

Отчёт по лабораторной работе №4

Дисциплина: архитектура компьютера

Горобцова Арина Романовна

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 3.1 | Основы работы с тс | 7 |
| 3.2 | Структура программы на языке ассемблера NASM | 8 |
| 3.3 | Подключение внешнего файла | 11 |
| 3.4 | Выполнение заданий для самостоятельной работы | 14 |
| 4 | Выводы | 20 |

Список иллюстраций

| | | |
|------|---|----|
| 3.1 | Открытый тс | 7 |
| 3.2 | Создание каталога | 8 |
| 3.3 | Перемещение между директориями | 8 |
| 3.4 | Создание файла | 8 |
| 3.5 | Открытие файла для редактирования | 9 |
| 3.6 | Редактирование файла | 9 |
| 3.7 | Открытие файла для просмотра | 10 |
| 3.8 | Компиляция файла и передача на обработку компоновщику | 10 |
| 3.9 | Исполнение файла | 10 |
| 3.10 | Копирование файла | 11 |
| 3.11 | Копирование файла | 12 |
| 3.12 | Редактирование файла | 12 |
| 3.13 | Исполнение файла | 13 |
| 3.14 | Отредактированный файл | 13 |
| 3.15 | Исполнение файла | 14 |
| 3.16 | Копирование файла | 14 |
| 3.17 | Редактирование файла | 15 |
| 3.18 | Исполнение файла | 15 |
| 3.19 | Копирование файла | 17 |
| 3.20 | Редактирование файла | 18 |
| 3.21 | Исполнение файла | 18 |

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

1. Основы работы с тс
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Выполнение лабораторной работы

3.1 Основы работы с mc

Открываю Midnight Commander, введя в терминал mc и перехожу в каталог ~/work/arch-рс/lab05, используя файловый менеджер mc(рис. 3.1).

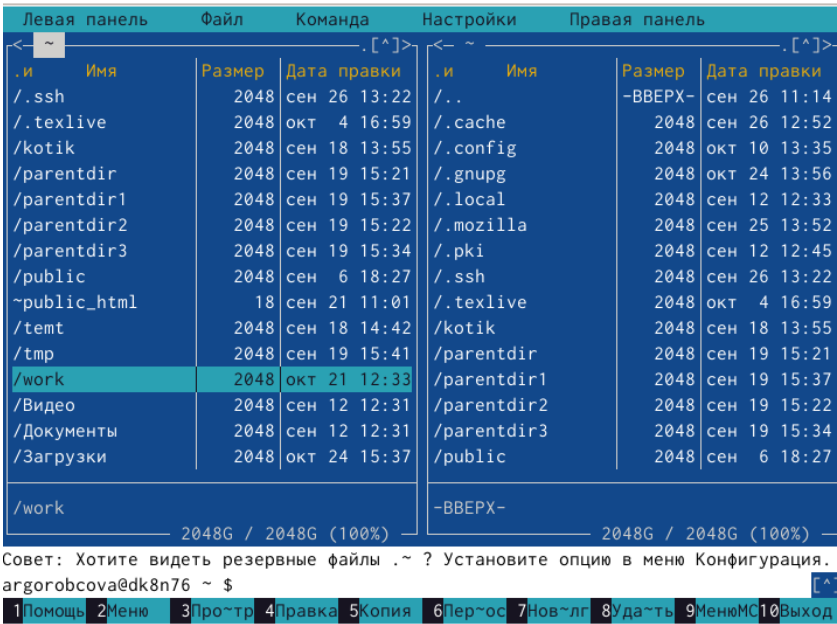


Рис. 3.1: Открытый mc

С помощью функциональной клавиши F7 создаю каталог lab05 (рис. 3.2).

