

Отчёт по лабораторной работе 5

Структура программ на языке ассемблера NASM. Системные
вызовы в ОС GNU Linux

Тимошенко Анна Михайловна

Содержание

Цель работы	1
Задание	1
Теоретическое введение	2
Выполнение лабораторной работы	2
Выводы.....	8
Список литературы.....	8

1. Цель работы
2. Задание
3. Теоретическое введение
4. Выполнение лабораторной работы
5. Вывод

Цель работы

Изучить структуру программы на языке ассемблера NASM

Задание

1. Открыть Midnight Commander
2. Создать папку lab05 и внутри нее создать файл lab5-1.asm
3. Открыть файл lab5-1.asm, ввести информацию из листинга 5.1 и сохранить изменения
4. Убедится, что файл содержит информацию
5. Оттранслировать текст файла lab5-1.asm, выполнить компоновку объектного файла
6. Запустить файл
7. Скачать и скопировать файл in_out.asm с помощью клавиши f5
8. С помощью клавиши f6 скопировать файл lab5-1.asm с именем lab5-2.asm
9. Исправить файл lab5-2.asm в соответствии с листингом 5.2
10. В файле lab5-2.asm заменить подпрограмму sprintLF на sprint
11. Создать исполняемый файл и проверить его работу

12. Создать копию файла lab5-1.asm и внести изменения, чтобы выводила введенная строка на экран
13. Создать копию файла lab5-2.asm и внести изменения, чтобы выводила введенная строка на экран

Теоретическое введение

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). Таким образом, общая структура программы имеет следующий вид: SECTION .data ; Секция содержит переменные, для ... ; которых задано начальное значение

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде mov dst,src Здесь операнд dst — приёмник, а src — источник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const). В табл. 5.4 приведены варианты использования mov с разными операндами.

Таблица 5.4. Варианты использования mov с разными операндами

Тип операндов	Пример	Пояснение
mov eax,ebx	пересылает значение регистра ebx в регистр eax	
mov cx,[eax]	пересылает в регистр cx значение из памяти, указанной в eax	
mov rez,ebx	пересылает в переменную rez значение из регистра ebx	
mov eax,403045h	пишет в регистр eax значение 403045h	
mov byte[rez],0	записывает в переменную rez значение 0	

Выполнение лабораторной работы

1. Открыть Midnight Commander (см рис 1)

