

Отчёт по лабораторной работе №5

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

Корчагин Алексей Павлович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Задания самостоятельной работы	13
5	Выводы	16
	Список литературы	17

Список иллюстраций

3.1	Midnight Commander	8
3.2	Создание Папки	8
3.3	Создание файла	9
3.4	Текст программы	9
3.5	Проверка файл	9
3.6	запуск lab5-1	10
3.7	Ввод данных	10
3.8	Каталоги открытые в МС	10
3.9	Результат копирования	11
3.10	Файл lab5-2.asm	11
3.11	Проверка lab5-2.asm	11
3.12	Заменна sprintLF	12
3.13	Проверка lab5-2 после замены sprintLF	12
4.1	Копирование файла lab5-1.asm	13
4.2	Корректировка файла lab5-12.asm	14
4.3	Работа lab5-12	14
4.4	копия файла lab5-2.asm	15
4.5	Корректировка файла lab5-22.asm	15
4.6	Работа lab5-22	15

Список таблиц

1 Цель работы

Получения опыта работы с 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

С помощью tc открыл Midnight Commander и перешёл в каталог arch-rc(рис. 3.1).

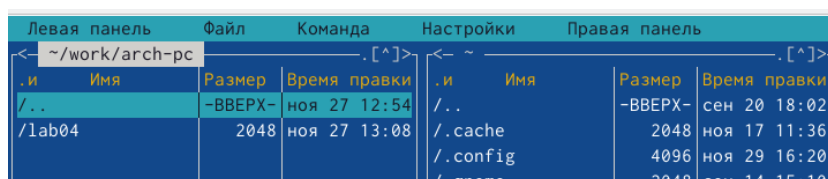


Рис. 3.1: Midnight Commander

Создал Папку lab05(рис.@fig:002).

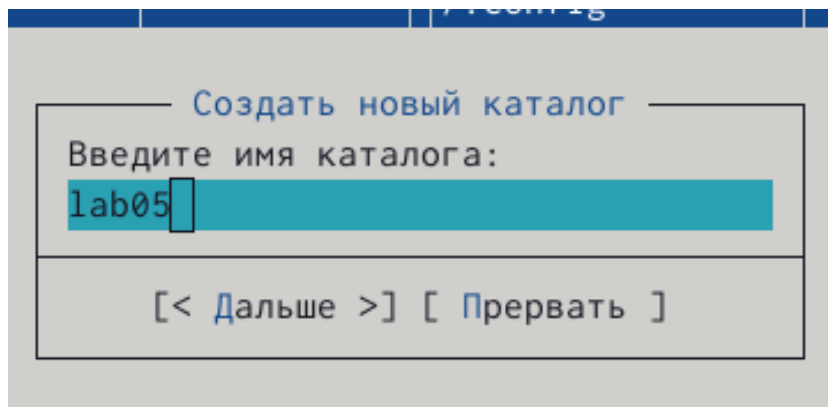


Рис. 3.2: Создание Папки

Создал файл lab5-1.asm(рис.@fig:003).


```
Совет: Формат списка файлов может быть изменен; наберите "man mc" для деталей.  
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ touch lab5-1.asm
```

Рис. 3.3: Создание файла

Ввёл текст программы в файл(рис.@fig:004).

```
/afs/.dk.sci.pfu.edu.ru/home/a/p/apkorchagin/work/arch-pc/lab05/lab5-1.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
```

Рис. 3.4: Текст программы

Проверяю файл с помощью f3(рис. 3.5).

```
/afs/.dk.sci.pfu.edu.ru~ch-pc/lab05/lab5-1.asm 1448/2432 5  
;-----  
; 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
```

Рис. 3.5: Проверка файл

Транслировал текст программы lab5-1.asm в объектный файл. Выполнял компоновку объектного файла и запустил получившийся исполняемый файл(рис.@fig:006).

```
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ./lab5-1
```

Рис. 3.6: запуск lab5-1

Ввёл своё имя и фамилию(рис.@fig:007).



Рис. 3.7: Ввод данных

Открыл два разных каталога одновременно на правой и левой панелях(рис.@fig:008).

Левая панель			Файл	Команда	Настройки	Правая панель		
<- ~/work/arch-pc/lab05						<- ~/Загрузки		
.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки	
/..		-ВВЕРХ-	ноя 29 16:45	/..		-ВВЕРХ-	ноя 29 17:03	
*lab5-1		8744	ноя 29 17:05	in_out.asm		3942	ноя 29 17:11	
lab5-1.asm		2432	ноя 29 16:57					
lab5-1.o		752	ноя 29 17:03					

Рис. 3.8: Каталоги открытые в МС

Копирую файл in_out.asm в каталог lab05 с помощью F5 (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 29 16:45		/..			ноя 29 17:03
in_out.asm			3942	ноя 29 17:11		in_out.asm			3942
*lab5-1			8744	ноя 29 17:05					
lab5-1.asm			2432	ноя 29 16:57					
lab5-1.o			752	ноя 29 17:03					

Рис. 3.9: Результат копирования

Создал копию файла lab5-1.asm(рис.@fig:010).

