

Отчёт по лабораторной работе №5

дисциплина: Архитектура компьютера

Баранова Анна Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Подключение внешнего файла in_out.asm	13
4.2	Задание для самостоятельной работы	18
5	Выводы	21

List of Figures

4.1	Открытие Midnight Commander	8
4.2	Открытие Midnight Commander	8
4.3	Создание папки lab05	9
4.4	Создание папки lab05	9
4.5	Создание файла lab5-1.asm	10
4.6	Создание файла lab5-1.asm	10
4.7	Открытие файла lab5-1.asm с помощью F4	10
4.8	Изменения в файле lab5-1.asm	11
4.9	Проверка текста файла	11
4.10	Компиляция текста	12
4.11	Компиляция текста	12
4.12	Передача объектного файла на обработку компоновщику	12
4.13	Передача объектного файла на обработку компоновщику	12
4.14	Запуск исполняемого файла	12
4.15	Запуск исполняемого файла	12
4.16	Открытие Midnight Commander	13
4.17	Копирование файла in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5	13
4.18	Копирование файла in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5	14
4.19	Создание копии файла lab5-1.asm с именем lab5-2.asm	14
4.20	Создание копии файла lab5-1.asm с именем lab5-2.asm	15
4.21	Открытие файла lab5-2.asm с помощью F4	15
4.22	Изменения в файле lab5-2.asm	16
4.23	Компиляция текста	16
4.24	Передача объектного файла на обработку компоновщику	16
4.25	Компиляция текста и передача объектного файла на обработку компоновщику	16
4.26	Запуск исполняемого файла	17
4.27	Запуск исполняемого файла	17
4.28	Изменения в файле lab5-2.asm	17
4.29	Создание исполняемого файла и проверка его работы	17
4.30	Создание копии файла lab5-1.asm с именем lab5-12.asm	18
4.31	Изменения в файле lab5-12.asm	18
4.32	Создание исполняемого файла и проверка его работы	19
4.33	Создание исполняемого файла и проверка его работы	19

4.34 Создание копии файла lab5-2.asm с именем lab5-22.asm, и изменение текста промграммы	19
4.35 Создание исполняемого файла и проверка его работы	20
4.36 Создание исполняемого файла и проверка его работы	20

1 Цель работы

Цель работы - приобрести практические навыки работы в Midnight Commander и освоить инструкции языка ассемблера mov и int.

2 Задание

В ходе выполнения данной лабораторной работы необходимо изучить:

- Основы работы с Midnight Commander;
- структуру программы на языке ассемблера NASM;
- инструкции `mov` и `int`;
- системные вызовы для обеспечения диалога с пользователем.

Выполнив эту работу, мы приобретём практические навыки работы в Midnight Commander и освоим инструкции языка ассемблера `mov` и `int`.

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером.

4 Выполнение лабораторной работы

Откроем Midnight Commander (рис. 4.1), (рис. 4.2).

