More from Boost

- Asynchronous Input and Output
- Accessing the File System
- Loosely Coupled Components
- ... and some place for more ...

Boost: Asio

The purpose of Boost.Asio is to support event-driven program designs by providing a framework to dispatch incoming events to previously registered event handlers:

- The largest group of events deals with I/O, especially the arrival of data from asynchronous sources (like sockets).
- Besides that it allows also an applications to send events to itself, maybe with a specified delay.

Event handlers are run single-threaded and hence there is no need for synchronisation as is in multi-threaded designs. For good responsiveness an event driven program design must keep event handlers small.

> Especially an event handler should **never** delay or wait for responses of an external client – rather register an handler to be started when the response arrives.

Boost: File System

Even with C++11 there is no portable way to access the file system to

- search through directories and sub-directories,
- determine and change file properties,
- delete, rename, link, or copy files.

> Boost had tried to tackle this since a long time – with more or less success – and is currently in its 3rd major release Boost.Filesystem V3.

This file system library may also – via the TR2 Path – become available with recent Standard C++ versions and is part of Microsoft Visual Studio 2013.

*: At the time of writing the final state of affairs is not quite clear. A major obstacle through all the years seems to have been uniting the classic and also modern 8-bit-char API of Unix/Linux (using UTF-8 now, which the clients – by and large – can handle "content-agnostic") with the 16-bit-char API of MS-Windows in a portable way ...

Signals2

From the Boost Documentation:

> The Boost.Signal Library is an implementation of a managed signals and slots system. Signals represent callbacks with multiple targets, and are also called publishers or events in similar systems. Signals are connected to some set of slots, which are callback receivers (also called event targets or subscribers), which are called when the signal is "emitted."