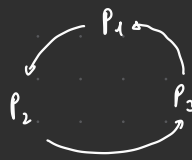


Min 3 procesos

- $S.C \rightarrow P_1$
- Esperando para entrar $\rightarrow P_3$
- No quiere entrar $\rightarrow P_2$



* Todos los nodos hacen lo mismo

int testigo = 1, pido-testigo = 0, ID, dentro_SC = 0
 int[] peticiones, atendida, id_nodos

Proceso 1

if (NOT testigo)

pido-testigo = pido-testigo + 1
 send(ID, pido-testigo)
 receive(TOKEN, atendida)
 testigo = 1

if (testigo = 1)

dentro_SC = 1
 S.C
 atendida[ID] = pido-testigo
 dentro_SC = 0

if (peticiones > atendida)

send-token(id_nodo)

Receptor:

int nodo-orig, pet-orig = 0
 receive(nodo-orig, pet-orig)
 peticiones[nodo-orig] = MAX(peticiones[nodo-orig], pet-orig)
 if (testigo AND (NOT dentro) AND peticiones > atendida)
 send-token(nodo-orig)

Proceso 3

if (NOT testigo)

pido-testigo = pido-testigo + 1
 send(ID, pido-testigo)
 receive(TOKEN, atendida)
 testigo = 1

if (testigo = 1)

dentro_SC = 1
 S.C
 atendida[ID] = pido-testigo
 dentro_SC = 0

if (peticiones > atendida)

send-token(id_nodo)

Receptor:

int nodo-orig, pet-orig = 0
 receive(nodo-orig, pet-orig)
 peticiones[nodo-orig] = MAX(peticiones[nodo-orig], pet-orig)
 if (testigo AND (NOT dentro) AND peticiones > atendida)
 send-token(nodo-orig)

Proceso 2

if (NOT testigo)

pido-testigo = pido-testigo + 1
 send(ID, pido-testigo)
 receive(TOKEN, atendida)
 testigo = 1

if (testigo = 1)

dentro_SC = 1
 S.C
 atendida[ID] = pido-testigo
 dentro_SC = 0

if (peticiones > atendida)

send-token(id_nodo)

Receptor:

int nodo-orig, pet-orig = 0
 receive(nodo-orig, pet-orig)
 peticiones[nodo-orig] = MAX(peticiones[nodo-orig], pet-orig)
 if (testigo AND (NOT dentro) AND peticiones > atendida)
 send-token(nodo-orig)