

Лабораторная работа №6

Простейший вариант

Кучеренко София

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Самостоятельная работа	10
4	Выводы	13

Список иллюстраций

2.1	Создание папки	6
2.2	Создание файла	6
2.3	Текст программы	7
2.4	Запуск программы	7
2.5	Копирование файла	8
2.6	Создание файла lab6-2.asm	8
2.7	Результат запуска программы	9
2.8	Результат запуска новой программы	9
3.1	Код программы lab6-3.asm	11
3.2	Результат запуска программы	11
3.3	Код программы lab6-4.asm	12
3.4	Результат запуска программы	12

Список таблиц

1 Цель работы

Приобрести практические навыки работы в Midnight Commander и освоить инструкции языка ассемблера mov и int.

2 Выполнение лабораторной работы

1. Откроем Midnight Commander и создадим папку lab06:

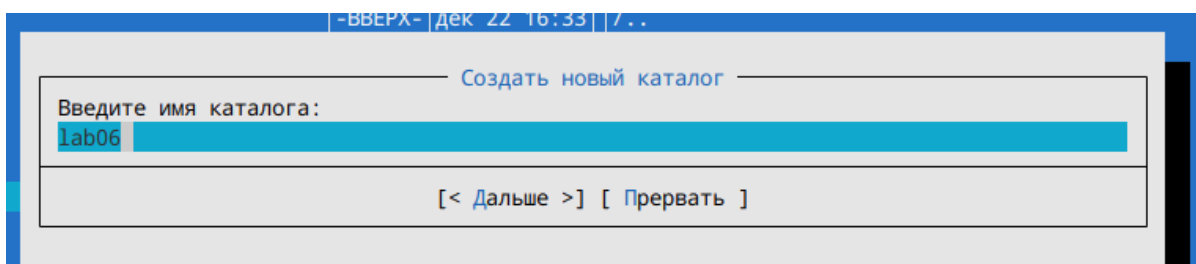


Рис. 2.1: Создание папки

2. Создадим файл lab6-1.asm:

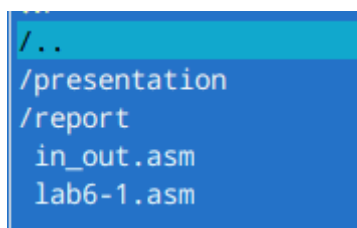


Рис. 2.2: Создание файла

3. С помощью клавиши F4 откроем файл lab6-1.asm, введем текст программы и сохраним изменения (F2). С помощью клавиши F3 посмотрим файл:

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.r
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'

^G Справка      ^O Записать     ^W Поиск        ^K Вырезать
^X Выход        ^R ЧитФайл     ^\ Замена       ^U Вставить
```

Рис. 2.3: Текст программы

4. Создадим исполняемый файл и запустим его:

