

Лабораторная работа №3

Архитектура компьютера

Мурашов Иван Вячеславович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Программа Hello world!	7
3.2	Работа с транслятором NASM	8
3.3	Работа с расширенным синтаксисом командной строки NASM . .	8
3.4	Работа с компоновщиком LD	9
3.5	Запуск исполняемого файла	9
3.6	Задания для самостоятельной работы	10
4	Выводы	13

Список иллюстраций

3.1	Создание каталога и перемещение между директориями	7
3.2	Создание файла и его открытие	7
3.3	Заполнение файла	8
3.4	Компиляция программы и просмотр каталога	8
3.5	Компиляция файлов и просмотр каталога	9
3.6	Компоновка файла и просмотр каталога	9
3.7	Запуск исполняемого файла	9
3.8	Заполнение файла	10
3.9	Компиляция файла и просмотр каталога	10
3.10	Компиляция файла и просмотр каталога	11
3.11	Запуск исполняемого файла	11
3.12	Каталог текущей лабораторной работы	11
3.13	Добавление и загрузка файлов на github	12

Список таблиц

1 Цель работы

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

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Выполнение лабораторной работы

3.1 Программа Hello world!

Создаю каталог для работы с программами на языке ассемблера NASM и перехожу в созданный каталог (рис. [3.1]).

```
[ivmurashov@fedora ~]$ mkdir -p ~/work/arch-pc/lab04  
[ivmurashov@fedora ~]$ cd ~/work/arch-pc/lab04
```

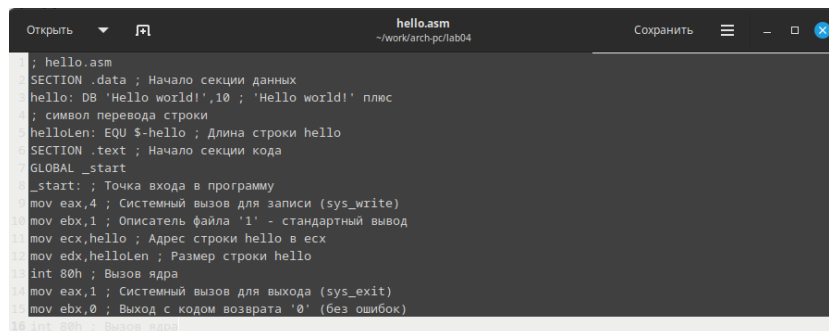
Рис. 3.1: Создание каталога и перемещение между директориями

Создаю текстовый файл с именем 'hello.asm' и открываю этот файл с помощью текстового редактора gedit (рис. [3.2]).

```
[ivmurashov@fedora lab04]$ touch hello.asm  
[ivmurashov@fedora lab04]$ gedit hello.asm
```

Рис. 3.2: Создание файла и его открытие

Ввожу приведённый в лекции текст в файл 'hello.asm' (рис. [3.3])

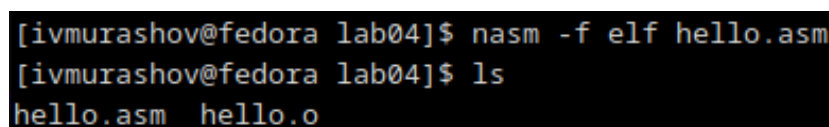


```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для 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. 3.3: Заполнение файла

3.2 Работа с транслятором NASM

Для компиляции текста программы “Hello World!” ввожу команду ‘nasm -f elf hello.asm’ и затем проверяю наличие скомпилированного файла ‘hello.o’ (рис. [3.4])



```
[ivmurashov@fedora lab04]$ nasm -f elf hello.asm
[ivmurashov@fedora lab04]$ ls
hello.asm  hello.o
```

Рис. 3.4: Компиляция программы и просмотр каталога

3.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду ‘nasm -o obj.o -f elf -g -l list.lst hello.asm’ для компиляции исходного файла ‘hello.asm’ в ‘obj.o’ и файла листинга ‘list.lst’. Проверяю корректность созданных файлов (рис. [3.5]).


```
[ivmurashov@fedora lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[ivmurashov@fedora lab04]$ ls
hello.asm  hello.o  list.lst  obj.o  presentation  report
```

Рис. 3.5: Компиляция файлов и просмотр каталога

3.4 Работа с компоновщиком LD

Передаю объектный файл на обработку компоновщику с помощью команды 'ld -m elf_i386 hello.o -o hello' и проверяю, что исполняемый файл 'hello' был создан (рис. [??]).

```
[ivmurashov@fedora lab04]$ ld -m elf_i386 hello.o -o hello
[ivmurashov@fedora lab04]$ ls
hello  hello.asm  hello.o  list.lst  obj.o  presentation  report
```

Выполняю команду 'ld -m elf_i386 obj.o -o main'. Исполняемый файл будет называться 'main', так как после ключа -o задано значение 'main'. Объектный файл, из которого собран этот исполняемый файл называется 'obj.o'. Просматриваю текущий каталог (рис. [3.6]).

