

Programmazione Avanzata e parallela

Lezione 19

Python e C

Ritorno delle librerie condivise

- Come possiamo chiamare C da Python?
- La libreria ctypes
- Convertire gli argomenti
- Specificare i tipi di ritorno

Chiamare C da Python

Problemi

- È possibile chiamare una libreria condivisa C a partire da Python
- Serve prestare attenzione a:
 - Il tipo degli argomenti
 - Il tipo di ritorno delle funzioni
 - Chi gestisce la memoria

Chiamare C da Python

Problemi di tipo

- Il tipo “int” in Python non corrisponde al tipo “int” in C
 - In Python il tipo ha precisione arbitraria, quindi un numero di bit di rappresentazione non fissato...
 - ...mentre in C ha un numero fissato di bit (32/64)
- I char in C sono pensati per rappresentare la codifica ASCII...
- ... ma Python un carattere può essere un simbolo in Unicode e quindi occupare più di un byte)

Chiamare C da Python

Libreria ctypes

- La libreria ctypes è inclusa nelle librerie standard di Python
- Permette di accedere a librerie condivise scritte in C
- Consente di effettuare conversione di tipo tra i tipi di C e i tipi di Python
- Permette di specificare il tipo degli argomenti delle funzioni C...
- ...e il tipo di ritorno

Chiamare C da Python

Conversione di tipo

Tipo C	Tipo in ctypes	Tipo Python
int	c_int	int
unsigned int	c_uint	int
float	c_float	float
double	c_double	float
char	c_char	Oggetto di tipo “bytes” con un solo carattere
size_t	c_size_t	int

Chiamare C da Python

Libreria condivisa

- Ricordate che per è possibile compilare un file come libreria condivisa con le opzioni -fPIC e -shared del compilatore
- `gcc -fPIC -shared -o library.so library.c`
- Servirà poi caricare la libreria da Python usando ctypes:
 - **`ctypes.cdll.LoadLibrary("library.so")`**
- Il valore ritornato è un oggetto che ci permette di accedere alla funzioni definite nella libreria

Chiamare C da Python

Chiamare funzioni

- Supponiamo di avere una funzione **sum** che prende due interi come argomento e ritorna un intero
- Supponendo che “code” sia l’oggetto creato caricando da libreria con ctypes possiamo semplicemente chiamare:
 - `code.sum(3, 4)`
 - Questo ci fornisce il risultato corretto
- E se il tipo non fosse un intero?

Chiamare C da Python

Specificare il tipo degli argomenti

- Ogni funzione C ha nell'oggetto creato da ctypes un attributo "argtypes" che permette di fornire una lista di tipi di argomenti.
 - Esempio:
`code.multiply.argtypes = [ctypes.c_int, ctypes.c_float]`
- Se il tipo Python è convertibile nei tipi ctypes indicati da "argtypes" allora la conversione avviene in automatico
- Similmente è necessario specificare il tipo di ritorno di una funzione:
 - Esempio: **`code.multiply.restype = ctypes.c_float`**

Chiamare C da Python

Puntatori

- Sui può ottenere il puntatore a un oggetto con tipo fornito da ctypes (e.g., c_int) con “pointer()”:
 - Esempio:
a = ctypes.c_int(4)
pa = pointer(a)
- Il tipo corrispondente (da usare se si specificano gli argomenti delle funzioni) è **POINTER(c_int)**