```
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ./lab6-1
Введите строку:
Кучеренко София Михайловна
```

Рис. 2.4: Запуск программы

5. Скопируем файл in_out.asm с помощью клавиши F6:

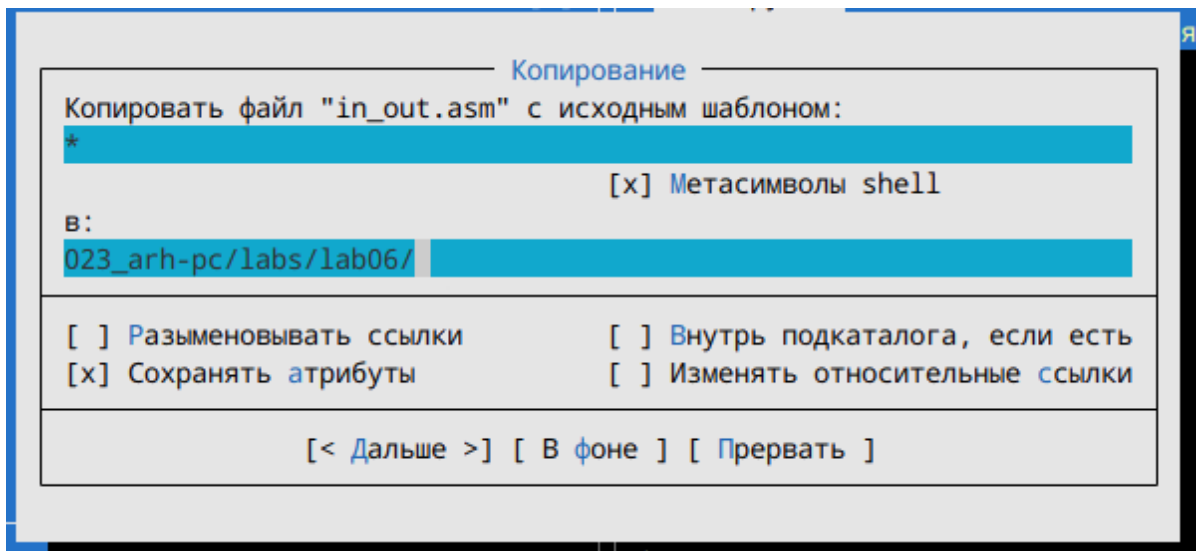


Рис. 2.5: Копирование файла

- С помощью клавиши F6 создадим копию файла lab6-1.asm с именем lab6-2.asm:

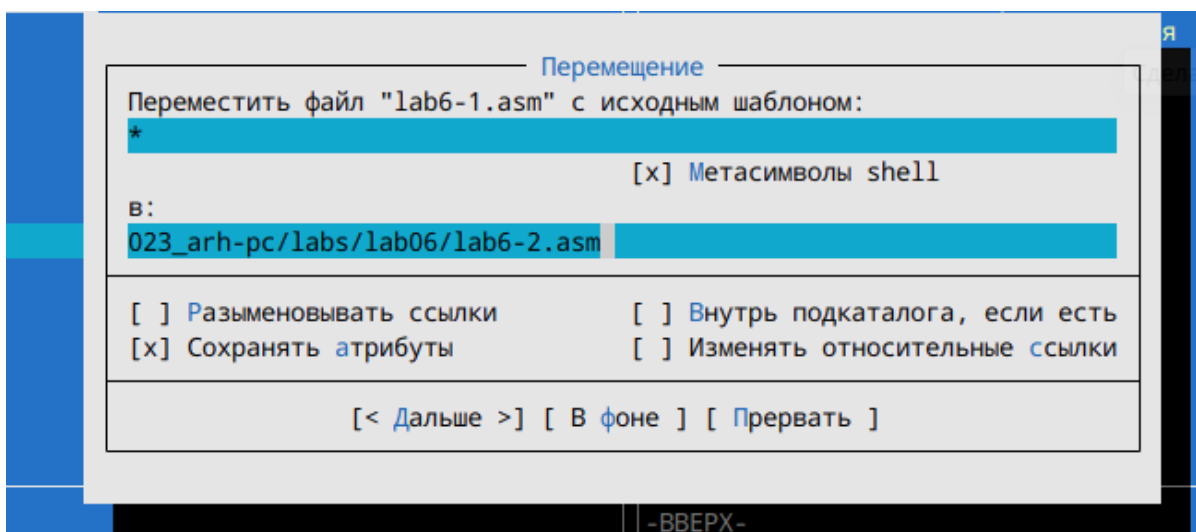


Рис. 2.6: Создание файла lab6-2.asm

- Исправим текст программы lab6-2.asm , создадим исполняемый файл и запустим:


```
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ nasm -f elf lab6-2.asm
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ./lab6-2
Введите строку:
Кучеренко София Михайловна
```

Рис. 2.7: Результат запуска программы

8. Заменим в файле lab6-2.asm подпрограмму sprintLF на sprint, создадим исполняемый файл и запустим(мы увидим, что строка вывелась не с новой строки):

