

Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Козина Дарья Александровна

Содержание

Цель работы	4
Задание.....	5
Выполнение лабораторной работы.....	6
Выполнение заданий для самостоятельной работы	11
Выводы	15

Список иллюстраций

Команда mc	6
Открытый mc	6
Перемещение между директориями	6
Созданный каталог lab5. Создание файла lab5-1.asm	7
Редактор программы	7
Файл с программой	7
Трансляция файла в объектный. Выполнение компоновки. Запуск полученного файла...	8
Установка файла.....	8
Директории	8
Копирование файла.....	8
Копирование файла.....	9
Редактирование файла.....	9
Исполнение файла	9
Отредактированный файл	9
Исполнение файла	10
Копирование файла.....	11
Редактирование файла.....	11
Исполнение файла	12
Копирование файла.....	13
Редактирование файла.....	13
Исполнение файла	14

Цель работы

Приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера mov и int.

Задание

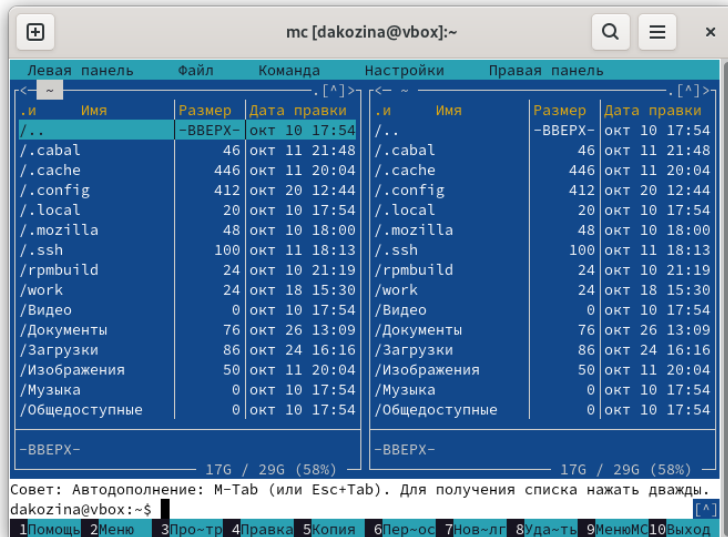
1. Основы работы с тс;
2. Структура программы на языке ассемблера NASM;
3. Подключение внешнего файла;
4. Выполнение заданий для самостоятельной работы.

Выполнение лабораторной работы

С помощью команды `mc` открываем Midnight Commander (рис. [-@fig:001], [-@fig:002]).

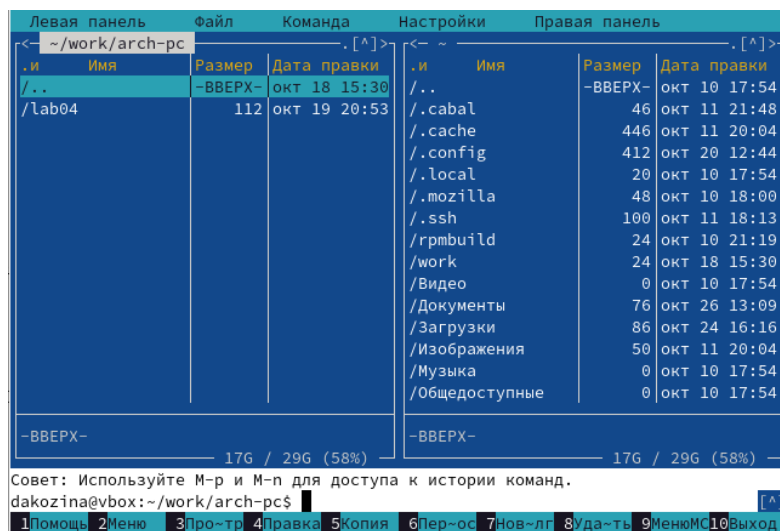
```
dakozina@vbox:~$ mc
```

Команда `mc`



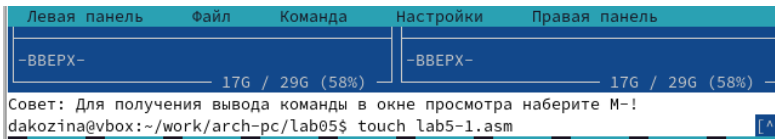
Открытый `mc`

С помощью клавиш переходим в каталог `~/work/arch-pc` (рис. [-@fig:003]).