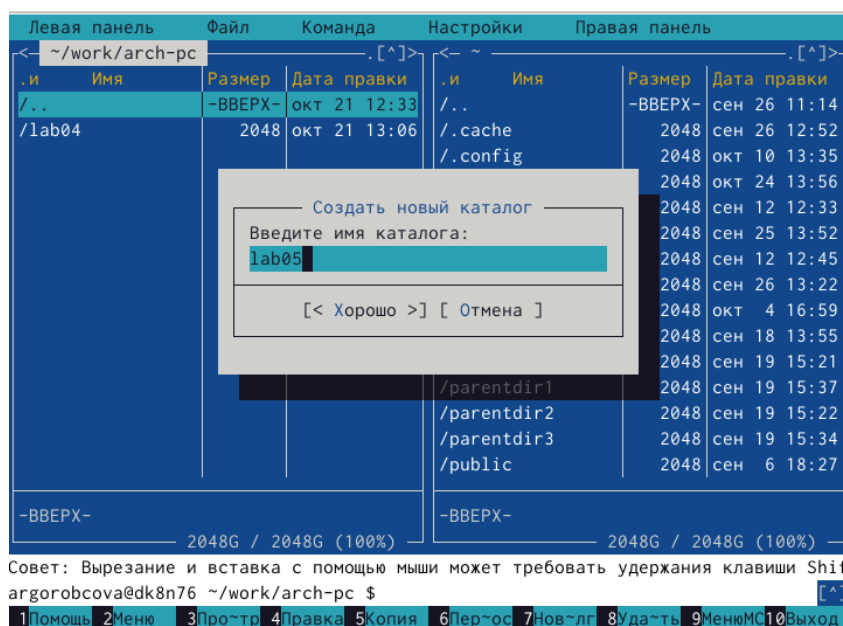


Рис. 3.2: Создание каталога

Перехожу в созданный каталог (рис. 3.3).

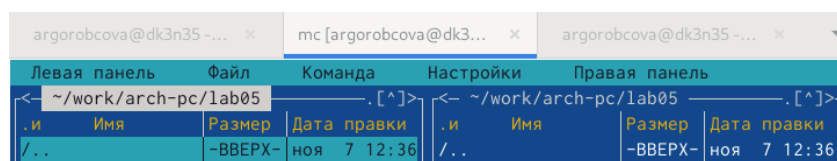


Рис. 3.3: Перемещение между директориями

В строке ввода прописываю команду `touch lab5-1.asm`, чтобы создать файл, в котором буду работать (рис. 3.4).

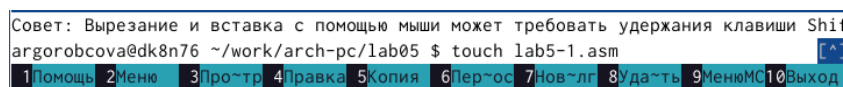


Рис. 3.4: Создание файла

3.2 Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования во встроенном редакторе (рис. 3.5).

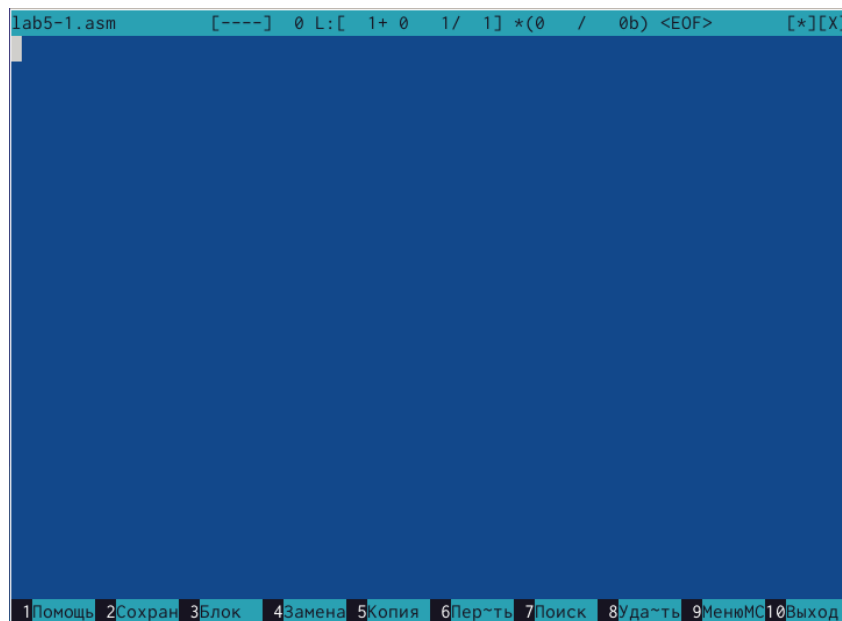


Рис. 3.5: Открытие файла для редактирования

Ввожу в файл код программы для запроса строки у пользователя (рис. 3.6). Далее выхожу из файла (Ctrl+X), сохраняя изменения (Y, Enter). Далее проверяю с помощью утилиты ls правильность выполнения команды.

