

考试时间：120 分钟

考试答案检查：所有源代码必须在考试结束前在网络学堂完成提交。考试结束后开始考试代码的现场演示，检查代码版本以此为准，迟交代码酌情扣分。

代码提交方式：在本地机器上以自己的“学号”为名称建立目录，在此目录中为每个题目建立子目录（如 1、2 ……），将答案源码拷贝到相应目录中，再将整个目录压缩成 zip 或 rar 文件，通过网络学堂上传答案。请确保上载内容正确，责任自负。

1. 文件传输程序

用 C/S 模型进行网络传输文件程序的开发。客户端、服务器端均用 Qt 创建窗口界面。要求如下：

- 首先，服务器端直接进入“等待连接”的状态，客户端启动并与服务器建立连接。
- 客户端有 2 个标签，2 个 QLineEdit 输入框，1 个“传输”按钮，1 个 TextEdit。两个标签分别写有“文件原位置”和“目标文件夹”；两个输入框分别指定服务器端保存的需要传送文件的路径及文件名，传到客户端保存该文件的文件夹。
- 服务器端 1 个 TextEdit 部件，用于显示服务器端保存的需要传送文件的路径及文件名。
- 点击客户端的传送按钮，服务器端显示传送文件名及路径，客户端开始从服务器端获取文件，并通过客户端的 TextEdit 部件展示接收进度，每 16KB 汇报一次，如输出“已接收 16KB，共计**MB”、“已接收 128KB，共计**MB”。如果服务器找不到文件，则返回一个错误信息，客户端显示在 TextEdit 中。要求这些信息，不是递增，而是反复显示在同一行。
- 因为测试文件较大，且为可执行文件，客户端在内存中收到文件并进行整合，保存到指定文件夹位置。
- 一个文件传输结束了，还可以启动新的传输任务。

评分标准：

- (1) 正确建立连接，客户端可以正确指定两个路径（1 分）
- (2) 服务器端显示路径及文件名，成功传输文件（1 分）
- (3) 客户端文件传输进度展示正确（1 分）
- (4) 文件传输到新位置内容不变（1 分）
- (5) 支持多次启动新的传输任务（1 分）

2019-9-2 日

2. 多阶段流水线的模拟

程序设计中的流水线模式是指将每个计算任务 task 分成多个阶段，每个线程负责一个阶段的功能实现，并将当前阶段的计算结果通过任务队列传递给下一个线程。

当一个线程需要将自己完成的结果传递给另一个线程时，利用队列来实现：当前线程作为“生产者”，将任务结构 task 放入队列尾，下一个线程作为“消费者”，从队列头取出该任务结构 task，并进行相应的操作。

队列的实现可以设计为一个 task 结构数组+队头/队尾变量，在多个“生产者/消费者”的情况下，队头/队尾变量还需要锁保护。如果“生产者”发现队列已满，则需要阻塞等待队列有空闲位置再向队尾放入任务；同理，如果“消费者”发现队列已空，则需要阻塞等待队列有任务再从队头取任务来做。

本题目共需要三个线程进行协作，下面给出三个线程需要执行的伪代码。

函数 1：（对应线程 1）

输入：int 类型 a, b

操作：

打印开始 log

计算 $c=\text{gcd}(a, b)$, $d=\text{lcm}(a, b)$ 最大公约数 最小公倍数

Sleep 1 秒钟

打印完成 log

输出：int 类型 c, d

函数 2：（对应线程 2）

输入：int 类型 a, b

操作：

打印开始 log

计算 $c=a+b$, $d=a*b$

Sleep 2 秒钟

打印完成 log

输出：int 类型 c, d

函数 3：（对应线程 3）

输入：int 类型 a, b

操作：

打印开始 log

计算 $c=a+b$, $d=b-a$

2019-9-2 日

Sleep 3 秒钟
打印完成 log
输出: int 类型 c, d

要求界面有一个“开始”按钮，每一个线程对应一个 TextEdit，用于显示该线程输出的 log 信息，log 信息的输出要求以追加的形式。点击开始按钮，线程 1 从 input.txt 中依次读入任务，并且执行函数 1 的操作，执行完后将执行结果放入队列 1；线程 2 从队列 1 中取出任务，执行函数 2，并把结果放入队列 2；线程 3 从队列 2 中取出任务，执行函数 3，并把结果显示在 log 中。

input.txt 中每一行代表一个任务，共 3 个数：任务编号，任务内容相关的两个数。

开始 log 的格式为 “[时间戳] Task 任务编号 start a b（任务中的两个输入）”

