

# **Отчет по лабораторной работе №5**

**Архитектура компьютера**

Маваси Башар

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение самостоятельной работй	8
4	Выводы	14

# Список иллюстраций

2.1	Открытие файла и ввод текста . . . . .	6
2.2	Оттранслирование, компоновка и запуск файла, ввод данных . .	6
2.3	Создание и редактирование файла . . . . .	7
2.4	Оттранслирование, компоновка и запуск файла, ввод данных . .	7
3.1	Копирование файла . . . . .	8
3.2	Изменение программы . . . . .	9
3.3	Запуск программы . . . . .	10
3.4	Копирование файла . . . . .	11
3.5	Изменение программы . . . . .	12
3.6	Запуск программы . . . . .	13

## Список таблиц

# 1 Цель работы

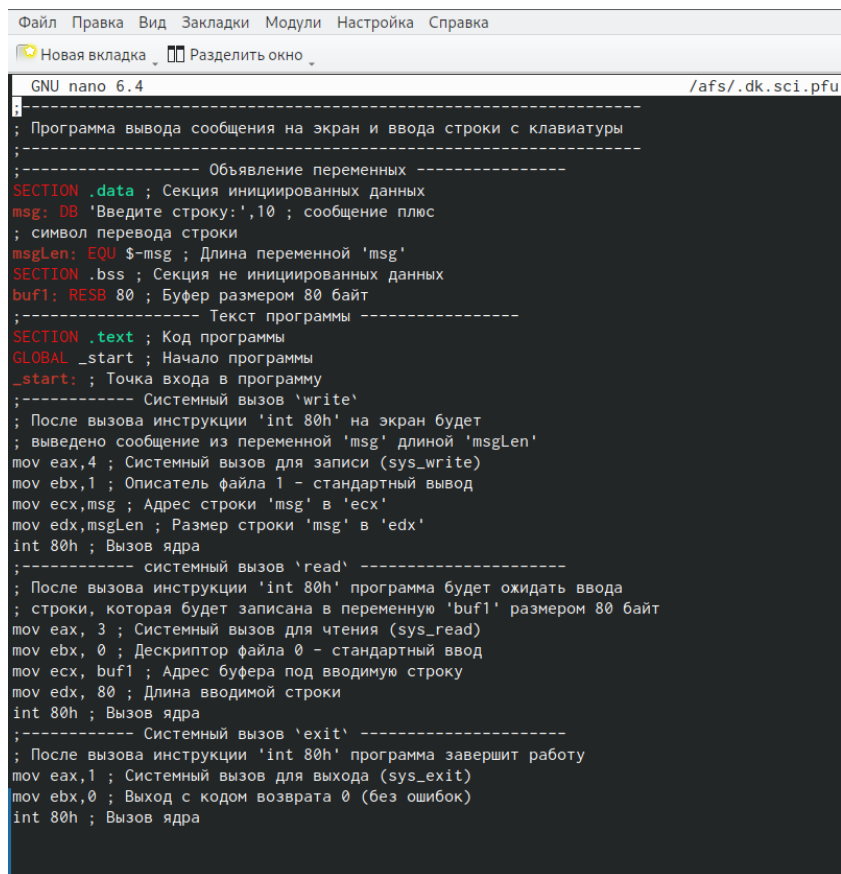
Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.



## Создание и редактирование файла

Рис. 2.3: Создание и редактирование файла

Оттранслируйте текст программы lab5-2.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл и ввел свои имя и фамилию. (рис. 2.4).



```
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно
GNU nano 6.4 /afs/.dk.sci.pfu
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный 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
;----- 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
```

Рис. 2.4: Оттранслирование, компоновка и запуск файла, ввод данных

### 3 Выполнение самостоятельной работы

Копирую файл lab5-1.asm с именем lab5-1.1.asm (рис. 3.1).