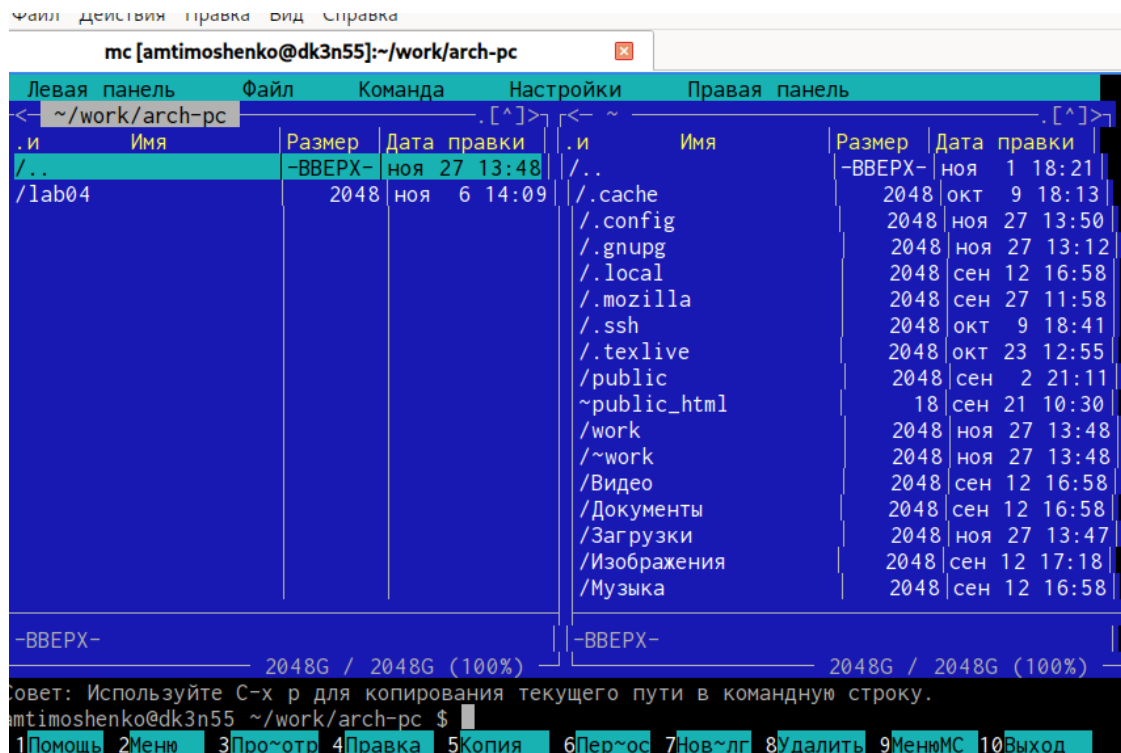


Рис.1 Открытый MC

2. Создать папку lab05 и внутри нее создать файл lab5-1.asm (см рис 2)

