

# Лабораторная работа №4

## Создание и процесс обработки программ на языке ассемблера NASM

Тимошенко Анна Михайловна

### Содержание

Цель работы .....	1
Задание .....	1
Теоретическое введение .....	2
Выводы.....	5

Цель работы  
Задание  
Теоретическое введение  
Выполнение лабораторной работы  
Выводы

### Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере  
NASM

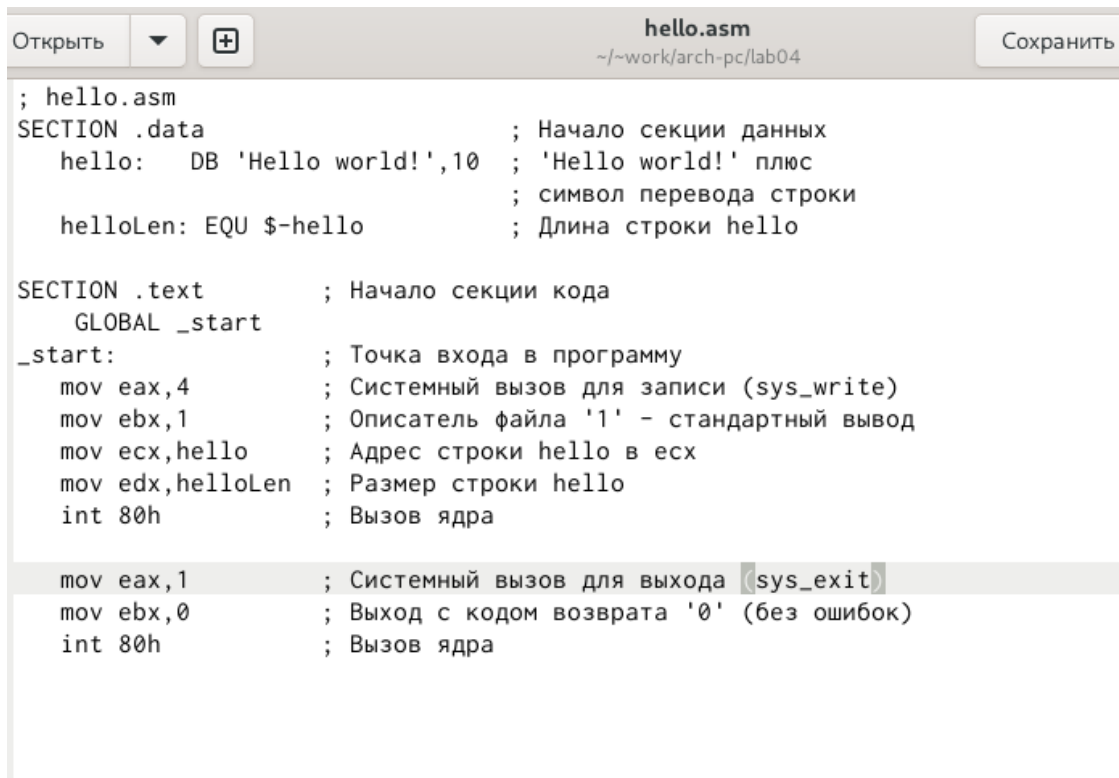
### Задание

1. Программа HELLO WORLD! 1.1 создать каталог для работы с программами на языке NASM 1.2 перейти в созданный каталог 1.3 создать текстовый файл с именем hello.asm 1.4 открыть файл 1.5 ввести в него указанный текст
2. Транслятор NASM 2.1 выполнить компиляцию в объектный код
3. Расширенный синтаксис 3.1 выполнить компиляцию файла
4. Компоновщик LD 4.1 передать объектный файл на обработку компоновщику
5. Запустить исполняемый файл
6. Задание для самостоятельной работы 6.1 создать копию файла hello.asm с именем lab4.asm 6.2 изменить скопированный файл, чтобы выводилась строка с именем и фамилией 6.3 оттранслировать полученный текст программы lab4.asm в объектный файл 6.4 скопировать файлы hello.asm и lab4.asm в локальный репозиторий

# Теоретическое введение

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня, таких как C/C++, Perl, Python и пр. Заметим, что получить полный доступ к ресурсам компьютера в современных архитектурах нельзя, самым низким уровнем работы прикладной программы является обращение напрямую к ядру операционной системы. Именно на этом уровне и работают программы, написанные на ассемблере. Но в отличие от языков высокого уровня ассемблерная программа содержит только тот код, который ввёл программист. Таким образом язык ассемблера — это язык, с помощью которого понятным для человека образом пишутся команды для процессора. Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды. До появления языков ассемблера программистам приходилось писать программы, используя только лишь машинные коды, которые были крайне сложны для запоминания, так как представляли собой числа, записанные в двоичной или шестнадцатеричной системе счисления. Преобразование или трансляция команд с языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер # Выполнение лабораторной работы

