

De stack is het geheugen dat is gereserveerd als kladruijme voor een uitvoeringsreeks. Wanneer een functie wordt aangeroepen, wordt er bovenaan de stack een blok gereserveerd voor lokale variabelen en enkele boekhoudkundige gegevens. Wanneer die functie terugkeert, wordt het blok ongebruikt en kan het de volgende keer dat een functie wordt aangeroepen, worden gebruikt. De stack wordt altijd gereserveerd in een LIFO-volgorde (last in first out); het meest recent gereserveerde blok is altijd het volgende vrij te maken blok. Dit maakt het heel eenvoudig om de stack bij te houden; een blok van de stack bevrijden is niets meer dan één wijzer aanpassen.



In een stapel items liggen items op elkaar in de volgorde waarin ze daar zijn geplaatst, en je kunt alleen de bovenste verwijderen (zonder het hele ding omver te werpen).

Stapel als een stapel papieren

De eenvoud van een stapel is dat u geen tabel hoeft bij te houden met een record van elke sectie toegewezen geheugen; de enige statusinformatie die u nodig hebt, is een enkele aanwijzer naar het einde van de stapel. Om toe te wijzen en de toewijzing ongedaan te maken, verhoogt en verlaagt u gewoon die enkele aanwijzer. Opmerking: soms kan een stack worden geïmplementeerd om bovenaan een deel van het geheugen te beginnen en naar beneden uit te breiden in plaats van naar boven te groeien.

De heap is geheugen dat is gereserveerd voor dynamische toewijzing. In tegenstelling tot de stack is er geen afgedwongen patroon voor de toewijzing en de toewijzing van blokken van de heap; je kunt op elk moment een blok toewijzen en het op elk moment vrijmaken. Dit maakt het veel complexer om bij te houden welke delen van de heap op een bepaald moment zijn toegewezen of vrij zijn.

Elke thread krijgt een stack, terwijl er meestal maar één heap is voor de toepassing (hoewel het niet ongebruikelijk is om meerdere heaps te hebben voor verschillende soorten toewijzing).

In een hoop (de heap) is er geen bepaalde volgorde in de manier waarop items worden geplaatst. Je kunt items in elke volgorde bereiken en verwijderen omdat er geen duidelijk 'top'-item is.

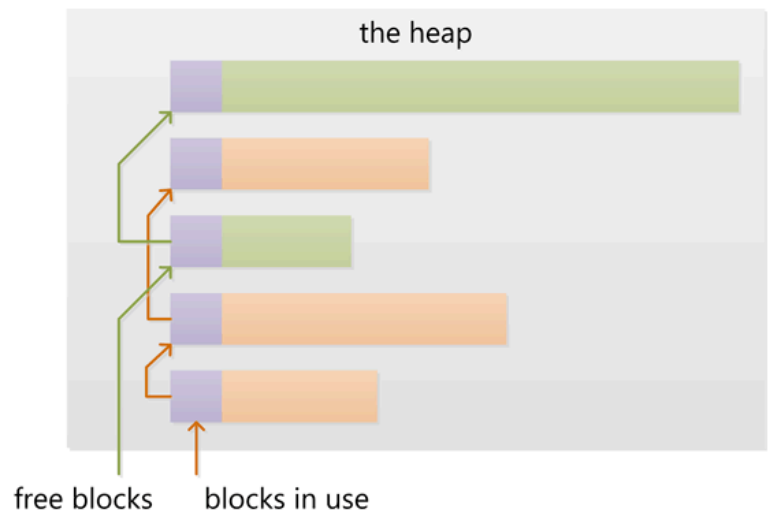


Hoop als een hoop drop allerhande

Heaptoewijzing vereist het bijhouden van een volledig overzicht van welk geheugen is toegewezen en wat niet, evenals wat overheadonderhoud om fragmentatie te verminderen, aaneengesloten geheugensegmenten te vinden die groot genoeg zijn om in de gevraagde grootte te passen, enzovoort. Geheugen kan op elk moment ongedaan worden gemaakt, waardoor er vrije ruimte overblijft. Soms voert een geheugentoe wijzer onderhoudstaken uit, zoals het defragmenteren van geheugen door toegewezen geheugen te verplaatsen, of het verzamelen van afval - tijdens runtime identificeren wanneer geheugen niet langer binnen bereik is en de toewijzing ervan ongedaan maken.

Als een gebruikt blok dat grenst aan een vrij blok, wordt opgeheven, kan het nieuwe vrije blok

worden samengevoegd met het aangrenzende vrije blok om een groter vrij blok te creëren, waardoor de fragmentatie van de heap effectief wordt verminderd.



Zie ook [deze post](#).