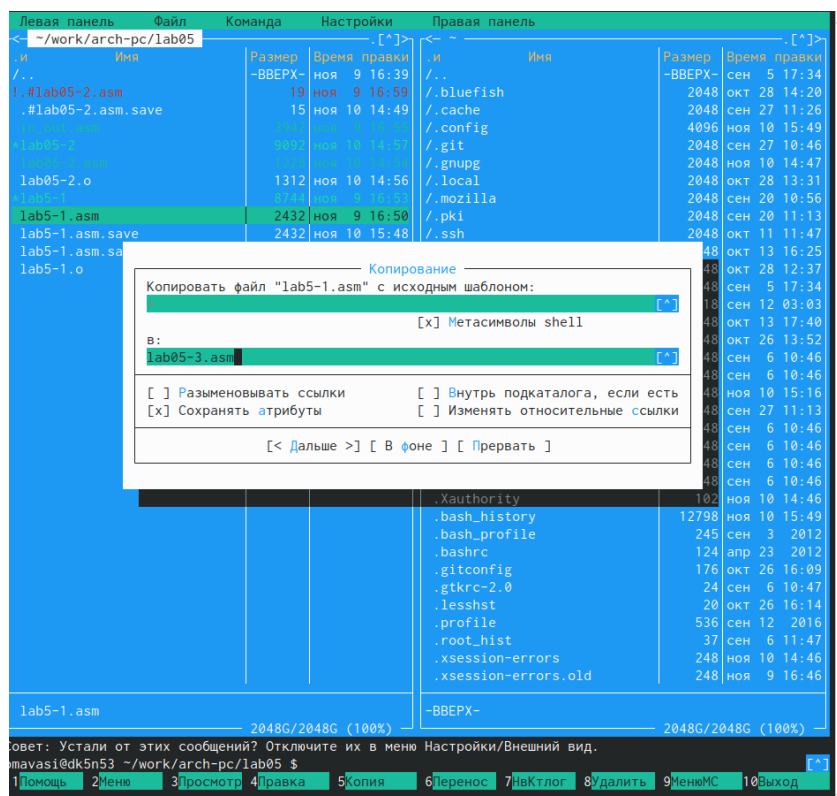
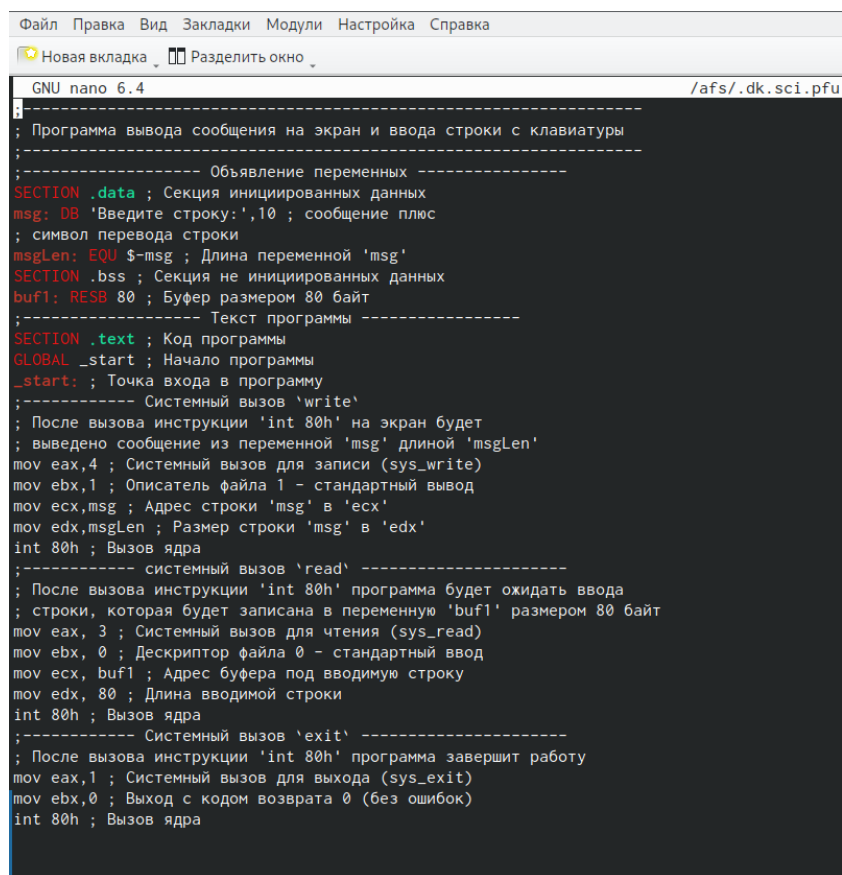


Рис. 3.1: Копирование файла

Изменяю код программы, добавляя вывод введенной строки (рис. 3.2).

