

**Отчёт по лабораторной работе №4.
Создание и процесс обработки программ
на языке ассемблера NASM**

Айдарбекова Алия НММбд-01-23

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Задание для самостоятельной работы	10
6	Выводы	11

Список иллюстраций

4.1	Создание файла для работы с программой Hello world!	8
4.2	Текст нужный	8
4.3	Превращаем текст в объектный код	8
4.4	Компиляция	9
4.5	Создание файла и и обработка	9
4.6	Запуск исполняемого файла	9
5.1	Копия	10
5.2	Запуск файла	10

Список таблиц

1 Цель работы

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

2 Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла
6. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Язык ассемблера - машинно-ориентированный язык низкого уровня. Он больше всего приближен к архитектуре ЭВМ и её аппаратным возможностям. Это язык, с помощью которого пишутся команды для процессора понятным для человека образом. Следует отметить, процессор понимает машинные коды. NASM - это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. Типичный формат записи команд: '[метка:] мнемокод [операнд {, операнд}] [;комментарий]' Здесь мнемокод - непосредственно мнемоника инструкции процессору, являющейся обязательной частью команды.

4 Выполнение лабораторной работы

1. Создаем каталог для работы с программами на языке ассемблера NASM и переходим туда. Далее создаем текстовый файл внутри и открываем его с помощью текстового редактора gedit

```
arayjdarbekova@dk3n62 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/report $ gedit report.md
arayjdarbekova@dk3n62 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/report $ cd ~/work/arch-pc/lab04
arayjdarbekova@dk3n62 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 4.1: Создание файла для работы с программой Hello world!

2. Вводим необходимый текст. Компилируем его.

```
1; hello.asm
2SECTION .data          ; Начало секции данных
3hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4                      ; символ перевода строки
5helloLen: EQU $-hello   ; Длина строки hello
6
7SECTION .text ; Начало секции кода
8GLOBAL _start
9
10_start:              ; Точка входа в программу
11mov eax,4             ; Системный вызов для записи (sys_write)
12mov ebx,1             ; Описатель файла '1' - стандартный вывод
13mov ecx,hello         ; Адрес строки hello в ecx
14mov edx,helloLen      ; Размер строки hello
15int 80h              ; Вызов ядра
16
17mov eax,1             ; Системный вызов для выхода (sys_exit)
18mov ebx,0             ; Выход с кодом возврата '0' (без ошибок)
19int 80h              ; Вызов ядра
```

Рис. 4.2: Текст нужный

```
arayjdarbekova@dk3n62 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
arayjdarbekova@dk3n62 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.2nd list.lst main obj.o
```

Рис. 4.3: Превращаем текст в объектный код

3. Скомпилируем исходный файл в obj.o. Дополнительно будет создан файл листинга

```
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.2nd  list.lst  main  obj.o
```

Рис. 4.4: Компиляция

4. Передаем объектный файл на обработку компоновщику

```
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.2nd  list.lst  main  obj.o
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 4.5: Создание файла и и обработка

5. Запускаем на выполнение созданный файл

```
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $ ./hello
Hello world!
arraydarbekova@dk3n62 ~/work/arch-pc/lab04 $
```

Рис. 4.6: Запуск исполняемого файла

5 Задание для самостоятельной работы

1. Создаем копию файла

```
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
```

Рис. 5.1: Копия

2. С помощью текстового редактора меняем фразу Hello world! на свои фамилию и имя.
3. Оттранслируем полученный текст программы в объектный файл, выполняя компоновку и запускаем получившийся исполняемый файл.

```
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.2nd lab4.asm
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4.asm lab4.o list.2nd list.lst main obj.o
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4.asm lab4.o list.2nd list.lst main obj.o
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ./hello
Hello world!
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ./lab4
bash: ./lab4: Нет такого файла или каталога
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.2nd list.lst main obj.o
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
arayjdarbekova@dk4n65 ~/work/arch-pc/lab04 $ ./lab4
Айдарбекова Алия
```

Рис. 5.2: Запуск файла

6 Выводы

При выполнении данной лабораторной работы я познакомилась с ассемблером и его проектом NASM. Освоила компиляцию программы, написанной на ассемблере.