

Лабораторная работа 6

Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Ерёмин Даниил

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	11

Список иллюстраций

2.1	запуск окна Midnight Commander	5
2.2	созданный файл	5
2.3	редактирование файла	6
2.4	проверка текста программы	6
2.5	обработка файлов компоновщиком LD	7
2.6	копирование файла в другой каталог	7
2.7	создание копии файла	7
2.8	корректировка текста программы	8
2.9	замена подпрограммы sprintLF на sprint	8
2.10	запуск программного файла	9
2.11	корректировка текста программы с использованием подпрограмм	9
2.12	создание исполняемого файла	10

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Выполнение лабораторной работы

1) С помощью команды 'mc' запускаю Midnight Commander, после чего перехожу в каталог ~/work/arch-rc (рис. 2.1)

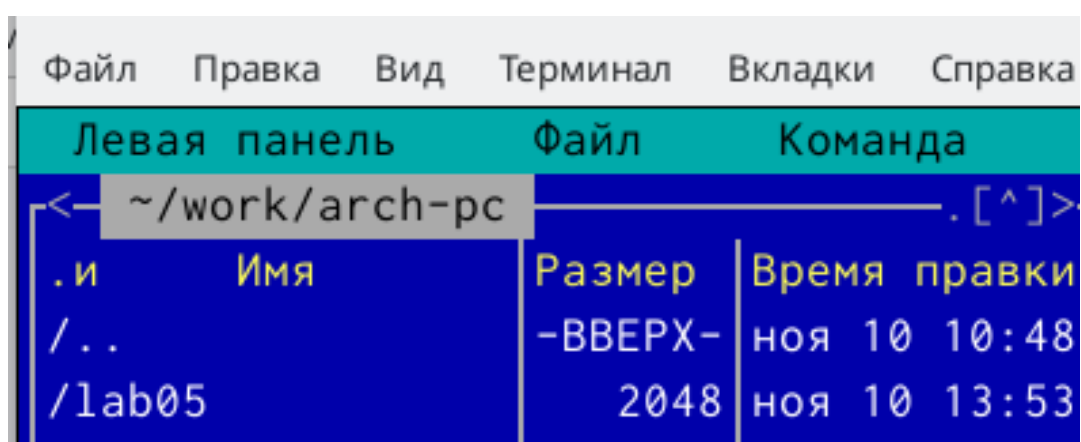


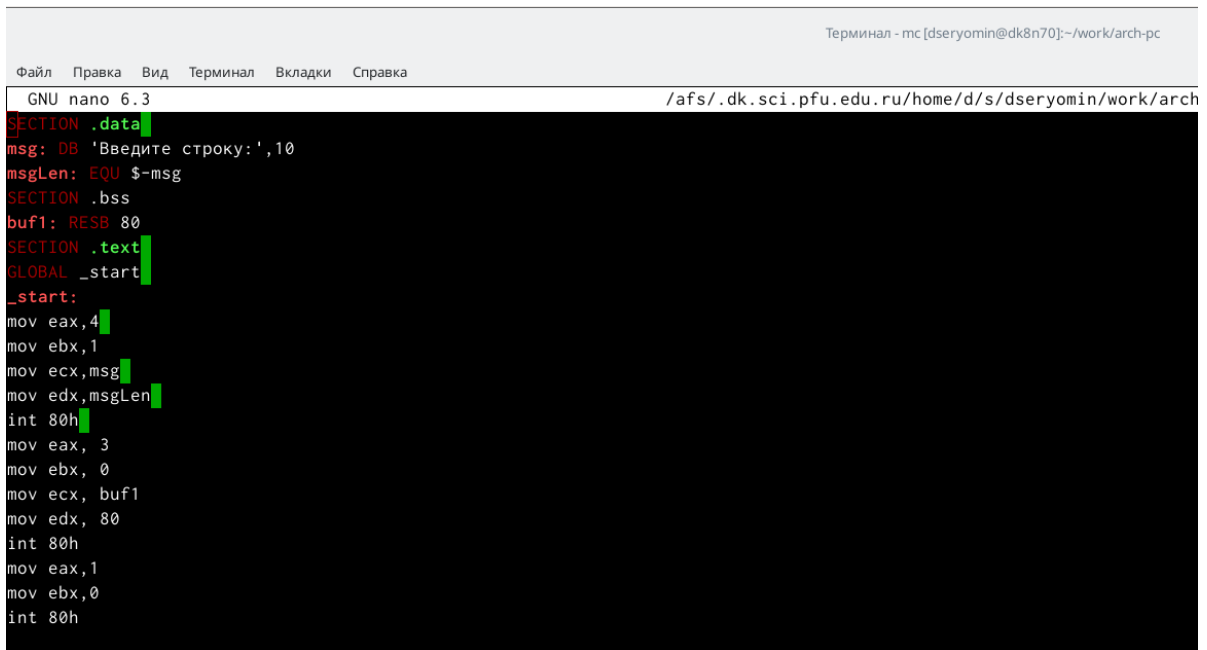
Рис. 2.1: запуск окна Midnight Commander

