

FIFO demo 13/05/2016

prepared for MG by James Thompson

Readme

built against: gcc 4.8.4 (linux Mint)

build options: pthread, -std=c++11

build instructions

make clean

make

To run:

./mcu

Program has four execution modes:

(1) low intensity of server msgs, worker on average has to wait (wait on msg demo)

Simulate slow msg generation from the writer, the worker (reader) thread will on average have to wait for new msgs to be generated. Demonstrates worker thread waiting for a new event msg.

(2) high intensity of server msg, worker on avg cannot keep up (large # queued msgs demo)

Simulate high level of writer msgs, worker cannot keep up, demonstrates queue of requests building up and being processed.

(3) server/worker activity level roughly equal

Balanced write/reader effort.

(4) simulated OOM exception while adding msg to queue

QueueMt will generate a test OOM error on Push();

Program structure (main classes only)

Notes:

- QueueMt utilizes mutex locks on all queue calls. A further mutex is used to handle events from Push() to Pop() in the event that the worker is waiting on a new work item.
- QueueMt is a template class, in this instance worker/writer expect message format of type Common::workPackageStruct
- QueueMt uses std::queue<T> to handle the FIFO data storage
- QueueMt throws exceptions on errors.
- Worker is standard priority
- Writer is set to higher priority (linux)
- One QueueMt object owned by Mcu and handle shared by worker/writer threads
- QueueMt event system assumes one worker thread only.
- All thread msgs are sent via Common(), not thread safe in this version. TODO
- Worker thread is terminated when it reaches the end of the requested number of msgs. In a real system would move this to an event or something similar.

