Отчёта по лабораторной работе 5

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM

Тукаев Тимур Ильшатович НММбд-03-23

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

1. Открыл Midnight Commander
2. Перешел в каталог ~/work/arch-pc
3. Создал каталог lab05

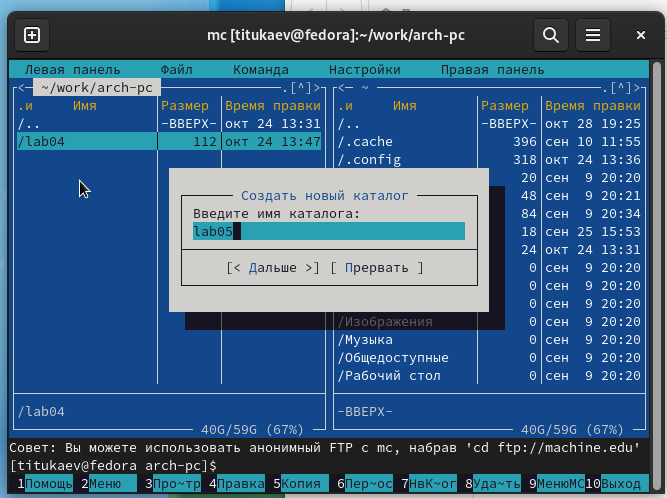


Figure 1: Создание каталога

1. Создал файл lab05-1.asm

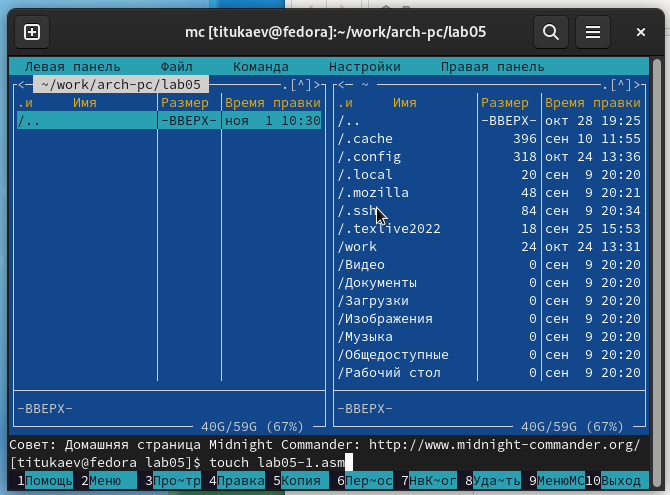


Figure 2: Создание файла lab05-1.asm

1. Открыл файл на редактирование
2. Написал код

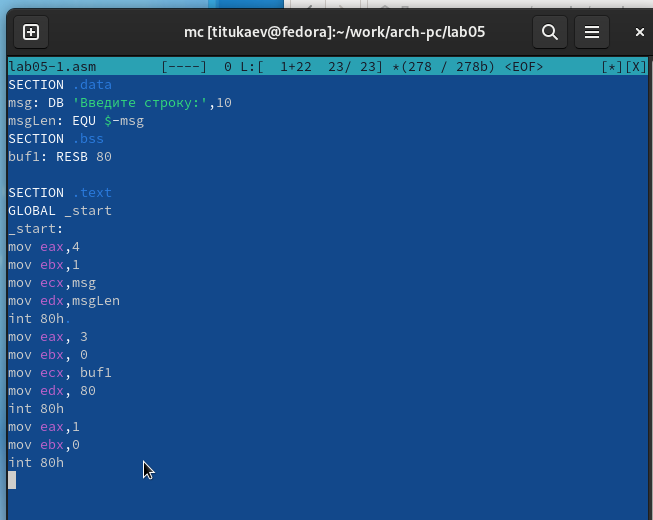


Figure 3: Программа в файле lab05-1.asm

Также размещаю код программы в отчете.

