Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Мусатова Екатерина Викторовна

Содержание

# 1 Цель работы

Цель данной лабораторной работы - приобретение практических навыков работы в Midnight Commander и освоение инструкций языка ассемблера mov и int.

# 2 Выполнение лабораторной работы

Открываю Midnight Commander (рис. [1](#fig%3A001)).

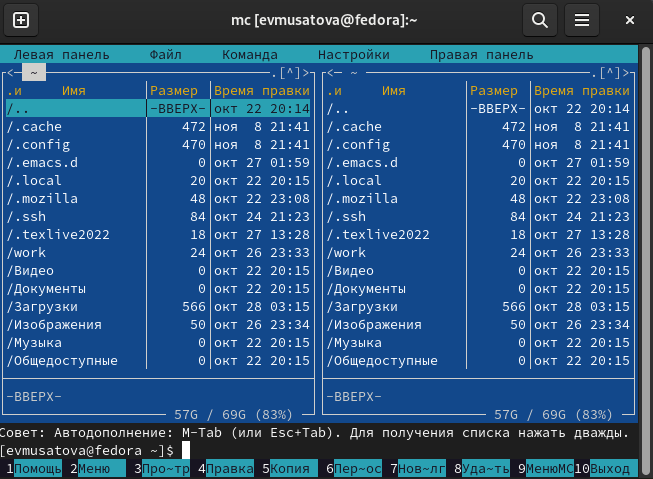


Figure 1: Midnight Commander

Пользуясь нужными клавишами, перехожу в каталог arch-pc (рис. [2](#fig%3A002)).

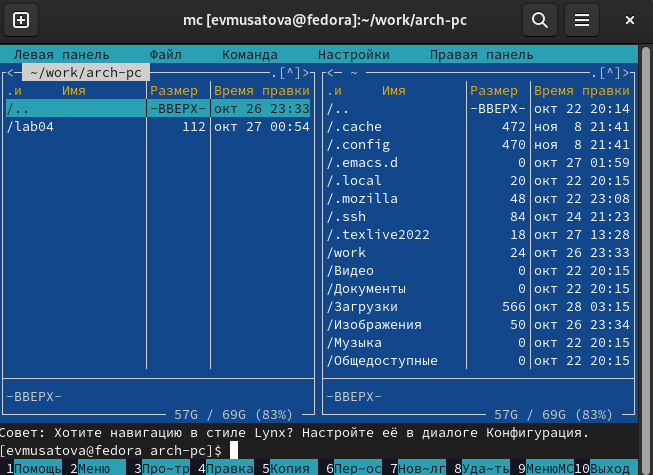


Figure 2: Переход в каталог

С помощью клавиши f7 создаю папку lab05 и прехожу в созданный каталог (рис. [3](#fig%3A003)).

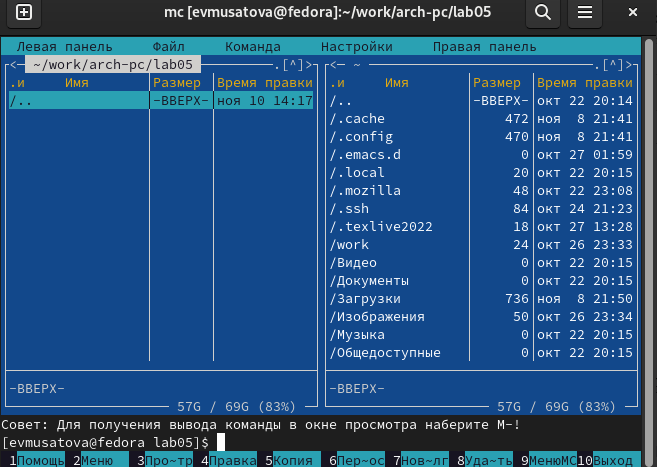


Figure 3: Создание папки

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm (рис. [4](#fig%3A004)).

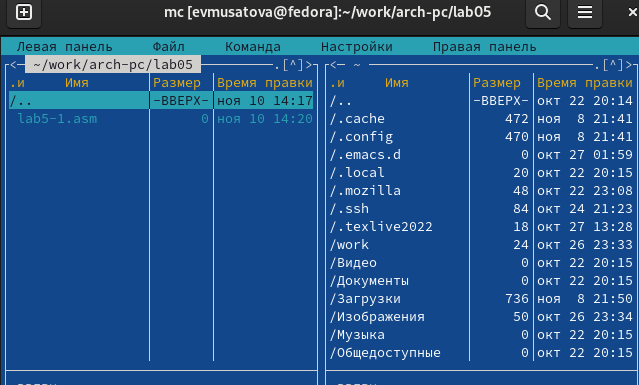


Figure 4: Создание файла

С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенном редакторе mcedit (рис. [5](#fig%3A005)).

