

## Il barbiere che dorme

Terzo esercizio sulla  
sincronizzazione dei processi

Sincronizzazione tra processi    Sistemi Operativi - Tullio Vardanega    14/21

### Specifica del problema

- ◆ Un negozio di barbiere con
  - 1 barbiere
  - 1 poltrona per il cliente servito
  - N sedie per clienti in attesa
- ◆ Protocollo di servizio
  - a. Se non vi sono clienti nel negozio il barbiere dorme sulla poltrona
  - b. Il primo cliente che entra nel negozio vuoto sveglia il barbiere
  - c. I clienti che entrano trovando la poltrona occupata si mettono in attesa su una sedia
  - d. Il cliente che non trova una sedia libera va a cercare un altro barbiere senza attendere

Sincronizzazione tra processi    Sistemi Operativi - Tullio Vardanega    15/21

### Illustrazione del problema

Sincronizzazione tra processi    Sistemi Operativi - Tullio Vardanega    16/21

### Uno schema di soluzione – 1

- ◆ Processo "barbiere"
  - Ripeti
    - { Attendi cliente
    - Servi cliente }
- ◆ Processo "cliente"
  - Entra nel negozio
    - Richiedi servizio
      - Se possibile, attendi
        - Ottieni servizio
      - Altrimenti cerca un altro barbiere

Sincronizzazione tra processi    Sistemi Operativi - Tullio Vardanega    17/21

### Uno schema di soluzione – 2

```

protected body Semaforo is
entry P when CiSonoClienti is
begin
  InCoda := InCoda - 1;
  CiSonoClienti := (InCoda > 0);
end P;
procedure V (Esito: out Boolean) is
begin
  InCoda := InCoda + 1;
  CiSonoClienti := (InCoda > 0);
  Esito := True;
exception
  -- quando ci sono più clienti di sedie la coda del negozio "tracima"
  --+ causando un overflow nel valore InCoda
  --+ il cui trattamento a programma notifica al cliente esito negativo
  when Constraint_Error =>
    Esito := False;
end V;
end Semaforo;
    
```

Sincronizzazione tra processi    Sistemi Operativi - Tullio Vardanega    18/21

### Uno schema di soluzione – 3

Sincronizzazione tra processi    Sistemi Operativi - Tullio Vardanega    19/21

## Uno schema di soluzione – 4

### Processo Barbiere

```
loop  
→ Semaforo.P; // dorme in attesa di clienti  
... // serve il primo cliente in attesa  
→ Servito.V; // rilascia il cliente servito (tramite un mutex)  
end loop;
```

### Processo Cliente

```
loop  
→ Semaforo.V (Esito); // verifica disponibilità di sedie in negozio  
if Esito then  
→ Servito.P; // attende il servizio  
... // se ne va servito  
else  
... // va a cercare un altro barbiere  
end if;  
end loop;
```

Sincronizzazione tra processi Sistemi Operativi - Tullio Vardanega 20/21

## Due eseguibili di prova

- ◆ Un cliente in arrivo ogni 5 secondi
- ◆ Tempo di servizio variabile tra 1 e 16 secondi
- ◆ **Versione 1**
  - Per MS Windows (★)
- ◆ **Versione 2**
  - Per ambiente Java

Sincronizzazione tra processi Sistemi Operativi - Tullio Vardanega 21/21