

Отчёт по лабораторной работе №5

Архитектура компьютера

Морозова Мария Вячеславовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выполнение самостоятельной работы	11
6	Выводы	14

Список иллюстраций

4.1	Создание файла	8
4.2	Программа вывода и ввода	8
4.3	Просмотр файла	9
4.4	Компоновка и запуск	9
4.5	Проверка	9
4.6	Компоновка и запуск	10
4.7	Замена	10
5.1	Изменения	11
5.2	Запуск	11
5.3	Копия	12
5.4	Исправление	12
5.5	Запуск	12

Список таблиц

1 Цель работы

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

2 Задание

Создать программы с выводом приглашения: “Введите строку”.

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter. В Midnight Commander используются функциональные клавиши F1 — F10, к которым привязаны часто выполняемые операции.

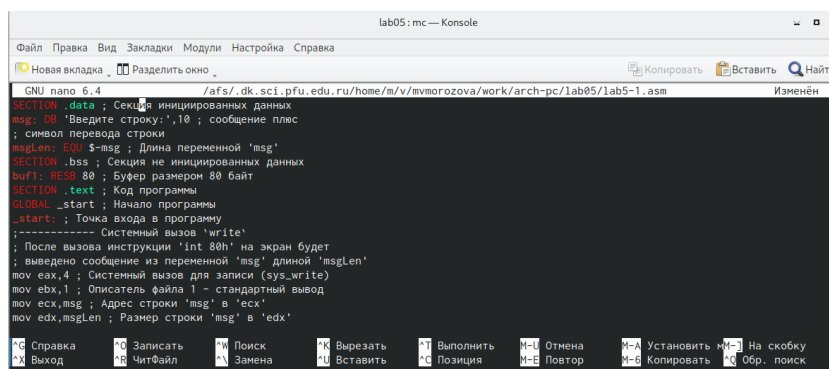
4 Выполнение лабораторной работы

Перешли в каталог `~/work/arch-pc`, с помощью строки ввода и команды `touch` создали файл `lab5-1.asm` в папке `lab05`. (рис. 4.1).

```
mvmorozova@dk8n80 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/report $ cd
mvmorozova@dk8n80 ~ $ mc
mvmorozova@dk8n80 ~/work/arch-pc/lab05 $ touch lab5-1.asm
```

Рис. 4.1: Создание файла

Вводим текст программы вывода сообщения на экран и ввода строки с клавиатуры. (рис. 4.2).



```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/m/v/mvmorozova/work/arch-pc/lab05/lab5-1.asm
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
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 'edx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
```

Рис. 4.2: Программа вывода и ввода

С помощью клавиши `F3` открыла файл `lab5-1.asm`, убедилась, что он содержит текст программы. (рис. 4.3).

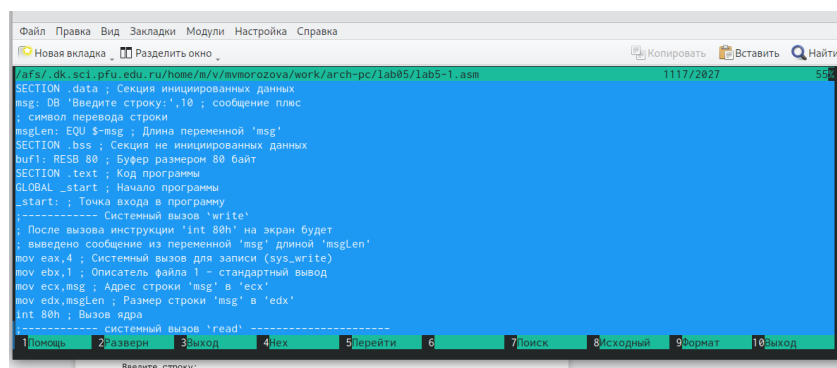


Рис. 4.3: Просмотр файла

Оттранслировала текст программы lab5-1.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл. На запрос ввела ФИО. (рис. 4.4).

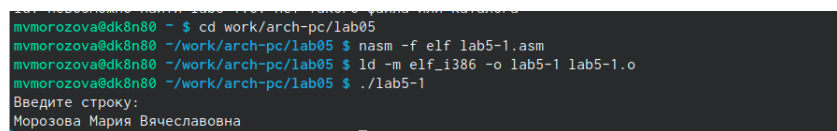


Рис. 4.4: Компоновка и запуск

Скопировала файл in_out.asm в каталог с файлом lab5-1.asm и проверила, что они действительно находятся в одной папке, создала копию файла lab5-1 с именем lab5-2. (рис. 4.5).

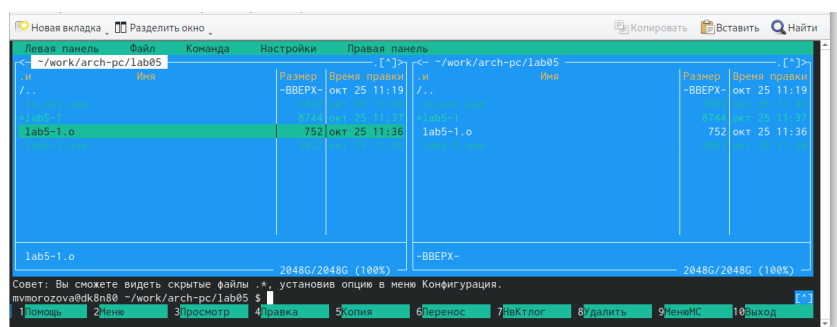


Рис. 4.5: Проверка

Скомпоновала объектный файл, запустила программу lab5-2. (рис. 4.6).

```

mvmorozova@dk8n54 ~ $ mc

mvmorozova@dk8n54 ~ $ nasm -f elf lab5-2.asm
nasm: fatal: unable to open input file 'lab5-2.asm' No such file or directory
mvmorozova@dk8n54 ~ $ cd work/arch-pc/lab05
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Морозова Мария Вячеславовна