```
[ivmurashov@fedora lab04]$ ld -m elf_i386 obj.o -o main
[ivmurashov@fedora lab04]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o  presentation  report
```

Рис. 3.6: Компоновка файла и просмотр каталога

3.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл, набрав команду './hello' (рис. [3.7]).

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

Рис. 3.7: Запуск исполняемого файла

3.6 Задания для самостоятельной работы

1. В каталоге lab04 с помощью команды 'cp' создаю копию файла 'hello.asm' с именем 'lab4.asm'. Проверяю наличие скопированного файла (рис. [3.7]).

```
[ivmurashov@fedora lab04]$ cp hello.asm lab4.asm
[ivmurashov@fedora lab04]$ ls
hello      hello.o   list.lst  obj.o     report
hello.asm  lab4.asm  main      presentation
```

2. С помощью текстового редактора gedit вношу изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моими фамилией и именем (рис. [3.8]).

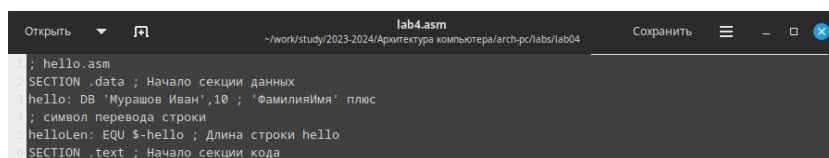


Рис. 3.8: Заполнение файла

3. Транслирую полученный текст программы lab4.asm в объектный файл. Проверяю наличие файла 'lab4.o' (рис. [3.9]).

```
[ivmurashov@fedora lab04]$ nasm -f elf lab4.asm
[ivmurashov@fedora lab04]$ ls
hello      hello.o   lab4.o    main      presentation
hello.asm  lab4.asm  list.lst  obj.o     report
```

Рис. 3.9: Компиляция файла и просмотр каталога

Выполняю компоновку объектного файла и проверяю наличие исполняемого файла 'lab4' (рис. [3.10]).

```
[ivmurashov@fedora lab04]$ ld -m elf_i386 lab4.o -o lab4
[ivmurashov@fedora lab04]$ ls
hello      hello.o  lab4.asm  list.lst  obj.o      report
hello.asm  lab4     lab4.o    main      presentation
```

Рис. 3.10: Компиляция файла и просмотр каталога

Запускаю получившийся исполняемый файл (рис. [3.11]).

```
[ivmurashov@fedora lab04]$ ./lab4
Мурашов Иван
```

Рис. 3.11: Запуск исполняемого файла

4. Копирую файлы hello.asm и lab4.asm в мой локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ через файловую систему (рис. [3.12]).

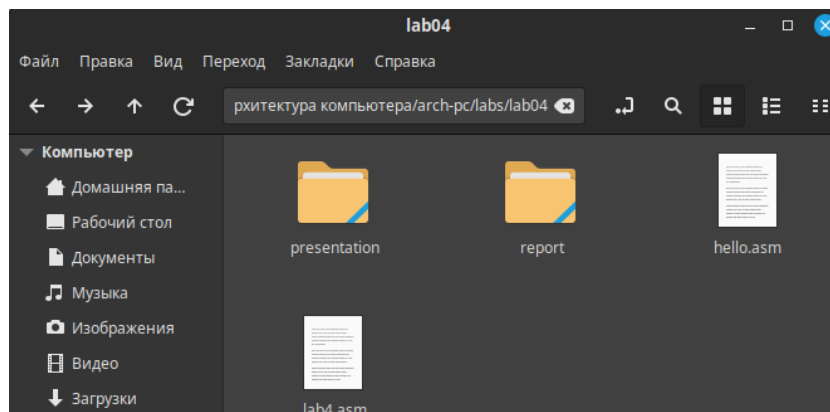


Рис. 3.12: Каталог текущей лабораторной работы

Загружаю файлы на Github (рис. [3.13]).

```

[ivmurashov@fedora lab04]$ git add .
[ivmurashov@fedora lab04]$ git commit -m 'Add files for Lab4'
Текущая ветка: master
Ваша ветка опережает «origin/master» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
.../.../prepare
.../.../presentation/

индекс пуст, но есть неотслеживаемые файлы
(используйте «git add», чтобы проиндексировать их)
[ivmurashov@fedora lab04]$ git push
Перечисление объектов: 28, готово.
Подсчет объектов: 100% (28/28), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (23/23), готово.
Запись объектов: 100% (23/23), 167.62 КиБ | 1.22 МиБ/с, готово.
Всего 23 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:neve7mind/study_2023-2024_arh-pc.git
 eb8d3e8..1dcd96f master -> master

```

Рис. 3.13: Добавление и загрузка файлов на github

4 Выводы

При выполнении данной лабораторной работой мной были освоены процедуры компиляции и сборки программ, написанных на ассемблере NASM.