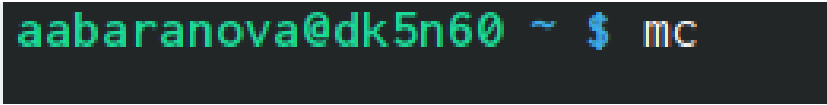


Рис. 4.1: Открытие Midnight Commander

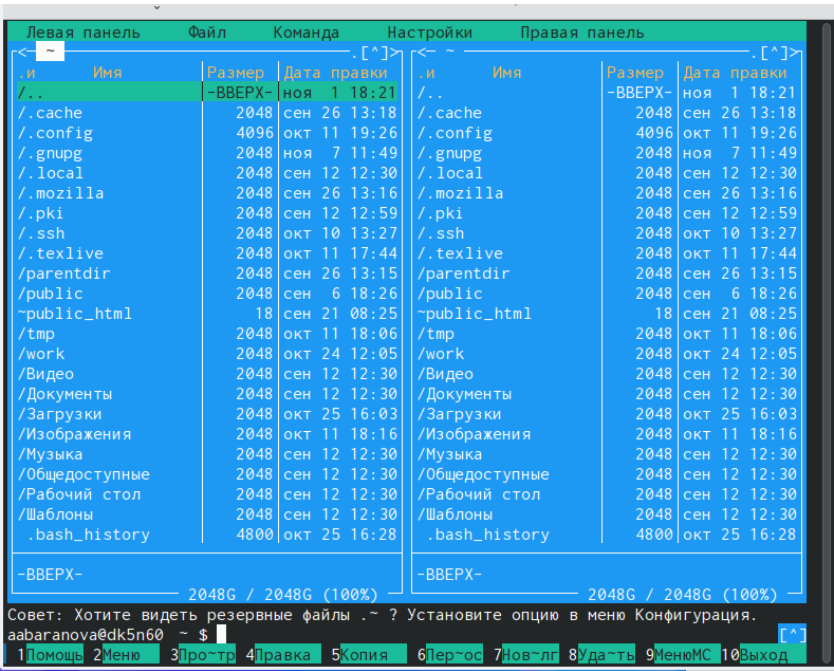


Рис. 4.2: Открытие Midnight Commander

Перейдём в каталог ~/work/arch-рс созданный при выполнении лабораторной работы №4, с помощью функциональной клавиши F7 создадим папку lab05 и перейдём в созданный каталог (рис. 4.3), (рис. 4.4).

