

# **Отчет по лабораторной работе №5**

**Архитектура вычислительных систем**

Диана Олеговна Шаяхметова

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>

## Список иллюстраций

4.1	переход в каталог . . . . .	8
4.2	создание папки . . . . .	9
4.3	создание файла . . . . .	10
4.4	ввод текста . . . . .	11
4.5	проверка файла . . . . .	12
4.6	оттранслировка, компоновка, запуск файла . . . . .	12
4.7	запуск файла . . . . .	13

## **Список таблиц**

# 1 Цель работы

Целью работы является приобретение практических навыков работы Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

1. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab6-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран. Не забудьте, подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.
4. Создайте исполняемый файл и проверьте его работу.

### **3 Теоретическое введение**

## 4 Выполнение лабораторной работы

1. Открываем Midnight commander и переходим в каталог ~/work/arch-pc



Рис. 4.1: переход в каталог

2. Создаем папку lab06 и переходим в созданный каталог



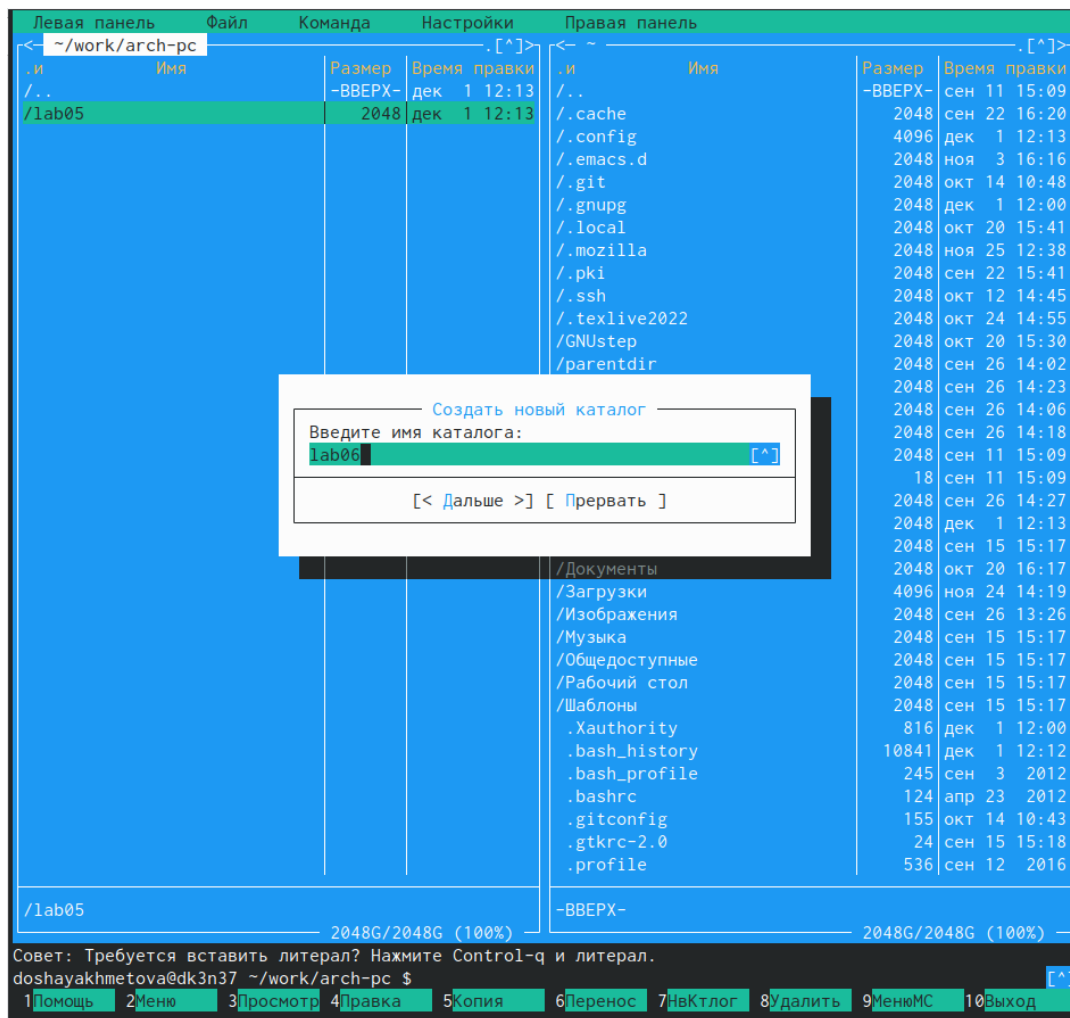


Рис. 4.2: создание папки

3. Пользуясь строкой ввода и командой `touch` создаем файл `lab6-1.asm`

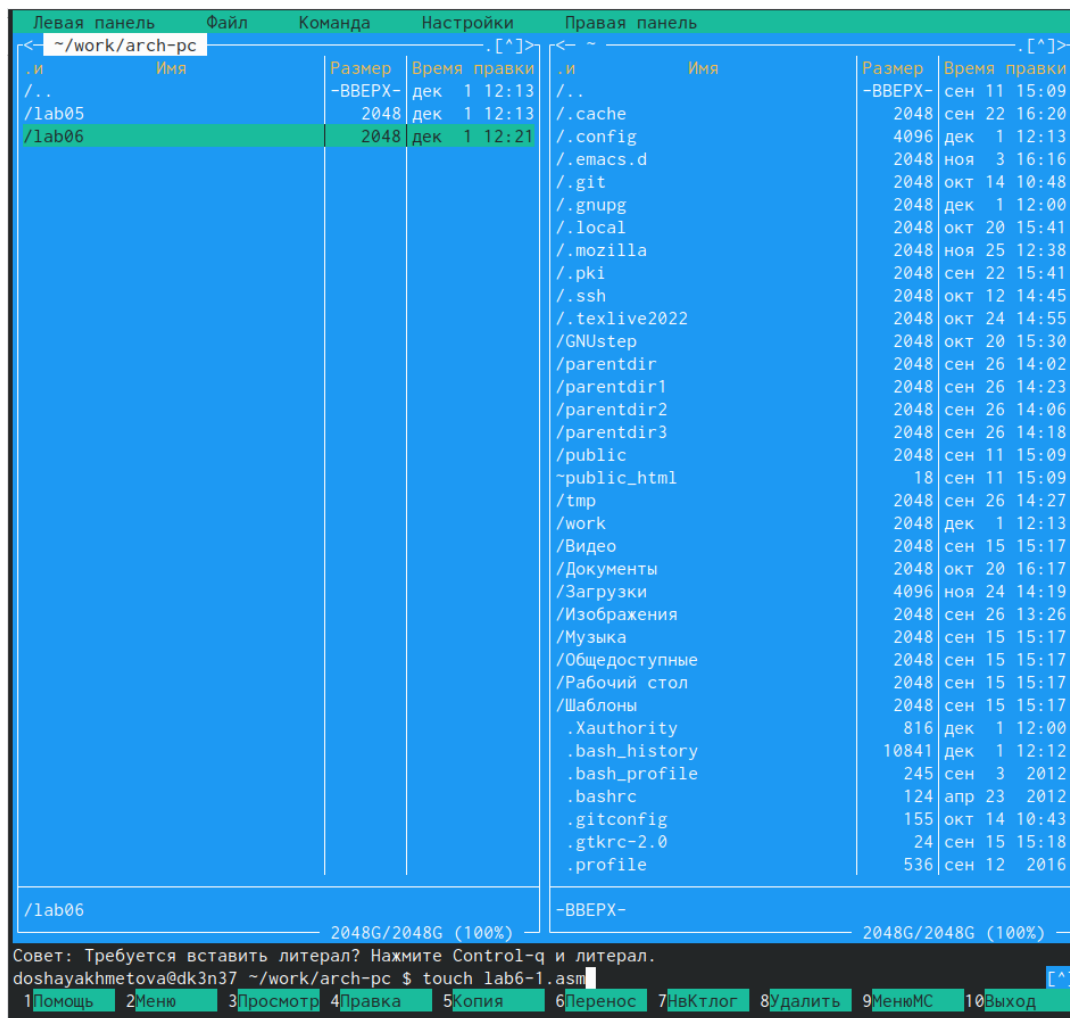


Рис. 4.3: создание файла

#### 4. Вводим текст программы, сохраняем и закрываем файл

```
/afs/.dk.sci.pfu.edu.ru/home/d/o/do-yakhmetova/work/arch-pc/lab6-1.asm 2434/2434 100%
;
; 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

1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рис. 4.4: ввод текста

## 5. Убедимся, что файл содержит текст программы

