

Лабораторная работа №5

Татур Стефан

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение самостоятельной работы	9
4	Выводы	12

Список иллюстраций

2.1	Создание каталога	6
2.2	Создание файла	6
2.3	Запуск текстового редактора	6
2.4	Работа в текстовом редакторе	7
2.5	Компиляция файла	7
2.6	Компоновка файла	7
2.7	Выполнение команды	8
2.8	Работа программы	8
3.1	Создание копии	9
3.2	Работа программы	10
3.3	Компоновка файла	10
3.4	Запуск программы	10
3.5	Копирование файлов	11
3.6	Созданные файлы	11

Список таблиц

1 Цель работы

Ознакомиться с низкоуровневым языком программирования `Assembler`. Освоение процедуры компиляции и сборки программ.

2 Выполнение лабораторной работы

1. Создал каталог для работы с программами на языке ассемблера NASM. И перешел в созданный каталог. (рис. ??)

```
satatur@dk4n65 ~ $ mkdir ~/work/study/2022-2023/'Архитектура компьютера'/arch-pc/lab05
satatur@dk4n65 ~ $ cd ~/work/study/2022-2023/'Архитектура компьютера'/arch-pc/lab05
```

Рис. 2.1: Создание каталога

2. Создал текстовый файл с именем hello.asm

```
satatur@dk4n65 ~ $ cd ~/work/study/2022-2023/'Архитектура компьютера'/arch-pc/lab05
satatur@dk4n65 ~/work/study/2022-2023/'Архитектура компьютера'/arch-pc/lab05 $ touch hello.asm
```

Рис. 2.2: Создание файла

3. Открыл этот файл с помощью gedit. Далее ввел в него нужный текст.

```
satatur@dk4n65 ~/work/study/2022-2023/'Архитектура компьютера'/arch-pc/lab05 $ gedit hello.asm
```

Рис. 2.3: Запуск текстового редактора

```

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. 2.4: Работа в текстовом редакторе

4. Для компиляции приведённого выше текста программы «Hello World» воспользовался командой представленной ниже.

```

satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf hello.asm
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello.asm hello.o

```

5. Скомпилировал исходный файл hello.asm в obj.o с помощью следующей команды.

```

satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l
list.lst hello.asm

```

Рис. 2.5: Компиляция файла

6. Далее передал объектный файл на обработку компоновщику.

```

satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l
list.lst hello.asm
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello.asm hello.o list.lst obj.o

```

Рис. 2.6: Компоновка файла

7. Выполнил следующую команду и создал “main”.

```
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
```

Рис. 2.7: Выполнение команды

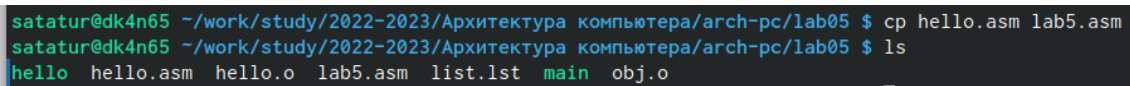
8. Запустил на выполнение созданный исполняемый файл, находящийся в текущем каталоге, набрав в командной строке команду “./hello”.

```
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./hello  
Hello world!
```

Рис. 2.8: Работа программы

3 Выполнение самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создал копию файла hello.asm с именем lab5.asm



```
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp hello.asm lab5.asm
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  lab5.asm  list.lst  main  obj.o
```

Рис. 3.1: Создание копии

2. Внес изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводиться строка с моей фамилией и именем.

