

Лабораторная работа №6

Дисциплина: Архитектура компьютера

Алиева Милена Арифовна

Содержание

1 Цель работы

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

2 Задание

1. Основы работы с `mc`
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

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`). Для объявления инициированных данных в секции `.data` используются директивы `DB`, `DW`, `DD`, `DQ` и `DT`, которые резервируют память и указывают, какие значения должны храниться в этой памяти: - `DB` (define byte) — определяет переменную размером в 1 байт; - `DW` (define word) — определяет переменную размером в 2 байта (слово); - `DD` (define double word) — определяет переменную размером в 4 байта (двойное слово); - `DQ` (define quad word) — определяет переменную размером в 8 байт (учетверенное слово); - `DT` (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых

переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

4 Выполнение лабораторной работы

1. Открываю Midnight Commander, введя в терминал mc (рис. 1)

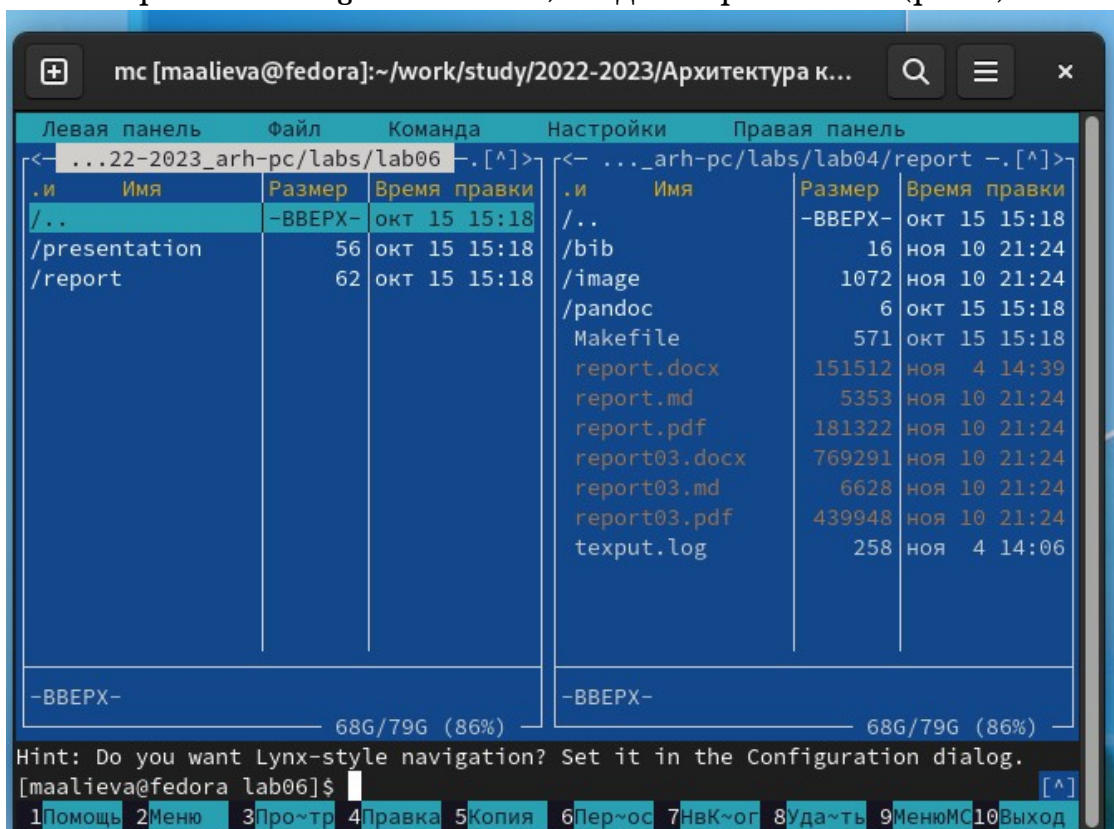


Рис. 1: Открытие Midnight Commander

2. Создаю каталог lab06_nasm (рис. 2)

