

Отчет по лабораторной работе №5

Архитектура вычислительных систем

Дмитрий Владимирович Орлюк

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	17
	Список литературы	18

Список иллюстраций

3.1	Открытие Midnight Commander	8
3.2	Создание папки для лабораторной работы	9
3.3	Проверка создания папки	9
3.4	Создания файла .asm	9
3.5	Редактирование файла в формате .asm	10
3.6	Проверка наличия текста	11
3.7	Транслирование команды, проверка работоспособности	11
3.8	Перенос файла in_out.asm в нужную папку	12
3.9	Создание копии файла lab6-1	13
3.10	Внесение изменений в файл	14
3.11	Проверка наличия текста	15
3.12	Проверка работоспособности файла	15
3.13	Добавление нужных команд из с.р	16
3.14	Транслирование текста в файл	16
3.15	Проверка работоспособности файла	16

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.
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 Выполнение лабораторной работы

1. С помощью команды `mc` откроем Mignight Commander, перейдем в нужный каталог

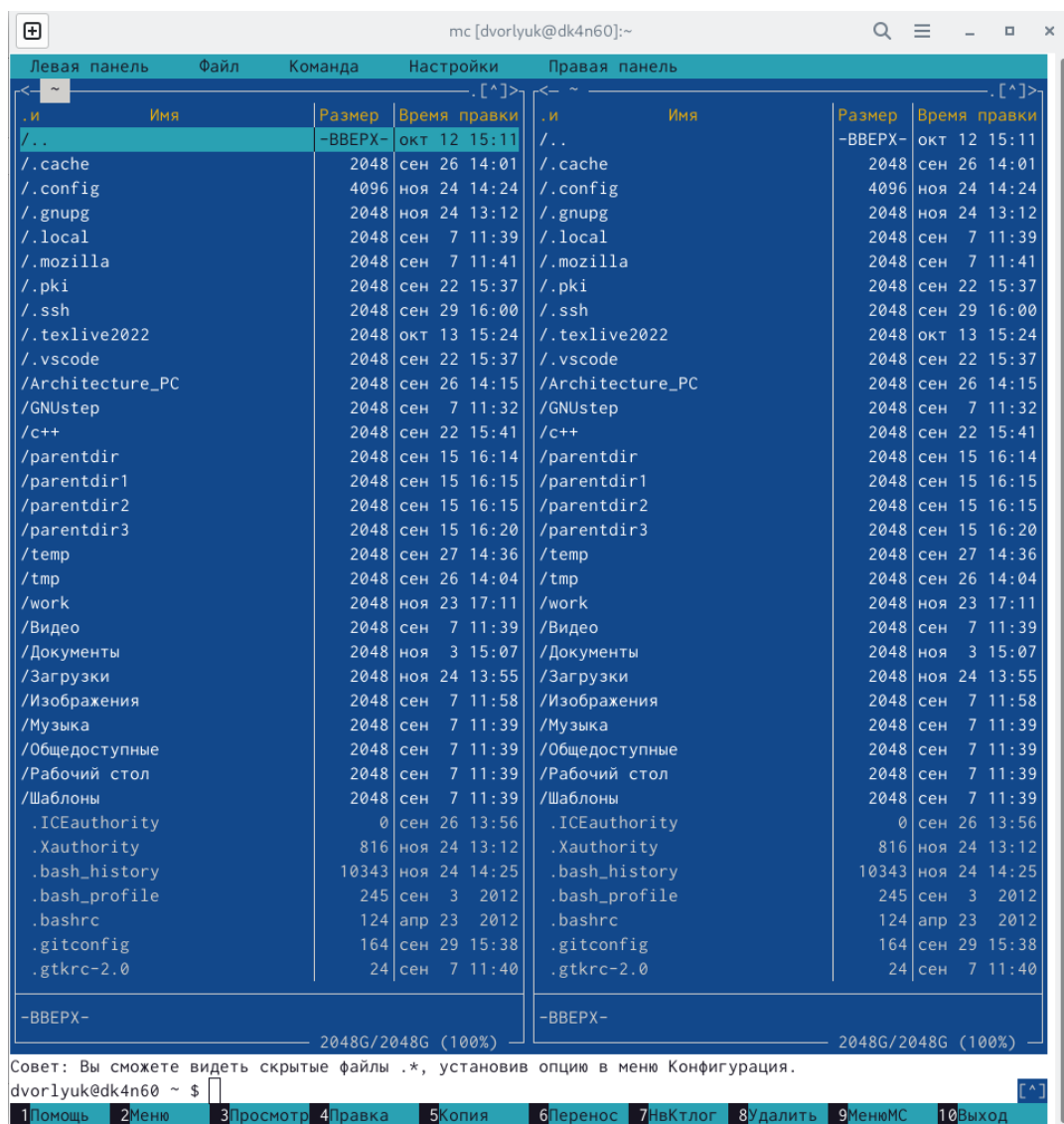


Рис. 3.1: Открытие Midnight Commander

2. С помощью F7 создаем папку lab05

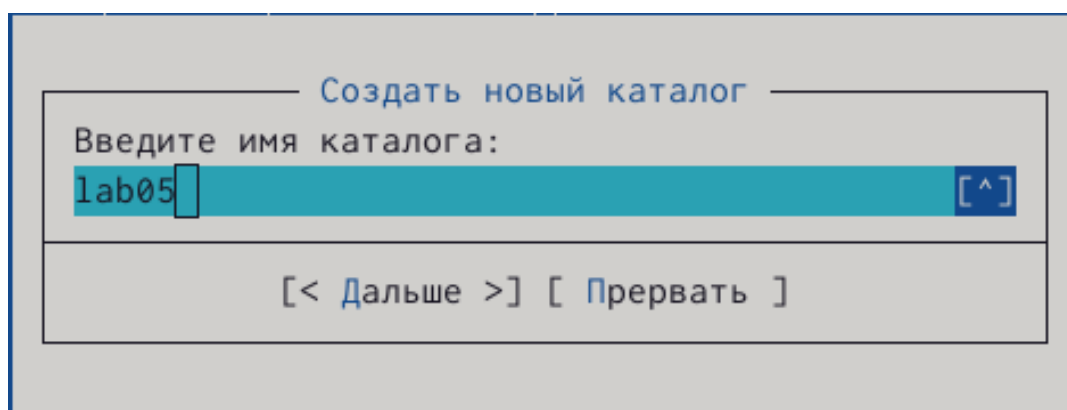


Рис. 3.2: Создание папки для лабораторной работы

3. Убедимся, что папка создана!

