Bretny Khamphavong

TinyOS Summarization

TinyOS is a free and open source component-based operating system and platform targeting wireless sensor networks (WSNs). This type of hardware platform enforces certain constraints on any operating system and applications that wish to run on it. Specifically, the available memory is typically very low. In addition, there is little processing power for maximum efficiency. TinyOS was written in such a way to allow for maximum concurrency with only one stack. While many of the tools around TinyOS are composed in either C or Java, programs for TinyOS are written in NesC. NesC is a dialect of C optimized for the limitations of sensor networks.

A key feature of TinyOS is that it is entirely non-blocking. Non-blocking input/output means that other processing can occur while waiting for input/output transmission to complete. A simple example of a non-blocking I/O operation would be a command line utility that asks for input from the user, and while waiting for that input, continues its normal execution. Once the input is received, it would be processed appropriately. In the world of TinyOS, this I/O operation would not be user input, but is more likely to be transmission of data from one wireless sensor to another. In order to achieve this non-blocking I/O with a single stack, all I/O operations that last longer than a few hundred microseconds are made asynchronous via a callback.

A callback is a way to reference an executable piece of code by another executable piece of code. Thus, applications written for TinyOS need to be able to provide pieces of code that can be executed by the OS when the transmission of data is complete. That implementation is the callback that is registered with TinyOS to be executed when the I/O operation completes. These callbacks are called events and since TinyOS uses them so extensively, they are linked into the application in such a way as to greatly increase performance.

In order to enable these performance abilities, developers for TinyOS must wrap their heads around some new programming concepts. Instead of writing procedural code, complex tasks must be linked together through a series of events. TinyOS also provides tasks, which can be scheduled at a later time from a FIFO queue. Typically these mechanisms are sufficient for high I/O applications but developers may run into issues for high CPU applications.

TinyOS is a lightweight operating system with an emphasis on low resource usage as well as high concurrency for I/O operations. While the concepts of events and event handlers are common in some programming areas, TinyOS may expose developers to a different level that they may not be used to.

References:

TinyOS Wikipedia: http://en.wikipedia.org/wiki/TinyOS
TinyOS Documentation Wiki: http://docs.tinyos.net/
Callback (computer science) Wikipedia: http://en.wikipedia.org/wiki/Callback_(computer_science)
Asynchronous I/O Wikipedia: http://en.wikipedia.org/wiki/Asynchronous_I/O