

# **Отчёт по лабораторной работе №5**

**Дисциплина: Архитектура компьютера**

Калашникова Ольга Сергеевна

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Цель работы</b>                             | <b>5</b>  |
| <b>2</b> | <b>Задание</b>                                 | <b>6</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>          | <b>7</b>  |
| 3.1      | Изучение Midnight Commander. . . . .           | 7         |
| 3.2      | Подключение внешнего файла in_out.asm. . . . . | 13        |
| 3.3      | Задание для самостоятельной работы . . . . .   | 15        |
| <b>4</b> | <b>Выводы</b>                                  | <b>21</b> |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 3.1  | Открытие Midnight Commander 1 . . . . .          | 7  |
| 3.2  | Открытие Midnight Commander 2 . . . . .          | 7  |
| 3.3  | Переход в каталог '~/.work/arch-pc' . . . . .    | 8  |
| 3.4  | Создание папки 'lab05' 1 . . . . .               | 8  |
| 3.5  | Создание папки 'lab05' 2 . . . . .               | 8  |
| 3.6  | Переход в папку 'lab05' . . . . .                | 9  |
| 3.7  | Создание файла 'lab5-1.asm' . . . . .            | 9  |
| 3.8  | Создание файла 'lab5-1.asm' (проверка) . . . . . | 9  |
| 3.9  | Выбор редактора . . . . .                        | 10 |
| 3.10 | Открытие файла 'lab5-1.asm' . . . . .            | 10 |
| 3.11 | Ввод текста программы . . . . .                  | 11 |
| 3.12 | Проверка . . . . .                               | 11 |
| 3.13 | Трансляция . . . . .                             | 12 |
| 3.14 | Проверка . . . . .                               | 12 |
| 3.15 | Компоновка . . . . .                             | 12 |
| 3.16 | Проверка . . . . .                               | 12 |
| 3.17 | Запуск файла . . . . .                           | 13 |
| 3.18 | Файл in_out.asm в нужной папке . . . . .         | 13 |
| 3.19 | Создание копии файла с другим именем . . . . .   | 13 |
| 3.20 | Исправление текста программы . . . . .           | 14 |
| 3.21 | Трансляция, компоновка, запуск файла . . . . .   | 14 |
| 3.22 | Проверка наличия файлов . . . . .                | 14 |
| 3.23 | Замена текста . . . . .                          | 15 |
| 3.24 | Проверка . . . . .                               | 15 |
| 3.25 | Создание копии файла с другим именем . . . . .   | 15 |
| 3.26 | Проверка . . . . .                               | 16 |
| 3.27 | Исправление текста программы . . . . .           | 16 |
| 3.28 | Трансляция, компоновка, запуск файла . . . . .   | 17 |
| 3.29 | Проверка наличия файлов . . . . .                | 17 |
| 3.30 | Создание копии файла с другим именем . . . . .   | 17 |
| 3.31 | Проверка . . . . .                               | 18 |
| 3.32 | Исправление текста программы . . . . .           | 19 |
| 3.33 | Трансляция, компоновка, запуск файла . . . . .   | 19 |
| 3.34 | Проверка наличия файлов . . . . .                | 20 |

## **Список таблиц**

# 1 Цель работы

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

## 2 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 исполняемый файл и проверить его работу.
9. Создать копию файла 'lab5-2.asm'. Внести изменения в программу, но уже с использованием подпрограмм из внешнего файла 'in\_out.asm', так чтобы она работала по определённому алгоритму.
10. Создать исполняемый файл и проверить его работу.