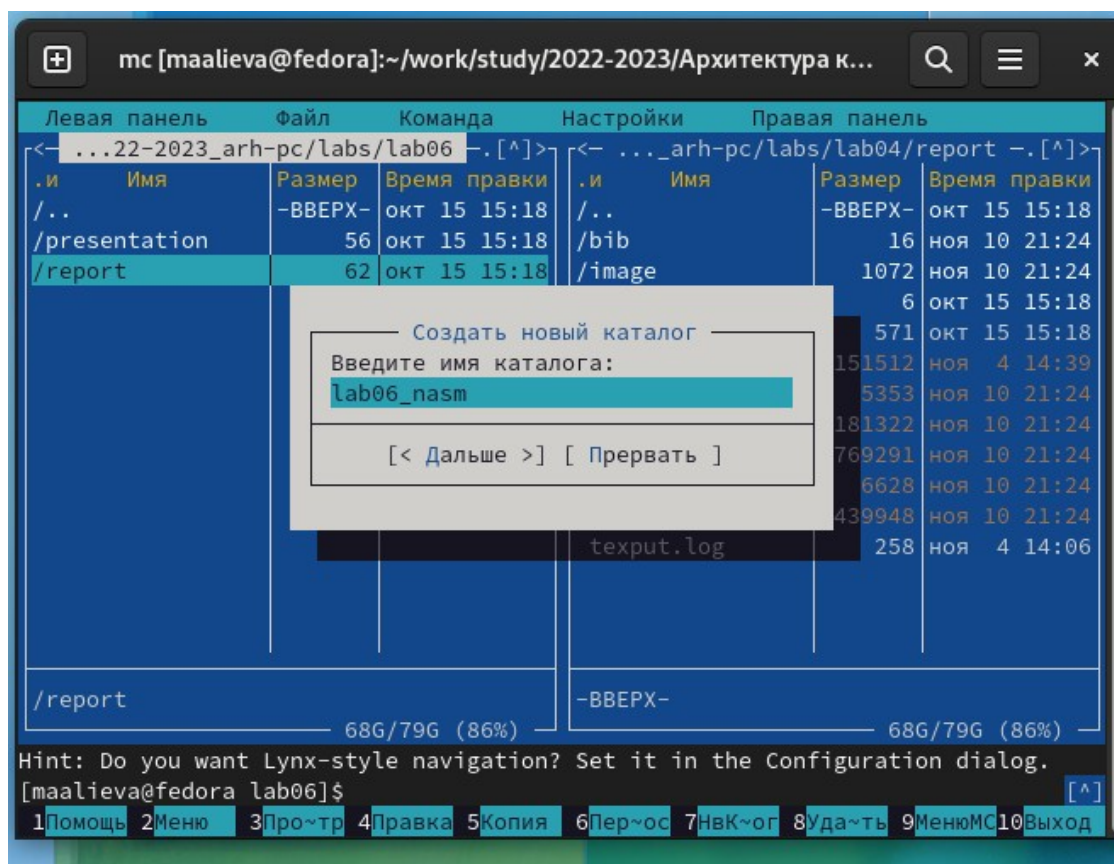


Рис. 2: Создание каталога lab06_nasm

3. С помощью команды touch lab6-1.asm создаю файл, в котором буду работать. С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Ввожу в файл код программы для запроса строки у пользователя (рис. 3)

```

mc [maalieva@fedora]:~/work/study/2022-2023/Архитектура к...
lab6-1.asm [-M--] 29 L: [ 1+ 0 1/ 37] *(29 /2812b) 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
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход

```

Рис. 3: Код программы для запроса строки у пользователя

4. Транслирую текст программы файла в объектный файл командой `nasm -f elf lab6-1.asm`, получаю объектный файл `lab6-1.o`, далее выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab6-1 lab6-1.o`, получаю исполняемый файл `lab6-1` (рис. 4)