2) Пользуясь строкой ввода и командой touch создаю файл lab6-1.asm в папке lab06, созданной мной клавишей f7 (рис.2.2)



Рис. 2.2: созданный файл

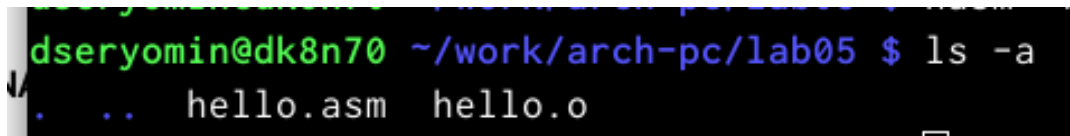
3) С помощью клавиши f4 редактирую созданный файл, вводя в него текст из листинга 6.1 (рис. 2.3)



```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/s/dseryomin/work/arch
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 2.3: редактирование файла

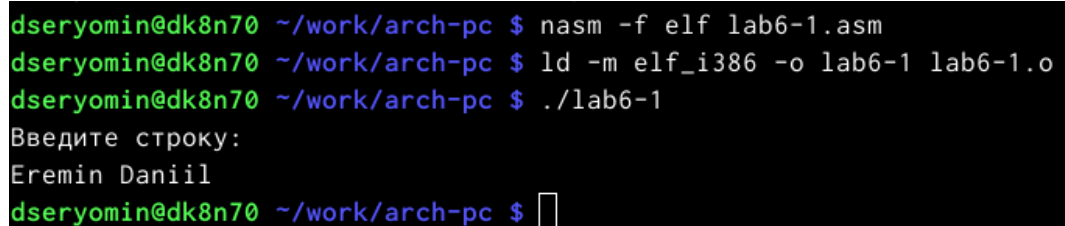
- 4) Убеждаюсь, что файл содержит текст программы с помощью клавиши f3 (рис. 2.4)



```
dseryomin@dk8n70 ~/work/arch-pc/lab05 $ ls -a
.  ..  hello.asm  hello.o
```

Рис. 2.4: проверка текста программы

- 5) Транслирую текст программы lab6-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл



```
dseryomin@dk8n70 ~/work/arch-pc $ nasm -f elf lab6-1.asm
dseryomin@dk8n70 ~/work/arch-pc $ ld -m elf_i386 -o lab6-1 lab6-1.o
dseryomin@dk8n70 ~/work/arch-pc $ ./lab6-1
Введите строку:
Eremin Daniil
dseryomin@dk8n70 ~/work/arch-pc $
```

(рис. ??)

- 6) передаю объектный файл на обработку компоновщику с помощью команды `ld -m elf_i386 hello.o -o hello`, на предыдущем скриншоте видно, что исполня-

емый файл был создан; после чего выполняю команду 'ld -m elf_i386 obj.o -o main' (рис. 2.5)

