

# **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

**дисциплина: Архитектура компьютера**

Мантуров Татархан Бесланович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Порядок выполнения лабораторной работы . . . . .	6
3.2	Задание для самостоятельной работы . . . . .	11
<b>4</b>	<b>Выводы</b>	<b>13</b>

## Список иллюстраций

3.1	ms . . . . .	6
3.2	Создаем каталог . . . . .	7
3.3	touch . . . . .	7
3.4	Открываем файл, заполняем . . . . .	8
3.5	Открываем файл и убеждаемся, что файл содержит текст программы	9
3.6	Проверка . . . . .	9
3.7	Копируем файл . . . . .	9
3.8	Создаем копию файла . . . . .	10
3.9	Проверяем скопировался ли файл . . . . .	10
3.10	Открываем и заполняем файл . . . . .	10
3.11	Редактируем файл . . . . .	10
3.12	Смотрим, как работает программа и сравниваем с прошлой . . .	11
3.13	Редактируем файл . . . . .	11
3.14	Проверяем правильность . . . . .	12
3.15	Создаем копию файла lab5-2.asm . . . . .	12
3.16	Редактируем файл . . . . .	12
3.17	Проверяем правильность программы . . . . .	12

# 1 Цель работы

Освоить инструкции языка ассемблера mov. Приобрести знания использования Midnight Commander.

## 2 Задание

Написать 2 программы по примеру и изменить их структуру по условию.

