

Отчёт по лабораторной работе №6

Уткина Алина Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Подключение внешнего файла in_out.asm	8
2.2	Самостоятельная работа	10
3	Выводы	12

Список иллюстраций

2.1	Смена текущего каталога в Midnight Commander	6
2.2	Создание каталога	6
2.3	Создание файла	6
2.4	Редактор mcedit	7
2.5	Листинг 1	7
2.6	Заполнение файла текстом программы	8
2.7	Трансляция, компоновка и запуск файла lab6-1	8
2.8	Копирование файла	9
2.9	Создание копии файла	9
2.10	Листинг 2	10
2.11	Работа дополненной программы lab6-1	11
2.12	Работа дополненной программы lab6-2	11

Список таблиц

1 Цель работы

Целью данной работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

2 Выполнение лабораторной работы

С помощью команды `mc` откроем Midnight Commander, пользуясь клавишами `⌘`, `⌘` и `Enter` перейдем в каталог `~/work/arch-pc`, созданный при выполнении лабораторной работы №5 (рис. 2.1).

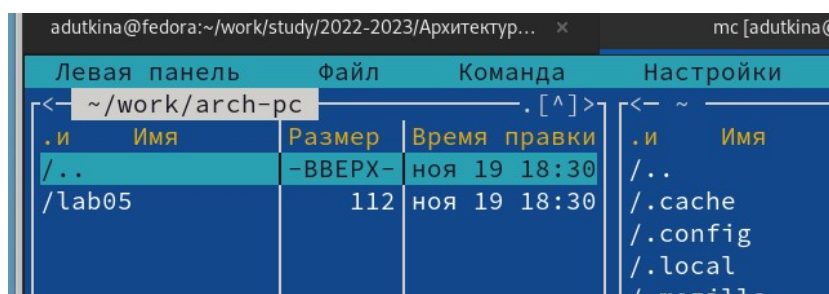


Рис. 2.1: Смена текущего каталога в Midnight Commander

С помощью функциональной клавиши `F7` создадим папку `lab06` (рис. 2.2) и перейдем в созданный каталог.

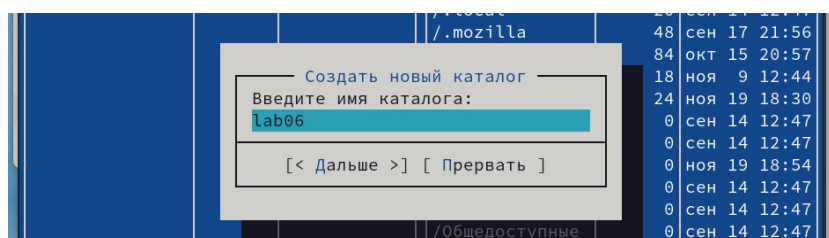


Рис. 2.2: Создание каталога

Пользуясь строкой ввода и командой `touch` создадим файл `lab6-1.asm` (рис. 2.3).

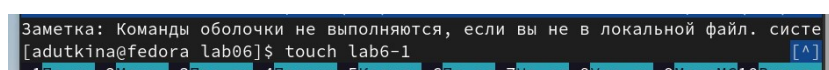


Рис. 2.3: Создание файла

С помощью функциональной клавиши F4 откроем файл lab6-1.asm для редактирования во встроенном редакторе mscedit (рис. 2.4).

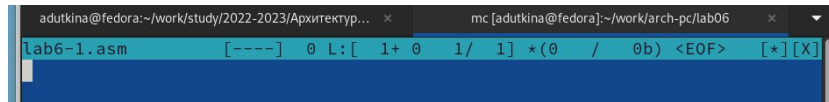


Рис. 2.4: Редактор mscedit

Введем текст программы из листинга 1 (рис. 2.5), сохраним изменения и закроем файл (рис. 2.6).

```

;-----
; 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

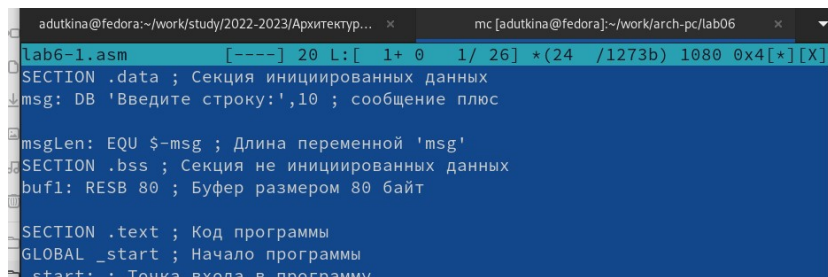
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра

;----- 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

```

Рис. 2.5: Листинг 1



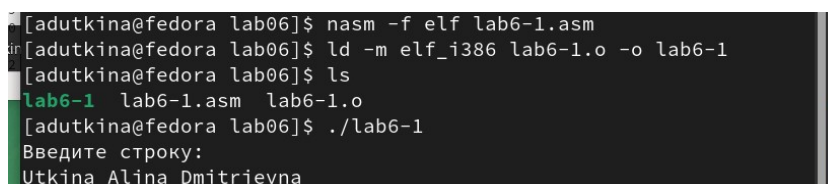
```
lab6-1.asm [----] 20 L: [ 1+ 0 1/ 26] *(24 /1273b) 1080 0x4[*][X]
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
msglen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт

SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
```

Рис. 2.6: Заполнение файла текстом программы

С помощью функциональной клавиши F3 откроем файл lab6-1.asm для просмотра. Убедимся, что файл содержит текст программы.