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,1  
mov ebx,0  
int 80h

1. Открыл файл на просмотр и убелился, что он содержит набранный код.

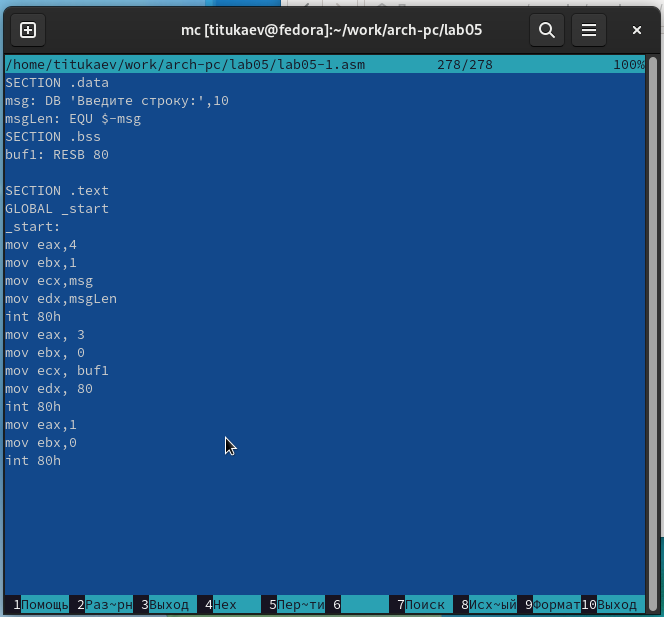


Figure 4: Просмотр файла lab05-1.asm

1. Получил исполняемый файл программы и провреил ее работу.

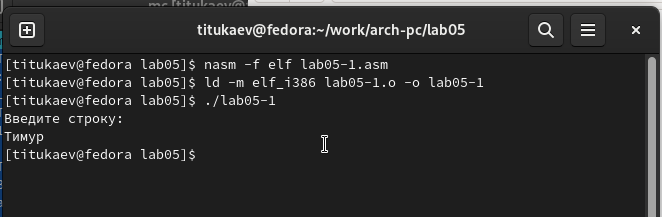


Figure 5: Запуск программы lab05-1.asm

1. Скачал файл in\_out.asm.
2. Добавил файл in\_out.asm в рабочий каталог.
3. Скопировал lab05-1.asm в lab05-2.asm.

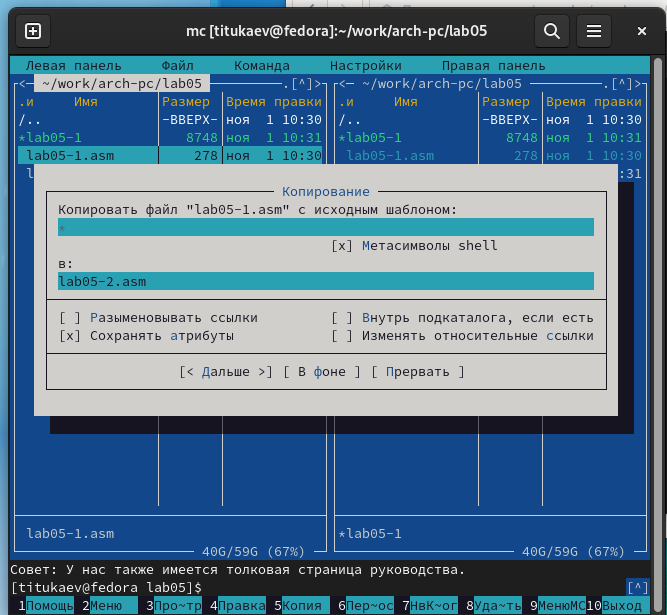


Figure 6: Копирование файла

1. Написал код программы lab05-2.asm. Скомпилировал программу и провреили запуск.

