

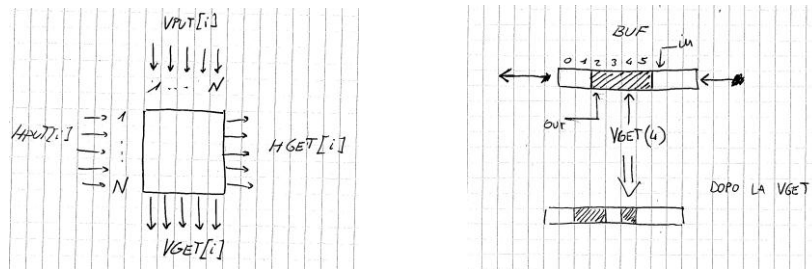
## Corso di Sistemi Operativi – Appello straordinario del 22 Marzo 2012

### Istruzioni

Se state leggendo, vuol dire che avete trovato sul vostro Desktop una cartella "CognomeNomeMatricola" che contiene la traccia dell'elaborato ed eventuali altri file utili per lo svolgimento della prova. Ai fini del superamento della prova è indispensabile rinominare tale cartella sostituendo "Cognome" "Nome" e "Matricola" con i vostri dati personali, e includere al SUO INTERNO tutti i file sorgente prodotti durante lo svolgimento della prova. Ad esempio, uno studente che si chiama Alex Britti ed ha matricola 66052 dovrà rinominare la cartella "CognomeNomeMatricola" in "BrittiAlex66052". Se si è sviluppato un progetto con Eclipse, i sorgenti prodotti dovrebbero essere rintracciabili nella cartella /home/\$NOMEUTENTE/workspace (dove \$NOMEUTENTE è il nome utente corrente, ad es. "esame").

Non saranno presi in considerazione file non chiaramente riconducibili al proprio autore.

**Si consiglia di salvare SPESSE il proprio lavoro.**



Si deve progettare una classe *BufferQuadrato*. Un *BufferQuadrato* è costituito da un array BUF di N buffer, ciascuno dei quali può contenere N elementi di tipo T. Ogni buffer BUF[i] si comporta singolarmente come un buffer circolare.

I metodi che devono essere implementati sono i seguenti:

- HPUT(int i, T E) : inserisce l'elemento E (di tipo T) nel buffer i-esimo. Il metodo pone il thread chiamante in stato di attesa se tale buffer dovesse essere pieno.
- T HGET(i) : estrae un elemento dal buffer i-esimo e lo restituisce. Il metodo pone il thread chiamante in stato di attesa se tale buffer dovesse essere pieno.
- VPUT(int j, T E) : introduce l'elemento E nel primo, tra i buffer BUF[0]..BUF[N] che si ritrovi ad avere la posizione j-esima libera. Il metodo pone il thread chiamante in stato di attesa se nessuno dei buffer dovesse avere tale posizione libera.
- T VGET(j) : estrae e restituisce l'elemento j-esimo dal primo buffer BUF che risulti avere tale posizione occupata. Il metodo pone il thread chiamante in stato di attesa se nessuno dei buffer dovesse avere tale posizione libera.

Le funzioni VPUT e VGET possono eventualmente creare delle zone vuote (o piene) non contigue nei singoli buffer. Tale situazione deve essere gestita opportunamente continuando a garantire la politica FIFO per inserimento e estrazione.

La libreria deve essere implementata garantendo **la mutua esclusione** nell'accesso ai dati condivisi **ove necessario**, al fine di garantire il massimo grado di parallelismo; devono essere inoltre evitate potenziali situazioni di deadlock e, possibilmente, di starvation.

*E' parte integrante della prova di esame completare le specifiche date nei punti non esplicitamente definiti, introducendo tutte le strutture dati che si ritengano necessarie, e risolvendo eventuali ambiguità. Non è consentito modificare il prototipo dei metodi se questo è stato fornito.*

**La prova può essere svolta** in C++ con libreria JTC + socket, oppure in Java. Nel caso dovessero servire al candidato, si può assumere di avere a disposizione i costrutti di sincronizzazione lock, read/write lock, barriera e buffer sincronizzato. Il candidato può fare uso di una qualsiasi delle funzioni di libreria Java 6, nel caso opti per svolgere l'elaborato in Java.

*E' consentito scrivere un main() e dei thread di prova A PROPRIO PIACIMENTO.*