Python 进阶训练营

尹会生

7 多进程与多线程

目　录

CONTENTS

学习目标

1.掌握异常捕获的原理和用方法

1.掌握进程、线程、锁的概念

1.多进程与多线程模块介绍

掌握 threading 模块与锁机制

掌握上下文管理器与协程

掌握 asyncio 并发处理的原理与使用方法

进程与线程的概念

为什么计算机程序可以并行
并行与并发有什么差别
全局解释器锁 GIL 对 Python 多线程有哪些影响

多进程模块 multiprocessing 详解
多线程模块 threading 详解
进程与线程的同步机制
队列机制
进程池与线程池

进程相关的模块

子进程模块 os.fork()

多进程模块 multiprocessing

一般用法：

```
from multiprocessing import Process

join([timeout])
```

进程相关的模块

面向对象的进程创建方法
- 继承 Process 类的 run() 方法

multiprocessing 的两种通信机制
- 队列 Queue
- 管道 Pipe

数据共享
- 共享内存 Value 和 Array

进程锁

进程其他模块

服务器进程

Manager

进程池

multiprocessing.pool

注意死锁问题

线程模块

多进程和多线程的区别

多线程模块 threading 详解

GIL 问题

python 的线程安全是伪指令级的

线程的其他相关知识

线程同步机制	阻塞线程
锁机制	互斥锁 可重入锁 条件锁
守护线程	
定时器	

多进程还是多线程

CPU 密集	I/O 密集
偏重计算	偏重输入输出 包括网络 I/O
建议使用多进程	建议使用多线程

迭代器

迭代器与可迭代的区别
利用 yield 实现生成器
itertools 库的用法

生成器-1

1. 在函数中使用 yield 关键字，可以实现生成器。
2. 生成器可以让函数返回可迭代对象。
3. yield 和 return 不同，return 返回后，函数状态终止，yield 保持函数的执行状态，返回后，函数回到之前保存的状态继续执行。
4. 函数被 yield 会暂停，局部变量也会被保存。
5. 迭代器终止时，会抛出 StopIteration 异常。

生成器-2

```
print([ i for i in range(0,11)])
```

替换为

```
print(( i for i in range(0,11)))
```

```
gennumber = ( i for i in range(0,11))
print(next(gennumber))
print(next(gennumber))
print(next(gennumber))
# print(list(gennumber))
print([i for i in gennumber ])
```

协程

协程和线程的区别是什么
yield/send 与yield from 作为协程如何使用
asyncio 模块
asyncio.coroutine 和 yield from 的关系
事件循环机制

THANKS!