

Лабораторная работа №4

Создание и процесс обработки программ на языке ассемблера NASM

Богату Ирина Владимировна

Содержание

Цель работы	5
Выполнение лабораторной работы	6
Выполнение задания для самостоятельной работы	10
Выводы	15

Список иллюстраций

1	Создание каталога lab4	6
2	Перемещение в созданный каталог	6
3	Создание .asm файлы	6
4	Открытие созданного файла с помощью gedit	7
5	Редактирование файла	7
6	Компиляция файла с помощью nasm	7
7	Проверка на успешное создание файла	8
8	Использование команды nasm с большим количеством аргументов	8
9	Проверка на успешное создание файлов	8
10	Сборка исполняемого файла с помощью ld	9
11	Проверка на успешное создание исполняемого файла	9
12	Сборка исполняемого файла main из файла obj.o	9
13	Проверка на успешное создание исполняемого файла	9
14	Запуск исполняемого файла	9
1	Копирование файла	10
2	Открытие файла для редактирования	11
3	Процесс редактирования файла	11
4	Компиляция файла в объектный	11
5	Сборка объектного файла в исполняемый	12
6	Запуск собранного файла	12
7	Копирование файла hello.asm в каталог 4 лабораторной работы .	12
8	Копирование файла lab4.asm в каталог 4 лабораторной работы . .	13
9	Загрузка проделанной работы на GitHub	14

Список таблиц

Цель работы

Научиться писать базовые программы на языке ассемблера NASM, компилировать их в объектные файлы и собирать из них исполняемые программы с помощью компоновщика.

Выполнение лабораторной работы

Перед выполнением лабораторной работы необходимо создать нужную директорию с помощью команды `mkdir` (Рис. 2.1):

```
ivbogatu@ivbogatu-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04
ivbogatu@ivbogatu-VirtualBox:~$
```

Рис. 1: Создание каталога lab4

Теперь переместимся в созданный нами каталог (Рис. 2.2):

```
ivbogatu@ivbogatu-VirtualBox:~$ cd ~/work/arch-pc/lab04
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 2: Перемещение в созданный каталог

Теперь создадим файл `hello` с расширением `.asm`, в котором мы будем писать код на ассемблере (Рис. 2.3):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ touch hello.asm
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 3: Создание .asm файлы

Для того, чтобы редактировать созданный файл, воспользуемся текстовым редактором `gedit` (Рис. 2.4):

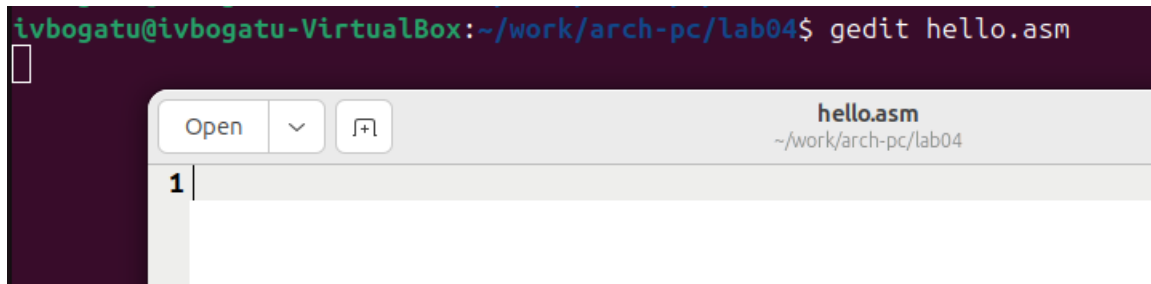


Рис. 4: Открытие созданного файла с помощью gedit

Вставим в открытый файл следующий код (Рис. 2.5):

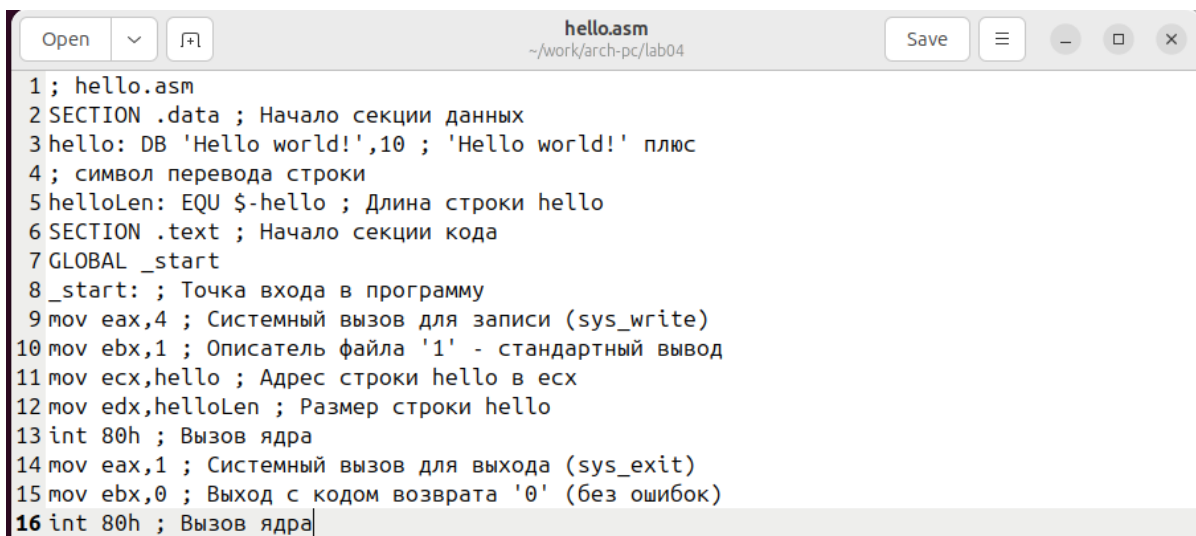


Рис. 5: Редактирование файла

Теперь нам необходимо превратить наш файл в объектный. Этим занимается транслятор NASM. Введём следующую команду (Рис. 2.6):

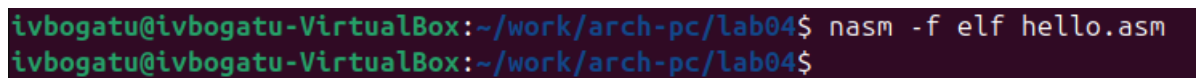


Рис. 6: Компиляция файла с помощью nasm

Здесь мы говорим создать из файла hello.asm объектный, указывая при этом формат файла elf (с помощью аргумента -f), то есть формат, работающий в си-

стемах семейства Linux. Далее проверим, создался ли объектный файл с помощью команды `ls` (Рис. 2.7):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 7: Проверка на успешное создание файла

Теперь попробуем использовать полный вариант команды `NASM` (Рис. 2.8):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 8: Использование команды `nasm` с большим количеством аргументов

Здесь мы указываем, что файл `hello.asm` должен быть скомпилирован в файл с названием `obj.o` (название указывается с помощью аргумента `-o`) в формате `elf` (аргументом `-f`) и включить туда символы для отладки (аргумент `-g`). Кроме того, мы укажем, что необходимо создать файл листинга `list.lst` (аргументом `-l`). Проверим, создался ли файл с помощью команды `ls` (Рис. 2.9):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 9: Проверка на успешное создание файлов

Для создания исполняемого файла необходимо использовать компоновщик `ld`, который соберёт объектный файл. Напишем следующую команду (Рис. 2.10):


