

Отчёта по лабораторной работе №4

Создание и процесс обработки программ на языке ассемблера NASM

Кочкина Кристина Андреевна НММбд-02-23

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Программа Hello world!	6
3.2	Транслятор NASM	7
3.3	Расширенный синтаксис командной строки NASM	8
3.4	Компоновщик LD	8
3.5	Запуск исполняемого файла	8
3.6	Задание для самостоятельной работы	9
4	Выводы	11

Список иллюстраций

3.1	Создаем каталоги с помощью команды <code>mkdir</code>	6
3.2	Переходим в каталог с помощью команды <code>cd</code>	6
3.3	Создаем текстовый файл <code>hello.asm</code> и проверяем, создался ли он .	6
3.4	Открываем файл	7
3.5	Заполняем файл по примеру	7
3.6	Используем команду <code>nasm</code> и проверяем его работу	7
3.7	Преобразуем файл <code>hello.asm</code> в <code>obj.o</code> и проверяем файлы	8
3.8	Используем команду <code>ld</code> и проверяем создался ли исполняемый файл	8
3.9	Используем команду <code>ld</code> , создавая файл <code>main</code>	8
3.10	Используем команду <code>./hello</code>	8
3.11	Используем команду <code>cp</code>	9
3.12	Редактируем файл для своего имени и фамилии	9
3.13	Запускаем программу и проверяем вывод	10
3.14	Копируем файлы в каталог с 4 Лабораторной работой	10

1 Цель работы

Освоить процедуры компиляции, познакомиться с языком ассемблера NASM.

2 Задание

Написать 2 программы(Hello world и lab4(Имя Фамилия))

3 Выполнение лабораторной работы

3.1 Программа Hello world!

Создаем каталог для работы с программами на языке ассемблера NASM (рис. 3.1).

```
kakochkina@dk3n64 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Создаем каталоги с помощью команды mkdir

Переходим в созданный каталог (рис. 3.2).

```
kakochkina@dk3n64 ~ $ cd ~/work/arch-pc/lab04  
kakochkina@dk3n64 ~/work/arch-pc/lab04 $
```

Рис. 3.2: Переходим в каталог с помощью команды cd

Создаем текстовый файл (рис. 3.3).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ touch hello.asm  
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ls  
hello.asm
```

Рис. 3.3: Создаем текстовый файл hello.asm и проверяем, создался ли он

Открываем данный файл в текстовом редакторе (рис. 3.4).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.4: Открываем файл

```
1 ; hello.asm
2 SECTION .data
3     hello: DB 'Hello world!',10
4     helloLen: EQU $-hello
5 SECTION .text
6     GLOBAL _start
7 _start:
8     mov eax,4
9     mov ebx,1
10    mov ecx,hello
11    mov edx,helloLen
12    int 80h
13
14    mov eax,1
15    mov ebx,0
16    int 80h
```

Рис. 3.5: Заполняем файл по примеру

3.2 Транслятор NASM

Преобразуем текст программы в объектный код (рис. 3.6).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Рис. 3.6: Используем команду nasm и проверяем его работу

3.3 Расширенный синтаксис командной строки NASM

Компилируем исходный файл (рис. 3.7).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.7: Преобразуем файл hello.asm в obj.o и проверяем файлы

3.4 Компоновщик LD

Передаем объектный файл на обработку компоновщику (рис. 3.8).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.8: Используем команду ld и проверяем создался ли исполняемый файл

Передаем объектный файл на обработку компоновщику (рис. 3.9).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
kakochkina@dk3n64 ~/work/arch-pc/lab04 $
```

Рис. 3.9: Используем команду ld, создавая файл main

3.5 Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл (рис. 3.10).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ./hello
Hello world!
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ gedit lab4.asm
```