```

1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Татур Стефан',10 ; 'Татур Стефан' 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. 3.2: Работа программы

- Оттранслировал полученный текст программы lab5.asm в объектный файл. Выполнил компоновку объектного файла и запустил получившийся исполняемый файл.

```

satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5.asm
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 lab5.o -o hello

```

Рис. 3.3: Компоновка файла

```

satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./hello
Татур Стефан
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $

```

Рис. 3.4: Запуск программы

- Скопировал файлы hello.asm и lab5.asm в свой локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/.

```

satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp hello.asm '/afs/.dk.sci.
pfu.edu.ru/home/s/a/satatur/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05'
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp lab5.asm '/afs/.dk.sci.p
fu.edu.ru/home/s/a/satatur/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05'
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  lab5.asm  lab5.o  list.lst  main  obj.o
satatur@dk4n65 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $

```

Рис. 3.5: Копирование файлов

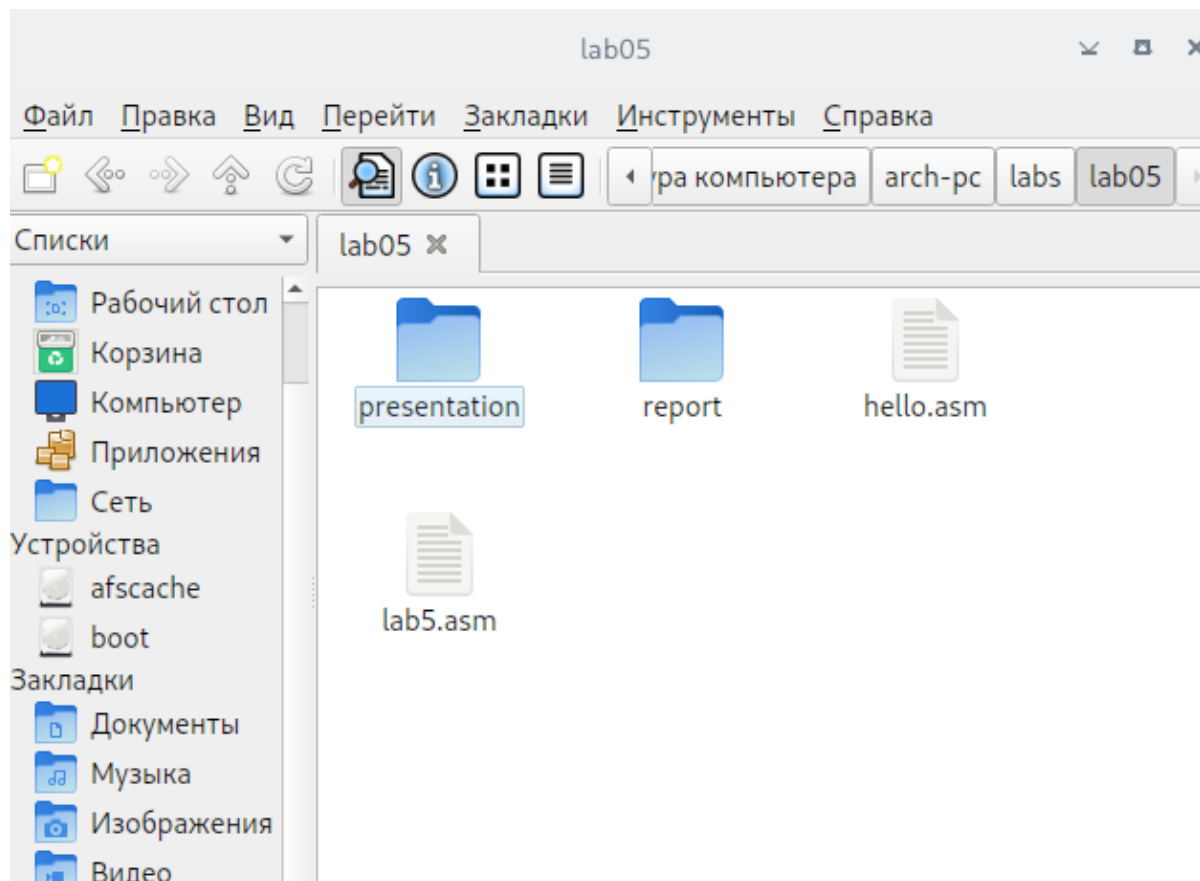


Рис. 3.6: Созданные файлы

5. Загрузил файлы на Github.

4 Выводы

Ознакомился с низкоуровневым языком программирования Assembler. Освоил процедуры компиляции и сборки программ. Разместил файлы на github.