Отчёта по лабораторной работе №4

Дисциплина: архитектура компьютера

Глущенко Евгений НКАбд-02-23

1 Цель работы

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

2 Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла
6. Выполнение заданий для самостоятельной работы.

3 Выполнение лабораторной работы

3.1 Создание программы Hello world!

С помощью команды cd перемещаюсь в каталог этой лабораторной раьоты (рис. 1).

```
[eigluthenko@38 ~]$ cd work/study/2023-2024/архитектура\ компьютера/stady_2023-2024_arch_pc/labs/lab04
[eigluthenko@38 lab04]$
```

Рис. 1: Перемещение между директориями

Создаю в текущем каталоге пустой текстовый файл hello.asm с помощью (рис. 2).

```
[eigluthenko@37 lab04]$ touch hello.asm
[eigluthenko@37 lab04]$
```

Рис. 2: Создание пустого файла

Открываю этот файл в текстовом редакторе gedit (рис. 3).

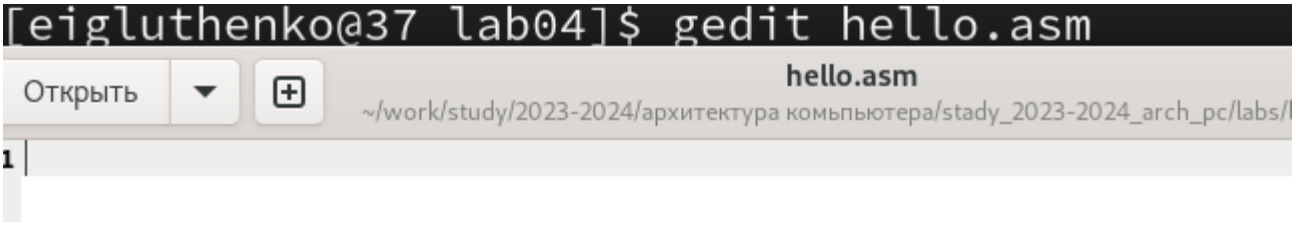

Рис. 3: Открытие файла в текстовом редакторе

Заполняю файл, вставляя в него программу для вывода “Hello word!” (рис. 4).

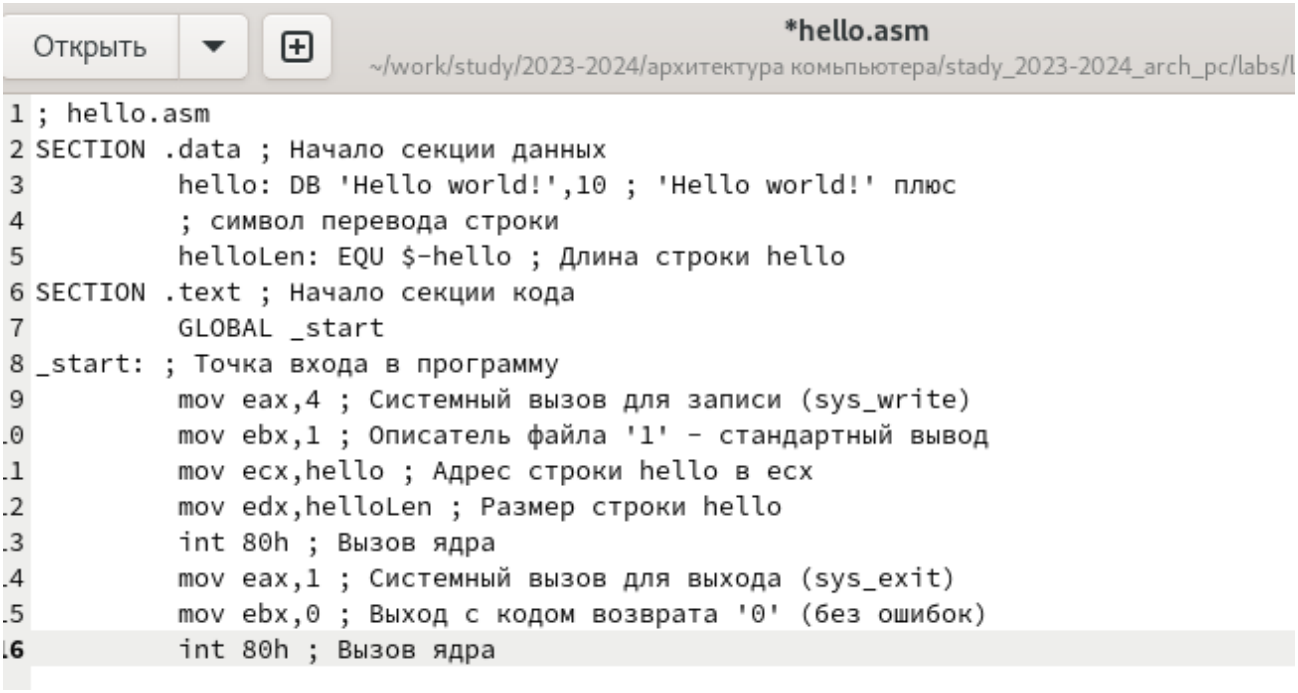

Рис. 4: Заполнение файла

3.2 Работа с транслятором NASM

Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду, указанную на рисунке. (рис. 5).

