传智播客

《Python 程序开发案例教程》

教学设计

课程名称：Python 程序开发案例教程

授课年级：2019 年级

授课学期：2019 学年第一学期

教师姓名：某某老师

2019 年 09 月 09 日

课题名称	第 13 章 进程和线程	计划学时	5 学时
内容分析	在计算机多核的环境下，为了能充分地利用 CPU 以提高执行的效率，Python 提供了两种常见的多任务编程的方式，分别为进程和线程。		
教学目标及基本要求	1. 了解什么是进程和线程 2. 掌握创建进程的几种方式 3. 掌握进程通信的原理，会使用 Queue 类实现进程间通信 4. 掌握线程的基本操作 5. 掌握线程中锁的使用 6. 理解同步机制，会使用 Condition 和 Queue 实现线程同步		
教学重点	1. 通过 Process 类创建进程 2. 通过 Pool 类批量创建进程 3. 线程的阻塞 4. 互斥锁 5. 可重入锁		
教学难点	1. 通过 Pool 类批量创建进程 2. 进程间通信——Queue 3. 互斥锁 4. 死锁 5. 可重入锁 6. 通过 Condition 类实现线程同步 7. 通过 Queue 类实现线程同步		
教学方式	教学采用教师课堂讲授为主，使用教学 PPT 讲解		

教学过程

第一课时

（什么是进程、通过 fork()函数创建进程、通过 Process 类创建进程、通过 Pool 类批量创建进程）

一、创设情境，引出文件的打开与关闭操作

（1） 教师通过提出需求，引出什么是进程。

（2） 明确学习目标

- 要求学生了解什么是进程
- 要求学生掌握通过 fork()函数创建进程
- 要求学生掌握 Process 类创建进程
- 要求学生掌握 Pool 类批量创建进程

二、进行重点知识的讲解

（1） 教师根据课件，讲述什么是进程

程序是一个没有生命的实体，它包含许多由程序设计语言编写的、但未被执行的指令，这些指令经过编译和执行才能完成指定动作。程序被执行后成为了一个活动的实体，这个实体就是进程。

（2） 教师根据课件，讲述通过 fork()函数创建进程。

在 Unix/linux 操作系统中，通过 Python 的 os 模块中封装的 fork()函数可以轻松创建一个进程，其语法格式为：fork()，当调用 fork()函数后，操作系统会创建当前进程的副本以实现进程的创建，此时原有的进程被称为父进程，复制的进程被称为子进程。

（3） 教师根据课件，讲述通过 Process 类创建进程。

multiprocessing 模块提供的 Process 类可通过两种方式创建子进程：一种是直接通过 Process 类创建子进程，另一种方式是通过 Process 子类创建子进程。

（4） 教师根据课件，讲述通过 Pool 类批量创建进程。

若创建的进程数量不多，可以直接使用 Process 类创建多个子进程。但是有时需要操作多个文件目录，或者远程控制多态计算机，这时对进程数量的需求会非常大，手动地创建多个进程的方式显然不可取的，不仅低效繁琐，而且工作量巨大。因此多进程模块 multprocessing 中提供了 Pool 类，可以批量创建子进程。

其语法格式为：

Pool(processes=None,initializer=None,initargs=(),maxtasksperchild =None,context=None)。

三、归纳总结，布置作业/随堂练习

（1） 回顾上课前的学习目标，并对本节课的内容进行总结。

教师总结本节课需要掌握的知识点，包括什么是进程、通过 fork()函数创建进程、通过 Process 类创建进程、通过 Pool 类批量创建进程。

（2） 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

（3） 使用博学谷系统下发课后作业。

第二课时

（进程间通信——Queue、什么是线程、线程的创建和启动、线程的阻塞）

一、回顾上节课的内容，继续讲解本课时的知识

（1） 教师对学生们的疑问进行统一答疑。

（2） 回顾总结上节课内容，继续介绍本课时的内容。

上节课我们学习了什么是进程、进程的三种创建方式，本节课将带领大家学习进程间通信——Queue、什么是线程、线程的创建和启动、线程的阻塞。

（3） 明确学习目标

- 要求学生掌握进程间通信——Queue
- 要求学生了解什么线程
- 要求学生掌握线程的创建和启动
- 要求学生掌握线程的阻塞

二、进行重点知识的讲解

（1） 教师根据课件，讲解进程间通信——Queue 的使用。

每个进程中所拥有的数据都是独有的，无法与其它进程共享。但大多数进程之间需要进行通信。例如，所有的子进程执行完任务之后，通知处于阻塞转态的主进程继续向下执行。为此，Python 的 mulitprocessing 模块中提供了能实现进程间的通信 Queue 类，该类用于创建和管理存储共享资源的队列。

（2） 教师根据课件，讲解什么是线程。

线程是系统进行运算调度的最小单位，也被称为轻量级进程，它包含在进程之中，是进程的实际运作单位。进程中可以包含多个线程，每个线程是进程中单一顺序的控制流，可以并行执行不同的任务。

