

# **Отчёт по лабараторной работе No5**

**Основы работы с Midnight Commander**

Бурыкина Софья Дмитриевна

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	14
5	Выводы	17
	Список литературы	18

# Список иллюстраций

3.1	Midnight Commander . . . . .	7
3.2	Перемещение между дирикториями . . . . .	8
3.3	Каталог создан . . . . .	9
3.4	Файл создан . . . . .	10
3.5	папо с программой . . . . .	11
3.6	Программа вывода . . . . .	12
3.7	Запкск пограммы . . . . .	12
3.8	Запкск пограммы . . . . .	13
4.1	Копирование файла . . . . .	14
4.2	Запуск программы . . . . .	15
4.3	Программа . . . . .	15
4.4	Запуск программы . . . . .	15
4.5	Программа . . . . .	16

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - приобретение практических навыков работы в Midnight Commander и освоение инструкций языка ассемблера `mov` и `int`

## **2 Теоретическое введение**

Более подробно об Unix см. в [1–6].

### 3 Выполнение лабораторной работы

Открыла Midnight Commander (рис. 3.1).

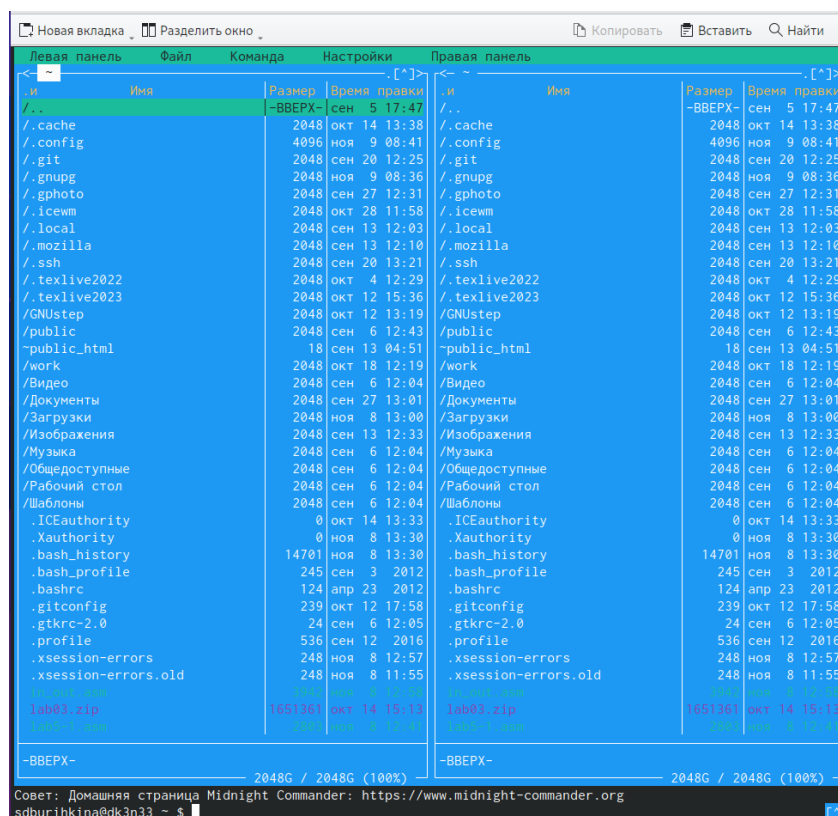


Рис. 3.1: Midnight Commander

Перешла в каталог study (рис. 3.2).