Рис. 3.10: Используем команду ./hello

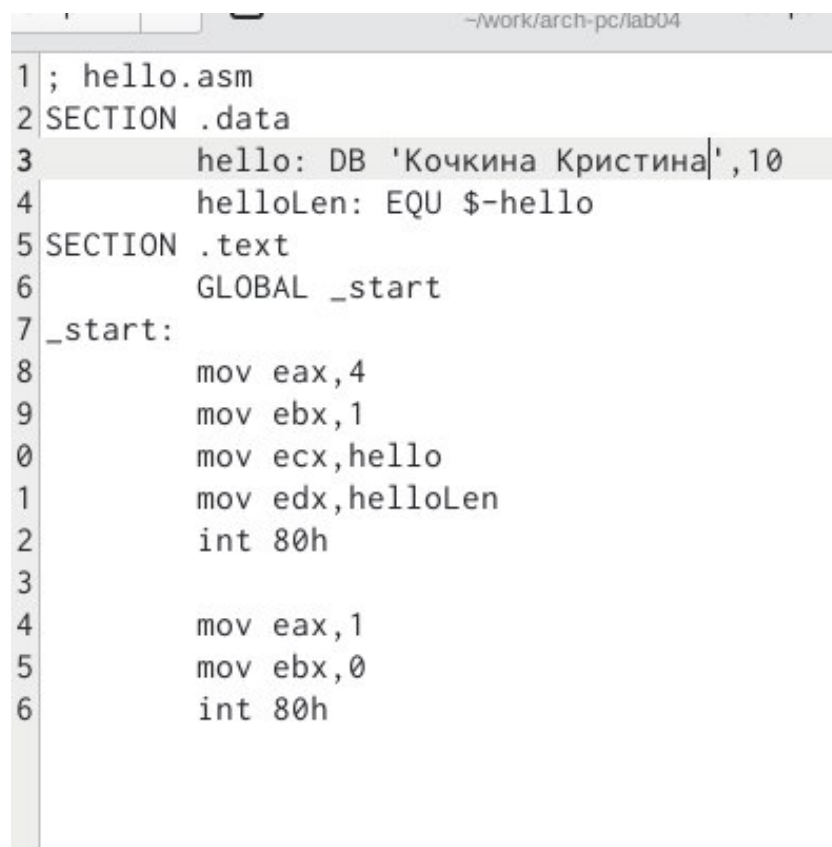
3.6 Задание для самостоятельной работы

Создаем копию файла hello.asm (рис. 3.11).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ gedit lab4.asm
```

Рис. 3.11: Используем команду cp

Открываем файл и редактируем его (рис. 3.12).



```
1 ; hello.asm
2 SECTION .data
3     hello: DB 'Кочкина Кристина',10
4     helloLen: EQU $-hello
5 SECTION .text
6     GLOBAL _start
7 _start:
8     mov eax,4
9     mov ebx,1
10    mov ecx,hello
11    mov edx,helloLen
12    int 80h
13
14    mov eax,1
15    mov ebx,0
16    int 80h
```

Рис. 3.12: Редактируем файл для своего имени и фамилии

Прописываем те же команды, что и с первой программой.

Запускаем команду (рис. 3.13).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ ./hello
Кочкина Кристина
```

Рис. 3.13: Запускаем программу и проверяем вывод

Копируем файлы в локальный репозиторий (рис. 3.14).

```
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ cp hello.asm ~/work/study/2023-2024/"Архитектура компьюте
pa"/arch-pc/labs/lab04/
kakochkina@dk3n64 ~/work/arch-pc/lab04 $ cp lab4.asm ~/work/study/2023-2024/"Архитектура компьюте
a"/arch-pc/labs/lab04/
kakochkina@dk3n64 ~/work/arch-pc/lab04 $
```

Рис. 3.14: Копируем файлы в каталог с 4 Лабораторной работой

Переходим в каталог лабораторных работ и загружаем файлы на Github.

4 Выводы

Мы познакомились с языком ассемблера NASM и создали две работающих программы.