## 3 Выполнение лабораторной работы

### 3.1 Изучение Midnight Commander.

Открываем Midnight Commander с помощью команды 'mc' (рис. 3.1),(рис. 3.2)

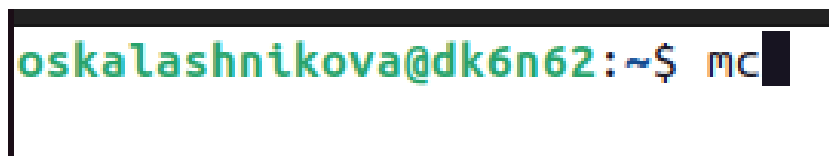


Рис. 3.1: Открытие Midnight Commander 1

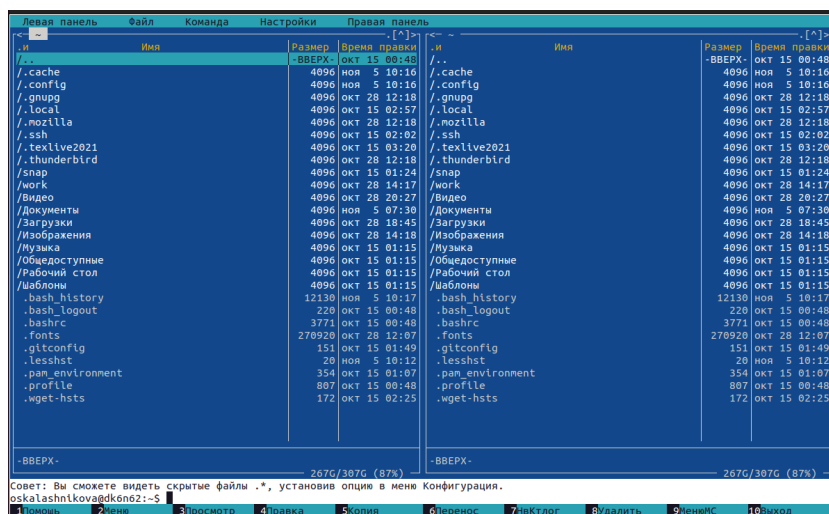


Рис. 3.2: Открытие Midnight Commander 2

Пользуясь клавишами на клавиатуре стрелочка вверх, стрелочка вниз и 'enter' переходим в каталог '~/.work/arch-pc' (рис. [3.3])

| Левая панель          | Файл | Команда | Настройки    | Правая панель |
|-----------------------|------|---------|--------------|---------------|
| ~ /work/arch-pc .[^]> |      |         |              |               |
| .и                    | Имя  | Размер  | Время правки |               |
| /..                   |      | -ВВЕРХ- | окт 28 14:17 |               |
| /lab04                |      | 4096    | окт 28 17:14 |               |

Рис. 3.3: Переход в каталог ‘~/work/arch-pc’

С помощью функциональной клавиши ‘F7’ создаём папку ‘lab05’ и переходим в неё (рис. [3.4]), (рис. [3.5]), рис. [3.6])

Создать новый каталог

Введите имя каталога:

lab05 [^]

[< Дальше >] [ Прервать ]

Рис. 3.4: Создание папки ‘lab05’ 1

| Левая панель       |     | Файл    | Команда      | Настройки       | Правая панель |         |              |
|--------------------|-----|---------|--------------|-----------------|---------------|---------|--------------|
| <- ~ /work/arch-pc |     |         |              | <- ~            | >.[^]>        |         |              |
| .и                 | Имя | Размер  | Время правки | .и              | Имя           | Размер  | Время правки |
| /..                |     | -ВВЕРХ- | окт 28 14:17 | /..             |               | -ВВЕРХ- | окт 15 00:48 |
| /lab04             |     | 4096    | окт 28 17:14 | /.cache         |               | 4096    | ноя 5 10:16  |
| /lab05             |     | 4096    | ноя 5 10:20  | /.config        |               | 4096    | ноя 5 10:16  |
|                    |     |         |              | /.gnupg         |               | 4096    | окт 28 12:18 |
|                    |     |         |              | /.local         |               | 4096    | окт 15 02:57 |
|                    |     |         |              | /.mozilla       |               | 4096    | окт 28 12:18 |
|                    |     |         |              | /.ssh           |               | 4096    | окт 15 02:02 |
|                    |     |         |              | /.texlive2021   |               | 4096    | окт 15 03:20 |
|                    |     |         |              | /.thunderbird   |               | 4096    | окт 28 12:18 |
|                    |     |         |              | /snap           |               | 4096    | окт 15 01:24 |
|                    |     |         |              | /work           |               | 4096    | окт 28 14:17 |
|                    |     |         |              | /Видео          |               | 4096    | окт 28 20:27 |
|                    |     |         |              | /Документы      |               | 4096    | ноя 5 07:30  |
|                    |     |         |              | /Загрузки       |               | 4096    | окт 28 18:45 |
|                    |     |         |              | /Изображения    |               | 4096    | окт 28 14:18 |
| /lab05             |     |         |              | -ВВЕРХ-         |               |         |              |
| 267G/307G (87%)    |     |         |              | 267G/307G (87%) |               |         |              |

