**计算机科学系**

**CS240操作系统、通信和并发**

**实际的10号**

在实践中，你使用java套接字解决方案实现了一个简单的客户机/服务器应用程序版本。客户端与服务器通信以获取日期。

在实践7、8和9中，你为生产者消费者问题、餐饮哲学家问题和读者/作者问题实现了多线程模拟

使用这些实践中的代码和经验，重新访问客户端/服务器日期应用程序(实用的基于6套接字的解决方案- DateServer.java &文件java)，并进行必要的修改，使服务器成为多线程的。服务器应该能够接受来自客户机的多个连接，并同时向各自的客户机返回日期值。

注意，不需要更改客户机代码，只需更改服务器代码。你需要改变的主要方法后的服务器代码,以便接受客户端套接字连接,而不是进行处理请求,应该创建一个新的worker线程和run()方法的新工作线程应该包含的代码处理客户端请求和关闭连接。

工作线程通过向客户端发送日期来实现服务，而不是像实际中那样使用main()方法。main()方法只是侦听和接受连接并生成工作线程。您需要将连接的客户端套接字对象作为参数传递给工作线程的构造函数。

main()方法还应该记录已建立的连接的数量，并将每个连接的数量传递给工作线程的构造函数，这样它就可以在响应客户端后将“完成处理客户端n”这样的消息打印到输出流。

您将需要创建一个WorkerThread类(基于早期实践中提供的模板)。显然，WorkerThread的行为(正如在run()方法中定义的那样)需要修改，以实现上述实际6中DateServer类中给出的部分功能。最简单的方法是删除run()方法和线程类的构造函数中的所有代码，这些线程类是在以前的一个实验中创建的，并从实用6的DateServer.java的main()方法中粘贴所需的部分

修改。

尽管这个示例的应用程序功能相当简单，但涉及通信和多线程的应用程序在概念上对程序员来说很难。您可以使用这个参考框架向您介绍java多线程服务器，并帮助将来构造更复杂的分布式多线程代码。

**在MOODLE提交**

一个文本文件，包含:—

您创建的WorkerThread类

修改的DateServer类，它生成WorkerThreads来处理客户端连接。

这是你们这门课最后一次作业。我希望你喜欢这些实验室，并学到一些新东西。