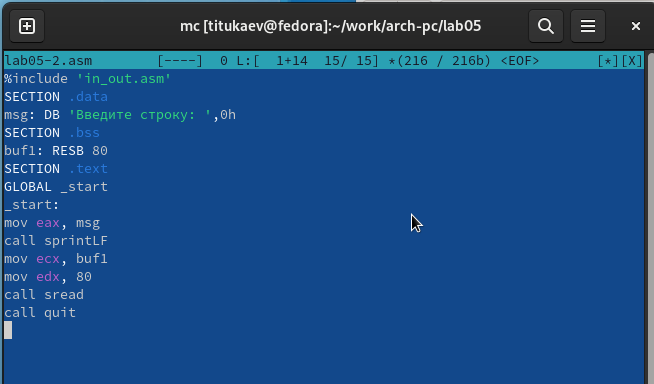


Figure 7: Программа в файле lab05-2.asm

Также размещаю код программы в отчете.

%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 sprintLF  
mov ecx, buf1  
mov edx, 80  
call sread  
call quit

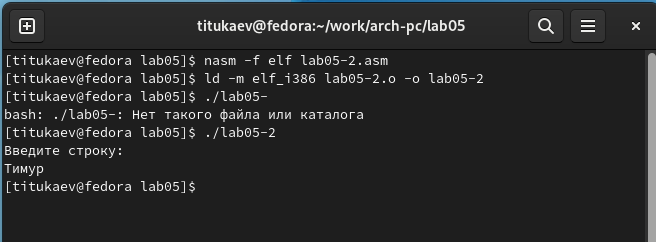


Figure 8: Запуск программы lab05-2.asm

1. В файле lab5-2.asm заменил подпрограмму sprintLF на sprint. Заново собрал исполняеый файл. Теперь после вывода строки она не завершается символом перехода на новую строку.

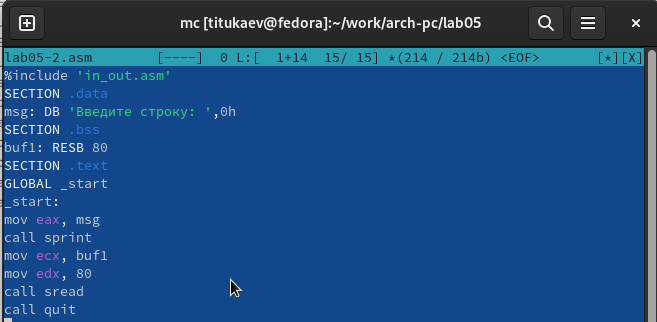


Figure 9: Программа в файле lab05-2.asm

Также размещаю код программы в отчете.

%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  
call quit

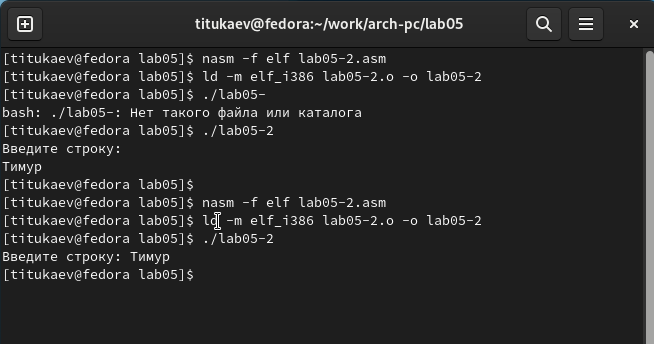


Figure 10: Запуск программы lab05-2.asm

1. Скопировал программу lab05-1.asm и измении код, чтобы вывести приглашение типа “Введите строку:”, ввести строку с клавиатуры, вывести введённую строку на экран.