Совет: Используйте C-x р для копирования текущего пути в командную строку.  
oskalahnikova@dk6n62:~/work/arch-pc\$

1Помощь

2Меню

3Про-тр

4Правка

5Копия

6Пер-ос

7НвК-ог

8Уда-ть

9МенюМС

10Выход

Рис. 3.5: Создание папки ‘lab05’ 2



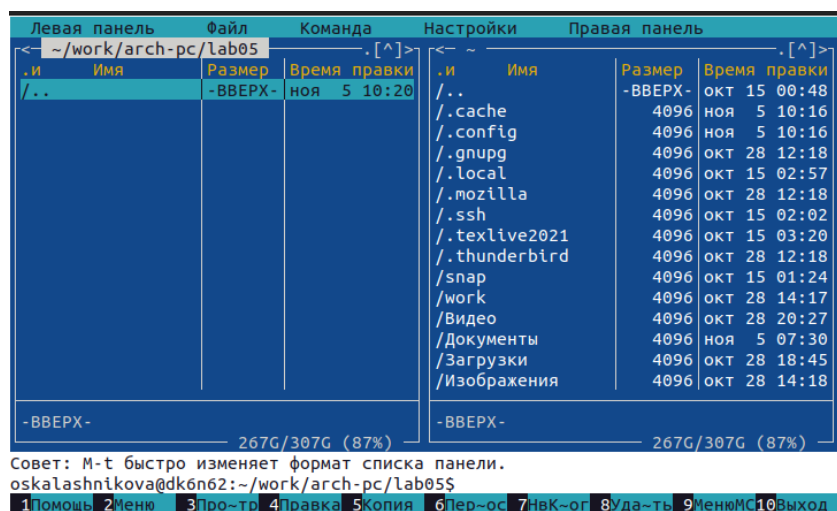


Рис. 3.6: Переход в папку 'lab05'

Пользуясь строкой ввода и командой 'touch' создаём файл 'lab5-1.asm' (рис.[3.7]), (рис. [3.8])

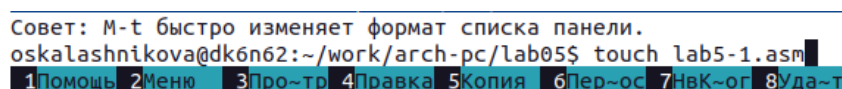


Рис. 3.7: Создание файла 'lab5-1.asm'

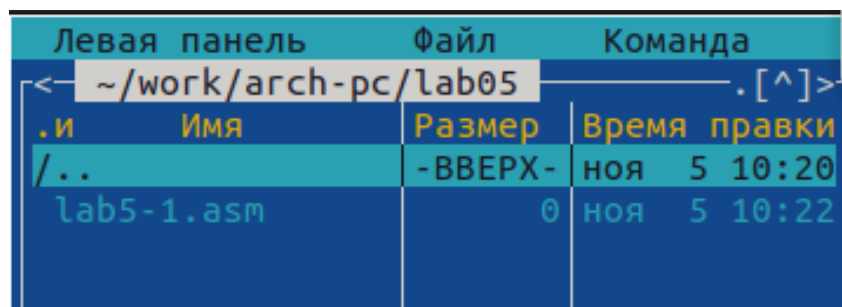


Рис. 3.8: Создание файла 'lab5-1.asm' (проверка)

