

**Отчет Лабораторная работа No5. Основы
работы с Midnight Commander (mc).
Структура программы на языке
ассемблера NASM. Системные вызовы в
ОС GNU Linux**

Простейший вариант

Матвеева Анастасия Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Ход работы	7
4	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	МС	7
3.2	МС	8
3.3	МС	8
3.4	МС	11
3.5	МС	12
3.6	МС	12
3.7	МС	13
3.8	МС	14

Список таблиц

1 Цель работы

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

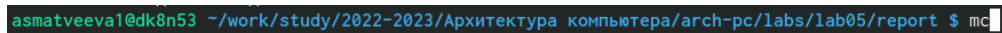
2 Задание

1. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран. Архитектура ЭВМ
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab6-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
4. Создайте исполняемый файл и проверьте его работу.

3 Ход работы

Лабораторная работа

1. Открываем Midnight Commander

A terminal window with a dark background. The prompt is 'asmatveeva1@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/report \$'. The user has entered 'mc' and the cursor is at the end of the command.

```
asmatveeva1@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/report $ mc
```

Рис. 3.1: MC

2. Используя клавиши переходим в каталог ~/work/arch-pc созданный при выполнении лабораторной работы No4.

Левая панель				Правая панель			
Файл		Команда	Настройки	Файл		Команда	Настройки
Имя	Размер	Время	правки	Имя	Размер	Время	правки
..	-BBERPX-	окт 26 12:42		..	-BBERPX-	сен 6 12:58	
/.git	2048	ноя 23 13:18		/.cache	2048	сен 9 09:36	
/config	2048	окт 26 12:43		/.config	4096	дек 7 10:44	
/labs	2048	окт 26 12:43		/.emacs.d	2048	ноя 23 11:36	
/template	2048	окт 26 12:43		/.fltk	2048	окт 19 11:09	
.gitattributes	1765	окт 26 12:43		/.git	2048	сен 21 10:50	
.gitignore	4637	окт 26 12:43		/.gnupg	2048	дек 7 10:38	
.gitmodules	278	окт 26 12:43		/.local	2048	сен 28 11:04	
CHANGELOG.md	2126	окт 26 12:43		/.mozilla	2048	сен 7 11:13	
COURSE	8	окт 26 12:46		/.pki	2048	сен 7 11:19	
LICENSE	18657	окт 26 12:43		/.ssh	2048	сен 21 11:53	
Makefile	815	окт 26 12:43		/.texlive2022	2048	окт 13 13:04	
README_en.md	152	окт 26 12:43		/CC++	2048	ноя 9 11:50	
README_git-flow.md	5653	окт 26 12:43		/GNUstep	2048	сен 28 10:56	
README.md	4477	окт 26 12:43		/parentdir	2048	сен 7 11:59	
prepare	0	окт 26 12:43		/public	2048	сен 2 21:26	
				~public_html	18	сен 2 21:26	
				/temp	2048	сен 10 10:48	
				/tmp	2048	сен 9 10:02	
				/work	2048	окт 26 12:36	
				/Видео	2048	сен 7 11:12	
				/Документы	2048	сен 7 11:12	
				/Загрузки	2048	ноя 23 10:55	
				/Изображения	2048	сен 9 09:33	
				/Музыка	2048	сен 7 11:12	
				/Общедоступные	2048	сен 7 11:12	
				/Рабочий стол	2048	сен 7 11:12	
				/Шаблоны	2048	сен 7 11:12	
				.Xauthority	867	дек 7 10:37	
				.bash_history	13053	ноя 23 13:45	
				.bash_profile	245	сен 3 20:12	
				.bashrc	124	апр 23 20:12	
				.dmrc	32	ноя 2 14:41	
				.gitconfig	248	окт 26 12:33	
				.gtkrc-2.0	24	сен 7 11:12	
-BBERPX- 2048G/2048G (100%)				-BBERPX- 2048G/2048G (100%)			

Рис. 3.2: MC

- С помощью функциональной клавиши F7 создайте папку lab05 и перейдите в созданный каталог

/.lab05	2048	дек 7 10:48		/.cache	2048	сен 9 09:36	
/lab5	2048	дек 7 10:48		/.config	4096	дек 7 10:44	
/presentation	2048	окт 26 12:43		/.emacs.d	2048	ноя 23 11:36	
/report	2048	ноя 23 13:53		/.fltk	2048	окт 19 11:09	

