Threads, External Processes, Sledgehammer

Fabian Immler

August 15, 2009

Sledgehammer

- Current goal $\longrightarrow$ External ATP
- In background
- Response: metis commands

Demo: sledgehammer

Prerequisites

- Isabelle 2009
- PolyML $\geq$ 5.2.1

Threads in ML

- modelled on pthread package
- simplified

Threads in Isabelle/ML

- `structure SimpleThread`
- `structure Synchronized`

Demo: creating Threads

Mutual Exclusion

- Mutex $\leftrightarrow$ Shared memory
- One lock
- Caution: interaction of lock and interrupt

Communication

- Via Shared Memory (eg Mailbox)
- No busy waiting!
- Notification

Communication with Condition Variables

- Condition Variable $\leftrightarrow$ Mutex
- Wait
  - Release Lock
- Condition Variable signal
  - Wake up with Lock aquired

Demo: pthreads primitives

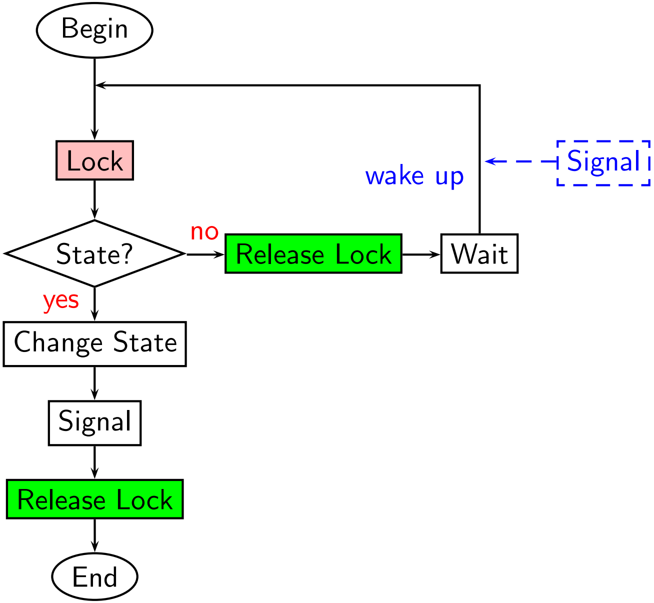
Communication

Example

Sledgehammer:

- global list of active threads
- changes invoke action

Common Communication



Synchronized State Variable

- Global, mutable state
- Synchronized access
- Exclusive locking
- Notification
- Without taking care of mutexes, condition variables, ...

Demo: Isabelle/ML combinators

Thread Attributes

- Thread is able to modify
- Control delivery of interrupt exceptions

Thread Attributes

- EnableBroadcastInterrupt
- InterruptState
    - InterruptDefer
    - InterruptSynch
    - InterruptAsynch
    - InterruptAsynchOnce

Isabelle/ML combinators

- `interruptible`
- `uninterruptible`
  - restore attributes inside

Demo: Thread attributes

system_out

- working with threads
- propagation of interrupts
- NOT OS.Process.system

system_out

- bash Script
- use File.shell_path

Demo: system_out

Further Information

- PolyML basis (http://www.polyml.org/docs/Threads.html)
- ~~/Pure/Concurrent
  - simple_thread.ML
  - synchronized.ML
  - mailbox.ML
- ~~/Pure/ML-Systems/multithreading_polyml.ML
- ~~/HOL/Tools/atp_manager.ML

What you need

- `SimpleThread.fork`
- `Thread.broadcastInterrupt` :-)
- `Synchronized.var`
- `system_out`

Thank you!