С помощью функциональной клавиши 'F4' открываем файл 'lab5-1.asm' для редактирования во встроенном редакторе. Я использовала mcedit (рис. [3.9]), (рис. [3.10])

```
oskalashnikova@dk6n62:~$ mc
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ touch lab5-1.asm

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/mcedit
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 2
```

Рис. 3.9: Выбор редактора

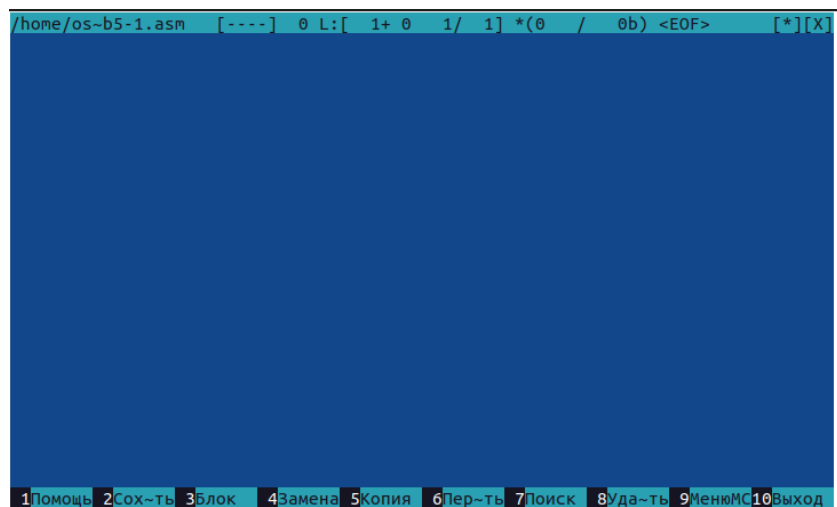


Рис. 3.10: Открытие файла 'lab5-1.asm'

Вводим предложенный текст программы и с помощью функциональной клавиши 'F2' сохраняем изменения и выходим с помощью 'F10' (рис. [3.11])

```

/home/os-b5-1.asm [-M--] 20 L:[ 1+ 6 7/ 35] *(515 /2431b) 1088 0x440 [*][X]
;-----
; 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 '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 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС 10Выход

Рис. 3.11: Ввод текста программы

С помощью функциональной клавиши 'F3' открываем изменённый файл, чтобы убедиться, что он содержит текст программы (рис. [3.12])

```

/home/oskashnikova/work/arch-pc/lab05/labs-1.asm 2431/2431 100%
;-----
; 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 '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 2Заверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход

Рис. 3.12: Проверка

Оттранслируем текст программы 'lab5-1.asm' в объектный файл. (рис. [3.13]), (рис. [3.14])

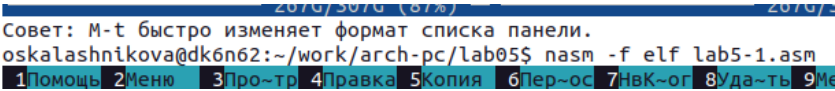


Рис. 3.13: Трансляция

| Левая панель           | Файл    | Команда      |
|------------------------|---------|--------------|
| < ~/work/arch-pc/lab05 |         | .[^]>        |
| .и                     | Имя     | Размер       |
| /..                    | -ВВЕРХ- | Время правки |
| lab5-1.asm             | 2431    | ноя 5 10:26  |
| lab5-1.o               | 752     | ноя 5 10:28  |

Рис. 3.14: Проверка

Выполним компоновку объектного файла. (рис. [3.15]), (рис. [3.16])

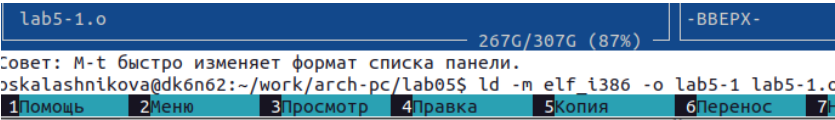


Рис. 3.15: Компоновка

| .и         | Имя     | Размер      | Время правки |
|------------|---------|-------------|--------------|
| /..        | -ВВЕРХ- | ноя 5 10:20 |              |
| *lab5-1    | 8744    | ноя 5 10:30 |              |
| lab5-1.asm | 2431    | ноя 5 10:26 |              |
| lab5-1.o   | 752     | ноя 5 10:28 |              |

