Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Сидельников Андрей Владимирович

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Открываю Midnight Commander, введя в терминал mc (рис. [1](#fig:001)).

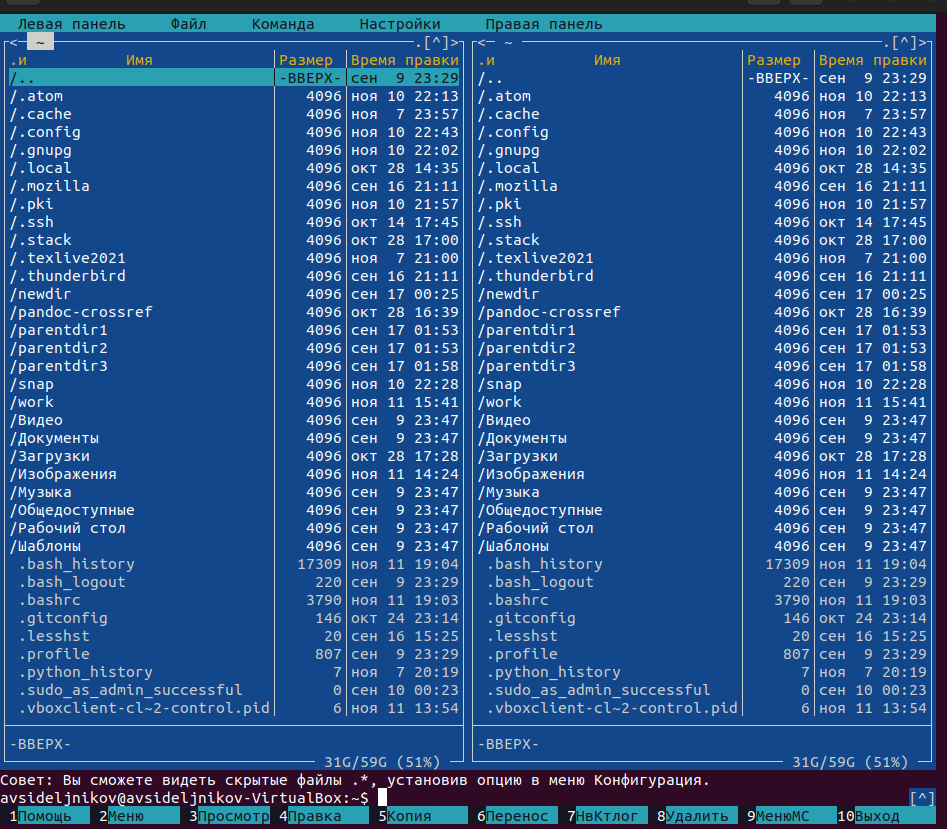
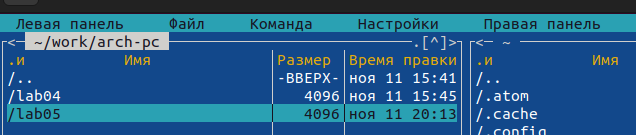


Figure 1: Открытый mc

Перехожу в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4 и создаю там каталог lab05 с помощью клавиши F7 (рис. [2](#fig:002)).



Создание каталога

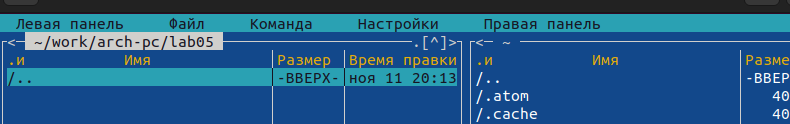


Figure 2: Переход в каталог lab05

В строке ввода прописываю команду touch lab5-1.asm, чтобы создать файл, в котором буду работать (рис. [3](#fig:003)).

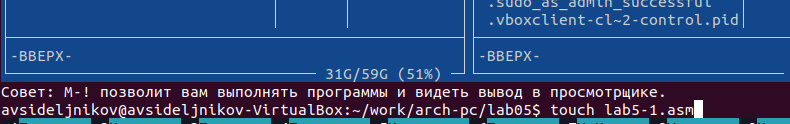


Figure 3: Создание файла

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano (рис. [4](#fig:004)).

