

Linux程序开发：QT的内部进程通信（1）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E7_A8_8B_E5_BA_c103_144885.htm

Qt 作为一种跨平台的基于 C 的 GUI 系统，能够提供给用户构造图形用户界面的强大功能。自从 1996 年 Qt 被 Trolltech 公司发布以来，该系统成为世界上很多成功的图形用户应用所使用的主要系统。更为重要的是，Linux 操作系统的桌面环境系统 KDE 也是基于 Qt 构造的。目前，Qt 已经提供了对包括 MS/Windows、Unix/X11 和嵌入式平台的支持，得到了越来越广泛的应用。在 Qt 系统中，不仅有着构造完善的系统结构，而且为了满足用户对编写图形用户界面应用的种种需求，它还创建了许多新的系统机制，其中 Qt 所特有的内部进程通信机制尤其值得一提。本文分析了基于 QT 的应用进程之间通信常用的三种机制

：QCOP 协议，Signal-Slot 机制和 FIFO 机制。给出了各自的使用方法，并指出了各自的使用场合。

1、QCOP 协议

QCOP 是 Qt 内部的一种通信协议，这种协议用于不同的客户之间在同一地址空间内部或者不同的进程之间的通信。目前，这种机制还只在 Qt 的嵌入式版本中提供。为实现这种通信机制，Qt 中包括了由 QObject 类继承而来的 QCopChannel 类，该类提供了诸如 send()、isRegistered() 等静态函数，它们可以在脱离对象的情况下使用。为了在 channel 中接收通信数据，用户需要构造一个 QCopChannel 的子类并提供 receive() 函数的重载函数，或者利用 connect() 函数与接收到的信号相联系。值得一提的是，在 Qt 系统中，只提供了 QCOP 协议机制和用于接收消息的类，而如何发送消息则没有提供相应

的类供用户使用。在基于 Qt 的桌面系统 Qtopia (QPE) 中，则提供了相应的发送类：QCopEnvelope。用户可以通过该类利用 channel 向其他进程发送消息。该类将通过 QCopChannel 发送 QCop 消息的过程进行了封装，用户只需要调用该类中的相关函数就可以方便地实现进程之间的通信过程。一方面，QCop 消息的发送要利用 QCopEnvelope 类，另一方面，接收消息则是通过与一个 QCopChannel 相关联。在发送消息时，将利用如下的协议机制：QCopEnvelope e(channelname, messagename)。对于需要携带参数的消息，必须使用" e 对于不带参数的消息，只需要利用：QCopEnvelope e(channelname, messagename)。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com