Рис. 3.16: Проверка

Запустим получившийся исполняемый файл. На запрос вводим наше ФИО. (рис. [3.17])

```
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Olga Kalashnikova
```

Рис. 3.17: Запуск файла

## 3.2 Подключение внешнего файла in\_out.asm.

Скачиваем файл 'in\_out.asm' с ТУИС и помещаем в нужную папку с помощью функциональной клавиши 'F5' (забыла сделать скрины копирования файла с помощью mc) (рис. [3.18])

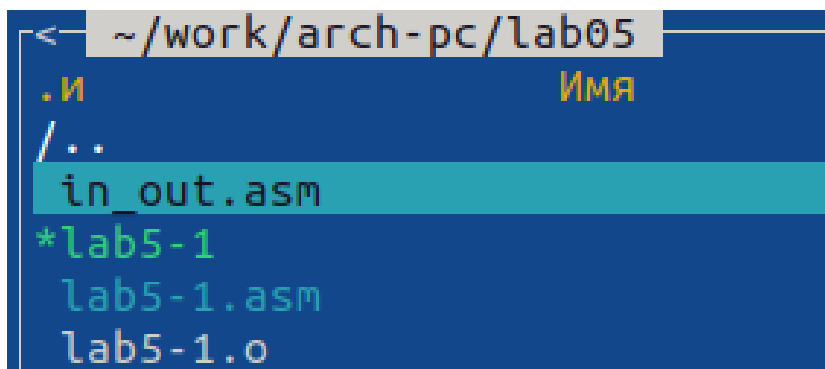


Рис. 3.18: Файл in\_out.asm в нужной папке

С помощью той же функциональной клавиши 'F5' создаём копию файла 'lab5-1.asm' с именем 'lab5-2.asm' (рис. [3.19])

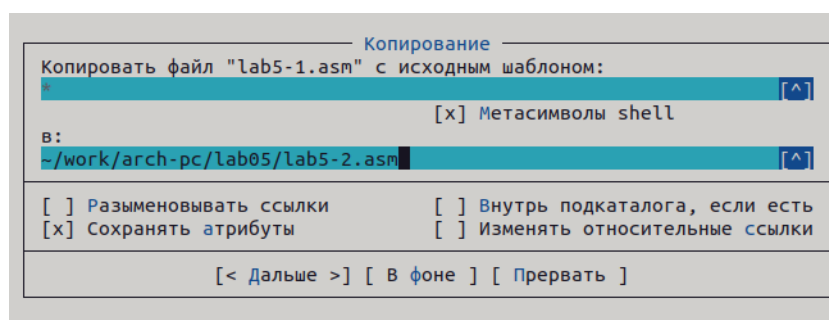


Рис. 3.19: Создание копии файла с другим именем

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. [3.20])

```

/home/oskalashnikova/work/arch-pc/lab05/lab5-2.asm [----] 42 L: [ 1+21 22/ 22] *(1239/1239b) <EOF> [X]
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла

SECTION .data ; Секция иницированных данных
msg: DB "Введите строку: ",0h ; сообщение

SECTION .bss ; Секция не иницированных 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 выводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис. 3.20: Исправление текста программы

Транслируем текст программы в объектный файл, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. (рис. [3.21]), (рис. [3.22])