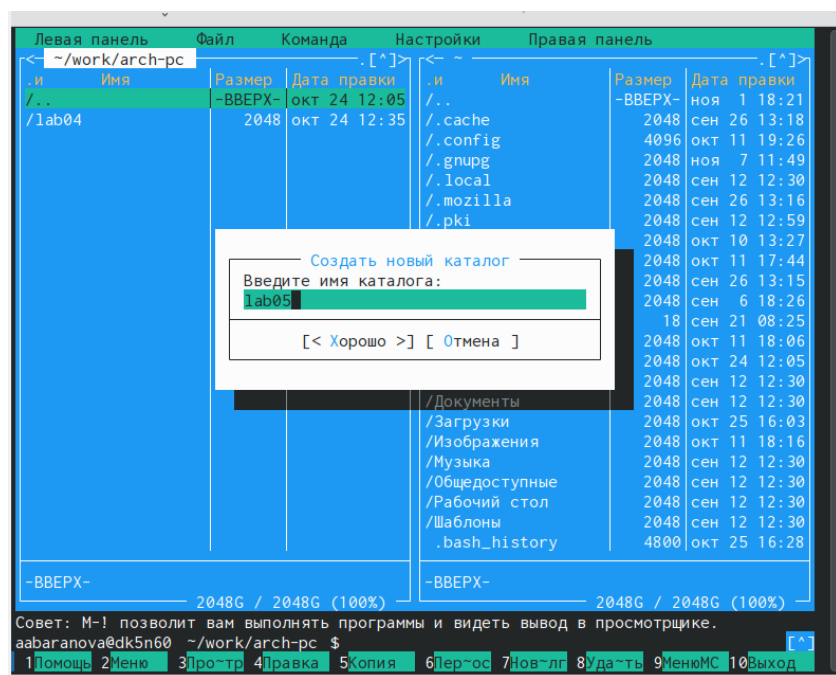


Рис. 4.3: Создание папки lab05

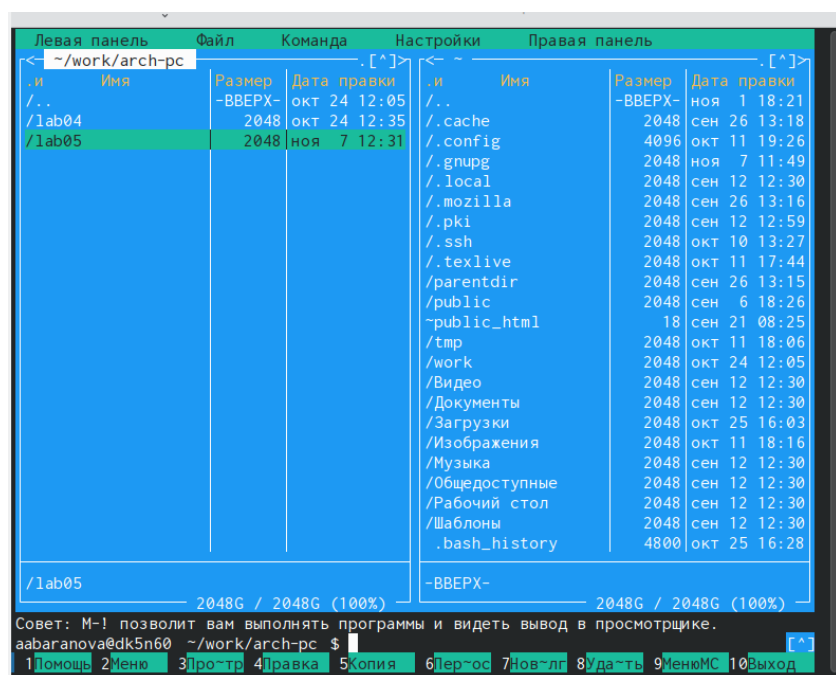


Рис. 4.4: Создание папки lab05

Пользуясь строкой ввода и командой `touch` создадим файл `lab5-1.asm` (рис. 4.5), (рис. 4.6).

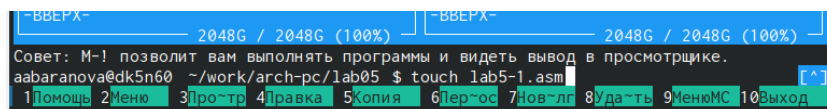


Рис. 4.5: Создание файла lab5-1.asm

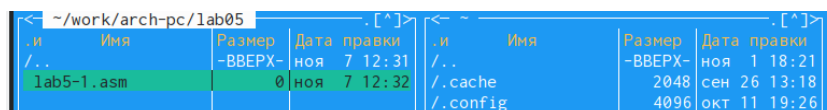


Рис. 4.6: Создание файла lab5-1.asm

С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенном редакторе (рис. 4.7).

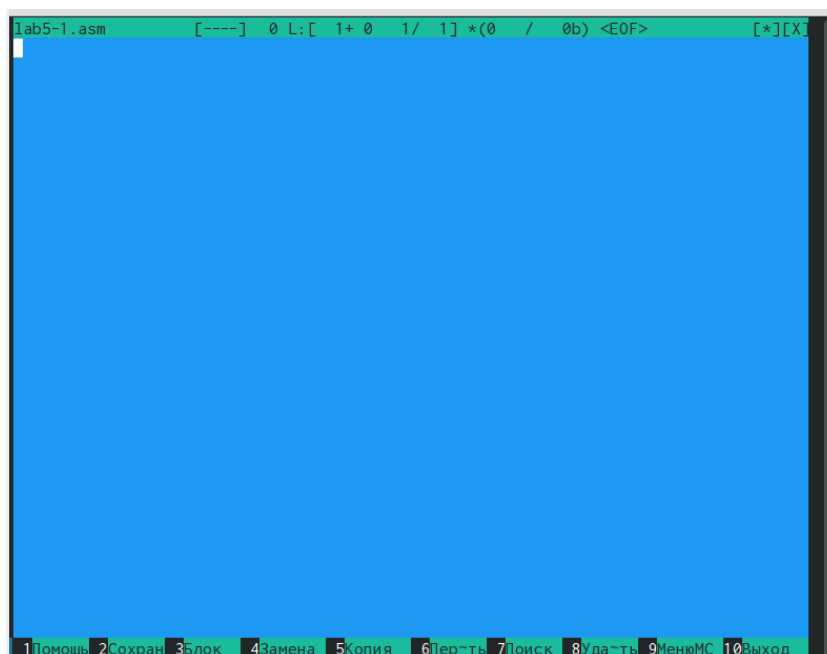


Рис. 4.7: Открытие файла lab5-1.asm с помощью F4

Введём текст программы в файл, сохраним изменения и закроем файл. (рис. 4.8).

```

lab5-1.asm [-M--] 19 L: [ 1+ 0 1/ 35] *(19 /2626b) 0045 0x02D [*][X]
; 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
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС 10Выход

