Компоновка	24
33	Запуск программы	24
34	Исправление программы	25
35	Проверка	25
36	Создание копии файла	26
37	Проверка	26

38	Текст программы	28
39	Проверка	28
40	Создание копии файла	29
41	Проверка	29
42	Текст программы	31
43	Проверка	31

Цель работы

Приобрести практические навыки работы в Midnight Commander и освоить инструкции языка ассемблера `mov` и `int`.

Задание

1. Открыть Midnight Commander.
2. Создать папку lab05, где будут храниться файлы лабораторной работы №5.
3. Создать файл «lab5-1.asm», ввести текст программы. Оттранслировать текст программы, выполнить компоновку объектного файла и запустить получившийся исполняемый файл.
4. Скачать с ТУИС файл «in_out.asm» и переместить его в каталог lab05.
5. Скопировать файл «lab5-1.asm» с именем «lab5-2.asm» и исправить текст программы так чтобы использовались программы из внешнего файла «in_out.asm».
6. Создать исполняемый файл и проверить его работу.
7. Создать копию файла «lab5-1.asm». Внести изменения в программу (без использования внешнего файла «in_out.asm»), так чтобы она работала по определённому алгоритму.
8. Создать копию файла «lab5-2.asm». Также исправить текст программы, но уже с использованием подпрограмм из внешнего файла «in_out.asm», так чтобы она работала по определённому алгоритму.
9. Создать исполняемые файлы и проверить их работу.

Выполнение лабораторной работы

Изучение Midnight Commander

Открываем Midnight Commander с помощью команды 'mc' (рис. [-@fig:001]), (рис. [-@fig:002])



Рис. 1: Открытие Midnight Commander

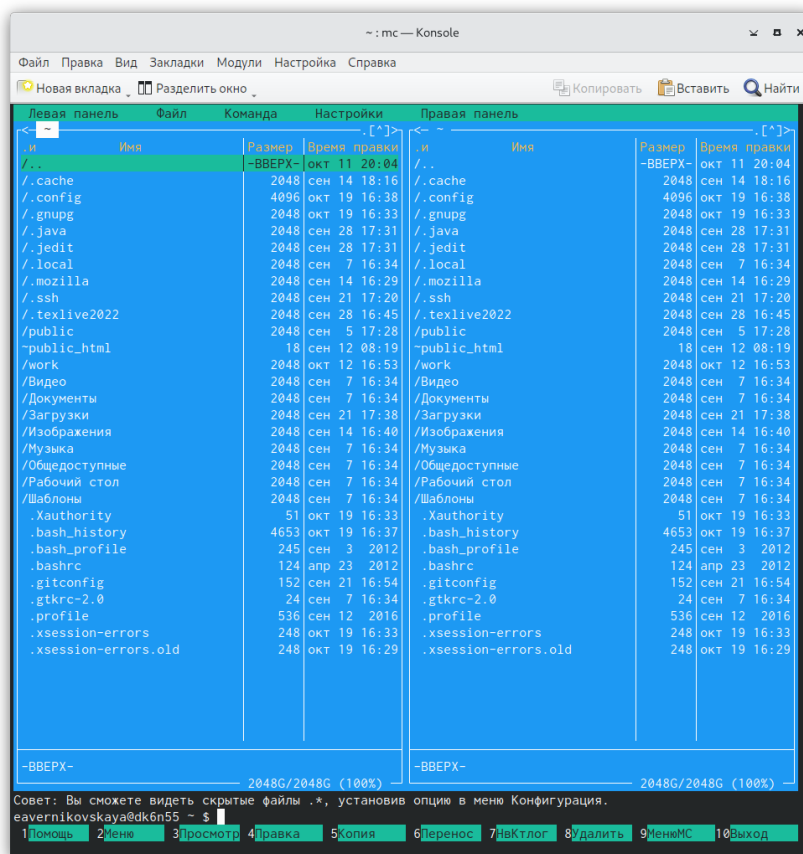


Рис. 2: Открытый Midnight Commander

Пользуясь клавишами на клавиатуре ‘вверх’, ‘вниз’ и ‘enter’ переходим в каталог «~/work/arch-pc» (рис. [-@fig:003])

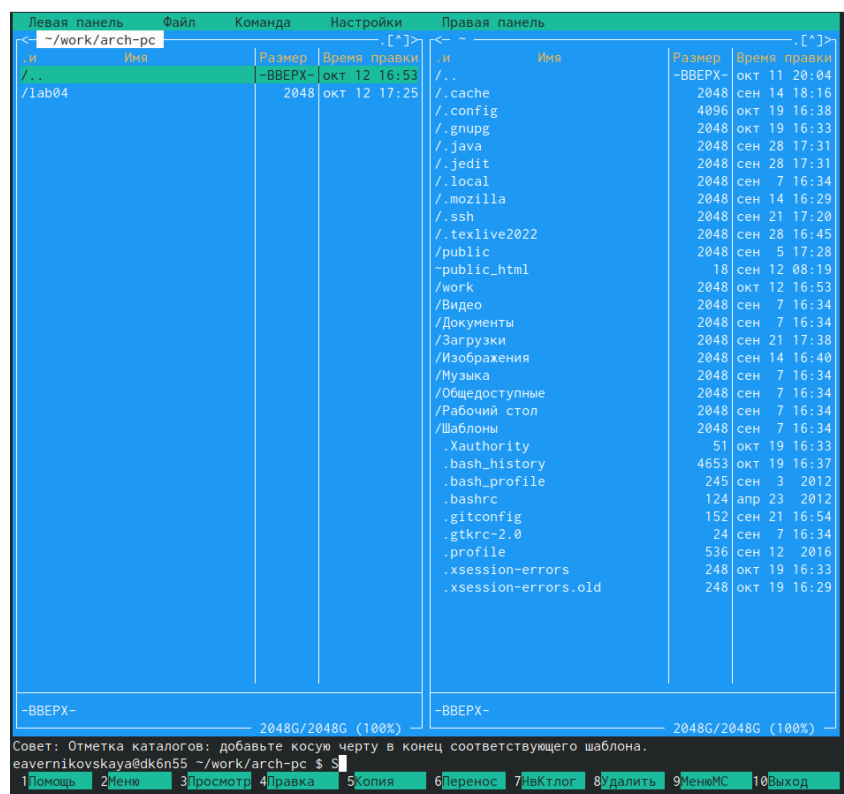


Рис. 3: Переход в нужный каталог

С помощью функциональной клавиши 'F7' создаём папку «lab05» и переходим в неё рис. [-@fig:004]), (рис. [-@fig:005]), рис. [-@fig:006])

