1 - Algo Ricart et Agrawala (1983)

- var état = (dehors ; demandeur ; dedans) init à dehors ;
  h_i : entier init 0 ;
  $last_i$: " " 0 ;
  $priorité_i$: Booleen init faux ;
  $nbrtt_i$: entier init 0 ;
  $diffare_i$: ensemble de id_proc init ϕ (ensemble vide)
  R_i : " " " " $i \in \{1, 2, \ldots, n\} - \{i\}$

- proc acquérir
  début
  $etat_i$:= demandeur ;
  $h_i := h_i + 1$; $last_i := h_i$;
  $nbrtt_i$ = card (R_i) ;
  pourtt j de R_i faire
  | envoyer requete($last_i$, i) a C_j
  fin pourtt
  attendre $nbrtt_i = 0$;
  $etat_i := dedans$;
  fin

- proc liberer
  début
  pourtt j de $diffare_i$ faire
  | envoyer permission a C_j
  fin pourtt
  $diffare_i := \phi$
  $etat_i$:= dehors.
  fin

- lors de reception requete(K, j)
  début
  $h_i := max(h_i, K)$
  $priorité_i := (etat_i \neq dehors) \wedge ((last_i, i) < (K, j))$
  Si (priorité) alors
  | $diffare_i := diffare_i \cup \{j\}$
  Sinon
  | envoyer permission a C_j ;
  Fs ;
  Fin

- lors de reception permission de j
  début
  | $nbrtt_i := nbrtt_i - 1$
  Fin

2 - Algo Carvalho-Roucairol (1983):

var $etat_i$: (dehors, demandeur, dedans) init dehors ;
h_i : entier init à 0 ;
$last_i$: entier init a à 0 ;
$priorité_i$: Booleen init à faux
$diffare_i$: ensemble de id_proc init à ϕ.
R_i : ensemble de id-proc init $i \in R_j$ OUEXCL $j \in R_i$

- proc acquérir
  début
    $etat_i$:= demandeur ;
    $last_i = h_i + 1$;
    pourtt j de R_i faire
      envoyer requete($last_i$, i) a C_j ;
    Finpourtt
    attendre $R_i = \phi$;
    $etat_i$:= dedans ;
  Fin

- proc libérer
  début
    pourtt j de $diffare_i$ faire
      envoyer permission à C_j ;
    Finpourtt
    R_i := $diffare_i$;
    $diffare_i$:= ϕ ;
    $etat_i$:= dehors
  Fin

- lors de reception permission de j
  début
    R_i := $R_i - \{j\}$
  Fin

- lors de reception requete (k, j)
  début
    h_i := max(h_i; k)
    $priorité_i$:= ($etat_i$ = dedans) $\vee$ ((etat = demandeur) $\wedge$ ($last_i$; i) < (k; j))
    Si priorité alors
      $diffare_i$:= $diffare_i \cup \{j\}$
    sinon
      envoyer permission à C_j
      $R_i = R_i \cup \{j\}$
      Si ($etat_i$ = demandeur) alors
        envoyer requete($last_i$; i) a C_j
      Fsi
    Fsi
  Fin