

ESERCITAZIONE 6

12 dicembre 2013

Concorrenza in ADA

Politiche basate su priorità: necessità di selezionare la richieste da servire

- Vettore delle operazioni di servizio: **Family of entries**
- Es. 2 livelli di priorità P1 e P2
- Il gestore di una risorsa deve servire prima le richieste di priorità P1, poi P2, e per ultima, P3.

```
type prio is(P1, P2);
```

```
task type server is  
    entry richiesta(prio) (<parametri formali>);  
    ..  
end server;
```

Struttura server

- Primo metodo: usare il connettore **else** all'interno di select (possibilità di innestare select):

```
select
    accept richiesta(P1)...
else
    select
        accept richiesta(P2)...
    else
        select
            accept richiesta(P3) ...
        or
            delay 1.0;
        end select;
    end select;
end select;
```

Struttura server

Secondo metodo (consigliato): usare l'attributo '**COUNT**' (applicabile alle entries). Ad esempio:

Richiesta(P1) 'COUNT

Restituisce il numero di richieste pendenti relative alla entry Richiesta(P1).

- **SCHEMA da seguire:**

```
select  
    accept richiesta(P1) do.. end;  
or  
    when richiesta(P1) 'COUNT=0 =>  
        accept richiesta(P2) do.. end;  
or  
    when richiesta(P1) 'COUNT=0 and  
        richiesta(P2) 'COUNT=0 =>  
        accept richiesta(P3) do.. end;  
end select;
```

Esercizio

Si consideri il laboratorio di un'azienda artigianale che produce dolci. L'azienda è specializzata nella produzione di torte; in particolare, i tipi di torte prodotti sono 2:

- **Torta al cioccolato,**
- **Crostata alla marmellata.**

Le torte vengono vendute in scatole pre-confezionate. L'azienda commercializza 3 tipi di confezioni:

- confezione semplice "**Cioccolato**", contenente 1 torta al cioccolato;
- confezione semplice "**Marmellata**", contenente 1 crostata.
- Confezione multipla "**Famiglia**", contenente 1 torta al cioccolato e 1 crostata.

Nel laboratorio dell'azienda vi è un **tavolo** per il deposito delle torte in attesa di essere confezionate al quale accedono:

- gli operai **dedicati alla produzione (OP)**, che accedono ciclicamente al tavolo per depositarvi ogni torta appena sfornata; ogni OP deposita sul tavolo 1 torta alla volta.
- gli operai **dedicati alle confezioni (OC)**, ognuno dedicato alla confezione di scatole di un tipo predefinito dato (Cioccolato, Marmellata o Famiglia); essi accedono ciclicamente al tavolo per **prelevare** la/le torte necessaria/e a realizzare la confezione del tipo assegnato.

Il tavolo ha una capacità massima pari a **MaxC**, costante che esprime il massimo numero di torte che possono stare contemporaneamente su di esso.

Si sviluppi un'applicazione distribuita **ADA**, che rappresenti operai (*clienti*) e gestore del tavolo(*server*) con task concorrenti. L'applicazione deve realizzare una politica di gestione del tavolo che soddisfi i vincoli dati e che, inoltre, soddisfi i seguenti **vincoli di priorità**:

- **tra gli operai OP**: i portatori di crostate siano favoriti rispetto ai portatori di torte al cioccolato;
- **tra gli operai OC**: gli operai dedicati alla confezione di scatole Famiglia siano favoriti rispetto a quelli dedicati alle scatole semplici (Cioccolato, Marmellata); inoltre, tra gli OC dedicati alle confezioni semplici, venga data priorità alle confezioni "Marmellata".

Impostazione server: famiglie di entries

```
type torta is (cioccolata, marmellata);
```

```
type confezione is (cioc, marm, family);
```

```
task type server is
```

```
    entry deposito(torta) (<par. formali>);
```

```
    entry prelievo(confezione) (<par. formali>);
```

```
end server;
```

```
S: server;
```


Impostazione clienti OP/OC

```
type cliente_ID is range 1..10;
```

```
task type clienteOP (ID: cliente_ID; T:torta);
```

```
task body cliente is
```

```
begin
```

```
    S. deposito(T)(ID);
```

```
end;
```

```
task type clienteOC(ID: cliente_ID; C:confezione);
```

```
task body cliente is
```

```
begin
```

```
    S. prelievo(C)(ID);
```

```
end;
```

Politica

- E' realizzata all'interno del server:

```
task body server is
    <variabili locali per rappr. Stato risorsa>
begin
    <INIZIALIZZAZIONI>
loop
    select
        <accettazione/definizione entries>
    end select;
end loop;
end server;
```