```
┌──────────────────── MODULE VectorClocks ────────────────────┐
EXTENDS Integers, TLC, Sequences
CONSTANTS Procs, MAX

--algorithm VectorClocks
variables
  msgs = [p ∈ Procs ↦ [q ∈ Procs ↦ 0]] ;   defined as Vector Clock

define
    returns the maximum value for each element of two vectors
  Max(v1, v2) ≜ [p ∈ Procs ↦ IF v1[p] > v2[p] THEN v1[p] ELSE v2[p]]
end define ;

fair process VectorClock ∈ Procs
variables
  vc = [p ∈ Procs ↦ 0]  Initially all clocks are zero
begin Main:
  while vc[self] < MAX do
    either Internal:  increments local clock
      vc[self] := vc[self] + 1 ;
    or Send:  increments local clock and sends it to another process
      vc[self] := vc[self] + 1 ;
      with p ∈ Procs \ {self} do   send vc to 'p' via msgs[p]
        msgs[p] := vc ;
      end with ;
    or Receive:  increments local clock and calcs maximum of two clocks
      vc := Max(vc, msgs[self]) ;
      goto Internal ;   vc[self] := vc[self] + 1;
    end either ;
  end while ;
end process ;

end algorithm   ;
 BEGIN TRANSLATION
VARIABLES msgs, pc

 define statement
Max(v1, v2) ≜ [p ∈ Procs ↦ IF v1[p] > v2[p] THEN v1[p] ELSE v2[p]]

VARIABLE vc

vars ≜ ⟨msgs, pc, vc⟩

ProcSet ≜ (Procs)

Init ≜  Global variables
          ∧ msgs = [p ∈ Procs ↦ [q ∈ Procs ↦ 0]]
           Process VectorClock
           $\land\ vc = [self \in Procs \mapsto [p \in Procs \mapsto 0]]$
           $\land\ pc = [self \in ProcSet \mapsto \text{"Main"}]$

$Main(self) \triangleq$  $\land\ pc[self] = \text{"Main"}$
                  $\land$ IF $vc[self][self] < MAX$
                        THEN  $\land \lor \land\ pc' = [pc$ EXCEPT $![self] = \text{"Internal"}]$
                                 $\lor \land\ pc' = [pc$ EXCEPT $![self] = \text{"Send"}]$
                                 $\lor \land\ pc' = [pc$ EXCEPT $![self] = \text{"Receive"}]$
                        ELSE  $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Done"}]$
                  $\land$ UNCHANGED $\langle msgs, vc \rangle$

$Internal(self) \triangleq$  $\land\ pc[self] = \text{"Internal"}$
                     $\land\ vc' = [vc$ EXCEPT $![self][self] = vc[self][self] + 1]$
                     $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Main"}]$
                     $\land\ msgs' = msgs$

$Send(self) \triangleq$  $\land\ pc[self] = \text{"Send"}$
                 $\land\ vc' = [vc$ EXCEPT $![self][self] = vc[self][self] + 1]$
                 $\land \exists\, p \in Procs \setminus \{self\}:$
                      $msgs' = [msgs$ EXCEPT $![p] = vc'[self]]$
                 $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Main"}]$

$Receive(self) \triangleq$  $\land\ pc[self] = \text{"Receive"}$
                    $\land\ vc' = [vc$ EXCEPT $![self] = Max(vc[self], msgs[self])]$
                    $\land\ pc' = [pc$ EXCEPT $![self] = \text{"Internal"}]$
                    $\land\ msgs' = msgs$

$VectorClock(self) \triangleq Main(self) \lor Internal(self) \lor Send(self)$
                          $\lor\ Receive(self)$

$Next \triangleq (\exists\, self \in Procs : VectorClock(self))$
           $\lor$  Disjunct to prevent deadlock on termination
             $((\forall\, self \in ProcSet : pc[self] = \text{"Done"}) \land$ UNCHANGED $vars)$

$Spec \triangleq \land\ Init \land \Box[Next]_{vars}$
         $\land \forall\, self \in Procs : \mathrm{WF}_{vars}(VectorClock(self))$

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

 END TRANSLATION

 Invariants
$VectorClockOK \triangleq (\forall\, k, l \in Procs : vc[k][k] \geq vc[l][k])$
```

* Modification History
* Last modified Sun *Nov* 25 21:39:03 *PST* 2018 by *ocosta*
* Created Sat *Nov* 17 15:07:39 *PST* 2018 by *ocosta*