Multi-threaded Strategies

- Multi-threading overview:

- Single RT audio thread initialising and processing FIFO

- Multiple “worker” threads processing the FIFO

- Questions:

- How many threads do you start?

- Trade-off between CPU use and throughput

- Fully real-time implementation means no system calls (locks, CVs, events etc.)

- Requires worker threads spinning on the FIFO waiting for available Nodes

- Worker threads can use CPU pause instructions

- Or brief yields/sleeps - very difficult to get them to wake up at the right time to process

- Non-real-time solution can use condition variables to sleep/wake worker threads

- NOT REAL-TIME SAFE!

Multi-threaded Strategies

- Multi-threading overview:
  - Single RT audio thread initialising and processing FIFO
  - Multiple “worker” threads processing the FIFO
- Questions:
  - How many threads do you start?
  - Trade-off between CPU use and throughput
- Fully real-time implementation means no system calls (locks, CVs, events etc.)
  - Requires worker threads spinning on the FIFO waiting for available `Nodes`
  - Worker threads can use CPU pause instructions
  - Or brief yields/sleeps - very difficult to get them to wake up at the right time to process
- Non-real-time solution can use condition variables to sleep/wake worker threads
  - NOT REAL-TIME SAFE!

Uses