Fakultet tehničkih nauka

Univerzitet u Novom Sadu

**Publisher-subscriber projekat**

Industrijski komunikacioni protokoli

**Uvod**

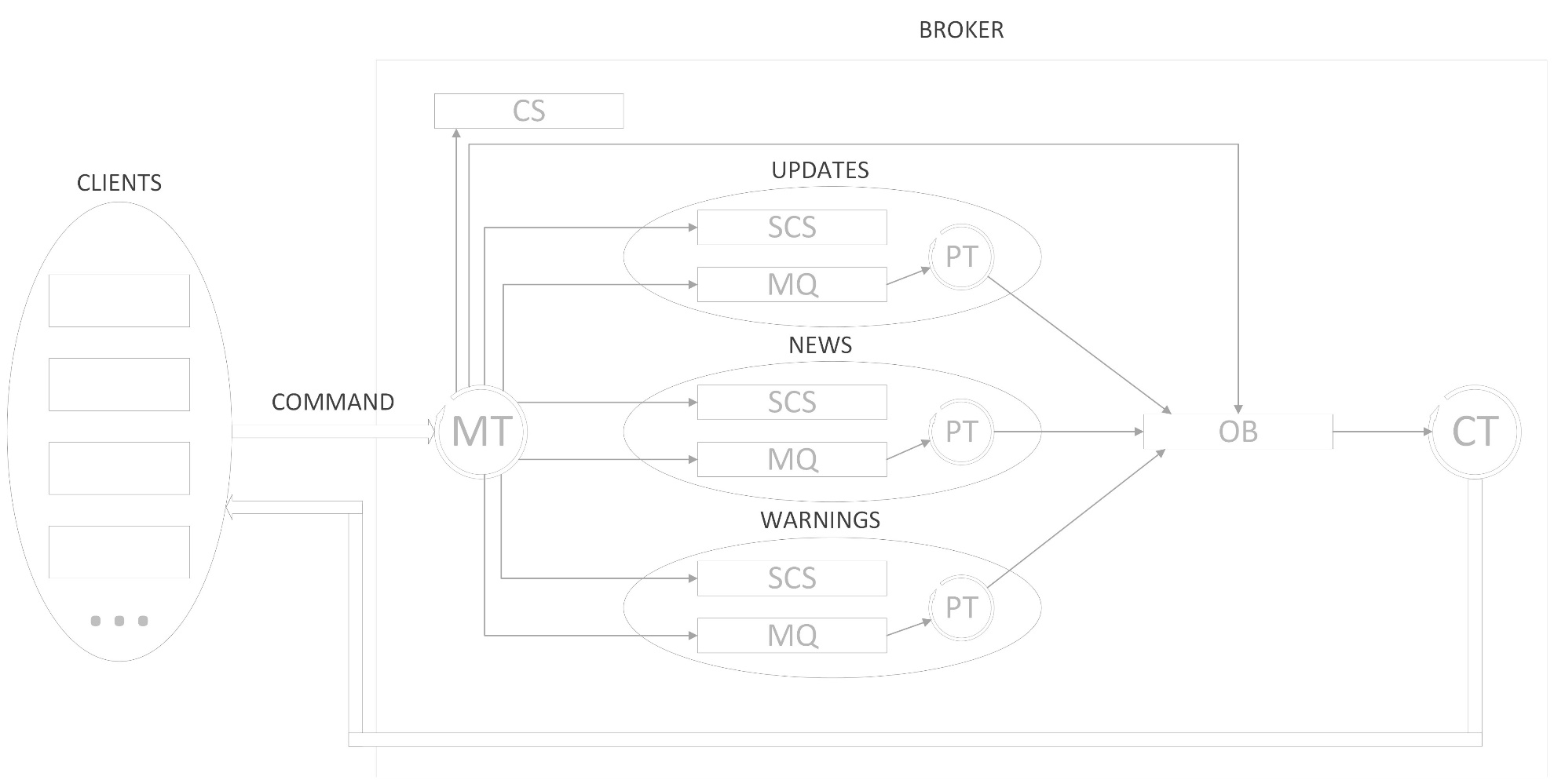
Publisher-subscriber projekat implementira publisher-subscriber pattern. Centralna komponenta je broker i on pruža klijentima sledeće usluge:

