

Prova Scritta di Calcolatori Elettronici II
24 Aprile 1998

Monitor

Una stretta strada di montagna non consente il passaggio contemporaneo di autoveicoli in entrambe le carreggiate, è quindi percorribile solo a senso unico alternato.

La strada è normalmente percorsa da **automobili** e **camion**.

Ogni tanto, inoltre, la strada è percorsa da un veicolo **spazzaneve**: in questo caso, l'accesso alla strada deve essere impedito a ogni altro veicolo, in qualunque verso di percorrenza, finché lo spazzaneve non ha terminato la pulizia e lasciato la strada. Lo spazzaneve può occupare la strada solo se questa è libera.

Utilizzando il costrutto monitor, si realizzi una politica di gestione della strada che tenga conto dei seguenti vincoli:

- lo spazzaneve ha la precedenza sugli autoveicoli (sia automobili che camion);
- le automobili hanno **sempre** la precedenza sui camion.

```
program StradaSensoUnicoAlternatoConSpazzaneve
```

```
const camion=1;
```

```
auto=0;
```

```
type veicolo=(auto, camion);
```

```
type dir=(nord, sud);
```

```
type strada=monitor
```

```
var attesa: array[dir] of condition;
```

```
spazzaneve: condition;
```

```
n:integer;
```

```
autosospese: array[dir] of integer;
```

```
pulizia: boolean;
```

```
dircorr:dir;
```

```
procedure entry Entra(v:veicolo; d:dir)
```

```
begin
```

```
while (d<>dircorr and n>0) or
```

```
(pulizia) or
```

```
(spazzaneve.queue) or
```

```
(v=camion and autosospese[other(d)]>0) do
```

```
begin
```

```
if v=auto then autosospese[d]:= autosospese[d]+1;
```

```
attesa[d].wait(v);
```

```
if v=auto then autosospese[d]:= autosospese[d]-1;
```

```
end;
```

```
dircorr:=d;
```

```
n:=n+1;
```

```
attesa[d].signal;
```

```
end;
```

procedure entry **Esci**

```
begin
  n:=n-1;
  if (n=0)
    then if (spazzaneve.queue)
      then spazzaneve.signal;
      else attesa[other(dircorr)].signal;
  end;
end;
```

procedure entry **InizioSpazzaneve**

```
begin
  if (n>0) then spazzaneve.wait;
  pulizia:=true;
end;
```

procedure entry **FineSpazzaneve**

```
begin
  pulizia:=false;
  if (autosospese[dircorr]>0) then attesa[dircorr].signal;
  else if attesa[other(dircorr)].queue;
    then attesa[other(dircorr)].signal;
    else attesa[dircorr].signal;
  end;
end;
```

function **other**(d:dir):dir

```
begin
  if (d=nord) then other := sud;
  else other:=nord;
end;
```

begin

```
  n:=0;
  dircorr:=nord;
  autosospese[nord]:=0;
  autosospese[sud]:=0;
  pulizia:=false;
end; {fine monitor}
```

type **spazzaneve**= process

```
begin
  ...
  M.InizioSpazzaneve;
  <pulizia strada>
  M.FineSpazzaneve;
  ...
end;
```

type **veicolo**=process(v:veicolo; d:dir)

```
begin
  ...
  M.Entra(v,d);
  <uso della strada>
  M.Esci;
  ...
end;
```

```
var A11(auto, sud), ..., A1n(auto, sud),
    A21(auto, nord), ..., A2n(auto, nord),
    A31(camion, sud), ..., A3n(camion, sud),
    A41(camion, nord), ..., A4n(camion, nord):veicolo;
    puliscistrada:spazzaneve;
    M:strada;
```

begin {applicazione concorrente} end.