结束 log 的格式为 “[时间戳] Task 任务编号 complete c d（任务中的两个输出）”

例如 [22:56:41.412] Task 3 complete 1 30

提示：本题中任务结构是 Task 编号和两个 int，可能需要的一些类或函数：QDateTime, QThread::sleep()。

评分标准：

- (1) 正确绘制开始按钮和三个 log 输出区 (1 分)
- (2) 三个线程能按照格式输出任务开始和结束的 log (1 分)
- (3) 三个线程输出 log 时有明显的间隔，能体现流水线特点，与时间戳吻合 (2 分)
- (4) 线程 1/2/3 输出的结果正确 (1 分)

注：若输出结果正确但不满足 (3)，则 (4) 不得分

3 支持并发的键值存储程序开发

所谓 key-value 键值存储就是一个 key 对应一个 value，key 是关键词，一般在系统中不重复，value 是值。本题要求大家实现一个 C/S 模型的键值存储系统：客户端用 QTcpSocket 开发一个发送操作文件，接收操作结果并展示的工具；服务器端用 QTcpServer、QTcpSocket 开发一个接收操作文件、依据文件对键值系统进行相应操作、返回操作结果给客户端的服务器。

(1) 客户端模拟三个用户，每个用户有 2 个按钮，1 个 textEdit。1 个“选择”按钮用于选择操作文件的位置，1 个“开始”按钮用于确认发送该操作文件。textEdit 用于显示操作结果。

(2) 用户在客户端通过点击“选择”按钮用 QFileDialog 指定操作文件的位置，并点击对应的“开始”按钮，此时该按钮 disable，对应结果窗口清空；客户端为该请求生成一个 QTcpSocket 并与 Server 建立连接，发送操作文件给服务器（如果不重新选择文件，则以上一次选择的文件为准，不用每次点按钮之前都得重新选择文件）；

(3) 对应刚才的操作结果返回后，textEdit 显示返回的操作结果，并将对应的操作按钮 enable。

(4) 三个用户可以不断提交操作文件给服务器来处理，所以服务器必须采用多线程编程。

(5) 服务器端进行监听，当 1 个客户端连接请求到来后，创建线程来进行文件接收，并根据客户端发来的操作文件对 key-value 进行处理；

对 key-value 类型及两种操作说明如下：

key 为 int，value 为 string

get a: 获取 key 为 a 所对应的 value，该操作需要返回结果 value 作为一行打印在客户端相应的 TextEdit 中。如果 key 为空，则返回字符串 null。

put a b: 将 key-value 系统中 key 为 a 的 value 设置为 b，该操作不需要返回结果。

(6) 服务器端界面有一个 textEdit 部件，每接收到一个操作，就将操作以追加的形式输出在 textEdit 中；

(7) 操作完成后，该线程返回操作结果给客户端；

(8) 客户端接到返回结果后，对应连接关闭，服务端对应线程结束；

提示：该题目在服务器端的实现可能用到互斥锁，多线程对于维护 key-value 的数据结构以及某些部件的访问是并发的，不加锁可能导致错误，简单的处理方法是在服务器端线程在操作 key-value 之前加锁，执行完操作文件后解锁。

2019-9-2 日

评分标准：

- (1) 正确绘制客户端/服务器端界面。（1 分）
- (2) 客户端点击 1 个按钮，两端能正确显示结果。（1 分）
- (3) 客户端同时点击多个按钮，对应部件能正确显示返回的操作结果。（2 分）
- (4) 客户端同时点击多个按钮，服务器端能展示已经执行的操作。（1 分）