MODULE *actor*

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

$between01(n1,\ nb,\ n2) \triangleq ((n1 < n2) \Rightarrow ((n1 < nb) \land (nb \leq n2))) \lor ((n1 \geq n2) \Rightarrow ((n1 < nb) \lor (nb \leq n2)))$
$between00(n1,\ nb,\ n2) \triangleq ((n1 < n2) \Rightarrow ((n1 < nb) \land (nb < n2))) \lor ((n1 \geq n2) \Rightarrow ((n1 < nb) \lor (nb < n2)))$

```
 --fair algorithm ActorStuff {
variables actorInboxes = (0 :> ⟨⟨"FindPredecessor", 6, 0⟩⟩) @@ (1 :> ⟨⟩) @@ (3 :> ⟨⟩) ;
          fingerTables = (0 :> (1 :> 1) @@ (2 :> 3) @@ (4 :> 0))
           @@ (1 :> (2 :> 3) @@ (3 :> 5) @@ (5 :> 0))
           @@ (3 :> (4 :> 0) @@ (5 :> 0) @@ (7 :> 0)) ;
          triggered = FALSE ;
          m = 3 ;

procedure trigger( trigger_content = "?" ) {
    triggerLabel:
      triggered := TRUE ;
      print triggered ;
      return ;
 }

  mod
fair process ( actor ∈ {0, 1, 3} )
variables currentMessage = ⟨"?", "no_content"⟩ ;
  kind = "?" ;
  content = "no_content" ;
  id ;
  asker ;
  i ;
{
  Send:
    actorInboxes["actor"] := Append(actorInboxes["actor"], ⟨"trigger", "foo"⟩);
  WaitForMessages:+
    while ( TRUE ) {
      if ( actorInboxes[self] ≠ ⟨⟩ ) {
         currentMessage := Head(actorInboxes[self]) ;
         content := Head(Tail(currentMessage)) ;
         kind := Head(currentMessage) ;
         actorInboxes[self] := Tail(actorInboxes[self]) ;
         } ;
        ProcessMessage:
         if ( kind = "FindPredecessor" ) {
           id := currentMessage[2] ;
           asker := currentMessage[3] ;
           if ( between01(self, id, fingerTables[self][self + 1]) ) {
```

```
            actorInboxes[asker] := Append(actorInboxes[asker], ⟨"Predecessor", self⟩) ;
            } else {
            i := m ;
            LOOP:
             while ( i > 0 ∧ ¬(between00(self, fingerTables[self][self + (2^(i−1))], id)) ) {
              i := i − 1 ;
              } ;
            if ( i = 0 ) {
            actorInboxes[fingerTables[self][self + (2^(m−1))]] :=
               Append(actorInboxes[fingerTables[self][self + (2^(m−1))]], currentMessage) ;
             } else {
            actorInboxes[fingerTables[self][self + (2^(i−1))]] :=
             Append(actorInboxes[fingerTables[self][self + (2^(i−1))]], currentMessage) ;
             }

            }
          } else {
            if ( kind = "Predecessor" ) {
             call trigger(content) ;
             }
          }
    }
}
}
```

BEGIN TRANSLATION ($chksum(pcal) =$ "62e423bf" $\wedge\ chksum(tla) =$ "9acec44b")

CONSTANT *defaultInitValue*

VARIABLES *actorInboxes*, *fingerTables*, *triggered*, *m*, *pc*, *stack*,
trigger_content, *currentMessage*, *kind*, *content*, *id*, *asker*, *i*

$vars \triangleq \langle actorInboxes, fingerTables, triggered, m, pc, stack,$
$trigger\_content, currentMessage, kind, content, id, asker, i\rangle$

$ProcSet \triangleq (\{0, 1, 3\})$

$Init \triangleq$ Global variables

$\wedge\ actorInboxes = (0 :> \langle\langle \text{"FindPredecessor"}, 6, 0\rangle\rangle) \mathbin{@@} (1 :> \langle\rangle) \mathbin{@@} (3 :> \langle\rangle)$