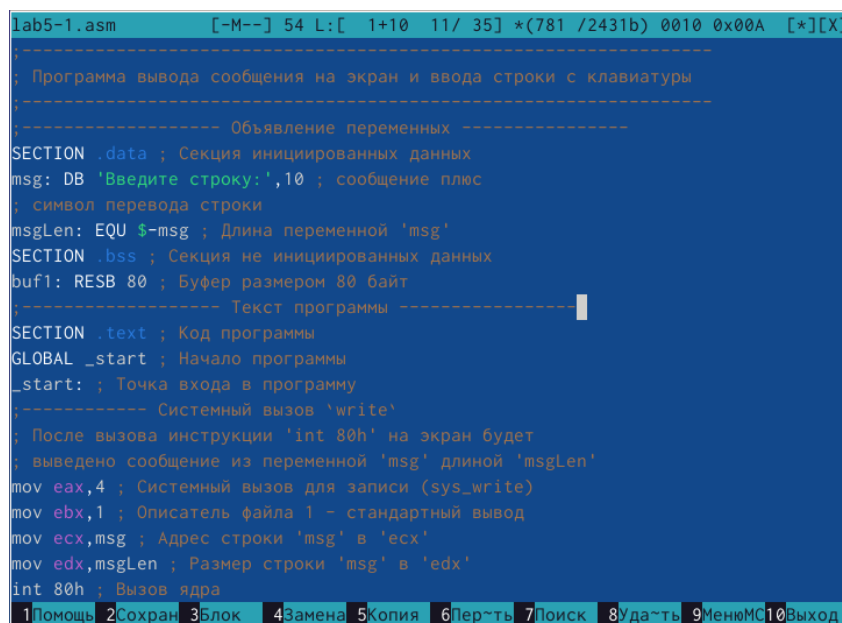


Рис. 3.6: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, что-

бы проверить, содержит ли файл текст программы (рис. 3.7).

```
/afs/.dk.sci.pfu.edu.ru~ch-pc/lab05/lab5-1.asm 1522/2432 62
;-----
; 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' -----
```

Рис. 3.7: Открытие файла для просмотра

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-1.asm`. Создался объектный файл `lab5-1.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o`. Создался исполняемый файл `lab5-1`. (рис. 3.8).

```
argorobcova@dk8n76 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
argorobcova@dk8n76 ~/work/arch-pc/lab05 $ ls
lab5-1.asm lab5-1.asm.save lab5-1.o
argorobcova@dk8n76 ~/work/arch-pc/lab05 $ ld -m elf_i386 lab5-1.o
argorobcova@dk8n76 ~/work/arch-pc/lab05 $ ls
a.out lab5-1.asm lab5-1.asm.save lab5-1.o
argorobcova@dk8n76 ~/work/arch-pc/lab05 $
```

Рис. 3.8: Компиляция файла и передача на обработку компоновщику

Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. 3.9).

```
argorobcova@dk8n76 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Горобцова Арина Романовна
```

Рис. 3.9: Исполнение файла

3.3 Подключение внешнего файла

Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки”. С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05 (рис. 3.10).

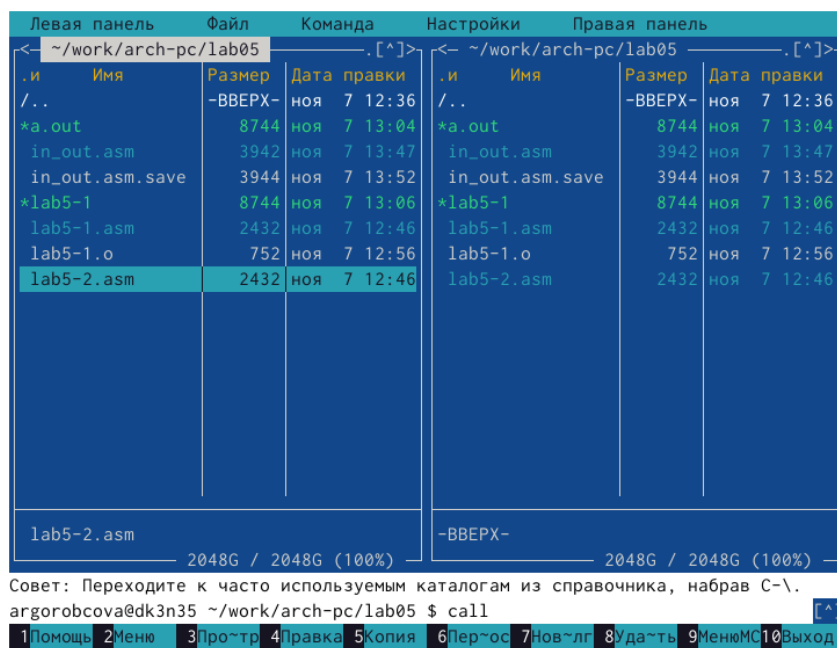


Рис. 3.10: Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла (рис. 3.11).

