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

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

CONSTANTS m, bm

Clients ≜ {0, 1, 3}

  --fair algorithm ActorStuff {
variables triggered = FALSE ;
           fingerTables =  (0 :> ((1 :> 1) @@ (2 :> 3) @@ (4 :> 0)))
            @@ (1 :> ((2 :> 3) @@ (3 :> 5) @@ (5 :> 0)))
            @@ (3 :> ((4 :> 0) @@ (5 :> 0) @@ (7 :> 0))) ;
            Channels = InitChannels(Clients) ;

 ⟨"FindPredecessor", id, asker⟩

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

fair process ( actor ∈ Clients )
variables currentMessage = ⟨"?", −1, −1⟩ ;
  kind = "?" ;
  id = −1 ;
  asker = −1 ;
  i ;
{
  Channels := Send(Channels, 1, 0, ⟨"FindPredecessor", 6, 0⟩,
              "FindPredcessor60",
              "Start");
  await HasMessage(Channels, self) ;
  with ( wrapped_msg ∈ NextMessages(Channels, self) do
   with currentMessage = Payload(wrapped_msg) do
    kind := Head(currentMessage) ;
    if (kind = "FindPredecessor"){
          id := currentMessage[2] ;
          asker := currentMessage[3] ;
          if (between01(self, id, fingerTables[self][(self + 1)%bm])){
          Channels := Send(Channels, self, asker, ⟨"Predecessor", self⟩,
                "Predecessor",
                "End") ;
          }else {
            $i := m$;
            FindFirstSuitableI:
             while ($i > 0 \land \neg((self + (2^{(i-1)}))\%bm \in \text{DOMAIN } fingerTables[self])$){
              $i := i - 1$;
             };
            MainLoop:
             while ($i > 0 \land \neg(between00(self, fingerTables[self][(self + (2^{(i-1)}))\%bm], id))$){
              $i := i - 1$;
              FindSuitableI:
              while ($i > 0 \land \neg((self + (2^{(i-1)}))\%bm \in \text{DOMAIN } fingerTables[self])$){
               $i := i - 1$;
              };
             };
            if ($i = 0$){
            $Channels := Send(Channels, self, fingerTables[self][(self + (2^{(m-1)}))\%bm], currentMessage,$
                 "Transit",
                 "Transit");
             }else {
             $Channels := Send(Channels, self, fingerTables[self][(self + (2^{(i-1)}))\%bm], currentMessage,$
                 "Transit",
                 "Transit");
             }
           }
         }else {
          if ($kind =$ "Predecessor" $\land currentMessage[2] = 3$){
            call $trigger(content)$;
           $triggered :=$ TRUE;
          }
         };
   end with ;
  $Channels := MarkMessageReceived(Channels, self, wrapped\_msg,$ "Received")
  end with ;
}
 }
```

BEGIN TRANSLATION ($chksum(pcal) =$ "$afae019c$" $\land\ chksum(tla) =$ "$5415e04a$")

```
CONSTANT  $defaultInitValue$
VARIABLES $actorInboxes, triggered, fingerTables, pc, currentMessage, kind, id,$
          $asker, i$

$vars \triangleq \langle actorInboxes, triggered, fingerTables, pc, currentMessage, kind,$
          $id, asker, i\rangle$

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

$Init \triangleq$  Global variables
         $\land\ actorInboxes = (0 :> \langle\langle$"FindPredecessor"$, 6, 0\rangle\rangle)\ @@\ (1 :> \langle\rangle)\ @@\ (3 :> \langle\rangle)$
```

```
      ∧ triggered = FALSE
      ∧ fingerTables =                     (0 :> ((1 :> 1) @@ (2 :> 3) @@ (4 :> 0)))
                          @@ (1 :> ((2 :> 3) @@ (3 :> 5) @@ (5 :> 0)))
                          @@ (3 :> ((4 :> 0) @@ (5 :> 0) @@ (7 :> 0)))
       Process actor
      ∧ currentMessage = [self ∈ {0, 1, 3} ↦ ⟨"?", −1, −1⟩]
      ∧ kind = [self ∈ {0, 1, 3} ↦ "?"]
      ∧ id = [self ∈ {0, 1, 3} ↦ −1]
      ∧ asker = [self ∈ {0, 1, 3} ↦ −1]
      ∧ i = [self ∈ {0, 1, 3} ↦ defaultInitValue]
      ∧ pc = [self ∈ ProcSet ↦ "WaitForMessages"]

WaitForMessages(self) ≜ ∧ pc[self] = "WaitForMessages"
                        ∧ IF actorInboxes[self] ≠ ⟨⟩
                              THEN ∧ currentMessage′ = [currentMessage EXCEPT ![self] = Head(actorIn
                                   ∧ kind′ = [kind EXCEPT ![self] = Head(currentMessage′[self])]
                                   ∧ actorInboxes′ = [actorInboxes EXCEPT ![self] = Tail(actorInboxes[s
                              ELSE ∧ TRUE
                                   ∧ UNCHANGED ⟨actorInboxes,
                                                currentMessage, kind⟩
                        ∧ pc′ = [pc EXCEPT ![self] = "ProcessMessage"]
                        ∧ UNCHANGED ⟨triggered, fingerTables, id, asker, i⟩