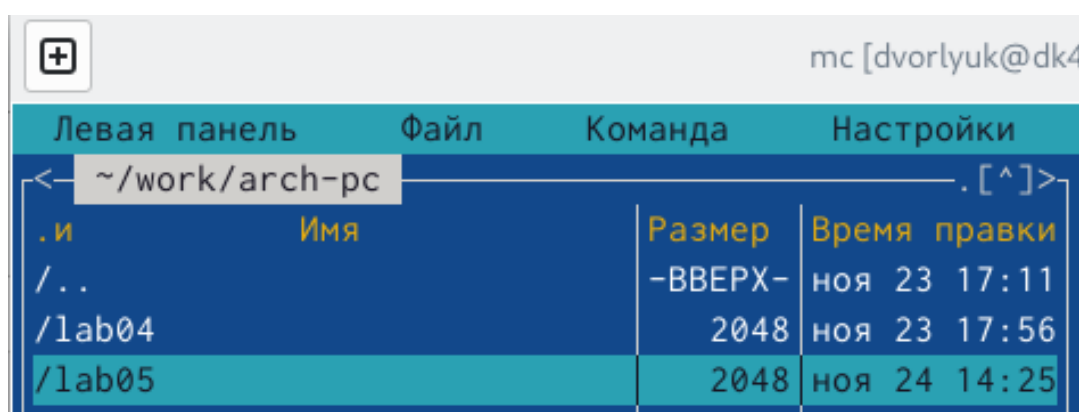


Рис. 3.3: Проверка создания папки

4. С помощью строки ввода создадим файл lab5-1 в формате .asm

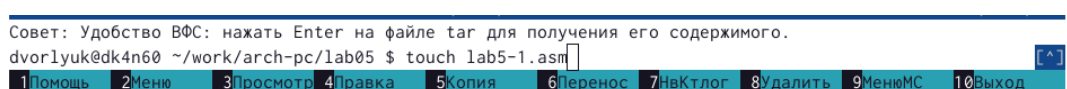


Рис. 3.4: Создания файла .asm

5. С помощью клавиши F4 откроем файл и внесем нужный текст из листинга

6.1

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/v/dvorlyuk/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

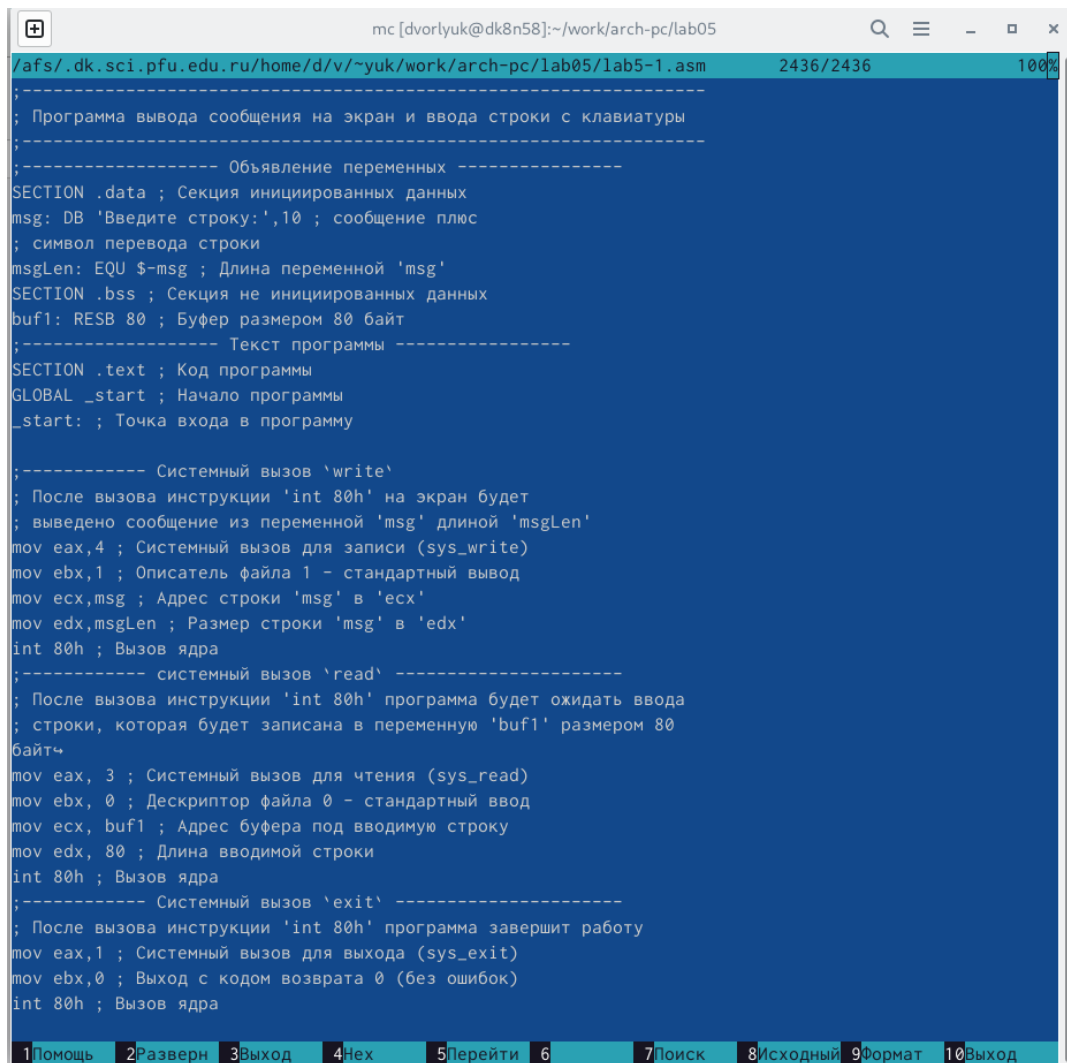
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер 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)

