

Лабораторная работа №4

Создание и процесс обработки программ на языке ассемблера NASM

Бунин Арсений Викторович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	Создание папки	9
4.2	Создание файла	9
4.3	Ввод текста программы в файл	10
4.4	Создание объектных файлов	10
4.5	объектные файлы	11
4.6	Создание исполняемого файла	11
4.7	Результат работы программы	12

Список таблиц

1 Цель работы

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

2 Задание

1. Создать файл `hello.asm` в заданной директории.
2. С помощью любого текстового редактора вставить в файл текст программы.
3. Оттранслировать полученный текст программы в объектный файл, получить исполняемый файл.
4. Запустить исполняемый файл.

3 Теоретическое введение

Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства: арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в качестве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, преобразование (арифметические или логические операции) данных хранящихся в регистрах.

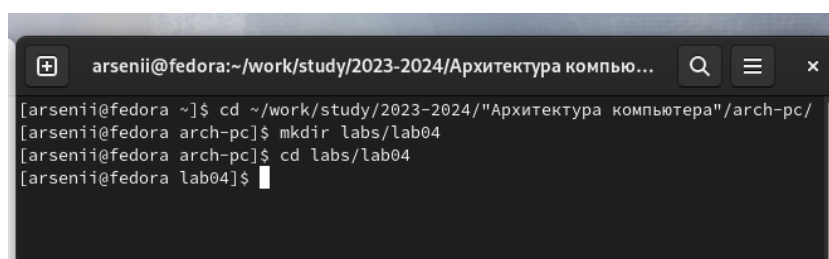
В процессе создания ассемблерной программы можно выделить четыре шага:

1. Набор текста программы в текстовом редакторе и сохранение её в отдельном файле. Каждый файл имеет свой тип (или расширение), который определяет назначение файла. Файлы с исходным текстом программ на языке ассемблера имеют тип `asm`.
2. Трансляция — преобразование с помощью транслятора, например `nasm`, текста программы в машинный код, называемый объектным.

На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — o, файла листинга — lst. 3. Компоновка или линковка — этап обработки объектного кода компоновщиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расширения. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение map. 4. Запуск программы. Конечной целью является работоспособный исполняемый файл. Ошибки на предыдущих этапах могут привести к некорректной работе программы, поэтому может присутствовать этап отладки программы при помощи специальной программы — отладчика. При нахождении ошибки необходимо провести коррекцию программы, начиная с первого шага. Из-за специфики программирования, а также по традиции для создания программ на языке ассемблера обычно пользуются утилитами командной строки (хотя поддержка ассемблера есть в некоторых универсальных интегрированных средах).

4 Выполнение лабораторной работы

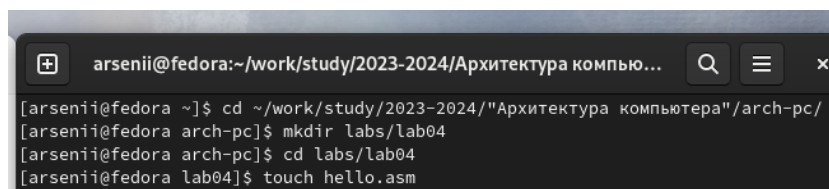
Открываем терминал и переходим в каталог курса, создали папку lab04(рис. 4.1).



```
arsenii@fedora:~/work/study/2023-2024/Архитектура компью...
[arsenii@fedora ~]$ cd ~/work/study/2023-2024/"Архитектура компьюте..."/arch-pc/
[arsenii@fedora arch-pc]$ mkdir labs/lab04
[arsenii@fedora arch-pc]$ cd labs/lab04
[arsenii@fedora lab04]$
```

Рис. 4.1: Создание папки

Создали файл hello.asm (рис. 4.2).



```
arsenii@fedora:~/work/study/2023-2024/Архитектура компью...
[arsenii@fedora ~]$ cd ~/work/study/2023-2024/"Архитектура компьюте..."/arch-pc/
[arsenii@fedora arch-pc]$ mkdir labs/lab04
[arsenii@fedora arch-pc]$ cd labs/lab04
[arsenii@fedora lab04]$ touch hello.asm
```

Рис. 4.2: Создание файла

Отредактировали файл (рис. 4.3).

