

Documentatia

Implementare coada

Coadă permite trei operații: send, recv, close. Se folosește un semafor pentru a implementa mecanismul de așteptare la coadă.

Operația send adaugă un element în coadă și face release la semafor. Operația recv așteaptă să existe cel puțin un element în coadă (facând acquire la semafor), și apoi scoate acel element din coadă și îl returnează.

Accesul la coadă este protejat prin intermediul unui mutex.

Operația close semnalează sfârșitul adăugării elementelor în coadă. Atunci când coada se termină, se eliberează toate thread-urile care mai așteaptă la semafor. Odată ce coada este închisă, nu se mai permite adăugarea a noi elemente.

Analiza

Variantă single-threaded tinde să fie mai rapidă decât cea multi-threaded.

Variantă multi-threaded pare să meargă mai repede cu mai multe thread-uri dedicate citirii.

Cred că o explicație posibilă pentru creșterea performanței odată cu creșterea numărului de thread-uri dedicate citirii, este că aceasta scade numărul de worker thread-uri. Un număr mai mic de worker thread-uri îmbunătățește performanța, pentru că scade overhead-ul asociat cu fiecare thread (blocarea de mutex, așteptarea la semafor, etc.; toate introduc overhead).

Tabel rulări

configuration time

p=0	0.00220
p=4,p_r=1	0.00400
p=4,p_r=2	0.00375
p=6,p_r=1	0.00492
p=6,p_r=2	0.00471
p=8,p_r=1	0.00620
p=8,p_r=2	0.00609
p=16,p_r=1	0.00921
p=16,p_r=2	0.00904