

Лабораторная работа №4

Архитектура внутренних систем

Ежова Алиса Михайловна

Содержание

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

Список иллюстраций

3.1	Каталог	6
3.2	Созданный каталог	6
3.3	Текстовый файл	6
3.4	Открытие файла	6
3.5	Текст в файле	7
3.6	Hello world	7
3.7	Проверка	7
3.8	Выполнение команды	7
3.9	Проверка	7
3.10	Обработка компоновщику	8
3.11	Проверка	8
3.12	Выполнение команды	8
3.13	Формат командной строки	8
3.14	Запуск созданного файла	9
3.15	Копия файла	9
3.16	Изменения файла	9
3.17	Оттранслированный текст	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 `lab04.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab04.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab04.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab04.asm` в Ваш локальный репозиторий в каталог `~/work/study/2022-2023/“Архитектура компьютера”/archpc/labs/lab04/`. Загрузите файлы на Github.

3 Выполнение лабораторной работы

- 1) Рассмотрим пример простой программы на языке ассемблера NASM. Создадим каталог для работы с программами на языке ассемблера NASM (уже существует):

```
amezhova@dk5n60 ~ $ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04
mkdir: невозможно создать каталог «/afs/.dk.sci.pfu.edu.ru/home/a/m/amezhova/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04»: Файл существует
```

Рис. 3.1: Каталог

- 2) Перейдем в созданный каталог:

```
amezhova@dk5n60 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.2: Созданный каталог

- 3) Создадим текстовый файл с именем hello.asm:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ touch hello.asm
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

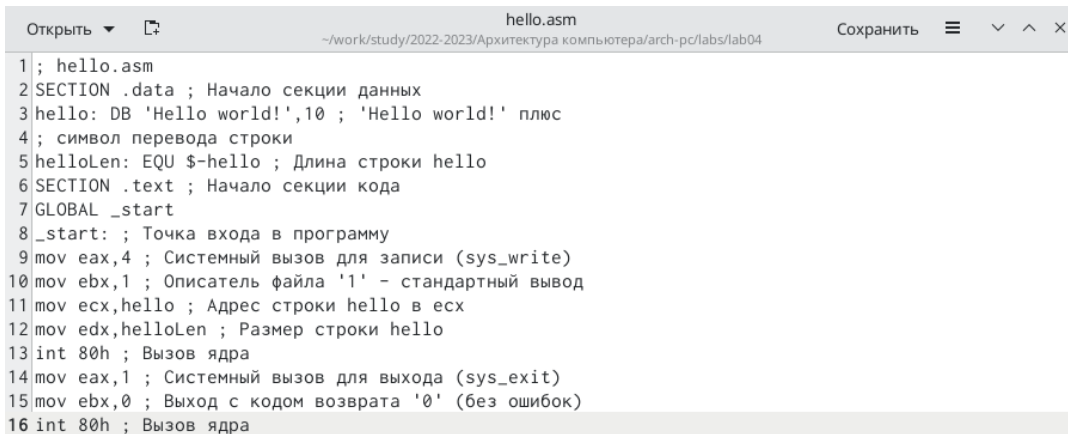
Рис. 3.3: Текстовый файл

- 4) Откроем этот файл с помощью любого текстового редактора, например, getit:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ gedit hello.asm
```

Рис. 3.4: Открытие файла

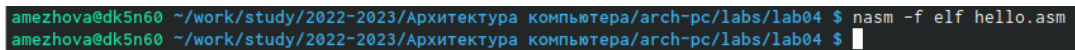
и введем в него следующий текст:



```
hello.asm
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. 3.5: Текст в файле

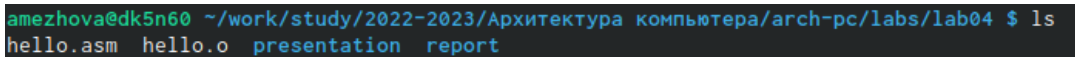
- 5) Напишем, необходимый для компиляции приведенного выше текст программы “Hello World”:



```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -f elf hello.asm
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.6: Hello world

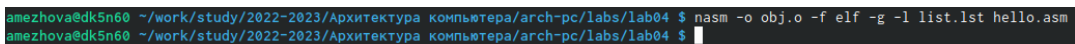
- 6) С помощью команды ls проверим, что объектный файл был создан:



```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm  hello.o  presentation  report
```

Рис. 3.7: Проверка

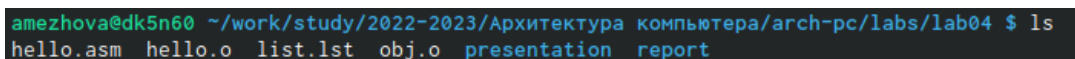
- 7) Выполним следующую команду:



```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.8: Выполнение команды

- 8) С помощью команды ls проверим, что файлы были созданы:



```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o  presentation  report
```

Рис. 3.9: Проверка

- 9) Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 hello.o -o hello
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.10: Обработка компоновщику

- 10) С помощью команды `ls` проверим, что исполняемый файл `hello` был создан:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o  presentation  report
```

Рис. 3.11: Проверка

- 11) Выполним следующую команду:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 obj.o -o main
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.12: Выполнение команды

- 12) Формат командной строки LD можно увидеть, набрав `ld --help`:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld --help
Использование ld [параметры] файл...
Параметры:
  -a КЛЮЧЕВОЕ СЛОВО                Управление общей библиотекой для совместимости с HP/UX
  -A АРХИТЕКТУРА, --architecture АРХИТЕКТУРА  Задать архитектуру
  -b ЦЕЛЬ, --format ЦЕЛЬ            Задать цель для следующих входных файлов
  -c ФАЙЛ, --mri-script ФАЙЛ        Прочитать сценарий компоновщика в формате MRI
  -d, -dc, -dp                     Принудительно делать общие символы определёнными
  --dependency-file ФАЙЛ            Write dependency file
  --force-group-allocation          Принудительно удалить членов группы из групп
  -e АДРЕС, --entry АДРЕС
```

Рис. 3.13: Формат командной строки

- 13) Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге, набрав в командной строке:


```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./hello
Hello world!
```

Рис. 3.14: Запуск созданного файла

- 14) В каталоге `~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/labs04` с помощью команды `cp` создала копию файла `hello.asm` с именем `lab4.asm`:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ cp hello.asm lab04.asm
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $
```

Рис. 3.15: Копия файла

- 15) С помощью любого текстового редактора внесла изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 lab04.o -o hello
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ld -m elf_i386 obj2.o -o main
```

Рис. 3.16: Изменения файла

- 16) Оттранслировала полученный текст программы `lab04.asm` в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл:

```
amezhova@dk5n60 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab04 $ ./hello
Ежова Алиса
```

Рис. 3.17: Оттранслированный текст

- 17) Загрузила файлы Github.

4 Вывод

Я освоила процедуру компиляции и сборки программ, написанных на ассемблере NASM.