
MODULE *OpCounter*

EXTENDS
Naturals, Sequences, SEC

VARIABLES

<i>counter</i> ,	<i>counter</i> [<i>r</i>]: current value of the counter at replica <i>r</i> ∈ <i>Replica</i>
<i>buffer</i> ,	<i>buffer</i> [<i>r</i>]: number of increments performed since the last broadcast at replica <i>r</i> ∈ <i>Replica</i>
<i>seq</i> ,	<i>seq</i> [<i>r</i>]: local sequence number at replica <i>r</i> ∈ <i>Replica</i>
<i>incoming</i> ,	<i>incoming</i> [<i>r</i>]: incoming messages at replica <i>r</i> ∈ <i>Replica</i>
<i>msg</i> ,	<i>incoming</i> [<i>r</i>]: current message at replica <i>r</i> ∈ <i>Replica</i>
<i>messageSet</i>	network variable

vars \triangleq $\langle counter, buffer, seq, incoming, msg, messageSet, SECvars \rangle$

Msg \triangleq [*r* : *Replica*, *buf* : *Nat*, *seq* : *Nat*, *update* : SUBSET *Update*]

Network \triangleq INSTANCE *ReliableNetwork*

TypeOK \triangleq

\wedge *counter* ∈ [*Replica* → *Nat*]
 \wedge *buffer* ∈ [*Replica* → *Nat*]
 \wedge *seq* ∈ [*Replica* → *Nat*]

Init \triangleq

$\wedge counter = [r \in Replica \mapsto 0]$
 $\wedge buffer = [r \in Replica \mapsto 0]$
 $\wedge seq = [r \in Replica \mapsto 0]$
 $\wedge Network!RInit$
 $\wedge SECInit$

Read(*r*) \triangleq *counter*[*r*]

Inc(*r*) \triangleq

$\wedge counter' = [counter \text{ EXCEPT } ![r] = @ + 1]$
 $\wedge buffer' = [buffer \text{ EXCEPT } ![r] = @ + 1]$
 $\wedge seq' = [seq \text{ EXCEPT } ![r] = @ + 1]$
 $\wedge SECUpdate(r, seq[r])$
 $\wedge \text{UNCHANGED } \langle incoming, msg, messageSet \rangle$

Send(*r*) \triangleq

$\wedge buffer[r] \neq 0$
 $\wedge buffer' = [buffer \text{ EXCEPT } ![r] = 0]$
 $\wedge Network!RBroadcast(r, [r \mapsto r, buf \mapsto buffer[r], seq \mapsto seq[r], update \mapsto OpUpdate(r)])$
 $\wedge SECSend(r)$
 $\wedge \text{UNCHANGED } \langle counter, seq \rangle$

Receive(*r*) \triangleq

$$\begin{array}{l}
\wedge \text{ Network!}RD\text{deliver}(r) \\
\wedge \text{ SECDeliver}(r, \text{msg}'[r]) \\
\wedge \text{ counter}' = [\text{counter} \text{ EXCEPT } ![r] = @ + \text{msg}'[r].\text{buf}] \\
\wedge \text{ UNCHANGED } \langle \text{buffer}, \text{seq} \rangle \\
\hline
\text{Next} \triangleq \exists r \in \text{Replica} : \text{Inc}(r) \vee \text{Send}(r) \vee \text{Receive}(r) \\
\hline
\text{Spec} \triangleq \text{Init} \wedge \Box[\text{Next}]_{\text{vars}} \\
\hline
\text{EmptyBuffer} \triangleq \text{buffer} = [r \in \text{Replica} \mapsto 0] \\
\text{EC} \triangleq \text{Network!EmptyChannel} \wedge \text{EmptyBuffer} \\
\quad \Rightarrow \forall r1, r2 \in \text{Replica} : \text{Read}(r1) = \text{Read}(r2) \\
\text{SEC} \triangleq \forall r1, r2 \in \text{Replica} : \text{SameUpdate}(r1, r2) \Rightarrow \text{Read}(r1) = \text{Read}(r2) \\
\hline
\backslash * \text{ Modification History} \\
\backslash * \text{ Last modified Thu May 16 09:23:01 CST 2019 by zfwang} \\
\backslash * \text{ Last modified Tue May 07 00:57:30 CST 2019 by xhdn} \\
\backslash * \text{ Last modified Mon May 06 15:51:30 CST 2019 by jywelling} \\
\backslash * \text{ Created Fri Mar 22 20:43:27 CST 2019 by jywelling}
\end{array}$$