

Шаблон отчёта по лабораторной работе

4

Бембо Жозе Лумингу , НКАбд-01-23

Содержание

• Цель работы	5
• Выполнение лабораторной работы	6
• Создание программы Hello world!	7
• Работа с транслятором NASM и работа с расширенным синтаксисом команд-ной строки NASM	9
• Работа с компоновщиком LD	10
• Запуск исполняемого файла	11
• Выполнение заданий для самостоятельной работы.	12
• Выводы	15
Список литературы	16

Список иллюстраций

3.1	перемещение между директорами	7
3.2	создание пустого файла	7
3.3	редактора gedig	7
3.4	редактора gedig	8
4.1	компиляция текста программы	9
5.1	передача объектного файла на обработку компоновщику	10
6.1	запуск исполняемого файла.....	11
7.1	создание копии файла.....	12
7.2	изменения программы	13
7.3	компиляция трека программы	13
7.4	передача объектного файла на обработку компоновщику.....	14
7.5	передача объектного файла на обработку компоновщику.....	14

Список таблиц

1 Цель работы

- Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Выполнение лабораторной работы

3 Создание программы Hello world!

- С помощью утилиты `cd` перемещаюсь в каталог, в котором буду работать (рис. 3.1).

```
[zlbembo@fedora ~]$ mkdir -p ~/work/arch-pc/lab04  
[zlbembo@fedora ~]$ cd ~/work/arch-pc/lab04  
[zlbembo@fedora lab04]$
```

Рис. 3.1: перемещение между директориями

- Создаю в текущем каталоге пустой текстовый файл `hello.asm` с помощью утилиты `touch` (рис. 3.2).

```
[zlbembo@fedora lab04]$ touch hello.asm  
[zlbembo@fedora lab04]$
```

Рис. 3.2: создание пустого файла

- Открываю созданный файл в текстовом редакторе `gedit` `hello.asm` (рис. 3.3).

```
[zlbembo@fedora lab04]$ touch hello.asm  
[zlbembo@fedora lab04]$ gedit hello.asm
```

Рис. 3.3: редактора gedit

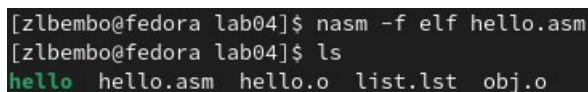
- Заполняю файл, вставляя в него программу для вывода "Hello word!" (рис. 3.4).

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

Рис. 3.4: редактора gedig

4 Работа с транслятором NASM и работа с расширенным синтаксисом командной строки NASM

1 Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`, ключ `-f` указывает транслятору `nasm`, что требуется создать бинарный файл в формате ELF (рис. [4.1]). Далее проверяю правильность выполнения команды с помощью утилиты `ls`: действительно, создан файл “hello.o”. Ввожу команду, которая скомпилирует файл `hello.asm` в файл `obj.o`, при этом в файл будут включены символы для отладки (ключ `-g`), также с помощью ключа `-l` будет создан файл листинга `list.lst` (рис. [4.1]). Далее проверяю с помощью утилиты `ls` правильность выполнения команды.(рис. 4.1).



```
[zlbembo@fedora lab04]$ nasm -f elf hello.asm
[zlbembo@fedora lab04]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.1: компиляция текста программы

5 Работа с компоновщиком LD


2 Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello (рис. [5.1]). Ключ -o задает имя создаваемого исполняемого файла. Далее проверяю с помощью утилиты ls правильность выполнения команды. Выполняю следующую команду (рис. [5.1]). Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o. (рис. 5.1).

```
[zlbembo@fedora lab04]$ ld -m elf_i386 obj.o -o main
[zlbembo@fedora lab04]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 5.1: передача объектного файла на обработку компоновщику

6 Запуск исполняемого файла

3 Запускаю на выполнение созданный исполняемый файл hello (рис. 6.1).

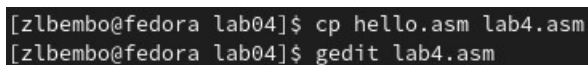
A terminal window with a dark background. The prompt is [zlbembo@fedora lab04]\$ and the command entered is ./hello. The output is Hello world!

```
[zlbembo@fedora lab04]$ ./hello
Hello world!
```

Рис. 6.1: запуск исполняемого файла

7 Выполнение заданий для самостоятельной работы.

