

二、信号和槽机制

信号槽是Qt框架引以为豪的机制之一。所谓信号槽，实际就是观察者模式。当某个事件发生之后，比如，按钮检测到自己被点击了一下，它就会发出一个信号(signal)。这种发出是没有目的的，类似广播。如果有对象对这个信号感兴趣，它就会使用连接(connect)函数，意思是，将想要处理的信号和自己的一个函数(称为槽(slot))绑定来处理这个信号也就是说，当信号发出时，被连接的槽函数会自动被回调。

也就是说，信号槽的本质就是将一个可以发出信号的信号函数与另一个要被调用的槽函数进行链接(connect)后，实现通过某一事件，导致信号发出，来执行另一个函数的功能。

2.1 信号和槽

某个自定义的类想使用信号槽，这个类MainWindow就必须继承 QObject (或者是他的祖先类有继承QObject) 而且在类内第一行必须有 Q_OBJECT // 信号槽，宏定义
QObject 基本上是所有类的基类。

而在MainWindow类中
MainWindow类继承了 QMainWindow类，而QMainWindow类继承了QWidget类
而QWidget类继承了QObject类

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    QPushButton bnt(&w); // 有参构造，传入parent参数，保证新的按钮在 mianwindow窗口中，
    bnt.show();
    bnt.move(100,100);

    // 某个自定义的类想使用信号槽，这个类MainWindow就必须继承 QObject (或者是他的祖先类有继承QObject)
    // 而且在类内第一行必须有 Q_OBJECT // 信号槽，宏定义
    // QObject 基本上是所有类的基类。
    QPushButton bnt1(&w);
    bnt1.show();
    bnt1.move(200,200);
    bnt1.setText("退出");

    // 对象链接（绑定）
    // 参数：发送信号的对象，发送的信号，接收信号的对象，接收信号后对象要执行的函数
    QObject::connect(&bnt1,&QPushButton::clicked,&w,&MainWindow::close);

    return a.exec(); // 循环队列，接收事件
}
```

connect 函数最常用的一般形式：

```
connect(sender, signal, receiver, slot);
```

参数：

sender：发出信号的对象

signal：发送对象发出的信号

receiver：接收信号的对象

slot: 接收对象在接收到信号之后所需要调用的函数

2.2 自定义信号槽

使用connect函数可以让我们连接系统提供的信号和槽。但是，Qt的信号槽机制并不仅仅是使用系统提供的那部分，还会允许我们自己设计自己的信号和槽。

报社和读者

有一个报纸售卖员类 Newspaper，有一个订阅者类Reader。Reader 可以订阅报纸。这样当报纸有了新的内容的时候，Reader可以立即得到通知。

```
main.c
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    // 定义两个对象
    Newspaper n;
    Reader r;
    // 链接对象，newspaper信号，链接reader函数。
    QObject::connect(&n, &Newspaper::newspaper, &r, &Reader::reader);
    n.sendSignals(); // 手动发送信号
    // 创建一个按钮
    QPushButton bnt1(&w);
    bnt1.show();
    // 让按钮的点击信号，链接newspaper信号，而newspaper信号又链接了 reader函数
    // 因此，点击按钮即可打印收到
    QObject::connect(&bnt1, &QPushButton::clicked, &n, &Newspaper::newspaper);
    return a.exec(); // 循环队列，接收事件
}
```

首先看 Newspaper 这个类。这个类继承了QObject 类。只有继承了 QObject 类的类，才具有信号槽的能力。所以，为了使用信号必须继承 QObject。(不管是 QObject的直接子类还是间接子类)，都应该在**第一行代码写上 Q_OBJECT**，不管是不是使用信号槽，都应该添加这个宏。

Newspaper 类的public和private 代码块都比较简单，只不过它新加了一个 signals。signals 块所列出的，就是该类的信号。信号就是一个一个的函数名，返回值是 void(因为无法获得信号的返回值，所以也就无需返回任何值)，参数是该类需要让外界知道的数据。信号作为函数名，不需要在 cpp 函数中添加任何实现

Reader 类更简单。因为这个类需要接受信号，所以我们将其继承了 QObject，并且添加了Q_OBJECT 宏。后面则是默认构造函数和一个普通的成员函数。Qt5中，任何成员函数、static函数、全局函数和 Lambda 表达式都可以作为槽函数。与信号函数不同，槽函数必须自己完成实现代码。槽函数就是普通的成员函数，因此作为成员函数也会受到 public、private 等访问控制符的影响。(如果信号是 private 的这个信号就不能在类的外面连接，也就没有任何意义。)

信号槽要求信号和槽的参数一致，所谓一致，是参数类型一致。如果不一致，允许的情况是，槽函数的参数可以比信号的少，即便如此，槽函数存在的那些参数的顺序也必须和信号的前面几个一致起来。这是因为，你可以在槽函数中选择忽略信号传来的数据(也就是槽函数的参数比信号的少)，但是不能说信号根本没有这个数据，你就要在槽函数中使用(就是槽函数的参数比信号的多，这是不允许的)。

2.3 自定义信号槽的注意事项

- (1) 发送者和接收者都需要是QObject的子类(当然，槽函数是全局函数、Lambda表达式等无需接收者的时候除外)；
- (2) 使用 signals 标记信号函数，信号是一个函数声明，返回 void，不需要实现函数代码；
- (3) 槽函数是普通的成员函数，作为成员函数，会受到public、private、protected 的影响；

- (4) 使用emit 在恰当的位置发送信号;
- (5) 使用QObject::connect函数连接信号和槽任何成员函数、static 函数、全局函数和 Lambda 表达式都可以作为槽函数
- (6) 信号与槽函数的参数个数和类型需要一致, 至少信号的参数不能少于槽函数的参数。

2.4 信号槽的更多用法

一个信号可以和多个槽相连

如果是这种情况, 这些槽会一个接一个的被调用, 但是它们的调用顺序是不确定的。

多个信号可以连接到一个槽

只要任意一个信号发出, 这个槽就会被调用

一个信号可以连接到另外的一个信号

当第一个信号发出时, 第二个信号被发出。除此之外, 这种信号信号的形式和信号-槽的形式没有什么区别。

槽可以被取消链接

这种情况并不经常出现, 因为当一个对象 delete之后, Qt自动取消所有连接到这个对象上面的槽

使用 Lambda 表达式

在使用 Qt5 的时候, 能够支持 Qt5 的编译器都是支持 Lambda 表达式的。在连接信号和槽的时候, 槽函数可以使用 Lambda表达式的方式进行处理。

2.5 Lambda 表达式