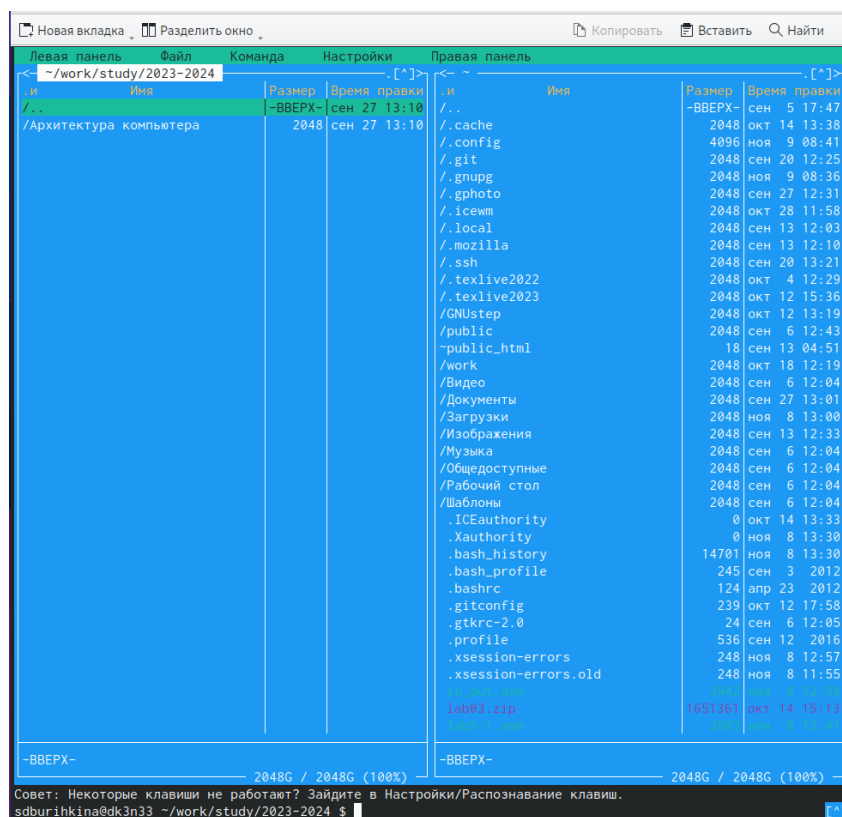


Рис. 3.2: Перемещение между дирикториями

Создала каталог lab05 (рис. 3.3).



Левая панель				Правая панель			
~/work/arch-pc							
Имя	Размер	Время	Правки	Имя	Размер	Время	Правки
./	-ВВЕРХ-	окт 18 12:19		./	-ВВЕРХ-	сен 5 17:3	
./lab04	2048	окт 18 13:18		./cache	2048	окт 14 13:3	
./lab05	2048	ноя 8 12:42		./config	4096	ноя 8 12:3	
				./git	2048	сен 20 12:3	
				./gnupg	2048	ноя 8 11:3	
				./gphoto	2048	сен 27 12:3	
				./icewm	2048	окт 28 11:3	
				./local	2048	сен 13 12:3	
				./mozilla	2048	сен 13 12:3	
				./ssh	2048	сен 20 13:3	
				./texlive2022	2048	окт 4 12:3	
				./texlive2023	2048	окт 12 15:3	
				/GNUstep	2048	окт 12 13:3	
				/public	2048	сен 6 12:3	
				~public_html	18	сен 13 04:3	
				/work	2048	окт 18 12:3	
				/Видео	2048	сен 6 12:3	
				/Документы	2048	сен 27 13:3	
				/Загрузки	2048	окт 25 12:3	
				/Изображения	2048	сен 13 12:3	
				/Музыка	2048	сен 6 12:3	
				/Общедоступные	2048	сен 6 12:3	
				/Рабочий стол	2048	сен 6 12:3	
				/Шаблоны	2048	сен 6 12:3	
				.ICEauthority	0	окт 14 13:3	
				.Xauthority	102	ноя 8 11:3	
				.bash_history	14553	ноя 8 12:3	
				.bash_profile	245	сен 3 20:3	
				.bashrc	124	апр 23 20:3	
				.gitconfig	239	окт 12 17:3	
				.gitkr-2.0	24	сен 6 12:3	
				.profile	536	сен 12 20:3	
				.xsession-errors	248	ноя 8 11:3	
				.xsession-errors.old	248	ноя 1 11:3	
				lab03.zip	1851261	окт 14 15:3	

Рис. 3.3: Каталог создан

Пользуясь строкой ввода и командой touch создала файл lab5-1.asm (рис. 3.4).