ProcessMessage(self) ≜ ∧ pc[self] = "ProcessMessage"
                       ∧ IF kind[self] = "FindPredecessor"
                             THEN ∧ id′ = [id EXCEPT ![self] = currentMessage[self][2]]
                                  ∧ asker′ = [asker EXCEPT ![self] = currentMessage[self][3]]
                                  ∧ IF between01(self, id′[self], fingerTables[self][(self + 1)%bm])
                                        THEN ∧ actorInboxes′ = [actorInboxes EXCEPT ![asker′[self]] = A
                                             ∧ pc′ = [pc EXCEPT ![self] = "DefaultsBack"]
                                             ∧ i′ = i
                                        ELSE ∧ i′ = [i EXCEPT ![self] = m]
                                             ∧ pc′ = [pc EXCEPT ![self] = "FindFirstSuitableI"]
                                             ∧ UNCHANGED actorInboxes
                                  ∧ UNCHANGED triggered
                             ELSE ∧ IF kind[self] = "Predecessor" ∧ currentMessage[self][2] = 3
                                        THEN ∧ triggered′ = TRUE
                                        ELSE ∧ TRUE
                                             ∧ UNCHANGED triggered
                                  ∧ pc′ = [pc EXCEPT ![self] = "DefaultsBack"]
                                  ∧ UNCHANGED ⟨actorInboxes, id, asker, i⟩
                       ∧ UNCHANGED ⟨fingerTables, currentMessage, kind⟩

FindFirstSuitableI(self) ≜ ∧ pc[self] = "FindFirstSuitableI"
                           ∧ IF i[self] > 0 ∧ ¬((self + (2^(i[self]−1)))%bm ∈ DOMAIN fingerTables[self])
                                 THEN ∧ i′ = [i EXCEPT ![self] = i[self] − 1]
```

```
                                        $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{"FindFirstSuitableI"}]$
                                 $\text{ELSE}\ \land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{"MainLoop"}]$
                                        $\land\ i' = i$
                         $\land\ \text{UNCHANGED}\ \langle actorInboxes, triggered,$
                                        $fingerTables, currentMessage, kind,$
                                        $id, asker\rangle$

$MainLoop(self) \triangleq\ \land\ pc[self] = \text{"MainLoop"}$
                  $\land\ \text{IF}\ i[self] > 0 \land \neg(between00(self, fingerTables[self][(self + (2^{(i[self]-1)}))\%bm], id[self]))$
                        $\text{THEN}\ \land\ i' = [i\ \text{EXCEPT}\ ![self] = i[self] - 1]$
                               $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{"FindSuitableI"}]$
                               $\land\ \text{UNCHANGED}\ actorInboxes$
                        $\text{ELSE}\ \land\ \text{IF}\ i[self] = 0$
                                   $\text{THEN}\ \land\ actorInboxes' = [actorInboxes\ \text{EXCEPT}\ ![fingerTables[self][(sel$
                                   $\text{ELSE}\ \land\ actorInboxes' = [actorInboxes\ \text{EXCEPT}\ ![fingerTables[self][(sel$
                               $\land\ pc' = [pc\ \text{EXCEPT}\ ![self]\ = \text{"DefaultsBack"}]$
                               $\land\ i' = i$
                  $\land\ \text{UNCHANGED}\ \langle triggered, fingerTables, currentMessage,$
                                 $kind, id, asker\rangle$

$FindSuitableI(self) \triangleq\ \land\ pc[self] = \text{"FindSuitableI"}$
                      $\land\ \text{IF}\ i[self] > 0 \land \neg((self + (2^{(i[self]-1)}))\%bm \in \text{DOMAIN}\ fingerTables[self])$
                            $\text{THEN}\ \land\ i' = [i\ \text{EXCEPT}\ ![self] = i[self] - 1]$
                                   $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{"FindSuitableI"}]$
                            $\text{ELSE}\ \land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{"MainLoop"}]$
                                   $\land\ i' = i$
                      $\land\ \text{UNCHANGED}\ \langle actorInboxes, triggered, fingerTables,$
                                     $currentMessage, kind, id, asker\rangle$

$DefaultsBack(self) \triangleq\ \land\ pc[self] = \text{"DefaultsBack"}$
                     $\land\ currentMessage' = [currentMessage\ \text{EXCEPT}\ ![self] = \langle\text{"?"}, -1, -1\rangle]$
                     $\land\ kind' = [kind\ \text{EXCEPT}\ ![self] = \text{"?"}]$
                     $\land\ id' = [id\ \text{EXCEPT}\ ![self] = -1]$
                     $\land\ asker' = [asker\ \text{EXCEPT}\ ![self] = -1]$
                     $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{"WaitForMessages"}]$
                     $\land\ \text{UNCHANGED}\ \langle actorInboxes, triggered, fingerTables, i\rangle$

$actor(self) \triangleq\ WaitForMessages(self) \lor ProcessMessage(self)$
                 $\lor\ FindFirstSuitableI(self) \lor MainLoop(self)$
                 $\lor\ FindSuitableI(self) \lor DefaultsBack(self)$

$Next \triangleq (\exists\, self \in \{0, 1, 3\} : actor(self))$

$Spec \triangleq\ \land\ Init \land \Box[Next]_{vars}$
           $\land\ \text{WF}_{vars}(Next)$
           $\land\ \forall\, self \in \{0, 1, 3\} : \text{WF}_{vars}(actor(self)) \land \text{SF}_{vars}(WaitForMessages(self))$
```

END TRANSLATION

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

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

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

* Modification History
* Last modified Sun *Feb* 20 23:17:30 *YEKT* 2022 by *pervu*
* Created Sun *Jan* 30 18:34:11 *YEKT* 2022 by *pervu*