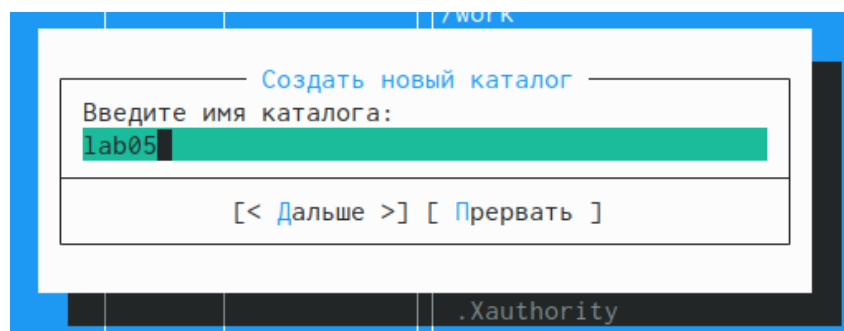


Рис. 4: Создание папки

Левая панель				Правая панель			
Файл		Команда		Настройки		Правая панель	
Имя	Размер	Время	Правки	Имя	Размер	Время	Правки
~/work/arch-pc	-BBERX-	окт 12 16:53		..	-BBERX-	окт 11 20:04	
/lab04	2048	окт 12 17:25		./cache	2048	сен 14 18:16	
/lab05	2048	окт 19 16:43		./config	4096	окт 19 16:38	
				./gnupg	2048	окт 19 16:33	
				./java	2048	сен 28 17:31	
				./jedit	2048	сен 28 17:31	
				./local	2048	сен 7 16:34	
				./mozilla	2048	сен 14 16:29	
				./ssh	2048	сен 21 17:20	
				./texlive2022	2048	сен 28 16:45	
				/public	2048	сен 5 17:28	
				~public_html	18	сен 12 08:19	
				/work	2048	окт 12 16:53	
				/Видео	2048	сен 7 16:34	
				/Документы	2048	сен 7 16:34	
				/Загрузки	2048	сен 21 17:38	
				/Изображения	2048	сен 14 16:40	
				/Музыка	2048	сен 7 16:34	
				/Общедоступные	2048	сен 7 16:34	
				/Рабочий стол	2048	сен 7 16:34	
				/Шаблоны	2048	сен 7 16:34	
				.Xauthority	51	окт 19 16:33	
				.bash_history	4653	окт 19 16:37	
				.bash_profile	245	сен 3 20:12	
				.bashrc	124	апр 23 20:12	
				.gitconfig	152	сен 21 16:54	
				.gitkr-2.0	24	сен 7 16:34	
				.profile	536	сен 12 20:16	
				.xsession-errors	248	окт 19 16:33	
				.xsession-errors.old	248	окт 19 16:29	
/lab05 2048G/2048G (100%)				-BBERX- 2048G/2048G (100%)			

Совет: F13 (или Shift-F3) вызывают просмотрщик в режиме необработанных данных.

eaavnikovskaya@dk6n55 ~/work/arch-pc \$

1 Помощь 2 Меню 3 Просмотр 4 Правка 5 Копия 6 Перенос 7 ВКтлог 8 Удалить 9 МенюМС 10 Выход

Рис. 5: Проверка

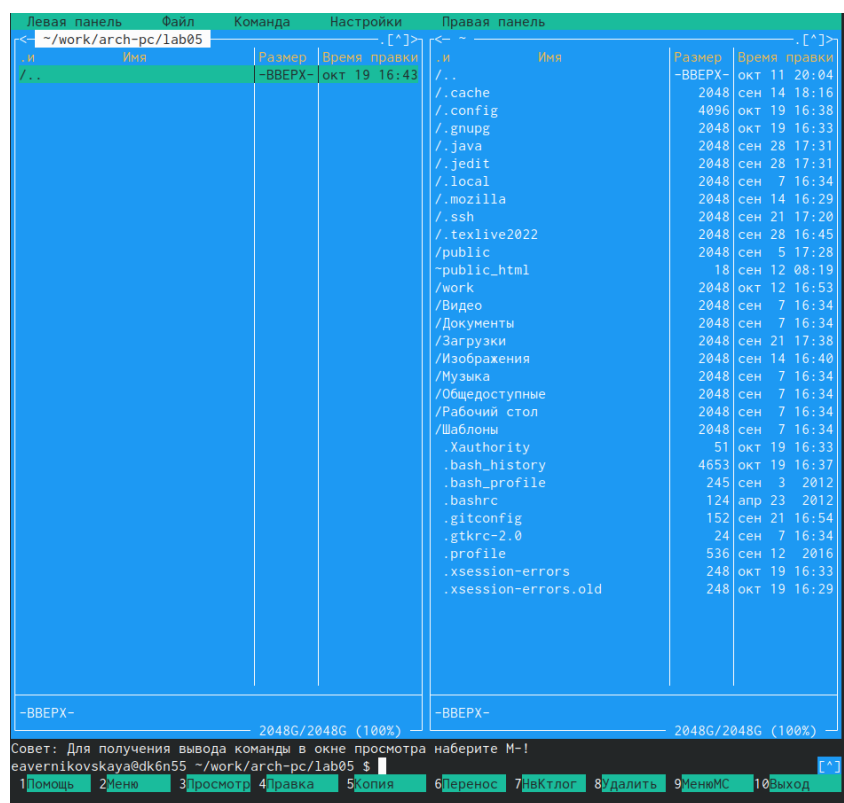


Рис. 6: Переход в созданную папку

Пользуясь строкой ввода и командой 'touch' создаём файл «lab5-1.asm» рис. [-@fig:007]), (рис. [-@fig:008])