```
Открыть ▾ + lab6-1.asm
~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----
4 ;----- Объявление переменных -----
5 SECTION .data ; Секция иницированных данных
6 msg: DB 'Введите строку:',10 ; сообщение плюс
7 ; символ перевода строки
8 msgLen: EQU $-msg ; Длина переменной 'msg'
9 SECTION .bss ; Секция не иницированных данных
10 buf1: RESB 80 ; Буфер размером 80 байт
11 ;----- Текст программы -----
12 SECTION .text ; Код программы
13 GLOBAL _start ; Начало программы
14 _start: ; Точка входа в программу
15 ;----- Системный вызов 'write' -----
16 ; После вызова инструкции 'int 80h' на экран будет
17 ; выведено сообщение из переменной 'msg' длиной 'msgLen'
18 mov eax,4 ; Системный вызов для записи (sys_write)
19 mov ebx,1 ; Описатель файла 1 - стандартный вывод
20 mov ecx,msg ; Адрес строки 'msg' в 'ecx'
21 mov edx,msgLen ; Размер строки 'msg' в 'edx'
22 int 80h ; Вызов ядра
23 ;----- системный вызов 'read' -----
24 ; После вызова инструкции 'int 80h' программа будет ожидать ввода
25 ; строки, которая будет записана в переменную 'buf1' размером 80
26 байт
27 mov eax, 3 ; Системный вызов для чтения (sys_read)
28 mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
29 mov ecx, buf1 ; Адрес буфера под вводимую строку
30 mov edx, 80 ; Длина вводимой строки
```

Рис. 4.5: проверка файла

6. Выполняем оттранслировку и компоновку объектного файла и запускаем получившийся файл

```
doshayakhmetova@dk3n37 ~ $ nasm -f elf lab6-1.asm
nasm: fatal: unable to open input file 'lab6-1.asm' No such file or directory
doshayakhmetova@dk3n37 ~ $ cd ~/work/arch-pc
doshayakhmetova@dk3n37 ~/work/arch-pc $ nasm -f elf lab6-1.asm
lab6-1.asm:26: warning: label alone on a line without a colon might be in error [-w+label-orphan]
doshayakhmetova@dk3n37 ~/work/arch-pc $ ld -m elf_i386 -o lab6-1 lab6-1.o
doshayakhmetova@dk3n37 ~/work/arch-pc $ ./lab6-1
Введите строку:
Шаяхметова Диана Олеговна
doshayakhmetova@dk3n37 ~/work/arch-pc $
```

Рис. 4.6: оттранслировка, компоновка, запуск файла

7. Создаем копию файла lab6-1.asm с именем lab6-2.asm, исправляем текст программы в файле lab6-2.asm в соответствии с листингом 6.2

```
lab6-2.asm      [-M--] 51 L:[ 1+12 13/ 17] *(919 /1224b) 0010 0x00A
;-----
; 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
;-----
; 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
```

8. Создаем исполняемый файл и проверяем его работу

```
doshayakhmetova@dk8n59 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
doshayakhmetova@dk8n59 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
doshayakhmetova@dk8n59 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:
Шаяхметова Диана Олеговна
```

Рис. 4.7: запуск файла

9. Заменяем подпрограмму `sprintLF` на `sprint` в файле `lab6-2.asm`, создаем исполняемый файл и проверяем его работу

```
lab6-2.asm      [----] 13 L: [ 1+20 21/ 22] *(1197/1266b) 0010 0x00A
;-----
; 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

mov eax, buf1
mov ebx, 80
call sprintLF
call quit ; вызов подпрограммы завершения

doshayakhmetova@dk8n59 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
doshayakhmetova@dk8n59 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
doshayakhmetova@dk8n59 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:
Шаяхметова Диана Олеговна
Шаяхметова Диана Олеговна

doshayakhmetova@dk8n59 ~/work/arch-pc/lab06 $
```

## 5 Выводы

Я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера `mov` и `int`.

## **Список литературы**