

Organizace a správa dynamické objektové paměti. Základní techniky čištění: Mark & Sweep, Mark & Compact, Baker, Incremental Mark & Sweep

Objekt tabule

- tabulka obsahující adresy všech existujících objektů
- pro přímý přístup index v tabulce nebo jiný identifikátor

⊕ - jednotný přístup k objektům

- měním jedinou adresu pokud změním objekt

⊖ - časová náročnost přístupu

- GC musí hledat i kudy tabulka
- volné ide po odstraňování nahrazení

Direct reference

- charakteristika přímé na daný objekt

⊕ - výšší výkon

⊖ - složitější přemístění objektů (GC)

Immediate value

- pro malé hodnoty je rychléjší odkazovat objekt

→ každý přímý do vlastní adresy

- když je nejmenším kódu → můžeme vyuzít celý adresní prostor (adresy jsou rozložené na celou množinu čísel)

Garbage collection = automatická správa paměti

⊕ minimalizace neobsazené paměti + chybí jako dangling pointer a memory leak nejsou

⊖ náročnější na paměť, CPU

Reference counting

- obvious

- paměťové náročnější & nedokáže cykly

Mark & Sweep

- všechny objekty mají přímou ne

- algoritmus prochází všechny objekty od konstantní množiny (konstanty a registry) a dává ano

→ nahradí odhadem všechny s ne

- řešení se může stát velmi pomalým

- ⊕ - paměťové manažování (+1 bod)
 - řeší cyklické závislosti

- ⊖ - program nastaven při GC
 - dochází k fragmentaci paměti

Mark & Compact - řeší fragmentaci

- mark stejný jako předchozí
- compact přeštáčí celou haldu cel mýchých k vysokým adresám
 → různé bloky přesouvá na první prázdnou adresu (a pamatují si offsety)
- ve druhém průchodu aktualizuje adresy

Baker (copy compactor)

- haldu rozdělí na dvě části (s jednou se nepracuje)
- když dojde místo, různé objekty se přkopírují do druhé polky (a ba se může používat)
 - může buď přepsat při dalším přechodu

Incrementální algoritmy

- GC dostává vždy malý čas při kterém může pracovat
 → program se nerestavuje na dlouho

Tri color Mark & Sweep

- všechny uzly jsou bílé, kořenové jsou šedé
- šedé uzly procházejí všechny potomky ručením na šedé a uzly na černé
 → když už není šedý, máme všechny bílé
- abych nemuseli nové vytvořit uzel referenční činností
 → černý a šedý uzel je označen jako šedý

Generační algoritmy

prorování: - většina objektů umře mladá

- mají přímý vztah k dlouho
- mají velmi malé referenční mladá

→ skupina pro různé generace, s každou jinakě čas GC