<https://blog.csdn.net/luotuo44/article/details/38300779>

**转载请注明出处:**[**http://blog.csdn.net/luotuo44/article/details/38300779**](http://blog.csdn.net/luotuo44/article/details/38300779)

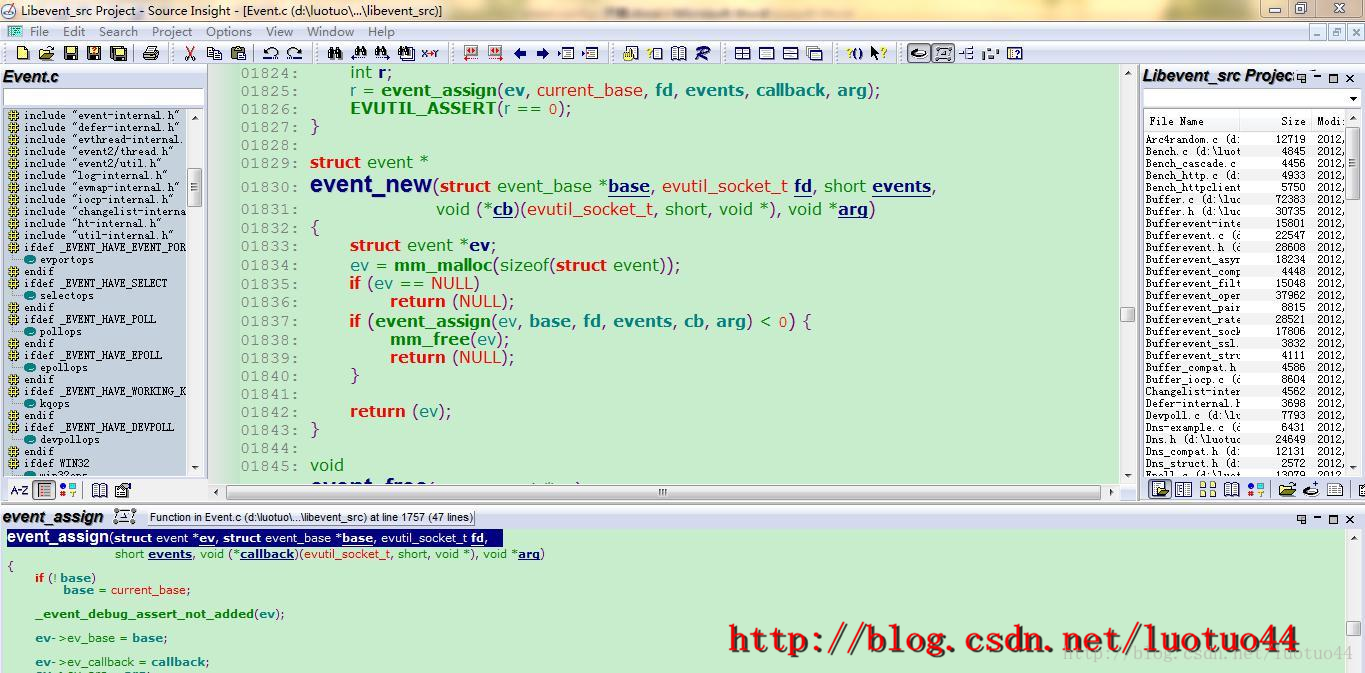
        我所分析的Libevent版本是2.0.21版本，是目前最新的稳定版本。看这系列博文中，需要读者有Linux编程的一些基础。因为像POSIX、文件描述符、多线程等等这些概念，我并不会去解释，我默认读者已经熟悉这些概念了。如果读者读过《UNIX环境高级编程》，那就完全没问题了。

        因为Libevent是跨平台的，所以它使用了很多它自己定义的通用跨平台类型，比如evutil\_socket\_t。此外，Libevent也定义了一些跨平台通用的API，这些都可以在《[通用类型和函数](http://blog.csdn.net/luotuo44/article/details/38780157)》一文中找到。

        相信来看本系列的文章的读者，都不会是刚刚接触Libevent的用户。这里就不说Libevent的优点和怎么安装使用Libevent了。我是想介绍其他东西。

# Source Insight：

        这个工具是阅读代码的神器。下面是一个截图。



        正中央是代码窗口。在代码中，如果你想看一下event\_assign函数的内部是怎么实现的，那么你不用找到event\_assign函数实现文件，然后打开，再Ctrl + F查找。在Source Insight中，你只需用鼠标单击一下event\_assign函数。那么就会在下面的那个窗口显示event\_assign函数的具体实现。是不是很厉害的功能？

