```
──────────────────────────── MODULE actor ────────────────────────────
EXTENDS TLC, Integers, Sequences

between01(n1, nb, n2) ≜ (nb ≥ 0) ∧ (((n1 < n2) ∧ ((n1 < nb) ∧ (nb ≤ n2))) ∨ ((n1 ≥ n2) ∧ ((n1 < nb) ∨ (nb
between00(n1, nb, n2) ≜ (nb ≥ 0) ∧ (((n1 < n2) ∧ ((n1 < nb) ∧ (nb < n2))) ∨ ((n1 ≥ n2) ∧ ((n1 < nb) ∨ (nb

fingerStart(myId, k, m) ≜ (myId + (2^(k−1)))%m

CONSTANTS m, bm, NULL

  --fair algorithm ActorStuff {
variables actorInboxes = (0 :> ⟨⟩) @@ (1 :> ⟨⟩) @@ (3 :> ⟨⟩) ;
          triggered = FALSE ;
            make local
          fingerTables = (0 :> ((1 :> 0) @@ (2 :> 0) @@ (4 :> 0)))
           @@ (1 :> ((2 :> NULL) @@ (3 :> NULL) @@ (5 :> NULL)))
           @@ (3 :> ((4 :> NULL) @@ (5 :> NULL) @@ (7 :> NULL))) ;
          predecessors = (0 :> 0) @@ (1 :> NULL) @@ (3 :> NULL)   ;

fair process ( actor ∈ {0, 1, 3} )
variables currentMessage = ⟨"?", NULL, NULL⟩ ;
  kind = "?" ;
  id = NULL ;
  i ;
  joined = FALSE ;
{
  Join:+
    if ( self = 0 ) {
        joined := TRUE ;
     } else {
         actorInboxes[0] := Append(actorInboxes[0], ⟨"FindPredecessor", fingerStart(self, 1, bm), self⟩) ;
     } ;
  WaitForMessagesBeforeJoin:+
    if ( joined ≠ TRUE ) {
        WhileWaitForMessagesBeforeJoin:+

        await actorInboxes[self] ≠ ⟨⟩ ;
        {
         currentMessage := Head(actorInboxes[self]) ;
         kind := Head(currentMessage) ;
         actorInboxes[self] := Tail(actorInboxes[self]) ;
         } ;
        ProcessMessageBeforeJoin:+
        {
            if ( kind = "Predecessor" ) {
                predecessors[self] := currentMessage[2] ;
                fingerTables[self][fingerStart(self, 1, bm)] := currentMessage[3] ;
```

```
                  joined := TRUE ;
                }
            } ;
          ReturnDefaultsBeforeJoin:+
           currentMessage := ⟨"?", NULL, NULL⟩ ;
           kind := "?" ;
           id := NULL ;
    } ;
Stabilize : +
 if (joined){
     actorInboxes[self] := Append(actorInboxes[self], ⟨"Stabilize", 1⟩);
 };
WaitForMessages : +
 if (joined)
 {
 WhileWaitForMessages : +
  while (TRUE) {
  if (actorInboxes[self] ≠ ⟨⟩) {
     currentMessage := Head(actorInboxes[self]);
     kind := Head(currentMessage);
     actorInboxes[self] := Tail(actorInboxes[self]);
    };
    ProcessMessage : +
    {
     if (kind = "FindPredecessor"){
         id := currentMessage[2];
     \*await fingerTables[self][fingerStart(self, 1, bm)] ≠ NULL;
     if (between01(self, id, fingerTables[self][fingerStart(self, 1, bm)])){
         actorInboxes[currentMessage[3]] :=
         Append(actorInboxes[currentMessage[3]], ⟨"Predecessor", self, fingerTables[self][fingerStart(self, 1, bm)]⟩);
     } else {
         i := m;
         await fingerTables[self] ≠ NULL;
         FindFirstSuitableI:
         while (i > 0 ∧ ¬(fingerStart(self, i, bm) ∈ DOMAIN fingerTables[self])){
           i := i − 1;
         };
         await fingerTables[self][fingerStart(self, i, bm)] ≠ NULL;
         MainLoop:
         while (i > 0 ∧ ¬(between00(self, fingerTables[self][fingerStart(self, i, bm)], id))){
             i := i − 1;
             FindSuitableI:
             while (i > 0 ∧ ¬(fingerStart(self, i, bm) ∈ DOMAIN fingerTables[self])){
                 i := i − 1;
             };
    };
    if (i = 0){
      actorInboxes[fingerTables[self][fingerStart(self, m, bm)]] :=
      Append(actorInboxes[fingerTables[self][fingerStart(self, m, bm)]], currentMessage);
    }else{
      actorInboxes[fingerTables[self][fingerStart(self, i, bm)]] :=
      Append(actorInboxes[fingerTables[self][fingerStart(self, i, bm)]], currentMessage);
    };
  };
  } else {
    {
      if (kind = “Stabilize”) {
        actorInboxes[fingerTables[self][fingerStart(self, currentMessage[2], bm)]] :=
        Append(actorInboxes[fingerTables[self][fingerStart(self, currentMessage[2], bm)]], ⟨“GetPredecessor”, self⟩);
      } else {
        if (kind = “GetPredecessor”) {
          actorInboxes[currentMessage[2]] :=
          Append(actorInboxes[currentMessage[2]], ⟨“GotPredecessor”, predecessors[self]⟩);
        } else {
          if (kind = “GotPredecessor”) {
            if (between00(currentMessage[2], fingerTables[self][fingerStart(self, 1, bm)], self)){
              fingerTables[self][fingerStart(self, 1, bm)] := currentMessage[2];
            };
            actorInboxes[fingerTables[self][fingerStart(self, 1, bm)]] :=
            Append(actorInboxes[fingerTables[self][fingerStart(self, 1, bm)]], ⟨“Notify”, self⟩);
          } else {
            if (kind = “Notify”) {
              if (predecessors[self] ≠ NULL){
                if (between00(currentMessage[2], self, predecessors[self])){
                  predecessors[self] := currentMessage[2];
                };
              };
            }
          }
        }
      }
    }
  };
  };
  ReturnDefaults : +
  currentMessage := ⟨“?”, NULL, NULL⟩;
  kind := “?”;
  id := NULL;
  };
};
```

```
} ;
}  ;
 BEGIN TRANSLATION (chksum(pcal) = “1ca98b0a” ∧ chksum(tla) = “1c215811”)
CONSTANT defaultInitValue
VARIABLES actorInboxes, triggered, fingerTables, predecessors, pc,
          currentMessage, kind, id, i, joined

