

# **Отчет по лабораторно работе №5**

**Дисциплина: архитектура компьютера**

Провоторов Антон Григорьевич

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	12
4	Выводы	17

# Список иллюстраций

2.1	Midnight Commander . . . . .	6
2.2	Перемещение между директориями . . . . .	6
2.3	Создание директории . . . . .	7
2.4	Создание файла . . . . .	8
2.5	Изменение файла . . . . .	8
2.6	Просмотр файла . . . . .	9
2.7	Запуск программы . . . . .	9
2.8	Копирование файла . . . . .	9
2.9	Копирование файла . . . . .	9
2.10	Изменение программы . . . . .	10
2.11	Запуск программы . . . . .	10
2.12	Изменение программы . . . . .	11
2.13	Запуск программы . . . . .	11
3.1	Копирование файла . . . . .	12
3.2	Изменение программы . . . . .	12
3.3	Запуск программы . . . . .	13
3.4	Копирование файла . . . . .	14
3.5	Изменение файла . . . . .	15
3.6	Запуск программы . . . . .	15

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - приобретение практических навыков работы в Midnight Commander и освоение инструкций языка ассемблера `mov` и `int`

## 2 Выполнение лабораторной работы

Открыл Midnight Commander (рис. [2.1]).

Midnight Commander

Рис. 2.1: Midnight Commander

Перешел в каталог arch-rc (рис. [2.2]).

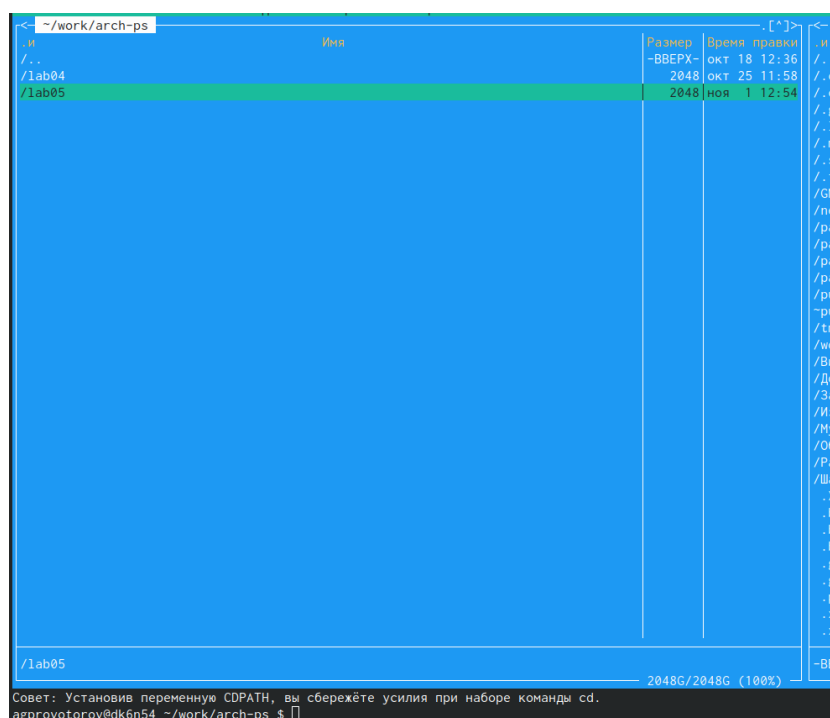


Рис. 2.2: Перемещение между директориями

Создаю каталог lab05 (рис. [2.3]).

