

Sistemi di Calcolo - Modulo 2 (A.A. 2015-2016)

Terzo appello - 16 Settembre 2016

Tempo a disposizione: 1h 30'.

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

Esercizio 1 - EchoServer multi-process con comunicazione via socket

Il processo server si avvia e predispone una socket per accettare connessioni di processi client. Ad ogni connessione di un client, il server crea un processo figlio che si occuperà dello scambio messaggi con il client. Il processo padre invece si rimette in attesa di nuove connessioni così da poter gestire più client in parallelo.

Ogni messaggio che il client invia al server è seguito da una risposta del server con lo stesso messaggio. Il comando di terminazione per un client è QUIT.

Per connettersi al server, il client deve essere avviato con i parametri `SERVER_ADDRESS` e `SERVER_PORT` specificati in `common.h`.

Obiettivi principali

1. Gestire creazione processi figlio
2. Scambio messaggi di lunghezza variabile su socket con gestione interruzioni
3. Chiusura descrittori e rilascio risorse allocate dinamicamente
4. Gestione degli errori

Esercizio 2 - Produttore/consumatore su N buffer in parallelo

Un processo inizializza `N` buffer circolari da gestire secondo il paradigma *single prod/single cons*, ed avvia in parallelo due thread: un orchestratore dei produttori e un orchestratore di consumatori. L'orchestratore dei produttori avvia `N` produttori in parallelo, dove l'`i`-esimo produttore inserisce `ELEMENT_COUNT` elementi nell'`i`-esimo buffer. L'orchestratore dei consumatori avvia `N` consumatori in parallelo, dove l'`i`-esimo consumatore legge `ELEMENT_COUNT` elementi dall'`i`-esimo buffer.

Ogni orchestratore attende il termine dei thread che ha avviato, ed il main thread attende il termine dei due orchestratori.

Obiettivi principali

1. Gestione multi-thread: creazione/attesa terminazione thread, gestione argomenti
2. Sincronizzazione: inizializzazione/rimozione semafori, paradigma prod/cons

Altro

- **i commenti nel codice contengono molte informazioni utili per lo svolgimento della prova, si consiglia quindi di tenerli in debita considerazione**
- in caso di necessità, nella cartella `backup/` è presente una copia della traccia
- 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

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.**
- 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.**