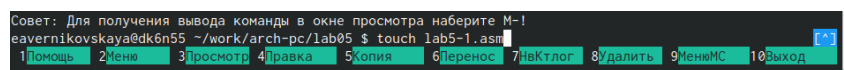


Рис. 7: Создание файла

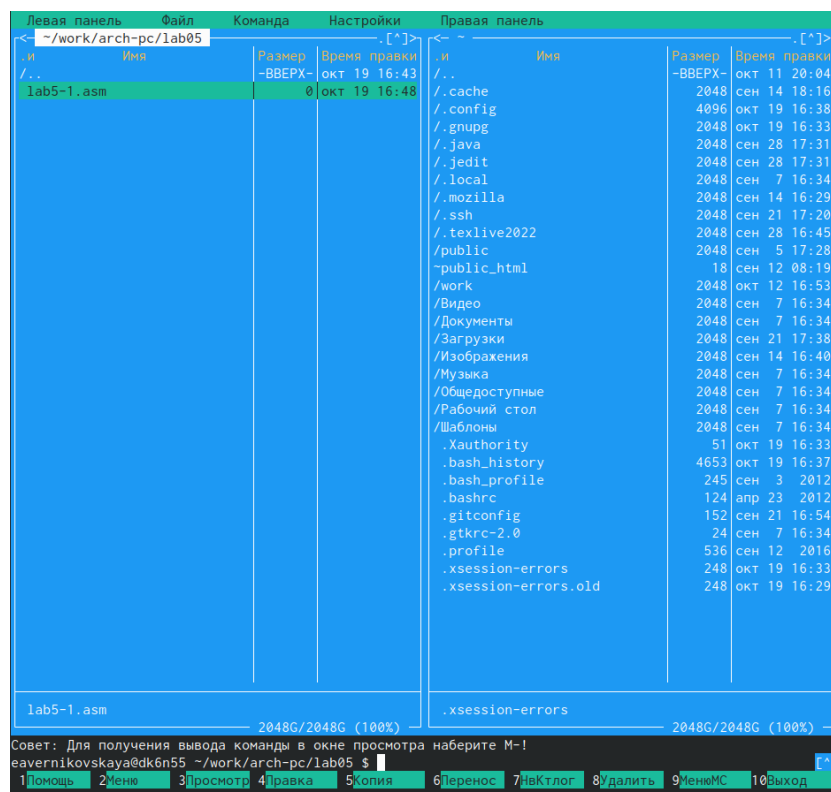


Рис. 8: Проверка

С помощью функциональной клавиши 'F4' открываем файл «lab5-1.asm» для редактирования во встроенном редакторе. В нашем случае это mcedit (рис. [-@fig:009])

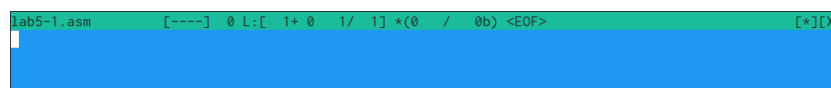


Рис. 9: Открытие файла

Вводим текст программы (рис. [-@fig:010])

Текст программы:

```
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
```

```

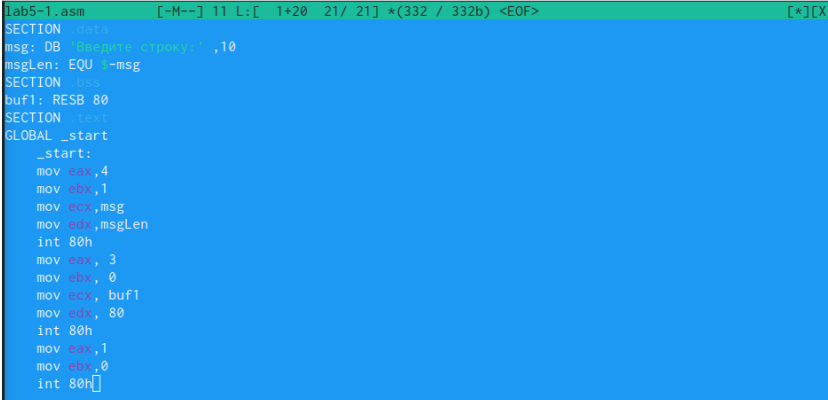
buf1: RESB 80
SECTION .text
GLOBAL _start

_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,1
    mov ebx,0
    int 80h

```



```

lab5-1.asm  [-M--] 11 L: [ 1+20 21/ 21] *(332 / 332b) <EOF>  [*][X]
SECTION .data
msg: DB "Program: myprog",10
msgLen: EQU $-msg
SECTION .text
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,1
    mov ebx,0
    int 80h

```

Рис. 10: Ввод текста программы

С помощью функциональной клавиши ‘F2’ сохраняем изменения и выходим с помощью ‘F10’ (рис. [-@fig:011])

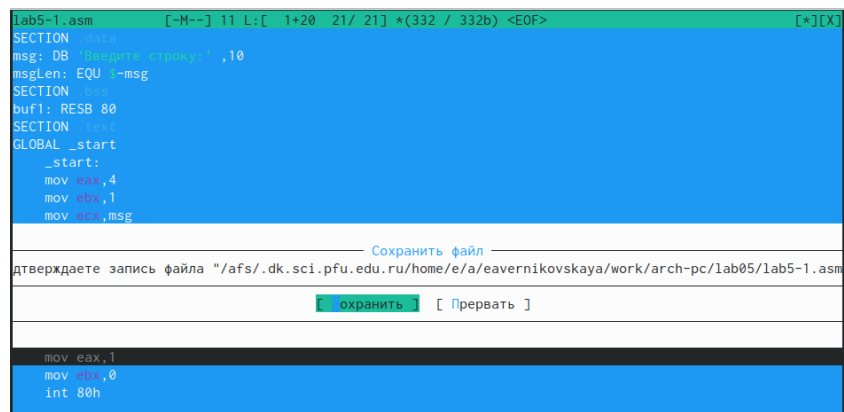


Рис. 11: Сохранение файла

С помощью функциональной клавиши 'F3' открываем изменённый файл, чтобы убедиться, что он содержит текст программы (рис. [-@fig:012])

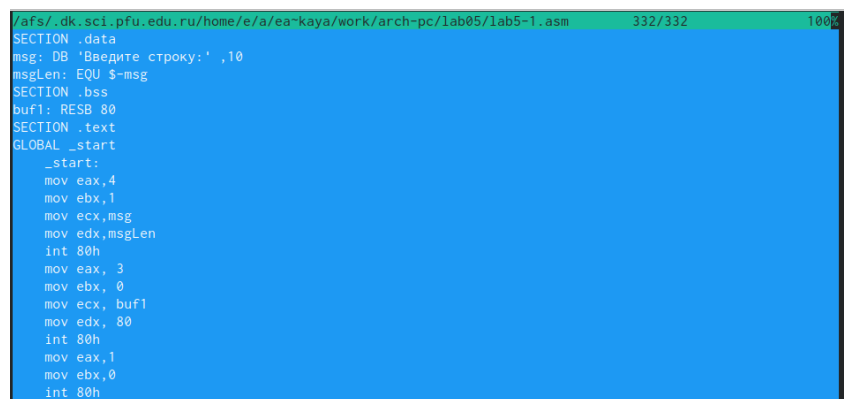


Рис. 12: Проверка

Транслируем текст программы в объектный файл, компилируем исходный файл, передаём объектный файл на обработку компоновщику (рис. [-@fig:013]), (рис. [-@fig:014]), (рис. [-@fig:015]), (рис. [-@fig:016]), (рис. [-@fig:017]), (рис. [-@fig:018]), (рис. [-@fig:019]), (рис. [-@fig:020])

