

Отчёт по лабораторной работе №5

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Грачева Мария Валерьевна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Самостоятельная работа	13
5	Выводы	17
	Список литературы	18

Список иллюстраций

3.1	Команда mc	8
3.2	Каталог arch-pc	8
3.3	Создание папки lab05	8
3.4	Создание файла lab5.asm	9
3.5	Ввод текста программы	9
3.6	Проверка текста	10
3.7	Транслирование, компоновка, запуск lab5.asm	10
3.8	Ввод имени	10
3.9	Работа с панелями	11
3.10	Копирование файла lab5-1.asm с именем lab5-2.asm	11
3.11	Изменение текста, транслирование, компоновка, запуск lab5-2.asm	12
3.12	Замена подпрограммы	12
3.13	Транслирование, компоновка, запуск lab5-2.asm. 2	12
4.1	Копирование файла lab5.asm	13
4.2	Изменение текста. 2	14
4.3	Транслирование, компоновка, запуск lab5с.asm	14
4.4	Копирование файла lab5-2.asm	15
4.5	Изменение текста. 3	15
4.6	Транслирование, компоновка, запуск lab5-2с.asm	16

Список таблиц

1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

2 Теоретическое введение

Midnight Commander (или просто `mc`) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. `mc` является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (`SECTION .text`), секция инициированных (известных во время компиляции) данных (`SECTION .data`) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (`SECTION .bss`).

Простейший диалог с пользователем требует наличия двух функций — вывода текста на экран и ввода текста с клавиатуры. Простейший способ вывести строку на экран — использовать системный вызов `write`. Этот системный вызов имеет номер 4, поэтому перед вызовом инструкции `int` необходимо поместить значение 4 в регистр `eax`. Первым аргументом `write`, помещаемым в регистр `ebx`, задаётся дескриптор файла. Для вывода на экран в качестве дескриптора файла нужно указать 1 (это означает «стандартный вывод», т. е. вывод на экран). Вторым аргументом задаётся адрес выводимой строки (помещаем его в регистр `ecx`, например, инструкцией `mov ecx, msg`). Строка может иметь любую длину. Последним аргументом (т.е. в регистре `edx`) должна задаваться максимальная длина выводимой строки. Для ввода строки с клавиатуры можно использовать аналогичный системный вызов `read`. Его аргументы – такие же, как у вызова `write`, только для «чтения» с клавиатуры используется файловый дескриптор 0

(стандартный ввод). Системный вызов `exit` является обязательным в конце любой программы на языке ассемблер. Для обозначения конца программы перед вызовом инструкции `int 80h` необходимо поместить в регистр `eax` значение 1, а в регистр `ebx` код завершения 0.

3 Выполнение лабораторной работы

Открываю Midnight Commander (рис. 3.1).

```
mvgracheva@dk8n54 ~ $ mc
```

Рис. 3.1: Команда mc

Перехожу в каталог arch-rc (рис. 3.2).

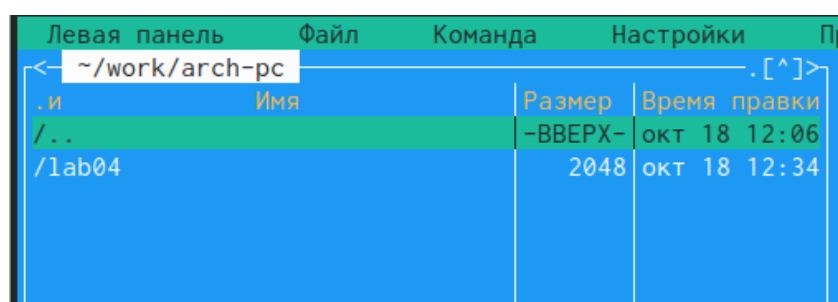


Рис. 3.2: Каталог arch-rc

Создаю папку lab05 (рис. 3.3).

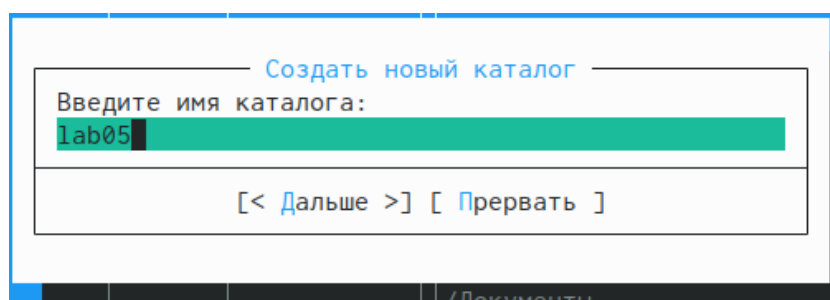


Рис. 3.3: Создание папки lab05

Создаю файл lab5.asm (рис. 3.4).

