

Лабораторная работа №5.

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

Саидова Маржина Авдулвохидовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	Midnight Commander	8
3.2	Создать папку	9
3.3	Создать файл	9
3.4	Открыть файлы	10
3.5	Компановка и запуск	10
3.6	Создание копии файла	11
3.7	Листинг 5.2	11
3.8	sprint	11
3.9	Выполнение алгоритма	12
3.10	Изменение в тексте	13
3.11	Выполнение алгоритма	14
3.12	Изменение в тексте	14

Список таблиц

2.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

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

2 Задание

1. Открыть Midnight Commander
2. Перейти в каталог `~/work/arch-рс` созданный при выполнении лабораторной работы №4
3. Создать папку `lab05` и перейти в созданный каталог.
4. Создать файл `lab5-1.asm`
5. Открыть файл `lab5-1.asm` для редактирования во встроенном редакторе.
6. Ввести текст программы из листинга 5.1, сохранить изменения и закрыть файл.
7. Оттранслировать текст программы `lab5-1.asm` в объектный файл. Выполнить компоновку объектного файла и запустить получившийся исполняемый файл.
8. Скачать файл `in_out.asm` со страницы курса в ТУИС.
9. Создайте копию файла `lab5-1.asm` с именем `lab5-2.asm`.
10. Исправить текст программы в файле `lab5-2.asm` с использованием подпрограмм из внешнего файла `in_out.asm` в соответствии с листингом 5.2. Создать исполняемый файл и проверить его работу.
11. В файле `lab5-2.asm` заменить подпрограмму `sprintLF` на `sprint`. Создать исполняемый файл и проверьте его работу.
12. Выполнить самостоятельную работу # Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 2.1 приведено краткое описание стандартных каталогов

Unix.

