

Отчёт по лабораторной работе №4

Архитектура компьютера

Андреева Софья Владимировна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Программа Hello world!	5
2.2	Транслятор NASM	6
2.3	Расширенный синтаксис командной строки NASM	6
2.4	Компоновщик LD.	6
2.5	Запуск исполняемого файла.	7
3	Задание для самостоятельной работы.	8
4	Выводы	10

Список иллюстраций

2.1	hello.asm	5
2.2	Ввод текста	5
2.3	Созданный объектный файл	6
2.4	Компиляция исходного файла, создание файлов	6
2.5	Исполняемый файл hello	6
2.6	Выполнение команды	7
2.7	ld —help	7
2.8	Запуск исполняемого файла	7
3.1	Создание копии файла hello.asm с именем lab4.asm	8
3.2	Изменения в тексте программы	8
3.3	Выполнение 3 задания	9
3.4	Копирование файлов	9
3.5	Загрузка файлов на Github	9

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Выполнение лабораторной работы

2.1 Программа Hello world!

Создадим каталог для работы с программами на языке ассемблера NASM, перейдем в него. Создадим текстовый файл с именем `hello.asm` и откроем его с помощью `gedit` (рис. 2.1).

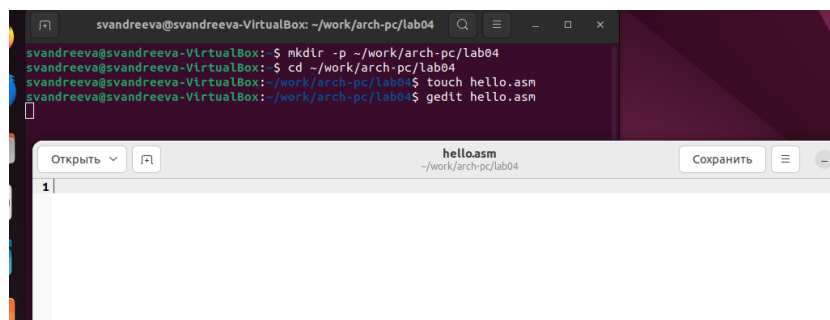


Рис. 2.1: hello.asm

Введем следующий текст (рис. 2.2).

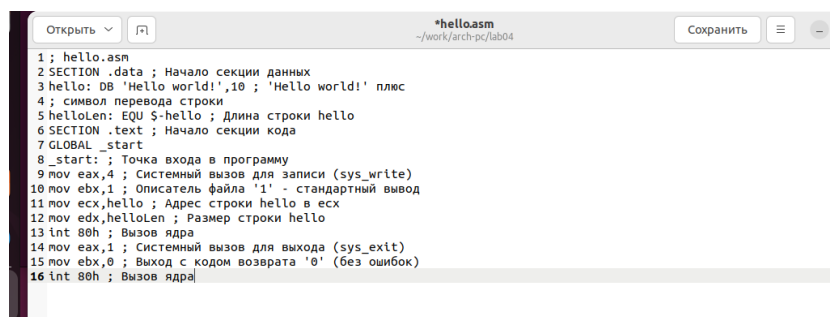


Рис. 2.2: Ввод текста

2.2 Транслятор NASM

С помощью транслятора преобразуем текст программы из файла hello.asm в объектный код, который запишется в файл hello.o. С помощью команды ls проверим, что объектный файл был создан. Объектный файл называется hello.o (рис. 2.3).

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

Рис. 2.3: Созданный объектный файл

2.3 Расширенный синтаксис командной строки NASM

Скомпилируем исходный файл hello.asm в obj.o, при этом формат выходного файла будет elf, и в него будут включены символы для отладки, кроме того, будет создан файл листинга list.lst. С помощью команды ls проверим, что файлы были созданы (рис. 2.4).

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

Рис. 2.4: Компиляция исходного файла, создание файлов

2.4 Компоновщик LD.

Передадим на обработку компоновщику объектный файл. С помощью команды ls проверим, что исполняемый файл hello был создан (рис. 2.5).

```
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 2.5: Исполняемый файл hello

Выполним следующую команду. Исполняемый файл будет иметь имя `main`. Объектный файл, из которого собран этот исполняемый файл, будет иметь имя `main.o` (рис. 2.6).

```
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
```

Рис. 2.6: Выполнение команды

Формат командной строки LD можно увидеть, набрав `ld --help` (рис. 2.7).