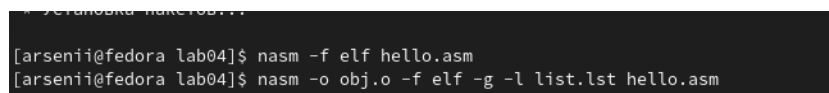


```
hello.asm
~/.work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04

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

Рис. 4.3: Ввод текста программы в файл

Сделали объектный файл двумя различными командами(рис. 4.4, 4.5).



```
~/.work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
[arsenii@fedora lab04]$ nasm -f elf hello.asm
[arsenii@fedora lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 4.4: Создание объектных файлов

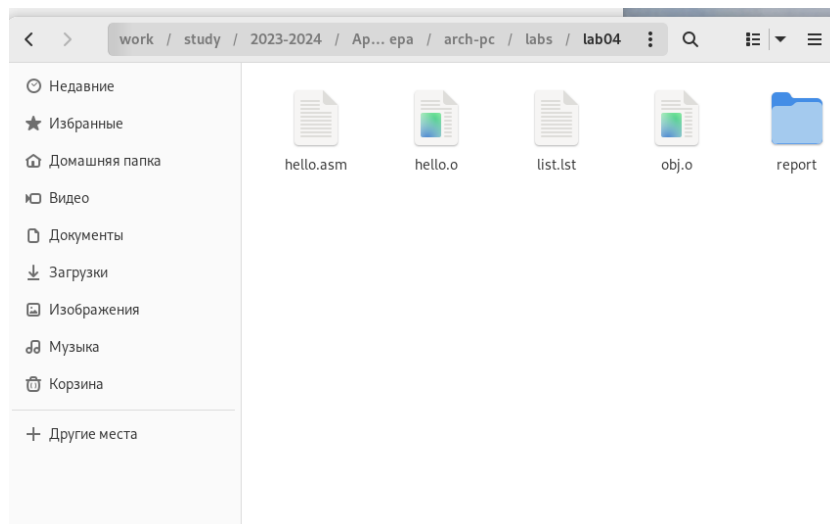


Рис. 4.5: объектные файлы

Сделали исполняемый файл (рис. 4.6)

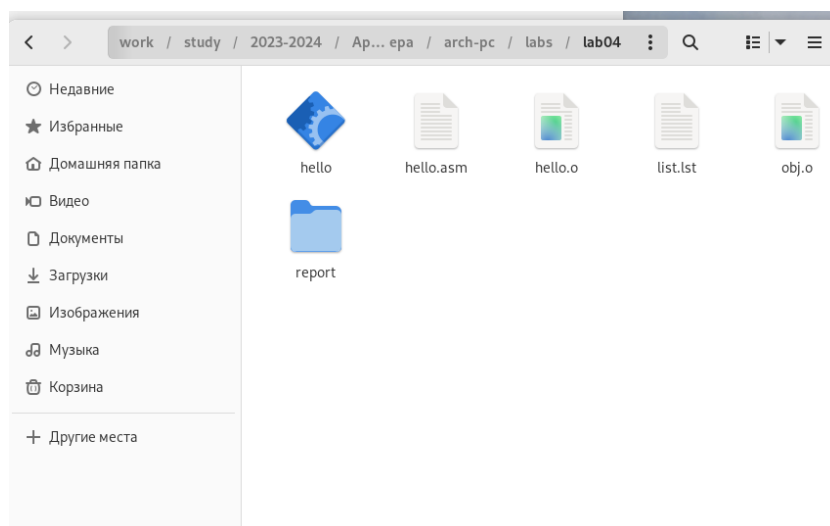
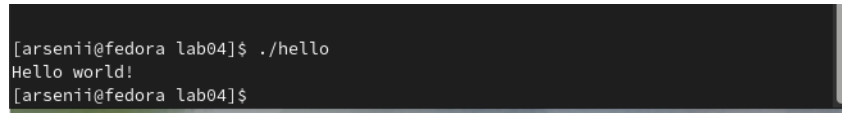


Рис. 4.6: Создание исполняемого файла

Запустили исполняемый файл (рис. 4.7)

A screenshot of a terminal window with a dark background. The text is white. It shows a user prompt, a command being executed, the output of the command, and the prompt again.

```
[arsenii@fedora lab04]$ ./hello  
Hello world!  
[arsenii@fedora lab04]$
```

Рис. 4.7: Результат работы программы

5 Выводы

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

Список литературы