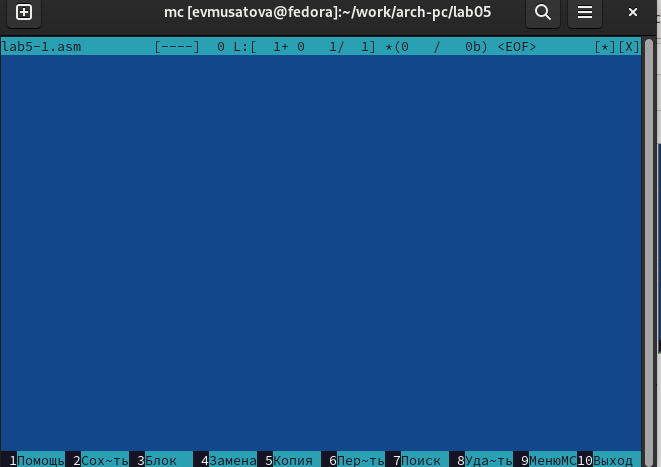


Figure 5: Открытие файла

Ввожу текст программы из листинга 5.1 (рис. [6](#fig%3A006)). Затем сохраняю изменения и закрываю файл.

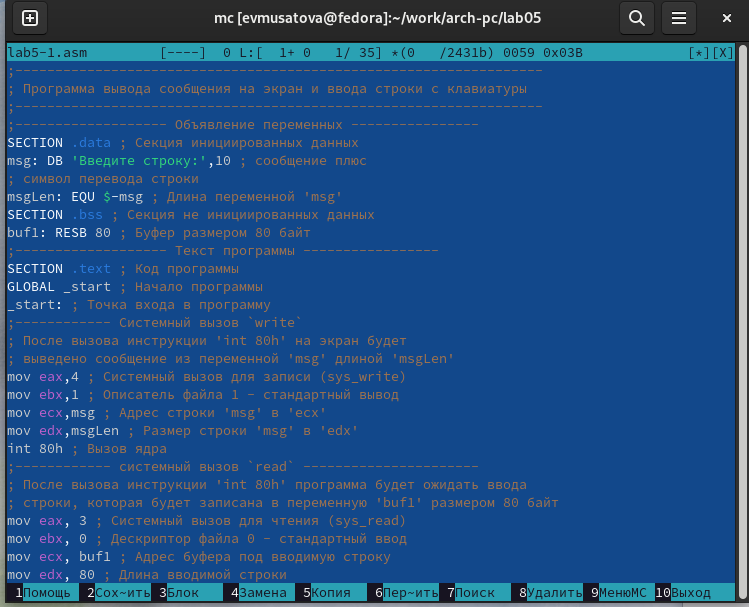


Figure 6: Ввод текста программы

С помощью функциональной клавиши f3 открываю файл, чтобы убедиться, что он содержит текст программы (рис. [7](#fig%3A007)).

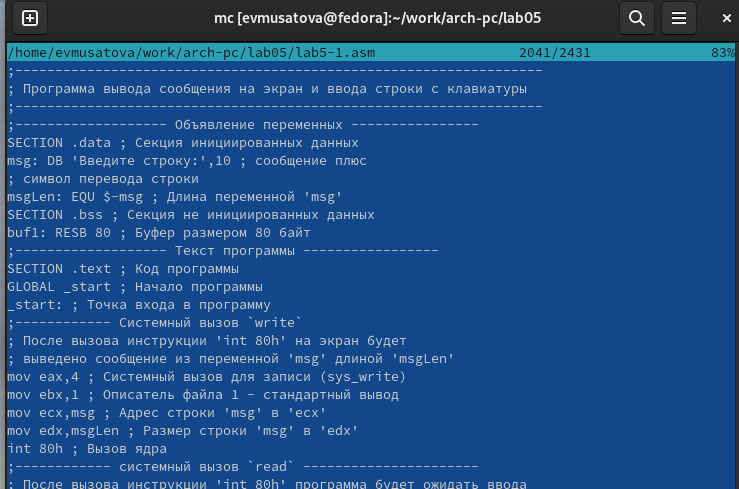


Figure 7: Проверка

Оттранслировала текст программы lab5-1.asm в объектный файл, выполнила компоновку объектного файла и запустила получившийся исполняемый файл, на запрос ввела свою фамилию и имя (рис. [8](#fig%3A008)).

