

Soru-1: Heap ve Stack alanları nedir, nasıl çalışır?

Cevap-1:

Heap ve Stack'i tanımlayacak olursak RAM içinde programların veri depoladığı iki farklı bellek alanıdır diyebiliriz.

Stack LIFO dediğimiz sistemle çalışır yani bir stack'e son giren eleman stack'ten ilk çıkacak elemandır. Stackler Heap'in aksine daha küçüktür ve hızlı yönetilebilir. Stack'te daha çok yerel değişkenlerimiz ve fonksiyonlarımızın bilgileri burada tutulur. Fonksiyon her çağrıldığında, o fonksiyona ait yerel değişkenler ve parametreler stack'te bir blok olarak saklanır. Fonksiyon işini bitirip sona erdiğinde, bu blok "stack'ten" temizlenir. Yani Memory, fonksiyon çağrıldığında tahsis edilir ve fonksiyon bitince serbest bırakılır. Heap'e göre hızlıdır çünkü bellek alanı çok hızlı bir şekilde ayrılır ve serbest bırakılır. Dezavantaj olarak da bellek alanı sınırlı olduğu için büyük veriler için yetersiz olabilir diyebiliriz.

Heap ise dinamik bellek yönetimi ve tahsisi için kullanılır. C++'da kullandığımız malloc, alloc, realloc gibi fonksiyonlar bunları sağlar. Burada Stack'in aksine Memory tahsis ettiğimiz veri program bitene kadar hafızada kalabilir, tabii bu tamamen kullanıcının isteğine bağlıdır. Özellikle bellek boyutunun bilinemediği durumlarda, örneğin bir listenin uzunluğu program çalışırken belirlendiğinde heap kullanmak gerekir. Heap için daha esnektir diyebiliriz ama yönetimi de bir o kadar daha zordur çünkü eğer doğru yönetilmezse memory leak dediğimiz hafızada işi biten memory'nin boşaltılmaması gibi durumlar yaşanabilir.

---

Soru-2: Garbage Collector nedir, ne işe yarar, nasıl çalışır?

Cevap-2:

Garbage Collector de aslında tam heap'in bu sorunu için devreye giriyor. Programın otomatik bellek yönetimi yapmasını sağlayan bir mekanizmadır diyebiliriz. Programın artık kullanılmayan daha teknik demek gerekirse referansı olmayan nesneleri ve verileri tespit eder ve bellekten serbest bırakır ve memory leak'i önler. Çalışma şekli olarak da bir nesneye kaç referans olduğunu sayar. Referans sayısı sıfır olduğunda, nesne çöplük olarak işaretlenir yani mark işlemi yapılır. Daha sonra da sweep yapılarak memory'den silinir. Ek olarak generation garbage collector mekanizması da vardır ve bu mekanizma belleği yaşlı ve genç nesneler olarak ikiye ayrılır. Genellikle daha yeni nesneler daha kısa ömürlü olduğundan, bu nesneler daha hızlı temizlenir.

Soru-3: x = 2 ataması yapıldığında x ve 2 nerelerde, nasıl tutulurlar?

Cevap-3:

x değişkeni, primitive bir tür olduğu için bellekte stack bölgesinde saklanır ve değeri olan 2 doğrudan x değişkeninin içinde yer alır. Eğer x dinamik olarak tahsis edilmiş olsaydı, heap bellekte depolanırdı. Ancak bu örnekte dinamik tahsis edilmediğinden, stack bellekte tutulur diyebiliriz.

---

Soru-4: arabalar = ['ford', 'dodge'] şeklinde bir liste oluşturulursa arabalar değişkeni, liste ve listenin elemanları nerede nasıl tutulurlar?

Cevap-4:

Bu örnekte de arabalar değişkeni bir liste olup stack bellekte saklanır ve heap bölgesinde oluşturulmuş olan liste nesnesinin adresini içerir. Öte yandan, asıl liste nesnesi heap bellekte bulunur ve içerisindeki elemanların adreslerini tutar. Yani şu şekilde görselleştirebiliriz;

liste nesnesi -> Heap

arabalar -> Stack

değişkenler -> Heap