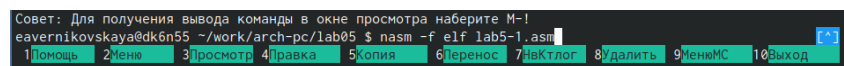


Рис. 13: Транслирование

Рис. 18: Проверка 3

Совет: Для получения вывода команды в окне просмотра наберите M!

```
eavernikovskaya@dk6n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 lab5-1.o -o main5
```

1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НаКтолог 8Удалить 9МенюМС 10Выход

Рис. 19: Компоновка 2

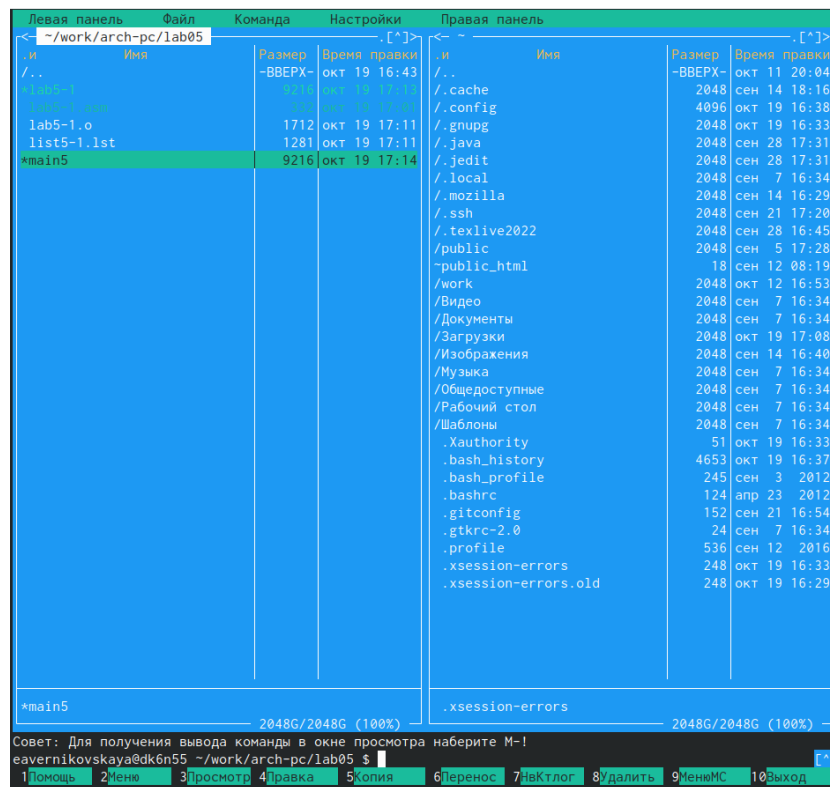


Рис. 20: Проверка 4

Запускаем получившийся исполняемый файл (рис. [-@fig:021])

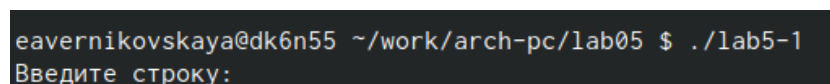


Рис. 21: Запуск файла

На запрос вводим наше ФИО (рис. [-@fig:022])

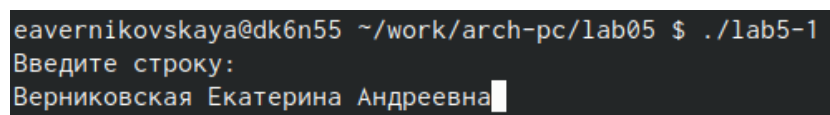


Рис. 22: Ввод с клавиатуры

Подключение внешнего файла in_out.asm

Скачиваем файл «in_out.asm» с ТУИС (рис. [-@fig:023])

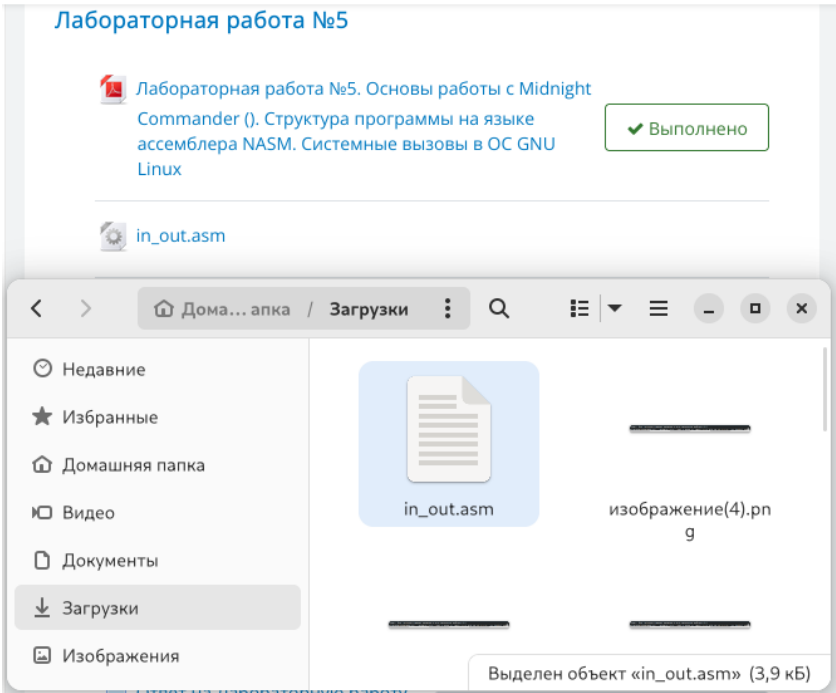


Рис. 23: Скачивание файла

В одной из панелей mc открываем каталог с файлом «lab5-1.asm», в другой - каталог со скачанным файлом «in_out.asm» (рис. [-@fig:024])

