Copyright © 2007 – The McGraw-Hill Companies srl

Esercizio E3.1

Impostazione

Si suppone che ogni processo consumatore disponga di una propria porta di tipo `T` (`dati`) da cui ricevere i messaggi. Quando un consumatore è pronto a ricevere un messaggio invia un segnale al processo `mailbox` per indicare la sua disponibilà a ricevere. Il processo `mailbox` riceverà tale segnale dalla sua porta `pronto_cons`.

Con tali indicazioni, il codice del generico consumatore può essere quindi il seguente:

```
process Cj{  // per 0≤j≤4
  port T dati;
  T messaggio;
  process p;
  signal s;
      ……
  send(s)to mailbox.pronto_cons; /*richesta di un messaggio*/
  p=receive(messaggio)from dati; /*ricezione del un messaggio*/
  <consuma il messaggio>;
      ……
}
```

Analogamente, un processo produttore invia un segnale al processo `mailbox` per indicare che è disponibile a inviare un messaggio. Il processo `mailbox` riceverà tale segnale dalla sua porta `pronto_prod`. Una volta inviato il segnale, il produttore attende un segnale di acknowledge dal processo `mailbox` attraverso la sua porta `ok_to_send`. La ricezione di tale segnale garantisce al produttore che il messaggio può essere inviato in quanto la `mailbox` non è piena. Il messaggio viene inviato attraverso la porta `dati` di tipo `T` del processo `mailbox`.

Il codice di un generico processo produttore è quindi il seguente:

```
process produttorei{
  port signal ok_to_send;
  T messaggio;
  process p;
  signal s;
      ……
  <produci il messaggio>;
  send(s)to mailbox.pronto_prod; /*richesta di invio un messaggio*/
  p=receive(s)from ok_to_send; /*attesa del permesso di invio*/
  send (messaggio)to mailbox.dati; /* invio del messaggio*/
      ……
}
```

Soluzione

Utilizzando le primitive asincrone la porta `dati` del processo `mailbox` contiene già un buffer. È quindi inutile riservare esplicitamente un buffer come parte della struttura dati della `mailbox`. In base alle precedenti indicazioni, il codice del processo `mailbox` è il seguente:

Copyright © 2007 – The McGraw-Hill Companies srl

```
process mailbox{
  port T dati;
  port signal pronto_prod, pronto_cons;

  T messaggio;
  process proc, prod;
  signal s;
  int contatore=0; /*conta il numero dei messaggi presenti nella mailbox*/
  int sospesi=0; /*conta il numero di consumatori sospesi*/
  boolean bloccato[5]; /*bloccato[i]==true indica che Ci è sospeso*/

  process cons[5]; /*cons[i] contiene l'identificatore di Ci */

{ for (int i=0; i<5; i++)bloccato[i]=false;
   cons[0]= C0; cons[1]= C1; cons[2]= C2; cons[3]= C3;
   cons[4]= C4;
  } /*inizializzazione*/

  do
   [] (contatore<10); proc=receive(s)from pronto_prod; ->
          /* c'è spazio nella mailbox e un produttore chiede di inviare un messaggio per cui
             può essere autorizzato*/
          send(s)to prod.ok_to_send;
          if(sospesi==0) contatore ++; /* se non ci sono consumatori sospesi
                    viene incrementato il numero dei messaggi presenti nella mailbox*/
          else /* altrimenti viene ricevuto il messaggio che viene poi immediatamente
                  inviato al consumatore sospeso di indice più basso*/
               { proc=receive(messaggio)from dati;
                 int i=0; while(!bloccato[i])i++;
                 bloccato[i]=false;
                 sospesi--;
                 send(messaggio)to cons[i].dati;
               }
   [] proc:=receive(s)from pronto_cons; ->
          if(contatore>0) /* se un consumatore vuole un messaggio e nella mailbox
               sono presenti messaggi, ne viene ricevuto uno che che viene poi
               immediatamente inviato al richiedente*/
              { contatore--;
                prod=receive(messaggio)from dati;
                send(messaggio)to proc.dati;
              }
          else /* altrimenti il richiedente si sospende*/
              { sospesi++;
                  if (proc==C0) bloccato[0]=true;
                  else if (proc==C1) bloccato[1]=true;
                   else if (proc==C2) bloccato[2]=true;
                    else if (proc==C3) bloccato[3]=true;
                     else bloccato[4]=true;
               }
  od;
}
```

Mc Graw-Hill
Tutti i diritti riservati