```
──────────────────── MODULE Mutex_acks ────────────────────
EXTENDS Naturals, Sequences, FiniteSets, TLC

CONSTANTS N, MaxClock

Pid ≜ 1 . . N
ClockVal ≜ 0 . . MaxClock + 1
Message ≜ [time : ClockVal, type : {"Request", "Release", "AckReq"}]

  --algorithm LogicalClocks

variables
  channel = [source ∈ Pid ↦ [destination ∈ Pid ↦ ⟨⟩]],
  crit = {}

define
  LogClockLt(reqs, p, q) ≜
    ∨ reqs[q] = 0
    ∨ reqs[p] < reqs[q]
    ∨ reqs[p] = reqs[q] ∧ p < q

  ChanHead(dst, type) ≜
    {src ∈ Pid : ∧ Len(channel[src][dst]) > 0
                 ∧ Head(channel[src][dst]).type = type
    }

  ChanHeadP(dst) ≜
    {src ∈ Pid : ∧ Len(channel[src][dst]) > 0
    }

  Max(a, b) ≜ IF a ≤ b THEN b ELSE a
end define

macro Receive(type, clock, src, time)begin
  with s ∈ ChanHead(self, type) do
    src := s ;
    time := Head(channel[src][self]).time ;
    channel := [channel EXCEPT ![src][self] = Tail(channel[src][self])]
  end with
end macro

macro Broadcast(clock, msg)begin
  channel :=
    [channel EXCEPT ![self] =
      [dst ∈ Pid ↦
        IF dst = self THEN channel[self][self]
                      ELSE Append(channel[self][dst], msg)]]
end macro
```

```
macro $SendTo(clock, dst, msg)$ begin
  $channel :=$
    $[channel \text{ EXCEPT } ![self][dst] = Append(channel[self][dst], msg)]$
end macro

macro $EnterCritSec()$ begin
  $crit := crit \cup \{self\}$
end macro

macro $ExitCritSec()$ begin
  $crit := crit \setminus \{self\}$
end macro

process $Proc \in Pid$
variables
  $clock = 1,$
  $acks = \{\},$
  $requests = [pid \in Pid \mapsto 0],$
  $time,$
  $src$

begin
  $loop$: while TRUE do
      either
          when $requests[self] = 0$ ;
            $Broadcast(clock, [time \mapsto clock, type \mapsto \text{“Request”}])$ ;
            $requests := [requests \text{ EXCEPT } ![self] = clock]$ ;
            $acks := \{self\}$
      or
          $Receive(\text{“AckReq”}, clock, src, time)$ ;
            $clock := Max(clock, time)$ ;
            $acks := acks \cup \{src\}$
      or
          when $\land\ self \notin crit$
               $\land\ acks = Pid$
         $\land\ acks \in \text{SUBSET } Pid$
               $\land\ \forall\, p \in Pid : p \neq self \Rightarrow$
                                    $LogClockLt(requests, self, p)$ ;
            $EnterCritSec()$ ;
      or
          when $self \in crit$ ;
            $requests := [requests \text{ EXCEPT } ![self] = 0]$ ;
            $ExitCritSec()$ ;
            $acks := \{\}$ ;
            $Broadcast(clock, [time \mapsto clock, type \mapsto \text{“Release”}])$
    or
      $Receive(\text{“Request”}, clock, src, time)$ ;
        $requests := [requests \text{ EXCEPT } ![src] = time]$ ;
        $clock := Max(clock, time)$ ;
     $L2$: $SendTo(clock, src, [time \mapsto clock + 1, type \mapsto \text{“AckReq”}])$
    or
      $Receive(\text{“Release”}, clock, src, time)$ ;
        $clock := Max(clock, time)$ ;
        $requests := [requests \text{ EXCEPT } ![src] = 0]$ ;
   or
     with $s \in ChanHeadP(self)$ do
       $channel :=$ [channel EXCEPT $![s][self] = Tail(channel[s][self])$]
     end with
    end either ;
    $tic$:   $clock := clock + 1$
  end while ;
end process

end algorithm
```

```
 BEGIN TRANSLATION ($chksum(pcal) =$ “$2db1399c$” $\wedge$ $chksum(tla) =$ “$dc4dbd89$”)
CONSTANT $defaultInitValue$
VARIABLES $channel, crit, pc$

 define statement
$LogClockLt(reqs, p, q) \triangleq$
  $\vee reqs[q] = 0$
  $\vee reqs[p] < reqs[q]$
  $\vee reqs[p] = reqs[q] \wedge p < q$

$ChanHead(dst, type) \triangleq$
  $\{src \in Pid :$ $\wedge Len(channel[src][dst]) > 0$
                  $\wedge Head(channel[src][dst]).type = type$
  $\}$

$ChanHeadP(dst) \triangleq$
  $\{src \in Pid :$ $\wedge Len(channel[src][dst]) > 0$
  $\}$

$Max(a, b) \triangleq \text{IF } a \leq b \text{ THEN } b \text{ ELSE } a$

VARIABLES $clock, acks, requests, time, src$

$vars \triangleq \langle channel, crit, pc, clock, acks, requests, time, src \rangle$

$ProcSet \triangleq (Pid)$

$Init \triangleq$  Global variables
```

```
              $\land\ channel = [source \in Pid \mapsto [destination \in Pid \mapsto \langle\rangle]]$
              $\land\ crit = \{\}$
               Process $Proc$
              $\land\ clock = [self \in Pid \mapsto 1]$
              $\land\ acks = [self \in Pid \mapsto \{\}]$
              $\land\ requests = [self \in Pid \mapsto [pid \in Pid \mapsto 0]]$
              $\land\ time = [self \in Pid \mapsto defaultInitValue]$
              $\land\ src = [self \in Pid \mapsto defaultInitValue]$
              $\land\ pc = [self \in ProcSet \mapsto \text{“loop”}]$

$loop(self) \triangleq$  $\land\ pc[self] = \text{“loop”}$
                  $\land\ \lor\ \land\ requests[self][self] = 0$