```

Рис. 4.8: Изменения в файле lab5-1.asm

С помощью функциональной клавиши F3 откроем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы (рис. 4.9).

```

/afs/.dk.sci.pfu.edu.ru/h/arch-pc/lab05/lab5-1.asm 2237/2626 85%
;----- 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
1Помощь 2Раз-рн 3Выход 4Hex 5Перейти 6 7Поиск 8Исх-ый 9Формат 10Выход

```

Рис. 4.9: Проверка текста файла

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. (рис. 4.10), (рис. 4.11), (рис. 4.12), (рис. 4.13), (рис. 4.14), (рис. 4.15).

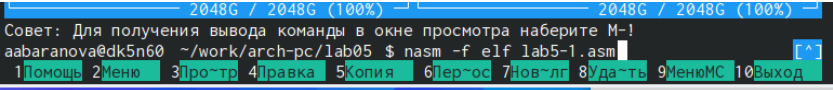


Рис. 4.10: Компиляция текста

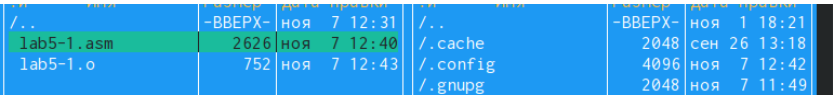


Рис. 4.11: Компиляция текста

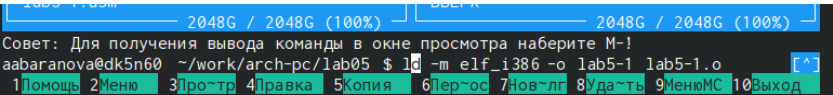


Рис. 4.12: Передача объектного файла на обработку компоновщику

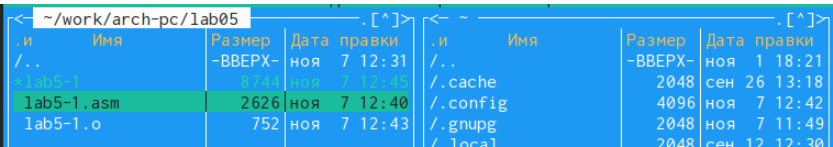


Рис. 4.13: Передача объектного файла на обработку компоновщику

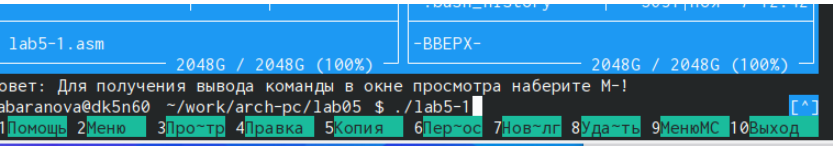


Рис. 4.14: Запуск исполняемого файла

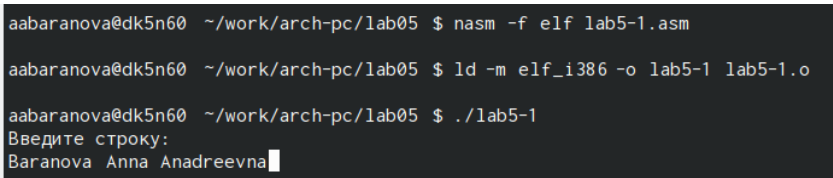


Рис. 4.15: Запуск исполняемого файла

4.1 Подключение внешнего файла in_out.asm

В одной из панелей mc откроем каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in_out.asm. Скопируйте файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5 (рис. 4.16), (рис. 4.17), (рис. 4.18).