Левая панель			Файл	Команда
<- ~/work/arch-pc/lab05				. [^]>
.и	Имя	Размер	Время правки	
/..			-ВВЕРХ-	ноя 29 16:45
in_out.asm			3942	ноя 29 17:11
*lab5-1			8744	ноя 29 17:05
lab5-1.asm			2432	ноя 29 16:57
lab5-1.o			752	ноя 29 17:03
lab5-2.asm			2432	ноя 29 16:57

Рис. 3.10: Файл lab5-2.asm

Исправил текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm. Создал исполняемый файл и проверел его работу (рис.@fig:011).

```

apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ./lab-2
bash: ./lab-2: Нет такого файла или каталога
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Корчагин Алексей Павлович
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $

```

Рис. 3.11: Проверка lab5-2.asm

Заменяю подпрограмму `sprintLF` на `sprint`(рис. @fig:012).

```
;-----  
; 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  
call quit ; вызов подпрограммы завершения
```

Рис. 3.12: Замена `sprintLF`

Создал файлы и проверил работу(рис. 3.13).

```
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm  
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o  
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ./lab5-2  
Введите строку: Корчагин Алексей  
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $
```

Рис. 3.13: Проверка `lab5-2` после замены `sprintLF`

4 Задания самостоятельной работы

Создал копию файла lab5-1.asm(рис.@fig:014).

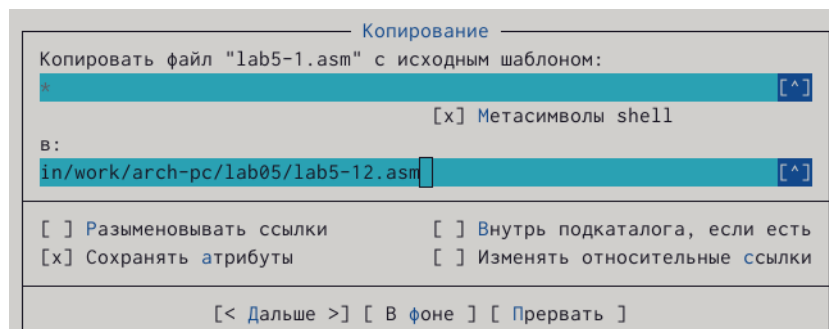


Рис. 4.1: Копирование файла lab5-1.asm

Внёс изменения в программу(рис.@fig:015).

```

; 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
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,4 ; Системный вызов для выхода (sys_exit)
mov ebx,1
mov ecx,buf1
mov,edx buf1.[]
int 80h ; Вызов ядра
mov eax,1
mov,ebx,0
int 80h

```

Рис. 4.2: Корректировка файла lab5-12.asm

Получил исполняемый файл и проверил его работу(рис.@fig:016).

```

apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ nasm -f elf lab5-12.asm
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-12 lab5-12.o
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ./lab5-12
Введите строку:
Корчагин Алексей Павлович
Корчагин Алексей Павлович
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ █

```

Рис. 4.3: Работа lab5-12

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

*lab5-12	8748	ноя 29 17:56
lab5-12.asm	1933	ноя 29 17:56
lab5-12.o	784	ноя 29 17:56
*lab5-2	9092	ноя 29 17:41
lab5-2.asm	1223	ноя 29 17:40
lab5-2.o	1312	ноя 29 17:41
lab5-22.asm	1223	ноя 29 17:40

Рис. 4.4: копия файла lab5-2.asm

Внёс изменения в программу(рис.@fig:018).

```

;-----
; 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
mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit ; вызов подпрограммы завершения

```

Рис. 4.5: Корректировка файла lab5-22.asm

Получил исполняемый файл и проверил его работу(рис.@fig:019).

```

apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ nasm -f elf lab5-22.asm
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-22 lab5-22.o
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ ./lab5-22
Введите строку: Корчагин Алексей Павлович
Корчагин Алексей Павлович
apkorchagin@dk8n68 ~/work/arch-pc/lab05 $ 

```

Рис. 4.6: Работа lab5-22

5 Выводы

Я приобрел опыт работы с Midnight Commander. Освоил инструкции языка ассемблера `mov` и `int`.

Список литературы