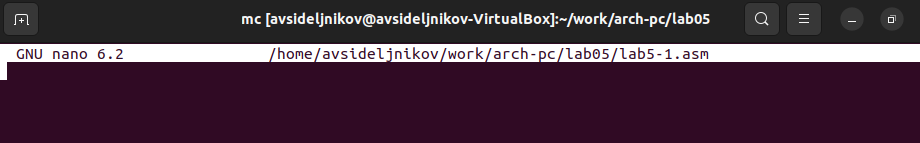


Figure 4: Открытие файла для редактирования

Ввожу в файл код программы для запроса строки у пользователя (рис. [5](#fig:005)).

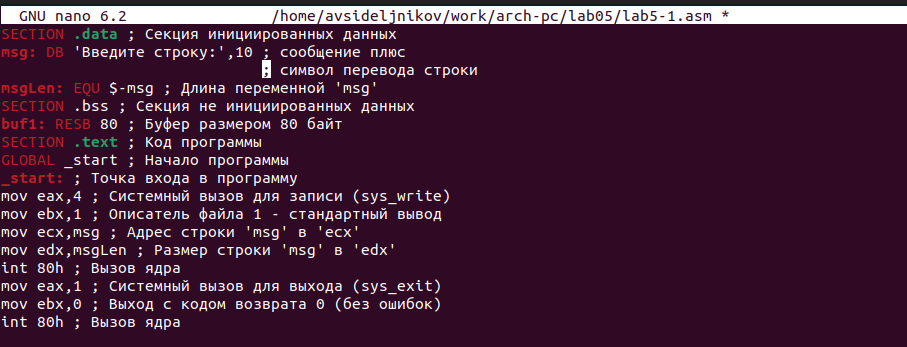


Figure 5: Редактирование файла

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. [6](#fig:006)).

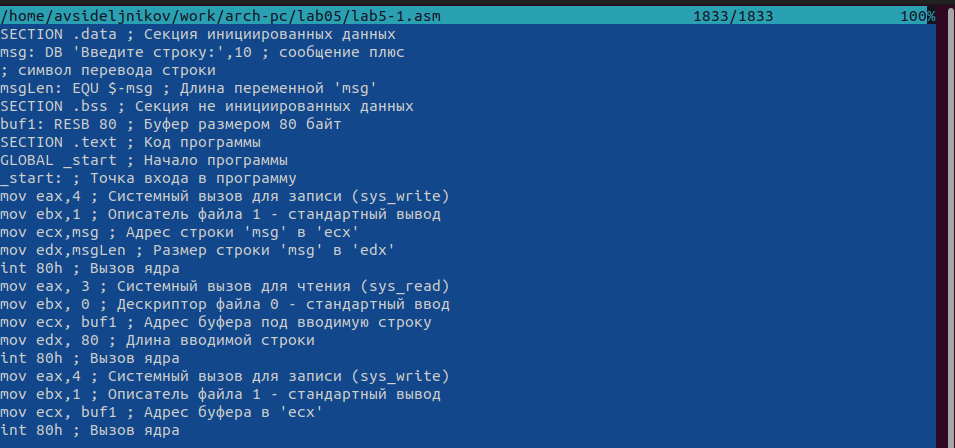


Figure 6: Открытие файла для просмотра

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 (рис. [7](#fig:007)).