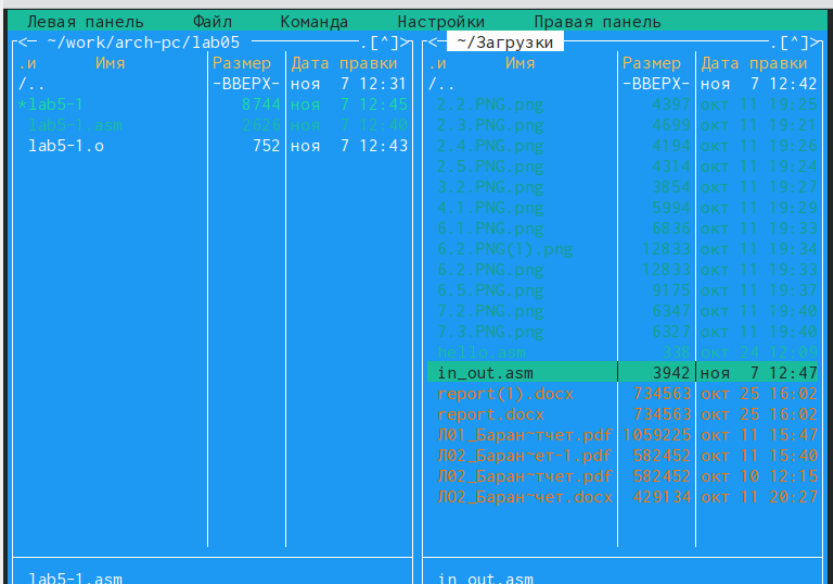


Рис. 4.16: Открытие Midnight Commander

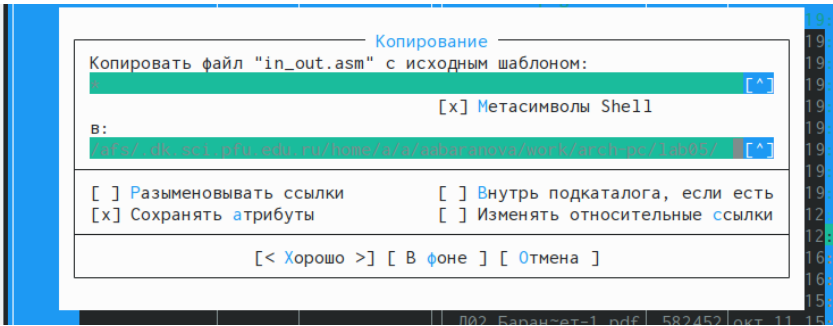


Рис. 4.17: Копирование файла in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5

< ~/work/arch-pc/lab05 .[*]>				< ~/Загрузки .[*]>			
Имя	Размер	Дата правки		Имя	Размер	Дата правки	
..	-ВВЕРХ-	ноя 7 12:31		..	-ВВЕРХ-	ноя 7 12:42	
in_out.asm	3942	ноя 7 12:47		2.2.PNG.png	4397	окт 11 19:25	
lab05-1	8148	ноя 7 12:40		2.3.PNG.png	4698	окт 11 19:21	
lab5-1.asm	2626	ноя 7 12:40		2.4.PNG.png	4194	окт 11 19:26	
lab5-1.o	752	ноя 7 12:43		2.5.PNG.png	4314	окт 11 19:24	
				3.2.PNG.png	3854	окт 11 19:27	
				4.1.PNG.png	5994	окт 11 19:29	
				6.1.PNG.png	6836	окт 11 19:33	
				6.2.PNG(1).png	12833	окт 11 19:34	
				6.2.PNG.png	12833	окт 11 19:33	
				6.5.PNG.png	9175	окт 11 19:37	
				7.2.PNG.png	6347	окт 11 19:40	
				7.3.PNG.png	6327	окт 11 19:40	
				hello.asm	138	окт 24 12:00	
				in_out.asm	3942	ноя 7 12:47	

Рис. 4.18: Копирование файла 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 (рис. 4.19), (рис. 4.20).

