

# Proširivanje Pythona programskim jezicima C/C++

**Ivo Majić**

Mentor: Doc. dr. sc. Domagoj Jakobović

Sveučilište u Zagrebu  
Fakultet elektrotehnike i računarstva

Ak. god. 2011/12

- Uvod
- Python/C API
- Dodatni alati
- Ubrzanje

- <http://www.python.org/>
- različite implementacije
  - Jython (Java)
  - IronPython (C#)
  - CPython (C, referentna implementacija)
- zamišljen kao jako proširiv programski jezik
- dio standardne biblioteke (CPython) pisan u C-u

- definira određeni broj funkcija, makroa i varijabli
- jedan od ciljeva - ubrzanje konačnog programa
- datoteka zaglavlja

```
#include <Python.h>
```

Funkcija koju želimo pozivati u Pythonu:

```
int zbroji(int broj1, int broj2) {  
    return broj1+broj2;  
}
```

Omotač funkcija (eng. wrapper function):

```
static PyObject* w_zbroji(PyObject *self, PyObject *args) {  
    int broj1, broj2;  
    if (!PyArg_ParseTuple(args, "ii", &broj1, &broj2))  
        return NULL;  
    int rezultat = zbroji(broj1, broj2);  
    return Py_BuildValue("i", rezultat);  
}
```

Tablica: Format string znakovi

Znak	Opis
i	pretvara Python integer u C int
c	pretvara Python string duljine 1 u C char
f	pretvara Python float u C float
d	pretvara Python float u C double
s	pretvara Python string u niz C charova
O	vraća pokazivač na PyObject bez pretvorbe

- format znak 'O' - Python specifični tipovi podataka (PyObject, PyDictObject, PySetObject, ...)

Tablica: PyObject funkcije

Funkcija	Opis
PyObject_Check(PyObject *p)	vraća True ako je tipa PyObject
PyObject_Size(PyObject *list)	vraća duljinu liste
PyObject_GetItem(PyObject *list, int i)	vraća i-ti element liste

- popis funkcija dostupnih unutar proširenja (eng. module)

```
static PyMethodDef racunop_methods [] = {  
    {"zbroji", w_zbroji, METH_VARARGS, zbroji_doc},  
    {NULL, NULL, 0, NULL} /* Sentinel */  
};
```

- inicijalizacijska funkcija (format **initIMEMODULA()**)

```
PyMODINIT_FUNC initracunop(void) {  
    Py_InitModule("racunop", racunop_methods, racunop_doc);  
}
```

- prevođenje (python setup.py build)

```
from distutils.core import setup
from distutils.core import Extension

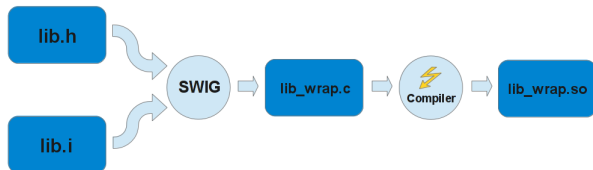
module = Extension("racunop", sources = ["racunop.c"])
setup(name="racunop", version="1.0", ext_modules=[module])
```

Slika: setup.py

- rezultat
  - racunop.so (Linux)
  - racunop.pyd (Windows)
- pozivanje unutar Pythona
  - import racunop



- SWIG
- Ctypes (Python proširenje)
- Cython



Slika: SWIG tok izvođenja

- alat za automatizirano generiranje poveznica (eng. bindings)
- <http://www.swig.org/>

```
#include <stdio.h>
#include " Brojevi.h"

Brojevi::Brojevi(int br1, int br2) {
    this->br1 = br1;
    this->br2 = br2;
}

void Brojevi::zbroj() {
    printf(" Zbroj je: %d\n", this->br1+this->br2);
}

void Brojevi::umnozak() {
    printf(" Umnozak je: %d\n", this->br1*this->br2);
}
```

Slika: Brojevi.cpp

# Dodatni alati - SWIG - primjer korištenja

- datoteka zaglavlja te konfiguracijska datoteka

```
class Brojevi {
    int br1;
    int br2;
public:
    Brojevi(int br1, int br2);
    void zbroj();
    void umnozak();
};
```

```
%module racunop
%{
#include "Brojevi.h"
%}
#include "Brojevi.h"
```

Slika: Brojevi.h

Slika: racunop.i

- prevođenje
  - `swig -python -c++ racunop.i`
  - `g++ -c racunop.cpp`
  - `g++ -I/usr/include/python2.7 -c racunop_wrap.cxx`
  - `ld -shared -o racunop.so racunop.o racunop_wrap.o libstdc++.so`

- pozivanje u Pythonu

```
>> import racunop
```

```
>> var = racunop.Brojevi(3,5)
```

```
>> var.zbroj()
```

```
Zbroj je: 8
```

```
>> var.umnozak()
```

```
Umnozak je: 15
```



Slika: Ctypes tok izvođenja

- dio Pythonove standardne biblioteke
- nema podrške za C++ klase
- podržani CDLL, OleDLL te WinDLL

```
import ctypes

biblioteka = ctypes.cdll.LoadLibrary("libc.so")

def sleep(sekundi):
    print "Spavam %d sekundi" % sekundi
    biblioteka.sleep(sekundi)

''' Testiranje '''
sleep(5)
```

Slika: Implementacija sleep() funkcije



Slika: Cython tok izvođenja

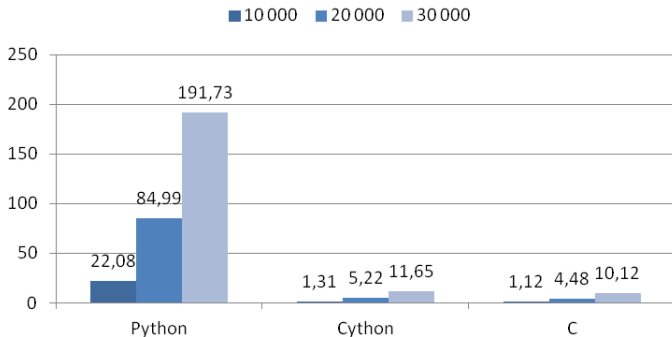
- <http://www.cython.org/>
- prevodioc koji proširuje sintaksu programskog jezika Python
- mogućnost statičkog definiranja C tipova podataka (**cdef** naredba)
- prevođenje korištenjem **distutils** proširenja



```
1  def prosti(int N):
2      cdef int n, k, i
3      cdef int p[N]
4      rezultat = []
5      k = 0
6      n = 2
7      while k < N:
8          i = 0
9          while i < k and n % p[i] != 0:
10             i = i + 1
11             if i == k:
12                 p[k] = n
13                 k = k + 1
14                 rezultat.append(n)
15             n = n + 1
16      return rezultat
```

Slika: prosti.pyx

## Trajanje izračuna prvih N prostih brojeva



Slika: Trajanje izračuna prvih N prostih brojeva (u sekundama)

- Hvala na pažnji!
- **Pitanja?**