

Kapitel 7: Garbage Collection

21. december 2017 13:26

Tre forskellige:

Reference counting

Garbage collection implies termination detection

Tracing

Side 110-121 i <http://www.cs.vu.nl/~tcs/da/dasides.pdf>

Processer er forsynet med hukommelse.

Objekter bærer pointer til lokale objekter og referencer til fjerne objekter.

Et rodobjekt kan oprettes i hukommelsen; Objekter er altid tilgængelige

ved at navigere fra et rodobjekt.

Formål med affaldssamling: At genvinde utilgængelige objekter.

Tre operationer ved processer til at opbygge eller slette en reference:

Oprettelse: Objektindehaveren sender en peger til en anden proces.

Duplikation: En proces, der ikke er objektejeren, sender en reference

til en anden proces.

Sletning: Referencen slettes ved sin proces.

Reference counting:

Reference tæller spor antallet af referencer til et objekt.

Hvis den falder til nul, og der ikke er pointere, er objektet skrald.

Fordel: Kan udføres i løbetid.

Ulempe: Kan ikke genvinde cyclic garbage.

Garbage collection implies termination detection

Garbage collection algoritmer kan omdannes til

(eksisterende og nye)
termineringsdetekteringsalgoritmer.
Givet en grundlæggende algoritme.
Lad hver proces p være vært for et kunstigt rodobjekt Op .
Der er også et specielt non-root objekt Z .
I begyndelsen holder kun initiativtagerne p en reference
fra Op til Z .
Hver grundlæggende meddelelse bærer en duplikering af
 Z -referencen.
Når en proces bliver passiv, sletter den sin Z -reference.
Grundalgoritmen afsluttes hvis og kun hvis Z er skrald.

Indirect reference counting \Rightarrow Dijkstra-Scholten
termination detection. Weighted reference counting \Rightarrow
Weight-throwing termination detection.