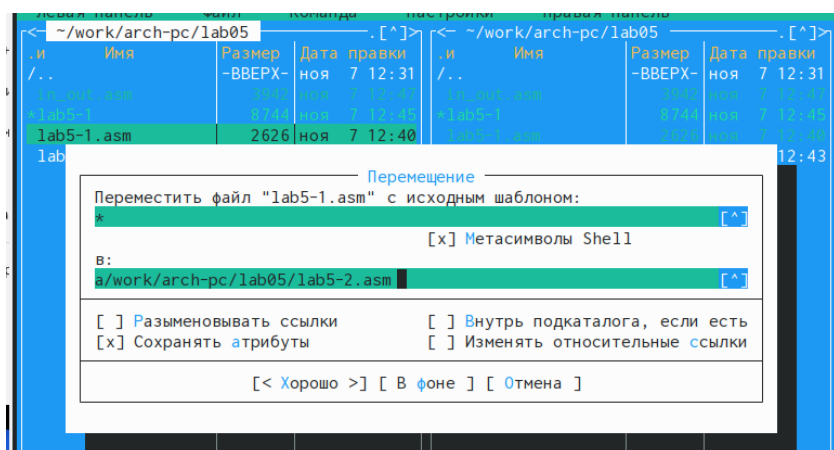


Рис. 4.19: Создание копии файла lab5-1.asm с именем lab5-2.asm

левая панель				правая панель			
файлы				настройка			
[< -/work/arch-pc/lab05 .[*]>				[< -/work/arch-pc/lab05 .[*]>			
Имя	Размер	Дата	правки	Имя	Размер	Дата	правки
./..	-ВВЕРХ-	ноя 7 12:31		./..	-ВВЕРХ-	ноя 7 12:31	
in_out.asm	3940	ноя 7 12:47		in_out.asm	3940	ноя 7 12:47	
lab5-1	752	ноя 7 12:43		lab5-1	752	ноя 7 12:43	
lab5-1.o	752	ноя 7 12:43		lab5-1.o	752	ноя 7 12:43	
lab5-2.asm	2626	ноя 7 12:48		lab5-2.asm	2626	ноя 7 12:48	

Рис. 4.20: Создание копии файла lab5-1.asm с именем lab5-2.asm

Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm (рис. 4.21), (рис. 4.22).

```

lab5-2.asm [-M--] 63 L: [ 1+ 0 1/ 25] *(63 /1350b) 0045 0x02D [*][X]
; 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 sprintf ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'

call sread ; вызов подпрограммы ввода сообщения

call quit ; вызов подпрограммы завершения

```

Рис. 4.21: Открытие файла lab5-2.asm с помощью F4

```

/afs/.dk.sci.pfu.edu.ru/home-k/arch-pc/lab05/lab5-2.asm 1350/1350 100%
;-----
; 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 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. 4.22: Изменения в файле lab5-2.asm

Создадим исполняемый файл и проверим его работу (рис. 4.23), (рис. 4.24), (рис. 4.25), (рис. 4.26), (рис. 4.27).

```

lab5-2.asm 2048G / 2048G (100%) ВВЕРХ 2048G / 2048G (100%)
Совет: На медленных терминалах может помочь флаг -s.
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
1Помощь 2Меню 3Протогр 4Правка 5Копия 6Пер-ос 7НовКтлг 8Удалить 9МенюМС 10Выход

```

Рис. 4.23: Компиляция текста

```

lab5-2.asm 2048G / 2048G (100%) ВВЕРХ 2048G / 2048G (100%)
Совет: На медленных терминалах может помочь флаг -s.
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
1Помощь 2Меню 3Протогр 4Правка 5Копия 6Пер-ос 7НовКтлг 8Удалить 9МенюМС 10Выход

```

Рис. 4.24: Передача объектного файла на обработку компоновщику

Левая панель				Правая панель			
Файл		Команда		Файл		Команда	
~/work/arch-pc/lab05		. [^]>		~/work/arch-pc/lab05		. [^]>	
.	Имя	Размер	Дата правки	.	Имя	Размер	Дата правки
./..	-ВВЕРХ-		ноя 7 12:31	./..	-ВВЕРХ-		ноя 7 12:31
in_out.asm		3942	ноя 7 12:47	in_out.asm		3942	ноя 7 12:47
*lab5-1		8734	ноя 7 13:05	*lab5-1		8734	ноя 7 13:05
lab5-1.o		752	ноя 7 12:43	lab5-1.o		752	ноя 7 12:43
*lab5-2		9802	ноя 7 13:08	*lab5-2		9802	ноя 7 13:08
lab5-2.asm		1350	ноя 7 13:07	lab5-2.asm		1350	ноя 7 13:07
lab5-2.o		1312	ноя 7 13:08	lab5-2.o		1312	ноя 7 13:08

Рис. 4.25: Компиляция текста и передача объектного файла на обработку компоновщику


