SISTEMI OPERATIVI

ESERCIZIO N. 1 del 17 NOVEMBRE 2000

In un **centro per il prelievo** del sangue lavora un **medico** che ha a disposizione **L lettini**. Le **persone** che effettuano il prelievo si dividono in due categorie: **donatori** e **pazienti**. Ogni persona può iniziare il prelievo solo quando è disponibile il medico e c'è almeno un lettino vuoto, altrimenti aspetta. Dopo che il medico ha iniziato il prelievo, la persona aspetta che il medico finisca il prelievo. Terminato il prelievo, dopo essersi ripresa, la persona libera il lettino. Nella soluzione si tenga presente che i donatori hanno la precedenza sui pazienti.
Si implementi una soluzione usando il costrutto monitor per modellare il **centro prelievi** e i processi per modellare il **medico** e le **persone** e si descriva la sincronizzazione tra i processi. Nella soluzione si massimizzi l'utilizzo delle risorse. Si discuta se la soluzione proposta può presentare starvation e in caso positivo per quali processi, e si propongano modifiche e/o aggiunte per evitare starvation.

```
program ospedale

const L = ...; { numero di lettini }
type   tipo = (D, P);  { Donatore e Paziente }

type medico = process
begin
        repeat
          centro.inizio_prelievo;
          <effettua il prelievo>
          centro.fine_prelievo;
        until false
end

type persona = process (t: tipo)
begin
        centro.entra (t);
        <riprenditi>
        centro.esci;
end

type centro_prelievi = monitor

{ variabili del monitor }
var  lettini_occupati : integer;
      { numero di lettini occupati }
      coda_medico : condition;
      { coda su cui sospendere il medico }
      coda_fuori : array[tipo] of condition;
      { persone che aspettano un lettino }
      coda_dentro : array[tipo] of condition;
      { persone che aspettano il medico }
      prelievo: condition;
      { coda su cui sospendere chi è in prelievo }
```

```
procedure entry entra (t: tipo)
begin
      if lettini_occupati = L then
      { controllo se c'è un lettino libero }
      { non c'è bisogno di controllare la priorità }
            coda_fuori [t].wait;

      lettini_occupati ++; { sono entrato }

      if not coda_medico.queue then
      { controllo la disponibilità del medico }
      { se il medico è occupato (= non sospeso) }
      { mi sospendo }
            coda_dentro[t].wait;
      else
      { se il medico è sospeso, viene risvegliato }
            coda_medico.signal;

      prelievo.wait; { aspetto la fine del prelievo }
end

procedure entry esci
begin
      lettini_occupati -- ;
      { risveglio una sola persona }
      { dando priorità ai donatori }
      if coda_fuori [D].queue
      then
            coda_fuori [D].signal;
      else
            coda_fuori [P].signal;
end
```

```
procedure entry inizio_prelievo
var s, i: integer;
begin
      if   not (coda_dentro[D].queue or
            coda_dentro[P].queue) then
      { mi sospendo se non c'è nessuna persona in coda }
               coda_medico.wait;
      else { vedi considerazioni }
      { risveglio una persona }
            if coda_dentro[D].queue then
                  coda_dentro[D].signal;
            else
                  coda_dentro[P].signal;
end

procedure entry fine_prelievo
begin
      prelievo.signal;
end

begin { inizializzazione delle variabili }
      lettini_occupati := 0;
end

var  centro: centro_prelievi; { il nostro monitor }
      d1, d2, ... : persona (D);
      p1, p2, ... : persona (P);
      m : medico;

begin end.
```

Considerazioni

Non c'è un flag per dire se il medico è libero o occupato, ma viene controllata la coda: medico in coda => libero; medico non in coda => occupato.

Il risveglio delle persone deve essere fatto in alternativa alla sospensione del medico, poiché il medico viene risvegliato da una persona, per cui non deve risvegliarne un'altra.

Starvation

La soluzione proposta presenta starvation, infatti è possibile che i donatori non lascino entrare i pazienti.

Un modo per evitare starvation è quello di utilizzare dei contatori, in modo che dopo che sono passati N donatori, si facciano entrare i pazienti.