Таблица 2.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

3 Выполнение лабораторной работы

1. Открытие Midnight Commander

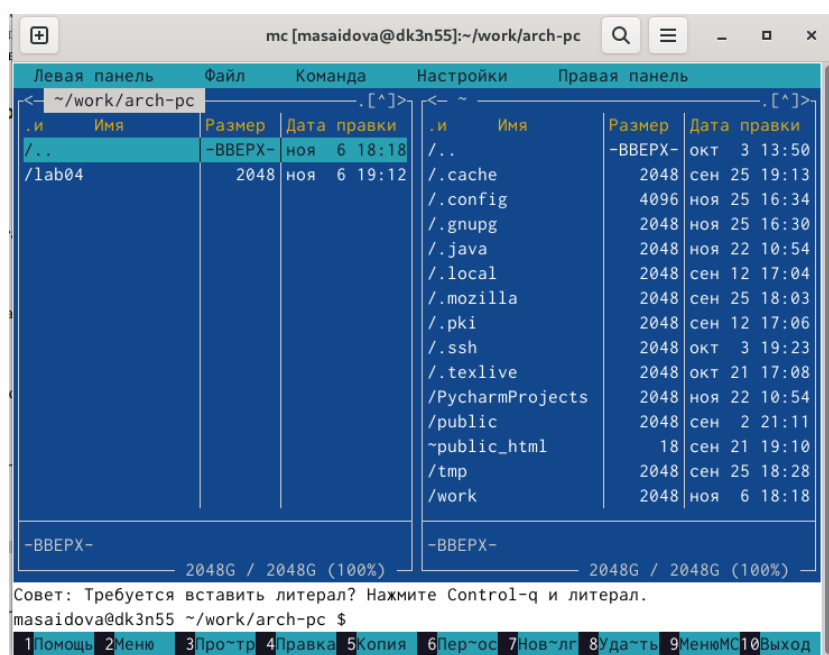


Рис. 3.1: Midnight Commander

2. Создание папки lab05 и переход в созданный каталог

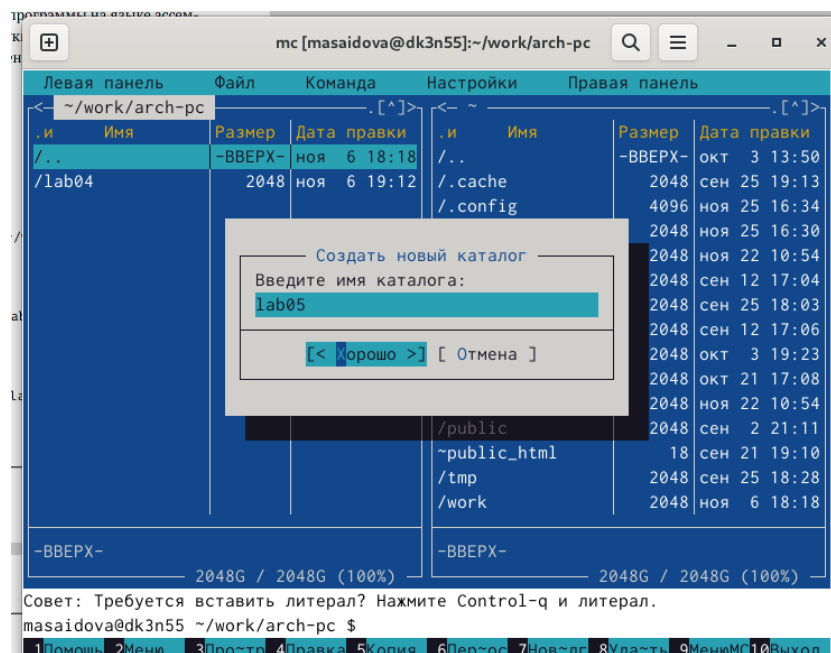


Рис. 3.2: Создать папку

3. Создание файла lab5-1.asm и открытие его

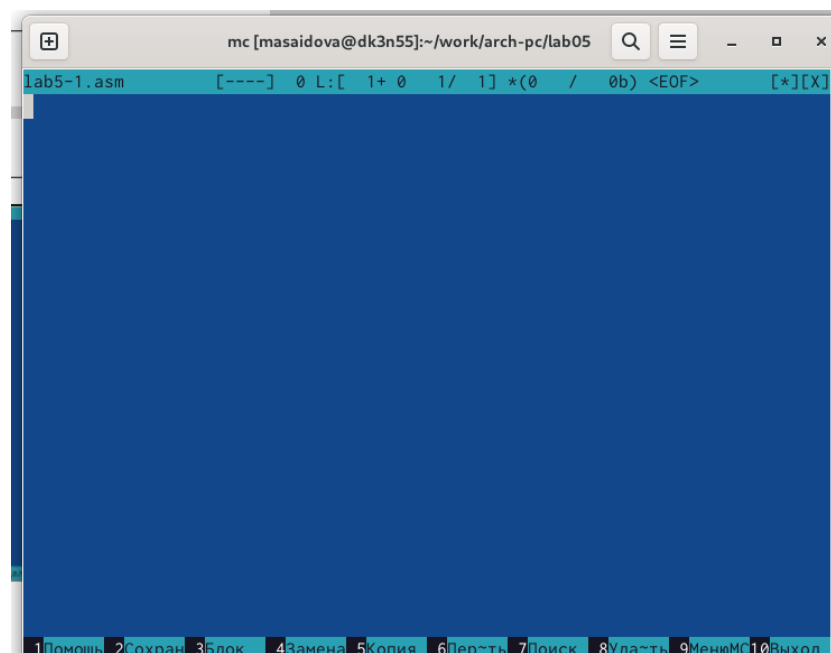
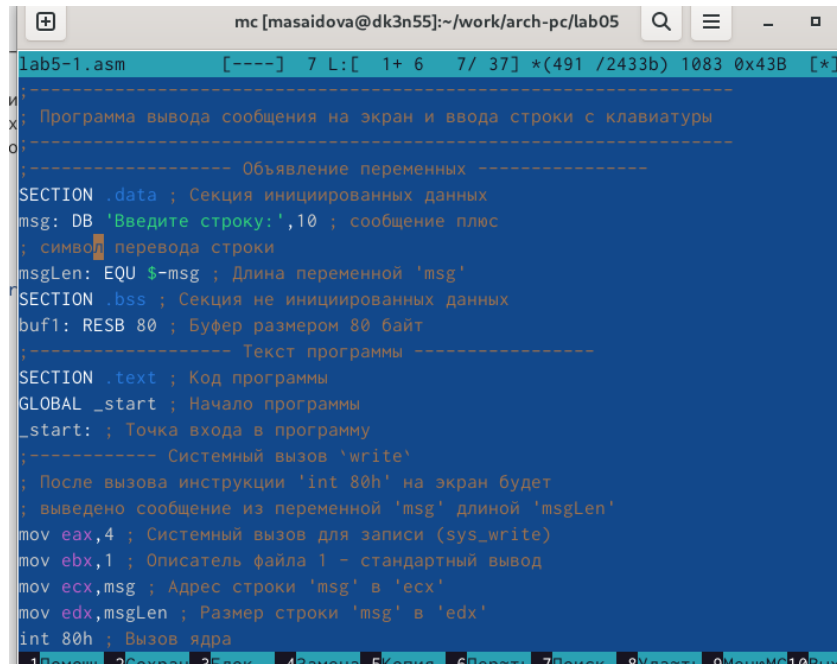


