Отчёт по лабораторной работе №5

Дисциплина: Архитектура Компьютера

Иванов Сергей Владимирович

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Откроем Midnight Commander и перейдем в каталог созданный при выполнении лабораторной работы №4.(Рис. [1](#fig:001))

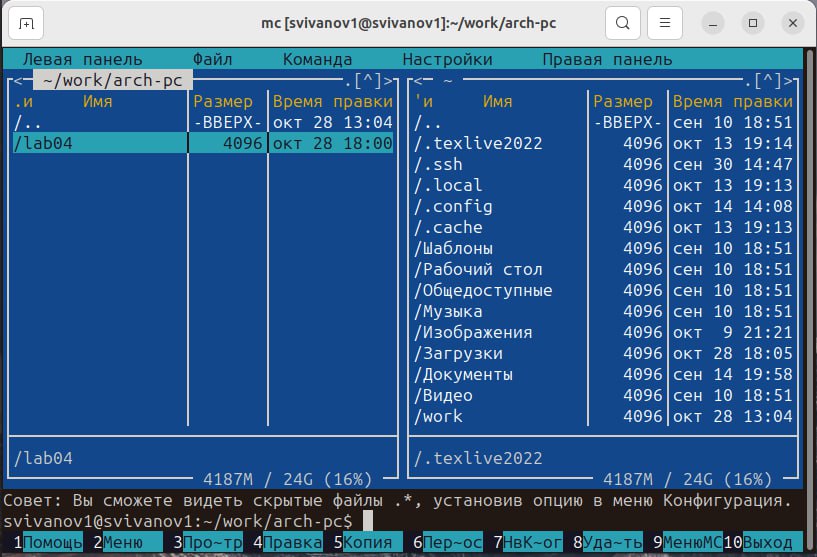


Figure 1: Midnight Kommander

Создадим папку lab05, а в ней файл lab5-1.asm (Рис. [2](#fig:002))

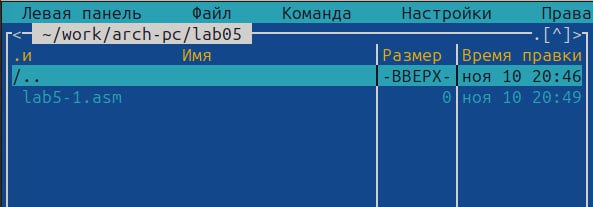


Figure 2: Создание папки и файла lab5-1.asm

Откроем файл lab5-1.asm и отредактируем его. (Рис. [3](#fig:003))

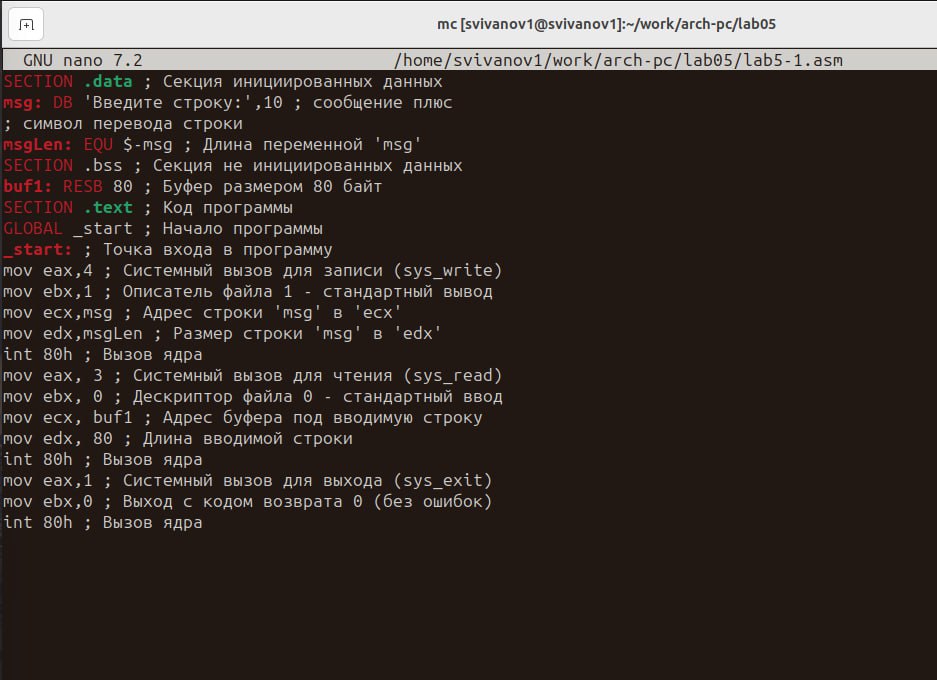


Figure 3: Редактируем lab5-1.asm

Оттранслируем текст программы в объектный файл. Выполним компоновку объектного файла и запустим исполняемый файл. На запрос вводим ФИО. (Рис. [4](#fig:004))