```

oskalashnikova@dk6n62:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Kalashnikova Olga
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ █
```

Рис. 3.21: Трансляция, компоновка, запуск файла

| Левая панель |  | Файл                 | Команда | Настройки |              | Правая панель |  |
|--------------|--|----------------------|---------|-----------|--------------|---------------|--|
| < ~          |  | ~/work/arch-pc/lab05 |         |           |              | .[^]>         |  |
| .и           |  | Имя                  |         | Размер    | Время правки | .и            |  |
| /..          |  |                      |         | -ВВЕРХ-   | ноя 5 11:25  | /..           |  |
| in_out.asm   |  |                      |         | 3942      | ноя 5 10:34  | /.cache       |  |
| *lab5-1      |  |                      |         | 8744      | ноя 5 10:30  | /.config      |  |
| lab5-1.asm   |  |                      |         | 2431      | ноя 5 10:26  | /.gnupg       |  |
| lab5-1.o     |  |                      |         | 752       | ноя 5 10:28  | /.local       |  |
| *lab5-2      |  |                      |         | 9092      | ноя 5 11:39  | /.mozilla     |  |
| lab5-2.asm   |  |                      |         | 1239      | ноя 5 11:38  | /.ssh         |  |
| lab5-2.o     |  |                      |         | 1312      | ноя 5 11:39  | /.texlive2    |  |
|              |  |                      |         |           |              | /.thunderb    |  |
|              |  |                      |         |           |              | /snap         |  |
|              |  |                      |         |           |              | /work         |  |

Рис. 3.22: Проверка наличия файлов

В файле ‘lab5-2.asm’ заменяем подпрограмму ‘sprintf’ на ‘print’ (рис. [3.23])

```

/home/oskalashnikova/work/arch-pc/lab05/lab5-2.asm [-М-] 12 L: [ 1+17 18/ 22] *(858 /1237b) 0032 0x020 [*] [X]
; 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 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.23: Замена текста

Создаём исполняемый файл и проверяем его работу (рис. [3.24])

```

oskalashnikova@dk6n62:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Kalashnikova Olga
oskalashnikova@dk6n62:~/work/arch-pc/lab05$

```

Рис. 3.24: Проверка

Ответ на вопрос в чем разница между 'sprintLF' и 'sprint': Разница заключается в том, что 'sprintLF' дополнительно печатает символ перевода строки (0AH) и потом вызывает 'sprint'.

### 3.3 Задание для самостоятельной работы

Создаём копию файла 'lab5-1.asm' с именем 'lab5-3.asm' (рис. [3.25]), (рис. [3.26])

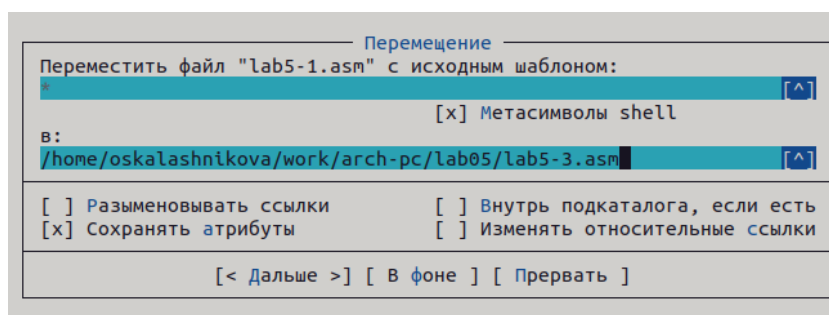


Рис. 3.25: Создание копии файла с другим именем

| ~ /work/arch-pc/lab05 |     | . [^]>  |       |        |
|-----------------------|-----|---------|-------|--------|
| .и                    | Имя | Размер  | Время | правки |
| ./..                  |     | -ВВЕРХ- | ноя 5 | 11:25  |
| in_out.asm            |     | 3942    | ноя 5 | 10:34  |
| *lab5-1               |     | 8744    | ноя 5 | 10:30  |
| lab5-1.o              |     | 752     | ноя 5 | 10:28  |
| *lab5-2               |     | 9092    | ноя 5 | 11:42  |
| lab5-2.asm            |     | 1237    | ноя 5 | 11:42  |
| lab5-2.o              |     | 1312    | ноя 5 | 11:42  |
| lab5-3.asm            |     | 2431    | ноя 5 | 10:26  |