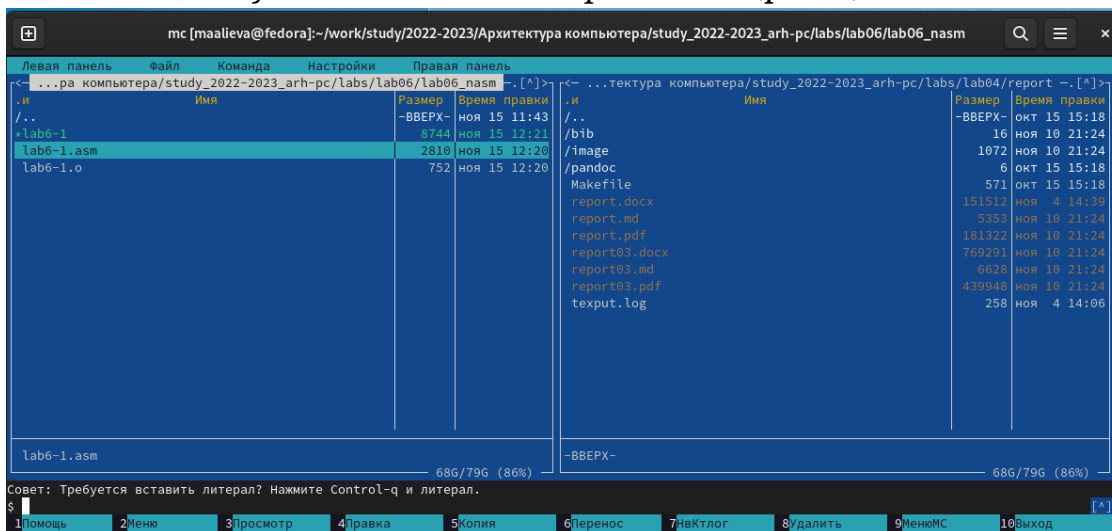


Рис. 4: Созданные файлы `lab6-1.o` и `lab6-1`

5. Запускаю исполняемый файл, программа просит ввести строку, ввожу имя и фамилию (рис. 5)

```
$ ./lab6-1
Введите строку:
Алиева Милена
```

Рис. 5: Исполнение файла, ввод имени и фамилии

6. Скачиваю файл in_out.asm со страницы курса в ТУИС. С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога загрузки в созданный каталог lab06_nasm (рис. 6)

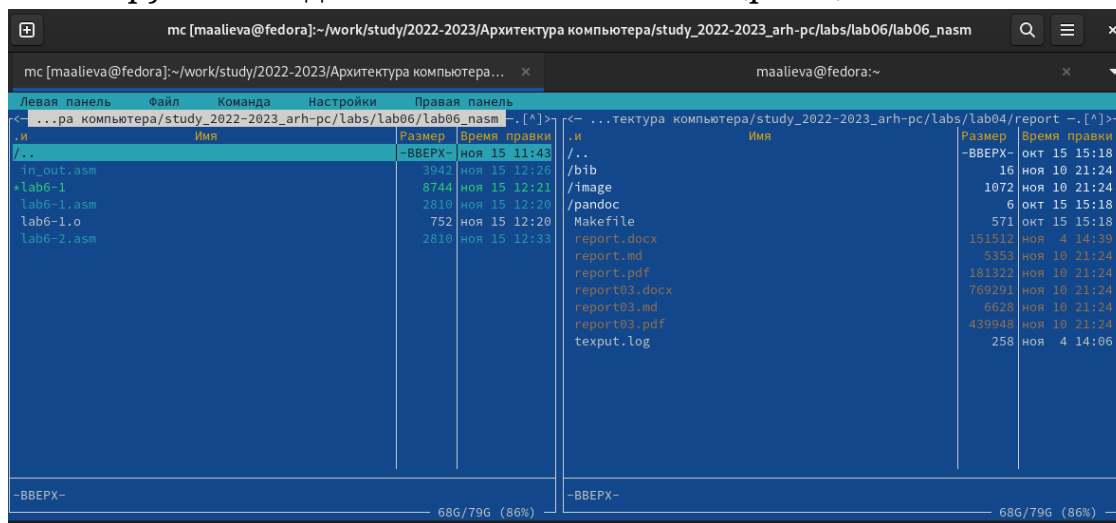


Рис. 6: Скачанный файл in_out.asm

7. С помощью функциональной клавиши F6 создаю копию файла lab6-1.asm с именем lab6-2.asm. Исправляю текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm (использую подпрограммы sprintLF, sread и quit) (рис. 7)


```

maalieva@fedora:~
...рхитектура компьютера/study_2022-2023_arh-pc/labs/lab06/lab06_nasm/lab6-2.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:',10 ; сообщение
; символ перевода строки
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
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

[ Прочитано 20 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/ К строке