                        $\land\ channel' = [channel\ \text{EXCEPT}\ ![self] =$
                                       $[dst \in Pid \mapsto$
                                         $\text{IF}\ dst = self\ \text{THEN}\ channel[self][self]$
                                                          $\text{ELSE}\ \ Append(channel[self][dst], ([time \mapsto clock[self], type \mapsto$
                        $\land\ requests' = [requests\ \text{EXCEPT}\ ![self] = [requests[self]\ \text{EXCEPT}\ ![self] = clock[self]]]$
                        $\land\ acks' = [acks\ \text{EXCEPT}\ ![self] = \{self\}]$
                        $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“tic”}]$
                        $\land\ \text{UNCHANGED}\ \langle crit, clock, time, src\rangle$
                     $\lor\ \land\ \exists\, s \in ChanHead(self, \text{“AckReq”}):$
                             $\land\ src' = [src\ \text{EXCEPT}\ ![self] = s]$
                             $\land\ time' = [time\ \text{EXCEPT}\ ![self] = Head(channel[src'[self]][self]).time]$
                             $\land\ channel' = [channel\ \text{EXCEPT}\ ![src'[self]][self] = Tail(channel[src'[self]][self])]$
                        $\land\ clock' = [clock\ \text{EXCEPT}\ ![self] = Max(clock[self], time'[self])]$
                        $\land\ acks' = [acks\ \text{EXCEPT}\ ![self] = acks[self] \cup \{src'[self]\}]$
                        $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“tic”}]$
                        $\land\ \text{UNCHANGED}\ \langle crit, requests\rangle$
                     $\lor\ \land\ \land\ self \notin crit$
                           $\land\ acks[self] = Pid$

                           $\land\ \forall\, p \in Pid : p \neq self \Rightarrow$
                                        $LogClockLt(requests[self], self, p)$
                        $\land\ crit' = (crit \cup \{self\})$
                        $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“tic”}]$
                        $\land\ \text{UNCHANGED}\ \langle channel, clock, acks, requests, time, src\rangle$
                     $\lor\ \land\ self \in crit$
                        $\land\ requests' = [requests\ \text{EXCEPT}\ ![self] = [requests[self]\ \text{EXCEPT}\ ![self] = 0]]$
                        $\land\ crit'\ = crit \setminus \{self\}$
                        $\land\ acks' = [acks\ \text{EXCEPT}\ ![self] = \{\}]$
                        $\land\ channel' = [channel\ \text{EXCEPT}\ ![self] =$
                                       $[dst \in Pid \mapsto$
                                         $\text{IF}\ dst = self\ \text{THEN}\ channel[self][self]$
                                                          $\text{ELSE}\ \ Append(channel[self][dst], ([time \mapsto clock[self], type \mapsto$
                        $\land\ pc' = [pc\ \text{EXCEPT}\ ![self] = \text{“tic”}]$
```

```
                ∧ UNCHANGED ⟨clock, time, src⟩
             ∨ ∧ ∃ s ∈ ChanHead(self, "Request") :
                    ∧ src′ = [src EXCEPT ![self] = s]
                    ∧ time′ = [time EXCEPT ![self] = Head(channel[src′[self]][self]).time]
                    ∧ channel′ = [channel EXCEPT ![src′[self]][self] = Tail(channel[src′[self]][self])]
                ∧ requests′ = [requests EXCEPT ![self] = [requests[self] EXCEPT ![src′[self]] = time′[self]]]
                ∧ clock′ = [clock EXCEPT ![self] = Max(clock[self], time′[self])]
                ∧ pc′ = [pc EXCEPT ![self] = "L2"]
                ∧ UNCHANGED ⟨crit, acks⟩
             ∨ ∧ ∃ s ∈ ChanHead(self, "Release") :
                    ∧ src′ = [src EXCEPT ![self] = s]
                    ∧ time′ = [time EXCEPT ![self] = Head(channel[src′[self]][self]).time]
                    ∧ channel′ = [channel EXCEPT ![src′[self]][self] = Tail(channel[src′[self]][self])]
                ∧ clock′ = [clock EXCEPT ![self] = Max(clock[self], time′[self])]
                ∧ requests′ = [requests EXCEPT ![self] = [requests[self] EXCEPT ![src′[self]] = 0]]
                ∧ pc′ = [pc EXCEPT ![self] = "tic"]
                ∧ UNCHANGED ⟨crit, acks⟩

tic(self) ≜ ∧ pc[self] = "tic"
            ∧ clock′ = [clock EXCEPT ![self] = clock[self] + 1]
            ∧ pc′ = [pc EXCEPT ![self] = "loop"]
            ∧ UNCHANGED ⟨channel, crit, acks, requests, time, src⟩

L2(self) ≜ ∧ pc[self]  = "L2"
           ∧ channel′ = [channel EXCEPT ![self][src[self]] = Append(channel[self][src[self]], ([time ↦ clock
           ∧ pc′ = [pc EXCEPT ![self] = "tic"]
           ∧ UNCHANGED ⟨crit, clock, acks, requests, time, src⟩

Proc(self) ≜ loop(self) ∨ tic(self) ∨ L2(self)

Next ≜ (∃ self ∈ Pid : Proc(self))

Spec ≜ Init ∧ □[Next]_vars

 END TRANSLATION

View ≜ ⟨channel, crit, clock, acks, requests, pc⟩

MutualExclusion ≜ Cardinality(crit) < 2
```