Implementacao de Fractal de Mandelbrot em C e Python

Eduardo S. Porto, Miguel Carraro

A aplicação implementa o fractal de Mandelbrot, com a lógica de cálculo escrita em C e a interface de usuário desenvolvida em Python.

A integração entre as duas linguagens é realizada através da biblioteca `ctypes` do Python.

Para a utilização do “ctypes” o programa C é copilado como uma bibilioteca compartilhada. Para copilar o codigo desta maneira os parametros **shared** e **-fPIC** são associados a copilação pelo GCC. Pyhton carrega esta bibilioteca utilizando **ctypes.CDLL('./fractal_mandel.so')**.

O programa Python cria um array com as dimensões especificadas (largura e altura) e o tipo de dados “float64”, que é equivalente ao tipo “double” em C.

result = np.zeros((colum, line), dtype=np.float64)

O array NumPy é convertido em um ponteiro de “double” para ser compatível com o código C:

result_ptr = result.ctypes.data_as(ctypes.POINTER(ctypes.c_double))

A função `mandelbrot` na biblioteca C é chamada a partir do Python, passando os parâmetros necessários e o ponteiro para o array:

loaded_library.mandelbrot(line, colum, limit_of_iterations, result_ptr)

O programa python invoca a função o programa em C.

fractal = mandelbrot_c_program(line, colum, limit_of_iterations)

Que acessa a capacidade de performance do programa em C.
Cria um array com largura e altura especificadas, preenchido com zeros, do tipo float64 (equivalente a double em C)

result = np.zeros((colum, line), dtype=np.float64)

Converte o array numpy para um ponteiro de double em C usando ctypes

result_ptr = result.ctypes.data_as(ctypes.POINTER(ctypes.c_double))

Chama a função 'mandelbrot' na biblioteca C, passando os parâmetros e o ponteiro para o array

loaded_library.main(line, colum, limit_of_iterations, result_ptr)