```

Рис. 7: Текст программы в файле lab6-2.asm

8. Транслирую текст программы файла в объектный файл командой `nasm -f elf lab6-2.asm`, получаю объектный файл `lab6-2.o`, далее выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab6-2 lab6-2.o`, получаю исполняемый файл `lab6-2` (рис. 8)

Имя	Размер	Время	Имя	Размер	Время
in_out.asm	3942	ноя 15 11:43	report.docx	151512	ноя 4 11:43
lab6-1.asm	8744	ноя 15 12:21	report.md	5353	ноя 10 12:21
lab6-1.o	2810	ноя 15 12:20	report.pdf	181322	ноя 10 12:20
lab6-2.asm	752	ноя 15 12:47	report03.docx	769291	ноя 10 12:47
lab6-2.o	9092	ноя 15 12:46	report03.md	6628	ноя 10 12:46
	1561	ноя 15 12:46	report03.pdf	439948	ноя 10 12:46
	1312	ноя 15 12:46	texput.log	258	ноя 4 11:43

Рис. 8: Созданные файлы lab6-2.o и lab6-2

9. Запускаю исполняемый файл, программа просит ввести строку, ввожу имя и фамилию (рис. 9)

```
[maalieva@fedora lab06_nasm]$ ./lab6-2
Введите строку:
Alieva Milena
[maalieva@fedora lab06_nasm]$
```

Рис. 9: Исполнение файла, ввод имени и фамилии

10. Начинаю выполнять задания для самостоятельной работы. Создаю копию файла lab6-1.asm с именем lab6-1-1.asm с помощью функциональной клавиши F5 (рис. 10)

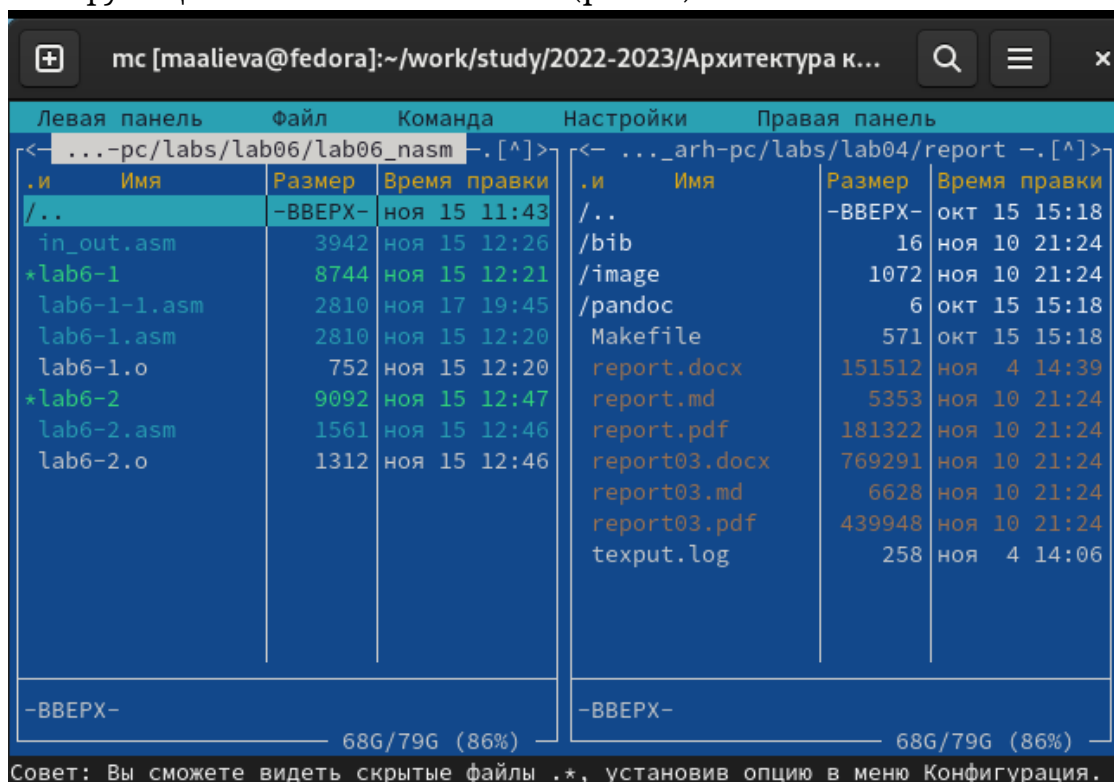


Рис. 10: Созданный файл с именем lab6-1-1.asm

11. С помощью функциональной клавиши F4 открываю созданный файл и изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 11)

```

lab6-1-1.asm  [----]  0 L: [ 13+30  43/ 43] *(2735/2735b) <EOF>  [*][X]
GLOBAL _start
_start:      ; Начало программы
; 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,4    ; Системный вызов для выхода (sys_exit)
mov ebx,1    ; Системный вызов для выхода (sys_exit)
mov ecx,buf1
mov edx,buf1
int 80h
mov eax,1
mov ebx,0
int 80h

