

Отчёт по лабораторной работе №5

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

Попутников Егор Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выполнение самостоятельной работы	11
5	Выводы	15

Список иллюстраций

3.1	Создание каталога	7
3.2	Создание файла	8
3.3	Текст программы	8
3.4	Трансляция текста программы	9
3.5	Создание копии файла lab5-1.asm	9
3.6	Использование подпрограмм из файла in_out.asm	10
3.7	Трансляция файла lab5-2.asm	10
4.1	Видоизменённая программа без использования внешнего файла in_out.asm	13
4.2	Трансляция полученного файла	13
4.3	Видоизменённая программа с использованием внешнего файла in_out.asm	14
4.4	Трансляция полученного файла	14

Список таблиц

1 Цель работы

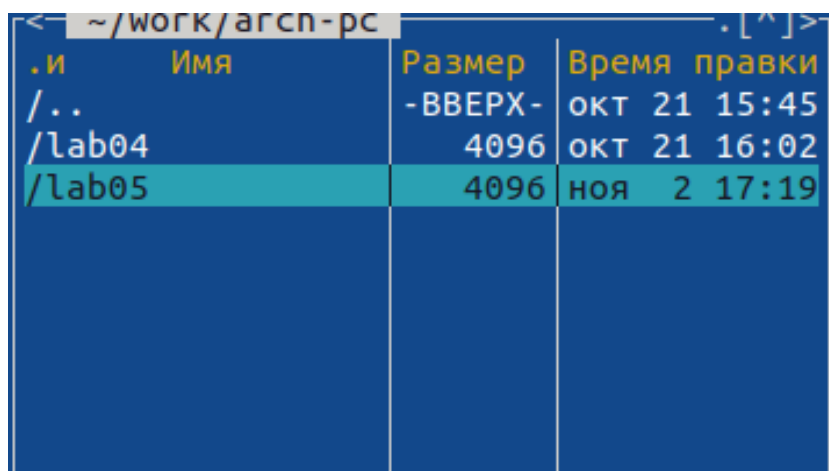
Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

1. Создайте копию файла lab5-1.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.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab5-2.asm. Исправьте текст программы с использование под-программ 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 Выполнение лабораторной работы

1. Откроем Midnight Commander с помощью команды `mc`. Перейдем в каталог `~/work/arch-pc` созданный при выполнении лабораторной работы №4 и создадим каталог `lab05`.(3.1)



<-- ~/work/arch-pc .[^]>			
.и	Имя	Размер	Время правки
/..		-ВВЕРХ-	окт 21 15:45
/lab04		4096	окт 21 16:02
/lab05		4096	ноя 2 17:19

Рис. 3.1: Создание каталога

2. Пользуясь строкой ввода и командой `touch` создадим файл `lab5-1.asm`.(3.2)

The screenshot shows a file manager window with the address bar set to `~/work/arch-pc/lab05`. The window displays a table of files:

.и	Имя	Размер	Время правки
/..		-ВВЕРХ-	ноя 2 17:19
	lab5-1.asm	0	ноя 2 17:21

Рис. 3.2: Создание файла

3. С помощью функциональной клавиши F4 откроем файл `lab5-1.asm` для редактирования во встроенном редакторе и напечатаем следующий текст программы (3.3) `SECTION .data msg: DB 'Введите строку:', 10 msgLen: EQU $-msg SECTION .bss buf1: RESB 80 SECTION .text GLOBAL _start _start: mov eax,4 mov ebx,1 mov ecx,msg mov edx,msgLen int 80h mov eax,3 mov ebx,0 mov ecx,buf1 mov edx,80 int 80h mov eax,1 mov ebx,0 int 80h`

The screenshot shows an assembly code editor with the following code:

```

/home/eg-b5-1.asm [-M--] 0 L:[ 18+21 39/ 39] *(2410/2410b) <EOF> [*][X]
;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 вывода
;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         ;Для 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. 3.3: Текст программы

- Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введем ФИО.(3.4).

```

egor@esputnik-dk3n56:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
egor@esputnik-dk3n56:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
egor@esputnik-dk3n56:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Попутников Егор Сергеевич
egor@esputnik-dk3n56:~/work/arch-pc/lab05$

```

Рис. 3.4: Трансляция текста программы

- Скачаем файл in_out.asm со страницы курса в ТУИС. Подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется. В одной из панелей mc откроем каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in_out.asm (для перемещения между панелями используем Tab). Скопируем файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5.С помощью функциональной клавиши F6 создадим копию файла lab5-1.asm с именем lab5-2.asm.(3.5)

<- ~/work/arch-pc/lab05 .[^]>				<- ~/Загрузки .[^]>			
.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
./..		-ВВЕРХ-	ноя 2 17:19	./..		-ВВЕРХ-	ноя 4 14:58
*lab5-1		8744	ноя 4 14:59	iloveimg-converted.zip		499016	окт 21 21:23
lab5-1.asm		250	ноя 4 14:56	in_out.asm		3942	ноя 3 13:49
lab5-1.o		736	ноя 4 14:59	install-tl.zip		25419K	окт 8 05:52
				lab02-1.pdf		706116	окт 9 00:42
				lab02.pdf		706116	сен 29 21:39

<- ~/work/arch-pc/lab05 .[^]>			
.и	Имя	Размер	Время правки
./..		-ВВЕРХ-	ноя 2 17:19
in_out.asm		3942	ноя 3 13:49
*lab5-1		8744	ноя 4 14:59
lab5-1.o		736	ноя 4 14:59
lab5-2.asm		250	ноя 4 14:56

Рис. 3.5: Создание копии файла lab5-1.asm

6. Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используем подпрограммы sprintLF, sread и quit)(3.6) %include 'in_out.asm' SECTION .data msg: DB 'Введите строку:',0h SECTION .bss buf1: RESB 80 SECTION .text GLOBAL _start _start: mov eax,msg call sprint mov ecx, buf1 mov edx,80 call sread call quit

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprintLF
mov ecx, buf1
mov edx,80
call sread
call quit
```

