Отчёта по лабораторной работе 5

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

Субанов Юсуф Жура угли НПМбв-01-21

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. Откроем Midnight Commander
2. Перейдем в каталог ~/work/arch-pc
3. Создадим каталог lab05

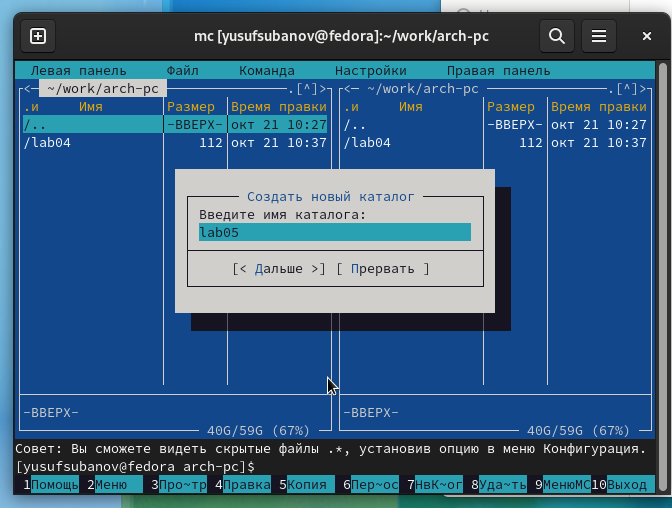


Figure 1: Создание каталога

1. Создадим файл lab05-1.asm

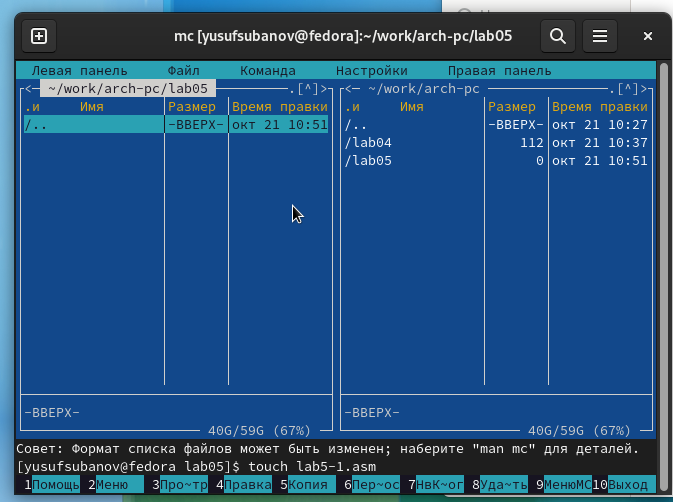


Figure 2: Создание файла lab05-1.asm

1. Откроем файл на редактирование
2. Напишем код

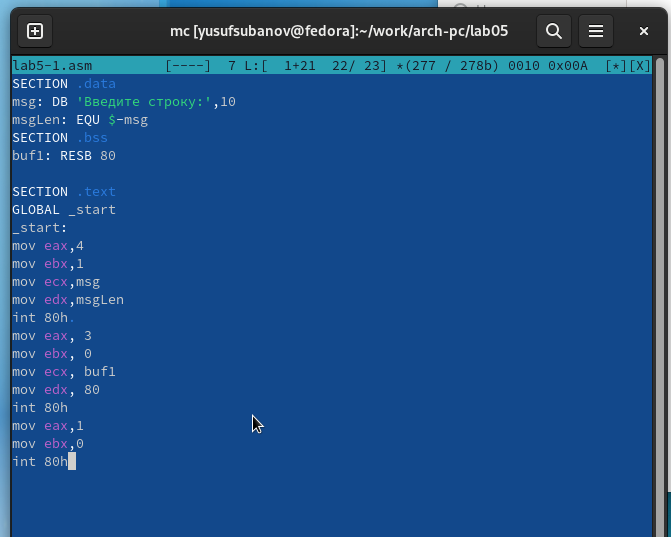


Figure 3: Программа в файле lab05-1.asm

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. Откроем файл на просмотр и проверим.

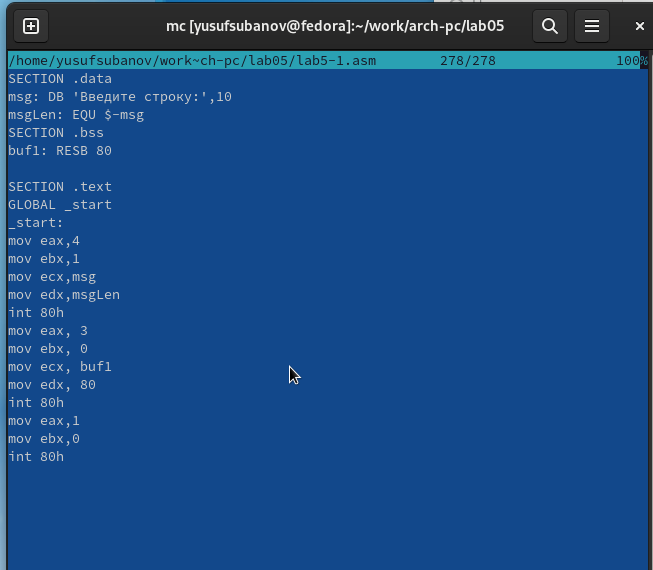


Figure 4: Просмотр файла lab05-1.asm

1. Скомпилировали программу и провреили запуск.