* Subscribe: Postoje tri teme (topics) i svaki klijent može da se prijavi na jednu ili vise temu. Broker prosleđuje klijentima koji su prijavljeni na određenu temu poruke koje pripadaju toj temi.
* Publish: Klijent može da posalje poruku i da izabere kojoj temi će ta poruka da pripada.
* TopicExists: Klijent može da traži od brokera da mu kaže da li tema sa određenim nazivom postoji.
* SubscriberNumber: Klijent može da zahtjeva od brokera da mu on kaže koliko klijenata je prijavljeno na određenu temu.

Svaki klijent može da ima i ulogu publisher-a i ulogu subscriber-a.

**Dizajn**

Na slici je prikazan dizajn projekta.



Skraćenice:

* MT - main thread
* PT - producer thread
* CT - consumer thread
* CS - connection sockets (list)
* SCS - subscriber connection sockets (list)
* MQ - message queue (queue)
* OB - outgoing buffer (circular buffer)

Main thread prihvata konekcije sa klijentima i prima komande klijenata.

Kada se konekcija ostvari, main thread dodaje connection socket klijenta u SCS.

Kada klijent posalje Publish komandu, main thread stavi poruku iz te komande u message queue odgovarajuće teme.

Kada klijent posalje Subscribe komandu, main thread stavi connection socket klijenta u SCS listu odgovarajuće teme.

Kada klijent posalje TopicExists ili SubscriberNumber komandu, main thread stavlja odgovor u outgoing buffer.

Producer thread-ovi uzimaju poruke sa početka message queue-ova i stavljaju ih u outgoing buffer.

Consumer thread čita poruke (i odgovore) iz outgoing buffer-a i salje ih klijentima.

**Strukture podataka**

Socket list

Socket-i se čuvaju u jednostruko spregnutoj listi. Socket-i se uvijek dodaju na početak liste jer je to najefikasnije, a redosled socket-a u listi nije bitan.

Message queue

Poruke se stavljaju na kraj queue-a, a uzimaju se sa početka queue-a. Queue se koristi zato što je bitno da poruke koje ranije stignu budu ranije i prosleđene subscriber-ima.

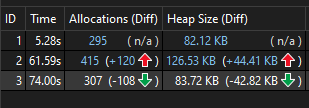
Outgoing buffer

Outgoing buffer je cirkularni buffer. Kada se on koristi na ovakav način, to predstavlja klasični producer/consumer problem za koji postoji dobro poznato i efikasno rješenje.