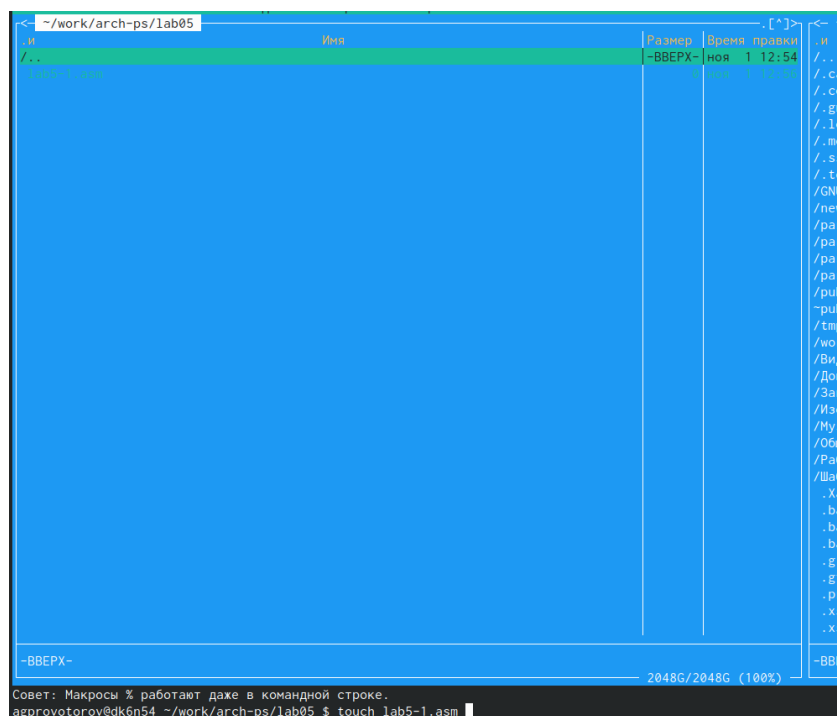


Рис. 2.3: Создание директории

Перехожу в созданную директорию и создаю файл `lab5-1.asm` (рис. [2.4]).

```

SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку ",10 ; сообщение плюс
; символ перевода строки
msgLen: EQU 1-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
52 Денидова А. В.
Архитектура 386

;----- 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: Создание файла

Открываю файл lfb6-1.asm для редактирования во встроенном редакторе nano и вставляю в него программу (рис. [2.5]). После сохраняю изменения и закрываю файл.

### Изменение файла

Рис. 2.5: Изменение файла

Открываю файл lab6-1.asm для просмотра (рис. [2.6]).





```

;-----
; 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 sprintLF ; 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. 2.10: Изменение программы

Создаю объектный файл lab6-2.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. [2.11]).

```

agprovotorov@dk8n59 ~/work/arch-ps/lab05 $ nasm -f elf lab5-2.asm
agprovotorov@dk8n59 ~/work/arch-ps/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
agprovotorov@dk8n59 ~/work/arch-ps/lab05 $ ./lab5-2
Введите строку:
Првоторов Антон Григорьевич
agprovotorov@dk8n59 ~/work/arch-ps/lab05 $

```

Рис. 2.11: Запуск программы

В тексте программы lab6-2.asm заменяю sprintLF на sprint (рис. [2.12]).

```

;-----
; 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 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
call quit ; вызов подпрограммы завершения

```

Рис. 2.12: Изменение программы

Создаю объектный файл lab6-2.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. [2.13]). Теперь ввод производится на той же строке, что и вывод, убран символ перевода строки после вывода.

```

agprovotorov@dk8n59 ~/work/arch-ps/lab05 $ nasm -f elf lab5-2.asm
agprovotorov@dk8n59 ~/work/arch-ps/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
agprovotorov@dk8n59 ~/work/arch-ps/lab05 $ ./lab5-2
Введите строку:
Провторов Антон Григорьевич

```

Рис. 2.13: Запуск программы

### 3 Выполнение заданий для самостоятельной работы

1. Копирую файл lab5-1.asm с именем lab5-3.asm (рис. [3.1]).

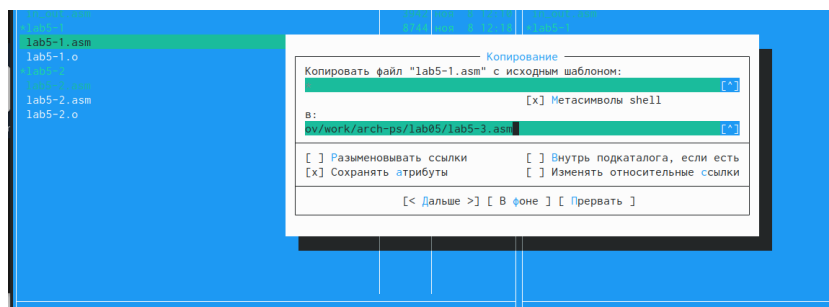


Рис. 3.1: Копирование файла

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

```
agprovotorov@dk6n55 ~/work/arch-ps/lab05 $ nasm -f elf lab5-3.asm
agprovotorov@dk6n55 ~/work/arch-ps/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
agprovotorov@dk6n55 ~/work/arch-ps/lab05 $ ./lab5-3
bash: ./lab5-3: Нет такого файла или каталога
agprovotorov@dk6n55 ~/work/arch-ps/lab05 $ ./lab5-3
bash: ./lab5-3: Нет такого файла или каталога
agprovotorov@dk6n55 ~/work/arch-ps/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
agprovotorov@dk6n55 ~/work/arch-ps/lab05 $ ./lab5-3
Введите строку:
Провоторов Антон
Провоторов Антон
```

