

---

MODULE *RoundRobin*

---

EXTENDS *Naturals*

VARIABLES *PC*, *CHAN*, *PHONE*, *ACCESSCOUNT*

CONSTANT *NodeCount*

---

*IDs*  $\triangleq 0 \dots \text{NodeCount} - 1$

Channel is instantiated as a boolean, representing either an empty or full buffer (containing the token) Phone is the shared resource. Access Count is used to assert that no more than one process uses the *resource(phone)* at once.

*TypeInvariant*  $\triangleq \begin{aligned} &\wedge PC \in [IDs \rightarrow 1 \dots 4] \\ &\wedge CHAN \in [IDs \rightarrow \text{BOOLEAN}] \\ &\wedge PHONE \in [IDs \rightarrow \text{BOOLEAN}] \\ &\wedge ACCESSCOUNT \in (0 \dots \text{NodeCount}) \\ &\wedge ACCESSCOUNT \leq 1 \end{aligned}$

---

*Init*  $\triangleq \begin{aligned} &\wedge PC = [p \in IDs \mapsto 1] \\ &\wedge CHAN = [[p \in IDs \mapsto \text{FALSE}] \text{ EXCEPT } ![0] = \text{TRUE}] \\ &\wedge PHONE = [p \in IDs \mapsto \text{FALSE}] \\ &\wedge ACCESSCOUNT = 0 \end{aligned}$

Allows for processes to wait

*WaitForToken(id)*  $\triangleq \begin{aligned} &\wedge PC[id] = 1 \\ &\wedge CHAN[id] = \text{FALSE} \\ &\wedge \text{UNCHANGED } \langle PHONE, CHAN, PC, ACCESSCOUNT \rangle \end{aligned}$

Allows for processes to receive the token when their channel is set to true

*ReceiveToken(id)*  $\triangleq \begin{aligned} &\wedge PC[id] = 1 \\ &\wedge CHAN[id] = \text{TRUE} \\ &\wedge PC' = [PC \text{ EXCEPT } ![id] = 2] \\ &\wedge CHAN' = [CHAN \text{ EXCEPT } ![id] = \text{FALSE}] \\ &\wedge \text{UNCHANGED } \langle PHONE, ACCESSCOUNT \rangle \end{aligned}$

Allows for a process to pick up the phone and increase the accesscount

*PickUpPhone(id)*  $\triangleq \begin{aligned} &\wedge PC[id] = 2 \\ &\wedge PC' = [PC \text{ EXCEPT } ![id] = 3] \\ &\wedge PHONE' = [PHONE \text{ EXCEPT } ![id] = \text{TRUE}] \\ &\wedge ACCESSCOUNT' = ACCESSCOUNT + 1 \\ &\wedge \text{UNCHANGED } \langle CHAN \rangle \end{aligned}$

Decreases the accesscount and stops using the phone

*HangUpPhone(id)*  $\triangleq \wedge PC[id] = 3$

$$\begin{aligned}
& \wedge PHONE' = [PHONE \text{ EXCEPT } ![id] = \text{FALSE}] \\
& \wedge PC' = [PC \text{ EXCEPT } ![id] = 4] \\
& \wedge ACCESSCOUNT' = ACCESSCOUNT - 1 \\
& \wedge \text{UNCHANGED } \langle CHAN \rangle
\end{aligned}$$

Allows for a process to pass along the token, setting its own channel to false and the subsequent channel to true

$$\begin{aligned}
SendToken(id, idNext) & \triangleq \wedge PC[id] = 4 \\
& \wedge CHAN' = [CHAN \text{ EXCEPT } ![id] = \text{FALSE}, ![idNext] = \text{TRUE}] \\
& \wedge PC' = [PC \text{ EXCEPT } ![id] = 1] \\
& \wedge \text{UNCHANGED } \langle PHONE, ACCESSCOUNT \rangle
\end{aligned}$$

$$\begin{aligned}
Proc(n, x) & \triangleq \vee WaitForToken(n) \\
& \vee ReceiveToken(n) \\
& \vee PickUpPhone(n) \\
& \vee HangUpPhone(n) \\
& \vee SendToken(n, x)
\end{aligned}$$

$$Next \triangleq \exists n \in IDs : Proc(n, (n+1)\%NodeCount)$$

Liveness property, securing that if the process counter of a process is 1, then eventually the process will receive the token and access the phone

$$NoStarvation \triangleq \forall n \in IDs : \Box(PC[n] = 1 \leadsto PHONE[n] = \text{TRUE})$$

---


$$Spec \triangleq Init \wedge \Box[Next]_{\langle PC, CHAN, PHONE, ACCESSCOUNT \rangle} \wedge NoStarvation$$


---

\ \* Modification History  
\ \* Last modified Tue Apr 17 09:42:03 CEST 2018 by jensk  
\ \* Created Mon Apr 16 12:30:39 CEST 2018 by jensk