Отчёт по лабораторной работе №5

Дисциплина: Архитектура компьютера

Зайцева Ульяна Владимировна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Начало работы с Midnight Commander.
2. Подключение внешнего файла in\_out.asm.
3. Задания для самостоятельной работы.

# 3 Теоретическое введение

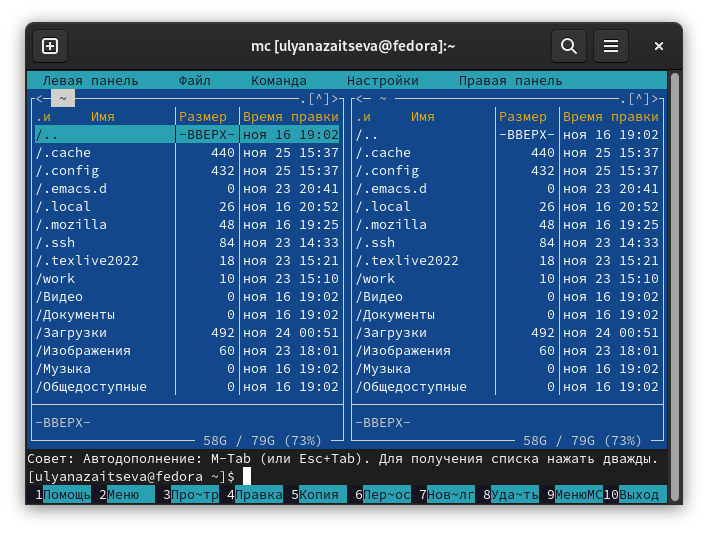
Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter.. В Midnight Commander используются функциональные клавиши F1 — F10 , к которым привязаны часто выполняемые операции.

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

# 4 Выполнение лабораторной работы

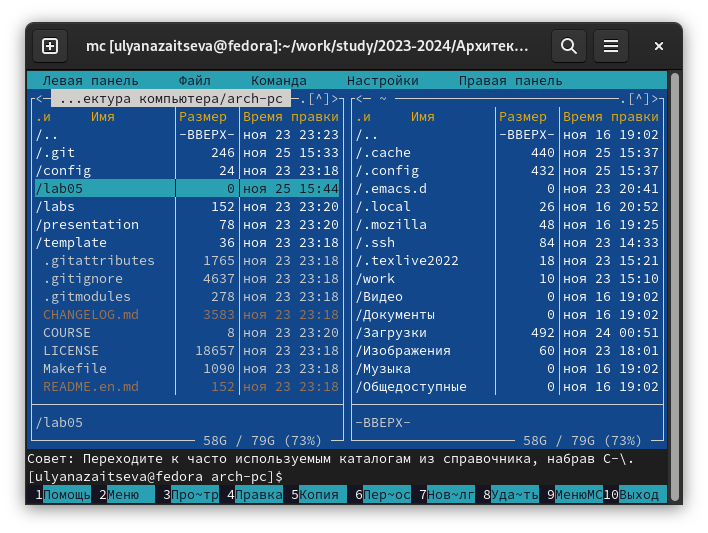
1. Начало работы с Midnight Commander.

Открываю Midnight Commander с помощью команды mc.(рис. ??).



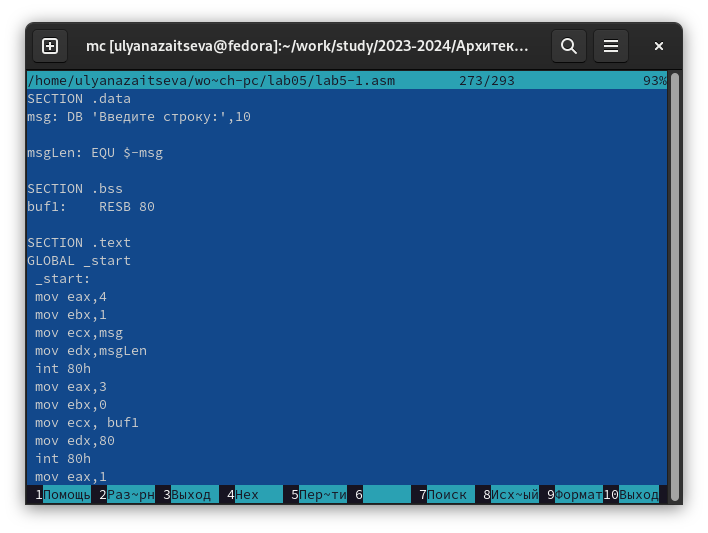
Окно mc

Пользуясь клавишами вверх, вниз и Enter перехожу в каталог, созданный при выполнении лабораторной работы №4, с помощью функциональной клавиши F7 создаю папку lab05 и перехожу в созданный каталог.(рис. ??).