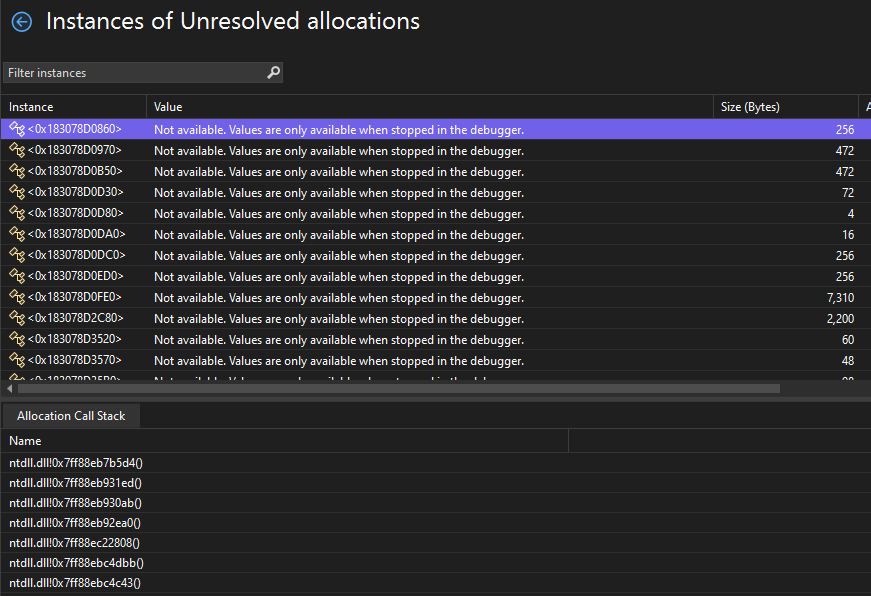
**Rezultati testiranja**

Test za provjeru curenja memorije

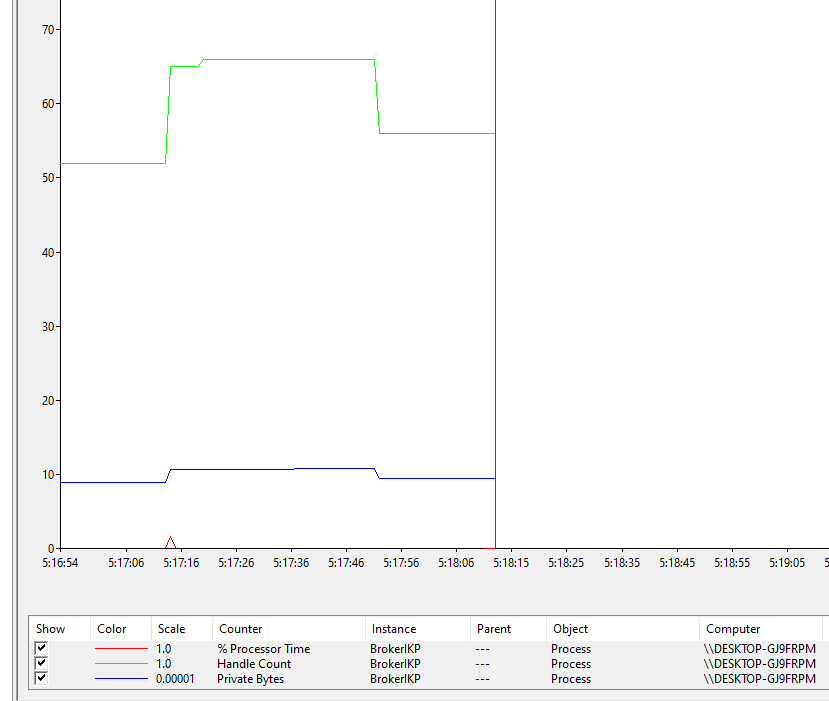
Na slici je prikazan rezultat iz Visual studio profiler-a. Prvi red predstavlja snapshot sa početka broker programa, drugi red predstavlja snapshot u toku opsluživanja klijenata, a treći red predstavlja snapshot sa kraja programa, nakon što je sva memorija na heap-u oslobođena.



U trećem snapshot-u je veličina heap-a malo veća nego u prvom, ali to nije zbog broker procesa nego zbog windows operativnog sistema koji zauzima memoriju kako bi upravljao ovim procesom. To se može vidjeti na sledećoj slici na kojoj se vidi da je tu memoriju zauzeo program iz fajla ntdll.dll koji pripada windows operativnom sistemu.



Na sledećoj slici vidi se rezultat test-a u Performance monitor alatu.