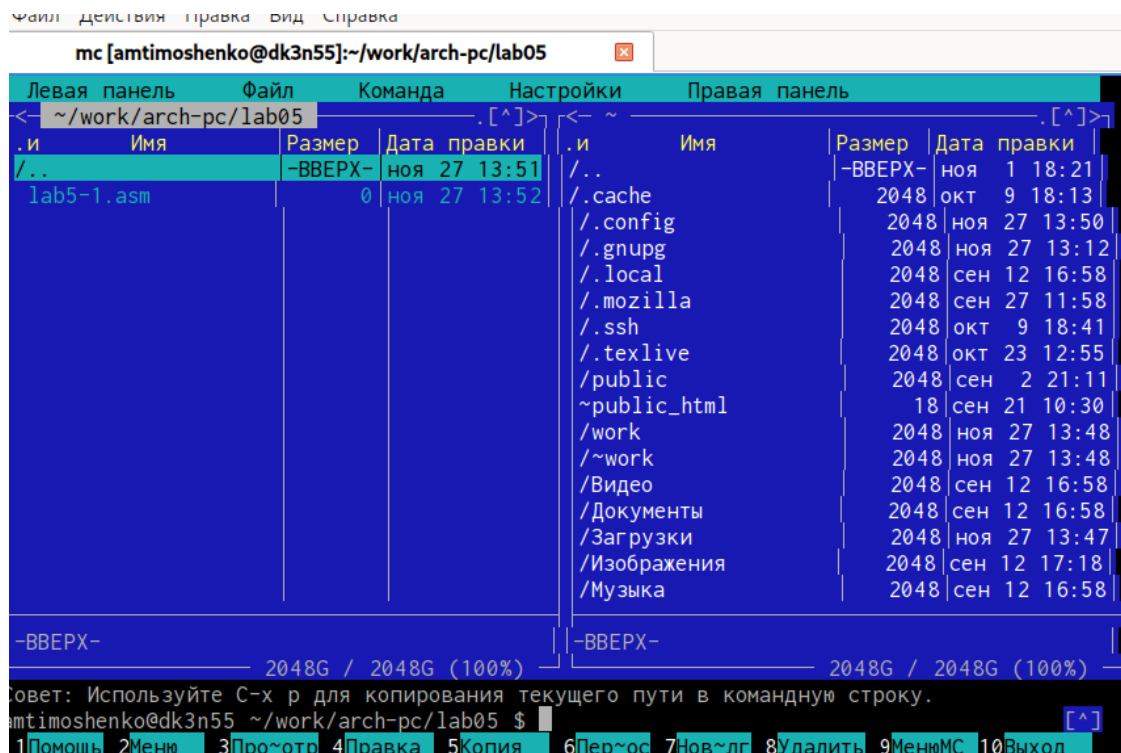


Рис. 2 Создание папки lab05 и файла lab5-1.asm

3. Открыть файл lab5-1.asm, ввести информацию из листинга 5.1 и сохранить изменения (см рис 3)

```
lab5-1.asm [-M--] 0 L: [ 1+21 22/ 22] *(277 / 277b) <EOF> [*][X]
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. 3 Открытый файл lab5-1.asm

4. Убедится, что файл содержит информацию (см рис 3)
5. Оттранслировать текст файла lab5-1.asm, выполнить компоновку объектного файла (см рис 4)