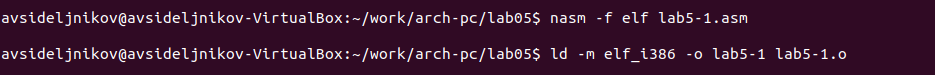


Figure 7: Компиляция файла и передача на обработку компоновщику

Запускаю исполняемый файл и ввожу свои ФИО, на этом программа заканчивает свою работу (рис. [8](#fig:008))

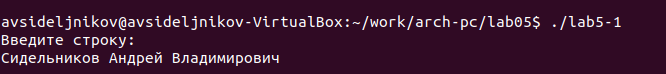


Figure 8: Исполнение файла

## 2.1 Подключение внешнего файла in\_out.asm

Скачиваю файл in\_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” (рис. [9](#fig:009))

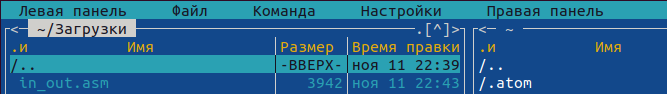


Figure 9: Скачанный файл

С помощью функциональной клавиши F5 копирую файл in\_out.asm из каталога Загрузки в созданный каталог lab05 (рис. [10](#fig:010)).

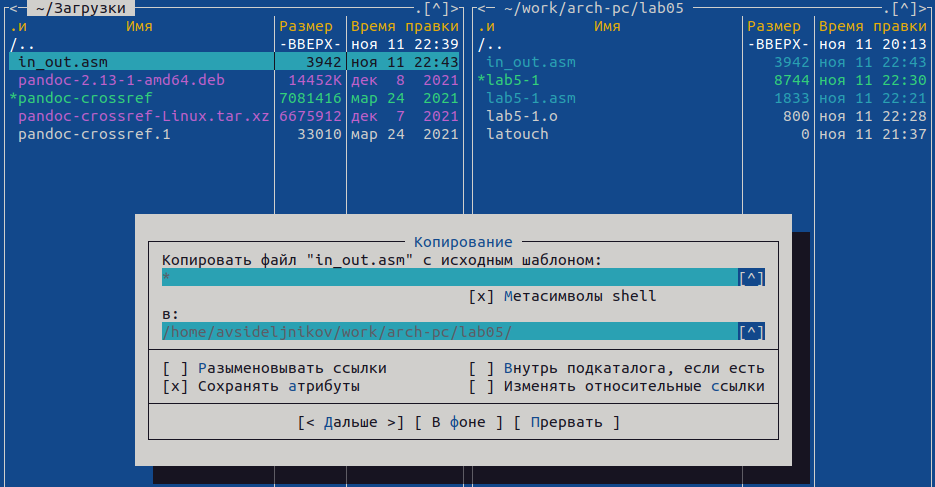


Figure 10: Копирование файла

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. [11](#fig:011)).

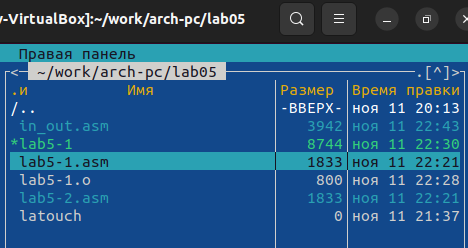


Figure 11: Копирование файла

Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm (рис. [12](#fig:012)).

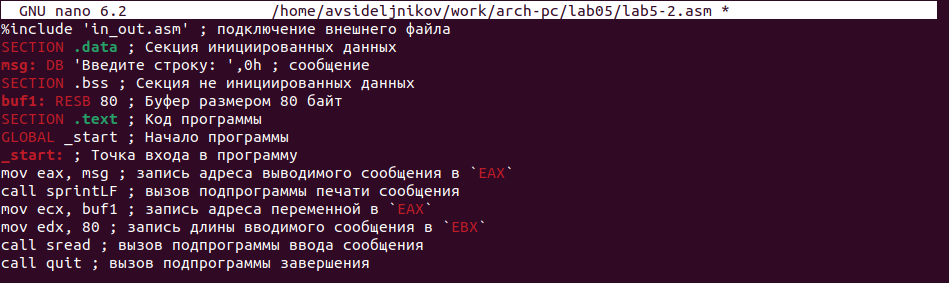


Figure 12: Редактирование файла

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. [13](#fig:013)).

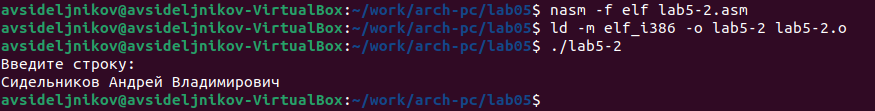


Figure 13: Исполнение файла

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. [14](#fig:014)).

