

## Prova Scritta di Calcolatori Elettronici II 31 Maggio 1997

### ESERCIZIO 1

Si consideri un castello di interesse storico.

L'accesso al castello è consentito a **due tipi** di visitatori: **adulti** o **bambini**.

La visita al castello non può avvenire in modo libero, ma deve essere sempre **guidata** dal proprietario del castello. (Essendo il proprietario **unico**, è implicito che in ogni istante ci può essere, al più, una ed una sola visita in atto.)

La visita assume caratteristiche diverse a seconda che l'insieme di visitatori da accompagnare sia costituito da adulti oppure da bambini. Per questo motivo, il gruppo di persone che partecipa ad ogni visita è quindi un insieme **omogeneo** di visitatori (cioè: o tutti adulti, oppure tutti bambini).

Per ottimizzare l'utilizzo del castello ogni visita può iniziare soltanto quando il numero  $P$  dei partecipanti ha raggiunto un valore prestabilito **P<sub>MAX</sub>**.

Quando il gruppo dei partecipanti è completo, la visita può avere inizio attraverso l'attivazione di un opportuno **processo "proprietario"** che rappresenta la guida di ogni visita.

Al termine della visita, il processo proprietario provvede a far uscire i partecipanti, e successivamente si pone in attesa di un nuovo gruppo di visitatori da guidare nella visita successiva.

Alla fine di ogni visita, il tipo dei partecipanti della visita successiva viene stabilito in base al numero e al tipo di visitatori in attesa; in particolare, detti **AS** ed **BS** rispettivamente il numero di adulti in attesa ed il numero di bambini in attesa:

- se **AS > BS** allora la prossima visita sarà per **adulti**;
- se **BS > AS** allora la prossima visita sarà per **bambini**;
- se **BS = AS** venga data la precedenza ai bambini: la prossima visita sarà per bambini.

Descrivere la politica di gestione delle visite al castello e la si implementi, usando il costrutto **monitor**, **avendo cura di esplicitare** la struttura dei vari tipi di processo.

La politica di sincronizzazione e la conseguente soluzione **deve** essere adeguatamente **commentata**.

## **Soluzione:**

program **fortezza**

const PMAX=...;

type visitatore=(adulto, bambino);

type **castello**=monitor

var attesa: array[visitatore] of condition;

    coda\_guida: condition;

    P: 0..PMAX;

    tipo:visitatore;

    SOSP:array[visitatore] of integer;

procedure entry **Entra**(t:visitatore)

begin

    while (t<>tipo and P>0) or

        (P=PMAX) do

        begin

            SOSP[t]:=SOSP[t]+1;

            attesa[t].wait;

            SOSP[t]:=SOSP[t]-1;

        end;

    tipo:=t;

    P:=P+1;

    if (P=PMAX)

        then coda\_guida.signal;

        else attesa[t].signal;

end;

procedure entry **Inizio\_Visita**

begin

    if (P<PMAX)

        then coda\_guida.wait;

end;

procedure entry **Fine\_Visita**

begin

    P:=0;

    if SOSP[adulti]>SOSP[bambini]

        then attesa[adulto].signal;

        else attesa[bambino].signal;

end;

```
begin
  P:=0;
  tipo:=bambino;
  SOSP[bambino]:=0;
  SOSP[adulto]:=0;
end; {fine monitor}
```

```
type proprietario= process
begin
  while true
    do begin
      C.Inizio_Visita;
      <visita>
      C.Fine_Visita;
    end;
```

```
type turista=process(t:visitatore)
begin
  C.Entra(t);
end;
```

```
var  B1(bambino), B2(bambino),.....:turista;
     A1(adulto), A2(adulto), .....: turista;
     C: castello;
```

```
begin {applicazione concorrente}
end.
```