

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、学生信息远程录入系统：

用 C/S 模型进行网络通讯程序开发，客户端、服务器端均用 Qt 创建对话框界面，在客户端用主窗口录入学生信息、传输给服务器端在信息窗口展示并存成服务器端文件。

- 客户端录入信息窗口包括如下四个“label”和“输入窗口”对：姓名、学号、性别、成绩。客户端进行初步的合法性检查：姓名不超过 10 个字符；学号为 10 位数字；性别为单选组合框，用户选择男、女；成绩在 0-100 之间。
- 当客户端点击“提交”按钮时，对应的事件处理函数向服务端以 UDP 包的形式发送一条学生信息。
- 服务器端接到“客户端发来的学生信息”时，则相应地在信息窗口（TextEdit 部件）展示并以追加写的方式保存成本地文件（不同字段之间用“，”，不同学生信息之间用“\r\n”）。

每一次传输，有 4 个变量，分别表示姓名、学号、性别、成绩。如果服务器已经接到过具有相同学号的学生信息，则在信息窗口提示，且不存到文件；否则，在信息窗口展示并存成服务器端文件。

总分：30 分

- （1） 客户端、服务器端的图形绘制，得 5 分。
- （2） 要求使用 QUdpSocket 完成客户端和服务端数据传输，并在信息窗口展示，得 15 分；
- （3） 客户端的输入合法性检查、服务器端的重复判定，得 5 分；
- （4） 服务器端可以写入文本文件，得 5 分。

## 2、蒙特卡洛模拟的并程序序：

问题背景：假定系统里有  $k$  块磁盘， $n$  块数据块，且  $n \gg k$ 。为了将任一数据块放入磁盘，给它生成一个随机数  $r$ ，则将该数据放入磁盘  $r \bmod k$  上。为了验证这种方法放置数据是否可以保证各个磁盘上放置的数据块总数是不是大体均等的，使用蒙特卡洛模拟的方法，依次生成  $n$  个随机数，来计算  $k$  个盘上的数据块个数，并给出最后的标准方差。

假定随机数序列已经生成并存在一个大文件里。现在要求一次性将该文件读入内存数组 `array`，并且：

(1) 将该内存平均分成 4 段，主线程启动 4 个子线程，每个子线程负责处理其中一段。

(2) 每个子线程将依据随机数对  $k$  取模的值，将对应磁盘的数据个数加 1。这样，形成 `B[i][j]` ( $i$  为子线程逻辑编号， $j$  为模  $k$  的余数) 的  $4*k$  大小的数组。

(3) 都完成步骤 2 之后，主线程进行合并操作。它计算出 4 个线程对应某个磁盘的数据块个数总和，并计算出这  $k$  个数的标准差。

评分标准 (30 分)：

- (1) 正确读取输入文件、用 `QThread` 创建 4 个线程 (输出线程 ID)，得 10 分；
- (2) 多线程正确完成子计算任务，每个线程正确输出各自统计结果，得 10 分；
- (3) 主线程等待所有子线程运行结束，正确得到归并结果并显示，得 10 分；

注：每个子线程在进行计算时，输出处理的前 3 个数对  $\langle i, j \rangle$ ， $i$  代表线程 ID， $j$  代表随机数对  $k$  取模结果。该输出将作为 (2) 的部分评分依据。

样例文件有  $n$  行，每行包含一个 `int` 型的数。

### 3、具有统计功能的云计算系统：

模仿云计算系统的传输和计算功能，用 C/S 模型进行网络通讯程序开发，客户端用 QTcpSocket 开发一个文件发送、指定计算函数、接收计算结果的工具，服务器端用 QTcpServer、QTcpSocket 开发一个文件接收、对该文件内容进行指定统计工作、返回计算结果的服务器。

- (1) 客户端使用 QFileDialog 来选择发送的待计算文件的位置。下面有 4 个“按钮”和 textEdit 对，按钮分别代表 min/max/average/median 四种运算，按钮下面的 textEdit 显示计算结果。
- (2) 用户在客户端通过 QFileDialog 指定待计算文件的位置，并点击 4 个按钮中的一个，此时按钮 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，发送文件连同对应的操作给服务器（要求不重新选择文件的情况下以上一次 QFileDialog 选择的文件为准，不要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 显示返回的计算结果，并将对应的操作按钮 enable。
- (4) 上述四种操作可以不断提交给服务器来运算，所以服务器必须采用多线程来编程。
- (5) 服务器端在 2018 端口进行监听，当 1 个客户端连接请求到来后，创建线程进行文件接收，并根据客户端发来的指示调用对应的计算操作；
- (6) 计算完成后，该线程返回计算结果给客户端；
- (7) 客户端接到返回结果后，对应的连接关闭，服务端线程结束；
- (8) 服务器支持多客户端、每个客户端支持多个操作请求同时进行；
- (9) 文件传输之前，需要传输头信息，头信息包括：操作类型、文件大小；
- (10) 文件格式为，每行一个数。
- (11) debug 信息打印两端多线程的线程 id 作为评分依据。

#### 评分标准（40 分）

- (1) 实现客户端界面，能够用 QFileDialog 来选择上传文件，按钮按下能够触发对应函数。（10 分）
- (2) 能够正确传输文件内容，并得到正确计算结果。（10 分）
- (3) 服务器端能够用多线程技术正确执行客户端请求的操作。（10 分）
- (4) 能够实现正确的并发逻辑，也就是说：单个客户端可以同时发送多个计算请求。（10 分）