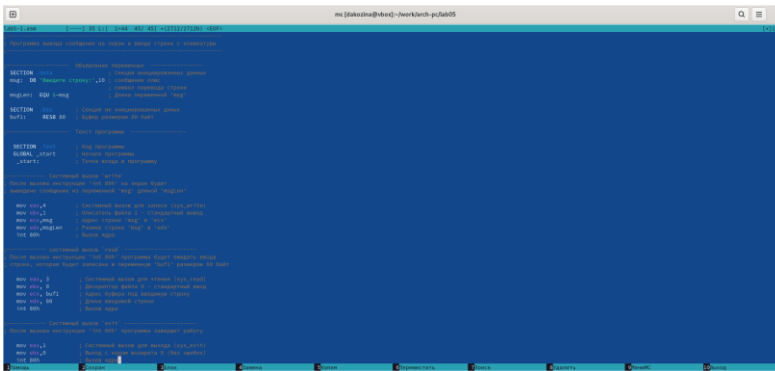
Перемещение между директориями

С помощью функциональной клавиши F7 создадим папку lab5 и перейдем в созданный каталог. Пользуясь строкой ввода, с помощью команды touch создадим файл lab5-1.asm (рис. [-@fig:004]).



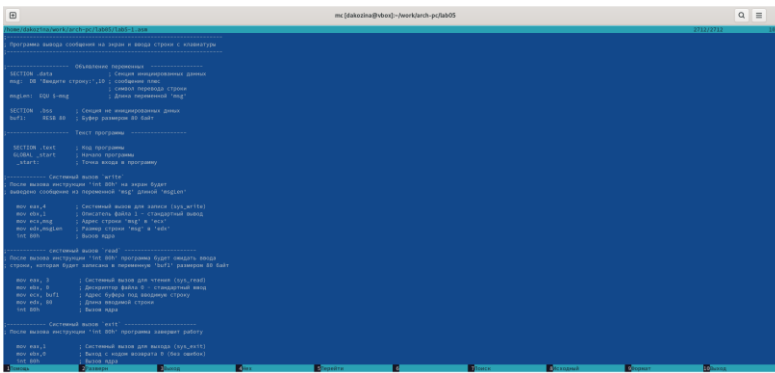
Созданный каталог lab5. Создание файла lab5-1.asm

С помощью функциональной клавиши F4 откроем созданный файл для редактирования во встроенном редакторе mcedit. Введем текст программы, сохраним изменения и закроем файл (рис. [-@fig:005]).



Редактор программы

С помощью функциональной клавиши F3 откроем файл для просмотра и убедимся, что файл содержит текст программы (рис. [-@fig:006]).



Файл с программой

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим полученный файл. На запрос ввести строку, введем ФИО (рис. [-@fig:007]).

```
dakozina@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
dakozina@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
ld: невозможно найти lab5-1.0: Нет такого файла или каталога
dakozina@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
dakozina@vbox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Козина Дарья Александровна
dakozina@vbox:~/work/arch-pc/lab05$
```

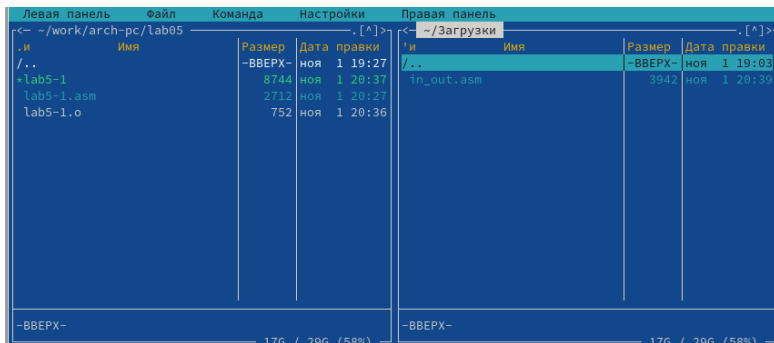
Трансляция файла в объектный. Выполнение компоновки. Запуск полученного файла

Скачаем файл in_out.asm со страницы курса в ТУИС (рис. [-@fig:008]).