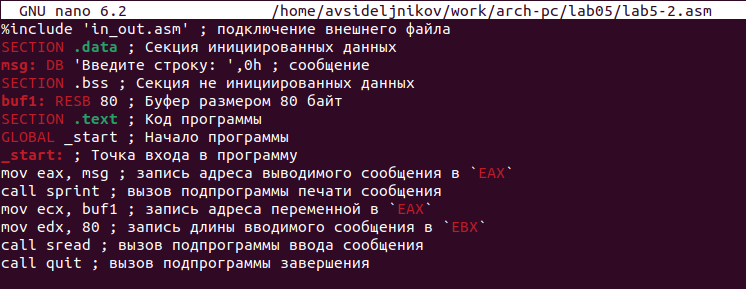


Figure 14: Отредактированный файл

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. [15](#fig:015))

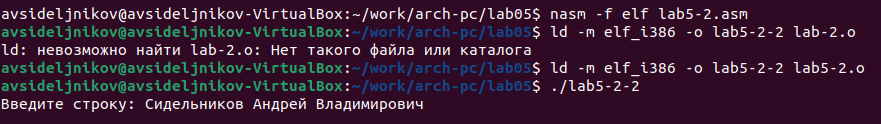


Figure 15: Исполнение файла

Разница между первым исполняемым файлом lab5-2 и вторым lab5-2-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку.

# 3 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью функциональной клавиши F5 (рис. [16](#fig:016)).

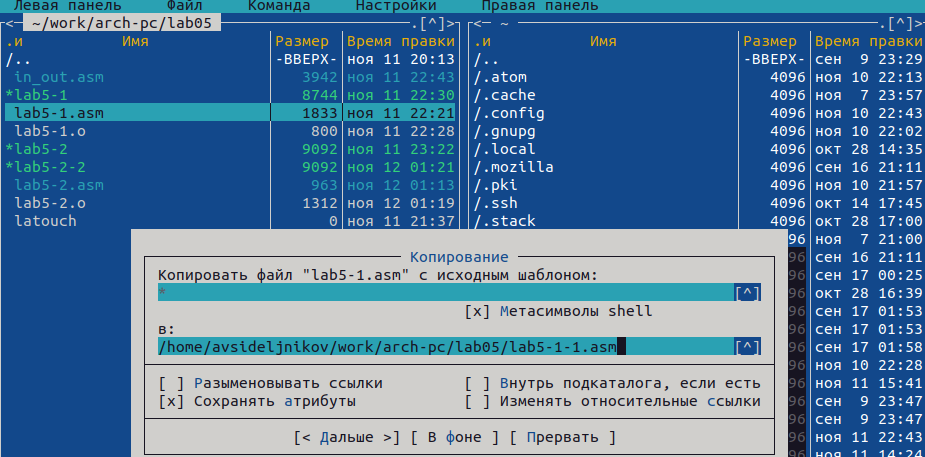


Figure 16: Копирование файла

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. [17](#fig:017)).

