

title: "Лабораторная работа №4" author: "ЧжуЖуйи" date: "26 октября 2025 г." — # Цель работы Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## Порядок выполнения лабораторной работы

### 1)Программа Hello world!

-Создадим каталог для работы с программами на языке ассемблера NASM:

```
mkdir -p ~/work/arch-pc/lab04
```

-Перейдём в созданный каталог и создадим текстовый файл с именем hello.asm :

```
cd ~/work/arch-pc/lab04
touch hello.asm
```

-откроем этот файл с помощью любого текстового редактора, например, gedit:

```
gedit hello.asm
```

-и введём в него следующий текст:

```
; 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 есх
    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
```

## 2)Транслятор NASM

-NASM превращает текст программы в объектный код и с помощью команды ls проверим, что объектный файл был создан.:

```
nasm -f elf hello.asm
ls
```

## 3)Расширенный синтаксис командной строки NASM

-Выполним следующую команду и с помощью команды ls проверим, что файлы были созданы.:

```
nasm -o obj.o -f elf -g -l list.lst hello.asm
ls -l
```

## 4)Компоновщик LD

-Как видно из схемы на 3), чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику и с помощью команды ls проверим, что исполняемый файл hello был создан:

```
ld -m elf_i386 hello.o -o hello
ls -l
```

-Выполним следующую команду:

```
ld -m elf_i386 obj.o -o main
```

-Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге,можно, набрав в командной строке:

```
./hello
```

## Задание для самостоятельной работы

**1)В каталоге ~/work/arch-pc/lab04 с помощью команды cp создайте копию файла hello.asm с именем lab4.asm**

```
cd ~/work/arch-pc/lab04
cp hello.asm lab4.asm
```

**2)С помощью любого текстового редактора внесите изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.**

```
gedit lab4.asm
```

```
; hello.asm
SECTION .data                ; Начало секции данных
    hello:    DB 'ЧжуЖуйи!',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 есх
    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
```

**3)Оттранслируйте полученный текст программы lab4.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.**

```
nasm -f elf lab4.asm
ld -m elf_i386 lab4.o -o lab4
./lab4
```

**4)Скопируйте файлы hello.asm и lab4.asm в Ваш локальный репозиторий в каталог ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/. Загрузите файлы на Github.**

```
cp hello.asm lab4.asm ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/1
cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/
```

```
git add .  
git commit -am 'feat(main): add lab04'  
git push
```

## **Выводы**

В ходе выполнения лабораторной работы №4 я освоила основы использования ассемблера NASM, поняла процесс компиляции и компоновки программ на ассемблере, а также научилась изменять и настраивать программы на ассемблере.