



Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

Programmazione Concorrente e Distribuita Thread: esercizi

Luigi Lavazza
Dipartimento di Scienze Teoriche e Applicate
luigi.lavazza@uninsubria.it



Esercizio 1

- Realizzare un sistema con un produttore e un consumatore
 - ▶ Usando un monitor
 - ▶ La cella è thread-safe, ma non blocca i consumatori su buffer vuoto o i produttori su buffer pieno
- Ovviamente, si osservano problemi...



Esercizio 1b

- Modificare la soluzione dell'esercizio 1 in modo che il sistema si comporti «bene», cioè
 - ▶ Il produttore aspetta quando il buffer è pieno
 - ▶ Il consumatore aspetta quando il buffer è vuoto



Esercizio 2

- Si realizzi un sistema con più produttori e consumatori



Esercizio 3

- Realizzare un sistema con un produttore e un consumatore
 - ▶ Usando i semafori, non i monitor
 - ▶ Tutta la sincronizzazione viene fatta nella coda (da realizzare)



Esercizio 4

- Realizzare un sistema con un produttore e un consumatore
 - ▶ Che fanno polling usando BlockingQueue
 - ▶ Usare
 - add (che solleva eccezione in caso di fallimento)
 - e poll (che restituisce null in caso di fallimento)



Esercizio 5

- Realizzare un sistema con un produttore e un consumatore
 - ▶ Con BlockingQueue
 - usando special value (offer e poll)



Esercizio 6

- Realizzare un sistema con un produttore e un consumatore
 - ▶ Con BlockingQueue
 - con time out



Esercizio 7

- Realizzare un sistema con due thread che condividono una risorsa
- I due thread devono accedere alternativamente alla risorsa.
 - ▶ Cioè non deve mai succedere che lo stesso thread acceda due volte consecutive alla risorsa.