```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Пере-тить 7Поиск 8Удалить 9МенюМС 10Выход

Рис. 11: Изменённая программа

12. Транслирую текст программы файла в объектный файл командой `nasm -f elf lab6-1-1.asm`, получаю объектный файл `lab6-1-1.o`, далее выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab6-1-1 lab6-1-1.o`, получаю исполняемый файл `lab6-1-1`. Программа просит ввести строку, ввожу свои ФИО, далее программа выводит введенные данные (рис. 12)

```

[maalieva@fedora lab06_nasm]$ ./lab6-1-1
Введите строку:
Alieva Milena Arifovna
Alieva Milena Arifovna
[maalieva@fedora lab06_nasm]$

```

Рис. 12: Исполнение файла, ввод имени, фамилии, отчества и их вывод

13. Создаю копию файла `lab6-2.asm` с именем `lab6-2-1.asm` с помощью функциональной клавиши F5 (рис. 13)

Левая панель		Файл	Команда	Настройки	Правая панель			
...		tudy_2022-2023_arh-pc/labs/lab06/lab06_nasm	-. [^]>		<-		...epa/study_2022-2023_arh-pc/labs/lab04/report	-. [^]>
и	Имя	Размер	Время правки		и	Имя	Размер	Время правки
..		-BVERX-	ноя 15 11:43		/..		-BVERX-	окт 15 15:18
in_out.asm		3942	ноя 15 12:26		/bib		16	ноя 10 21:24
lab6-1		8744	ноя 15 12:21		/image		1072	ноя 10 21:24
lab6-1-1		8748	ноя 17 20:00		/pandoc		6	окт 15 15:18
lab6-1-1.asm		2736	ноя 17 20:00		Makefile		571	окт 15 15:18
lab6-1-1.o		784	ноя 17 20:00		report.docx		151512	ноя 4 14:39
lab6-1.asm		2810	ноя 15 12:20		report.md		5353	ноя 10 21:24
lab6-1.o		752	ноя 15 12:20		report.pdf		181322	ноя 10 21:24
lab6-2		9092	ноя 15 12:47		report03.docx		769291	ноя 10 21:24
lab6-2-1.asm		1561	ноя 17 20:04		report03.md		6628	ноя 10 21:24
lab6-2.asm		1561	ноя 15 12:46		report03.pdf		439948	ноя 10 21:24
lab6-2.o		1312	ноя 15 12:46		texput.log		258	ноя 4 14:06

Рис. 13: Создание файла lab6-2-1.asm

- С помощью функциональной клавиши F4 открываю созданный файл и изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 14)

```

maalieva@fedora:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab06...
/home/maalieva/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/labs/lab06/lab06_nasm/lab6-2-1.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:',10 ; сообщение
; символ перевода строки
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
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,4
mov ebx,1
mov ecx,buf1
int 80h
call quit                     ; вызов подпрограммы завершения

```

Рис. 14: Изменённый код программы

- Транслирую текст программы файла в объектный файл командой `nasm -f elf lab6-2-1.asm`, получаю объектный файл lab6-2-1.o, далее выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab6-2-1 lab6-2-1.o`, получаю исполняемый файл lab6-2-1. Программа просит ввести строку, ввожу свои ФИО, далее программа выводит введенные данные (рис. 15)

```
[maalieva@fedora lab06_nasm]$ ./lab6-2-1
Введите строку:
Alieva Milena Arifovna
Alieva Milena Arifovna
[maalieva@fedora lab06_nasm]$
```

Рис. 15: Исполнение файла, ввод имени, фамилии, отчества и их вывод

5 Выводы

Я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера `mov` и `int`.

Список литературы