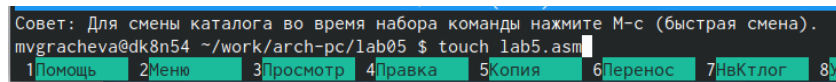


Рис. 3.4: Создание файла lab5.asm

Открываю этот файл и ввожу текст программы (рис. 3.5).

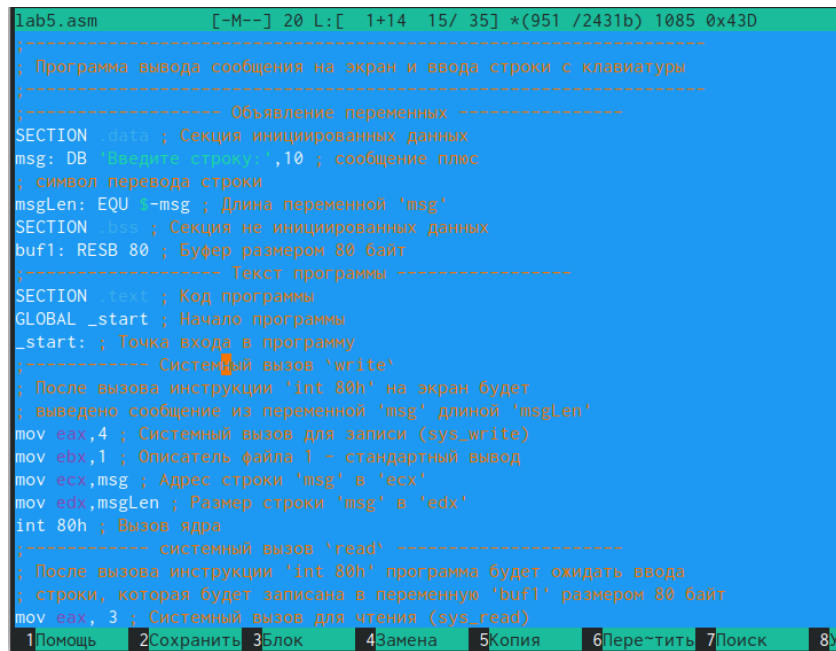


Рис. 3.5: Ввод текста программы

С помощью функциональной клавиши f3 проверяю текст (рис. 3.6).

```

/afs/.dk.sci.pfu.edu.ru/home/m/v/mvgracheva/work/arch-pc/lab05/lab5.asm
;
; 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)

```

Рис. 3.6: Проверка текста

Транслирую текст программы lab5.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл (рис. 3.7).

```

mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5.asm
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5 lab5.o
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ./lab5

```

Рис. 3.7: Транслирование, компоновка, запуск lab5.asm

Ввожу своё ФИО (рис. 3.8).

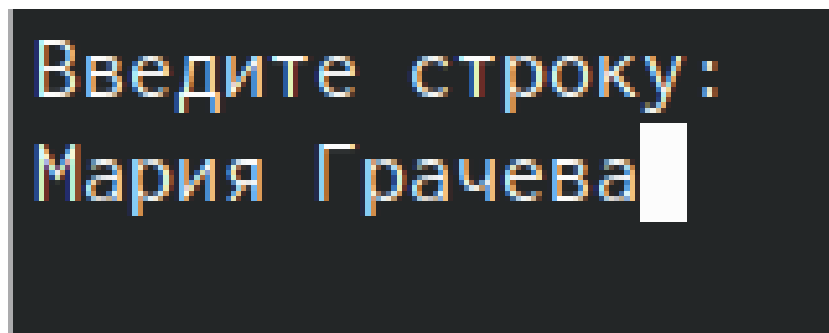


Рис. 3.8: Ввод имени

Открываю два разных каталога одновременно на правой и левой панелях (рис. 3.9).

Левая панель		Файл	Команда	Настройки	Правая панель			
< ~/work/arch-pc/lab05				[^]	< ~/Загрузки			[^]
.и	Имя		Размер	Время правки	.и	Имя	Размер	Время правки
./..			-ВВЕРХ-	ноя 11 12:11	./..		-ВВЕРХ-	ноя 11 12:10
lab5					files.zip		247339	окт 14 15:47
lab5.asm					in_out.asm		3942	ноя 11 12:17
lab5.o			752	ноя 11 12:13				

Рис. 3.9: Работа с панелями

Копирую файл in_out.asm в каталог с файлом lab5.asm с помощью функциональной клавиши F5 (рис. ??).

Левая панель		Файл	Команда	Настройки	Правая панель			
<- ~/work/arch-pc/lab05				[^]	<- ~/Загрузки			[^]
.и	Имя	Размер	Время правки		.и	Имя	Размер	Время правки
./..		-ВВЕРХ-	ноя 11 12:11		./..		-ВВЕРХ-	ноя 11 12:10
in_out.asm		3942	ноя 11 12:17		Files.zip		247339	окт 14 15:47
*lab5		8744	ноя 11 12:14		in_out.asm		3942	ноя 11 12:17
lab5.asm		2431	ноя 11 12:08					
lab5.o		752	ноя 11 12:13					

Создаю копию файла lab5-1.asm с именем lab5-2.asm (рис. 3.10).

