

Отчет по лабораторной работе №4

Архитектура вычислительных систем

Белослюдов Иван Евгеньевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выполнение лабораторной работы	8
5	Выводы	10

Список иллюстраций

3.1	Переходим в каталог	6
3.2	Вставляем текст	6
3.3	Код и проверка	7
3.4	Проверка	7
3.5	Проверка	7
3.6	Запуск	7
4.1	Копируем	8
4.2	Оттранслируем	8
4.3	Github	9

1 Цель работы

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

2 Задание

1. В каталоге `~/work/arch-рс/lab04` с помощью команды `ср` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2022-2023/“Архитектура компьютера”/arch-рс/labs/lab04/`. Загрузите файлы на Github.

3 Выполнение лабораторной работы

1) Переходим в каталог lab04. Создадим текстовый файл с именем hello.asm. Откроем этот файл с помощью любого текстового редактора gedit и вставим в него текст.

```
iebeloslyudov@dk8n76 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ touch hello.asm
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ gedit hello.asm
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.1: Переходим в каталог

```
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. 3.2: Вставляем текст

2) Напишем код для компиляции приведенного текста программы “Hello World”. С помощью команды ls проверим, что объектный файл был создан.

```
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -f elf hello.asm
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls hello.asm hello.o
hello.asm hello.o
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.3: Код и проверка

3) Скомпилируем исходный файл `hello.asm` в `obj.o`. С помощью команды `ls` проверим, что файлы были созданы.

```
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm hello.o list.lst obj.o presentation report
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.4: Проверка

4) Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику. С помощью команды `ls` проверим, что исполняемый файл `hello` создан.

```
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 hello.o -o hello
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello hello.asm hello.o list.lst obj.o presentation report
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 obj.o -o main
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello hello.asm hello.o list.lst main obj.o presentation report
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.5: Проверка

5) Запустим на выполнение созданный исполняемый файл.

```
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./hello
Hello world!
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.6: Запуск

4 Выполнение лабораторной работы

- 1) Скопируем с помощью команды `cp` создаем копию файла `hello.asm` с именем `lab4.asm`

```
iebeloslyudov@dk8n76 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ cp hello.asm lab04.asm
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 4.1: Копируем

- 2) С помощью редактора внесем изменения в текст программы в файле `lab04.asm` так, чтобы вместо `Hello World!` на экран выводилась строка с фамилией и именем.
- 3) Оттранслируем полученный текст программы `lab04.asm` в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -f elf -g -l list1.lst lab04.asm
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 lab04.o -o lab04
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./lab04
Белослюдов Иван
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 4.2: Оттранслируем

- 4) Загрузим файлы на GitHub.


```

iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04/report $ git add .
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04/report $ git commit -am "Добавил файл"
[master ed49141] Добавил файл
21 files changed, 209 insertions(+), 119 deletions(-)
create mode 100755 labs/lab04/hello
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/hello.o
create mode 100755 labs/lab04/lab04
create mode 100644 labs/lab04/lab04.asm
create mode 100644 labs/lab04/lab04.o
create mode 100644 labs/lab04/list.lst
create mode 100644 labs/lab04/list1.lst
create mode 100755 labs/lab04/main
create mode 100644 labs/lab04/obj.o
create mode 100644 labs/lab04/report/image/1.png
create mode 100644 labs/lab04/report/image/2.png
create mode 100644 labs/lab04/report/image/3.png
create mode 100644 labs/lab04/report/image/4.png
create mode 100644 labs/lab04/report/image/5.png
create mode 100644 labs/lab04/report/image/6.png
create mode 100644 labs/lab04/report/image/7.png
create mode 100644 labs/lab04/report/image/8.png
create mode 100644 labs/lab04/report/report.docx
rewrite labs/lab04/report/report.md (72%)
create mode 100644 labs/lab04/report/report.pdf
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04/report $ git push
Перечисление объектов: 33, готово.
Подсчет объектов: 100% (33/33), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (27/27), готово.
Запись объектов: 100% (27/27), 819.21 КиБ | 22.14 Миб/с, готово.
Всего 27 (изменений 7), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (7/7), completed with 2 local objects.
To github.com:VanuaBel/study_2022-2023_arh-pc.git
   b34cfb7..ed49141  master -> master
iebeloslyudov@dk8n76 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04/report $

```

Рис. 4.3: Github

5 Выводы

В ходе выполнения лабораторной работы №4 я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.