```
2048G / 2048G (100%) 2048G / 2048G (100%)
Совет: На медленных терминалах может помочь флаг -s.
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ./lab5-2
1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перейти 7Добавить 8Удалить 9МенюМС 10Выход
```

Рис. 4.26: Запуск исполняемого файла

```
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Baranova Anna Andreevna
```

Рис. 4.27: Запуск исполняемого файла

В файле lab5-2.asm заменим подпрограмму sprintLF на sprint (рис. 4.28).

```
lab5-2.asm [-M--] 12 L: [ 1+17 18/ 25] *(936 /1348b) 0032 0x020 [*][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 sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины вводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 4.28: Изменения в файле lab5-2.asm

Создадим исполняемый файл и проверим его работу (рис. 4.29).

```
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
bash: d: команда не найдена
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Baranova Anna Andreevna
```

Рис. 4.29: Создание исполняемого файла и проверка его работы

4.2 Задание для 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. (рис. 4.30), (рис. 4.31).



Рис. 4.30: Создание копии файла lab5-1.asm с именем lab5-12.asm

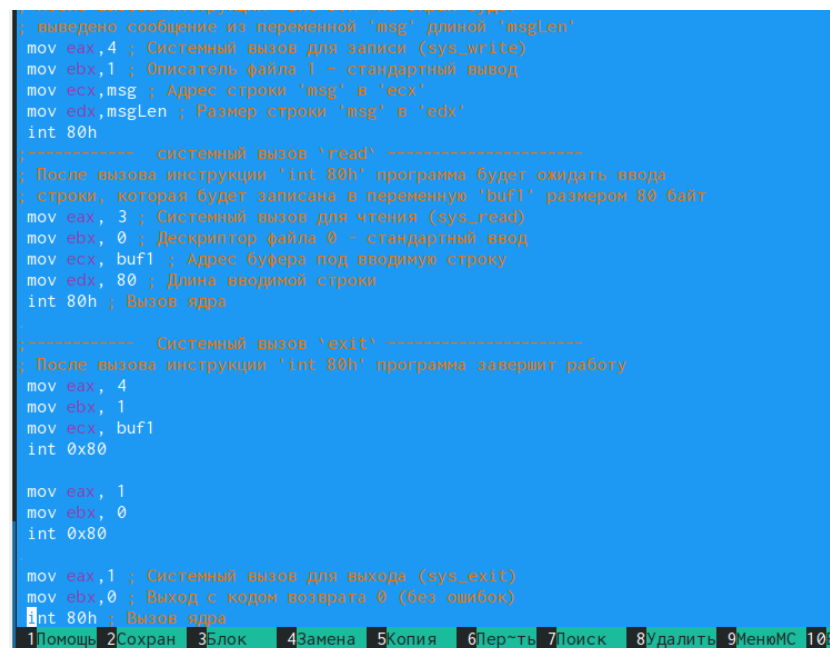


Рис. 4.31: Изменения в файле lab5-12.asm

Получим исполняемый файл и проверим его работу (рис. 4.32), (рис. 4.33).