```
dseryomin@dk8n70 ~/work/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
```

Рис. 2.5: обработка файлов компоновщиком LD

- 7) копирую файл in_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши f5 (рис. 2.6)

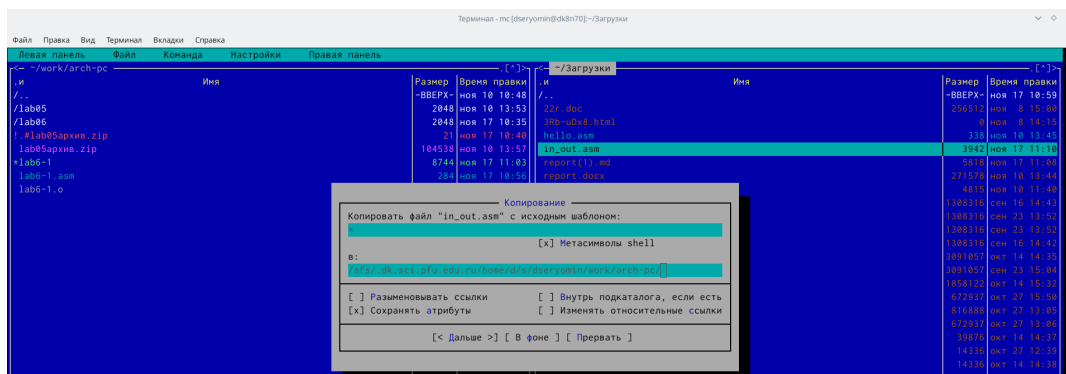


Рис. 2.6: копирование файла в другой каталог

- 8) с помощью клавиши f6 создаю копию файла lab6-1.asm с именем lab6-2.asm (рис. 2.7)

