

# Объектно-ориентированное программирование. Лабораторная работа №2. Основы программирования в UNIX-like ОС

Макаров П. А.

## Содержание

<b>1</b>	<b>Задания для самостоятельной работы</b>	<b>1</b>
1.1	Консольный вывод в C++ средствами <code>stdio</code> . . . . .	1
1.2	Консольный вывод в C++ средствами <code>iostream</code> . . . . .	3
1.3	Консольный вывод в Python. Вариант 1. . . . .	3
1.4	Консольный вывод в Python. Вариант 2. . . . .	3
<b>2</b>	<b>Список источников</b>	<b>4</b>

## 1 Задания для самостоятельной работы

### 1.1 Консольный вывод в C++ средствами `stdio`

1. Напишите в текстовом редакторе `vim` файл `hello-1.cpp` с исходным текстом программы на языке C++, приведённым в Листинге 1:

---

```
1 /*
2     hello-1.cpp
3     Makarov P.A., 21.09.2022
4 */
5
6 #include <stdio>
7
8 int main(void) {
9     printf("Hello from stdio!\n");
10    return 0;
11 }
```

---

Листинг 1: Содержимое файла `hello-1.cpp`

- Исследуйте результат работы транслятора `g++` с помощью следующих команд:

```
$ g++ hello-1.cpp -o hello-1
$ ./hello-1
$ time ./hello-1
$ file hello-1.cpp
$ file hello-1
$ hexdump -C hello-1.cpp
$ readelf -a hello-1 | less
$ hexdump -C hello-1 | less
$ objdump -D hello-1 | less
```

При выполнении последних двух команд, кроме всего прочего, обратите внимание на содержимое файла `hello-1` в диапазоне адресов от `0x2004` до `0x2014`.

- Исследуйте результат работы компилятора `g++` с помощью следующих команд:

```
$ g++ -S hello-1.cpp -o hello-1.s
$ ls -l
$ file hello-1.s
$ less hello-1.s
```

Прочитайте справочную литературу о синтаксисе ассемблера AT&T. Попытайтесь понять содержимое файла `hello-1.s`. Подумайте и ответьте на следующие вопросы. Что такое компилятор и транслятор? Какие функции они выполняют? Какой термин более универсальный, а какой более специальный?

- С помощью следующих команд изучите работу стандартного ассемблера `as`, используемого транслятором `g++`:

```
$ g++ -c hello-1.cpp -o hello-1.o
$ ls -l
$ file hello-1.o
$ hexdump -C hello-1.o | less
$ objdump -D hello-1.o | less
```

- Подумайте и ответьте на следующие вопросы. Является ли ассемблирование окончательным этапом в процессе трансляции? Что именно делает ассемблер? Какова функция компоновщика (линкера)?

## 1.2 Консольный вывод в C++ средствами iostream

Напишите в текстовом редакторе vim файл `hello-2.cpp` с исходным текстом программы на языке C++, приведённым в Листинге 2. Чем этот текст принципиально отличается от примера, приведённого в Листинге 1? Проведите полноценное исследование данной программы аналогично предыдущему пункту.

---

```
1 /*
2     hello-2.cpp
3     Makarov P.A.
4     21.09.2022
5 */
6
7 #include <iostream>
8
9 using namespace std;
10
11 int main(void) {
12     cout << "Hello from iostream!" << endl;
13     return 0;
14 }
```

---

Листинг 2: Содержимое файла `hello-2.cpp`

## 1.3 Консольный вывод в Python. Вариант 1.

Напишите в текстовом редакторе vim файл `hello-1.py` с исходным текстом программы на языке Python, приведённым в Листинге 3.

---

```
1 print('Hello from python')
```

---

Листинг 3: Содержимое файла `hello-1.py`

Исследуйте данную программу, выполняя следующие команды. Разберитесь с тем, что происходит и сравните результаты с предыдущими пунктами.

```
$ file hello-1.py
$ ./hello-1.py
$ python hello-1.py
$ python3 hello-1.py
$ time python hello-1.py
$ time python3 hello-1.py
```

## 1.4 Консольный вывод в Python. Вариант 2.

Напишите в текстовом редакторе vim файл `hello-2.py` с исходным текстом программы на языке Python, приведённым в Листинге 4.

---

```
1 #!/usr/bin/python3
2
3 print('Hello from python')
```

---

#### Листинг 4: Содержимое файла `hello-2.py`

Исследуйте данную программу, выполняя следующие команды. Сравните результаты с предыдущими пунктами.

```
$ chmod +x hello-2.py
$ ls -l
$ file hello-2.py
$ ./hello-2.py
$ python3 hello-2.py
$ time ./hello-2.py
$ time python3 hello-2.py
```

## 2 Список источников

1. Столяров А.В. [Операционная среда ОС UNIX для изучающих программирование.](#)
2. Википедия. [GNU Compiler Collection.](#)
3. [GCC, the GNU Compiler Collection.](#)
4. Википедия. [GNU Assembler.](#)
5. Википедия. [GNU linker.](#)
6. <http://www.cplusplus.com>.
7. <https://www.python.org/doc/>.