Рис. 3.3: MC

- Создаем файл спомощью touch. Открываем при помощи F4 и редактируем его. Вводим текст


```

;-----
; 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
Архитектура ЭВМ
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

```

```

;-----
; 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
Архитектура ЭВМ
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)
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

```

- Оттранслируем текст программы lab5-1.asm в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос пишем свою ФИО.

```

asmatveeva1@dk3n38 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f
elf lab5-1.asm
lab5-1.asm:30: warning: label alone on a line without a colon might be in error [-w+label-orp
han]
asmatveeva1@dk3n38 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ld

```

```

asmatveeva1@dk3n38 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
asmatveeva1@dk3n38 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-1
Введите строку:
Анастасия Матвеева
asmatveeva1@dk3n38 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $

```

6. Скачиваем файл in_out.asm со страницы курса в ТУИС. Подключаем файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется

Левая панель				Правая панель			
Файл		Команда		Настройки		Правая панель	
<- ...а компьютера/arch-pc/labs/lab05 -.[^]>				<- ~/Загрузки		.[^]>	
.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
/..		-ВВЕРХ-	окт 26 12:43	/..		-ВВЕРХ-	дек 7 12:43
/presentation		2048	окт 26 12:43	/путин - Гoo~Поиск_files		2048	ноя 9 14:34
/report		2048	окт 26 12:43	3.1_БРС_ОсП_НБ.doc		62976	ноя 23 10:55
in_out.asm		3942	дек 7 12:42	in_out.asm		3942	дек 7 12:42
*lab5-1		8772	дек 7 12:38	lab5-1.asm		2362	дек 7 11:42
lab5-1.asm		2362	дек 7 11:42	Л2_Матвеева_отчёт-1.pdf		995526	окт 13 11:57
lab5-1.asm.save		2362	дек 7 12:43	Л2_Матвеева_отчёт-2.md		4874	окт 26 13:17
lab5-1.o		768	дек 7 12:37	Л2_Матвеева_отчёт-2.pdf		995526	окт 26 16:15
				Л2_Матвеева_отчёт-2.zip		679547	окт 26 16:18
				Л2_Матвеева_отчёт.pdf		995526	окт 13 11:57
				путин - Гoo~ Поиск.html		1093982	ноя 9 14:34

Рис. 3.4: МС

7. С помощью функциональной клавиши F6 создаем копию файла lab5-1.asm с именем lab5-2.asm. Выделяем файл lab5-1.asm, нажимаем клавишу F6, вводим имя файла lab5-2.asm и нажимаем клавишу Enter. Далее исправляем текст в файле lab5-2.asm

```

.../a/s/asmatveeval/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/lab5-2.asm
;-----
; 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 sprintLF               ; 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. 3.5: MC

8. Исправляем sprintLF на sprint.

```

;-----
; 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

    call quit                   ; вызов подпрограммы завершения

```

Рис. 3.6: MC

Самостоятельная работа

1. Создаем копию файла lab5-1.asm. Вносим изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.

```
;-----  
; Программа вывода сообщения на экран и ввода строки с клавиатуры  
;-----  
;----- Объявление переменных -----  
SECTION .data ; Секция инициированных данных  
msg: DB 'Введите строку:',10 ; сообщение плюс  
; символ перевода строки  
Архитектура ЭВМ  
msgLen: EQU $-msg ; Длина переменной 'msg'  
SECTION .bss ; Секция не инициированных 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  
  
mov eax,4  
mov ebx,1  
mov ecx, buf1
```

Рис. 3.7: MC

Мы добавили несколько строк для вывода результата нашей программы

2. Выводим результат программы свою фамилию

```

asmatveeva1@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-1
Введите строку: Матвеева
asmatveeva1@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ █

```

Рис. 3.8: MC

3. Делаем тоже самое только подключаем одпрограмм из внешнего файла `in_out.asm`, так чтобы она работала по следующему алгоритму:
 - вывести приглашение типа “Введите строку:”;
 - ввести строку с клавиатуры;
 - вывести введённую строку на экран.

![MC](image/13.png){#fig:013 width=90%}

4. Выводим ответ

```

asmatveeva1@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-2
Введите строку: Matveeva Anastasiya
asmatveeva1@dk8n53 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ █

```

4 Выводы

Мы обучились работать в Midnight Commander и освоили язык ассамблер `mov` and `int`.

Список литературы