MODULE *IdGenerator*

EXTENDS *Integers*, *TLC*

CONSTANT *NumberOfProcesses*

--fair algorithm *IdGenerator*{

variable $lastIdUsed = 42$, $processIds = [i \in 1 \,..\, NumberOfProcesses \mapsto 0]$;

define {
$AllIsDone \triangleq (\forall\, i \in 1 \,..\, NumberOfProcesses : pc[i] = \text{“Done”})$
$IdsAreAllDifferent \triangleq (\forall\, i, j \in 1 \,..\, NumberOfProcesses : i \neq j \Rightarrow processIds[i] \neq processIds[j])$
$IdGeneratorInvariant \triangleq AllIsDone \Rightarrow IdsAreAllDifferent$
}

process ($id \in 1 \,..\, NumberOfProcesses$) {
update and read are separate steps
update: $lastIdUsed := lastIdUsed + 1$;
read: $processIds[self] := lastIdUsed$
}
}

BEGIN TRANSLATION
VARIABLES *lastIdUsed*, *processIds*, *pc*

define statement
$AllIsDone \triangleq (\forall\, i \in 1 \,..\, NumberOfProcesses : pc[i] = \text{“Done”})$
$IdsAreAllDifferent \triangleq (\forall\, i, j \in 1 \,..\, NumberOfProcesses : i \neq j \Rightarrow processIds[i] \neq processIds[j])$
$IdGeneratorInvariant \triangleq AllIsDone \Rightarrow IdsAreAllDifferent$

$vars \triangleq \langle lastIdUsed, processIds, pc \rangle$

$ProcSet \triangleq (1 \,..\, NumberOfProcesses)$

$Init \triangleq$ Global variables
$\land\, lastIdUsed = 42$
$\land\, processIds = [i \in 1 \,..\, NumberOfProcesses \mapsto 0]$
$\land\, pc = [self \in ProcSet \mapsto \text{“update”}]$

$update(self) \triangleq \land\, pc[self] = \text{“update”}$
$\land\, lastIdUsed' = lastIdUsed + 1$
$\land\, pc' = [pc \text{ EXCEPT } ![self] = \text{“read”}]$
$\land$ UNCHANGED *processIds*

$read(self) \triangleq \land\, pc[self] = \text{“read”}$
$\land\, processIds' = [processIds \text{ EXCEPT } ![self] = lastIdUsed]$

$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“Done”}]$
$\land$ UNCHANGED $lastIdUsed$

$id(self) \triangleq update(self) \lor read(self)$

$Next \triangleq (\exists\, self \in 1 \,..\, NumberOfProcesses : id(self))$
$\lor$ Disjunct to prevent deadlock on termination
$((\forall\, self \in ProcSet : pc[self] = \text{“Done”}) \land \text{UNCHANGED } vars)$

$Spec \triangleq \land Init \land \Box[Next]_{vars}$
$\land \text{WF}_{vars}(Next)$

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

END TRANSLATION

* Modification History
* Last modified Sun *Dec* 13 13:38:46 *EST* 2015 by *balopat*
* Created *Fri Dec* 04 17:18:27 *EST* 2015 by *balopat*