

Отчёт по лабораторной работе №4

Дисциплина: компьютерные науки и технологии программирования

Мирзоян Ян Игоревич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Создание программы Hello world!	8
4.2	Работа с транслятором NASM	9
4.3	Работа с расширенным синтаксисом командной строки NASM . .	9
4.4	Работа с компоновщиком LD	9
4.5	Запуск исполняемого файла	10
4.6	Выполнение заданий для самостоятельной работы.	10
5	Выводы	12
	Список литературы	13

Список иллюстраций

4.1	Создал каталог для работы и перешел в него	8
4.2	Создал необходимый файл и открыл его в текстовом редакторе .	8
4.3	Заполнил файл согласно шаблону	8
4.4	Преобразую текст файла в объективный код и проверяю наличие файла с помощью ls	9
4.5	Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o, при этом в файл будут включены символы для отладки (ключ -g), также с помощью ключа -l будет создан файл листинга list.lst. Далее проверяю с помощью утилиты ls правильность выполнения команды.	9
4.6	Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello и проверяю наличие файла с помощью ls	9
4.7	Выполняю команду. Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o	10
4.8	Запускаю файл с помощью указанной команды	10
4.9	Копирую файл открываю его в текстовом редакторе, корректирую согласно заданию	10
4.10	Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4	10
4.11	Запускаю исполняемый файл lab4, на экран действительно выводятся мои имя и фамилия	11
4.12	Копирую два файла в указанный каталог, проверяю их наличие .	11

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

Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства (рис. 4.1). Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской (системной) плате.

4 Выполнение лабораторной работы

4.1 Создание программы Hello world!

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

Рис. 4.1: Создал каталог для работы и перешел в него

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

Рис. 4.2: Создал необходимый файл и открыл его в текстовом редакторе

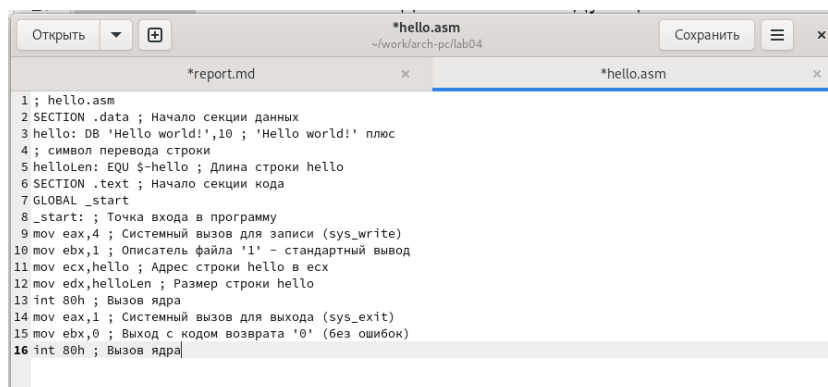


Рис. 4.3: Заполнил файл согласно шаблону

4.2 Работа с транслятором NASM

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

Рис. 4.4: Преобразую текст файла в объективный код и проверяю наличие файла с помощью ls

4.3 Работа с расширенным синтаксисом командной строки NASM

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

Рис. 4.5: Ввожу команду, которая скомпилирует файл hello.asm в файл obj.o, при этом в файл будут включены символы для отладки (ключ -g), также с помощью ключа -l будет создан файл листинга list.lst. Далее проверяю с помощью утилиты ls правильность выполнения команды.

4.4 Работа с компоновщиком LD

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

Рис. 4.6: Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello и проверяю наличие файла с помощью ls

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

Рис. 4.7: Выполняю команду. Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o

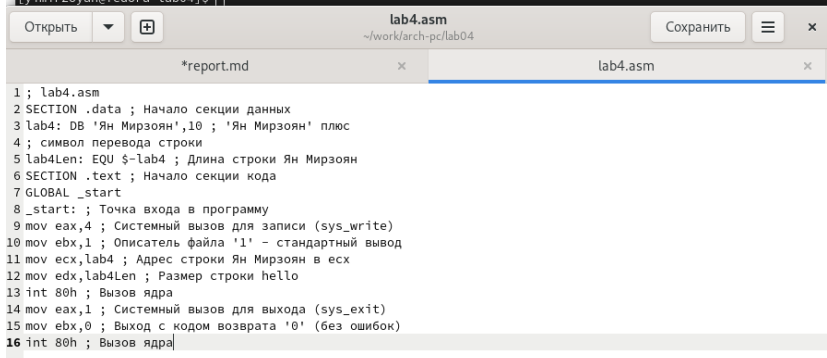
4.5 Запуск исполняемого файла

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

Рис. 4.8: Запускаю файл с помощью указанной команды

4.6 Выполнение заданий для самостоятельной работы.

```
[yimirzoyan@fedora lab04]$ cp hello.asm lab4.asm
[yimirzoyan@fedora lab04]$ gedit lab4.asm
[yimirzoyan@fedora lab04]$
```



```
1; lab4.asm
2 SECTION .data ; Начало секции данных
3 lab4: DB 'Ян Мирзоян',10 ; 'Ян Мирзоян' плюс
4 ; символ перевода строки
5 lab4Len: EQU $-lab4 ; Длина строки Ян Мирзоян
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,lab4 ; Адрес строки Ян Мирзоян в ecx
12 mov edx,lab4Len ; 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. 4.9: Копирую файл открываю его в текстовом редакторе, корректирую согласно заданию

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

Рис. 4.10: Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4

```
[yimirzoyan@fedora lab04]$ ./lab4
Ян Мирзоян
[yimirzoyan@fedora lab04]$
```

Рис. 4.11: Запускаю исполняемый файл lab4, на экран действительно выводятся мои имя и фамилия

```
[yimirzoyan@fedora lab04]$ cp hello.asm /home/yimirzoyan/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04
[yimirzoyan@fedora lab04]$ cp lab4.asm /home/yimirzoyan/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04
[yimirzoyan@fedora lab04]$ cd /home/yimirzoyan/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04
[yimirzoyan@fedora lab04]$ ls
hello.asm lab4.asm presentation report
[yimirzoyan@fedora lab04]$
```

Рис. 4.12: Копирую два файла в указанный каталог, проверяю их наличие

5 Выводы

При выполнении данной лабораторной работы освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
15. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-

- е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).