Рис. 3.2: Изменение программы

Создаю объектный файл lab5-3.o, komponую его в исполняемый файл, запускаю исполняемый файл (рис. [3.3]).

```

13
14 SECTION .text ; Код программы
15 GLOBAL _start ; Начало программы
16 _start: ; Точка входа в программу
17
18
19 ;----- Системный вызов 'write'
20 ; После вызова инструкции 'int 80h' на экран будет
21 ; выведено сообщение из переменной 'msg' длиной 'msgLen'
22
23
24 mov eax,4 ; Системный вызов для записи (sys_write)
25 mov ebx,1 ; Описатель файла 1 - стандартный вывод
26 mov ecx,msg ; Адрес строки 'msg' в 'ecx'
27 mov edx,msgLen ; Размер строки 'msg' в 'edx'
28 int 80h ; Вызов ядра
29
30 ;----- системный вызов 'read' -----
31 ; После вызова инструкции 'int 80h' программа будет ожидать ввода
32 ; строки, которая будет записана в переменную 'buf1' размером 80 байт
33
34
35 mov eax, 3 ; Системный вызов для чтения (sys_read)
36 mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
37 mov ecx, buf1 ; Адрес буфера под вводимую строку
38 mov edx, 80 ; Длина вводимой строки
39 int 80h
40 mov eax, 4
41 mov ebx, 1
42 mov ecx, buf1
43 mov edx, 80
44 int 80h

```

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 ;Descriptor 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

```

2. Копирую файл lab5-2.asm с именем lab5-4.asm (рис. [3.4]).

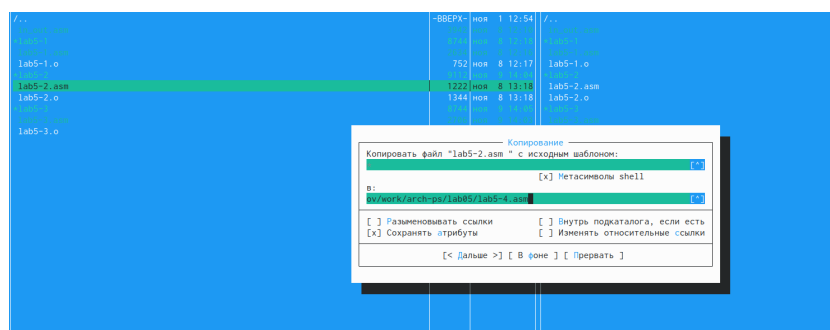


Рис. 3.4: Копирование файла

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

```

1;-----
agprovatorc 2; Программа вывода сообщения на экран и ввода строки с клавиатуры
agprovatorc 3;-----
agprovatorc 4%include 'in_out.asm' ; подключение внешнего файла
bash: ./lab5-4.asm: 5SECTION .data ; Секция иницированных данных
agprovatorc 6msg: DB 'Введите строку: ',0h ; сообщение
bash: ./lab5-4.asm: 7SECTION .bss ; Секция не иницированных данных
agprovatorc 8buf1: RESB 80 ; Буфер размером 80 байт
agprovatorc 9SECTION .text ; Код программы
Введите строку: 10GLOBAL _start ; Начало программы
Провоторов 11_start: ; Точка входа в программу
12mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
agprovatorc 13call sprint ; вызов подпрограммы печати сообщения
14mov ecx, buf1 ; запись адреса переменной в 'EAX'
agprovatorc 15mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
lab5-4.asm: 16call sread ; вызов подпрограммы ввода сообщения
agprovatorc 17mov eax,4 ; Системный вызов для записи (sys_write)
lab5-4.asm: 18mov ebx,1 ; Описатель файла '1' - стандартный вывод
agprovatorc 19mov ecx,buf1 ; Адрес строки buf1 в ecx
lab5-4.asm: 20int 80h ; Вызов ядра
agprovatorc 21call quit ; вызов подпрограммы завершения
lab5-4.asm:
lab5-4.asm:
agprovatorc
agprovatorc
lab5-4.asm:
lab5-4.asm:
agprovatorc
agprovatorc
Введите строку:
Провоторов

```

Рис. 3.5: Изменение файла

Создаю объектный файл lab5-4.o, компоную его в исполняемый файл, запуская исполняемый файл (рис. [3.6]).

```

agprovatorov@ek6n55 ~/work/arch-ps/lab05 $ nasm -f elf lab5-4.asm
lab5-4.asm:1: error: label or instruction expected at start of line
lab5-4.asm:24: error: parser: instruction expected
agprovatorov@ek6n55 ~/work/arch-ps/lab05 $ nasm -f elf lab5-4.asm
agprovatorov@ek6n55 ~/work/arch-ps/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
agprovatorov@ek6n55 ~/work/arch-ps/lab05 $ ./lab5-4
Введите строку: Провоторов Антон
Провоторов Антон

```

Рис. 3.6: Запуск программы

Программа из пункта 2:

```

;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

%include 'in_out.asm' ; подключение внешнего файла

SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение

SECTION .bss ; Секция не иницированных 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
```



## 4 Выводы

Я приобрел практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера mov и int.