Stress test

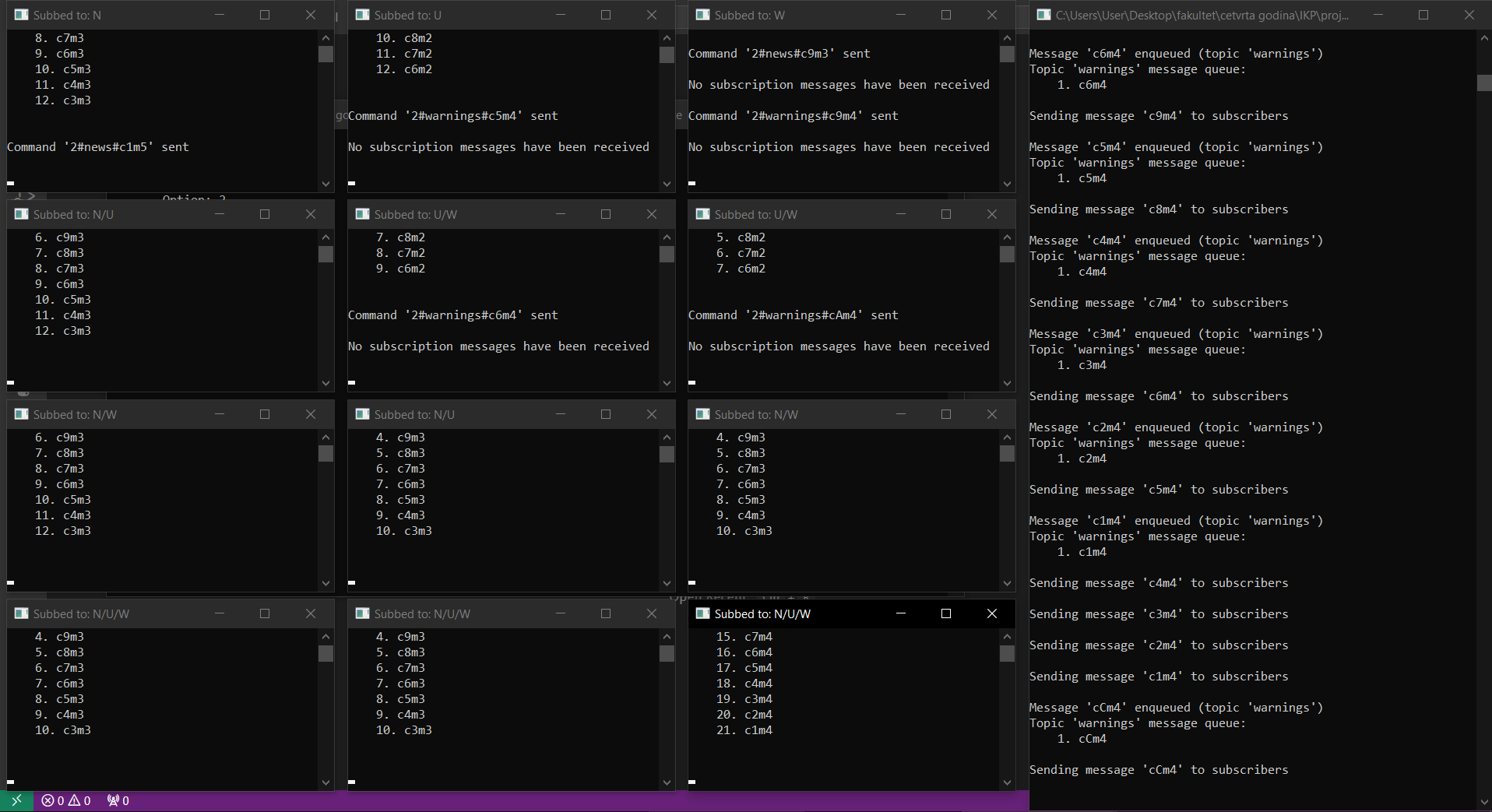
Urađen je i stress test u kome se 12 klijenata konektuje na broker, prijave se na određene teme (u prosijeku se svaki klijent prijavi na dvije od tri teme) i onda publish-uju poruke.

Prijavljivanje na teme:

A screenshot of a computer screen

Description automatically generated

Publish-ovanje je prikazano ovde. U periodu od oko 10 sekundi svaki klijent publish-uje po 9 komandi. Tj. ukupno broker primi i prosledi 108 poruka za 10 sekundi.



Ni u jednom trenutku se ne desi da u nekom message queue-u budu više od tri poruke, što pokazuje efikasnost broker-a. Svi subscriber-i prime poruku odmah nakon što je publisher publish-ovao.

