

Modern cpp 30 lectures — coroutines

Friday, December 25, 2020

4:34 PM

什么是协程:

```
def fibonacci():  
    a = 0  
    b = 1  
    while True:  
        yield b  
        a, b = b, a + b
```

yield = 生成器, generator, 返回一个可迭代的对象, 每次迭代就能得到一个 yield 出来的结果. 这是一种很常见的协程形式

使用这个生成器:

```
for i in islice(fibonacci(), 20):  
    print(i)
```

} 打印头 20 项.

```
for i in takewhile(  
    lambda x: x < 10000,  
    fibonacci()):  
    print(i)
```

} 打印 小于 10000 的数列项.

- islice 取一个范围的头若干项.
- takewhile 则在范围中逐项取出内容, 直到第 1 个参数的条件不能被满足.
- islice 和 takewhile 的结果都可被看作是 C++ 中的视图.

C++20 协程.

常见用途

- 生成器
- 异步 I/O
- 惰性求值
- 事件驱动应用.

关键字: co_await, co_yield, co_return.

三个关键字只要有一个出现在函数里, 这个函数就是一个协程了.

用协程实现的 fibonacci:

```
uint64_t resumable fibonacci()  
{  
    uint64_t a = 0;  
    uint64_t b = 1;  
    while(true) {  
        co_yield b;  
        auto tmp = a;  
        a = b;  
        b += tmp;  
    }  
}
```

调用:

```
auto res = fibonacci();  
while (res.resume()) {  
    auto i = res.get();  
    if (i >= 10000) {  
        break;  
    }  
    cout << i << endl;  
}
```

C++20 协程的高层抽象.

cppcoro 库, 高层接口包含 generator, fibonacci 协程可以这样实现.

```
#include <cppcoro/generator.hpp>  
using cppcoro::generator;
```

```
generator<uint64_t> fibonacci()  
{  
    uint64_t a = 0;  
    uint64_t b = 1;  
  
    while(true) {  
        co_yield b;  
        auto tmp = a;  
        a = b;  
        b += tmp;  
    }  
}
```

使用 fibonacci:

```
for (auto i : fibonacci()) {  
    if (i >= 10000) {  
        break;  
    }  
    cout << i << endl;  
}
```