

(a)

$P_i::$

var

q : queue of (int, pid) initially *null*;

$numAcks$: integer initially 0;

$numInCS$: integer initially 0;

k : constant integer

// Also assumes logical clock algorithm running

request:

send *request* with (*logicalClock*, i) to all other processes;

insert (*logicalClock*, i) in q ;

$numAcks := 0$;

On *receive*(*request*, (ts , j)) from P_j :

insert (ts , j) in q ;

send (*ack*, *logicalClock*) to P_j ;

On *receive*(*ack*, ts):

$numAcks := numAcks + 1$;

if ($numAcks = N - 1$) and P_i 's request smallest in q and ($numInCS < k$) **then**
enter;

On *receive*(*release*):

$numInCS := numInCS - 1$;

if ($numAcks = N - 1$) and P_i 's request smallest in q and ($numInCS < k$) **then**
enter;

enter:

delete the request by P_i from q

send *enter* to all processes

enter_critical_section;

On *receive*(*enter*) from P_j :

$numInCS := numInCS + 1$;

delete the request by P_j from q

if ($numAcks = N - 1$) and P_i 's request smallest in q and ($numInCS < k$) **then**
enter;

release:
 send *release* to all processes

(b)

P_i::

var

pendingQ: list of process ids initially *null*;
 myts: integer initially ∞ ;
 numOkay: integer initially 0;
 numInCS: integer initially 0;
 k: constant integer

request:

myts := *logical_clock*;
 send *request* with *myts* to all other processes;
 numOkay := 0;

On *receive(request, hists)* from *P_j*:

if (*hists, j*) < (*myts, i*) **then**
 send *okay* with *false* to process *P_j*;
 else *append(pendingQ, j)*;

On *receive(okay, entered)*:

numOkay := *numOkay* + 1;
 if (*entered* = *true*) **then** *numInCS* := *numInCS* + 1;
 if (*numOkay* = *N* - 1) and (*numInCS* < *k*) **then** *enter*;

enter:

myts := ∞ ;
 for *j* \in *pendingQ* **do**
 send *okay* with *true* to the process *j*;
 enter_critical_section;

On *receive(enter)*:

numInCS := *numInCS* + 1;
 if (*numOkay* = *N* - 1) and (*numInCS* < *k*) **then** *enter*;

On *receive(release)*:

numInCS := *numInCS* - 1;

if ($numOkay = N - 1$) and ($numInCS < k$) **then enter**;

release:

for $j \in pendingQ$ **do**

 send *release* to the process j ;

$pendingQ := null$;