

Отчёт по лабораторной работе №5

Дисциплина: Архитектура компьютеров

Мухина Ксения Николаевна

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Задания для самостоятельной работы	11
4 Выводы	14
Список литературы	15

Список иллюстраций

2.1	Создание каталога в Midnight Commander.	6
2.2	Создание файла asm.	7
2.3	Текст программы вывода сообщения и ввода строки.	7
2.4	Запуск программы lab5-1.	8
2.5	Копирование внешнего файла в каталог lab05.	8
2.6	Создание копии файла lab5-1.asm.	9
2.7	Изменённый текст программы вывода сообщения и ввода строки.	9
2.8	Запуск программы.	10
2.9	Запуск изменённой программы.	10
3.1	Работа программы lab5-1a.asm.	12
3.2	Работа программы lab5-2a.asm.	13

Список таблиц

1 Цель работы

Приобрести практические навыки работы в Midnight Commander. Освоить инструкции языка ассемблера mov и int.

2 Выполнение лабораторной работы

Далее описываемая работа была выполнена на виртуальной машине Oracle VirtualBox с ОС Linux Ubuntu.

Откроем Midnight Commander, введя mc. Далее перейдём в каталог ~/work/arch-pc и при помощи клавиши F7 создадим в нём каталог lab05.

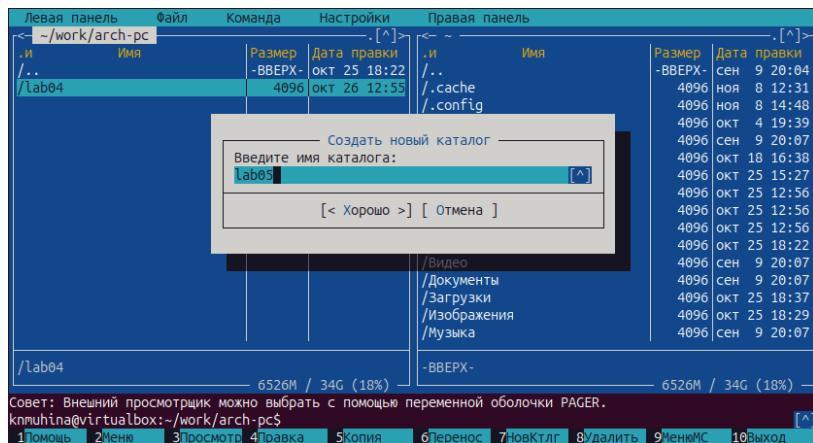


Рисунок 2.1: Создание каталога в Midnight Commander.

Перейдя в созданный каталог, используем строку ввода для выполнения команды touch и создания файла lab5-1.asm.

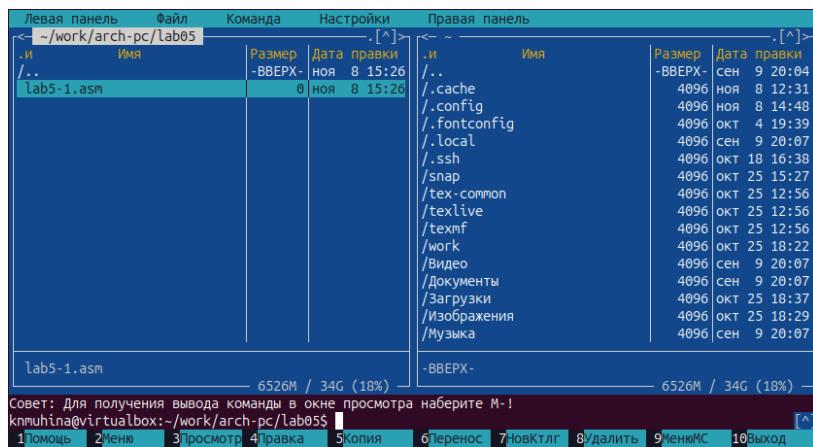


Рисунок 2.2: Создание файла asm.

Файл был успешно создан. При помощи клавиши F4 откроем файл в редакторе mcedit и введём текст будущей программы, далее сохраним изменения и закроем файл.

С помощью клавиши F3 откроем файл для просмотра и убедимся, что файл содержит введённый текст.

```
/home/klyukhina/work/arch-pc/lab05/lab5-1.asm 310/310 100%
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
1Помощь 2Разверн 3Выход 4Нех. 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рисунок 2.3: Текст программы вывода сообщения и ввода строки.

Файл действительно содержит текст будущей программы. Оттранслируем его в объектный файл, выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```

knmuhiina@virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
knmuhiina@virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
knmuhiina@virtualbox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку
Мухина Ксения Николаевна
knmuhiina@virtualbox:~/work/arch-pc/lab05$

```

Рисунок 2.4: Запуск программы lab5-1.

В качестве ответа на запрос мы ввели ФИО автора отчёта.

1. Подключение внешнего файла in_out.asm

Для упрощения написания программ в будущих лабораторных работах мы будем подключать внешний файл in_out.asm, содержащий необходимые подпрограммы. Данные подпрограммы будут использоваться для исключения одинаковых участков кода.

Внешние файлы подключаются с помощью директивы %include. Директива программы должна стоять раньше вызовов подпрограмм из подключаемого файла. Для вызова подпрограммы из внешнего файла используется инструкция call, имеющая вид:

call

где function - имя подпрограммы.

Скачаем файл со страницы курса в ТУИС. Скопируем файл из каталога Загрузки в каталог с файлом lab5-1 с использованием клавиши F5.