```
henko@dk3n55]:~/work/arch-pc/lab05 amtimoshenko@dk3n55 - lab05
amtimoshenko@dk3n55 ~ $ cd ~/work/arch-pc/lab05
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Тимошенко Анна Михайловна
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $
```

Рис.4 Выполнение команд

6. Запустить файл (см рис 4)

7. Скачать и скопировать файл in_out.asm с помощью клавиши f5 (см рис 5)

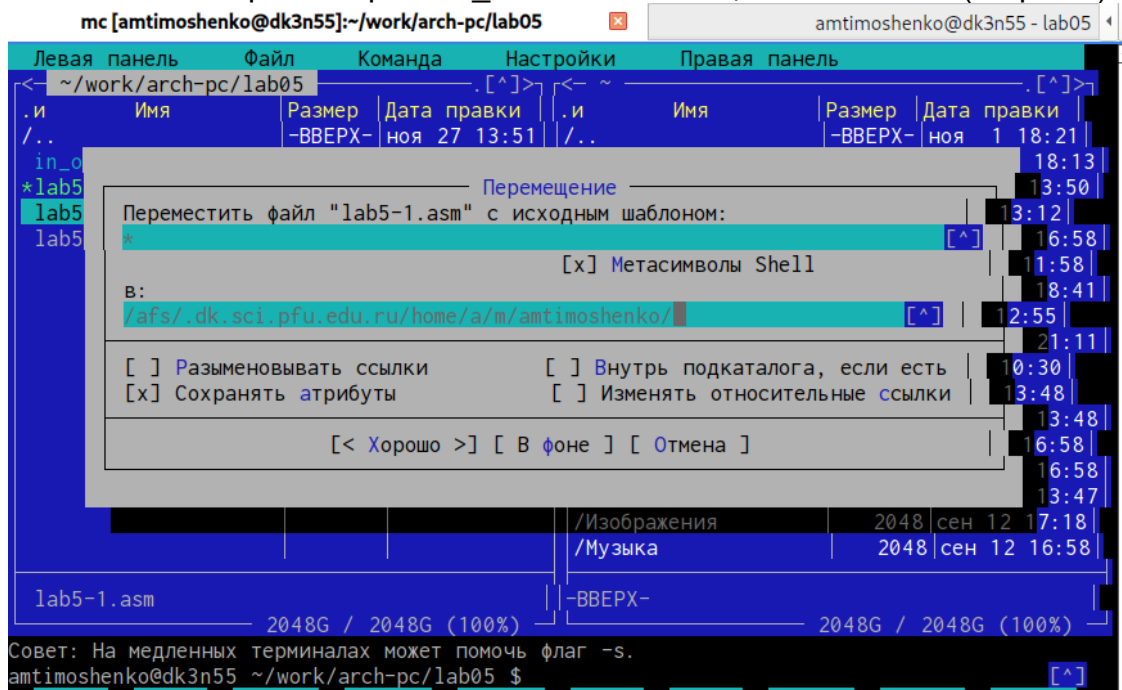


Рис.5 Скопированный in_out.asm через f5

8. С помощью клавиши f6 скопировать файл lab5-1.asm с именем lab5-2.asm (см рис 6)

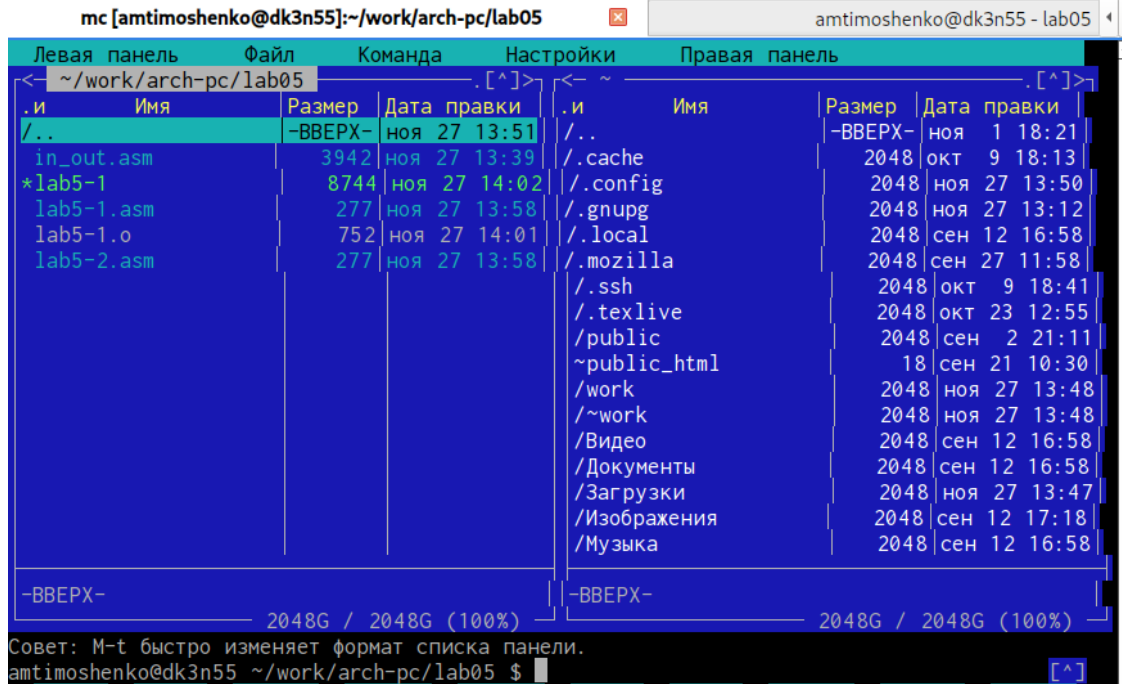
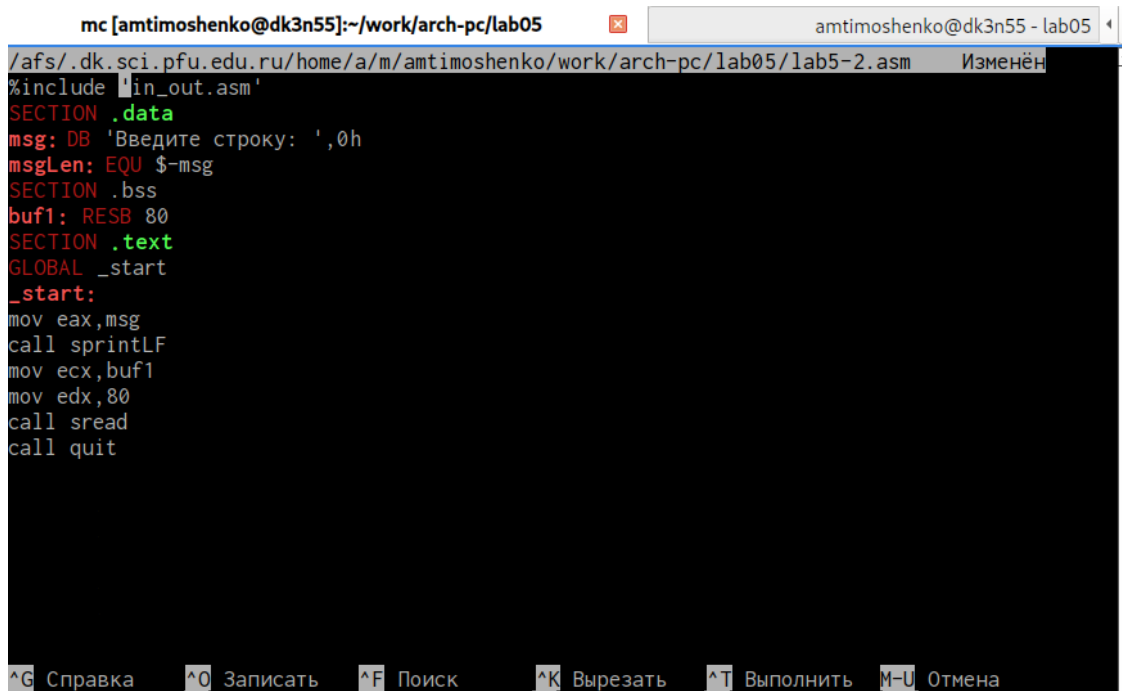


Рис.6 Скопированный файл lab5-1.asm с именем lab5-2.asm

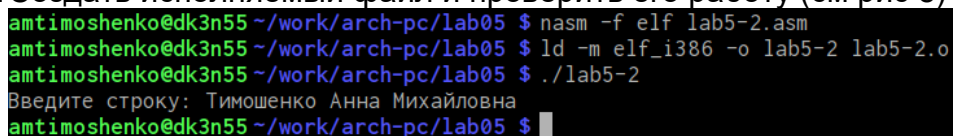
9. Исправить файл lab5-2.asm в соответствии с листингом 5.2 и заменить подпрограмму sprintLF на sprint (см рис 7)