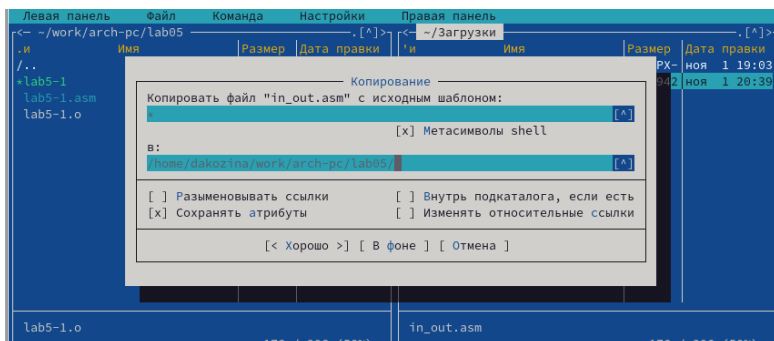
Установка файла

В одной панели mc откроем каталог с файлом lab5-1.asm, в другой панели каталог со скаченным файлом in_out.asm (рис. [-@fig:009]).



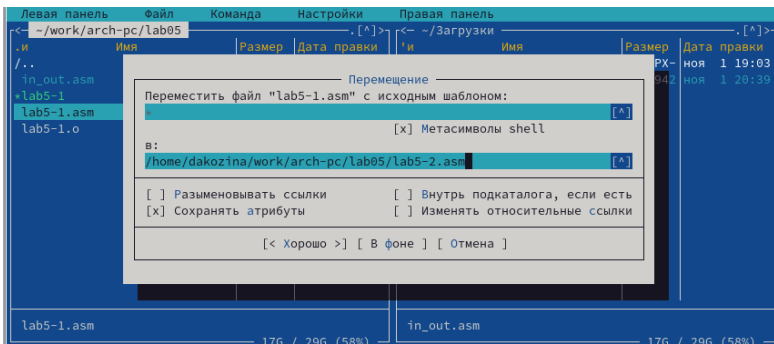
Директории

Скопируем файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5 (рис. [-@fig:010]).



Копирование файла

С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm (рис. [-@fig:011]).



Копирование файла

Исправим текст программы в файле lab5-2.asm с использованием программы из внешнего файла in_out.asm (рис. [-@fig:012]).

```
GNU nano 7.2 /home/dakozina/work/arch-pc/lab05/lab5-2.asm Изм
;-----
; 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 printf ; 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
```

Редактирование файла

Создадим исполняемый файл и проверим его работу (рис. [-@fig:013]).

```
dakozina@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
dakozina@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
dakozina@vbox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Козина Дарья Александровна
dakozina@vbox:~/work/arch-pc/lab05$
```

Исполнение файла

В файле lab5-2.asm заменим подпрограмму printf на sprint (рис. [-@fig:014]).

```
/home/dakozina/work/arch-pc/lab05/lab5-2.asm 1223/1223 100%
;-----
; 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
call quit ; вызов подпрограммы завершения
```

Отредактированный файл

Создадим исполняемый файл и проверим его работу (рис. [-@fig:015]).

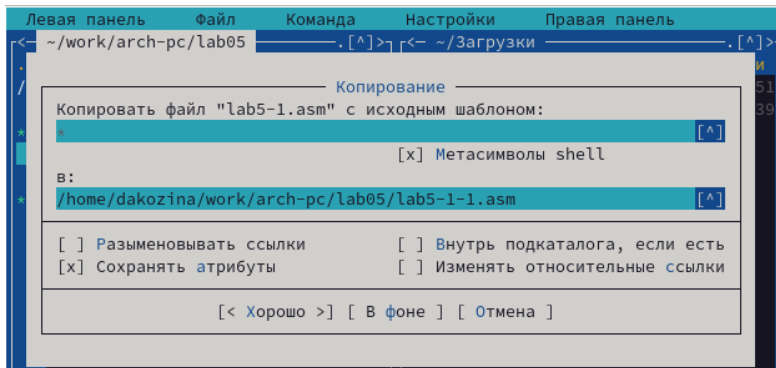
```
dakozina@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
dakozina@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
dakozina@vbox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Козина Дарья Александровна
dakozina@vbox:~/work/arch-pc/lab05$ █
```

Исполнение файла

Разница между первым исполняемым файлом и вторым в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается разница между подпрограммами `sprintLF` и `sprint`.