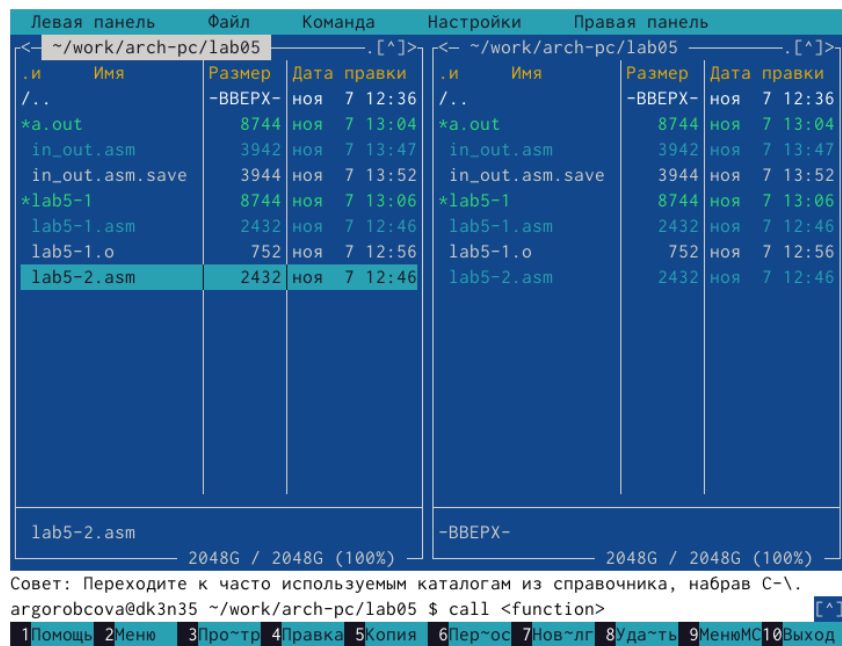


Рис. 3.11: Копирование файла

Изменяю содержимое файла lab5-2.asm во встроенном редакторе, чтобы в программе использовались подпрограммы из внешнего файла in_out.asm.(рис. 3.12).

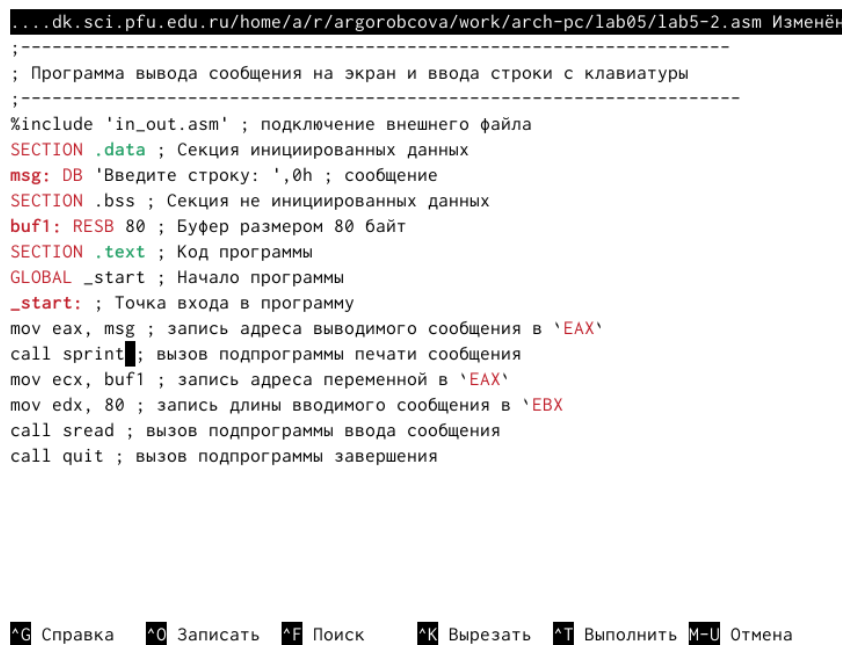


Рис. 3.12: Редактирование файла

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Создался объектный файл `lab5-2.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o`. Создался исполняемый файл `lab5-2`. Запускаю исполняемый файл (рис. 3.13).

```
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ls
a.out      in_out.asm.save  lab5-1.asm  lab5-2      lab5-2.o
in_out.asm lab5-1          lab5-1.o    lab5-2.asm
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Горбцова Арина Романовна
```