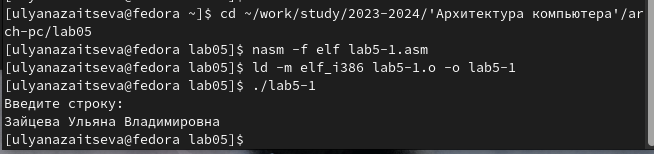
Новая папка

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm. С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе, ввожу текст программы из листинга 5.1, сохраняю изменения с помощью функциональной клавиши F2 и закрываю файл.(рис. ??)



Ввод текста листинга

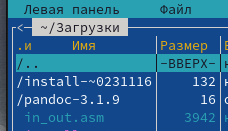
Компилирую текст программы lab5-1.asm в объектный файл, выполняю компоновку объектного файла и запускаю получившийся исполняемый файл, в строку ввожу моё ФИО.(рис. ??).



Компиляция программы

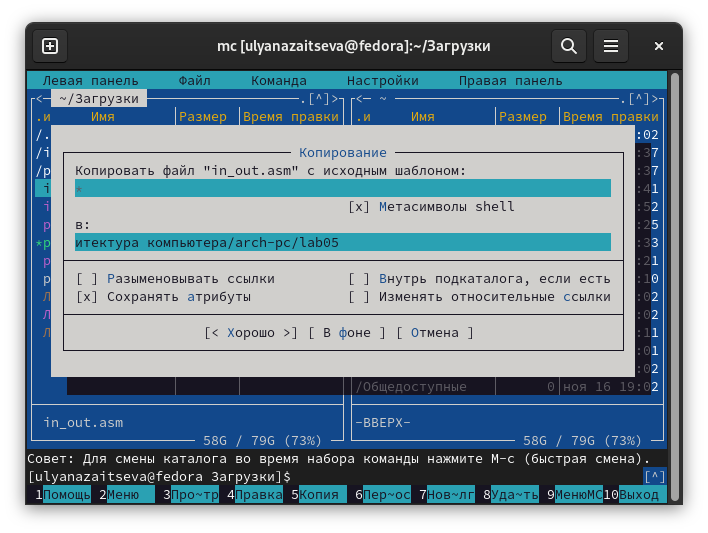
1. Подключение внешнего файла in\_out.asm.

Скачиваю файл in\_out.asm со страницы в ТУИС.(рис. ??).



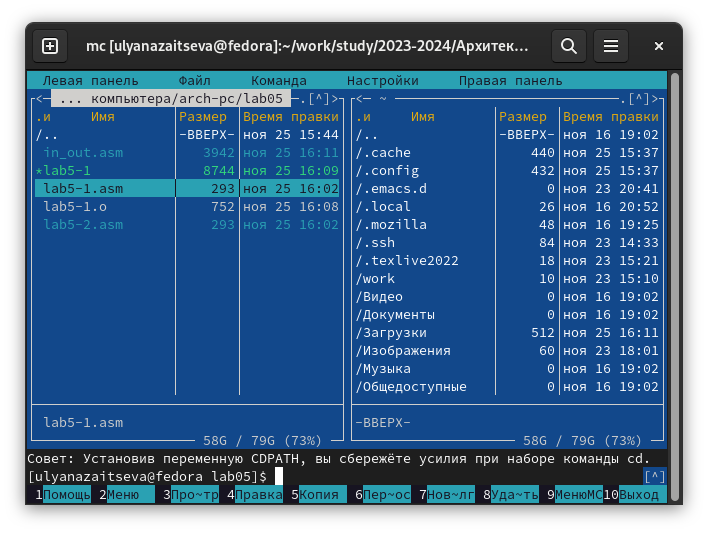
Скачанный файл in\_out.asm

Копирую файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5.(рис. ??).