```
mc [amtimoshenko@dk3n55]:~/work/arch-pc/lab05
/afs/.dk.sci.pfu.edu.ru/home/a/m/amtimoshenko/work/arch-pc/lab05/lab5-2.asm
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
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.7 Исправленный файл lab5-2.asm

10. Создать исполняемый файл и проверить его работу (см рис 8)



```
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Тимошенко Анна Михайловна
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $
```

Рис.8 Проверка и создание файла

Заметна разница: теперь после вывода сообщения нет перехода на новую строку.

11. Создать копию файла lab5-1.asm и внести изменения, чтобы выводила введенная строка на экран (см рис 9-10)

```

;----- 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

```

Рис.9 Вносим изменения

```

amtimoshenko@edk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
amtimoshenko@edk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
amtimoshenko@edk3n55 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Тимошенко Анна Михайловна
Тимошенко Анна Михайловна
amtimoshenko@edk3n55 ~/work/arch-pc/lab05 $ █

```

Рис.10 Вывод введенной строки на экран

12. Создать копию файла lab5-2.asm и внести изменения, чтобы выводила введенная строка на экран (см рис 11-12)

```

/afs/.dk.sci.pfu.edu.ru/home/a/m/amtimoshenko/work/arch-pc/lab05/lab5-4.asm  Изменён
#include 'in_out.asm' ; подключение внешнего файла
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
enterStroke DB 'Введите строку: ',0h ; сообщение

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

mov eax, enterStroke ; запись адреса выводимого сообщения в 'EAX'
call sprint ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; адрес строки переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'

call sread ; вызов подпрограммы ввода сообщения

```

Рис.11 Вносим изменения

```
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку: Тимошенко Анна Михайловна
Тимошенко Анна Михайловна
amtimoshenko@dk3n55 ~/work/arch-pc/lab05 $ █
```

Рис.12 Вывод введенной строки на экран

Выводы

В процессе выполнения лабораторной работы я ознакомилась со структурой программы на языке ассемблера NASM

Список литературы