Рис. 3.6: Использование подпрограмм из файла in_out.asm

7. В файле lab5-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу.(Разница этих подпрограмм в том,что sprintLF добавляет к сообщению символ перевода строки) (3.7)

```
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:Попутников Егор Сергеевич
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$
```

Рис. 3.7: Трансляция файла lab5-2.asm

4 Выполнение самостоятельной работы

Создадим копию файла lab5-1.asm. Внесём 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.(4.1) “SECTION .data msg: DB ‘Введите строку:’,10 msgLen: EQU \$-msg inputPrompt: DB ‘Введенная строка:’,10 inputPromptLen: EQU \$-inputPrompt SECTION .bss buf1: RESB 80 SECTION .text GLOBAL _start

```
_start: ; Вывести приглашение “Введите строку:” mov eax, 4 mov ebx, 1 mov ecx, msg mov edx, msgLen int 80h
```

```
; Ввести строку с клавиатуры
```

```
mov eax, 3
```

```
mov ebx, 0
```

```
mov ecx, buf1
```

```
mov edx, 80
```

```
int 80h
```

```
; Вывести введенную строку на экран
```

```
mov eax, 4
```

```
mov ebx, 1
```

```
mov ecx, inputPrompt
```

```
mov edx, inputPromptLen
```

```
int 80h
```

```
mov eax, 4
mov ebx, 1
mov ecx, buf1
mov edx, 80
int 80h
```

```
; Завершить программу
```

```
mov eax, 1
xor ebx, ebx
int 80h````
```

```

SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
inputPrompt: DB 'Введенная строка:',10
inputPromptLen: EQU $-inputPrompt
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start

_start:
    ; Вывести приглашение "Введите строку:"
    mov eax, 4
    mov ebx, 1
    mov ecx, msg
    mov edx, msgLen
    int 80h

    ; Ввести строку с клавиатуры
    mov eax, 3
    mov ebx, 0
    mov ecx, buf1
    mov edx, 80
    int 80h

    ; Вывести введенную строку на экран
    mov eax, 4
    mov ebx, 1
    mov ecx, inputPrompt
    mov edx, inputPromptLen
    int 80h

    mov eax, 4
    mov ebx, 1
    mov ecx, buf1
    mov edx, 80

```

Рис. 4.1: Видоизменённая программа без использования внешнего файла in_out.asm

Получим исполняемый файл и проверим его работу. На приглашение ввести строку введем ФИО.(4.2)

```

egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ nasm -f elf lab5-1c.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1c lab5-1c.o
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ ./lab5-1c
Введите строку:
Попутников Егор Сергеевич
Введенная строка:
Попутников Егор Сергеевич
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ █

```

Рис. 4.2: Трансляция полученного файла

Создадим копию файла lab5-2.asm. Исправим текст программы с использованием под- программ из 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.

(4.3) ““%include ‘in_out.asm’ SECTION .data msg: DB ‘Введите строку:’,0h SECTION .bss buf1: RESB 80 SECTION .text GLOBAL _start _start: ; Вывести приглашение “Введите строку:”
 mov eax, msg call sprintfLF
 ; Ввести строку с клавиатуры mov ecx, buf1 mov edx, 80 call sread
 ; Вывести введенную строку на экран mov eax, buf1 call sprintfLF
 call quit”“

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
; Вывести приглашение "Введите строку:"
mov eax, msg
call sprintfLF

; Ввести строку с клавиатуры
mov ecx, buf1
mov edx, 80
call sread

; Вывести введенную строку на экран
mov eax, buf1
call sprintfLF

call quit
```

Рис. 4.3: Видоизменённая программа с использованием внешнего файла in_out.asm

Создадим исполняемый файл и проверим его работу.(4.4)

```
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ nasm -f elf lab5-2c.asm
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2c lab5-2c.o
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$ ./lab5-2c
Введите строку:
Попутников Егор Сергеевич
Попутников Егор Сергеевич
egor@espoputnikov-dk3n56:~/work/arch-pc/lab05$
```

Рис. 4.4: Трансляция полученного файла

5 Выводы

Я приобрёл практические навыки работы в Midnight Commander. Освоил инструкции языка ассемблера `mov` и `int`.