Левая панель	Файл	Команда	Настройки	П
< ~/work/arch-pc/lab05			[^]	
.и	Имя		Размер	Время правки
./..			-ВВЕРХ-	ноя 11 12:11
in_out.asm			3942	ноя 11 12:17
*lab5			8744	ноя 11 12:14
lab5-2.asm			2431	ноя 11 12:08
lab5.asm			2431	ноя 11 12:08
lab5.o			752	ноя 11 12:13

Рис. 3.10: Копирование файла lab5-1.asm с именем lab5-2.asm

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm. Создаю исполняемый файл и проверяю его работу

(рис. 3.11).

```
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Грачева Мария Валерьевна
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $
```

Рис. 3.11: Изменение текста, транслирование, компоновка, запуск lab5-2.asm

Заменяю подпрограмму sprintf на sprintf (рис. 3.12).

```
lab5-2.asm [-M--] 11 L:[ 1+12 13/ 17] *(847 /1222b) 0032 0x020
;-----
; 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
```

Рис. 3.12: Замена подпрограммы

Создаю исполняемый файл и проверяю его работу (рис. 3.13).

```
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Грачева Мария
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $
```

Рис. 3.13: Транслирование, компоновка, запуск lab5-2.asm. 2

Разница заключается в переносе строки ввода

4 Самостоятельная работа

1) Создаю копию файла lab5.asm (рис. 4.1).

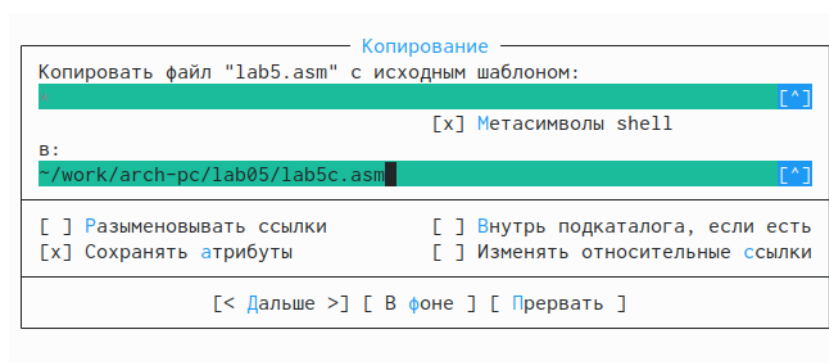


Рис. 4.1: Копирование файла lab5.asm

Вношу изменения в программу (рис. 4.2).

```

lab5c.asm      [----] 12 L: [ 2+23 25/ 29] *(1501/1745b) 0032 0x020
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
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

```

Рис. 4.2: Изменение текста. 2

Получаю исполняемый файл и проверяю его работу (рис. 4.3).

```

mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5c.asm
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5c lab5c.o
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ./lab5c
Введите строку:
Грачева Мария Валерьевна
Грачева Мария Валерьевна

```

Рис. 4.3: Транслирование, компоновка, запуск lab5c.asm

Создаю копию файла lab5-2.asm (рис. 4.4).

Левая панель	Файл	Команда	Настройки	П
<	~/work/arch-pc/lab05			[^]>
.и	Имя	Размер	Время правки	
/..		-BBERX-	ноя 11 12:11	
in_out.asm		3942	ноя 11 12:17	
*lab5		8744	ноя 11 12:14	
*lab5-2		9892	ноя 11 12:37	
lab5-2.asm		1222	ноя 11 12:36	
lab5-2.o		1312	ноя 11 12:37	
lab5-2c.asm		1222	ноя 11 12:36	
lab5.asm		2431	ноя 11 12:08	
lab5.o		752	ноя 11 12:13	
*lab5c		8744	ноя 11 13:28	
lab5c.asm		1745	ноя 11 13:30	
lab5c.o		784	ноя 11 13:28	

Рис. 4.4: Копирование файла lab5-2.asm

Вношу изменения в программу (рис. 4.5).

```
lab5-2c.asm [-M--] 41 L:[ 1+20 21/ 21] *(1459/1459b) <EOF>
;
; 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
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
```

Рис. 4.5: Изменение текста. 3

Получаю исполняемый файл и проверяю его работу (рис. 4.6).

```
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2c.asm
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2c lab5-2c.o
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $ ./lab5-2c
Введите строку: Грачева Мария
Грачева Мария
mvgracheva@dk8n54 ~/work/arch-pc/lab05 $
```

Рис. 4.6: Транслирование, компоновка, запуск lab5-2c.asm

5 Выводы

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

Список литературы