

Programowanie współbieżne

Lista zadań nr 14

Na ćwiczenia 3. lutego 2022

Zadanie 1. Do czego służy licznik współdzielony (ang. *shared counter*)? Podaj ogólny schemat licznika współdzielonego wykorzystującego sieć zliczającą (ang. *counting network*) oraz wyjaśnij, dlaczego wymagamy by ta sieć miała własność krokową (ang. *step property*). Podaj szczegółowy schemat sieci zliczającej z czterema wejściami/wyjściami złożonej z równoważników (ang. *balancers*) i opartego na niej licznika. Wyjaśnij, jak działa ta sieć.

Wskazówka: TAoMP2e, r. 12.5

Zadanie 2. Pokaż rekurencyjną konstrukcję sieci BITONIC[2^k], dla dowolnego k , i udowodnij jej poprawność.

Wskazówka: Slajdy do wykładu 13., 159 – 180

Zadanie 3. Podaj wydajną, niewstrzymywaną (ang. *lock-free*) implementację klasy **Balancer** (równoważnika).

Zadanie 4. Podaj wydajną, nieczekającą (ang. *wait-free*) implementację klasy **Balancer**.

Wskazówka: Jedyną trudność tego zadania polega na wybraniu odpowiedniej reprezentacji dla równoważnika. Innej niż obiekt dwustanowy, który zapewne występował w Twojej implementacji z zadania poprzedniego. Wybierz taką reprezentację, by móc wykorzystywać operacje atomowe na pewnym standardowym typie danych.

Zadanie 5. Opisz działanie metod **add()**, **contains()** oraz **remove()** klasy **LockFreeHashSet** implementującej opartą na rekurencyjnym porządku dzielonym. Dlaczego zależy nam, żeby operacja rozszerzania tablicy haszującej odbywała się inkrementacyjnie?

Wskazówka: TAoMP2e, r. 13.3

Zadanie 6. Do czego służy blokada odczytujący-zapisujący (ang. *readers-writers lock*). Porównaj klasę **SimpleReadWriteLock** z **FifoReadWriteLock** i wskaż, jaką pożądaną cechę ma ta druga.

Wskazówka: TAoMP2e, r. 8.3

Zadanie 7. Zadanie 8.2 z TAoMP2e.

Zadanie 8. Zadanie 8.3 z TAoMP2e.