Рис. 3.4: Файл создан

Открыла редактор папо (рис. 3.5).

```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/s/d/sdburikhina/work/arch-pc/lab05/lab5-1.asm Изменён
; 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

Сохранить изменённый буфер? 
Y Да
N Нет 
```

Рис. 3.5: nano с программой

Открыла файл lab5-1.asm написала программу (рис. 3.6).

```

/afs/.dk.sci.pfu.edu.ru/home/s/d/sdburikhina/work/arch-pc/lab05/lab5-1.asm Y 2803/2803 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                    ; Дескриптор 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. 3.6: Программа вывода

Создаю объектный файл lab5-1.o, выполняю компоновку объектного файла и запускаю исполняемый файл (рис. 3.7).

```

sdburikhina@dk4n64 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
sdburikhina@dk4n64 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
sdburikhina@dk4n64 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Бурыкина Софья Дмитриевна
sdburikhina@dk4n64 ~/work/arch-pc/lab05 $

```

Рис. 3.7: Запуск программы

Скачала файл in\_out.asm создала копию lab5-1.asm с именем lab5-2.asm. В файле lab5-2.asm заменила подпрограмму sprintLF на sprint. Создала исполняемый файл и проверила его работу (рис. 3.8). Ответ на вопрос: Отличается строкой вывода строки.

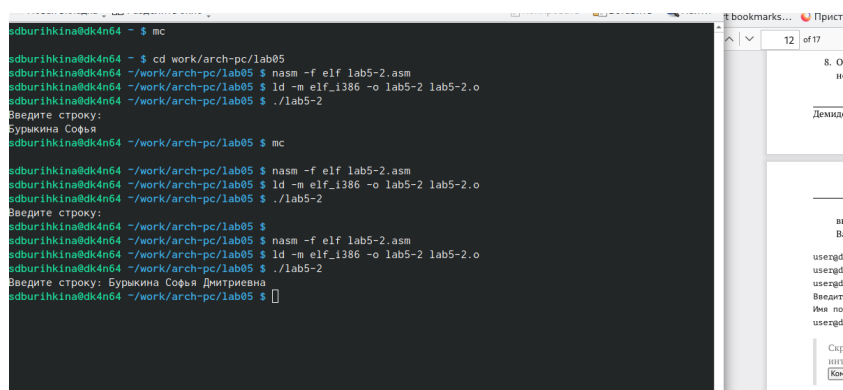


Рис. 3.8: Запуск программы

# 4 Самостоятельная работа

Копирую файл lab5-1.asm с именем lad5-3.asm (рис. 4.1).

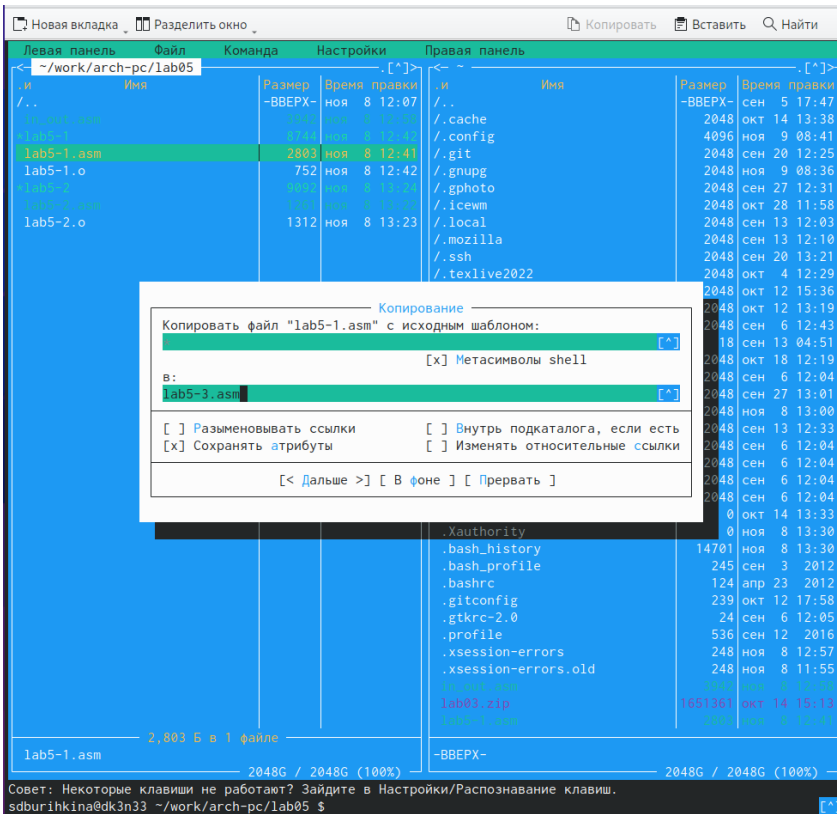


Рис. 4.1: Копирование файла

Изменила код программы и запустила исполняемый файл (рис. 4.2).

