

Отчет по лабораторной работе №4

Архитектура компьютера

Дмитрий Константинович Кобзев

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	9
5	Выводы	11
	Список литературы	12

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Задание

1. В каталоге `~/work/arch-рс/lab04` с помощью команды `ср` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-рс/labs/lab04/`. Загрузите файлы на Github.

3 Выполнение лабораторной работы I

[1–6]

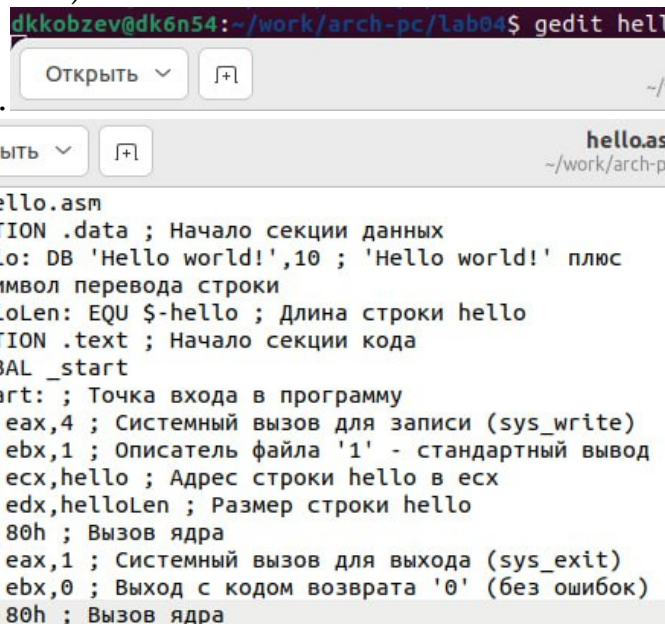
Создаем каталог для работы с программами на языке ассемблера NASM (рис.

1.1). `dkkobzev@dk6n54:~$ mkdir -p ~/work/arch-pc/lab04`

Переходим в созданный каталог (рис. 1.2). `dkkobzev@dk6n54:~$ cd ~/work/arch-pc/lab04`

Создаем текстовый файл с именем hello.asm (рис. 1.3). `dkkobzev@dk6n54:~/work/arch-pc/lab04$ gedit hello.asm`

Открываем этот файл с помощью gedit (рис. 1.4).



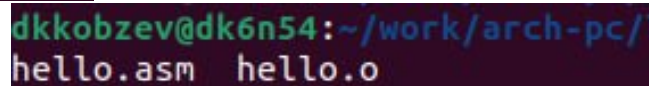
```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для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

Вводим в него следующий текст (рис. 1.5).

Компилируем приведенного выше текста программы “Hello World” (рис. 1.6).

`dkkobzev@dk6n54:~/work/arch-pc/lab04$ nasm -f elf hello.asm`

Проверяем, что объектный файл создан (рис. 1.7).



```
dkkobzev@dk6n54:~/work/arch-pc/lab04$ ls hello.asm hello.o
```

Компилируем исходный файл hello.asm в obj.o и создаем файл листинга list.lst

(рис. 1.8). `dkkobzev@dk6n54:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm`

Проверяем правильность выполнения программы (рис. 1.9).

```
dkkobzev@dk6n54:~/work/a  
hello.asm hello.o list
```

Передаем объектный файл на обработку компоновщику(рис. 1.10).

```
dkkobzev@dk6n54:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
```

Проверяем правильность выполнения программы (рис. 1.11).

```
dkkobzev@dk6n54:~/work/a  
hello hello.asm hello.o
```

Выполняем следующую команду (рис. 1.12).