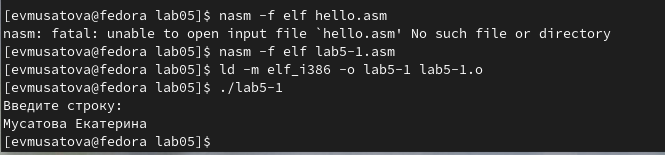


Figure 8: Запуск программы

Скачиваю файл in\_out.asm со страницы курса в ТУИС и помещаю его в тот же каталог, где находится файл с программой. Затем с помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. (рис. [9](#fig%3A009)).

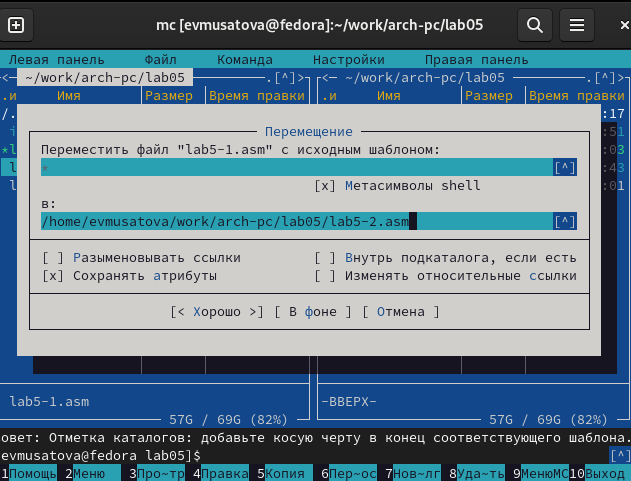


Figure 9: Создание файла lab5-2.asm

Исправляю текст программы в файле lab5-2.asm с использование подпрограмм из внешнего файла in\_out.asm в соответствии с листингом 5.2 (рис. [10](#fig%3A010)).

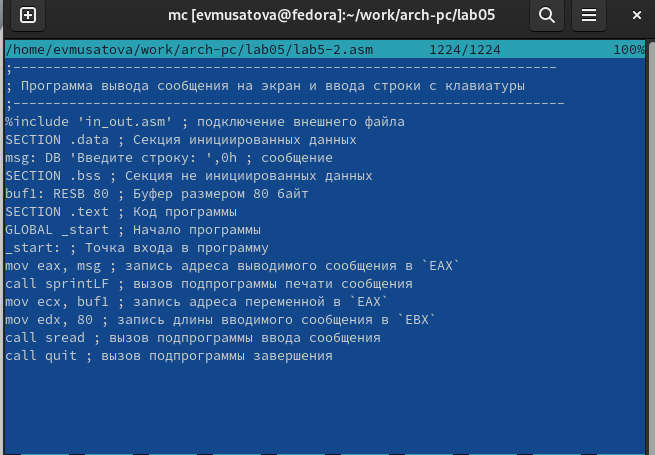


Figure 10: Исправление текста программы

Создаю исполняемый файл и проверяю работу программы (рис. [11](#fig%3A011)).

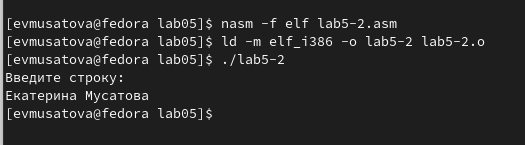


Figure 11: Проверка программы

В файле lab5-2.asm заменяю подпрограмму sprintLF на sprint. Создаю исполняемый файл и проверяю его работу (рис. [12](#fig%3A012)). Разница заключается в том, что теперь ввод производится на той же строке, что и вывод, убран символ перевода строки после вывода.

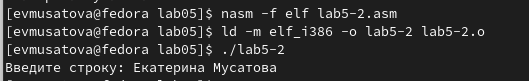


Figure 12: Проверка программы после замены

# 3 Выполнение заданий для самостоятельной работы

1. Копирую файл lab5-1.asm с именем lab5-3.asm (рис. [[13](#fig%3A013)]).

Figure 13: Копирование файла

Figure 13: Копирование файла

Изменяю код программы, добавляя вывод введенной строки (рис. [[14](#fig%3A014)]).