Оттранслируем текст программы lab6-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. Введем ФИО. (рис. 2.7)



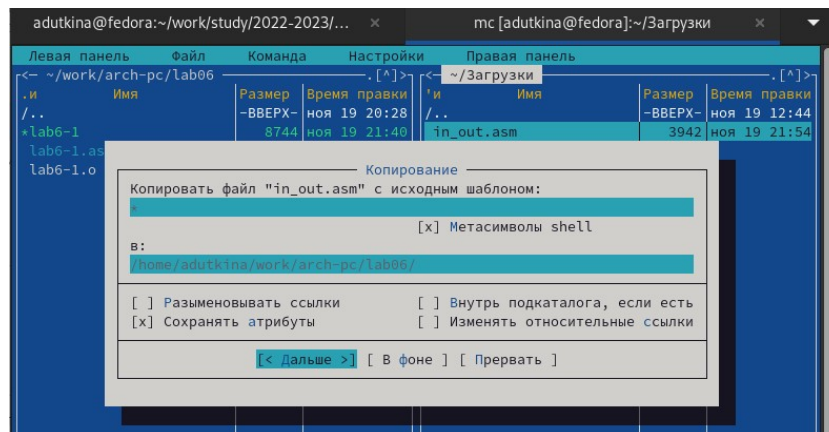
```
[adutkina@fedora lab06]$ nasm -f elf lab6-1.asm
[adutkina@fedora lab06]$ ld -m elf_i386 lab6-1.o -o lab6-1
[adutkina@fedora lab06]$ ls
lab6-1 lab6-1.asm lab6-1.o
[adutkina@fedora lab06]$ ./lab6-1
Введите строку:
Utkina Alina Dmitrievna
```

Рис. 2.7: Трансляция, компоновка и запуск файла lab6-1

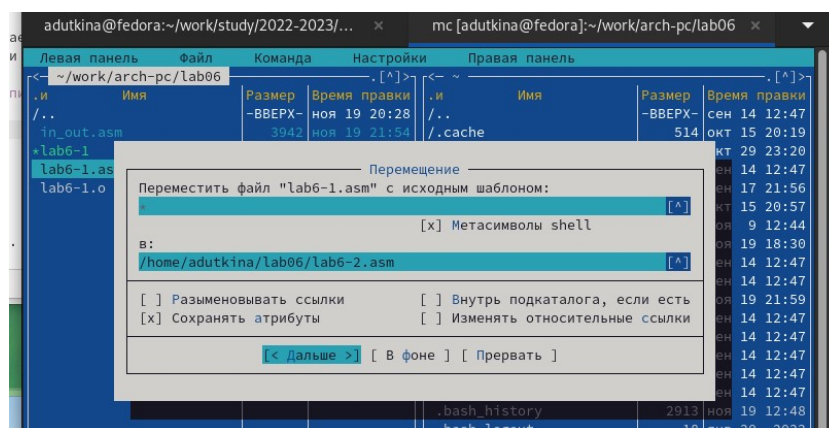
2.1 Подключение внешнего файла in_out.asm

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

В одной из панелей mc откроем каталог с файлом lab6-1.asm. В другой панели каталог со скаченным файлом in-out.asm. Скопируем файл in_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5 (рис. 2.8)



С помощью функциональной клавиши F6 создадим копию файла lab6-1.asm с именем lab6-2.asm (рис. 2.9)



Исправим текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 2 (рис. 2.10). Создадим исполняемый файл и проверим его работу.

```

;-----
; 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. 2.10: Листинг 2

В файле lab6-2.asm заменим подпрограмму sprintf на sprintf. При запуске программ разница в том, что в первом случае мы вводим данные в новую строку, а во втором - в ту же, где выводится сообщение “Введите строку:”

2.2 Самостоятельная работа

Создадим копию файла lab6-1.asm. Внесем 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.

Для этого вставим блок команд: - mov eax,4 ; Системный вызов для записи (sys_write) - mov ebx,1 ; Описатель файла 1 - стандартный вывод - mov ecx, buf1 ; Адрес строки 'buf1' в 'ecx' - int 80h ; Вызов ядра

(рис. 2.11)

```
[adutkina@fedora lab06]$ nasm -f elf lab6-1-2.asm
[adutkina@fedora lab06]$ ld -m elf_i386 lab6-1-2.o -o lab6-1-2
[adutkina@fedora lab06]$ ./lab6-1-2
Введите строку:
Уткина
Уткина
[adutkina@fedora lab06]$
```

Рис. 2.11: Работа дополненной программы lab6-1

Создадим копию файла lab6-2.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 введённую строку на экран.

Для этого можно использовать те же команды (рис. 2.12)

```
[adutkina@fedora lab06]$ cp lab6-2.asm lab6-2-2.asm
[adutkina@fedora lab06]$ nasm -f elf lab6-2-2.asm
[adutkina@fedora lab06]$ ld -m elf_i386 lab6-2-2.o -o lab6-2-2
[adutkina@fedora lab06]$ ./lab6-2-2
Введите строку:
Уткина
Уткина
[adutkina@fedora lab06]$
```

Рис. 2.12: Работа дополненной программы lab6-2

3 Выводы

В ходе лабораторной работы были изучены основы работы с Midnight Commander, структура программы на языке ассемблера NASM, элементы программирования. Также были приобретены практические навыки работы в Midnight Commander и освоены инструкции языка ассемблера `mov` и `int`.