



FAKULTET TEHNIČKIH NAUKA

Univerzitet u Novom Sadu  
Industrijski komunikacioni protokoli u EES

# Heap Memory Management

Autori:

Aleksandar Antonijev PR 109/2020

Veljko Kondić PR 122/2020

## Uvod:

Ovaj dokument opisuje implementaciju rukovaoca heap-om koji se oslanja na mark-sweep algoritam za automatsko rukovanje memorijom. Cilj ovog zadatka je implementirati thread-safe rukovalac koji omogućava alokaciju, dealokaciju i sakupljanje smeća (garbage collection) u segmentno organizovanom heap-u. Takođe, potrebno je da implementira merenje propusnosti alokacije i dealokacije u različitim brojevima niti.

## Ciljevi zadatka:

1. Implementacija mark-sweep algoritma: Razviti algoritam za automatsko rukovanje memorijom koji koristi mark-sweep pristup za identifikaciju i dealokaciju objekata.
2. Thread-safety: Osigurati da rukovalac heap-om bude thread-safe kako bi se izbegli problemi višestrukih pristupa iz različitih niti.
3. Segmentna organizacija heap-a: Podeliti heap na segmente kako bi se efikasnije upravljalo memorijom i smanjila fragmentacija.
4. Sakupljanje korena: Koristiti strukturu podataka koja će služiti za čuvanje skupa korenskih referenci.
5. Merenje propusnosti: Implementirati funkcije za merenje propusnosti alokacije i dealokacije u različitim brojevima niti.

## Strukture podataka:

- Niz vektora koji služe za čuvanje korenskih referenci blokova u memoriji.
- Segmenta organizacija heap-a radi efikasnijeg upravljanja memorijom. Svaki segment sadrži više blokova memorije.
- Blokovi memorije koji su skladišteni u okviru segmenata.

## Semantika podataka:

### 1. Mark-Sweep Algoritam:

Označavanje (Mark): Faza u kojoj se svi pristupačni objekti markiraju(označavaju) kako bi se identifikovali koreni i objekti koji nisu potrebni.

Brisanje (Sweep): Označeni objekti se dealociraju kako bi se oslobodio prostor.

### 2. Thread-Safety

Korišćenje mehanizama sinhronizacije, poput međusobne isključivosti (mutexa) , kako bi se omogućilo bezbedno rukovanje sa strukturama podataka i heap-om u višenitnom okruženju.

## Dizajn:

Kratak opis: Heap je podeljen u više segmenata koji skladište više blokova memorije. Moguće je alocirati jedan po jedan blok, takođe je moguće dealocirati blokove. Za potrebe merenja propusnosti alokacije i dealokacije moguće je koristiti različit broj niti.

## Zaključak i potencijalna unapređenja:

Implementacija mark-sweep algoritma za automatsko rukovanje memorijom pokazuje zadovoljavajuće performanse u kontekstu thread-safe rukovanja i segmentne organizacije heap-a. Merenje propusnosti alokacije i dealokacije u različitim brojevima niti pruža uvid u performanse rukovaoca. Lošiji rezultati mogu proizići iz visoke konkurencije između niti za pristup heap-u, što može dovesti do čekanja na resurse i povećanja vremena izvršavanja.

U zaključku, ova implementacija pruža balans između thread-safety, efikasnog rukovanja memorijom i segmentacije heap-a, uzimajući u obzir izazove u radu u višenitnom okruženju.

Moguće je unaprediti ovaj projekat korišćenjem Serverske i Klijentske komponente kako bi se bolje pokazao sam proces rukovanja memorijom, njenim zauzećem i oslobađanjem iste.