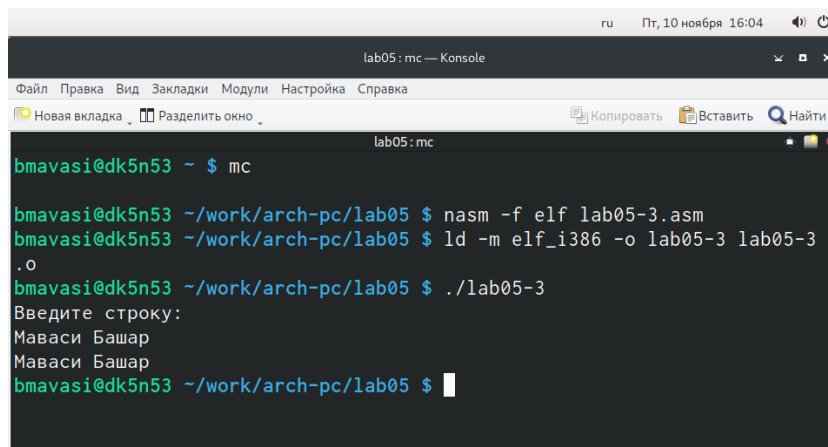




```
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно
GNU nano 6.4 /afs/.dk.sci.pfu
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;----- Объявление переменных -----
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный 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
;----- 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
```

Рис. 3.2: Изменение программы

Создаю объектный файл lab5-1.1.o, компоную его в исполняемый файл, запускаю исполняемый файл (рис. 3.3).



```
bmavasi@dk5n53 ~ $ mc
bmavasi@dk5n53 ~/work/arch-pc/lab05 $ nasm -f elf lab05-3.asm
bmavasi@dk5n53 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab05-3 lab05-3.o
bmavasi@dk5n53 ~/work/arch-pc/lab05 $ ./lab05-3
Введите строку:
Маваси Башар
Маваси Башар
bmavasi@dk5n53 ~/work/arch-pc/lab05 $
```

Рис. 3.3: Запуск программы

Программа из пункта 1: ;————— 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 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

Копирую файл lab5-2.asm с именем lab5-2.2.asm (рис. 3.4).