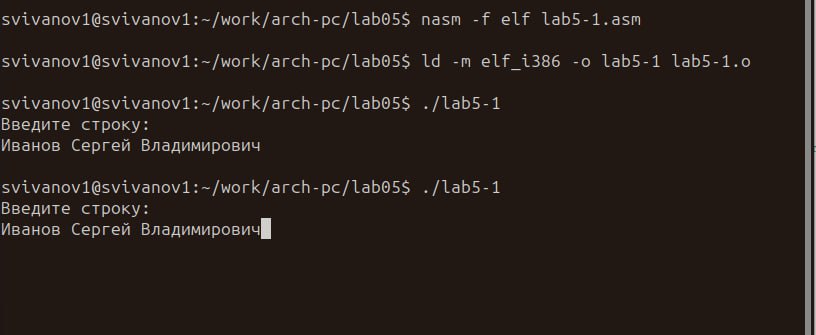


Figure 4: Трансляция и проверка исполняемого файла

Скачаем файл in\_out.asm со страницы курса в ТУИС. (Рис. [5](#fig:005))

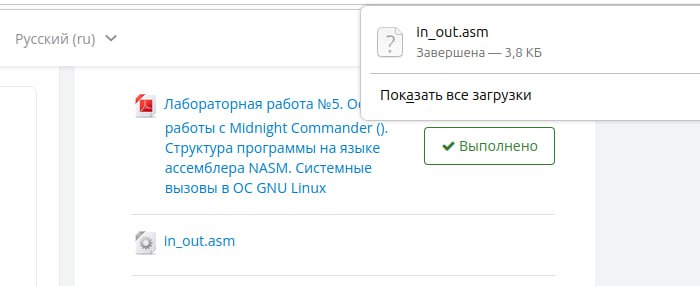


Figure 5: Скачиваем файл

Скопируем файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши f5. (Рис. [6](#fig:006))

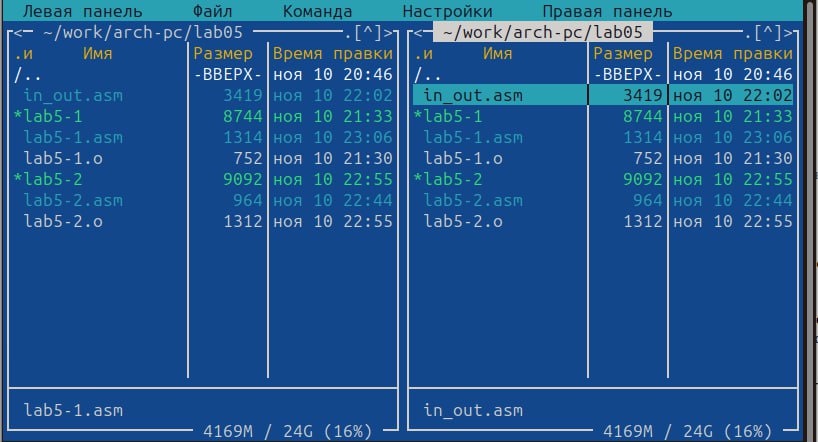


Figure 6: Копирование файла в каталог

Создадим копию файла lab5-1.asm с именем lab5-2.asm (Рис. [7](#fig:007))

