Sistemi Operativi M
Prof. Anna Ciampolini
Simulazione Prova di Laboratorio 18 dicembre 2009

Tema A

Si consideri la pista di pattinaggio sul ghiaccio di una località turistica montana.
La pista di pattinaggio è aperta a tutti.
In particolare i clienti dell'impianto si suddividono in due categorie: **principianti** e **esperti**. Ogni gruppo di principianti, durante la permanenza all'interno della pista, viene accompagnato da un istruttore messo a disposizione dalla società che gestisce l'impianto; a questo proposito, si supponga che il numero totale di istruttori sia **NI**.
I pattinatori entrano ed escono dalla pista a **gruppi omogenei** (ogni gruppo è formato solo da principanti o solo da esperti) e **monolitici** (ogni pattinatore fa parte dello stesso gruppo, sia in ingresso che in uscita), ognuno caratterizzato da una **consistenza numerica** data.
La capacità della pista è limitata dal valore **MAX**, che esprime il numero massimo di pattinatori che possono essere in pista contemporaneamente (gli istruttori non vengono conteggiati).
Inoltre, il regolamento dell'impianto prevede che debba essere sempre rispettata la relazione:

$$P \geq E$$

dove: P è il numero dei principianti in pista, e E è il numero degli esperti in pista.
Si sviluppi un'applicazione concorrente in **C/pthread** che rappresenti i gruppi di pattinatori come thread concorrenti.
In particolare, la soluzione deve implementare una politica di sincronizzazione dei thread che rispetti le specifiche date, ed inoltre i vincoli seguenti:

- **nell'accesso alla pista**: i **principianti abbiano la precedenza sugli esperti**; a parità di categoria, si privilegino i gruppi meno numerosi.
- **nell'uscita dalla pista**: gli **esperti abbiano la precedenza sui principianti**; a parità di categoria, si privilegino i gruppi più numerosi.

Sistemi Operativi M
Prof. Anna Ciampolini
Simulazione Prova di Laboratorio 18 dicembre 2009

Tema B

Si consideri un importante museo.
Il museo può essere visitato da gruppi di persone che possono essere *eventualmente* accompagnati nella visita da guide messe a disposizione dal museo. A questo proposito, si supponga che il numero totale delle guide disponibili nel museo sia pari a **NG**.
Pertanto, i gruppi possono essere di due tipi: ***guidati*** e ***liberi***; ogni gruppo è caratterizzato da una consistenza numerica costante.
La capacità del museo è limitata dal valore **MAX**, che esprime il numero massimo di visitatori che possono essere all'interno del museo contemporaneamente (le guide non vengono conteggiate).
Inoltre, per limitare il rischio di confusione, la società che gestisce il museo impone che debba essere sempre rispettata la seguente relazione:

$$G \geq L$$

dove: G è il numero dei visitatori appartenenti a gruppi guidati all'interno del museo, e L è il numero dei visitatori appartenenti a gruppi liberi all'interno museo.

Si sviluppi un'applicazione concorrente in **Java** che rappresenti i gruppi di visitatori come thread concorrenti.
La soluzione dovrà implementare una politica di sincronizzazione dei thread che rispetti le specifiche date ed inoltre rispetti in vincoli seguenti:

- **nell'accesso al museo**: i **gruppi guidati abbiano la precedenza su quelli liberi**; a parità di categoria, si privilegino i gruppi meno numerosi.
- **nell'uscita dal museo**: i **gruppi liberi abbiano la precedenza su quelli guidati**; a parità di categoria, si privilegino i gruppi più numerosi.

Sistemi Operativi M
Prof. Anna Ciampolini
Simulazione Prova di Laboratorio 18 dicembre 2009

Tema C

Si consideri la pista di pattinaggio sul ghiaccio di una località turistica montana.
La pista di pattinaggio è aperta a tutti.
In particolare i clienti dell'impianto si suddividono in due categorie: **principianti** e **esperti**. Ogni gruppo di principianti, durante la permanenza all'interno della pista, viene accompagnato da un istruttore messo a disposizione dalla società che gestisce l'impianto; a questo proposito, si supponga che il numero totale di istruttori sia **NI**.
I pattinatori entrano ed escono dalla pista a **gruppi omogenei** (ogni gruppo è formato solo da principanti o solo da esperti) e **monolitici** (ogni pattinatore fa parte dello stesso gruppo, sia in ingresso che in uscita), ognuno caratterizzato da una **consistenza numerica** data.
La capacità della pista è limitata dal valore **MAX**, che esprime il numero massimo di pattinatori che possono essere in pista contemporaneamente (gli istruttori non vengono conteggiati).
Inoltre, il regolamento dell'impianto prevede che debba essere sempre rispettata la relazione:

$$P \geq E$$

dove: P è il numero dei principianti in pista, e E è il numero degli esperti in pista.
Si sviluppi un'applicazione concorrente in **Java** che rappresenti i gruppi di pattinatori come thread concorrenti.
In particolare, la soluzione deve implementare una politica di sincronizzazione dei thread che rispetti le specifiche date, ed inoltre i vincoli seguenti:

- **nell'accesso alla pista**: i **principianti abbiano la precedenza sugli esperti**; a parità di categoria, si privilegino i gruppi meno numerosi.
- **nell'uscita dalla pista**: gli **esperti abbiano la precedenza sui principianti**; a parità di categoria, si privilegino i gruppi più numerosi.

Sistemi Operativi M
Prof. Anna Ciampolini
Simulazione Prova di Laboratorio 18 dicembre 2009

Tema D

Si consideri un importante museo.
Il museo può essere visitato da gruppi di persone che possono essere *eventualmente* accompagnati nella visita da guide messe a disposizione dal museo. A questo proposito, si supponga che il numero totale delle guide disponibili nel museo sia pari a **NG**.
Pertanto, i gruppi possono essere di due tipi: ***guidati*** e ***liberi***; ogni gruppo è caratterizzato da una consistenza numerica costante.
La capacità del museo è limitata dal valore **MAX**, che esprime il numero massimo di visitatori che possono essere all'interno del museo contemporaneamente (le guide non vengono conteggiate).
Inoltre, per limitare il rischio di confusione, la società che gestisce il museo impone che debba essere sempre rispettata la seguente relazione:

$$G \geq L$$

dove: G è il numero dei visitatori appartenenti a gruppi guidati all'interno del museo, e L è il numero dei visitatori appartenenti a gruppi liberi all'interno museo.

Si sviluppi un'applicazione concorrente in **c/pthread** che rappresenti i gruppi di visitatori come thread concorrenti.
La soluzione dovrà implementare una politica di sincronizzazione dei thread che rispetti le specifiche date ed inoltre rispetti in vincoli seguenti:

- **nell'accesso al museo**: i **gruppi guidati abbiano la precedenza su quelli liberi**; a parità di categoria, si privilegino i gruppi meno numerosi.
- **nell'uscita dal museo**: i **gruppi liberi abbiano la precedenza su quelli guidati**; a parità di categoria, si privilegino i gruppi più numerosi.