

Building Multithreaded Solutions with OmniThreadLibrary



Delphi European Conference

Primož Gabrijelčič, primoz@gabrijelcic.org



www.thedelphigeek.com

November 17/18/19 '10 VERONA





OmniThreadLibrary

ITDevCon

OmniThreadLibrary

- Multithreading library for Delphi 2007/2009 ⇒
- Task oriented
- Strong messaging support
 - Can be used with any threading infrastructure
- High level parallel programming

Project Status

- Free “As in Air” –*Theo de Raadt*
 - OpenBSD license
- Available
 - code.google.com/p/omnithreadlibrary/
- Actively developed
 - 830 commits
- Used
 - ~~1744~~ 1785 downloads of the latest release
- Almost no documentation
 - otl.17slon.com/tutorials.htm

Today's Topics

- Communication vs. Shared data
- Tasks vs. Threads
- Thread pools



Tasks

Task <> Thread

- *Task* is part of code that has to be executed
- *Thread* is execution environment
- You take care of the task,
OTL takes care of the thread

Execution Models

- CreateTask(*task_procedure*)
- CreateTask(*task_method*)
- CreateTask(*TOmniWorker_object*)
- CreateTask(*anonymous_procedure*)

- www.thedelphigeek.com/2008/09/omnithreadlibrary-patterns-how-to-not.html
- www.thedelphigeek.com/2009/11/omnithreadlibrary-patterns-task.html



Thread Pools

Thread pool

- Starting up a thread takes time
- Thread pool keeps threads alive and waits for tasks
- Automatic thread startup/shutdown
- User code executed at thread creation
 - Connection pool
- `.Run` \Rightarrow `.Schedule`



Communication

- **Pros**
 - Only one copy
 - Fast if only reading
- **Cons**
 - **Locking**
 - Bad scaling
 - Deadlocks, livelocks
- **Keep in mind**
 - Fine-grained is better
 - Optimistic locking

- Pros
 - No shared data – no* locking
- Cons
 - Hard to understand
 - Increased memory usage

**For sufficiently flexible definition of “No”*

Common Sense

- Sometimes you need both
- Minimize shared data
- Minimize interaction points

“If your solution depends on sharing data million times a second, you’re doomed.” -me

Messaging Solutions

- Windows messages
- Pipes
- Mailslots
- Sockets (TCP/IP)
- Shared memory + (micro)locking
 - Used by the OmniThreadLibrary communication primitives

Messaging in the OTL

- Bounded stack
- Bounded queue
- Dynamic queue
 - Dynamically allocated, $O(1)$ enqueue and dequeue
 - Useful for single-threaded applications too
 - www.thedelphigeek.com/2010/02/dynamic-lock-free-queue-doing-it-right.html
- All are threadsafe, microlocking, and support multiple readers and writers

TOmniValue

- Used in OTL messaging
- Record with operators
 - Can contain strings, interfaces, objects ...
- Faster than Variant
- Much faster than TValue (but slightly less powerfull)
 - www.thedelphigeek.com/2010/03/speed-comparison-variant-tvalue-and.html



Show me the code!

ITDevCon

Danger, Will Robinson!



*“New programmers
are drawn to multithreading
like moths to flame,
with similar results.”*

-Danny Thorpe

Be Afraid

- Be very afraid!
- Designing parallel solutions is hard
- Writing multithreaded code is hard
- Testing multicore apps is hard
- Debugging multithreading code is pure insanity

Keep in Mind

- Don't parallelize everything
- Don't create thousands of threads
- Rethink the algorithm
- Prove the improvements
- Test, test and test



Q & A

ITDevCon