Rule 14. Concurrency (CON)

- CON30-C. Clean up thread-specific storage
- CON31-C. Do not destroy a mutex while it is locked
- CON32-C. Prevent data races when accessing bit-fields from multiple threads
- CON33-C. Avoid race conditions when using library functions
- CON34-C. Declare objects shared between threads with appropriate storage durations
- CON35-C. Avoid deadlock by locking in a predefined order
- CON36-C. Wrap functions that can spuriously wake up in a loop
- CON37-C. Do not call signal() in a multithreaded program
- CON38-C. Preserve thread safety and liveness when using condition variables
- CON39-C. Do not join or detach a thread that was previously joined or detached
- CON40-C. Do not refer to an atomic variable twice in an expression
- CON41-C. Wrap functions that can fail spuriously in a loop
- CON43-C. Do not allow data races in multithreaded code

Information for Editors
To have a new guideline automatically listed above be sure to label it con and rule.

Risk Assessment Summary

<table>
<thead>
<tr>
<th>Rule</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Remediation Cost</th>
<th>Priority</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON30-C</td>
<td>Medium</td>
<td>Unlikely</td>
<td>Medium</td>
<td>P4</td>
<td>L3</td>
</tr>
<tr>
<td>CON31-C</td>
<td>Medium</td>
<td>Probable</td>
<td>High</td>
<td>P4</td>
<td>L3</td>
</tr>
<tr>
<td>CON32-C</td>
<td>Medium</td>
<td>Probable</td>
<td>Medium</td>
<td>P8</td>
<td>L2</td>
</tr>
<tr>
<td>CON33-C</td>
<td>Medium</td>
<td>Probable</td>
<td>High</td>
<td>P4</td>
<td>L3</td>
</tr>
<tr>
<td>CON34-C</td>
<td>Medium</td>
<td>Probable</td>
<td>High</td>
<td>P4</td>
<td>L3</td>
</tr>
<tr>
<td>CON35-C</td>
<td>Low</td>
<td>Probable</td>
<td>Medium</td>
<td>P4</td>
<td>L3</td>
</tr>
<tr>
<td>CON36-C</td>
<td>Low</td>
<td>Unlikely</td>
<td>Medium</td>
<td>P2</td>
<td>L3</td>
</tr>
<tr>
<td>CON37-C</td>
<td>Low</td>
<td>Probable</td>
<td>Low</td>
<td>P6</td>
<td>L2</td>
</tr>
<tr>
<td>CON38-C</td>
<td>Low</td>
<td>Unlikely</td>
<td>Medium</td>
<td>P2</td>
<td>L3</td>
</tr>
<tr>
<td>CON39-C</td>
<td>Low</td>
<td>Likely</td>
<td>Medium</td>
<td>P6</td>
<td>L2</td>
</tr>
<tr>
<td>CON40-C</td>
<td>Medium</td>
<td>Probable</td>
<td>Medium</td>
<td>P8</td>
<td>L2</td>
</tr>
<tr>
<td>CON41-C</td>
<td>Low</td>
<td>Unlikely</td>
<td>Medium</td>
<td>P2</td>
<td>L3</td>
</tr>
<tr>
<td>CON43-C</td>
<td>Medium</td>
<td>Probable</td>
<td>High</td>
<td>P4</td>
<td>L3</td>
</tr>
</tbody>
</table>

Related Rules and Recommendations

CON04-C. Join or detach threads even if their exit status is unimportant
CON30-C. Clean up thread-specific storage
CON31-C. Do not destroy a mutex while it is locked
CON32-C. Prevent data races when accessing bit-fields from multiple threads
CON33-C. Avoid race conditions when using library functions
CON34-C. Declare objects shared between threads with appropriate storage durations
CON35-C. Avoid deadlock by locking in a predefined order
CON37-C. Do not call signal() in a multithreaded program
CON43-C. Do not allow data races in multithreaded code
POS04-C. Avoid using PTHREAD_MUTEX_NORMAL type mutex locks
POS44-C. Do not use signals to terminate threads
POS47-C. Do not use threads that can be canceled asynchronously
POS48-C. Do not unlock or destroy another POSIX thread's mutex
POS49-C. When data must be accessed by multiple threads, provide a mutex and guarantee no adjacent data is also accessed
POS50-C. Declare objects shared between POSIX threads with appropriate storage durations
POS51-C. Avoid deadlock with POSIX threads by locking in predefined order
POS53-C. Do not use more than one mutex for concurrent waiting operations on a condition variable
WIN01-C. Do not forcibly terminate execution