Далее проверяю правильность выполнения команды с помощью утилиты ls: действительно, создан файл “hello.o”.

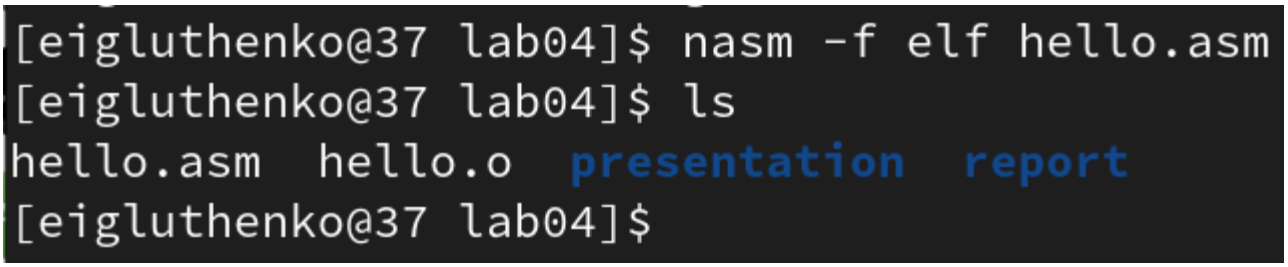

Рис. 5: Компиляция текста программы

3.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o, при этом в файл будут включены символы для отладки (рис. 6).

Далее проверяю с помощью команды ls правильность выполнения команды.

```
[eigluthenko@37 lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[eigluthenko@37 lab04]$ ls
hello.asm  hello.o  list.lst  obj.o  presentation  report
[eigluthenko@37 lab04]$
```

Рис. 6: Компиляция текста программы

3.4 Работа с компоновщиком LD

Передаю файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello (рис. 7).

Далее проверяю с помощью утилиты ls правильность выполнения команды.

```
[eigluthenko@37 lab04]$ ld -m elf_i386 hello.o -o hello
[eigluthenko@37 lab04]$ ls
hello  hello.asm  hello.o  list.lst  obj.o  presentation  report
[eigluthenko@37 lab04]$
```

Рис. 7: Передача объектного файла на обработку компоновщику

Выполняю следующую команду (рис. 8). 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

```
[eigluthenko@37 lab04]$ ld -m elf_i386 obj.o -o main
[eigluthenko@37 lab04]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o  presentation  report
[eigluthenko@37 lab04]$
```

Рис. 8: Передача объектного файла на обработку компоновщику

3.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello (рис. 9).

```
[eigluthenko@37 lab04]$ ./hello
Hello world!
[eigluthenko@37 lab04]$
```

Рис. 9: Запуск исполняемого файла

3.6 Выполнение заданий для самостоятельной работы.

С помощью команды cp создаю копию файла hello.asm с именем lab4.asm (рис. 10).

```
[eigluthenko@37 lab04]$ cp hello.asm lab4.asm
[eigluthenko@37 lab04]$
```

Рис. 10: Создание копии файла

С помощью gedit открываю файл и вношу изменения в программу так, чтобы она выводила мои имя и фамилию. (рис. 11).

```
; lab4.asm
SECTION .data ; Начало секции данных
        hello: DB 'Evgenii Glushchenko',10

        helloLen: EQU $-lab4 ; Длина строки lab4

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,lab4 ; Адрес строки lab4 в ecx
        mov edx,lab4Len ; Размер строки lab4
        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. 11: Изменение программы

Компилирую текст программы в объектный файл (рис. 12). Проверяю с помощью ls, что файл lab4.o создан.

```
[eigluthenko@37 lab04]$ nasm -f elf  lab4.asm
[eigluthenko@37 lab04]$ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o  presentation  report
```

Рис. 12: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD (рис. 13).

```
[eigluthenko@37 lab04]$ ld -m elf_i386 lab4.o -o lab4
[eigluthenko@37 lab04]$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o  presentation  report
[eigluthenko@37 lab04]$
```

Рис. 13: Передача объектного файла на обработку компоновщику

Запускаю файл lab4, и на экран выводятся мои имя и фамилия (рис. 14).

```
[eigluthenko@37 lab04]$ ./lab4
Evgenii Glushchenko
```

Рис. 14: Запуск исполняемого файла

С помощью команд git add . и git commit добавляю файлы на GitHub, комментируя действие как добавление файлов для лабораторной работы №4 (рис. 17).

```
[eigluthenko@37 stady_2023-2024_arch_pc]$ git add .
[eigluthenko@37 stady_2023-2024_arch_pc]$ git commit -m "Add fales for lab0
```

Рис. 15: Добавление файлов на GitHub

Отправляю файлы на сервер (рис. 18).

ы

```
[eigluthenko@37 stady_2023-2024_arch_pc]$ git push
Перечисление объектов: 23, готово.
Подсчет объектов: 100% (21/21), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (17/17), готово.
Запись объектов: 100% (17/17), 267.74 КиБ | 424.00 КиБ/с, готово.
Всего 17 (изменений 7), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (7/7), completed with 3 local objects.
To github.com:eigluthenko/stady_2023-2024_arch_pc.git
   1fe1d6f..1cfc3b1  master -> master
```

Рис. 16: Отправка файлов

4 Выводы

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