MODULE *SemaphoresTrivial*

EXTENDS *Integers*, *Sequences*, *TLC*

CONSTANT N

```
--algorithm AddByNProcesses{
  variable
      x = 0,
      semaphore = "FREE" ;
      numberOfProcesses = 1 .. N ;

  macro toggleSemaphore( ) {
      if ( semaphore = "LOCKED" ) {
      semaphore := "FREE" ;
       } else {
      semaphore := "LOCKED" ;
       } ;
   }

  process ( Increment ∈ numberOfProcesses )
  variable temp = -1 ;
  {
      lock:   await semaphore = "FREE" ;
              toggleSemaphore() ;

      read: temp := x ;
      write: x := temp + 1 ;

      unblock: toggleSemaphore() ;
   }

}
```

BEGIN TRANSLATION ($chksum(pcal) =$ "$d299ec71$" $\land\ chksum(tla) =$ "$9b8da42f$")

VARIABLES x, $semaphore$, $numberOfProcesses$, pc, $temp$

$vars \triangleq \langle x,\ semaphore,\ numberOfProcesses,\ pc,\ temp \rangle$

$ProcSet \triangleq (numberOfProcesses)$

$Init \triangleq$ Global variables
$\land\ x = 0$
$\land\ semaphore =$ "FREE"
$\land\ numberOfProcesses = 1 \,..\, N$
Process *Increment*
$\land\ temp = [self \in numberOfProcesses \mapsto -1]$
$\land\ pc = [self \in ProcSet \mapsto$ "lock"$]$

$lock(self) \triangleq \land\ pc[self] =$ "lock"
$\land\ semaphore =$ "FREE"

$$
\begin{array}{l}
\quad \land \text{IF } semaphore = \text{“LOCKED”} \\
\qquad \text{THEN } \land\, semaphore' = \text{“FREE”} \\
\qquad \text{ELSE } \land\, semaphore' = \text{“LOCKED”} \\
\quad \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“read”}] \\
\quad \land \text{UNCHANGED } \langle x,\, numberOfProcesses,\, temp \rangle
\end{array}
$$

$$
\begin{array}{rl}
read(self) \triangleq & \land\, pc[self] = \text{“read”} \\
& \land\, temp' = [temp \text{ EXCEPT } ![self] = x] \\
& \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“write”}] \\
& \land \text{UNCHANGED } \langle x,\, semaphore,\, numberOfProcesses \rangle
\end{array}
$$

$$
\begin{array}{rl}
write(self) \triangleq & \land\, pc[self] = \text{“write”} \\
& \land\, x' = temp[self] + 1 \\
& \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“unblock”}] \\
& \land \text{UNCHANGED } \langle semaphore,\, numberOfProcesses,\, temp \rangle
\end{array}
$$

$$
\begin{array}{rl}
unblock(self) \triangleq & \land\, pc[self] = \text{“unblock”} \\
& \land \text{IF } semaphore = \text{“LOCKED”} \\
& \qquad \text{THEN } \land\, semaphore' = \text{“FREE”} \\
& \qquad \text{ELSE } \land\, semaphore' = \text{“LOCKED”} \\
& \land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“Done”}] \\
& \land \text{UNCHANGED } \langle x,\, numberOfProcesses,\, temp \rangle
\end{array}
$$

$$
Increment(self) \triangleq lock(self) \lor read(self) \lor write(self) \lor unblock(self)
$$

Allow infinite stuttering to prevent deadlock on termination.

$$
\begin{array}{rl}
Terminating \triangleq & \land\, \forall\, self \in ProcSet : pc[self] = \text{“Done”} \\
& \land \text{UNCHANGED } vars
\end{array}
$$

$$
\begin{array}{rl}
Next \triangleq & (\exists\, self \in numberOfProcesses : Increment(self)) \\
& \quad \lor Terminating
\end{array}
$$

$$
Spec \triangleq Init \land \Box[Next]_{vars}
$$

$$
Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})
$$

END TRANSLATION

* Modification History
* Last modified *Tue Mar* 12 13:38:36 *CET* 2024 by *jeujeus*
* Created *Tue Mar* 12 12:58:14 *CET* 2024 by *jeujeus*