Левая панель				Правая панель			
Файл		Команда		Настройки		Правая панель	
<- ~/work/arch-pc/lab05		. [*]>		<- ~/work/arch-pc/lab05			
Имя	Размер	Дата	правки	Имя	Размер	Дата	правки
./..	-ВВЕРХ-	ноя	7 12:31	./..	-ВВЕРХ-	ноя	7 12:31
in_out.asm	3942	ноя	7 12:47	in_out.asm	3942	ноя	7 12:47
lab5-1	8744	ноя	7 13:40	lab5-1	8744	ноя	7 13:40
lab5-1.asm	1348	ноя	7 13:12	lab5-1.asm	1348	ноя	7 13:12
lab5-1.o	752	ноя	7 12:43	lab5-1.o	752	ноя	7 12:43
lab5-12	8748	ноя	7 14:03	lab5-12	8748	ноя	7 14:03
lab5-12.asm	2514	ноя	7 14:03	lab5-12.asm	2514	ноя	7 14:03
lab5-12.o	800	ноя	7 14:03	lab5-12.o	800	ноя	7 14:03
lab5-2	8002	ноя	7 13:08	lab5-2	8002	ноя	7 13:08
lab5-2.asm	1348	ноя	7 13:12	lab5-2.asm	1348	ноя	7 13:12
lab5-2.o	1312	ноя	7 13:13	lab5-2.o	1312	ноя	7 13:13

Рис. 4.32: Создание исполняемого файла и проверка его работы

```
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-12.asm
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-12 lab5-12.o
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ./lab5-12
Введите строку:
Baranova
Baranova
aabaranova@dk5n60 ~/work/arch-pc/lab05 $
```

Рис. 4.33: Создание исполняемого файла и проверка его работы

Создадим копию файла lab5-2.asm. Исправим 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 введенную строку на экран (рис. 4.34).

```
aabaranova@dk5n60 ~ $ mc
```

Рис. 4.34: Создание копии файла lab5-2.asm с именем lab5-22.asm, и изменение текста программы

Создадим исполняемый файл и проверим его работу (рис. 4.35), (рис. 4.36).

< ~/work/arch-pc/lab05 .[^>				< ~/work/arch-pc/lab05 .[^>			
.и	Имя	Размер	Дата правки	.и	Имя	Размер	Дата правки
./..	-ВВЕРХ-		ноя 7 12:31	./..	-ВВЕРХ-		ноя 7 12:31
in_out.asm		2902	ноя 7 12:47	in_out.asm		2902	ноя 7 12:47
*lab5-1		8744	ноя 7 13:48	*lab5-1		8744	ноя 7 13:48
lab5-1.asm		1348	ноя 7 13:12	lab5-1.asm		1348	ноя 7 13:12
lab5-1.o		752	ноя 7 12:43	lab5-1.o		752	ноя 7 12:43
*lab5-12		8744	ноя 7 14:04	*lab5-12		8744	ноя 7 14:04
lab5-12.asm		2514	ноя 7 14:03	lab5-12.asm		2514	ноя 7 14:03
lab5-12.o		800	ноя 7 14:03	lab5-12.o		800	ноя 7 14:03
*lab5-2		8744	ноя 7 13:08	*lab5-2		8744	ноя 7 13:08
lab5-2.asm		1348	ноя 7 13:12	lab5-2.asm		1348	ноя 7 13:12
lab5-2.o		1312	ноя 7 13:13	lab5-2.o		1312	ноя 7 13:13
*lab5-22		8744	ноя 7 14:18	*lab5-22		8744	ноя 7 14:18
lab5-22.asm		1379	ноя 7 14:18	lab5-22.asm		1379	ноя 7 14:18
lab5-22.o		1328	ноя 7 14:18	lab5-22.o		1328	ноя 7 14:18

Рис. 4.35: Создание исполняемого файла и проверка его работы

```

aabaranova@dk5n60 ~/work/arch-pc/lab05 $ nasm -f elf lab5-22.asm
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-22 lab5-22.o
aabaranova@dk5n60 ~/work/arch-pc/lab05 $ ./lab5-22
Введите строку: Baranova
Baranova
aabaranova@dk5n60 ~/work/arch-pc/lab05 $

```

Рис. 4.36: Создание исполняемого файла и проверка его работы

5 Выводы

В ходе выполнения данной лабораторной работы были приобретены практические навыки работы в Midnight Commander и освоены инструкции языка ассемблера `mov` и `int`.