Common Use Cases for Condition Variables

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

- Understand what condition variables are
- Note a human known use of condition variables
- Know what pattern they implement
- Recognize common use cases where condition variables are applied
Applying Condition Variables in Practice
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly

See [en.wikipedia.org/wiki/Grok](en.wikipedia.org/wiki/Grok)
Applying Condition Variables in Practice

• CVs are powerful, but can be hard to grok & apply correctly, e.g.
• The protocol for using CVs involves several “moving parts”

CAUTION
BE ALERT!! MOVING PARTS
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly, e.g.
  - The protocol for using CVs involves several “moving parts”, i.e.
    - Condition variable(s) & a lock
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly, e.g.
- The protocol for using CVs involves several “moving parts”, i.e.
  - Condition variable(s) & a lock
  - Several cooperating threads
CVs are powerful, but can be hard to grok & apply correctly, e.g.

- The protocol for using CVs involves several “moving parts”, i.e.
  - Condition variable(s) & a lock
  - Several cooperating threads
  - A monitor object that mediates access to mutable shared state

See [www.dre.vanderbilt.edu/~schmidt/PDF/monitor.pdf](http://www.dre.vanderbilt.edu/~schmidt/PDF/monitor.pdf)
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly, e.g.
  - The protocol for using CVs involves several “moving parts”, i.e.
    - Condition variable(s) & a lock
    - Several cooperating threads
    - A monitor object that mediates access to mutable shared state
    - An idiomatic sequence of steps
Applying Condition Variables in Practice

- CVs are powerful, but can be hard to grok & apply correctly, e.g.
  - The protocol for using CVs involves several “moving parts”
  - The non-determinism of concurrency is tricky

See [en.wikipedia.org/wiki/Nondeterministic_algorithm](en.wikipedia.org/wiki/Nondeterministic_algorithm)
Applying Condition Variables in Practice

• CVs are powerful, but can be hard to grok & apply correctly, e.g.
  • The protocol for using CVs involves several “moving parts”
  • The non-determinism of concurrency is tricky
    • i.e., looping may be needed to ensure a resource is available

See stackoverflow.com/a/38313778
• CVs are therefore often not used directly by apps, but instead are "hidden" within other abstractions
• CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions
• CVs form the basis for higher-level synchronizers in Java

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/locks/AbstractQueuedSynchronizer.ConditionObject.html
Applying Condition Variables in Practice

• CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions
  • CVs form the basis for higher-level synchronizers in Java, e.g.
    • Blocking queues & deques in java.util.concurrent* packages

See docs.oracle.com/javase/tutorial/collections/implementations/queue.html
CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions. CVs form the basis for higher-level synchronizers in Java, e.g.

- Blocking queues & deques in java.util.concurrent* packages
  - e.g., ArrayBlockingQueue

See upcoming discussion in “Example Application of Java ConditionObject”
Applying Condition Variables in Practice

- CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions
- CVs form the basis for higher-level synchronizers in Java, e.g.
  - Blocking queues & deques in java.util.concurrent* packages
  - Java built-in monitor objects

See upcoming lesson on “Java Built-in Monitor Objects”
Applying Condition Variables in Practice

- CVs are therefore often not used directly by apps, but instead are “hidden” within other abstractions.
- CVs form the basis for higher-level synchronizers in Java, e.g.
  - Blocking queues & deques in java.util.concurrent* packages
  - Java built-in monitor objects
  - The Monitor Object pattern

See www.dre.vanderbilt.edu/~schmidt/PDF/monitor.pdf
End of Common Use Cases for Condition Variables