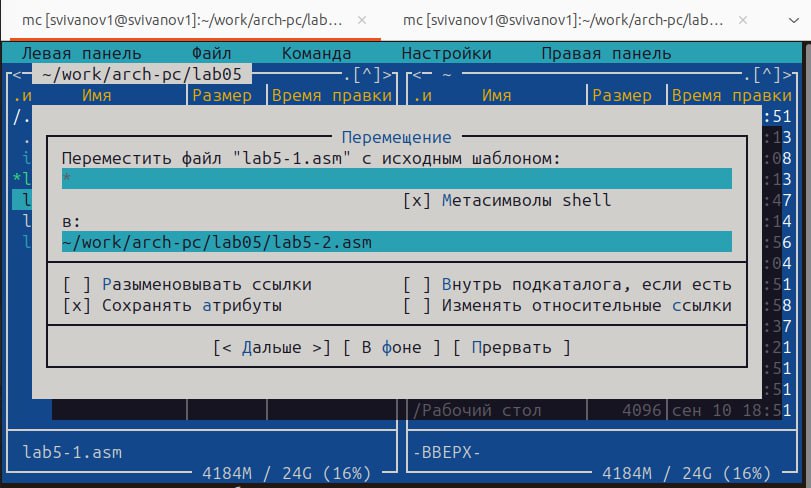


Figure 7: Создание копии файла

Исправим текст программы lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (Рис. [8](#fig:008))

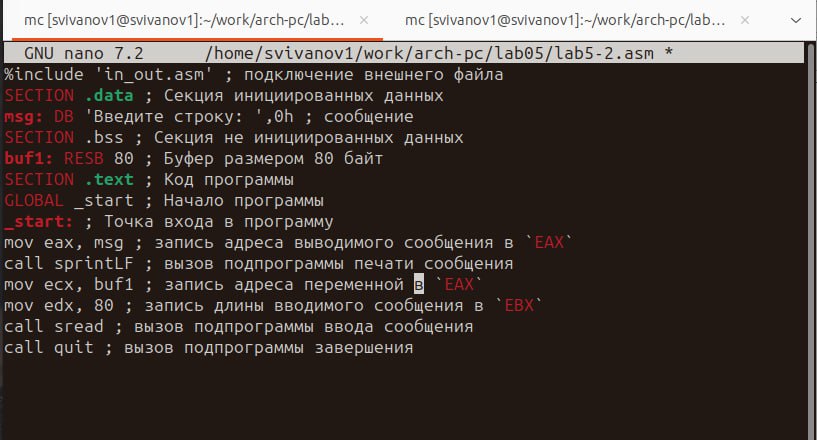


Figure 8: Редактируем программу

Создадим исполняемый файл и проверим его работу. Все работает правильно. (Рис. [9](#fig:009))

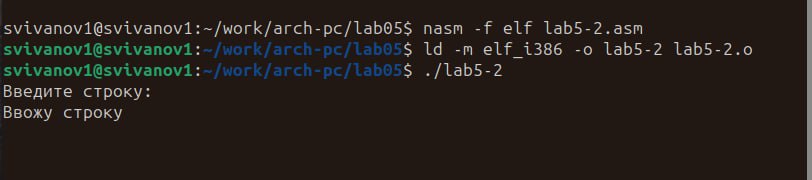


Figure 9: Создание и проверка иссполняемого файла

В файле lab5-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу. Разница в том, что теперь строка на ввод не переносится, как это было в прошлом варианте. (Рис. [10](#fig:010))

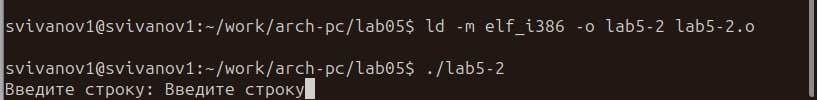


Figure 10: Замена подпрограммы

Создадим копию файла lab5-1.asm с именем lab5-3.asm. Внесем изменения в программу чтобы она выводила введённую строку на экран.(без использования внешнего файла in\_out.asm) (Рис. [11](#fig:011))