## **3 Выполнение лабораторной работы**

### **3.1 Порядок выполнения лабораторной работы**

Открываем Mid. Commander

mc

Рис. 3.1: mc

Создаем каталог lab05

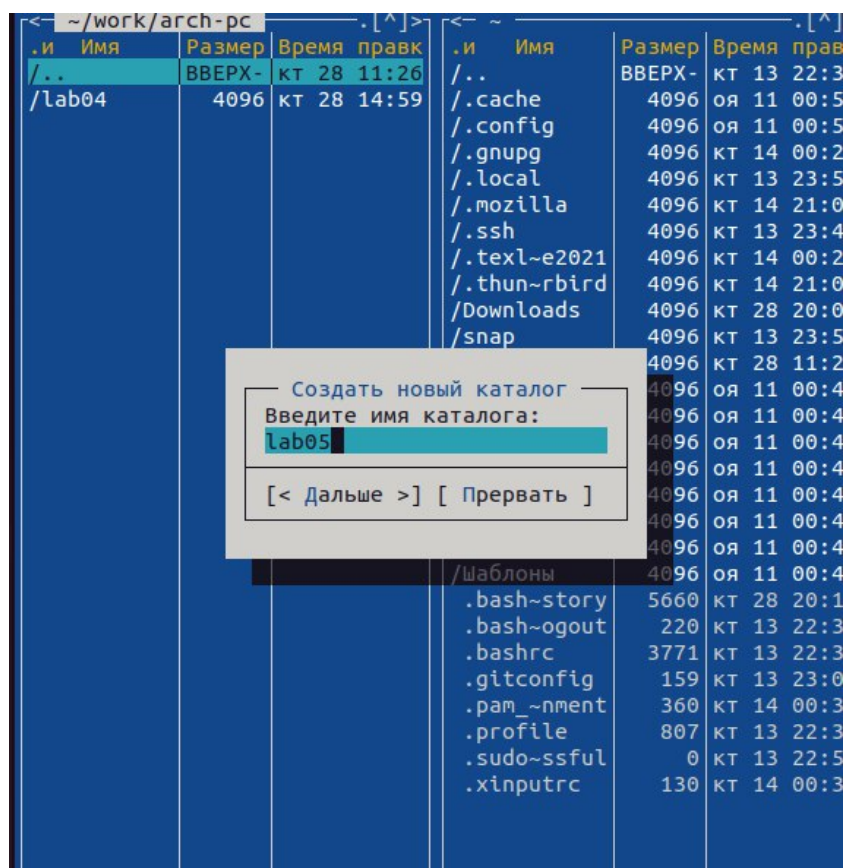


Рис. 3.2: Создаем каталог

Создаем файл lab5-1.asm

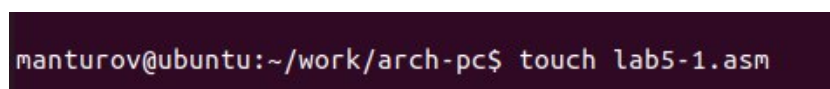


Рис. 3.3: touch

Открываем файл для редактирования и заполняем его по листингу

```

/home/manturov/~h-pc/lab5-1.asm 2432/2432 100%
;
;-----
; 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 ;Descriptor 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. 3.4: Открываем файл, заполняем

Открываем файл и просматриваем



```
/home/manturov/work/arch-pc/lab05/lab5-2.asm *
;----->
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----->
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
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
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: Открываем файл и убеждаемся, что файл содержит текст программы

Транслируем текст программы и запускаем файл

```
manturov@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
manturov@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
manturov@ubuntu:~/work/arch-pc/lab05$ ./lab5-1
Введите строку: Мантуров Татархан
```

Рис. 3.6: Проверка

Копируем файл в нужную директорию

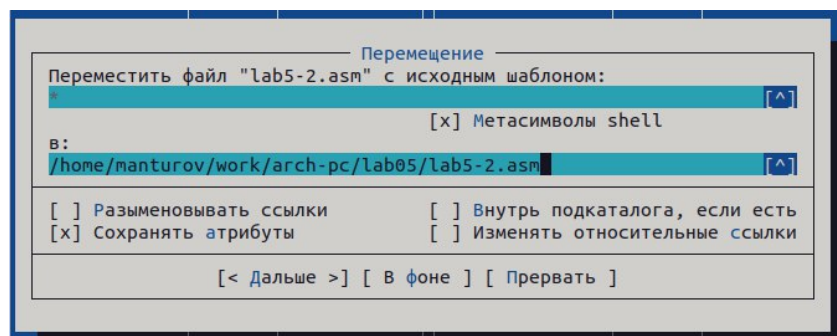


Рис. 3.7: Копируем файл

Создаем копию файла



Рис. 3.8: Создаем копию файла

Проверяем созданный файл

Проверяем скопировался ли файл

Рис. 3.9: Проверяем скопировался ли файл

Открываем новый файл и заполняем его

```
%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
```

Рис. 3.10: Открываем и заполняем файл

Открываем файл для редактирования и меняем sprintLF на sprint

```
mov ecx, msg ; запись адреса выводимого сообщения
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'EAX'
```

Рис. 3.11: Редактируем файл

Транслируем и запускаем файл

```
manturov@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
manturov@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
manturov@ubuntu:~/work/arch-pc/lab05$ ./lab5-1
Введите строку: Мантуров Татархан
```

Рис. 3.12: Смотрим, как работает программа и сравниваем с прошлой

## 3.2 Задание для самостоятельной работы

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль

```
/home/manturov/work/arch-pc/lab05/lab5-1.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
```

Рис. 3.13: Редактируем файл

Транслируем файл и запускаем программу

```
manturov@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
manturov@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
manturov@ubuntu:~/work/arch-pc/lab05$ ./lab5-1
Введите строку: Мантуров Татархан
```

Рис. 3.14: Проверяем правильность

Создаем копию файла lab5-2.asm

Создаем копию файла lab5-2.asm

Рис. 3.15: Создаем копию файла lab5-2.asm

Редактируем файл, чтобы введенный текст с клавиатуры выводился в консоль

```
%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
```

Рис. 3.16: Редактируем файл

Транслируем файл и запускаем

Проверяем правильность программы

Рис. 3.17: Проверяем правильность программы

## 4 Выводы

Мы приобрели навыки работы с Midnight Commander и освоили инструкции `mov`.