

# 1 С-расширения

## 1.1 Часть 1. ctypes

В `sum1.c` запишем функцию для подсчёта суммы цифр числа:

```
1 unsigned digit_sum(unsigned x) {
2     unsigned ans = 0;
3     while (x != 0) {
4         ans += x % 10;
5         x /= 10;
6     }
7     return ans;
8 }
```

Соберём *shared library* из нашего исходника следующим образом:

```
gcc -O3 -shared -o sum1.so -fPIC sum1.c
```

В `py1.py` при помощи библиотеки *ctypes* загрузим нашу *shared library* и посчитаем сумму цифр чисел от 0 до 100 млн.:

```
1 from ctypes import CDLL
2
3
4 libc = CDLL("sum1.so")
5 digit_sum = libc.digit_sum
6
7
8 res = 0
9 for i in range(100000000):
10     res += digit_sum(i)
11
12 print(res)
```

Запустим и посмотрим на результаты:

```
yrustt@yrustt:~/Desktop/Projects/M17-18-SPCL/Lab_6$ time python3 py1.py
3600000000
```

```
real    0m44.089s
user    0m44.064s
sys     0m0.004s
```

## 1.2 Часть 2. CPython Extension Module

В `sum2module.c` определим функцию `digit_sum`.

Также в этом файле определяем массив структур *PyMethodDef* - функции, которые могут вызываться из *python*-кода. Флаг *METH\_VARARGS* указывает, что мы не будем передавать *keyword* аргументы в функцию при вызове, что позволяет нам воспользоваться функцией *PyArg\_ParseTuple*.

Также определим структуру *PyModuleDef*.

```
1 | static PyObject* digit_sum(PyObject *self, PyObject *args) {
2 |     int x, ans = 0;
3 |     if (!PyArg_ParseTuple(args, "i", &x)) return NULL;
4 |
5 |     while (x != 0) {
6 |         ans += x % 10;
7 |         x /= 10;
8 |     }
9 |
10 |    return Py_BuildValue("i", ans);
11 | }
12 | ...
```

В `setup.py` определим:

```
1 | from distutils.core import setup, Extension
2 |
3 | setup(
4 |     name='sum2',
5 |     version='0.0.1',
6 |     ext_modules=[Extension('sum2', ['sum2module.c'])]
7 | )
```

Чтобы получить *shared library* из нашего *c*-кода, запускаем:

```
python3 setup.py install
```

В `py2.py` импортируем наш модуль и посчитаем сумму цифр чисел от 0 до 100 млн.:

```
1 | import sum2
2 |
3 | res = 0
4 | for i in range(100000000):
5 |     res += sum2.digit_sum(i)
6 |
7 | print(res)
```

Запустим и посмотрим на результаты:

```
yrustrt@yrustrt:~/Desktop/Projects/M17-18-SPCL/Lab_7$ time python3 py2.py
3600000000
```

```
real    0m29.391s
user    0m29.388s
sys     0m0.000s
```

### 1.3 Бенчмарк

Method	Time (s)
Python	150.192
ctypes	44.089
CPython Extension Module	29.391