

Отчет по лабораторной работе №4.

Создание и процесс обработки программ на языке ассемблера NASM

Валиева Марина Русланбековна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	11

Список иллюстраций

3.1	Создание каталога	6
3.2	Переход в созданный каталог	6
3.3	Создание текстового файла	6
3.4	Открытие этого файла	6
3.5	Введение текста	7
3.6	Программа “Hello, world”	7
3.7	Выполнение команды	7
3.8	Проверка	7
3.9	Передача файла на обработку компоновщику	8
3.10	Проверка	8
3.11	Выполнение команды	8
3.12	Запуск на выполнение созданного исполняемого файла	8
3.13	Копирование	8

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. 4.3.1. Программа Hello world!

Создадим каталог для работы с программами на языке ассемблера NASM:

```
mrvalieva@dk3n62 ~ $ mkdir ~/work/study/2022-2023/Архитектура\ компьютера/arch-  
pc/lab04
```

Рис. 3.1: Создание каталога

Перейдем в созданный каталог

```
mrvalieva@dk3n62 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/  
lab04
```

Рис. 3.2: Переход в созданный каталог

Создадим текстовый файл с именем hello.asm

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
touch hello.asm
```

Рис. 3.3: Создание текстового файла

откроем этот файл с помощью любого текстового редактора, например, gedit

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
gedit hello.asm
```

Рис. 3.4: Открытие этого файла

и введем в него текст:

```

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: Введение текста

4.3.2. Транслятор NASM NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать:

```

mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $
nasm -f elf hello.asm

```

Рис. 3.6: Программа “Hello, world”

4.3.3. Выполним следующую команду:

```

mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $
nasm -o obj.o -f elf -g -l list.lst hello.asm

```

Рис. 3.7: Выполнение команды

С помощью команды ls проверим, что файлы были созданы.

```

mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $
ls
hello hello.asm hello.o list.lst main obj.o

```

Рис. 3.8: Проверка

4.4. Компоновщик LD Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
ld -m elf_i386 hello.o -o hello
```

Рис. 3.9: Передача файла на обработку компоновщику

С помощью команды ls проверим, что исполняемый файл hello был создан.

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
ls  
hello hello.asm hello.o list.lst main obj.o
```

Рис. 3.10: Проверка

Выполним следующую команду:

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
ld -m elf_i386 obj.o -o main
```

Рис. 3.11: Выполнение команды

4.4.1. Запуск исполняемого файла Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге:

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
./hello  
Hello world!
```

Рис. 3.12: Запуск на выполнение созданного исполняемого файла

4.5. Выполнение самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создадим копию файла hello.asm с именем lab4.asm

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
cp hello.asm lab04.asm  
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
nasm -f elf -g -l list.lst lab04.asm
```

Рис. 3.13: Копирование

2. С помощью любого текстового редактора внесем изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с нашими фамилией и именем.

```
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
nasm -o Valieva.o -f elf -g -l list1.lst lab04.asm  
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
ld -m elf_i386 Valieva.o -o Valieva  
mrvalieva@dk3n62 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $  
./Valieva  
Valieva Marina
```

```
1 ; hello.asm  
2 SECTION .data  
3 hello: DB 'Valieva Marina',10  
4  
5 helloLen: EQU $-hello  
6  
7 SECTION .text  
8 GLOBAL _start  
9  
10 _start:  
11 mov eax,4  
12 mov ebx,1  
13 mov ecx,hello  
14 mov edx,helloLen  
15 int 80h  
16 mov eax,1  
17 mov ebx,0  
18 int 80h
```

3. Оттранслируем полученный текст программы lab4.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

4. Скопируем файлы `hello.asm` и `lab4.asm` в наш локальный репозиторий в каталог `~/work/study/2022-2023/“Архитектура компьютера”/arch-рс/labs/lab05/`. Загрузим файлы на Github.

4 Выводы

В результате лабораторной работы №4 я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.