```
dkkobzev@dk6n54:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o
```

Запускаем на выполнение созданный исполняемый файл, находящийся в теку-

щем каталоге (рис. 1.13).

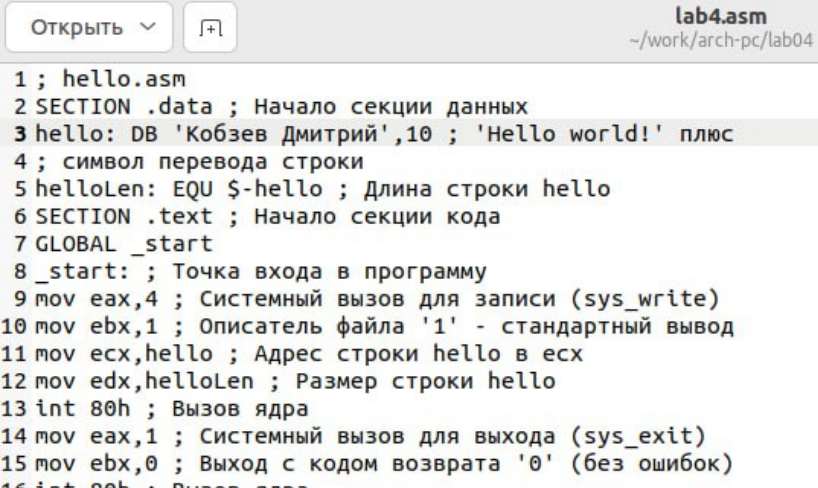
```
dkkobzev@dk6n54:~/work/arch-pc/lab04$ ./hello  
Hello world!
```


4 Самостоятельная работа

Задание 1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создаем копию файла `hello.asm` с именем `lab4.asm` (рис. 2.1).

```
dkkobzev@dk6n54:~/work/arch-pc/lab04$ cp hello.a
```

Задание 2. С помощью любого текстового редактора вносим изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с моими фамилией и именем. (рис. 2.2).



```
Открыть ▾ [icon] lab4.asm
~/work/arch-pc/lab04
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Кобзев Дмитрий',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
```

Задание 3. Оттранслируем полученный текст программы `lab4.asm` в объектный файл. Выполним компоновку объектного файла и запустим получившийся

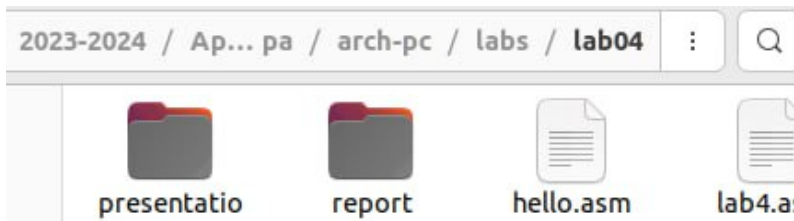
```
dkkobzev@dk6n54:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
dkkobzev@dk6n54:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.a
sm
dkkobzev@dk6n54:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
dkkobzev@dk6n54:~/work/arch-pc/lab04$ ./lab4
Кобзев Дмитрий
```

исполняемый файл.

Задание 4. Скопируем файлы `hello.asm` и `lab4.asm` в мой локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-`

pc/labs/lab04/. Загрузим файлы на Github.

```
dkkobzev@dk6n54:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
$ git add .
dkkobzev@dk6n54:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
$ git commit -am 'feat(main): make course structure'
[master 30c6c28] feat(main): make course structure
 5 files changed, 33 insertions(+), 1 deletion(-)
 create mode 100644 labs/lab04/hello.asm
 create mode 100644 labs/lab04/lab4.asm
dkkobzev@dk6n54:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
$ git push
Перечисление объектов: 19, готово.
Подсчет объектов: 100% (19/19), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (11/11), готово.
Запись объектов: 100% (11/11), 30.94 Киб | 2.81 Миб/с, готово.
Всего 11 (изменений 7), повторно использовано 0 (изменений 0), повторно использо
вано пакетов 0
remote: Resolving deltas: 100% (7/7), completed with 6 local objects.
To github.com:dkkobzev/study_2023-2024_arh-pc.git
 3d2c855..30c6c28 master -> master
```



5 Выводы

В ходе выполнения лабораторной работы мною были освоены процедуры компиляции и сборки программ, написанных на ассемблере NASM..

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.