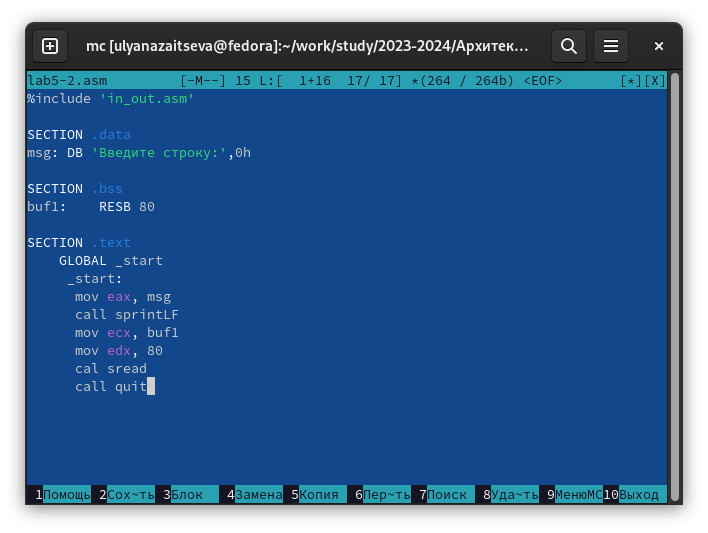
Копирую файл

С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm.(рис. ??)



Созданный файл в нужном каталоге

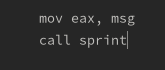
Исправляю текст программы в файле lab5-2.asm с использование подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2.(рис. ??)



Исправление программы

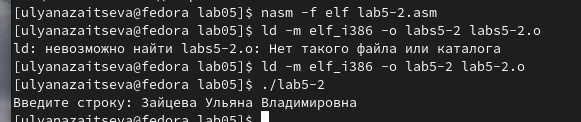
Не знаю где потерялся скрин:(( Я создаю исполняемый файл и проверяю его работу. Вывод получается как на (рис. ??).

Затем в файле lab5-2.asm заменяю подпрограмму sprintLF на sprint.(рис. ??)



Замена подпрограммы

Создаю исполняемый файл и проверяю его работу.(рис. ??).

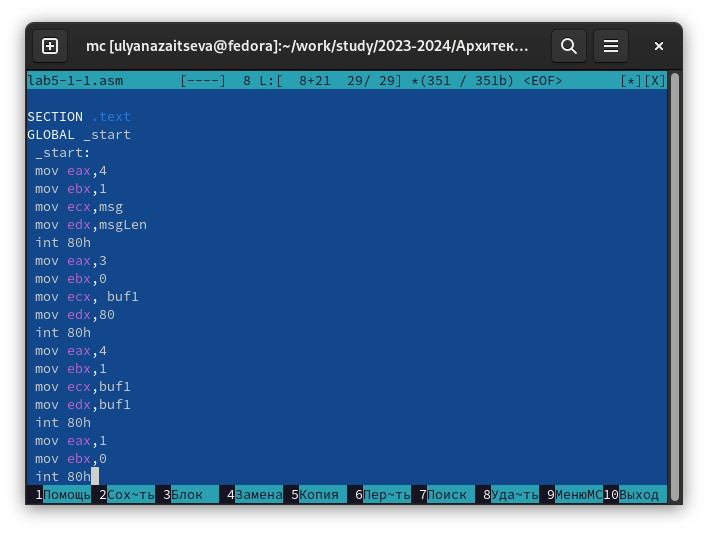


Проверка работы файла

Разница в том, что в первой программе ввод текста происходил с новой строки, в новой программе перехода на новую строку нет.

1. Задания для самостоятельной работы.

Создаю копию файла lab5-1.asm с именем lab5-1-1.asm. Вношу изменения в программу так, чтобы она выводила введённую строку на экран.(рис. ??)



Изменяю текст файла

Код:

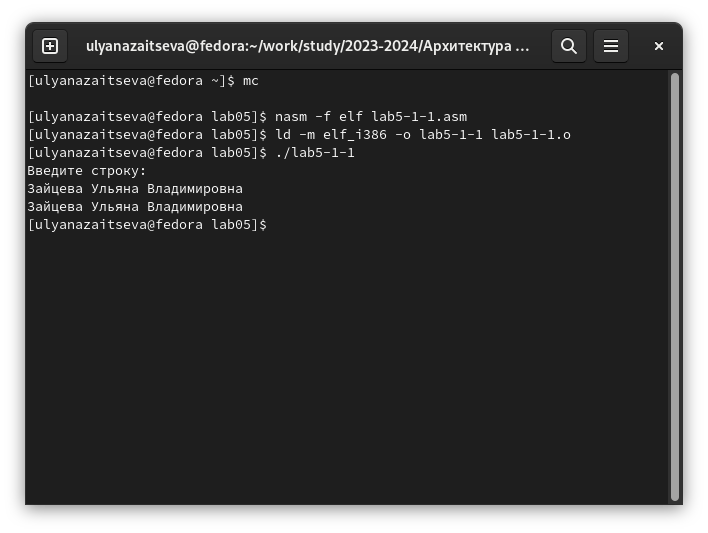
SECTION .data msg: DB ‘Введите строку:’,10