```
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ nasm -f elf lab6-2.asm
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ./lab6-2
Введите строку: Кучеренко София Михайловна
```

Рис. 2.8: Результат запуска новой программы

3 Самостоятельная работа

1. Напишем программу, исправив код файла lab6-1.asm и проверим её работу:

```

1  SECTION .data ; Секция инициализированных данных
2  msg: DB 'Введите строку:',10 ; сообщение плюс
3  ; символ перевода строки
4  msgLen: EQU $-msg ; Длина переменной 'msg'
5  SECTION .bss ; Секция не инициализированных данных
6  buf1: RESB 80 ; Буфер размером 80 байт
7  ;----- Текст программы -----
8  SECTION .text ; Код программы
9  GLOBAL _start ; Начало программы
10 _start: ; Точка входа в программу
11 ;----- Системный вызов `write`
12 ; После вызова инструкции 'int 80h' на экран будет
13 ; выведено сообщение из переменной 'msg' длиной 'msgLen'
14 mov eax,4 ; Системный вызов для записи (sys_write)
15 mov ebx,1 ; Описатель файла 1 - стандартный вывод
16 mov ecx,msg ; Адрес строки 'msg' в 'ecx'
17 mov edx,msgLen ; Размер строки 'msg' в 'edx'
18 int 80h ; Вызов ядра
19 ;----- системный вызов `read` -----
20 ; После вызова инструкции 'int 80h' программа будет ожидать ввода
21 ; строки, которая будет записана в переменную 'buf1' размером 80 байт
22 mov eax, 3 ; Системный вызов для чтения (sys_read)
23 mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
24 mov ecx, buf1 ; Адрес буфера под вводимую строку
25 mov edx, 80 ; Длина вводимой строки
26 int 80h ; Вызов ядра
27 mov eax,4 ; Системный вызов для записи (sys_write)
28 mov ebx,1 ; Описатель файла 1 - стандартный вывод
29 mov ecx,buf1 ; Адрес строки 'msg' в 'ecx'
30 mov edx,msgLen ; Размер строки 'msg' в 'edx'
31 int 80h ; Вызов ядра
32 ;----- Системный вызов `exit` -----
33 ; После вызова инструкции 'int 80h' программа завершит работу
34 mov eax,1 ; Системный вызов для выхода (sys_exit)
35 mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
36 int 80h ; Вызов ядра

```

Рис. 3.1: Код программы lab6-3.asm

```
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ nasm -f elf lab6-3.asm
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
^[[A][A][A[smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ./lab6-3
Введите строку:
rainbow
rainbow
```

Рис. 3.2: Результат запуска программы

2. Напишем программу, исправив код файла lab6-2.asm и проверим её работу:

```
1  %include 'in_out.asm' ; подключение внешнего файла
2  SECTION .data ; Секция инициированных данных
3  msg: DB 'Введите строку: ',0h ; сообщение
4  SECTION .bss ; Секция не инициированных данных
5  buf1: RESB 80 ; Буфер размером 80 байт
6  SECTION .text ; Код программы
7  GLOBAL _start ; Начало программы
8  _start: ; Точка входа в программу
9  mov eax, msg ; запись адреса выводимого сообщения в `EAX`
10 call sprintf ; вызов подпрограммы печати сообщения
11
12 mov ecx, buf1 ; запись адреса переменной в `EAX`
13 mov edx, 80 ; запись длины вводимого сообщения в `EBX`
14 call sread ; вызов подпрограммы ввода сообщения
15 mov eax, buf1
16 call sprintf
17 call quit ; вызов подпрограммы завершения
```

Рис. 3.3: Код программы lab6-4.asm

```
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ nasm -f elf lab6-4.asm
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ld -m elf_i386 -o lab6-4 lab6-4.o
smkucherenko@dk2n24 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab06 $ ./lab6-4
Введите строку:
rainbow
rainbow
```

Рис. 3.4: Результат запуска программы

4 Выводы

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