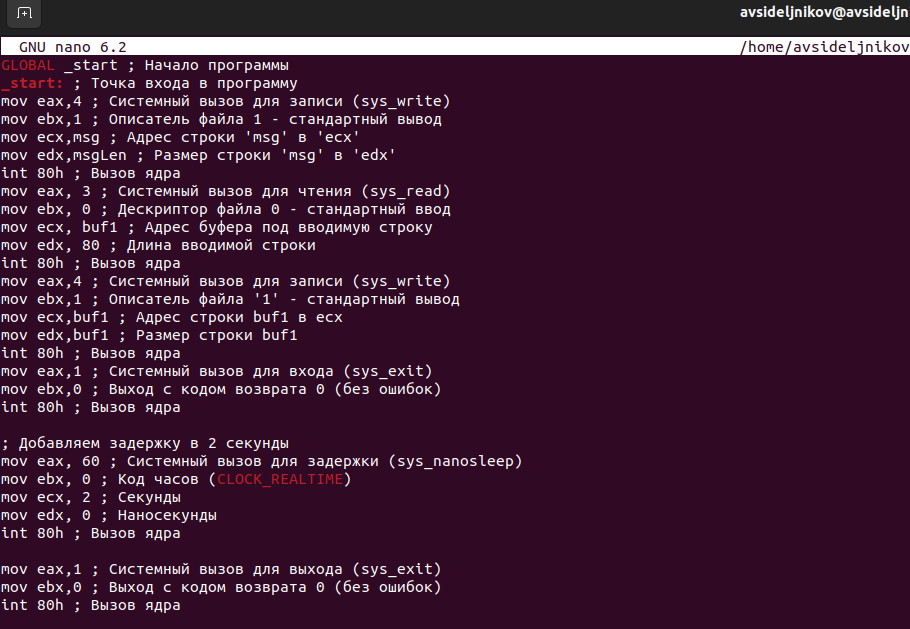


Figure 17: Редактирование файла

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. [18](#fig:018)).

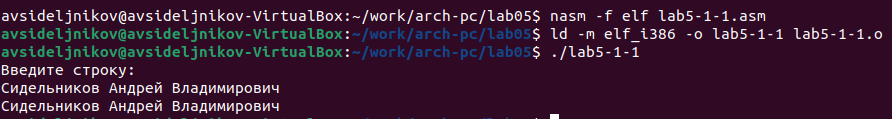


Figure 18: Исполнение файла

SECTION .data ; Секция инициированных 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

Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. [19](#fig:019)).

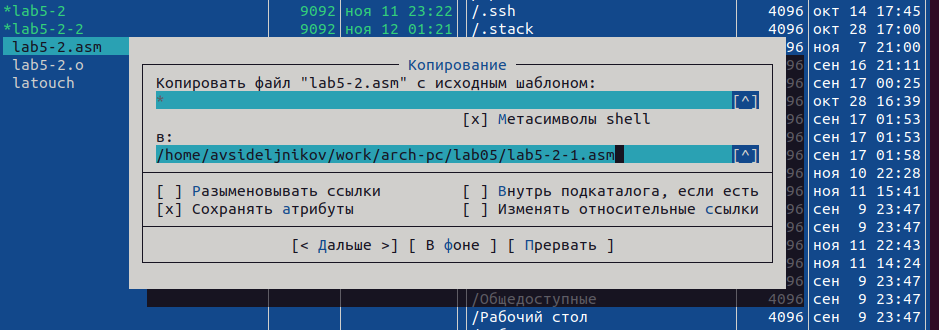


Figure 19: Копирование файла

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. [20](#fig:020)).

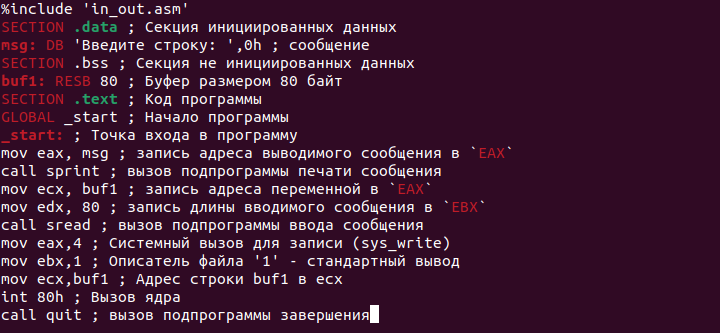


Figure 20: Редактирование файла

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. [21](#fig:021)).

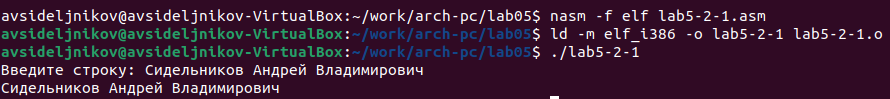


Figure 21: Исполнение файла

%include 'in\_out.asm' 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 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.