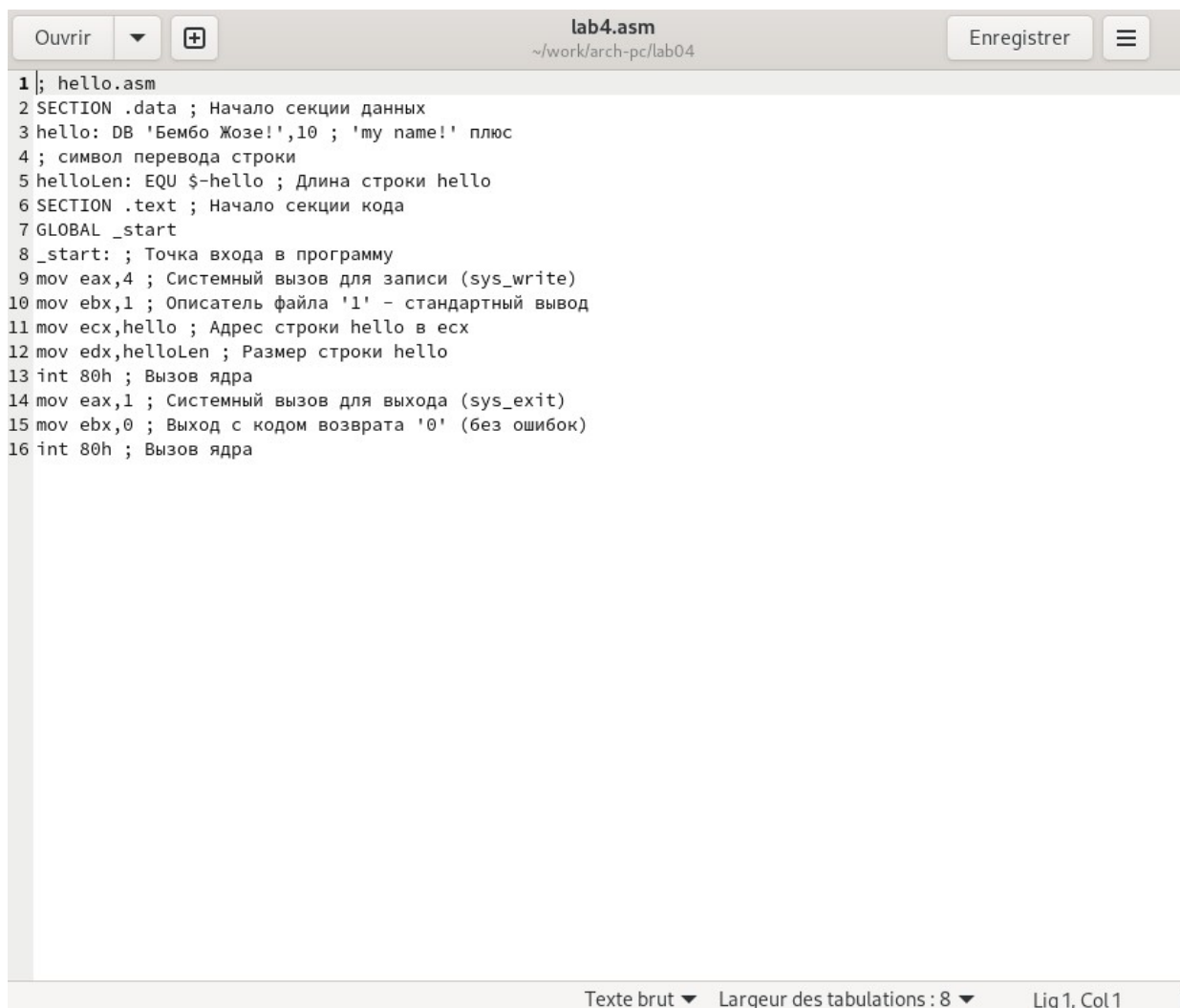
- 4 С помощью утилиты `cp` создаю в текущем каталоге копию файла `hello.asm` с именем `lab4.asm` (рис. 7.1).



```
[zlbembo@fedora lab04]$ cp hello.asm lab4.asm  
[zlbembo@fedora lab04]$ gedit lab4.asm
```

Рис. 7.1: создание копии файла

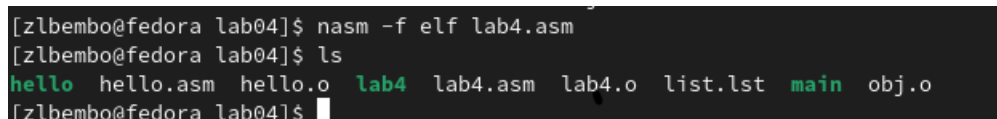
- 5 С помощью текстового редактора `gedit` открываю файл `lab4.asm` и вношу изменения в программу так, чтобы она выводила мои имя и фамилию.(рис. 7.2).



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

Рис. 7.2: изменения программы

6 Компилирую текст программы в объектный файл (рис. [7.3]). Проверяю с помощью утилиты `ls`, что файл `lab4.o` создан. (рис. 7.3).



```
[zlbembo@fedora lab04]$ nasm -f elf lab4.asm
[zlbembo@fedora lab04]$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
[zlbembo@fedora lab04]$
```

Рис. 7.3: компиляция трека программы

7 Передаю объектный файл `lab4.o` на обработку компоновщику `LD`, чтобы получить исполняемый файл `lab4` (рис. 7.4).

```
[zlbembo@fedora lab04]$ ld -m elf_i386 lab4.o -o lab4
[zlbembo@fedora lab04]$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
```

Рис. 7.4: передача объектного файла на обработку компоновщику

- 8** Запускаю исполняемый файл lab4, на экран действительно выводятся мои имя и фамилия (рис. 7.5).

```
[zlbembo@fedora lab04]$ ./lab4
Бембо Жозе!
```

Рис. 7.5: передача объектного файла на обработку компоновщику

8 Выводы

-При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Список литературы