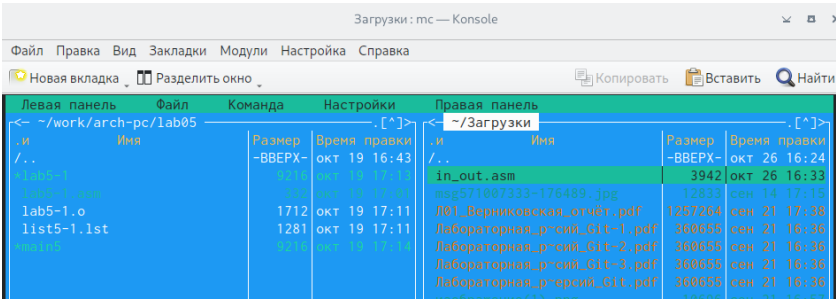


Рис. 24: Открытие двух панелей

С помощью функциональной клавиши ‘F5’ копируем файл «in_out.asm» в каталог с файлом «lab5-1.asm» (рис. [-@fig:025]), (рис. [-@fig:026])

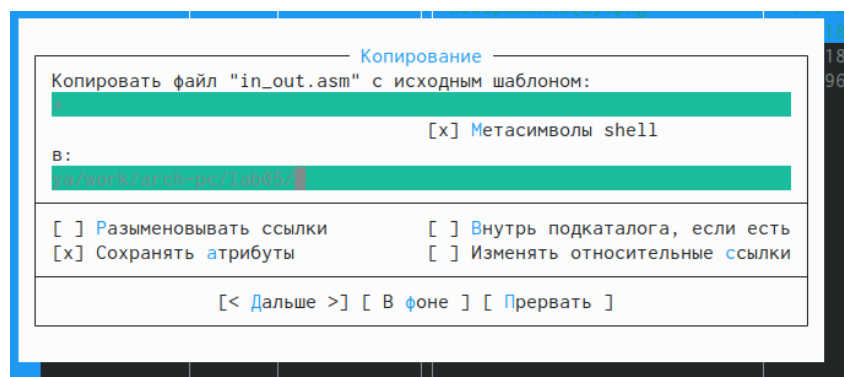


Рис. 25: Копирование файла в нужный каталог

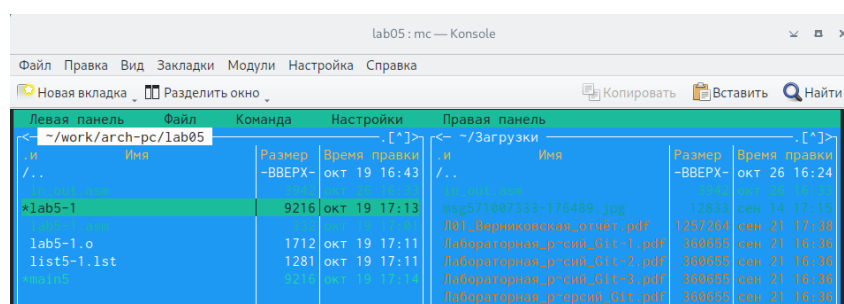


Рис. 26: Проверка

С помощью той же функциональной клавиши 'F5' создаём копию файла «lab5-1.asm» с именем «lab5-2.asm» (рис. [-@fig:027]), (рис. [-@fig:028])

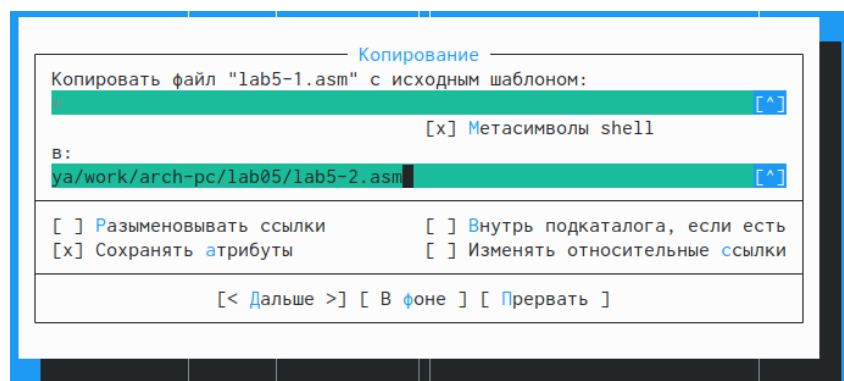


Рис. 27: Создание копии файла с другим именем

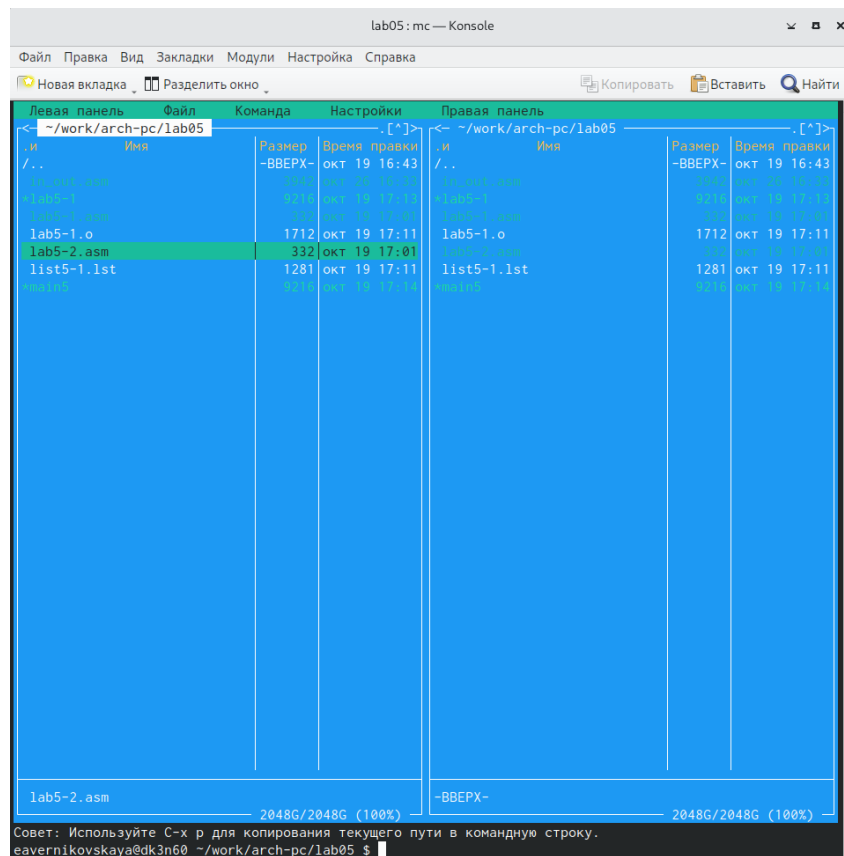


Рис. 28: Проверка

С помощью функциональной клавиши 'F4' открываем файл «lab5-2.asm» для редактирования и исправляем текст программы с использованием подпрограмм из внешнего файла «in_out.asm» (рис. [-@fig:029])