Рис. 3.13: Исполнение файла

Открываю файл `lab5-2.asm` для редактирования в nano функциональной клавишей F4. Изменяю в нем подпрограмму `sprintLF` на `sprint`. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий (рис. 3.14).

```
...dk.sci.pfu.edu.ru/home/a/r/argorobcova/work/arch-pc/lab05/lab5-2.asm Изменён
;-----
; 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 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

^G Справка ^O Записать ^F Поиск ^K Вырезать ^T Выполнить M-U Отмена
```

Рис. 3.14: Отредактированный файл

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. 3.15).

```

argorobcova@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ls
a.out      in_out.asm.save  lab5-1.asm  lab5-2      lab5-2.o
in_out.asm lab5-1           lab5-1.o    lab5-2.asm
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Горобцова Арина Романовна

```

Рис. 3.15: Исполнение файла

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2-1 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами `sprintLF` и `sprint`.

3.4 Выполнение заданий для самостоятельной работы

1. Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. 3.16).

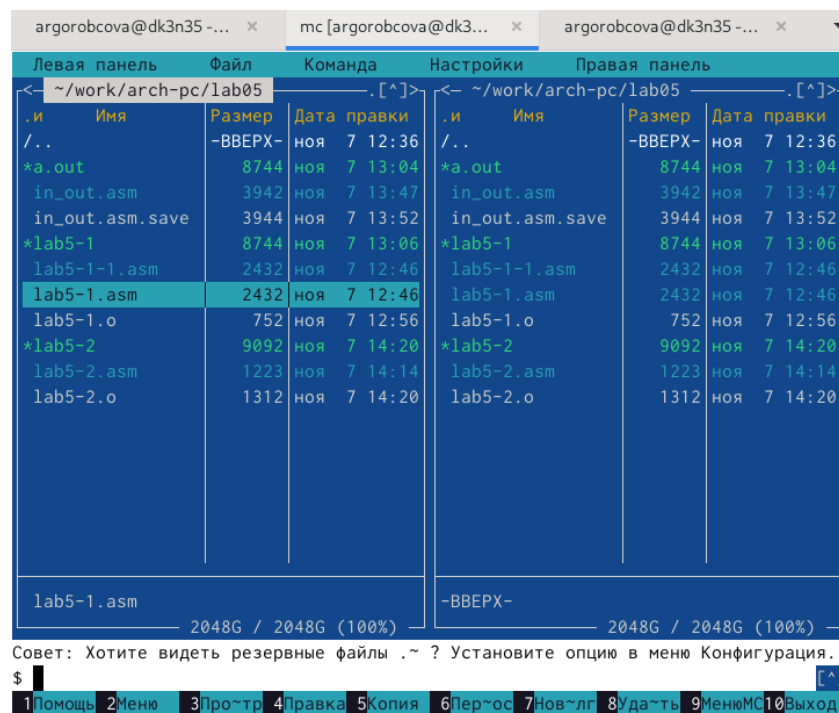


Рис. 3.16: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 3.17).

```
lab5-1-1.asm  [----] 20 L: [ 3+23 26/ 26] *(1521/1521b) <EOF>  [*][X]
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 ; Вызов ядра
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Перть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 3.17: Редактирование файла

2.Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 3.18).

```
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-1.asm
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-1-1
Введите строку:
Горобцова Арина Романовна
Горобцова Арина Романовна
argorobcova@dk3n35 ~/work/arch-pc/lab05 $
```

Рис. 3.18: Исполнение файла

Код программы из пункта 1:

```
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10
```

```

msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных 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 ; Вызов ядра

mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ; Вызов ядра

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

```

3.Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. 3.19).

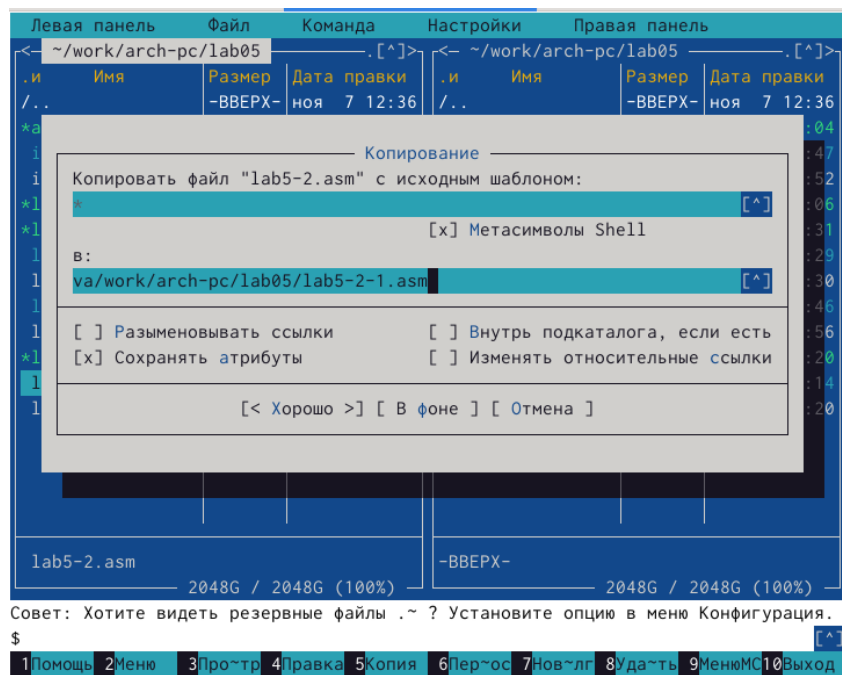
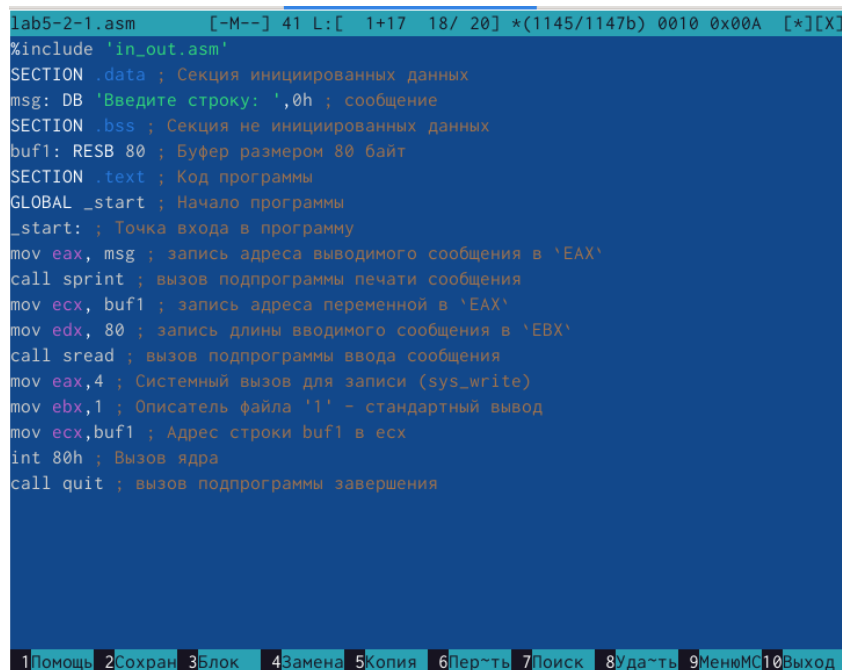


Рис. 3.19: Копирование файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 3.20).



```
lab5-2-1.asm [-M--] 41 L: [ 1+17 18/ 20] *(1145/1147b) 0010 0x00A [*][X]
#include 'in_out.asm'
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 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
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Описатель файла '1' - стандартный вывод
mov ecx, buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения
```

Рис. 3.20: Редактирование файла

4.Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. 3.21).

```
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2-1.asm
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
argorobcova@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-2-1
Введите строку: Горобцова Арина Романовна
Горобцова Арина Романовна
argorobcova@dk3n35 ~/work/arch-pc/lab05 $
```

Рис. 3.21: Исполнение файла

Код программы из пункта 3:

```
%include 'in_out.asm'
```

```
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 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
mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Описатель файла '1' - стандартный вывод
mov ecx, buf1 ; Адрес строки buf1 в ecx
int 80h ; Вызов ядра
call quit ; вызов подпрограммы завершения

4 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера `mov` и `int`.