

高级网工实战之旅

60个网络实战案例

¥99  
原价153

- 2021薛大龙软考报名入口
- Linux工程师10问10答
- 考华为认证，做高级网工
- 网络安全攻防实战系统班

## Qt 事件处理机制（下篇）

在Qt中，事件被封装成一个个对象，所有的事件均继承自抽象类QEvent。接下来依次谈谈Qt中有谁来产生、分发、接受和处理事件。

作者：佚名 来源：互联网 | 2011-07-01 14:20 收藏 分享

继续我们上一篇文章继续介绍，[Qt 事件处理机制（上篇）](#)介绍了Qt框架的事件处理机制：事件的产生、分发、接受和处理，并以视窗系统鼠标点击QWidget为例，对代码进行了剖析，向大家分析了Qt框架如何通过Event Loop处理进入处理消息队列循环，如何一步一步委派给平台相关的函数获取、打包用户输入事件交给视窗系统处理，函数调用栈如下：

```
1. main(int, char **)
2. QApplication::exec()
3. QCoreApplication::exec()
4. QEventLoop::exec(ProcessEventsFlags )
5. QEventLoop::processEvents(ProcessEventsFlags )
6. QEventDispatcherWin32::processEvents(QEventLoop::ProcessEventsFlags)
```

本文将介绍Qt app在视窗系统回调后，事件又是怎么一步步通过QApplication分发给最终事件的接受和处理者QWidget::event，（QWidget继承Object,重载其虚函数event），以下所有的讨论都将嵌入在源码之中。

```
1. QT_WIN_CALLBACK QtWndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM
2. bool QApplicationPrivate::sendMouseEvent(...)
3. inline bool QCoreApplication::sendSpontaneousEvent(QObject *receiver, QEvent *event)
4. bool QCoreApplication::notifyInternal(QObject *receiver, QEvent *event)
5. bool QApplication::notify(QObject *receiver, QEvent *e)
6. bool QApplicationPrivate::notify_helper(QObject *receiver, QEvent * e)
7. bool QWidget::event(QEvent *event)
8.
9. // (续上文Section 7) Section 2-1：
10. QT_WIN_CALLBACK QtWndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM
11. {
12. ...
13. //检查message是否属于Qt可转义的鼠标事件
14. if (qt_is_translatable_mouse_event(message)) {
15.     if (QApplication::activePopupWidget() != 0) {
16.         POINT curPos = msg.pt;
17.         //取得鼠标点击坐标所在的QWidget指针，它指向我们在main创建的widget实例
18.         QWidget* w = QApplication::widgetAt(curPos.x, curPos.y);
19.         if (w)
20.             widget = (QETWidget*)w;
21.     }
22.     if (!qt_tabletChokeMouse) {
23.         //对，就在这里。Windows的回调函数将鼠标事件分发回给了Qt Widget
24.         // => Section 2-2
25.         result = widget->translateMouseEvent(msg);
26.     }
27. }
28. // Section 2-2 $QTDIR\src\gui\kernel\application_win.cpp
29. //该函数所在与Windows平台相关，主要职责就是把已windows格式打包的鼠标事件解包、翻译成
30. bool QETWidget::translateMouseEvent(const MSG &msg)
```

精培学院

51CTO旗下精品培训平台

- 行业标杆课程体系
- 三对一辅导服务
- 为效果付费

立即进入

### 编辑推荐

头条	沉浸式故事：AR和VR如何改变2021年的市场营销？
热点	iOS版微信8.0.3更新了什么 朋友圈和自定义表情突破限制
热点	微信一口气更新了 12 个功能
聚焦	合理规划：如何为APP选择正确的数据库？
关注	为什么手机厂商越来越少做白色前面板的手机呢？

### 24H热文 一周话题 本月最赞

- iOS 14.5新功能大盘点：个个都是绝技
- 从安卓设备转移数据到iOS的几种方法，掌握...
- iPhone 12价格走势：2个月降10%，5个月...
- 推荐两款iOS端磁力下载工具
- 轻松找到微信接收文件存储位置
- 快捷指令怎么用？玩转iOS14快捷指令全攻略
- 一招搞定手机和电脑的多屏协同
- 合理规划：如何为APP选择正确的数据库？