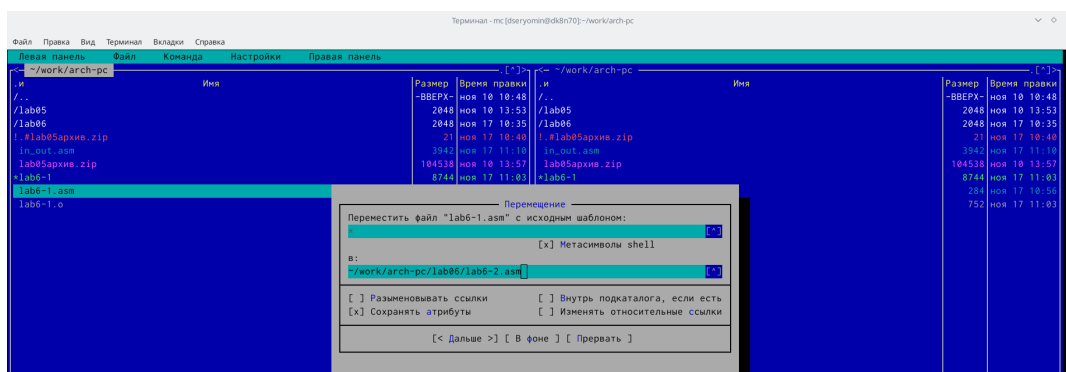
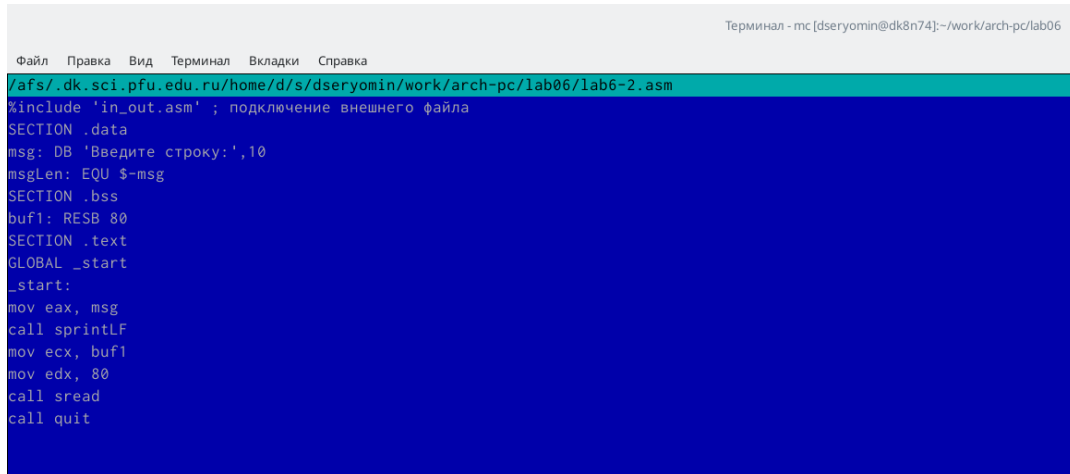


Рис. 2.7: создание копии файла

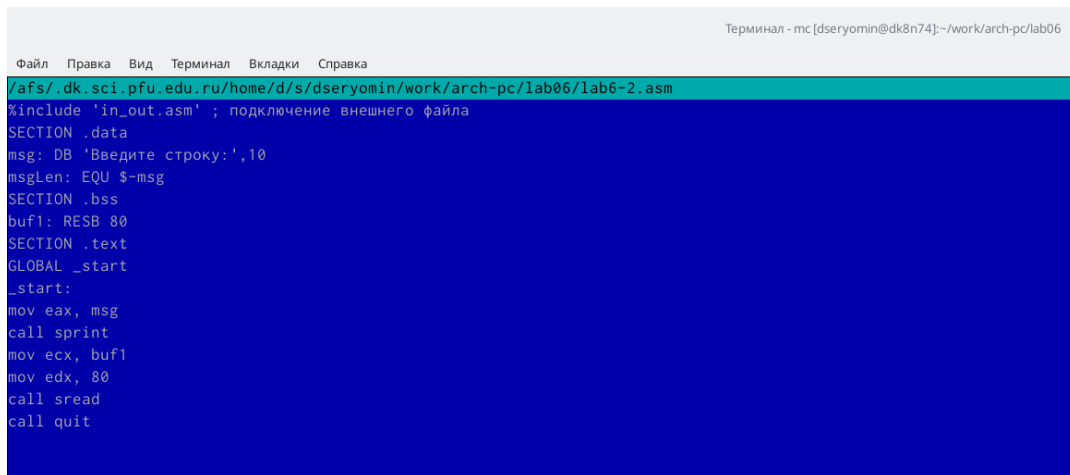
- 9) Исправляю текст программы в файле lab6-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используя подпрограммы `sprintLF`, `sread` и `quit`) в соответствии с листингом 6.2. После этого создаю исполняемый файл и выполняю работу. (рис. 2.8)