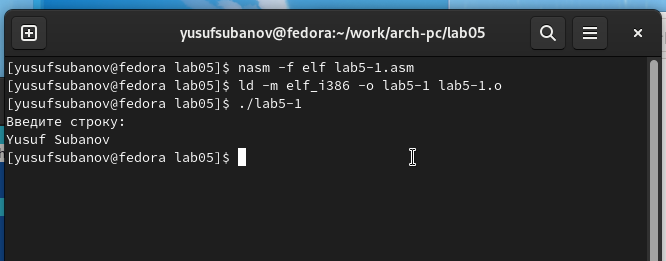


Figure 5: Запуск программы lab05-1.asm

1. Скачали файл in\_out.asm.
2. Добавили файл in\_out.asm в рабочий каталог.
3. Скопировали lab05-1.asm в lab05-2.asm.

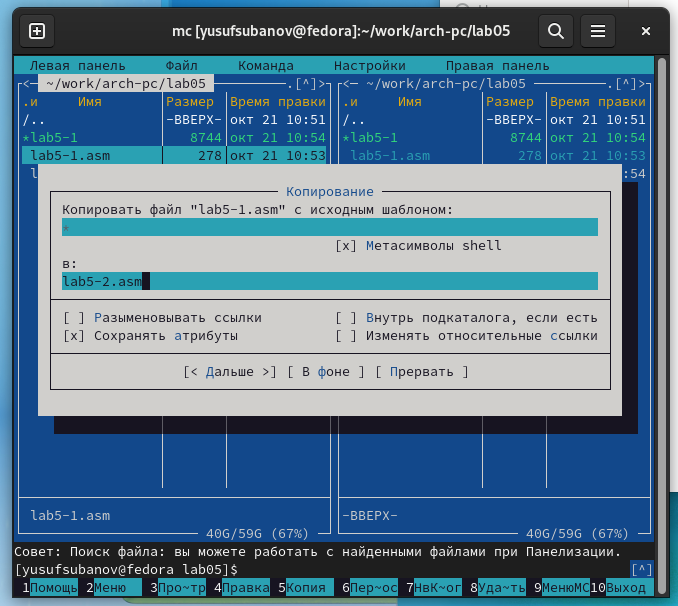


Figure 6: Копирование файла

1. Написали код программы lab05-2.asm. Скомпилировали программу и провреили запуск.

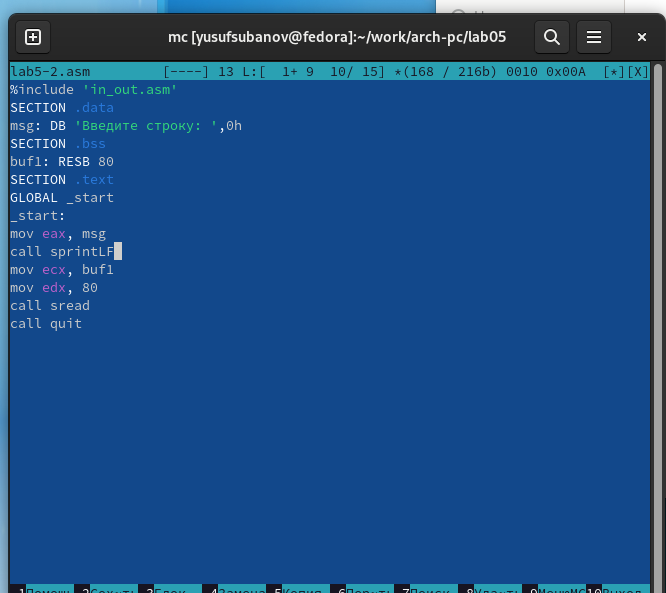


Figure 7: Программа в файле lab05-2.asm

%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

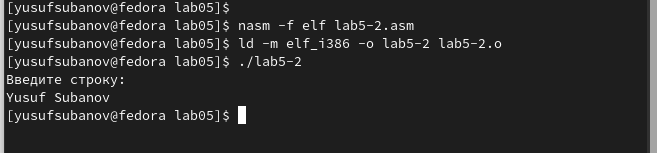


Figure 8: Запуск программы lab05-2.asm

1. В файле lab5-2.asm заменили подпрограмму sprintLF на sprint. Теперь после вывода строки ввод происходит в ту же строку.