```

sdburikhina@dk3n33 ~ $ cd work/arch-pc/lab05
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Бурыкина Софья самостоятельная
Бурыкина Софья самостоятельная
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $

```

Рис. 4.2: Запуск программы

Программа 1 ( рис. 4.3).

```

lab5-3.asm [----] 20 L: [ 1+32 33/ 35] *(2006/2008b) 0010 0x00A
;----- Объявление переменных -----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
SECTION .data ; Секция инициализированных данных
msg: DB "lab5-3\n" ; Сообщение
; Размер строки
msgLen: EQU $-msg ; Длина переменной 'msg'
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, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Файловый дескриптор 1 - стандартный вывод
mov ecx, msg ; Адрес строки 'msg' в 'data'
mov edx, msgLen ; Размер строки 'msg' в 'data'
int 80h ; Выход ядра

mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Файловый дескриптор 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводную строку
mov edx, 80 ; Длина вводной строки
int 80h ; Выход ядра

mov eax, 4 ; Системный вызов для записи (sys_write)
mov ebx, 1 ; Файловый дескриптор 1 - стандартный вывод
mov ecx, buf1 ; Адрес строки 'buf1' в 'bss'
mov edx, buf1 ; Размер строки 'buf1'
int 80h ; Выход ядра

mov eax, 1 ; Системный вызов для выхода (sys_exit)
mov ebx, 0 ; Выход с кодом возврата 0 (ока ошибок)
int 80h ; Выход ядра

```

Рис. 4.3: Программа

Скопировала файл lab5-2.asm с именем lab5-4.asm и скомпоновала его в исполняемый файл, запуская исполняемый файл (рис. 4.4).

```

sdburikhina@dk3n33 ~ $ cd work/arch-pc/lab05
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку: Бурыкина Софья самостоятельная
Бурыкина Софья самостоятельная
sdburikhina@dk3n33 ~/work/arch-pc/lab05 $

```

Рис. 4.4: Запуск программы

Программа 2 ( рис. 4.5).

```

lab5-4.asm          [----] 41 L: [ 1+ 5   6/ 22] *(470 /1456b) 0010 0x00A
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;~~~~~
Kinclude  _kernel_ ; подключение внешнего файла
SECTION  _text_ ; Секция инициализированных данных
msg: DB  "Привет!\n", 0h ; сообщение
SECTION  _data_ ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION  _start_ ; 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 ecx, 80 ; запись длины выводимого сообщения в EDI
call streed ; вызов подпрограммы ввода сообщения
mov ecx, 4 ; десятичный разряд для записи (edx, write)
mov edx, 1 ; опускатель файла '1' - стандартный вывод
mov ecx, buf1 ; адрес строки buf1 в ecx
int 80h ; вызов ядра
call quit ; вызов подпрограммы завершения

```

Рис. 4.5: Программа



## 5 Выводы

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

## Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.  
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.