$\wedge\ fingerTables = (0 :> (1 :> 1) \mathbin{@@} (2 :> 3) \mathbin{@@} (4 :> 0))$
$\mathbin{@@} (1 :> (2 :> 3) \mathbin{@@} (3 :> 5) \mathbin{@@} (5 :> 0))$
$\mathbin{@@} (3 :> (4 :> 0) \mathbin{@@} (5 :> 0) \mathbin{@@} (7 :> 0))$

$\wedge\ triggered = \text{FALSE}$

$\wedge\ m = 3$

Procedure trigger

$\wedge\ trigger\_content = [self \in ProcSet \mapsto \text{"?"}]$

Process actor

$\wedge\ currentMessage = [self \in \{0, 1, 3\} \mapsto \langle \text{"?"}, \text{"no\_content"}\rangle]$

$\wedge\ kind = [self \in \{0, 1, 3\} \mapsto \text{"?"}]$

$\land\ content = [self \in \{0, 1, 3\} \mapsto \text{“no\_content”}]$
$\land\ id = [self \in \{0, 1, 3\} \mapsto defaultInitValue]$
$\land\ asker = [self \in \{0, 1, 3\} \mapsto defaultInitValue]$
$\land\ i = [self \in \{0, 1, 3\} \mapsto defaultInitValue]$
$\land\ stack = [self \in ProcSet \mapsto \langle\rangle]$
$\land\ pc = [self \in ProcSet \mapsto \text{“WaitForMessages”}]$

$triggerLabel(self) \triangleq$
$\land\ pc[self] = \text{“triggerLabel”}$
$\land\ triggered' = \text{TRUE}$
$\land\ PrintT(triggered')$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = Head(stack[self]).pc]$
$\land\ trigger\_content' = [trigger\_content \text{ EXCEPT } ![self] = Head(stack[self]).trigger\_conten$
$\land\ stack' = [stack \text{ EXCEPT } ![self] = Tail(stack[self])]$
$\land \text{UNCHANGED } \langle actorInboxes, fingerTables, m,$
$currentMessage, kind, content, id, asker,$
$i\rangle$

$trigger(self) \triangleq triggerLabel(self)$

$WaitForMessages(self) \triangleq$
$\land\ pc[self] = \text{“WaitForMessages”}$
$\land \text{IF } actorInboxes[self] \neq \langle\rangle$
$\text{THEN } \land\ currentMessage' = [currentMessage \text{ EXCEPT } ![self] = Head(actorIn$
$\land\ content' = [content \text{ EXCEPT } ![self] = Head(Tail(currentMessage'[s$
$\land\ kind' = [kind \text{ EXCEPT } ![self] = Head(currentMessage'[self])]$
$\land\ actorInboxes' = [actorInboxes \text{ EXCEPT } ![self] = Tail(actorInboxes[s$
$\text{ELSE } \land \text{TRUE}$
$\land \text{UNCHANGED } \langle actorInboxes,$
$currentMessage, kind,$
$content\rangle$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“ProcessMessage”}]$
$\land \text{UNCHANGED } \langle fingerTables, triggered, m, stack,$
$trigger\_content, id, asker, i\rangle$

$ProcessMessage(self) \triangleq$
$\land\ pc[self] = \text{“ProcessMessage”}$
$\land \text{IF } kind[self] = \text{“FindPredecessor”}$
$\text{THEN } \land\ id' = [id \text{ EXCEPT } ![self] = currentMessage[self][2]]$
$\land\ asker' = [asker \text{ EXCEPT } ![self] = currentMessage[self][3]]$
$\land \text{IF } between01(self, id'[self], fingerTables[self][self + 1])$
$\text{THEN } \land\ actorInboxes' = [actorInboxes \text{ EXCEPT } ![asker'[self]] = A$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“WaitForMessages”}]$
$\land\ i' = i$
$\text{ELSE } \land\ i' = [i \text{ EXCEPT } ![self] = m]$
$\land\ pc' = [pc \text{ EXCEPT } ![self] = \text{“LOOP”}]$
$\land \text{UNCHANGED } actorInboxes$
$\land \text{UNCHANGED } \langle stack, trigger\_content\rangle$
$\text{ELSE } \land \text{IF } kind[self] = \text{“Predecessor”}$