```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 10: Сборка исполняемого файла с помощью ld

Здесь мы указываем формат `elf_i386` (с помощью аргумента `-m`) и файл для сборки, а аргументом `-o` указываем имя выходного файла. Мы назовём его `hello`. Проверим, создался ли файл с помощью команды `ls` (Рис. 2.11):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst obj.o
```

Рис. 11: Проверка на успешное создание исполняемого файла

Теперь соберём файл `obj.o` в файл `main` (Рис. 2.12):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 12: Сборка исполняемого файла `main` из файла `obj.o`

Теперь проверим, создался ли файл. Снова пропишем команду `ls` (Рис. 2.13):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Рис. 13: Проверка на успешное создание исполняемого файла

Теперь запустим файл `hello`, для этого мы должны написать `./` и название файла (Рис. 2.14):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 14: Запуск исполняемого файла

Выполнение задания для самостоятельной работы

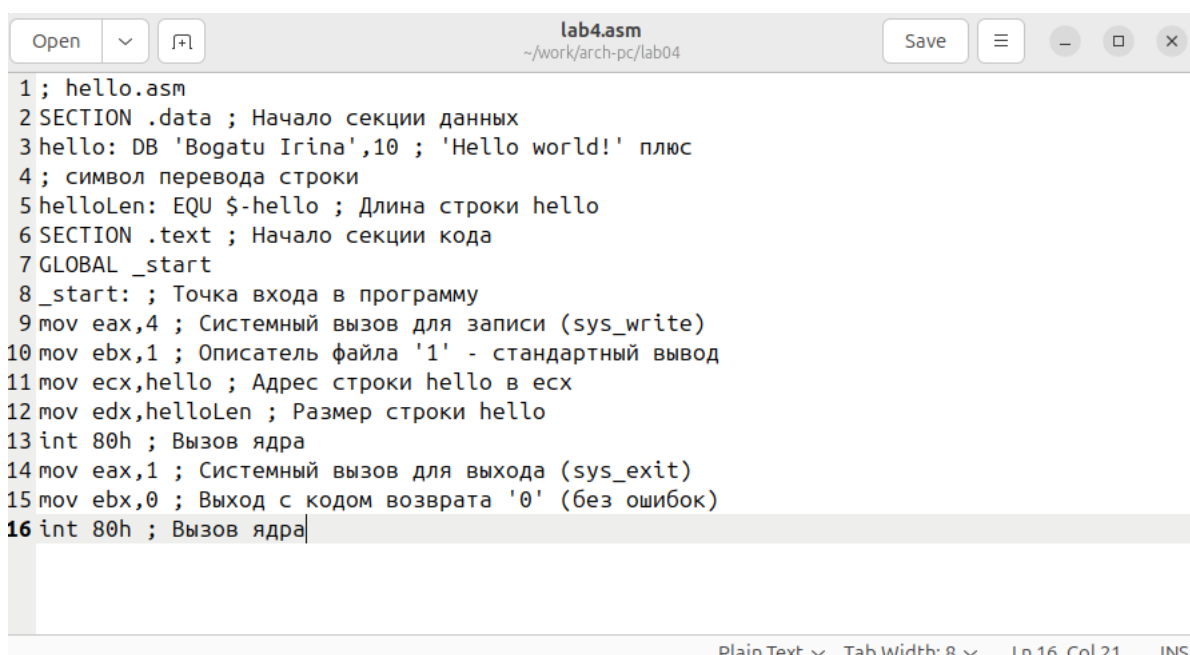
Скопируем файл `hello.asm` в каталог `~/work/arch-pc/lab04` под названием `lab4.asm` (Рис. 3.1):

A terminal window with a dark background. The prompt is `ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$` and the command entered is `gedit lab4.asm`.

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ gedit lab4.asm
```