Текст программы:

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',10
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

```

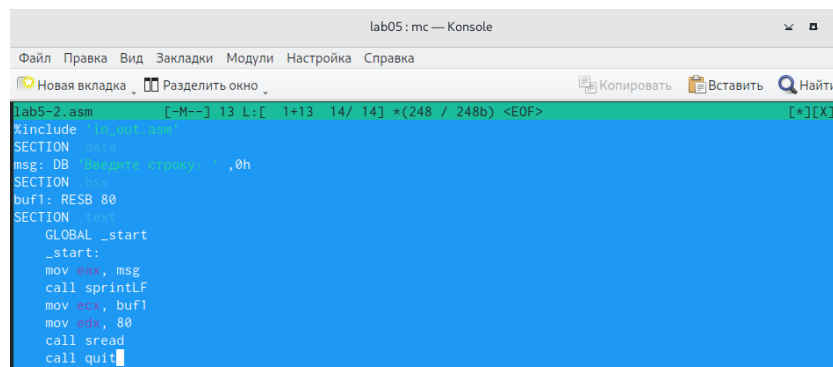


Рис. 29: Исправленная программа

Транслируем текст программы в объектный файл, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл (рис. [-@fig:030]), (рис. [-@fig:031]), (рис. [-@fig:032]), (рис. [-@fig:033])

```
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
```

Рис. 30: Транслирование

```

eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2.asm lab5-2.o list5-1.lst main5
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $

```

Рис. 31: Проверка

```
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o list5-1.lst main5
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $
```

Рис. 32: Компоновка

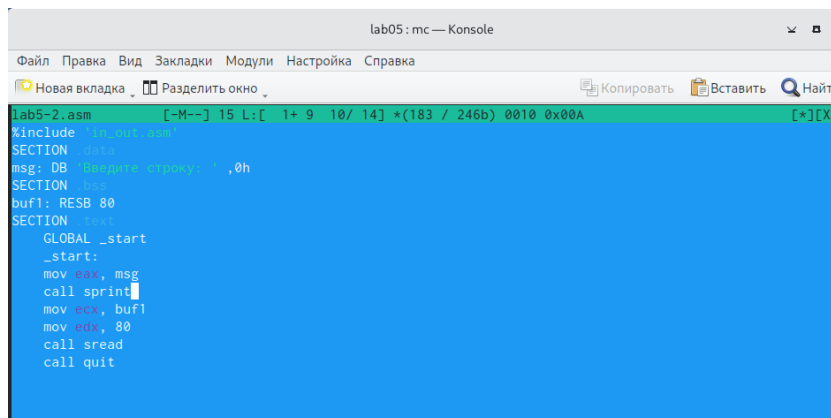
```
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Верниковская Екатерина
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $
```

Рис. 33: Запуск программы

В файле «lab5-2.asm» заменяем подпрограмму ‘sprintLF’ на ‘sprint’ (рис. [-@fig:034])

Текст исправленной программы:

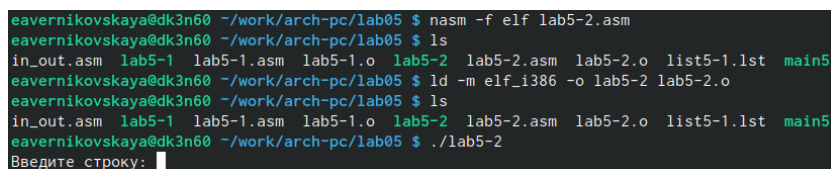
```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',10
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL __start
__start:
    mov eax, msg
    call sprint
    mov ecx, buf1
    mov edx, 80
    call sread
    call quit
```

```
lab5-2.asm [-M--] 15 L: [ 1+ 9 10/ 14] *(183 / 246b) 0010 0x00A [*][X]
#include "in_out.asm"
SECTION .data
msg: DB "Нажмите Ctrl+C, 0h"
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 34: Исправление программы

Создаём исполняемый файл и проверяем его работу (рис. [-@fig:035])



```
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o list5-1.lst main5
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o list5-1.lst main5
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: 
```

Рис. 35: Проверка

Ответ на вопрос: в первом случае программа просила нас ввести текст на следующей строке, во втором - сразу после знака двоеточия.

Задание для самостоятельной работы

Создаём копию файла «lab5-1.asm» с именем «lab5-1-1.asm» (рис. [-@fig:036]), (рис. [-@fig:037])