msgLen: EQU $-msg

SECTION .bss buf1: RESB 80

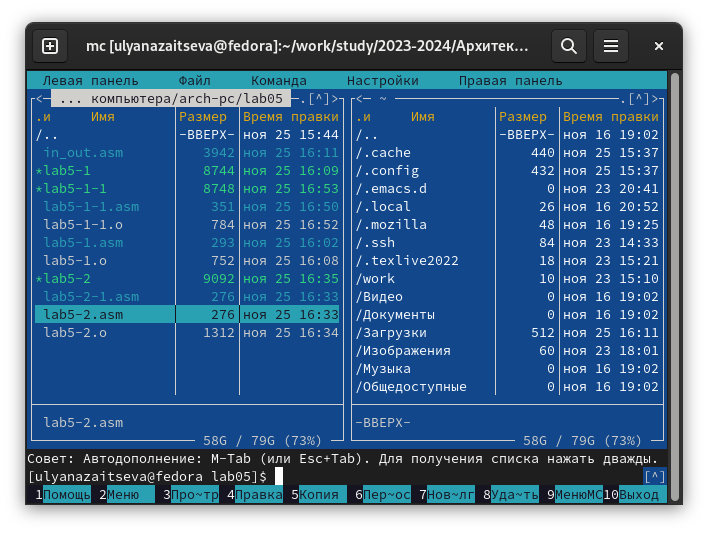
SECTION .text GLOBAL \_start \_start: mov eax,4 mov ebx,1 mov ecx,msg mov edx,msgLen int 80h mov eax,3 mov ebx,0 mov ecx, buf1 mov edx,80 int 80h mov eax,4 mov ebx,1 mov ecx,buf1 mov edx,buf1 int 80h mov eax,1 mov ebx,0 int 80h

Компилирую файл и проверяю его работу. Ввожу своё ФИО.(рис. ??)



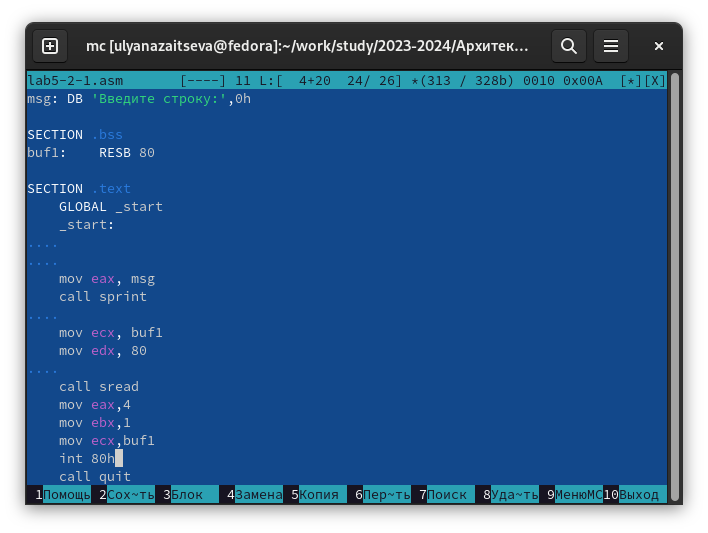
Компиляция файла

Создаю копию файла lab5-2.asm с названием lab5-2-1.asm(рис. ??)



Копирование файла

Вношу изменения в программу с использование подпрограмм из внешнего файла in\_out.asm так, чтобы она выводила введённую строку на экран.(рис. ??)



Изменяю текст файла

Код:

%include ‘in\_out.asm’

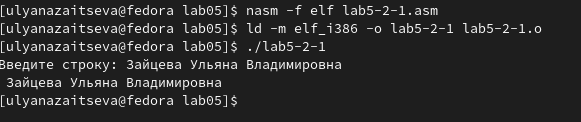
SECTION .data msg: DB ‘Введите строку:’,0h

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,4  
mov ebx,1  
mov ecx,buf1  
int 80h  
call quit

Создаю исполняемый файл и проверяю его работу.(рис. ??)



Проверка работы

Случайно поставила пробел, поэтому и он появился # Выводы

Во время выполнения лабораторной работы я приобрела навыки работы в Midnight Commander и освоила инструкции языка ассемблер mov и int.