### 订阅专栏

+更多

Redis运维秘籍


运维标配技术  
共15章 | one叶孤舟

39人订阅学习

```

31. {
32.     //.. 这里很长的代码给以忽略
33.     // 让我们看一下sendMouseEvent的声明
34.     // widget是事件的接受者; e是封装好的QMouseEvent
35.     // ==> Section 2-3
36.     res = QApplicationPrivate::sendMouseEvent(widget, &e, alienWidget, this, &qt_button_down
37. }
38. // Section 2-3 $QTDIR\src\gui\kernel\application.cpp
39. bool QApplicationPrivate::sendMouseEvent(QWidget *receiver, QMouseEvent *event,
40.     QWidget *alienWidget, QWidget *nativeWidget,
41.     QWidget **buttonDown, QPointer<QWidget> &lastMouseReceiver,
42.     bool spontaneous)
43. {
44.     //至此与平台相关代码处理完毕
45.     //MouseEvent默认的发送方式是spontaneous, 所以将执行sendSpontaneousEvent。 sendSp
46.     除了将QEvent的属性spontaneous标记不同。 这里是解释什么spontaneous事件: 如
47.     显然MousePressEvent是由视窗系统产生的一个的事件 (详见上文Section 1~ Section 7
48.
49.     if (spontaneous)
50.         result = QApplication::sendSpontaneousEvent(receiver, event); ==> Section 2-4
51.     else
52.         result = QApplication::sendEvent(receiver, event);
53. }
54. // (续上文Section 7) Section 2-1:
55. QT_WIN_CALLBACK QtWndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM
56. {
57.     ...
58.     //检查message是否属于Qt可转义的鼠标事件
59.     if (qt_is_translatable_mouse_event(message)) {
60.         if (QApplication::activePopupWidget() != 0) {
61.             POINT curPos = msg.pt;
62.             //取得鼠标点击坐标所在的QWidget指针, 它指向我们在main创建的widget实例
63.             QWidget* w = QApplication::widgetAt(curPos.x, curPos.y);
64.             if (w)
65.                 widget = (QETWidget*)w;
66.         }
67.         if (!qt_tabletChokeMouse) {
68.             //对, 就在这里。Windows的回调函数将鼠标事件分发回给了Qt Widget
69.             // ==> Section 2-2
70.             result = widget->translateMouseEvent(msg);
71.         }
72.     }
73.     // Section 2-2 $QTDIR\src\gui\kernel\application_win.cpp
74.     //该函数所在与Windows平台相关, 主要职责就是把已windows格式打包的鼠标事件解包、翻译
75.     bool QETWidget::translateMouseEvent(const MSG &msg)
76.     {
77.         //.. 这里很长的代码给以忽略
78.         // 让我们看一下sendMouseEvent的声明
79.         // widget是事件的接受者; e是封装好的QMouseEvent
80.         // ==> Section 2-3
81.         res = QApplicationPrivate::sendMouseEvent(widget, &e, alienWidget, this, &qt_button_down
82.     }
83.     // Section 2-3 $QTDIR\src\gui\kernel\application.cpp
84.     bool QApplicationPrivate::sendMouseEvent(QWidget *receiver, QMouseEvent *event,
85.         QWidget *alienWidget, QWidget *nativeWidget,
86.         QWidget **buttonDown, QPointer<QWidget> &lastMouseReceiver,
87.         bool spontaneous)
88.     {
89.         //至此与平台相关代码处理完毕
90.         //MouseEvent默认的发送方式是spontaneous, 所以将执行sendSpontaneousEvent。 sendSp
91.         除了将QEvent的属性spontaneous标记不同。 这里是解释什么spontaneous事件
92.         显然MousePressEvent是由视窗系统产生的一个的事件 (详见上文Section 1~ Secti
93.         if (spontaneous)
94.             result = QApplication::sendSpontaneousEvent(receiver, event); ==> Section 2-4
95.         else
96.             result = QApplication::sendEvent(receiver, event);

```



活学活用

**Ubuntu Server**

实战直通车  
共35章 | UbuntuServer

233人订阅学习



**Java EE**

速成指南

掌握Java核心  
共30章 | 51CTO王波

89人订阅学习

**51CTO学院** 新用户免费领VIP月度会员>>



企业云计算架构设计之存储架构设计



Nutanix 超融合基础架构设计指南

VMware vSAN超融合基础架构设计

## CTO品牌

+ 更多

CTO训练营第九季招募中

技术经理研习营2021年招募

### CTO训练营

申请入营

互联网班

体验营

### 技术经理

申请加入

能力地图

进化图谱

## 专题推荐

+更多

产品不行 技术来撑, 看苹果如何打响技术战

产品

不谈架构谈应用! 开源时代你将如何“起飞”?