（3） 教师根据课件，讲解线程的创建和启动。

模块 threading 中定义了 Thread 类，该类专门用于管理线程。线程的创建方式分为使用 Thread 类创建和 Thread 子类创建两种（启动详见 13.5.1）。

（4） 教师根据课件，讲解线程的阻塞。

线程在执行的过程中，会因为等待某个条件的触发进入则塞状态，例如，控制台阻塞等待接收用户的输入。为了避免线程处于无休止的阻塞态，可以为其指定超时时长。通过 join()方法可以等待其它线程的结束或指定等待的时长。

三、归纳总结，布置作业

（1） 回顾学习目标，总结本节课所学知识包括：进程间通信——Queue、什么是线程、线程的创建和启动、线程的阻塞。

（2） 布置随堂练习，检查学生掌握情况。

根据博学谷和随堂练习资源，给学生布置随堂练习，检测学生的掌握程度，并对学生出现的问题进行解决。

（3） 使用博学谷系统下发课后作业。

第三课时

（互斥锁、死锁、可重入锁）

一、回顾上节课内容，继续介绍本课时的内容

（1） 教师对学生们的疑问进行统一答疑。

（2） 教师通过提问学生问题，由上一课时引出本课时要讲解的内容。

（3） 明确学习目标

- 要求学生掌握互斥锁
- 要求学生掌握死锁
- 要求学生掌握可重入锁

二、进行重点知识的讲解

（1） 教师根据课件，讲解互斥锁。

互斥锁是最简单的加锁技术，它只有两种状态：锁定和非锁定。当某个线程

需要更改共享数据时，它会先对共享数据上锁，将当前的资源转换为“锁定”状态，其它线程无法对被锁定的共享数据进行修改；当前程序执行结束后，它会解锁共享数据，将资源转换为“非锁定”状态，以便其它线程可以对资源上锁后进行修改。

（2） 教师根据课件，讲解死锁。

死锁是指两个以上的线程在执行过程中，由于各自持有一部分共有资源或者彼此通信而造成一种阻塞的现象。若没有外力作用，线程们将无法继续执行，一直处于阻塞状态。在使用 Lock 对象给资源加锁时，若操作不当很容易造成死锁，常见的不当行为主要包括上锁与解锁的次数不匹配和两个线程格子持有一部分共享资源。

（3） 教师根据课件，讲解可重入锁。

为了避免因同一线程多次使用互斥锁造成的死锁，threading 模块中提供了 RLock 类。RLock 类代表可重入锁，它允许同一线程多次锁定和多次释放。

三、归纳总结，布置作业

（1） 回顾学习目标，总结本节课所学知识包括：单选按钮 Radiobutton、列表框 List、文本域 Text、pack 布局管理器。

（2） 使用博学谷系统下发课后作业。

第四课时

（线程同步、通过 Condition 类实现线程同步、通过 Queue 类实现线程同步、实例 1：生成者与消费者模式）

一、回顾上节课内容，继续介绍本课时的内容

（1） 教师对学生们的疑问进行统一答疑。

（2） 教师通过提问学生问题，由上一课时引出本课时要讲解的内容。

（3） 明确学习目标

- 要求学生了解线程同步
- 要求学生了解通过 Condition 类实现线程同步
- 要求学生了解通过 Queue 类实现线程同步
- 要求学生了解实例 1：生产者与消费者模式的实现过程

二、进行重点知识的讲解

（1） 教师根据课件，讲解什么是线程同步。

线程按预定的次序执行称为线程的同步。例如，线程 1 执行完任务 1，之后线程 2 执行任务 2…，threading 模块中提供的 Condition 类和 queue 模块中的 Queue 类能实现线程的同步。

（2） 教师根据课件，讲解 Condition 类实现线程同步。

Condition 类代表条件变量，它允许线程在触发某些事件或达到特定条件后才开始执行。通过 Condition 类提供的构造方法可以创建一个实例。

（3） 教师根据课件，讲解通过 Queue 类实现线程同步。

Queue 类表示一个 FIFO（先进先出）队列，即先插入的队列中的数据先获取，用于多个线程之间的信息传递。创建队列的方式比较简单，其语法格式为：Queue(maxsize=0)。

（4） 教师根据课件，讲解实例 1：生产者与消费者模式。

教师根据教学资源实现实例 1：生产者与消费者模式，并向学生讲解实现过程。

三、归纳总结，布置作业

（1） 回顾学习目标，总结本节课所学知识包括：线程同步、通过 Condition 类实现线程同步、通过 Queue 类实现线程同步、实例 1：生产者消费者模式。

（2） 使用博学谷系统下发课后作业。

第五课时
（上机练习）

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察同学对知识点的掌握情况，对代码的熟练程度。

上机一：（练习教材示例代码以及实例 1：生产者与消费者模式）

形式：单独完成

要求：

(1) 要求学生能够熟练掌握教材中示例代码。

(2) 要求学生能够自己实现实例 1 程序。

思考题和习题	见教材第 13 章配套的习题
教学后记	