

Specifikacija Zahteva za Projektovanje Alokatora Memorije

Uvod

Današnji softverski sistemi su veoma kompleksni i nije moguće unapred odrediti koliko memorije im je potrebno za rad niti je efikasno odvojiti memoriju za podatke koji nikada istovremeno neće biti potrebni. Zato se današnji softverski sistemi koriste konceptom dinamičke memorije. Dinamička memorija je memorija čija je upotreba određena za vreme izvršavanja programa a ne prilikom njegovog prevođenja. Neki programski jezici automatski koriste dinamičku memoriju (Python je jedan od njih), dok ostali zahtevaju od programera da eksplicitno naznači upotrebu dinamičke memorije (ključne reči *malloc* i *new* u C/C++-u). Kada programski jezik ili programer naznači upotrebu dinamičke memorije on zapravo traži da se u memoriji programa odvoji blok memorije tražene veličine (koja ne mora biti poznata pri prevođenju) i da programu na korišćenje. U zavisnosti od konkretne implementacije, od programera se može tražiti i da eksplicitno navede trenutak u kome se ta memorija više neće koristiti. Kod koji ispunjava ove zahteve programskog jezika ili programera i vodi računa o blokovima memorije datim na korišćenje se naziva Alokator Memorije.

Prva faza

Funkcionalni Zahtevi

1. Kako Python ne omogućava direktnu manipulaciju memorijom, memoriju ćemo simulirati velikom listom. Alokator treba da pruži sledeće usluge:
 - Alokaciju zahtevanog broja kontinualnih elemenata liste (rezultat može biti indeks prvog elementa liste ili neki drugi objekat koji predstavlja alociranu dinamičku memoriju).
 - Oslobađanje prethodno alociranog dela liste.
2. Pri odabiru dela liste kojim će ispuniti određen zahtev alokator može koristiti nekoliko algoritama, implementirati *FirstFit* algoritam i još jedan algoritam po izboru i omogućiti laku promenu odabranog algoritma i dodavanje još algoritama.

Nefunkcionalni Zahtevi

1. Sistem treba da bude dizajniran da bude otporan na greške, uključujući neispravne argumente i pogrešnu upotrebu zahteva.
2. Sistem treba da vodi računa o performansama, tako da minimalno utiče na vreme izvršavanja programa koji ga koristi.