vars ≜ ⟨actorInboxes, triggered, fingerTables, predecessors, pc,
        currentMessage, kind, id, i, joined⟩

ProcSet ≜ ({0, 1, 3})

Init ≜  Global variables
        ∧ actorInboxes = (0:> ⟨⟩) @@ (1:> ⟨⟩) @@ (3:> ⟨⟩)
        ∧ triggered = FALSE
        ∧ fingerTables =             (0:> ((1:> 0) @@ (2:> 0) @@ (4:> 0)))
                         @@ (1:> ((2:> NULL) @@ (3:> NULL) @@ (5:> NULL)))
                         @@ (3:> ((4:> NULL) @@ (5:> NULL) @@ (7:> NULL)))
        ∧ predecessors = (0:> 0) @@ (1:> NULL) @@ (3:> NULL)
        Process actor
        ∧ currentMessage = [self ∈ {0, 1, 3} ↦ ⟨“?”, NULL, NULL⟩]
        ∧ kind = [self ∈ {0, 1, 3} ↦ “?”]
        ∧ id = [self ∈ {0, 1, 3} ↦ NULL]
        ∧ i = [self ∈ {0, 1, 3} ↦ defaultInitValue]
        ∧ joined = [self ∈ {0, 1, 3} ↦ FALSE]
        ∧ pc = [self ∈ ProcSet ↦ “Join”]

Join(self) ≜ ∧ pc[self] = “Join”
             ∧ IF self = 0
                  THEN ∧ joined′ = [joined EXCEPT ![self] = TRUE]
                       ∧ UNCHANGED actorInboxes
                  ELSE ∧ actorInboxes′ = [actorInboxes EXCEPT ![0] = Append(actorInboxes[0], ⟨“FindPre
                       ∧ UNCHANGED joined
             ∧ pc′ = [pc EXCEPT ![self] = “WaitForMessagesBeforeJoin”]
             ∧ UNCHANGED ⟨triggered, fingerTables, predecessors,
                          currentMessage, kind, id, i⟩

WaitForMessagesBeforeJoin(self) ≜ ∧ pc[self] = “WaitForMessagesBeforeJoin”
                                  ∧ IF joined[self] ≠ TRUE
                                       THEN ∧ pc′ = [pc EXCEPT ![self] = “WhileWaitForMessagesBefore
                                       ELSE ∧ pc′ = [pc EXCEPT ![self] = “Stabilize”]
                                  ∧ UNCHANGED ⟨actorInboxes, triggered,
                                               fingerTables, predecessors,
                                               currentMessage, kind, id, i,
                                               joined⟩

WhileWaitForMessagesBeforeJoin(self) ≜ ∧ pc[self] = “WhileWaitForMessagesBeforeJoin”
```

```
                                      ∧ IF actorInboxes[self] ≠ ⟨⟩
                                           THEN ∧ currentMessage′ = [currentMessage EXCEPT ![self
                                                ∧ kind′ = [kind EXCEPT ![self] = Head(currentMess
                                                ∧ actorInboxes′ = [actorInboxes EXCEPT ![self] = Ta
                                           ELSE ∧ TRUE
                                                ∧ UNCHANGED ⟨actorInboxes,
                                                             currentMessage,
                                                             kind⟩
                                      ∧ pc′ = [pc EXCEPT ![self] = "ProcessMessageBeforeJoin"]
                                      ∧ UNCHANGED ⟨triggered,
                                                   fingerTables,
                                                   predecessors, id, i,
                                                   joined⟩

ProcessMessageBeforeJoin(self) ≜ ∧ pc[self] = "ProcessMessageBeforeJoin"
                                 ∧ IF kind[self] = "Predecessor"
                                      THEN ∧ predecessors′ = [predecessors EXCEPT ![self] = currentMes
                                           ∧ fingerTables′ = [fingerTables EXCEPT ![self][fingerStart(se
                                           ∧ joined′ = [joined EXCEPT ![self] = TRUE]
                                      ELSE ∧ TRUE
                                           ∧ UNCHANGED ⟨fingerTables,
                                                        predecessors,
                                                        joined⟩
                                 ∧ pc′ = [pc EXCEPT ![self] = "ReturnDefaultsBeforeJoin"]
                                 ∧ UNCHANGED ⟨actorInboxes, triggered,
                                              currentMessage, kind, id, i⟩

ReturnDefaultsBeforeJoin(self) ≜ ∧ pc[self] = "ReturnDefaultsBeforeJoin"
                                 ∧ currentMessage′ = [currentMessage EXCEPT ![self] = ⟨"?", NULL, NU
                                 ∧ kind′ = [kind EXCEPT ![self] = "?"]
                                 ∧ id′ = [id EXCEPT ![self] = NULL]
                                 ∧ pc′ = [pc EXCEPT ![self] = "WhileWaitForMessagesBeforeJoin"]
                                 ∧ UNCHANGED ⟨actorInboxes, triggered,
                                              fingerTables, predecessors,
                                              i, joined⟩

Stabilize(self) ≜ ∧ pc[self] = "Stabilize"
                  ∧ IF joined[self]
                       THEN ∧ actorInboxes′ = [actorInboxes EXCEPT ![self] = Append(actorInboxes[self], ⟨
                       ELSE ∧ TRUE
                            ∧ UNCHANGED actorInboxes
                  ∧ pc′ = [pc EXCEPT ![self] = "Done"]
                  ∧ UNCHANGED ⟨triggered, fingerTables, predecessors,
                               currentMessage, kind, id, i, joined⟩

actor(self) ≜ Join(self) ∨ WaitForMessagesBeforeJoin(self)
```