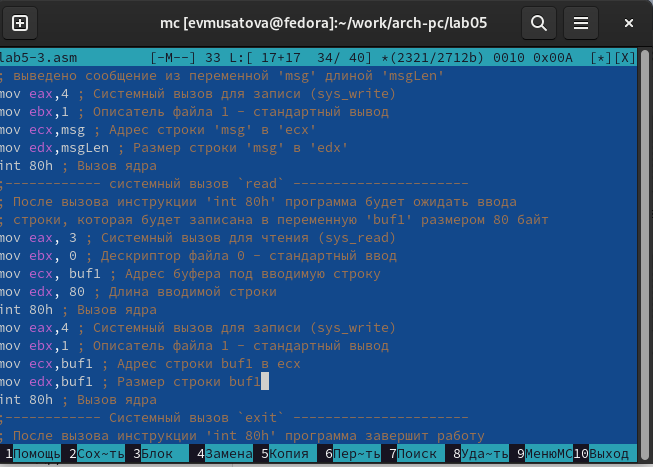


Figure 14: Изменение программы

Создаю объектный файл lab5-3.o и компоную его в исполняемый файл (рис. [[15](#fig%3A015)]).

Figure 15: Создание исполняемого файла

Figure 15: Создание исполняемого файла

Запускаю программу и ввожу свуою фамилию для проверки (рис. [[16](#fig%3A016)]).

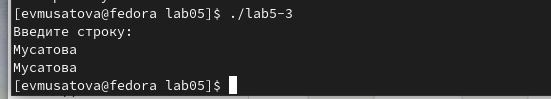


Figure 16: Проверка

Программа из пункта 1:

;—————————————————————— ; 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 ;———— Cистемный 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 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 ;———— 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

1. Копирую файл lab5-2.asm с именем lab5-4.asm (рис. [[17](#fig%3A017)]).

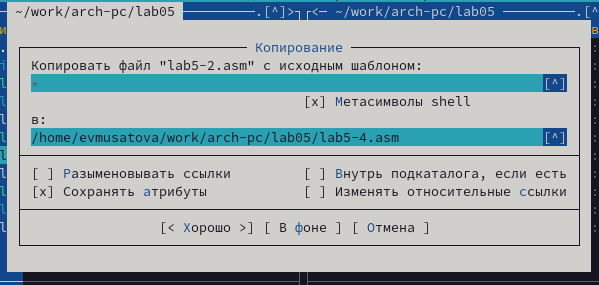


Figure 17: Копирование файла

Изменяю код программы, добавляя вывод введенной строки (рис. [[18](#fig%3A018)]).

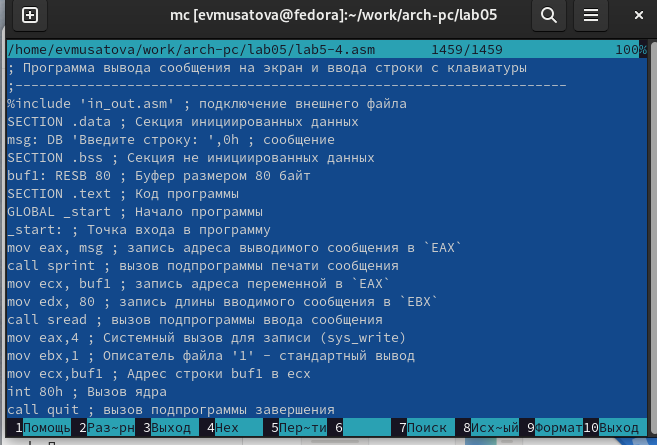


Figure 18: Изменение файла

Создаю исполняемый файл и запускаю его (рис. [[19](#fig%3A019)]).

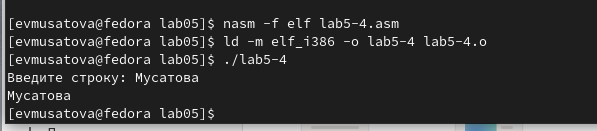


Figure 19: Запуск программы

Программа из пункта 2:

;——————————————————————– ; Программа вывода сообщения на экран и ввода строки с клавиатуры ;——————————————————————— %include ‘in\_out.asm’ ; подключение внешнего файла SECTION .data ; Секция инициированных данных msg: DB ‘Введите строку:’,0h ; сообщение 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, 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,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

Я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера mov и int.