Рис. 1: Копирование файла

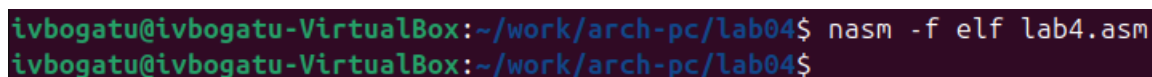
Внесём изменения в скопированный файл. Для этого откроем его в `gedit` (Рис. 3.2):



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Bogatu Irina',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. 2: Открытие файла для редактирования

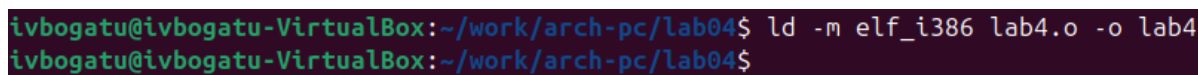
Теперь изменим третью строчку, заменив фразу Hello world! на фамилию и имя (Рис. 3.3):



```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 3: Процесс редактирования файла

Теперь скомпилируем полученный файл в объектный. Для этого воспользуемся командой `nasm` и укажем формат `elf` и нужный файл для компиляции (Рис. 3.4):



```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 4: Компиляция файла в объектный

Теперь соберём полученный объектный файл. Укажем формат `elf_i386` и объектный файл для сборки (`lab4.o`). Укажем, что выходной файл должен быть на-

зван lab4 (Рис. 3.5):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ ./lab4
Bogatu Irina
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$
```

Рис. 5: Сборка объектного файла в исполняемый

Убедимся в том, что сделали всё правильно. Для этого запустим собранный файл (Рис. 3.6):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04/
```

Рис. 6: Запуск собранного файла

Теперь скопируем файл hello.asm в каталог 4 лабораторной работы (Рис. 3.7):

```
ivbogatu@ivbogatu-VirtualBox:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04/
```

Рис. 7: Копирование файла hello.asm в каталог 4 лабораторной работы

Эту же операцию проведём для файла lab4.asm (Рис. 3.8):

```
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$ git add .  
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$ git commit -am "feat(main): added files hello.asm and lab4.asm"  
[master cc9bc26] feat(main): added files hello.asm and lab4.asm  
2 files changed, 32 insertions(+)  
create mode 100644 labs/lab04/hello.asm  
create mode 100644 labs/lab04/lab4.asm  
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$ git push  
Enumerating objects: 9, done.  
Counting objects: 100% (9/9), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (6/6), 985 bytes | 985.00 KiB/s, done.  
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.  
To github.com:qwetorri/study_2023-2024_arh--pc.git  
531f029..cc9bc26 master -> master  
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$
```

Рис. 8: Копирование файла lab4.asm в каталог 4 лабораторной работы

Теперь загрузим результат проделанной лабораторной работы на GitHub (Рис. 3.9):

```
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$ git add .  
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$ git commit -am "feat(main): added files hello.asm and lab4.asm"  
[master cc9bc26] feat(main): added files hello.asm and lab4.asm  
 2 files changed, 32 insertions(+)  
  create mode 100644 labs/lab04/hello.asm  
  create mode 100644 labs/lab04/lab4.asm  
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$ git push  
Enumerating objects: 9, done.  
Counting objects: 100% (9/9), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (6/6), 985 bytes | 985.00 KiB/s, done.  
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0  
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.  
To github.com:qwetorri/study_2023-2024_arh--pc.git  
 531f029..cc9bc26  master -> master  
ivbogatu@ivbogatu-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-  
pc$
```

Рис. 9: Загрузка проделанной работы на GitHub

Выводы

В результате выполнения лабораторной работы появилось понимание того, как работает алгоритм создания исполняемого файла из кода на ассемблере, а также появились навыки работы с языком `asm`, компиляции кода в объектный файл и сборкой исполняемых программ.