移动

基于React与Vue后, 移动开源项目Weex如何定义未来

React



```

97. }

1. // Section 2-4 C:\Qt\4.7.1-Vs\src\corelib\kernel\qcoreapplication.h
2. inline bool QCoreApplication::sendSpontaneousEvent(QObject *receiver, QEvent *event)
3. {
4.     //将event标记为自发事件
5.     //进一步调用 2-5 QCoreApplication::notifyInternal
6.     if (event) event->spont = true; return self ? self->notifyInternal(receiver, event) : false;
7. }
8. // Section 2-5: $QTDIR\gui\kernel\qapplication.cpp
9. bool QCoreApplication::notifyInternal(QObject *receiver, QEvent *event)
10. {
11.
12.     // 几行代码对于Qt Jambi (QT Java绑定版本) 和QSA (QT Script for Application)的支持
13.     ...
14.     // 以下代码主要意图为Qt强制事件只能发送给当前线程里的对象, 也就是说receiver->d_fur
15.         注意, 跨线程的事件需要借助Event Loop来派发
16.     QObjectPrivate *d = receiver->d_func();
17.     QThreadData *threadData = d->threadData;
18.     ++threadData->loopLevel;
19.     bool returnValue;
20.     QT_TRY {
21.         //哇, 终于来到大名鼎鼎的函数QCoreApplication::nofity()了 ==> Section 2-6
22.         returnValue = notify(receiver, event);
23.     } QT_CATCH (...) {
24.         --threadData->loopLevel;
25.         QT_RETHROW;
26.     }
27. }
28. // Section 2-6: $QTDIR\gui\kernel\qapplication.cpp
29. // QCoreApplication::notify和它的重载函数QApplication::notify在Qt的派发过程中起到核心的作用
30.     任何线程的任何对象的所有事件在发送时都会调用notify函数。
31. bool QApplication::notify(QObject *receiver, QEvent *e)
32. {
33.     //代码很长, 最主要的是一个大大的Switch,Case
34.     ..
35.     switch ( e->type())
36.     {
37.         ...
38.         case QEvent::MouseButtonPress:
39.         case QEvent::MouseButtonRelease:
40.         case QEvent::MouseButtonDblClick:
41.         case QEvent::MouseMove:
42.         ...
43.         //让自己私有类(d是私有类的句柄) 来进一步处理 ==> Section 2-7
44.         res = d->notify_helper(w, w == receiver ? mouse : &me);
45.         e->spont = false;
46.         break;
47.     }
48.     ...
49. }
50. // Section 2-7: $QTDIR\gui\kernel\qapplication.cpp
51. bool QApplicationPrivate::notify_helper(QObject *receiver, QEvent *e)
52. {
53.     ...
54.     // 向事件过滤器发送该事件, 这里介绍一下Event Filters. 事件过滤器是一个接受即将发送给目
55.     //如代码所示它开始处理事件在目标对象行动之前。过滤器的QObject::eventFilter () 实现被调
56.         允许或者拒绝事件的更进一步的处理。如果所有的事件过滤器允许更进一步的事
57.         如果他们中的一个停止处理, 目标和任何后来的事件过滤器不能看到任何事件。
58.     if (sendThroughObjectEventFilters(receiver, e))
59.         return true;
60.     // 递交事件给receiver => Section 2-8
61.     bool consumed = receiver->event(e);
62.     e->spont = false;
63. }

```



看看16年,移动开发都发生了什么

移动

精选博文 论坛热帖 下载排行

读书

+更多



Cisco网络技术教程 (第2版)

本书作为思科认证体系中的入门级教材, 主要讲述了网络的基本知识和思科设备的基本命令, 以及路由、交换等深层次网络知识的入门知识, 其体系...



订阅51CTO邮刊

点击这里查看样刊

立即订阅



51CTO服务号



51CTO播客

```

64. // Section 2-8 $QTDIR\gui\kernel\qwidget.cpp
65. // QApplication通过notify及其私有类notify_helper,将事件最终派发给了QObject的子类- QWidget
66. bool QWidget::event(QEvent *event)
67. {
68.     ...
69.     switch(event->type()) {
70.     case QEvent::MouseButtonPress:
71.         // Don't reset input context here. Whether reset or not is
72.         // a responsibility of input method. reset() will be
73.         // called by mouseHandler() of input method if necessary
74.         // via mousePressEvent() of text widgets.
75.         #if 0
76.             resetInputContext();
77.         #endif
78.         //mousePressEvent是虚函数, QWidget的子类可以通过重载重新定义mousePressEvent事件的行
79.         mousePressEvent((QMouseEvent*)event);
80.         break;
81.     }

```

小结: **Qt 事件处理机制** (下篇) 的内容介绍完了, 希望本文对你 有所帮助! 更多相关资料请参考编辑推荐!

#### 【编辑推荐】

1. [Qt For Symbian截获程序前后台切换事件](#)
2. [Qt 多线程之逐线程事件循环 下篇](#)
3. [详解 QT 源码之 Qt 事件机制原理](#)
4. [QT源码之Qt信号槽机制与事件机制的联系](#)
5. [详解 Qt 事件过滤器](#)

【责任编辑: [李程站](#) TEL: (010) 68476606】

点赞 2

Qt

事件

分享:



## 大家都在看 猜你喜欢



值得推荐的五款免费网络漏洞扫描器



将在2021年颠覆业务发展的十种技术



Angular、React与Vue, 那个框架更好?



今天我才发现Redis有7种数据类型...

## 好课推荐



企业云计算架构设计之存储架构设计



Nutanix 超融合基础架构设计指南



VMware vSAN超融合基础架构设计



备战2021软考--系统架构设计师视频课程专题