```
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ ld --help
Использование ld [параметры] файл...
Параметры:
  -a КЛЮЧЕВОЕ СЛОВО           Управление общей библиотекой для совместимости с HP/UX
  -A АРХИТЕКТУРА, --architecture АРХИТЕКТУРА      Принудительно делать общие символы определёнными
                                                    Задать архитектуру
  -b ЦЕЛЬ, --format ЦЕЛЬ      Задать цель для следующих входных файлов
  -c ФАЙЛ, --mri-script ФАЙЛ  Прочитать сценарий компоновщика в формате MRI
  -d, -dc, -dp               Принудительно делать общие символы определёнными
  --dependency-file ФАЙЛ     Write dependency file
  --force-group-allocation    Принудительно удалить членов группы из групп
  -e АДРЕС, --entry АДРЕС     Задать начальный адрес
  -E, --export-dynamic        Экспортировать все динамические символы
  --no-export-dynamic         Отменить действие --export-dynamic
  --enable-non-contiguous-regions
                              Enable support of non-contiguous memory regions
  --enable-non-contiguous-regions-warnings
                              Enable warnings when --enable-non-contiguous-regions may cause unexpected behavior
```

Рис. 2.7: `ld --help`

2.5 Запуск исполняемого файла.

Запустим исполняемый файл (рис. 2.8).

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

Рис. 2.8: Запуск исполняемого файла

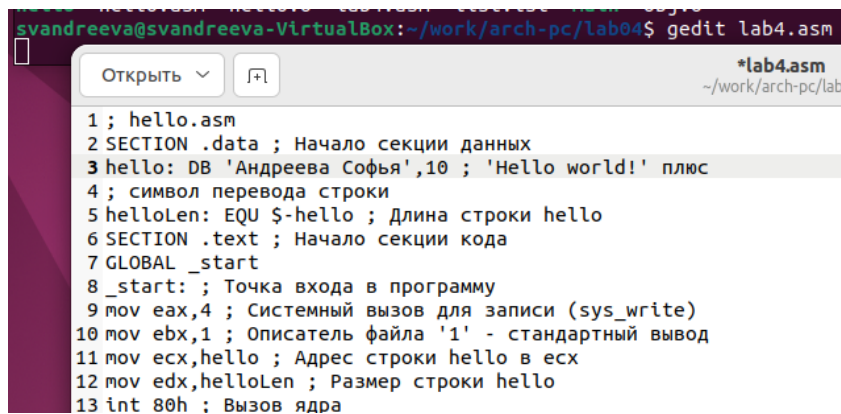
3 Задание для самостоятельной работы.

В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab4.asm` (рис. 3.1).

```
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

Рис. 3.1: Создание копии файла `hello.asm` с именем `lab4.asm`

С помощью любого текстового редактора внесем изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с моей фамилией и именем. (рис. 3.2).



```
svandreeva@svandreeva-VirtualBox:~/work/arch-pc/lab04$ gedit lab4.asm
Открыть v [F+] *lab4.asm
~/work/arch-pc/lab
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Андреева Софья',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
```

Рис. 3.2: Изменения в тексте программы

Оттранслируем полученный текст программы `lab4.asm` в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. (рис. 3.3).


```
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$ gcc -c lab4.asm
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$ ls
hello.o lab4 lab4.asm lab4.o list.lst main.o
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$ ./lab4
Андреева Софья
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$
```

Рис. 3.3: Выполнение 3 задания

Скопируем файлы hello.asm и lab4.asm в наш локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Загрузите файлы на Github.(рис. 3.4).

```
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2023-2024/“Архитектура компь
тера”/arch-pc/labs/lab04/
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2023-2024/“Архитектура компь
тера”/arch-pc/labs/lab04/
svandreeva@svandreeva-VirtualBox: ~/work/arch-pc/lab04$ cd ~/work/study/2023-2024/“Архитектура компьютера”/arc
h-pc/labs/lab04/
svandreeva@svandreeva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
svandreeva@svandreeva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 3.4: Копирование файлов

Загрузим файлы на Github (рис. 3.5).

```
svandreeva@svandreeva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git add .
svandreeva@svandreeva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git commit -am "feat(main):add fl
les lab04"
[master c4ce03c] feat(main):add files lab04
14 files changed, 32 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/image/1.png
create mode 100644 labs/lab04/report/image/10.png
create mode 100644 labs/lab04/report/image/11.png
create mode 100644 labs/lab04/report/image/12.png
create mode 100644 labs/lab04/report/image/2.png
create mode 100644 labs/lab04/report/image/3.png
create mode 100644 labs/lab04/report/image/4.jpg
create mode 100644 labs/lab04/report/image/5.png
create mode 100644 labs/lab04/report/image/6.png
create mode 100644 labs/lab04/report/image/7.png
create mode 100644 labs/lab04/report/image/8.png
create mode 100644 labs/lab04/report/image/9.png
svandreeva@svandreeva-VirtualBox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git push
Перечисление объектов: 25, готово.
Подсчет объектов: 100% (25/25), готово.
Сжатие объектов: 100% (20/20), готово.
Запись объектов: 100% (20/20), 559.19 КБ | 3.97 МБ/с, готово.
Всего: 30 (созданий: 1) — передача: 100% (20/20) — сжатие: 0% (0/0) — проверка: 100% (20/20) — готово.
```

Рис. 3.5: Загрузка файлов на Github

4 Выводы

В ходе лабораторной работы были освоены процедуры компиляции и сборки программ, написанных на машинноориентированном языке низкого уровня, ассемблере NASM.