        右边的窗口是文件列表，和其他IDE的功能差不多。

        左边那个窗口功能也是很强大的，特殊是当代码中出现了很多条件宏。在这样的条件下，这个宏会被定义成这样。在那样的条件下，又会定义成那样。左边的窗口可以清楚地看到。

# Doxgen工具：

        这是一个可以制作chm文件的工具。

        在Source Insight中，虽然是很容易追踪到某个函数的具体实现(实现都在c文件中，非头文件)。但Libevent的源文件中很少有注释，也没有这个函数的解释。而且Source Insight无法追踪到函数的声明，只能追踪到定义(就是函数的实现)（或许是我对Source Insight还不熟悉，如果有这样的功能，还望大家指出）。

        而由Doxgen工具生成的chm文件是离线把的帮助文档，它会列出函数的说明、参数和返回值。

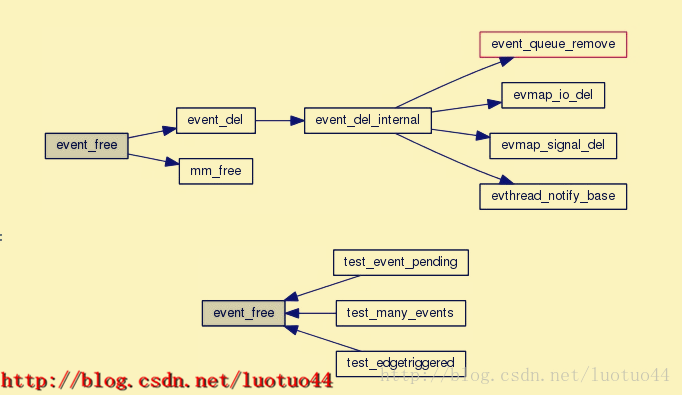
        怎么用Doxgen工具制作Libevent的离线版帮助文档，可以参考<http://blog.chinaunix.net/uid-12773189-id-3022656.html>

# do{}while(0)宏：

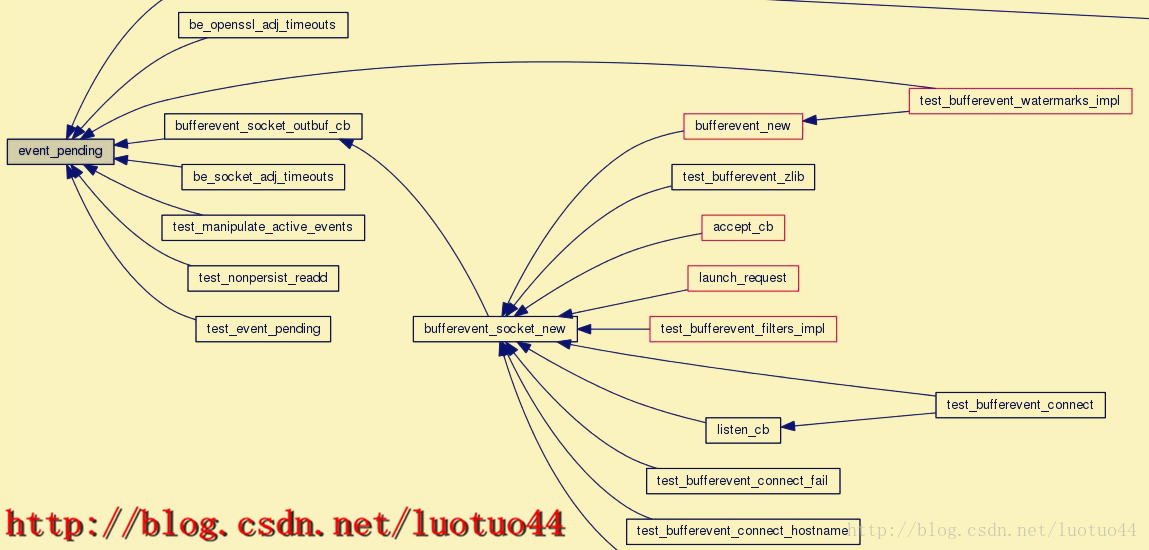
        在Libevent的源代码中，经常能看到do{}while(0)宏的使用。如果是第一次碰到这种写法，估计都会比较不解。可以参考<http://www.cnblogs.com/lanxuezaipiao/p/3535674.html>来解惑。

# 函数调用关系图：

        有时候追踪一个函数，想知道该函数的调用关系图。[有一个网站](http://sourcecodebrowser.com/libevent/2.0.3-alpha/evbuffer-internal_8h.html)提供了这个关系图。下面举两个例子。



        图中，上面的是event\_free函数会调用哪些函数，一颗主调用树。下面的是哪些函数会调用event\_free函数，是被调用关系。其中test\_event\_pending这些是Libevent提供的测试例子的测试函数。下面再给另外一个被调用关系的图：



        有一个不足之处，这个网站并没有和Libevent同步更新，目前提供的最高Libevent版本是2.0.3-alpha

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/luotuo44/article/details/38300779