

15.7 从C扩展中释放全局锁🔒

问题🔒

你想让C扩展代码和Python解释器中的其他进程一起正确的执行，那么你就需要去释放并重新获取全局解释器锁（GIL）。

解决方案🔒

在C扩展代码中，GIL可以通过在代码中插入下面这样的宏来释放和重新获取：

```
#include "Python.h"
...

PyObject *pyfunc(PyObject *self, PyObject *args) {
    ...
    Py_BEGIN_ALLOW_THREADS
    // Threaded C code. Must not use Python API functions
    ...
    Py_END_ALLOW_THREADS
    ...
    return result;
}
```

讨论🔒

只有当你确保没有Python C API函数在C中执行的时候你才能安全的释放GIL。GIL需要被释放的常见的场景是在计算密集型代码中需要在C数组上执行计算（比如在numpy中）或者是要执行阻塞的I/O操作时（比如在一个文件描述符上读取或写入时）。

当GIL被释放后，其他Python线程才被允许在解释器中执行。 `Py_END_ALLOW_THREADS` 宏会阻塞执行直到调用线程重新获取了GIL。