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    M-U Отмена      M-A Установить ме
^X Выход        ^R ЧитФайл     ^\ Замена       ^U Вставить     ^C Позиция      M-E Повтор      M-6 Копировать

```

Рис. 3.5: Редактирование файла в формате .asm

6. Убедимся, что текст добавлен простым путем открытия файла



```
mc [dvorlyuk@dk8n58]:~/work/arch-pc/lab05
/afs/.dk.sci.pfu.edu.ru/home/d/v/~yuk/work/arch-pc/lab05/lab5-1.asm 2436/2436 100%
; 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
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 ; 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

1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

Рис. 3.6: Проверка наличия текста

7. Оттранслируем текст программы в объектный файл, путем нескольких команд, затем проверим работоспособность.

```
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
lab5-1.asm:27: warning: label alone on a line without a colon might be in error [-w+label-orphan]
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Орлюк Дмитрий Владимирович
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $
```

Рис. 3.7: Транслирование команды, проверка работоспособности

8. Файл из ТУИС скачаем, с помощью F5 перенесем в тот каталог, где лежит

файл с программой.

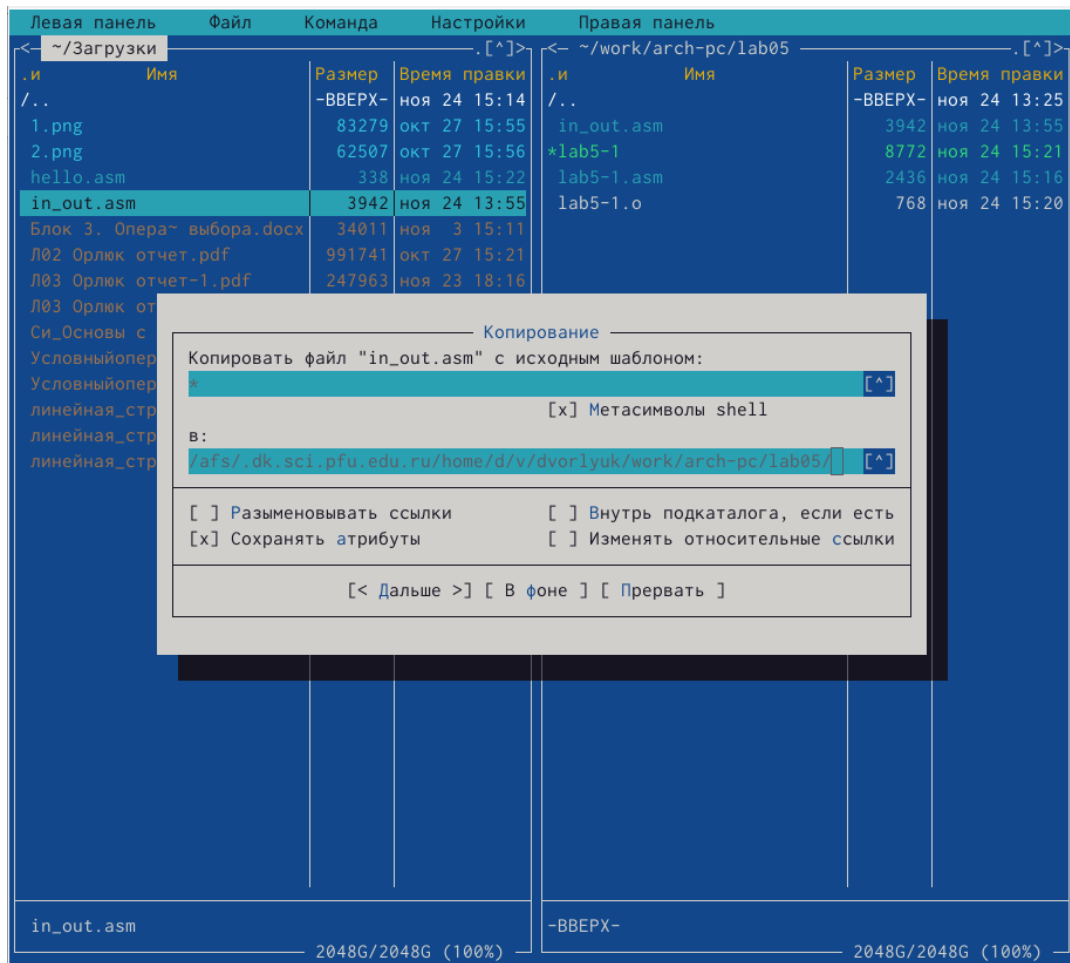


Рис. 3.8: Перенос файла in_out.asm в нужную папку

9. С помощью клавиши F6 создадим копию файла lab5-1 с именем lab6-2

Левая панель		Команда	Настройки	Правая панель			
Файл				Файл			
~/work/arch-pc/lab05				~/work/arch-pc/lab05			
Имя	Размер		Время правки	Имя	Размер		Время правки
./	-ВВЕРХ-		ноя 24 13:25	./	-ВВЕРХ-		ноя 24 13:25
in_out.asm	3942		ноя 24 13:55	in_out.asm	3942		ноя 24 13:55
*lab5-1	8772		ноя 24 15:21	*lab5-1	8772		ноя 24 15:21
lab5-1.o	768		ноя 24 15:20	lab5-1.o	768		ноя 24 15:20
lab5-2.asm	1224		ноя 24 15:27	lab5-2.asm	1224		ноя 24 15:27
</							

