

Sistemi di Calcolo 2 (A.A. 2019-2020 e successivi)

Prova congiunta Sistemi di calcolo 12 CFU - Seconda parte (A.A. 2017-2018 e precedenti)

Appello - 26 Gennaio 2022

Tempo a disposizione: 1h 45m.

Attenzione: assicurarsi di compilare il file **studente.txt** e che il codice prodotto non contenga **errori di compilazione**, pena una non correzione dell'elaborato.

Regole Esame

- Domande ammesse
Le domande possono riguardare solo la specifica dell'esame e la struttura di alto livello del codice, nessuna domanda può riguardare singole istruzioni.
- Oggetti vietati
I seguenti oggetti non devono essere presenti sulla scrivania, né tantomeno usati: smartphone, smartwatch, telefonini, tablet, portatili, dispositivi di archiviazione USB, copie cartacee della dispensa, astucci e qualsiasi forma di libri ed appunti. **Chi verrà sorpreso ad usare uno di questi oggetti verrà automaticamente espulso dall'esame.**
- Modalità di risposta
Le risposte alle domande di teoria vanno fornite nei file txt indicati. Il professore fornisce fogli per appunti e dove fornire illustrazioni o codice utile ad integrazione delle risposte di teoria. Qualsiasi altro foglio portato dallo studente non può essere usato.
- Azioni vietate
È assolutamente vietato comunicare in qualsiasi modo con gli altri studenti. **Chi verrà sorpreso a comunicare con gli altri studenti per la prima volta verrà richiamato, la seconda volta verrà invece automaticamente espulso dall'esame.**

Teoria 1 - Socket (rispondere nel file teoria1.txt)

Descrivere come si utilizza una datagram socket all'interno di un'applicazione client-server, avvalendosi di pseudo codice. Parlare inoltre del protocollo di comunicazione che ne fa uso.

Teoria 2 - Algoritmi di concorrenza (rispondere nel file teoria2.txt)

Considerate il seguente algoritmo del panettiere "modificato":

Initially

```
/* global info */
boolean choosing[N] = {false, ..., false};
integer num[N] = {0, ..., 0};
/* local info */
int i = <entity ID>; //  $i \in \{0, 1, \dots, N-1\}$ 
repeat
1   NCS
2   choosing[i] := true
3   num[i] := max {num[j] :  $0 \leq j < N$ }
4   choosing[i] := false
5   for j := 0 to N-1 do begin
6       while choosing[j];
7       while num[j] != 0 AND {num[j], j} < {num[i], i};
8   end
9   CS
10  num[i] := 0
forever
```

L'algoritmo del panettiere è stato modificato a linea 3 eliminando la somma di 1 al max, (istruzione corretta: $\text{num}[i] := 1 + \max \{ \text{num}[j] : 0 \leq j < N \}$)

- A. Descrivere cosa comporta tale modifica nell'esecuzione del programma
- B. Quale o quali sono le problematiche che si creano legate alle proprietà di Mutua Esclusione, No-Deadlock e No-Starvation? Se e quali vengono violate? Motivare la risposta.

Programmazione - Realizzazione di un sistema multi-process

All'interno di un sistema uno o più processi producer generano degli input che devono poi essere elaborati da un solo processo elaborator. Ogni producer ha un bilancio iniziale in euro e ogni volta che trasmette un input all'elaborator fornisce anche una ricompensa (reward) per la sua elaborazione. Lo scambio dei dati tra i processi avviene tramite una memoria condivisa che contiene un buffer in cui i processi producer e il processo elaborator accedono secondo la modalità produttore/consumatore; gli indici per l'accesso alla memoria, i semafori unnamed di sincronizzazione. Memoria e semafori sono istanziati dall'elaborator, che deve pertanto essere avviato prima di qualsiasi producer. Ogni volta che un producer genera un task, memorizza input e reward all'interno di una struttura dati che viene inserita all'interno del buffer e decrementa il proprio bilancio, quando il bilancio non è sufficiente a pagare il task il producer termina la propria attività. L'elaborator preleva un input del task dopo l'altro insieme al reward, che viene sommato al proprio bilancio.

Si chiede di completare i file `common.h`, `producer.c` e `elaborator.c`

Obiettivi

1. Gestione di una memoria condivisa
2. Paradigma produttore/consumatore tramite semafori unnamed inclusi nella memoria condivisa

Altro

- **i commenti nel codice contengono molte informazioni utili per lo svolgimento della prova, si consiglia quindi di tenerli in debita considerazione**
 - il file `dispensa.pdf` contiene una copia della dispensa *Primitive C per UNIX System Programming* preparata dai tutor di questo corso
 - il file `raccomandazioni.pdf` contiene una serie di considerazioni sugli errori riscontrati più di frequente
-