```
Терминал - mc [dseryomin@dk&n74]:~/work/arch-pc/lab06
Файл  Правка  Вид  Терминал  Вкладки  Справка
/afs/.dk.sci.pfu.edu.ru/home/d/s/dseryomin/work/arch-pc/lab06/lab6-2.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 2.8: корректировка текста программы

- 10) В файле lab6-2.asm заменяю подпрограмму `sprintLF` на `sprint`. Создаю исполняемый файл и проверяю его работу. (рис. 2.9)



```
Терминал - mc [dseryomin@dk&n74]:~/work/arch-pc/lab06
Файл  Правка  Вид  Терминал  Вкладки  Справка
/afs/.dk.sci.pfu.edu.ru/home/d/s/dseryomin/work/arch-pc/lab06/lab6-2.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 2.9: замена подпрограммы `sprintLF` на `sprint`


```
Терминал - dseryomin@dk8n74:~/work/arch-pc/lab06
Файл  Правка  Вид  Терминал  Вкладки  Справка
dseryomin@dk8n74 ~ $ cd ~/work/arch-pc/lab06
dseryomin@dk8n74 ~/work/arch-pc/lab06 $ nasm -f elf lab6-2.asm
dseryomin@dk8n74 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ld: невозможно найти lab6-2.0: Нет такого файла или каталога
dseryomin@dk8n74 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
ld: невозможно найти lab6-2.0: Нет такого файла или каталога
dseryomin@dk8n74 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
dseryomin@dk8n74 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку:
Eremin Daniil
dseryomin@dk8n74 ~/work/arch-pc/lab06 $
```

Рис. 2.10: запуск программного файла

Задание для самостоятельной работы:

- 1) Исправляю текст программы с использованием 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. (рис. 2.11)

```
Терминал - mc [dseryomin@dk8n74]:~/work/arch-pc
Файл  Правка  Вид  Терминал  Вкладки  Справка
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/s/dseryomin/work/arch-pc
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите строку:',10

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

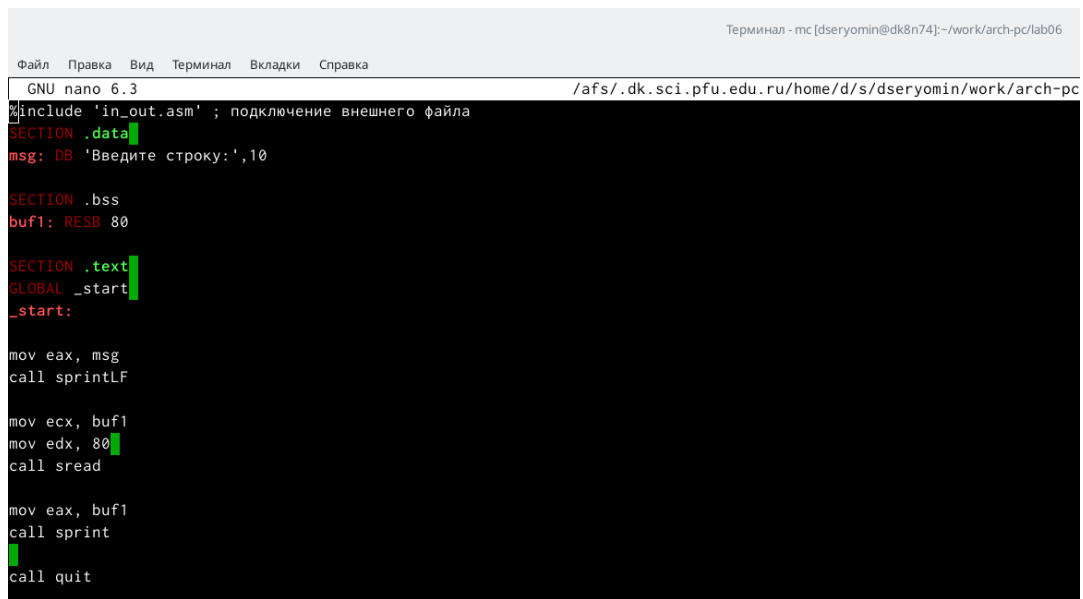
mov eax, msg
call sprintf

mov ecx, buf1
mov edx, 80
call sread

mov eax, buf1
call sprint
call quit
```

Рис. 2.11: корректировка текста программы с использованием подпрограмм

- 2) создаю исполняемый файл и проверяю его работу (рис. 2.12)



```
Терминал - mc [dseryomin@dk&n74]:~/work/arch-pc/lab06
Файл  Правка  Вид  Терминал  Вкладки  Справка
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/d/s/dseryomin/work/arch-pc
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите строку:',10

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintLF

mov ecx, buf1
mov edx, 80
call sread

mov eax, buf1
call sprint
call quit
```

Рис. 2.12: создание исполняемого файла

3) Делаю тоже самое с файлом lab6-1, но без использования подпрограмм из внешнего файла in_out.asm

3 Выводы

В ходе выполнения данной лабораторной работы я приобрел практические навыки работы в Midnight Commander, а также я освоил инструкцию языка ассемблера `mov` и `int`.