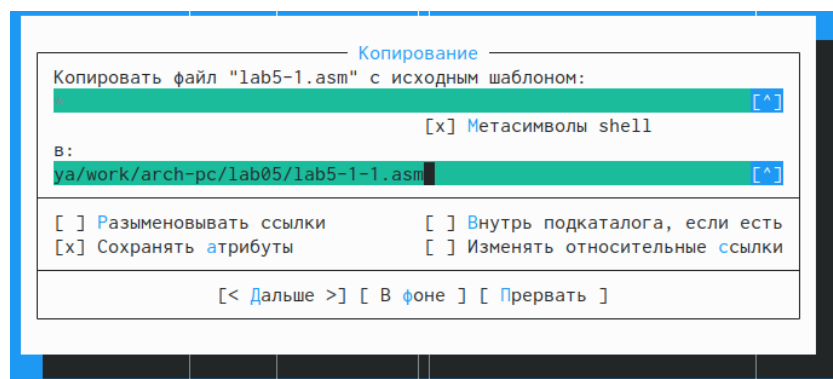


Рис. 36: Создание копии файла

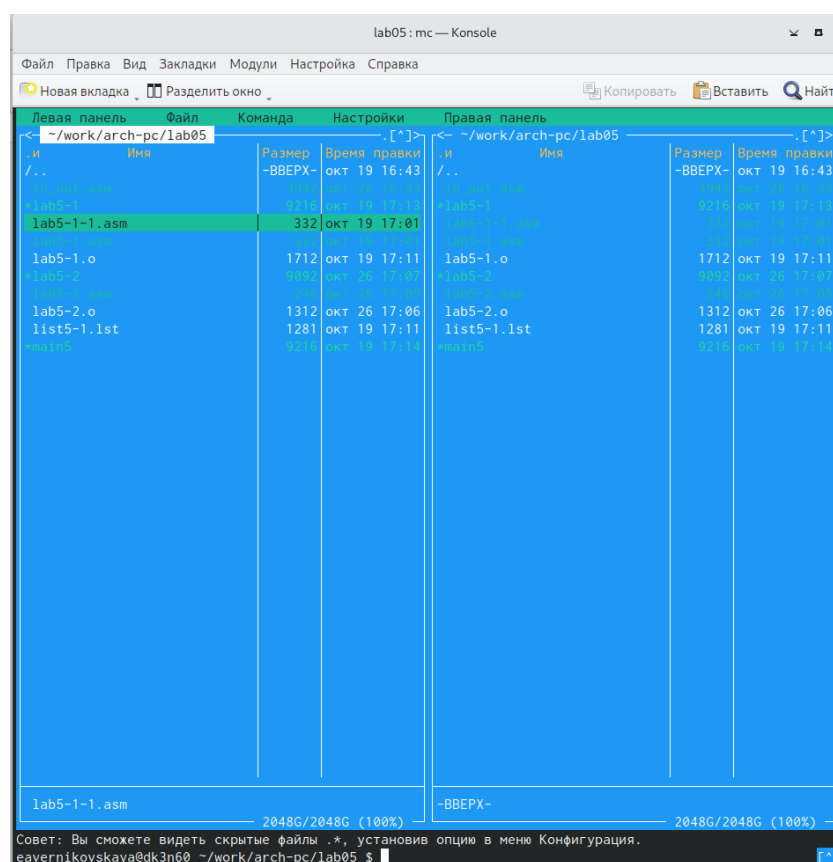


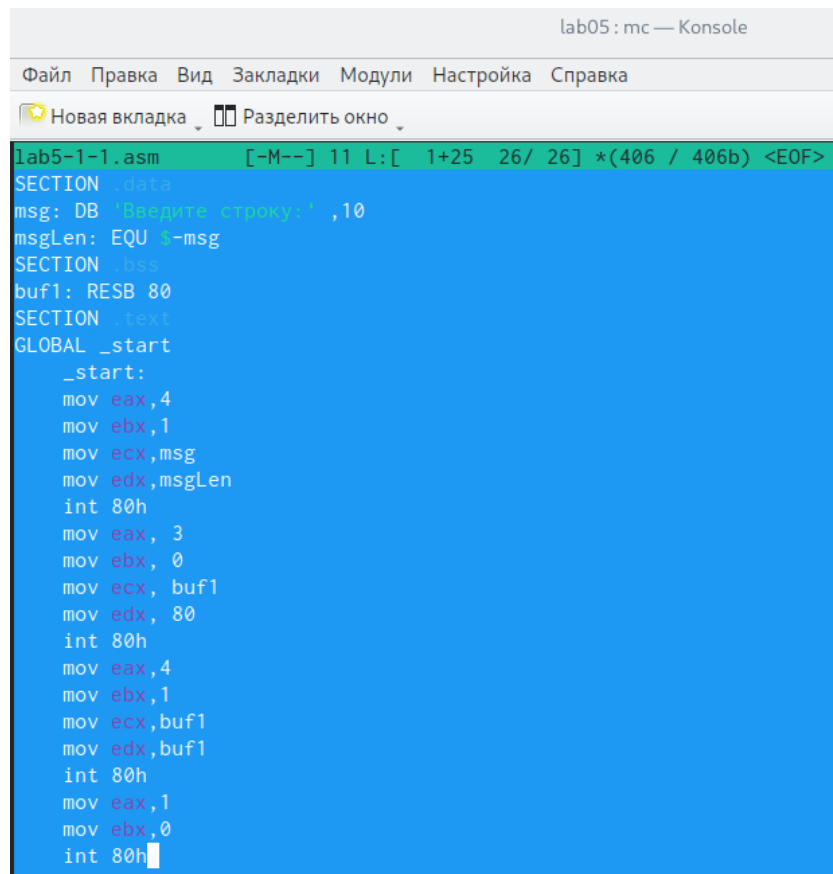
Рис. 37: Проверка

Вносим 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 введённую строку на экран; (рис. [-@fig:038])

Текст программы:

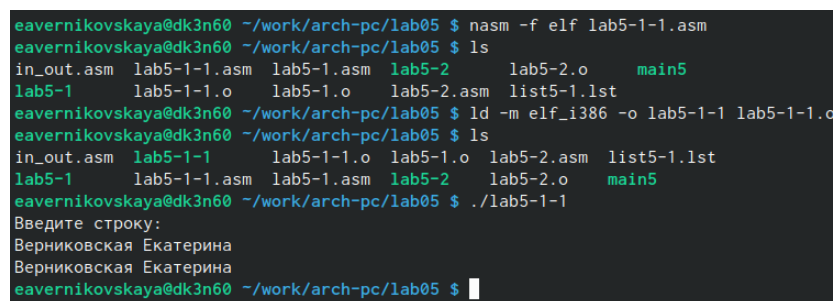
```
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
    mov eax,4
    mov ebx,1
    mov ecx,msg
    mov edx,msgLen
    int 80h
    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h
    mov eax,1
    mov ebx,0
    int 80h
```



```
lab5-1-1.asm [-M--] 11 L: [ 1+25 26/ 26] *(406 / 406b) <EOF>
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,buf1
    int 80h
    mov eax,1
    mov ebx,0
    int 80h
```

Рис. 38: Текст программы

Получаем исполняемый файл и проверяем его работу. На приглашение вводим свою фамилию и имя (рис. [-@fig:039])



```
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-1.asm
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1-1.asm lab5-1.o lab5-2 lab5-2.o main5
lab5-1 lab5-1-1.o lab5-1.o lab5-2.asm list5-1.lst
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1-1 lab5-1-1.o lab5-1.o lab5-2.asm list5-1.lst
lab5-1 lab5-1-1.asm lab5-1.o lab5-2 lab5-2.o main5
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ./lab5-1-1
Введите строку:
Верниковская Екатерина
Верниковская Екатерина
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $
```

Рис. 39: Проверка

Создаём копию файла «lab5-2.asm» с именем «lab5-2-2.asm» (рис. [-@fig:040]), (рис.

