

## 6 信号和槽

### 6.1 什么是信号和槽?

信号和槽是Qt自行定义的一种通信机制，实现对象间的数据交互。当用户或系统触发了一个动作，导致某个组件的状态发生了改变，该组件就会发射一个信号，即调用其类中一个特定的成员函数，同时还可能携带必要的参数。槽和普通成员函数几乎没有太多区别，可以是公有、保护或私有的，可以被重载，也可以被覆盖，其参数可以是任意类型，并可以像普通成员函数一样被调用。槽函数与普通成员函数的差别并不在于其语法特性，而在于其功能。槽函数更多体现为对某种特定信号的处理，可以将槽和其它对象的信号建立连接。当发射信号时，槽函数将被触发和执行，进而实现特定的功能。

### 6.2 如何定义信号?

定义信号类似下面这个样子：

```
1 class XXX: public QObject
2 {
3     Q_OBJECT
4
5     signals:
6     void sigfunc(...); // 信号函数
7 };
```

信号函数只需声明，不要定义。

### 6.3 如何定义槽?

定义槽类似下面这个样子：

```
1 class XXX: public QObject
2 {
3     Q_OBJECT
4
5     public slots:
6     void slotfunc(...) // 槽函数
7     {
8         ...
9     }
10 };
```

槽函数可被连接到某个信号上。信号发射时，槽函数被执行。槽函数可以被直接调用。

### 6.4 如何连接信号和槽?

用于连接信号和槽的connect方法：

```

1 QObject::Connection QObject::connect(
2     const QObject* sender, // 信号发送者
3     const char* signal, // 信号函数
4     const QObject* receiver, // 信号接收者
5     const char* method, // 槽函数
6     Qt::ConnectionType type = Qt::AutoConnection
7 );

```

## 6.5 案例

点击按钮关闭窗口。

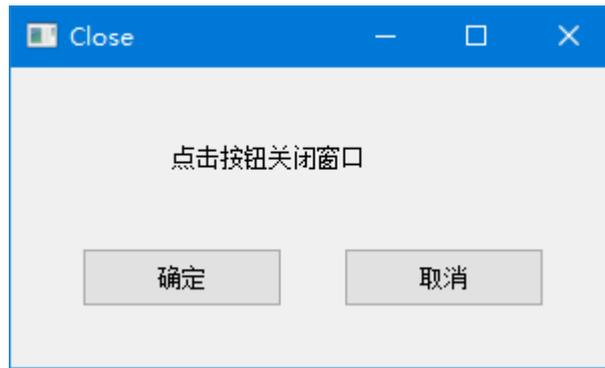
C:\Users\Minwei\Projects\Qt\Close\main.cpp:

```

1 #include <QApplication>
2 #include <QWidget>
3 #include <QLabel>
4 #include <QPushButton>
5
6 int main(int argc, char* argv[])
7 {
8     // 创建应用对象
9     QApplication app(argc, argv);
10
11     // 创建父窗口
12     QWidget parent;
13     parent.move(500, 400); // 设置父窗口的位置
14     parent.resize(300, 150); // 设置父窗口的大小
15
16     // 创建标签组件
17     QLabel label("点击按钮关闭窗口", &parent);
18     label.move(80, 30); // 设置子窗口的位置
19     label.resize(140, 30); // 设置子窗口的大小
20
21     // 创建确定按钮
22     QPushButton btnOk("确定", &parent);
23     btnOk.move(35, 90); // 设置子窗口的位置
24     btnOk.resize(100, 30); // 设置子窗口的大小
25
26     // 创建取消按钮
27     QPushButton* btnCancel = new QPushButton("取消", &parent);
28     btnCancel->move(165, 90); // 设置子窗口的位置
29     btnCancel->resize(100, 30); // 设置子窗口的大小
30
31     // 连接信号和槽
32     QObject::connect(&btnOk, &QPushButton::clicked,
33         &parent, &QWidget::close);
34     QObject::connect(btnCancel, SIGNAL(clicked()),
35         &parent, SLOT(close()));
36
37     // 显示父窗口
38     parent.show();
39
40     // 进入事件循环
41     return app.exec();

```

运行效果如图所示：



## 6.6 信号和槽的语法要求

原则上，信号函数和槽函数的参数表应严格一致，比如：

```
1 QObject::connect(sender, SIGNAL(sigfunc(int)),
2   receiver, SLOT(slotfunc(int)));
```

但有两个例外。一个是当槽函数的参数带有缺省值时，比如：

```
1 QObject::connect(sender, SIGNAL(sigfunc(int)),
2   receiver, SLOT(slotfunc(int,int=0)));
```

另一个是当信号函数的参数多于槽函数时，比如：

```
1 QObject::connect(sender, SIGNAL(sigfunc(int,int)),
2   receiver, SLOT(slotfunc(int)));
```

这时，多余参数将被忽略。

## 6.7 信号和槽的特殊用法

一般情况下，将一个信号连接到一个槽，已可以满足大多数应用开发的需要。但有时我们也需要将一个信号连接到多个槽，即信号广播，比如：

```
1 QObject::connect(sender, SIGNAL(sigfunc()),
2   receiver1, SLOT(slotfunc1()));
3 QObject::connect(sender, SIGNAL(sigfunc()),
4   receiver2, SLOT(slotfunc2()));
```

或者将多个信号连接到一个槽，即信号汇集，比如：

```
1 QObject::connect(sender1, SIGNAL(sigfunc1()),
2   receiver, SLOT(slotfunc()));
3 QObject::connect(sender2, SIGNAL(sigfunc2()),
4   receiver, SLOT(slotfunc()));
```

甚至将一个信号再连接到另一个信号，即信号级联，比如：

```
1 QObject::connect(sender1, SIGNAL(sigfunc1()),
2     sender2, SIGNAL(sigfunc2()));
```