Рис. 3.9: Создание копии файла lab6-1

10. Внесем нужные изменения в файл с помощью команды `msedit`

```
lab5-2.asm [B---] 41 L: [ 1+16 17/ 17] *(1224/1224b) <EOF> [*][X]
;-----
; 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 sprintf ; 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. 3.10: Внесение изменений в файл

11. Проверим, что текст изменен, простым путем открытия файла

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/v/dvorlyuk/work/arch-pc/lab05/lab5-2.asm Изменён
;-----
; 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. 3.11: Проверка наличия текста

12. Проверим работу файла

```
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Орлюк Дмитрий Владимирович
```

Рис. 3.12: Проверка работоспособности файла

13. (Задание из ср) Редактируем файл asm так, чтоб при вводе нужного текста, он отображался на новой. Для этого, допишем нужные команды (В КОНЕЦ НАПИШЕМ ВЫЗОВ ПРОГРАММЫ ЗАВЕРШЕНИЯ)

```
mov eax, buf1 ; запись адреса выводимого сообщения в 'EAX'  
call sprintf ; вызов подпрограммы печати сообщения  
call quit ; вызов подпрограммы завершения
```

Рис. 3.13: Добавление нужных команд из с.р

14. Оттранслируем текст программы в объектный файл.

```
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm  
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
```

Рис. 3.14: Транслирование текста в файл

15. Проверим работоспособность

```
dvorlyuk@dk8n58 ~/work/arch-pc/lab05 $ ./lab5-2  
Введите строку: Орлюк Дмитрий Владимирович  
Орлюк Дмитрий Владимирович
```

Рис. 3.15: Проверка работоспособности файла

4 Выводы

В ходе лабораторной работы мною были приобретены практические навыки работы в Midnight Commander, а также освоены инструкции языка ассемблера `mov` и `int`. Я научился работать с МС, и с его помощью работать с файлами (Создание, переименовывание, копирование, перемещение, удаление, и тд.)

Список литературы