```

Рис. 4.6: Компоновка и запуск

Запустила прорамму после замены sprintLF на sprint. (рис. 4.7).

```

Морозова Мария Вячеславовна
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ mc

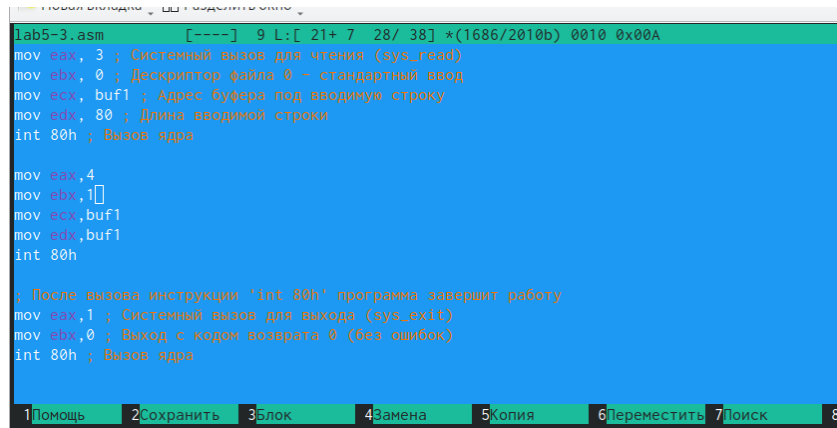
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Морозова Мария Вячеславовна
mvmorozova@dk8n54 ~/work/arch-pc/lab05 $ █

```

Рис. 4.7: Замена

5 Выполнение самостоятельной работы

Внесла изменения в программу. (рис. 5.1).



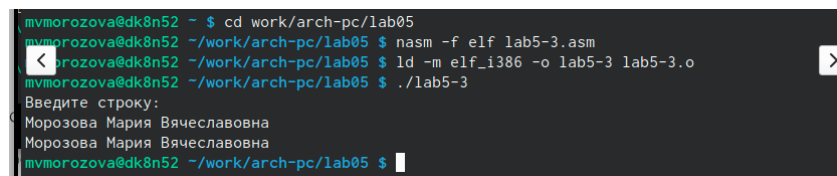
```
lab5-3.asm [---] 9 L:[ 21+ 7 28/ 38] *(1686/2010b) 0010 0x00A
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра

mov eax, 4
mov ebx, 1
mov ecx, buf1
mov edx, buf1
int 80h

; 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. 5.1: Изменения

Запустила программу с изменениями. (рис. 5.2).



```
mvtmorozova@dk8n52 ~ $ cd work/arch-pc/lab05
mvtmorozova@dk8n52 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
mvtmorozova@dk8n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
mvtmorozova@dk8n52 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Морозова Мария Вячеславовна
Морозова Мария Вячеславовна
mvtmorozova@dk8n52 ~/work/arch-pc/lab05 $
```

Рис. 5.2: Запуск

Создала копию файла lab5-2. (рис. 5.3).

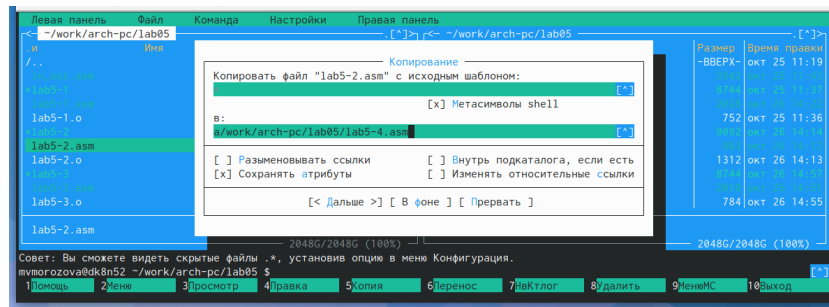


Рис. 5.3: Копия

Исправила текст программы. (рис. 5.4).

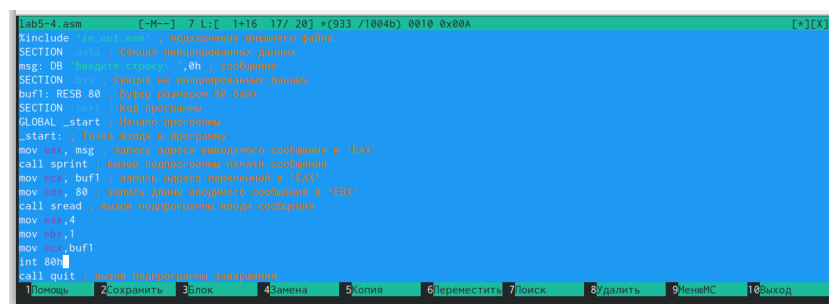


Рис. 5.4: Исправление

Запустила исправленную программу. (рис. 5.5).

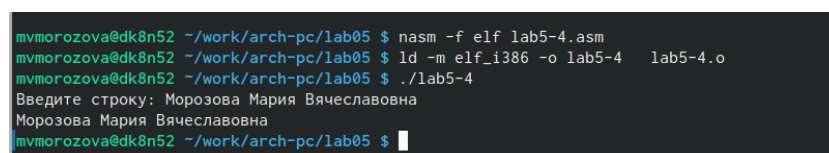


Рис. 5.5: Запуск

Листинги:

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 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 пере-
 менной '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
 mov eax,4 mov ebx,1 mov ecx,buf1 mov edx,buf1 int 80h
 ; 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

6 Выводы

Были приобретены практические навыки работы в Midnight Commander, освоены инструкции языка ассемблера `mov` и `int`.