1. Программа HELLO WORLD! (см. рис.1)
  - 1.1 создать каталог для работы с программами на языке NASM (см. рис.1)
  - 1.2 перейти в созданный каталог (см. рис.1)
  - 1.3 создать текстовый файл с именем hello.asm (см. рис.1)
  - 1.4 открыть файл (см. рис.1)
  - 1.5 ввести в него указанный текст (см. рис.2)



```
; hello.asm
SECTION .data                ; Начало секции данных
    hello: DB 'Hello world!',10 ; 'Hello world!' плюс
                                ; символ перевода строки
    helloLen: EQU $-hello      ; Длина строки hello

SECTION .text                ; Начало секции кода
    GLOBAL _start
_start:                      ; Точка входа в программу
    mov eax,4                ; Системный вызов для записи (sys_write)
    mov ebx,1                ; Описатель файла '1' - стандартный вывод
    mov ecx,hello            ; Адрес строки hello в ecx
    mov edx,helloLen         ; Размер строки hello
    int 80h                  ; Вызов ядра

    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 Ввод текста в файл hello.asm

2. Транслятор NASM (см. рис.1)
  - 2.1 выполнить компиляцию в объектный код (см. рис.1)
3. Расширенный синтаксис (см. рис.1)
  - 3.1 выполнить компиляцию файла (см. рис.1)
4. Компановщик LD (см. рис.1)
  - 4.1 передать объектный файл на обработку компоновщику (см. рис.1)
5. Запустить исполняемый файл (см. рис.1)

```

amtimoshenko@dk2n21 ~ $ cd ~/work/arch-pc/lab04
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ touch hello.asm
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ gedit hello.asm

amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ ./hello
Hello world!
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $

```

РИС.1 Выполнение лабораторной работы

## 6. Задание для самостоятельной работы

### 6.1 создать копию файла hello.asm с именем lab4.asm (см. рис.3)

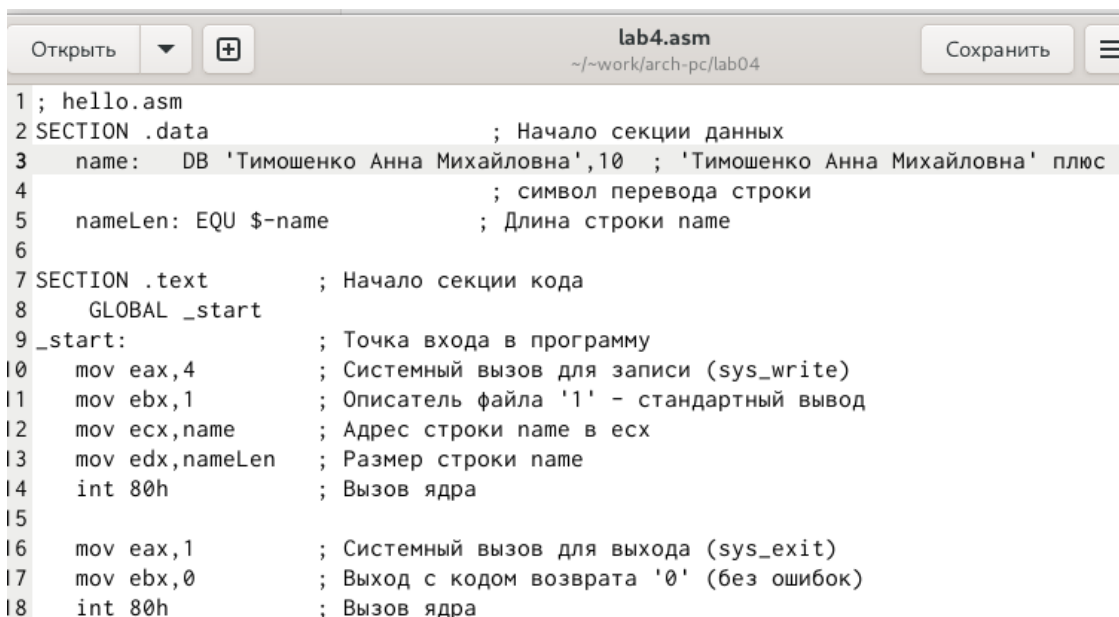
```

bash: cd: ~/work/arch-pc/lab04: Нет такого файла или каталога
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
amtimoshenko@dk2n21 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o

```

РИС.3 Создание копии файла hello.asm с именем lab4.asm

### 6.2 изменить скопированный файл, чтобы выводилась строка с именем и фамилией (см. рис.4)



```

1 ; hello.asm
2 SECTION .data                ; Начало секции данных
3     name: DB 'Тимошенко Анна Михайловна',10 ; 'Тимошенко Анна Михайловна' плюс
4                                     ; символ перевода строки
5     nameLen: EQU $-name        ; Длина строки name
6
7 SECTION .text                ; Начало секции кода
8     GLOBAL _start
9 _start:                      ; Точка входа в программу
10    mov eax,4                 ; Системный вызов для записи (sys_write)
11    mov ebx,1                 ; Описатель файла '1' - стандартный вывод
12    mov ecx,name              ; Адрес строки name в ecx
13    mov edx,nameLen           ; Размер строки name
14    int 80h                   ; Вызов ядра
15
16    mov eax,1                 ; Системный вызов для выхода (sys_exit)
17    mov ebx,0                 ; Выход с кодом возврата '0' (без ошибок)
18    int 80h                   ; Вызов ядра

```

РИС.4 Изменение файла lab4.asm

6.3 оттранслировать полученный текст программы lab4.asm в объектный файл (см. рис.5)

6.4 скопировать файлы hello.asm и lab4.asm в локальный репозиторий (см. рис.5)

```
amtimoshenko@dk2n21 ~/~work/arch-pc/lab04 $ nasm -f elf lab4.asm
amtimoshenko@dk2n21 ~/~work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o name
amtimoshenko@dk2n21 ~/~work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main_name
amtimoshenko@dk2n21 ~/~work/arch-pc/lab04 $ ./name
Тимошенко Анна Михайловна
```

РИС.5 скопировать файлы в локальный репозиторий

## Выводы

Я ознакомилась с созданием и процессом обработки программ на языке ассемблера NASM