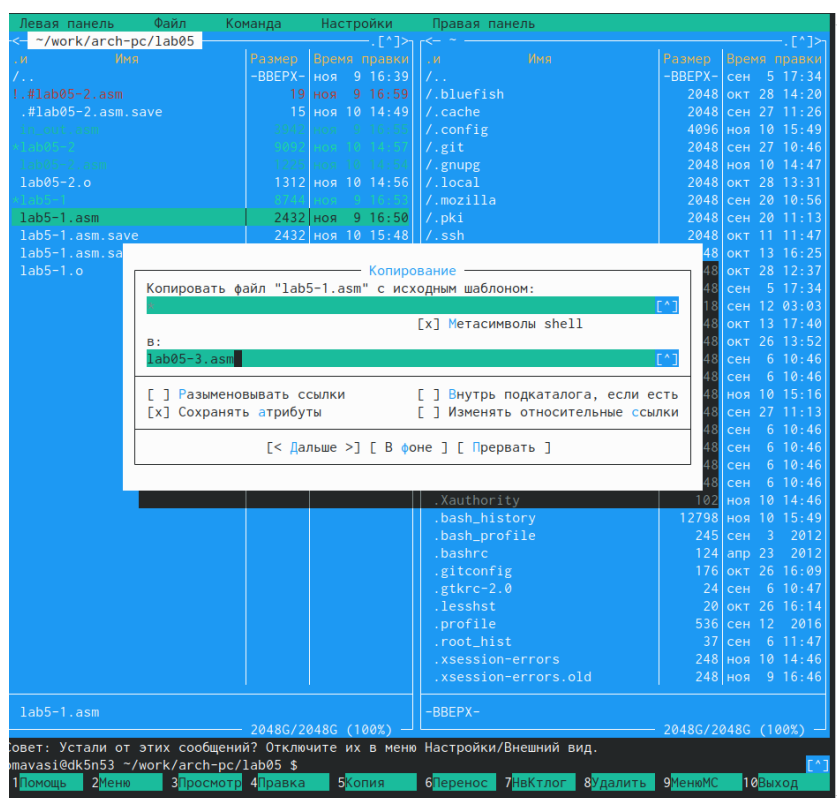
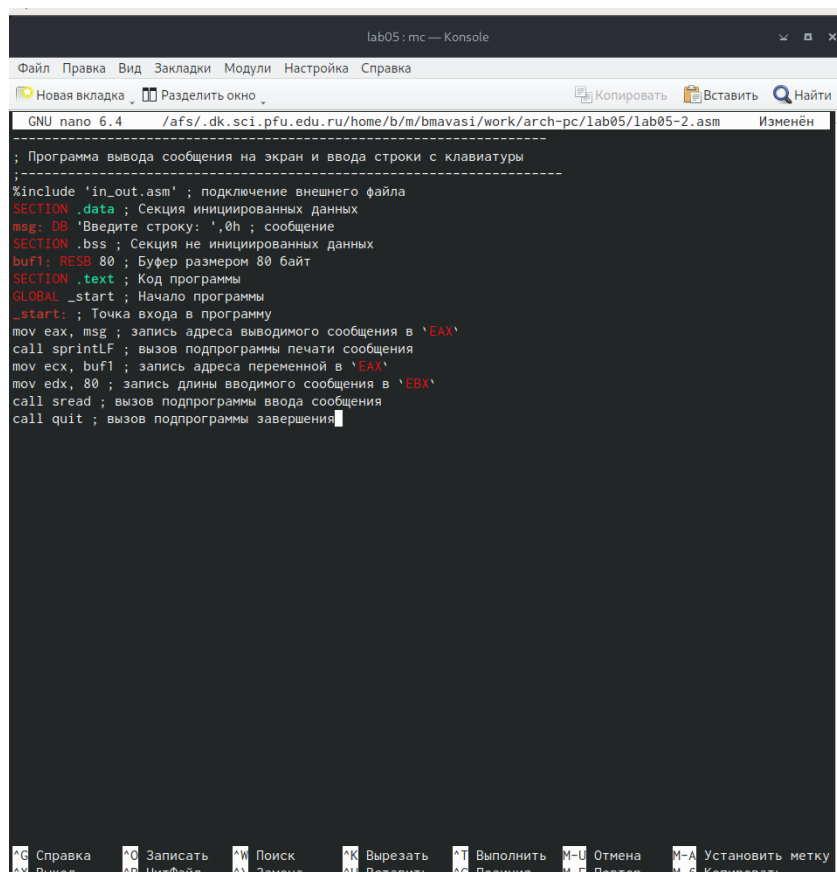


Рис. 3.4: Копирование файла

Изменяю код программы, добавляя вывод введенной строки (рис. 3.5).



```
lab05: mc — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно Копировать Вставить Найти
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/b/m/bmavasi/work/arch-pc/lab05/lab05-2.asm Изменён

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 sprintf ; 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
call quit ; вызов подпрограммы завершения
```

Рис. 3.5: Изменение программы

Создаю объектный файл lab5-2.2.o, componую его в исполняемый файл, запускаю исполняемый файл (рис. 3.6).

```
bmavasi@dk5n53 ~/work/arch-pc/lab05 $ nasm -f elf lab05-2.asm
bmavasi@dk5n53 ~/work/arch-pc/lab05 $ d -m elf_i386 -o lab05-2 lab05-2.o
bash: d: команда не найдена
bmavasi@dk5n53 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab05-2 lab05-2.o
bmavasi@dk5n53 ~/work/arch-pc/lab05 $ ./lab05-2
Введите строку:
Маваси Башар
bmavasi@dk5n53 ~/work/arch-pc/lab05 $
```

Рис. 3.6: Запуск программы

## 4 Выводы

Я приобрел практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.