$$
\begin{array}{l}
\text{THEN} \land \land stack' = [stack \text{ EXCEPT } ![self] = \langle [procedure \mapsto \text{“trigg}\\
\qquad pc \mapsto \text{“Wait}\\
\qquad trigger\_content \mapsto\\
\qquad \circ\, stack[self]]\\
\quad \land trigger\_content' = [trigger\_content \text{ EXCEPT } ![self] = \\
\quad \land pc' = [pc \text{ EXCEPT } ![self] = \text{“triggerLabel”}]\\
\text{ELSE} \land pc' = [pc \text{ EXCEPT } ![self] = \text{“WaitForMessages”}]\\
\quad \land \text{UNCHANGED } \langle stack, trigger\_content\rangle\\
\land \text{UNCHANGED } \langle actorInboxes, id, asker, i\rangle\\
\land \text{UNCHANGED } \langle fingerTables, triggered, m, currentMessage, kind, content\rangle
\end{array}
$$

$$
\begin{array}{l}
LOOP(self) \triangleq \land pc[self] = \text{“LOOP”}\\
\quad \land \text{IF } i[self] > 0 \land \neg(between00(self, fingerTables[self][self + (2^{(i[self]-1)})], id[self]))\\
\qquad \text{THEN} \land i' = [i \text{ EXCEPT } ![self] = i[self] - 1]\\
\qquad\qquad \land pc' = [pc \text{ EXCEPT } ![self] = \text{“LOOP”}]\\
\qquad\qquad \land \text{UNCHANGED } actorInboxes\\
\qquad \text{ELSE} \land \text{IF } i[self] = 0\\
\qquad\qquad\qquad \text{THEN} \land actorInboxes' = [actorInboxes \text{ EXCEPT } ![fingerTables[self][self + (\\
\qquad\qquad\qquad \text{ELSE} \land actorInboxes' = [actorInboxes \text{ EXCEPT } ![fingerTables[self][self + (\\
\qquad\qquad \land pc' = [pc \text{ EXCEPT } ![self] = \text{“WaitForMessages”}]\\
\qquad\qquad \land i' = i\\
\quad \land \text{UNCHANGED } \langle fingerTables, triggered, m, stack, trigger\_content, currentMessage, kind, content, id, asker\rangle
\end{array}
$$

$$actor(self) \triangleq WaitForMessages(self) \lor ProcessMessage(self) \lor LOOP(self)$$

$$
\begin{array}{l}
Next \triangleq (\exists\, self \in ProcSet : trigger(self))\\
\qquad \lor (\exists\, self \in \{0, 1, 3\} : actor(self))
\end{array}
$$

$$
\begin{array}{l}
Spec \triangleq \land Init \land \Box[Next]_{vars}\\
\qquad \land \mathrm{WF}_{vars}(Next)\\
\qquad \land \forall\, self \in \{0, 1, 3\} : \land \mathrm{WF}_{vars}(actor(self))\\
\qquad\qquad \land \mathrm{SF}_{vars}(WaitForMessages(self))\\
\qquad\qquad \land \mathrm{WF}_{vars}(trigger(self))
\end{array}
$$

END TRANSLATION

$$Triggered \triangleq triggered = \text{TRUE}$$

$$Liveness \triangleq \Diamond Triggered$$

* Modification History
* Last modified *Mon Jan* 31 02:59:31 *YEKT* 2022 by *pervu*
* Created Sun *Jan* 30 18:34:11 *YEKT* 2022 by *pervu*