Рис. 3.3: Создать файл

4. Открытие файла lab5-1.asm для редактирования во встроенном редакторе

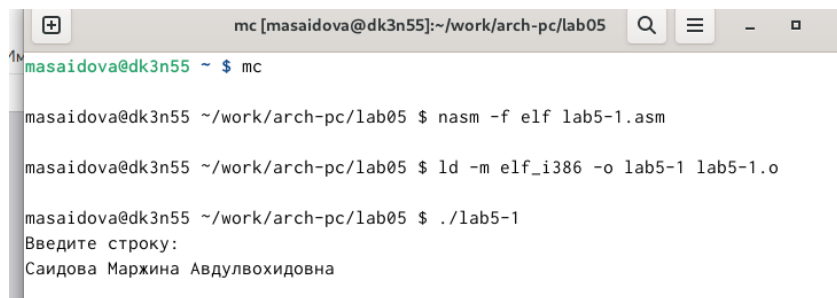
и введение текста программы из листинга 5.1



```
lab5-1.asm [----] 7 L: [ 1+ 6 7/ 37] *(491 /2433b) 1083 0x43B [*]
; 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. 3.4: Открыть файлы

5. Оттранслирование текста программы lab5-1.asm в объектный файл. Выполнение компоновки объектного файла и запуск получившегося исполняемого файла.



```
masaidova@dk3n55 ~ $ mc
masaidova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
masaidova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
masaidova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Саидова Маржина Авдулвохидовна
```

Рис. 3.5: Компоновка и запуск

6. Создание копии файла lab5-1.asm с именем lab5-2.asm.

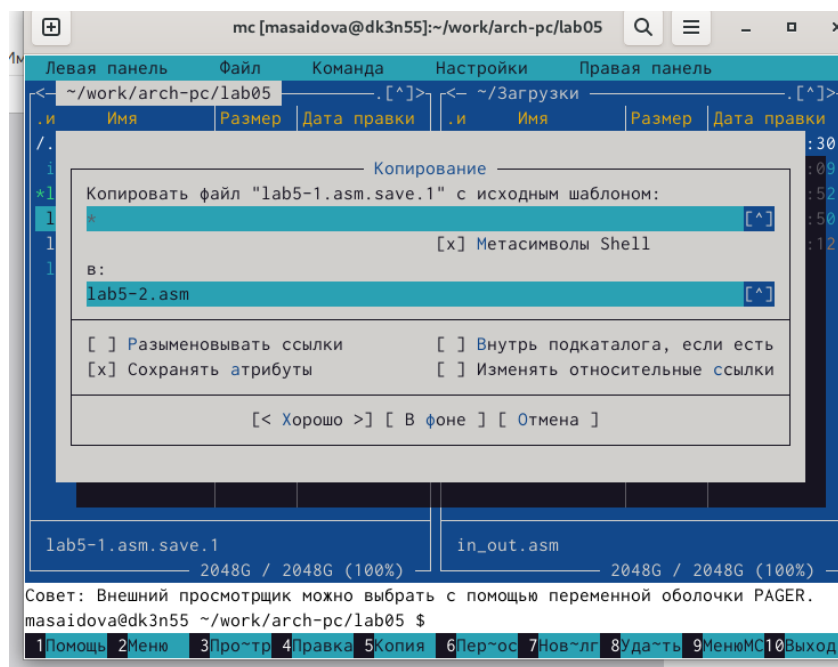


Рис. 3.6: Создание копии файла

7. Исправление текста программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2. Создание исполняемого файла и проверка его работы.

```
masaidova@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Саидова Маржина
masaidova@dk3n35 ~/work/arch-pc/lab05 $
```

Рис. 3.7: Листинг 5.2

8. В файле lab5-2.asm замена подпрограммы sprintLF на sprint. Создание исполняемого файла и проверка его работы.

```
masaidova@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку: Саидова Маржина
```

Рис. 3.8: sprint

9. Самостоятельная работа:

9.1 Создание копии файла lab5-1.asm. Внесение изменений для выполнения алгоритма: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.

```
masaidova@dk3n35 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку:
Саидова
Саидова
masaidova@dk3n35 ~/work/arch-pc/lab05 $
```

Рис. 3.9: Выполнение алгоритма

```

;----- 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 в
; строки, которая будет записана в переменную 'buf1' размером
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 '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,buf1 ; Адрес строки 'msg' в 'ecx'
mov edx,80 ; Размер строки 'msg' в 'edx'
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. 3.10: Изменение в тексте

9.2 Создание копии файла lab5-2.asm. Внесение изменений для выполнения алгоритма: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.

```

masaidova@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-5.asm
masaidova@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-5 lab5-5.o
masaidova@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-5
Введите строку: Саидова
Саидова

```

Рис. 3.11: Выполнение алгоритма

```

lab5-5.asm [----] 0 L:[ 1+23 24/ 24] *(1381/1381b) <EOF>
;
; 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, buf1 ; запись адреса переменной в 'EAX'
call sprint ; вызов подпрограммы печати сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.12: Изменение в тексте

4 Выводы

В процессе выполнения лабораторной работы приобрела практические навыки работы в Midnight Commander.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.