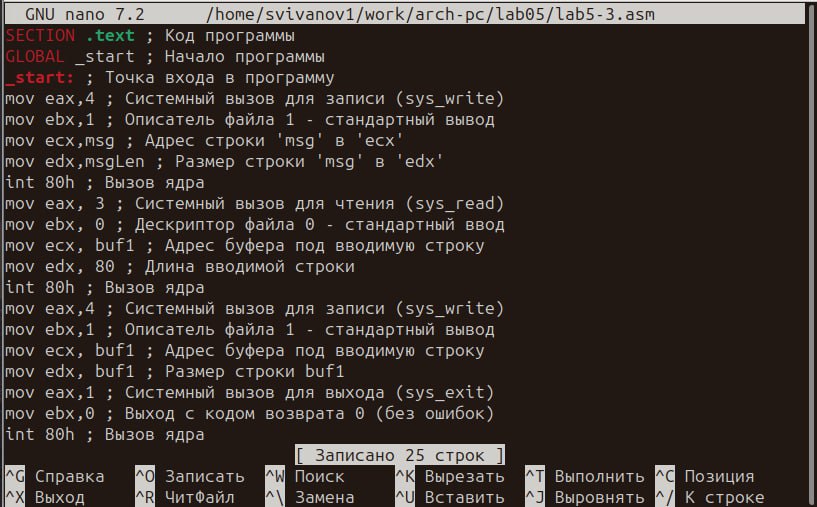


Figure 11: Копируем и редактируем lab5-3.asm

Получим исполняемый файл и убедимся что он работает правильно. На приглашение ввести строку вводим свою фамилию. (Рис. [12](#fig:012))

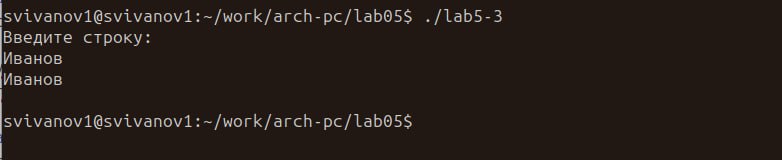


Figure 12: Проверяем программу

**Код программы из пункта 1 самостоятельной работы:**

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 буфера под вводимую строку  
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

Создадим копию файла lab5-1.asm с именем lab5-4.asm. Внесем изменения в программу чтобы она выводила введённую строку на экран.(с использованием внешнего файла in\_out.asm) (Рис. [13](#fig:013))

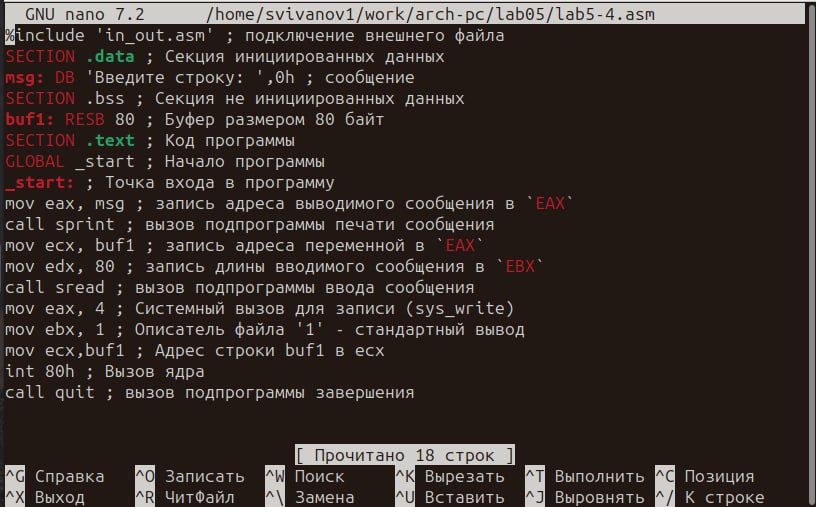


Figure 13: Копируем и редактируем lab5-4.asm

Получим исполняемый файл и убедимся что он работает правильно. (Рис. [14](#fig:014))

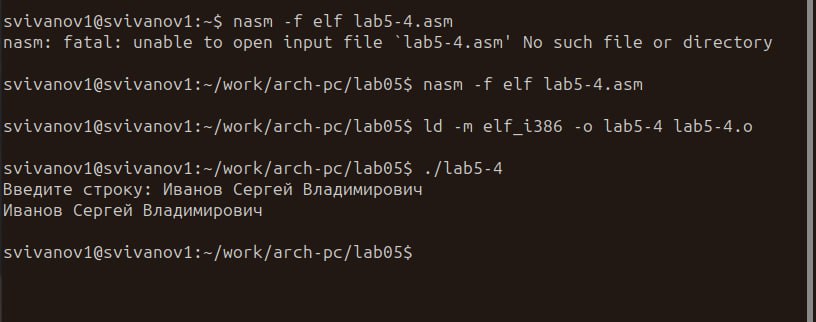


Figure 14: Проверяем программу

**Код программы из пункта 3 самостоятельной работы:**

%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 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

# 3 Выводы

В результате выполнения лабораторной работы мы приобрели практические навыки работы в Midnight Commander и освоенили инструкции языка ассемблера mov и int.