Рис. 3.26: 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 введенную строку на экран. (рис. [3.27])

```
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт

SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

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

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

mov eax,4
mov ebx,1
mov ecx, buf1
mov edx, 80
int 80h

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

Рис. 3.27: Исправление текста программы

Транслируем текст программы в объектный файл, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. (рис. [3.28]),



(рис. [3.29])

```
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Kalashnikova Olga
Kalashnikova Olga
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ █
```

Рис. 3.28: Трансляция, компоновка, запуск файла

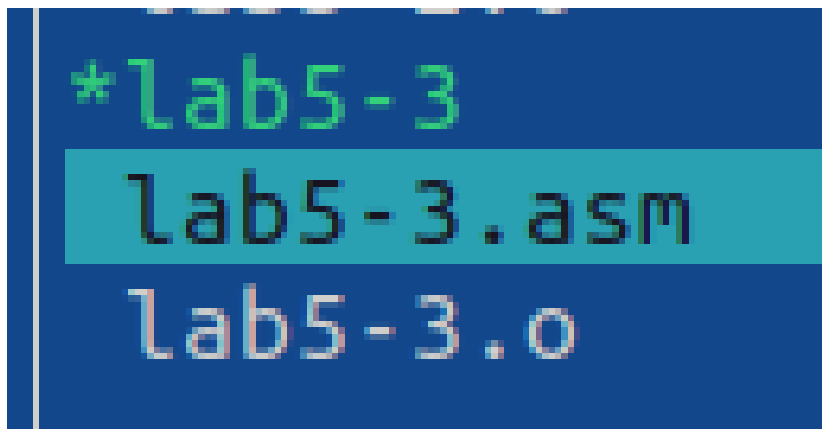


Рис. 3.29: Проверка наличия файлов

Создаём копию файла 'lab5-2.asm' с именем 'lab5-4.asm'. (рис. [3.30]), (рис. [3.31])

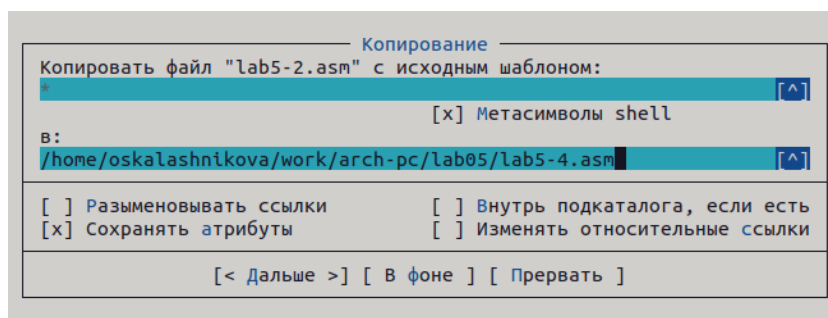


Рис. 3.30: Создание копии файла с другим именем

```
< ~/work/arch-pc/lab05
.и      Имя
/..
in_out.asm
*lab5-1
lab5-1.asm
lab5-1.o
*lab5-2
lab5-2.asm
lab5-2.o
*lab5-3
lab5-3.asm
lab5-3.o
lab5-4.asm
```

Рис. 3.31: Проверка

Исправляем текст программы 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. [3.32])

```

/home/oskalashnikova/work/arch-pc/lab05/lab5-4.asm [-M--] 0 L:
;-----
; 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 sprintLF ; 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
mov edx, 80
call sprintLF

call quit ; вызов подпрограммы завершения

```

Рис. 3.32: Исправление текста программы

Транслируем текст программы в объектный файл, выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. (рис. [3.33]), (рис. [3.34])

```

oskalashnikova@dk6n62:~/work/arch-pc/lab05$ nasm -f elf lab5-4.asm
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-4 lab5-4.o
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ ./lab5-4
Введите строку:
Kalashnikova Olga
Kalashnikova Olga
oskalashnikova@dk6n62:~/work/arch-pc/lab05$ █

```

Рис. 3.33: Трансляция, компоновка, запуск файла

```
*lab5-4  
lab5-4.asm  
lab5-4.o
```

Рис. 3.34: Проверка наличия файлов

## 4 Выводы

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