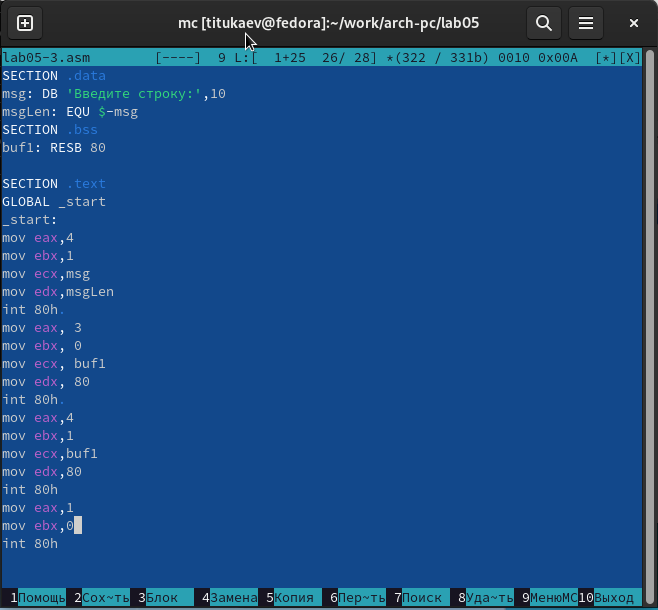


Figure 11: Программа в файле lab05-3.asm

Также размещаю код программы в отчете.

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,80  
int 80h  
mov eax,1  
mov ebx,0  
int 80h

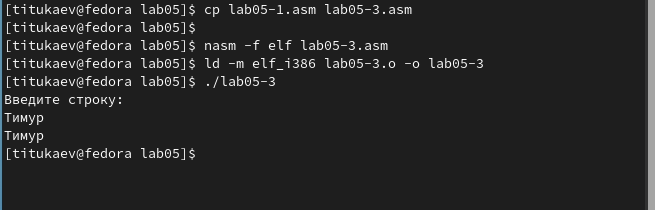


Figure 12: Запуск программы lab05-3.asm

1. Скопировал программу lab05-2.asm и изменил код, чтобы вывести приглашение типа “Введите строку:”, ввести строку с клавиатуры, вывести введённую строку на экран.

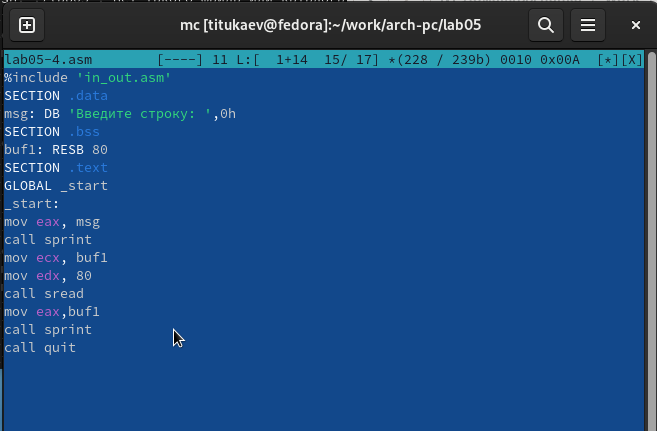


Figure 13: Программа в файле lab05-4.asm

Также размещаю код программы в отчете.

%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,buf1  
call sprint  
call quit

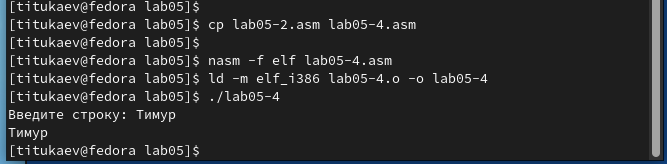


Figure 14: Запуск программы lab05-4.asm

Отличие этих двух реализаций в том, что файл in\_out.asm содержит уже готовые подпрограммы для обеспечения ввода/вывода. Таким образом, нам остается только разместить данные в нужных регистрах и вызвать желаемую подпрограмму с помощью call.

# 3 Выводы

Научились писать базовые ассемблерные программы. Освоили ассемблерные инструкции mov и int.