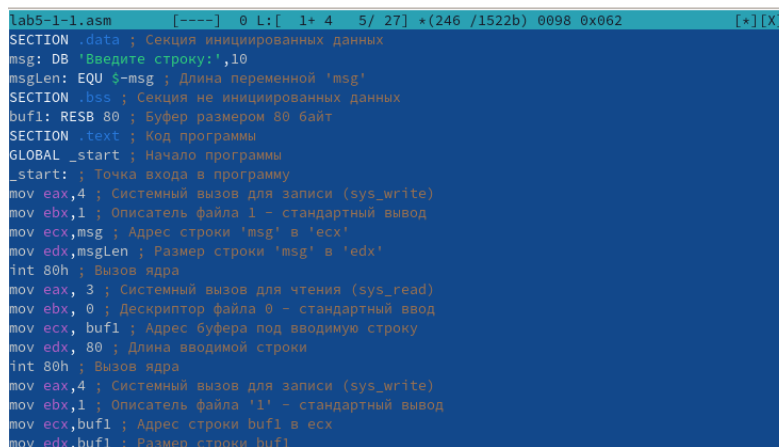
Выполнение заданий для самостоятельной работы

С помощью функциональной клавиши F5 создадим копию файла lab5-1.asm с именем lab5-1-1.asm (рис. [-@fig:016]).



Копирование файла

С помощью функциональной клавиши F4 откроем созданный файл в редакторе и внесем изменения в программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую строку (рис. [-@fig:017]).



Редактирование файла

Код измененной программы:

```
SECTION .data ; Секция инициированных данных
```

```
msg: DB 'Введите строку:',10
```

```
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

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)

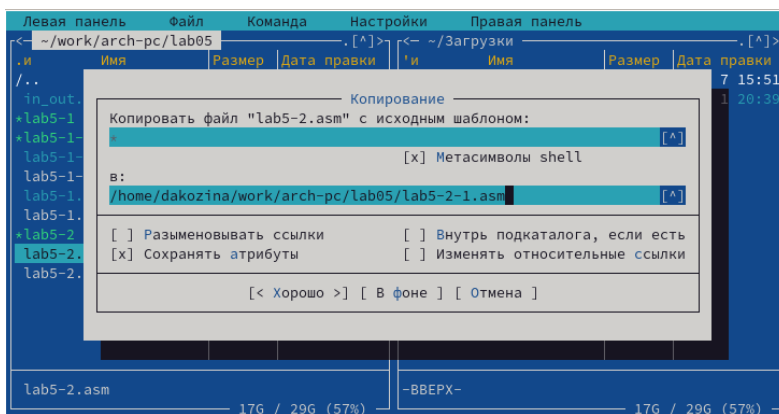
int 80h ; Вызов ядра

Проверим работу полученного файла(рис. [-@fig:018]).

```
dakozina@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
dakozina@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
dakozina@vbox:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Козина
Козина
-
```

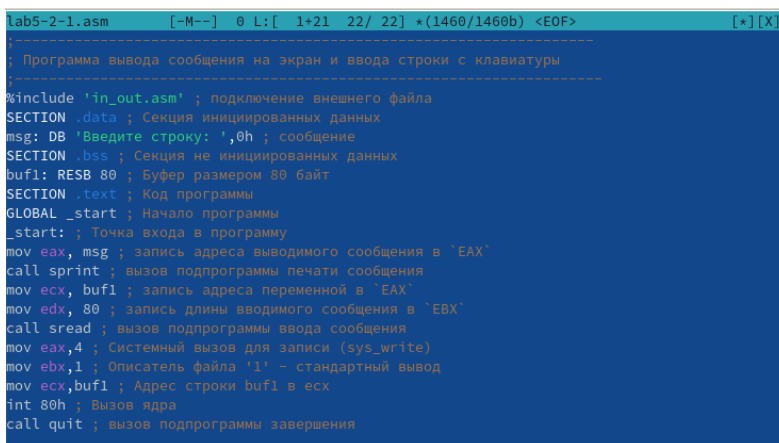
Исполнение файла

С помощью функциональной клавиши F5 создадим копию файла lab5-2.asm с именем lab5-2-1.asm (рис. [-@fig:019]).



Копирование файла

С помощью функциональной клавиши F4 откроем созданный файл в редакторе и внесем изменения в программу с использованием подпрограмм из внешнего файла in_out.asm так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую строку (рис. [-@fig:020]).



Редактирование файла

Код измененной программы:

```
%include 'in_out.asm'
```

```
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

Проверим работу созданного файла (рис. [-@fig:021]).

```
dakozina@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
dakozina@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab4-2-1.o
ld: невозможно найти lab4-2-1.o: Нет такого файла или каталога
dakozina@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
dakozina@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
dakozina@vbox:~/work/arch-pc/lab05$ ./lab5-2-1
Введите строку: Козина
Козина
```

Исполнение файла

Выводы

В ходе лабораторной работы мы приобрели практические навыки работы в Midnight Commander, освоили инструкции языка ассемблера `mov` и `int`.