

Отчёт по лабораторной работе№4

Создание и процесс обработки программ на языке ассемблера NASM

Зеленко Ирина Юрьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	12

Список иллюстраций

3.1	Создание каталога	7
3.2	Переход в каталог	7
3.3	Создание текстового файла	7
3.4	Открытие файла	7
3.5	Ввод текста	8
3.6	Компиляция текста	8
3.7	проверка, что объектный файл был создан	8
3.8	Создание файлов	9
3.9	Передача файла на компоновку	9
3.10	Запуск на выполнение созданный исполняемый файл	9
3.11	Создание копии файла с именем lab4.asm	9
3.12	открываем текстовый редактор	9
3.13	Внесение изменения в текст программы	10
3.14	Оттранслирование, компоновка, запуск	10
3.15	Копирование файлов в локальный репозиторий	11

Список таблиц

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Выполнение лабораторной работы

1. Создайте каталог для работы с программами на языке ассемблера NASM:

```
iyzelenko@dk2n26 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Создание каталога

2. Перейдём в созданный каталог:

```
iyzelenko@dk2n26 ~ $ cd ~/work/arch-pc/lab04
```

Рис. 3.2: Переход в каталог

3. Создадим текстовый файл с именем hello.asm:

```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ touch hello.asm
```

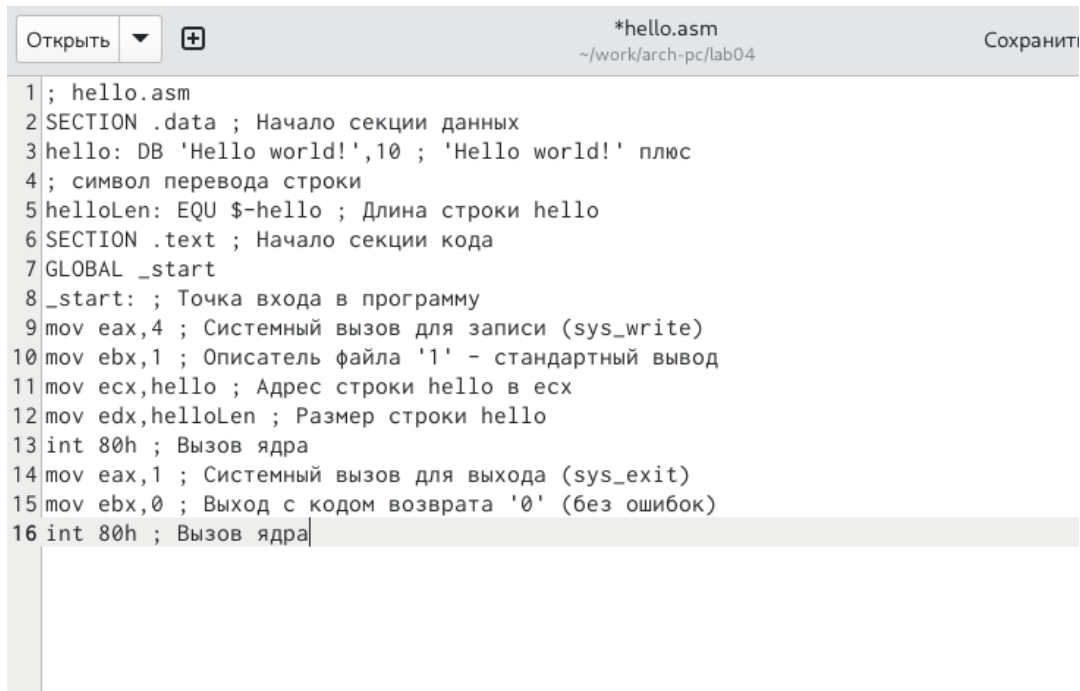
Рис. 3.3: Создание текстового файла

4. Откроем этот файл с помощью текстового редактора

```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.4: Открытие файла

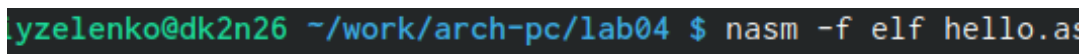
5. Введём в него текст:



```
Открыть ▼ + *hello.asm ~/work/arch-pc/lab04 Сохранить
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 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. 3.5: Ввод текста

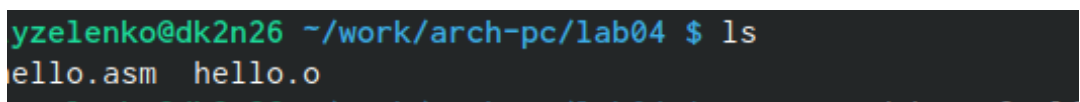
6. Скомпилируем данный текст



```
yzelenko@dk2n26 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
```

Рис. 3.6: Компиляция текста

7. Проверим, что объектный файл был создан:



```
yzelenko@dk2n26 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Рис. 3.7: проверка, что объектный файл был создан

8. Скомпилируем исходный файл hello.asm в obj.o и создадим файл листинга list.lst и проверяем, что файлы были созданы


```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.8: Создание файлов

9. Передадим объектный файл на обработку компоновщику.