[-@fig:041])

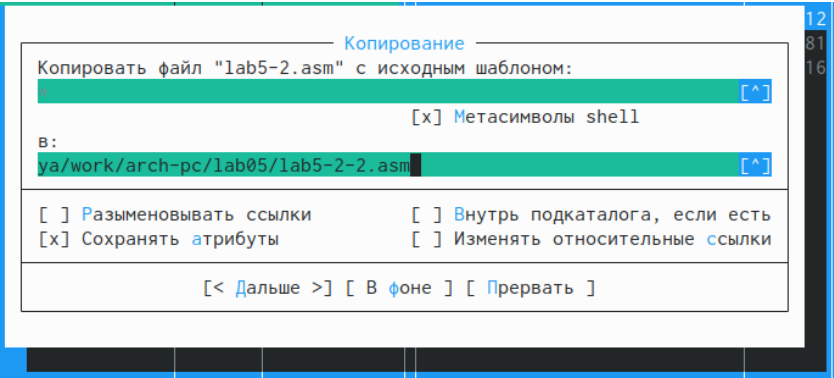


Рис. 40: Создание копии файла

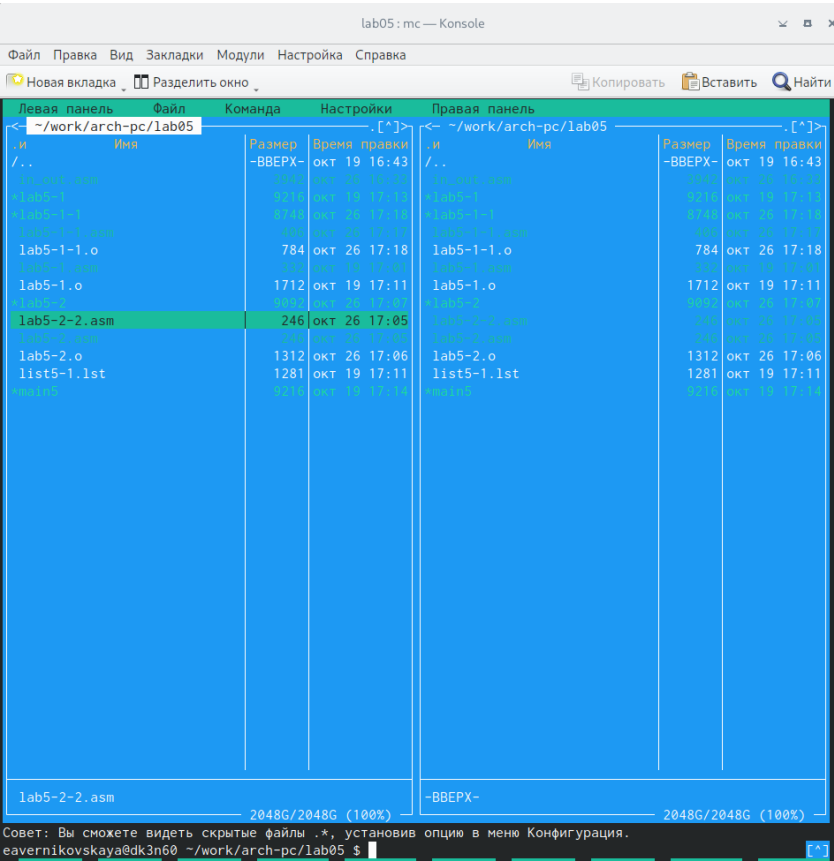


Рис. 41: Проверка

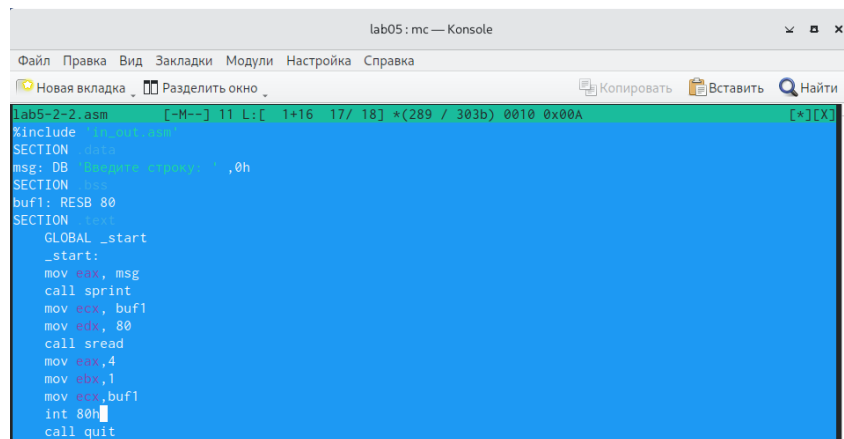
Вносим изменения в программу (уже с использованием подпрограмм из внешнего

файла «in_out.asm»), так чтобы она работала по следующему алгоритму:

- Вывести приглашение типа “Введите строку:”;
- Ввести строку с клавиатуры;
- Вывести введённую строку на экран; (рис. [-@fig:042])

Текст программы:

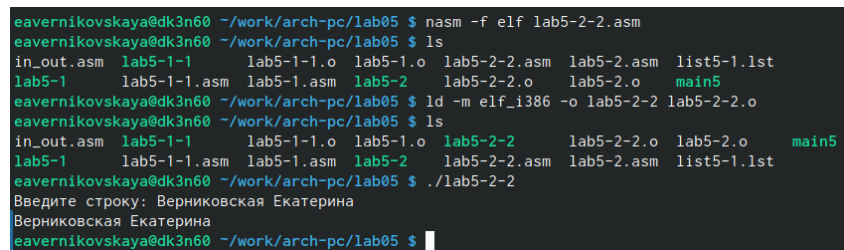
```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',10
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
    mov eax, msg
    call sprint
    mov ecx, buf1
    mov edx, 80
    call sread
    mov eax,4
    mov ebx,1
    mov ecx,buf1
    int 80h
    call quit
```



```
lab5-2-2.asm [-M--] 11 L: [ 1+16 17/ 18] *(289 / 303b) 0010 0x00A [*][X]
%include "in_out.asm"
SECTION .data
msg: DB "Верниковская Екатерина", 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 ecx, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit
```

Рис. 42: Текст программы

Создаём исполняемый файл и проверяем его работу (рис. [-@fig:043])



```
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2-2.asm
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1-1 lab5-1-1.o lab5-1.o lab5-2-2.asm lab5-2-2.o lab5-2.o list5-1.lst
lab5-1 lab5-1-1.asm lab5-1.asm lab5-2 lab5-2-2.o lab5-2.o main5
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-2 lab5-2-2.o
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ls
in_out.asm lab5-1-1 lab5-1-1.o lab5-1.o lab5-2-2 lab5-2-2.o lab5-2.o main5
lab5-1 lab5-1-1.asm lab5-1.asm lab5-2 lab5-2-2.asm lab5-2.o lab5-2.o list5-1.lst
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $ ./lab5-2-2
Введите строку: Верниковская Екатерина
Верниковская Екатерина
eavernikovskaya@dk3n60 ~/work/arch-pc/lab05 $
```

Рис. 43: Проверка

Выводы

В ходе выполнения лабораторной работы мы приобрели навыки работы в Midnight Commander, а так же освоили инструкции языка ассемблера `mov` и `int`.