

Лабораторная работа номер 5

Дисциплина: Архитектура компьютеров

Бабенко Роман Игоревич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задания для самостоятельной работы	8
4	Выводы	10

Список иллюстраций

2.1	Создание текстового файла	6
2.2	Копируем текст	6
2.3	Компилируем код	7
2.4	компилируем файл в 'obj.o'	7
2.5	Обрабатываем компоновщиком	7
2.6	Выполнение предложенной команды	7
2.7	Запускаем файл	7
3.1	Копируем файлы	8
3.2	Выполняем компоновку и запускаем исполняемый файл	8
3.3	Копируем файлы	8
3.4	Проверяем их наличие	8
3.5	Загружаем на github	9

Список таблиц

1 Цель работы

Научиться компилировать и собирать программы, написанные на ассемблере NASM.

2 Выполнение лабораторной работы

Создаём каталог, переходим в него и создаём текстовый файл hello.asm. (рис. 2.1)

```
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ mkdir -p ~/work/arch-pc/lab05
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ cd ~/work/arch-pc/lab05
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ touch hello.asm
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ gedit hello.asm
ribabenko@dk5n55 ~/work/arch-pc/lab05 $
```

Рис. 2.1: Создание текстового файла

Открываем его и вводим следующий текст. (рис. 2.2)

```
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. 2.2: Копируем текст

Компилируем код и проверяем что создан файл hello.o. (рис. 2.3)

```

ribabenko@dk5n55 ~/work/arch-pc/lab05 $ nasm -f elf hello.asm
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ls
hello.asm hello.o
ribabenko@dk5n55 ~/work/arch-pc/lab05 $

```

Рис. 2.3: Компилируем код

Компилируем исходный файл в obj.o и проверяем его наличие. (рис. 2.4)

```

ribabenko@dk5n55 ~/work/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ls
hello.asm hello.o list.lst obj.o
ribabenko@dk5n55 ~/work/arch-pc/lab05 $

```

Рис. 2.4: компилируем файл в 'obj.o'

Передаём объектный файл на обработку компоновщику и проверяем с помощью ls его наличие. (рис. 2.5)

```

ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 hello.o -o hello
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ls
hello hello.asm hello.o list.lst obj.o
ribabenko@dk5n55 ~/work/arch-pc/lab05 $

```

Рис. 2.5: Обрабатываем компоновщиком

Выполняем следующую команду. Файл будет иметь имя 'main', он был создан из файла 'obj.o'. (рис. 2.6)

```

ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main

```

Рис. 2.6: Выполнение предложенной команды

Запускаем на выполнение исполняемый файл. (рис. 2.7)

```

ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ./hello
Hello world!
ribabenko@dk5n55 ~/work/arch-pc/lab05 $

```

Рис. 2.7: Запускаем файл

3 Задания для самостоятельной работы

Копируем hello.asm и lab05.asm в 'lab05'. (рис. 3.1)

```
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ cp hello.asm lab05.asm
ribabenko@dk5n55 ~/work/arch-pc/lab05 $
```

Рис. 3.1: Копируем файлы

Изменяем файл lab05.asm. Оттранслируем текст в объектный файл. Выполняем компоновку и запускаем исполняемый файл. (рис. 3.2)

```
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ nasm -o lab05.o -f elf -g -l list.lst lab05.asm
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 lab05.o -o lab05
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ ./lab05
Бабенко Роман
```

Рис. 3.2: Выполняем компоновку и запускаем исполняемый файл

Копируем файлы в локальный репозиторий. (рис. 3.3) (рис. 3.4)

```
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ cp hello.asm ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/labs/lab05/hello.asm
ribabenko@dk5n55 ~/work/arch-pc/lab05 $ cp lab05.asm ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/labs/lab05/lab05.asm
```

Рис. 3.3: Копируем файлы

```
ribabenko@dk5n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ls
hello.asm lab05.asm presentation report
```

Рис. 3.4: Проверяем их наличие

Загружаем файлы на github. (рис. 3.5)


```

ribabenko@dk5n55 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc
ribabenko@dk5n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ git add .
ribabenko@dk5n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ git commit -am 'feat(main): add files lab-5'
[master 5df556a] feat(main): add files lab-5
 2 files changed, 32 insertions(+)
 create mode 100644 labs/lab05/hello.asm
 create mode 100644 labs/lab05/lab5.asm
ribabenko@dk5n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 954 байта | 954.00 КиБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:skrabik/study_2022-2023_arh-pc-.git
 8249618..5df556a  master -> master
ribabenko@dk5n55 ~/work/study/2022-2023/Архитектура компьютера/arch-pc $ █

```

Рис. 3.5: Загружаем на github

4 Выводы

Я научился компилировать и собирать команды на NASM.