Esempio di uso del costrutto monitor

L'algoritmo dell'ascensore (scan)

Algorimo *scan*

- In un edificio a N piani, l'ascensore deve servire le richieste di utenti che competono per usare l'ascensore, con l'obiettivo di spostarsi a un piano diverso.
- L'ascensore puo` servire una richiesta alla volta.
- **Movimento**: l'ascensore si puo` spostare tra gli N piani [0,..N-1] nelle due direzioni:
  - dal basso verso l'alto (SU);
  - nella direzione opposta (GIU);
- L'algoritmo SCAN e` una possibile politica di scheduling delle richieste di uso dell'ascensore, che ha come obiettivo la minimizzazione del numero dei cambiamenti di direzione.

SCAN

Politica:

- in ogni istante, all'ascensore e` associata una **direzione corrente** (SU, GIU) e una **posizione corrente** (piano associato alla prossima richiesta da servire).
- ogni richiesta e` caratterizzata da una destinazione T (che individua il piano di destinazione richiesto dall'utente in attesa):
  - se l'ascensore sta salendo (direzione SU), verranno servite tutte le richieste con T raggiungibile in quella direzione (T> posizione corrente)
  - se l'ascensore sta scendendo (direzione GIU), verranno servite tutte le richieste con T raggiungibile in quella direzione (T< posizione corrente).
- le richieste sono servite secondo l'ordine di vicinanza alla richiesta corrente.
- Quando nella direzione scelta non ci sono più richieste da servire, la direzione dell'ascensore viene invertita ed il procedimento è ripetuto.

N.B. Viene servita una richiesta alla volta.

Soluzione: monitor

- Modelliamo ogni utente dell'ascensore con un processo (thread) concorrente.
- Definiamo un monitor, il cui compito e` realizzare la politica di servizio mediante le due procedure entry:
  - **Richiesta(T)**: viene invocata dai processi per ottenere la fermata dell'ascensore al piano T:
    - se l'ascensore è ***occupato***, la richiesta del processo viene accodata in funzione dello stato corrente del'ascensore (direzione e richiesta corrente) in una coda associata ad una direzione in modo da rispettare la politica.
  - **Rilascio**: viene invocata da ogni processo arrivato a destinazione per rilasciare l'ascensore:
    - se vi sono richieste in attesa sulla stessa direzione, verra` risvegliata la prima (la piu` vicina alla posizione corrente dell'ascensore); altrimenti, la direzione verra` invertita e verra` risvegliato il primo processo in attesa nella nuova direzione.

N.B. Viene servita una richiesta alla volta.

Monitor

- Il monitor e` caratterizzato dalle seguenti variabili:
  - **occupato**: indica se l'ascensore sta servendo una richiesta;
  - **direzione**: indica la direzione corrente dell'ascensore (SU, GIU);
  - **posizione**: indica la destinazione corrente;
  - **dir_SU**, **dir_GIU**: condition associate alle due direzioni, sulle quali si sospenderanno i processi in attesa di servizio.

HP: è disponibile la *wait* con priorità

```
typedef int piano;
typedef enum{SU,GIU}dir;

monitor movimento_ascensore
{  piano posizione=0; /*inizialmente al piano terra..*/
   boolean occupato=false;    /* inizialmente libero..*/
   dir direzione=SU;
   condition dir_SU,dir_GIU;

   public void Richiesta(piano dest)
   { if (occupato==true)
        if ((direzione==SU)&& (dest<=posizione))
               dir_GIU.wait(dest);
        else dir_SU.wait(N-dest);
        occupato=true;
        posizione=dest;
   }
```

```
public void Rilascio
{ occupato=false;
   if (direzione==SU)
        if (dir_SU.queue)
                dir_SU.signal;
        else{   direzione=GIU;
                dir_GIU.signal; }

   else  if (dir_GIU.queue)
                dir_GIU.signal;
        else{   direzione=SU;
                dir_SU.signal; }
}
} /* fine monitor*/
```

```
movimento_ascensore A; /* istanza del monitor*/

void Utente(piano PART, piano ARR) /*codice di un
   generico utente */
{      /* chiamata ascensore dal piano PART: */
       A.Richiesta(PART);
       <uso dell'ascensore>;
       A.Rilascio();
       /*entrata nella cabina e
       richiesta del piano di destinazione ARR:*/
       A.Richiesta(ARR);
       <uso dell'ascensore>;
       A.Rilascio();

}
```