$$\begin{aligned}&\lor \mathit{WhileWaitForMessagesBeforeJoin}(\mathit{self})\\&\lor \mathit{ProcessMessageBeforeJoin}(\mathit{self})\\&\lor \mathit{ReturnDefaultsBeforeJoin}(\mathit{self}) \lor \mathit{Stabilize}(\mathit{self})\end{aligned}$$

Allow infinite stuttering to prevent deadlock on termination.

$$\begin{aligned}\mathit{Terminating} \triangleq\ &\land \forall\, \mathit{self} \in \mathit{ProcSet} : \mathit{pc}[\mathit{self}] = \text{“Done”}\\&\land \textsc{unchanged}\ \mathit{vars}\end{aligned}$$

$$\begin{aligned}\mathit{Next} \triangleq\ &(\exists\, \mathit{self} \in \{0,\ 1,\ 3\} : \mathit{actor}(\mathit{self}))\\&\quad \lor \mathit{Terminating}\end{aligned}$$

$$\begin{aligned}\mathit{Spec} \triangleq\ &\land \mathit{Init} \land \Box[\mathit{Next}]_{\mathit{vars}}\\&\land \mathrm{WF}_{\mathit{vars}}(\mathit{Next})\\&\land \forall\, \mathit{self} \in \{0,\ 1,\ 3\} : \land \mathrm{WF}_{\mathit{vars}}(\mathit{actor}(\mathit{self}))\\&\qquad\qquad\qquad\qquad\quad \land \mathrm{SF}_{\mathit{vars}}(\mathit{Join}(\mathit{self})) \land \mathrm{SF}_{\mathit{vars}}(\mathit{WaitForMessagesBeforeJoin}(\mathit{self})) \land \mathrm{SF}_{\mathit{vars}}(\mathit{Wh}\end{aligned}$$

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

END TRANSLATION

$$\mathit{Triggered} \triangleq \mathit{predecessors} = (0 :> 0) \mathbin{@@} (1 :> 0) \mathbin{@@} (3 :> 0)$$

$$\mathit{Liveness} \triangleq \Diamond \mathit{Triggered}$$

$$\mathit{LenStateConstraint} \triangleq \mathit{Len}(\mathit{actorInboxes}[0]) \leq 0 \land \mathit{Len}(\mathit{actorInboxes}[1]) \leq 0 \land \mathit{Len}(\mathit{actorInboxes}[3]) \leq 0$$

* Modification History
* Last modified Sun May 01 22:38:33 *YEKT* 2022 by *pervu*
* Created Sun *Jan* 30 18:34:11 *YEKT* 2022 by *pervu*