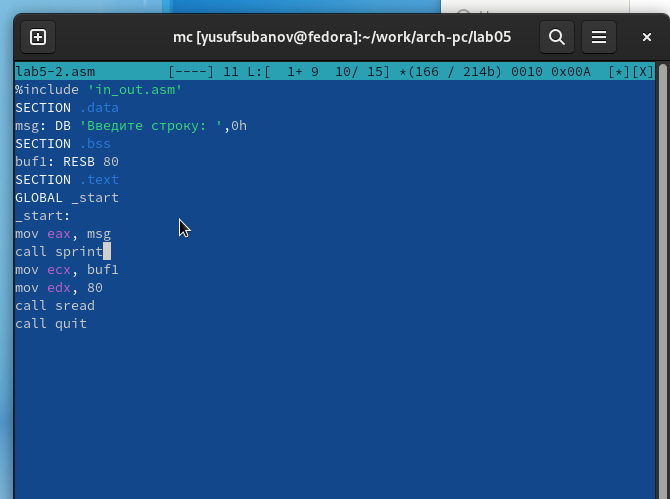


Figure 9: Программа в файле lab05-2.asm

%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

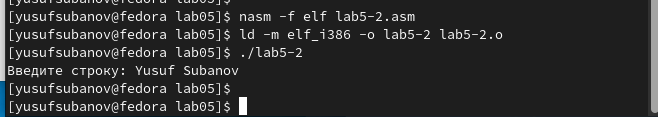


Figure 10: Запуск программы lab05-2.asm

1. Скопировали программу lab05-1.asm и изменили код, чтобы выводить введенную строку.

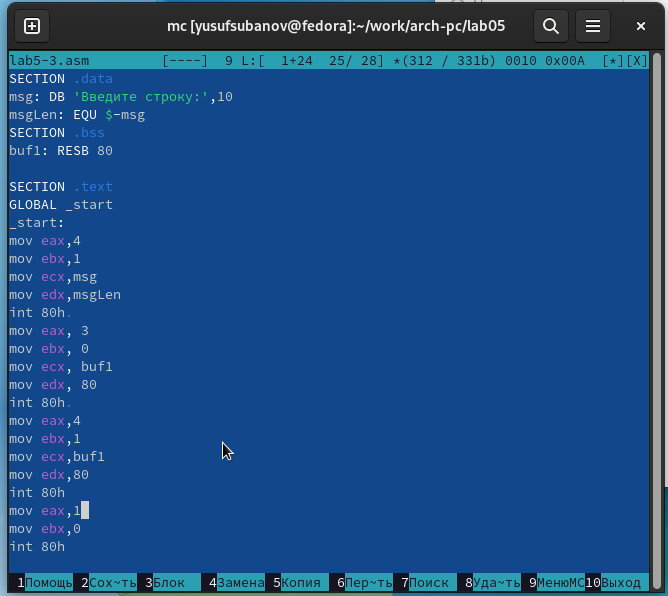


Figure 11: Программа в файле lab05-3.asm

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

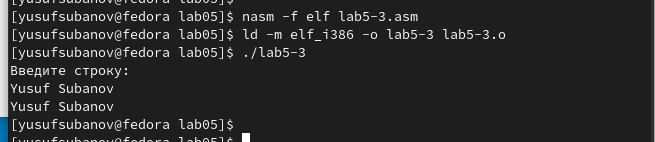


Figure 12: Запуск программы lab05-3.asm

1. Скопировали программу lab05-2.asm и изменили код, чтобы выводить введенную строку.

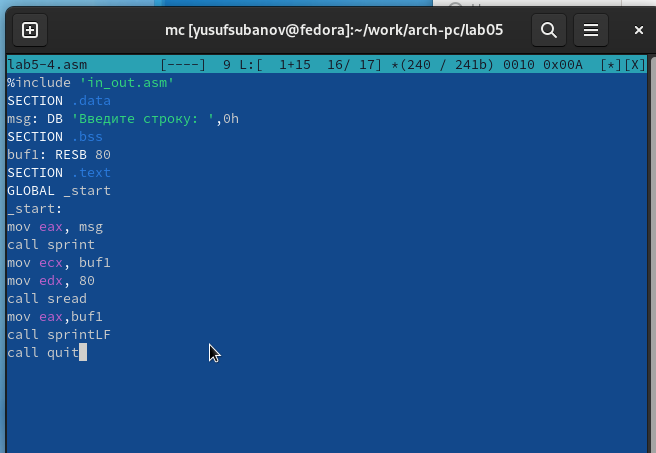


Figure 13: Программа в файле lab05-4.asm

%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 sprintLF  
call quit

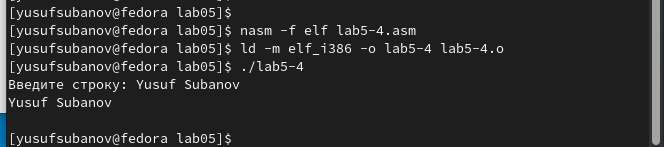


Figure 14: Запуск программы lab05-4.asm

# 3 Выводы

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