```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
```

Рис. 3.9: Передача файла на компоновку

10. Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге.

```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

Рис. 3.10: Запуск на выполнение созданный исполняемый файл

11. Создадим копию файла hello.asm с именем lab4.asm

```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
```

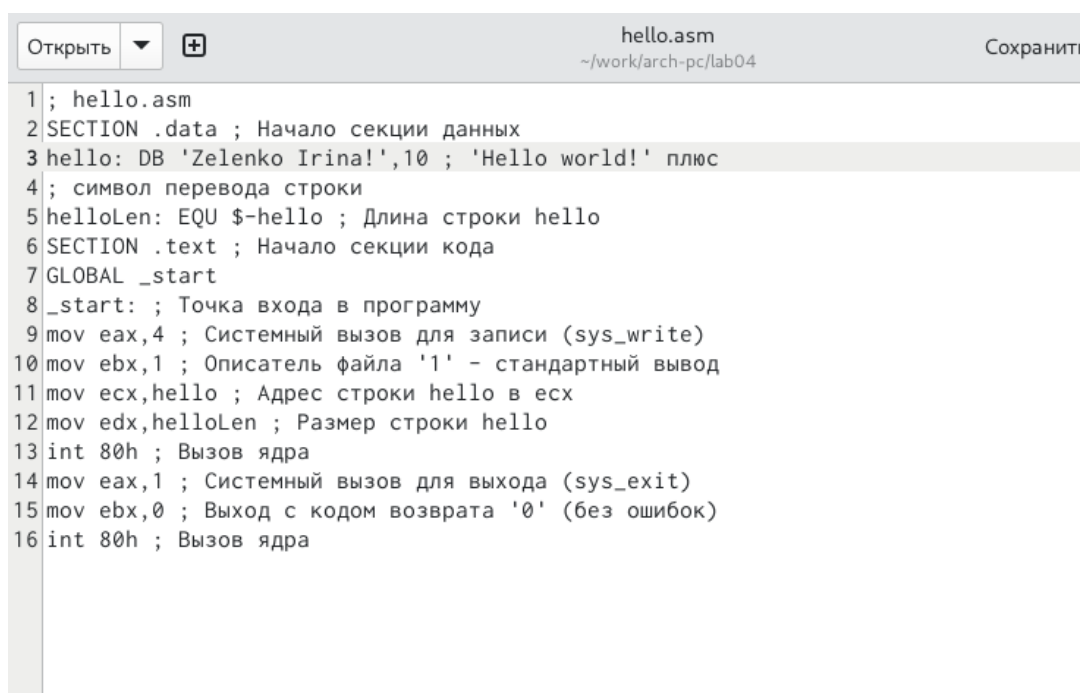
Рис. 3.11: Создание копии файла с именем lab4.asm

12. откроем текстовый редактор

```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.12: открываем текстовый редактор

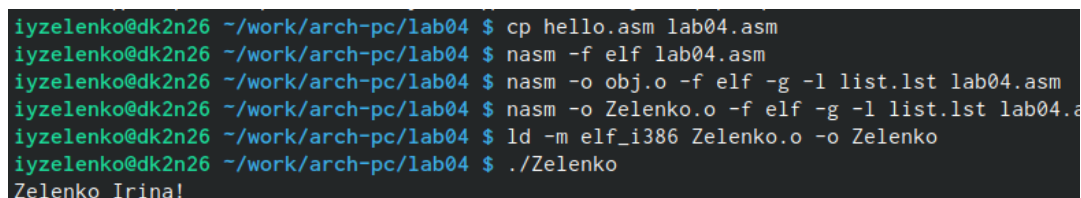
13. Внесём изменения в текст программы в файле lab4.asm



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Zelenko Irina!',10 ; 'Hello world!' 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. 3.13: Внесение изменения в текст программы

14. Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.



```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ cp hello.asm lab04.asm
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ nasm -f elf lab04.asm
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab04.asm
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ nasm -o Zelenko.o -f elf -g -l list.lst lab04.o
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ ld -m elf_i386 Zelenko.o -o Zelenko
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ ./Zelenko
Zelenko Irina!
```

Рис. 3.14: Оттранслирование, компоновка, запуск

15. Скопировала файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ с помощью утилиты cp и проверил наличие файлов с помощью утилиты ls

```
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ cp hello.asm ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/report
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ cp lab04.asm ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/report
iyzelenko@dk2n26 ~/work/arch-pc/lab04 $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/report
iyzelenko@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/report $ ls
bib    hello.asm  image      lab4.asm  Makefile  pandoc
hello  hello.o    lab04.asm  list.lst  obj.o     report.md
```

Рис. 3.15: Копирование файлов в локальный репозиторий

4 Выводы

В ходе выполнения работы, я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.