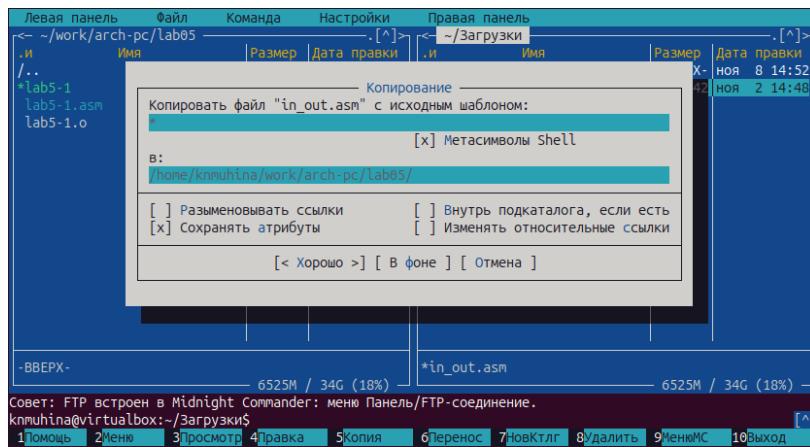


Рисунок 2.5: Копирование внешнего файла в каталог lab05.

После успешного копирования файла с подпрограммами создадим копию файла lab5-1.

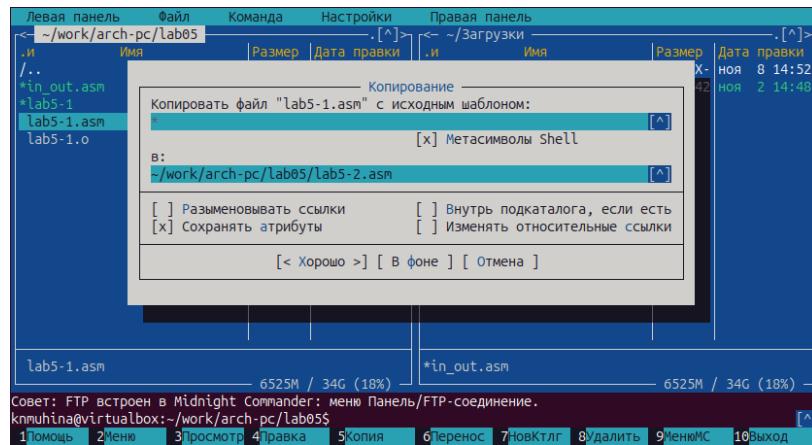


Рисунок 2.6: Создание копии файла lab5-1.asm.

В соответствии с листингом, приведённым в лабораторной работе, исправим текст программы в созданной копии файла с использованием подпрограмм из внешнего файла.

```
/home/knmuhina/work/- pc/lab05/lab5-2.asm [----] 11 L:[ 1+16 17/ 17] *(231 / 231b) <EOF> [*][X]
%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 sprintLF
    mov ecx, buf1
    mov edx, 80
    call sread
    call quit
```

Рисунок 2.7: Изменённый текст программы вывода сообщения и ввода строки.

Создадим исполняемый файл и проверим его работу.

```
knmuhina@virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
knmuhina@virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
knmuhina@virtualbox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Мухина Ксения Николаевна
knmuhina@virtualbox:~/work/arch-pc/lab05$
```

Рисунок 2.8: Запуск программы.

Программа работает успешно.

Далее в файле lab5-2.asm заменим подпрограмму sprintLF на sprint и проведём работу программы.

```
knmuhina@virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
knmuhina@virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
knmuhina@virtualbox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Мухина Ксения Николаевна
knmuhina@virtualbox:~/work/arch-pc/lab05$
```

Рисунок 2.9: Запуск изменённой программы.

Как мы видим, с изменением подпрограммы при запросе ввода с клавиатуры программа не переводит строку, а ожидает ответ на той же строке.

3 Задания для самостоятельной работы

Создадим копию файла lab5-1.asm с названием lab5-1a.asm. Внесём изменения в программу, чтобы после запроса ввода с клавиатуры она выдала введённую строку.

Листинг 1. Программа lab5-1a.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, 4
mov ebx, 1
mov ecx, buf1
mov edx, 80
int 80h
mov eax, 1
mov ebx, 0
int 80h
```

Проверим работу программы.

```
knmuhina@virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab5-1a.asm
knmuhina@virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1a lab5-1a.o
knmuhina@virtualbox:~/work/arch-pc/lab05$ ./lab5-1a
Введите строку:
Мухина
Мухина
knmuhina@virtualbox:~/work/arch-pc/lab05$
```

Рисунок 3.1: Работа программы lab5-1a.asm.

Программа работает корректно.

Далее создадим копию файла lab5-2.asm с названием lab5-2a.asm. Внесём изменения в программу, чтобы после запроса ввода с клавиатуры она выдавала введённую строку.

Листинг 2. Программа lab5-2a.asm

```
%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
```

```
    mov eax, buf1
```

```
    mov edx, 80
```

```
    call sprint
```

```
    call quit
```

Проверим работу программы.

```
knmuhina@virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2a.asm
knmuhina@virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2a lab5-2a.o
knmuhina@virtualbox:~/work/arch-pc/lab05$ ./lab5-2a
Введите строку: Мухина
Мухина
knmuhina@virtualbox:~/work/arch-pc/lab05$
```

Рисунок 3.2: Работа программы lab5-2a.asm.

Программа работает корректно.

4 Выводы

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

Список литературы

1. Файл «Лабораторная работа №5. Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux.pdf»