

Solutions aux exercices du labo #6 (2 novembre 2017)

Solution à l'exercice 1.

```
class ReentrantLock
  ...

  def lock
    @mutex.synchronize do
      while hold_count > 0 && @owner != Thread.current
        # Un autre thread a le verrou.
        @free.wait(@mutex)
      end

      DBC.assert @hold_count == 0 ||
                 @owner == Thread.current

      # Le thread courant a maintenant le verrou...
      # ou l'avait deja!
      @owner = Thread.current
      @hold_count += 1
    end
  end

  def unlock
    @mutex.synchronize do
      DBC.assert @owner == Thread.current

      @hold_count -= 1
      if @hold_count == 0
        # Dernier unlock en suspens:
        # on signale un des threads en attente.
        @owner = nil
        @free.signal
      end
    end
  end
end

#
```

```
def trylock
  @mutex.synchronize do
    return false if hold_count > 0 && owner != Thread.current

    # hold_count == 0 || owner == Thread.current
    @owner = Thread.current
    @hold_count += 1
    true
  end
end
end
end
```

Solution à l'exercice 2.

```
class Reducteur
  def initialize( val_initiale, nb_threads, &bloc )
    @val_initiale = val_initiale
    @nb_threads = nb_threads
    @bloc = bloc

    @mutex = Mutex.new
    @tous_arrives = ConditionVariable.new
    @nb_arrives = 0
    @total = @val_initiale # Total cumulatif.
    @resultat_final = nil
  end

  def reduire( valeur )
    @mutex.synchronize do
      @nb_arrives += 1
      @total = @bloc.call( @total, valeur )

      if @nb_arrives < @nb_threads
        # Pas le dernier, donc on attend...
        @tous_arrives.wait( @mutex )

        # Maintenant tous arrives: on retourne le resultat.
        @resultat_final
      else
        # Le dernier thread vient d'arriver...
        # On prend en note le resultat final.
        @resultat_final = @total

        # On reinitialise pour la vague suivante.
        @nb_arrives = 0
        @total = @val_initiale

        # On signale les threads arrives avant.
        @tous_arrives.broadcast

        # On retourne le resultat
        @resultat_final
      end
    end
  end
end
```