Bir program neden çok fazla bellek kullanır?



www.yzokul.com.tr

Bir program neden çok fazla bellek kullanır?



Bir programın neden çok fazla bellek kullandığını merak edebilirsin. Kısa yanıt şudur: büyük veriler saklar, ihtiyaç duymadığı verileri kopyalar veya artık kullanmadığı şeyleri tutar. Örneğin, devasa bir sayı listesi yükleyen bir program hepsini aynı anda bellekte tutar. Bu üç alışkanlık, yüksek bellek kullanımının yaygın nedenleridir.



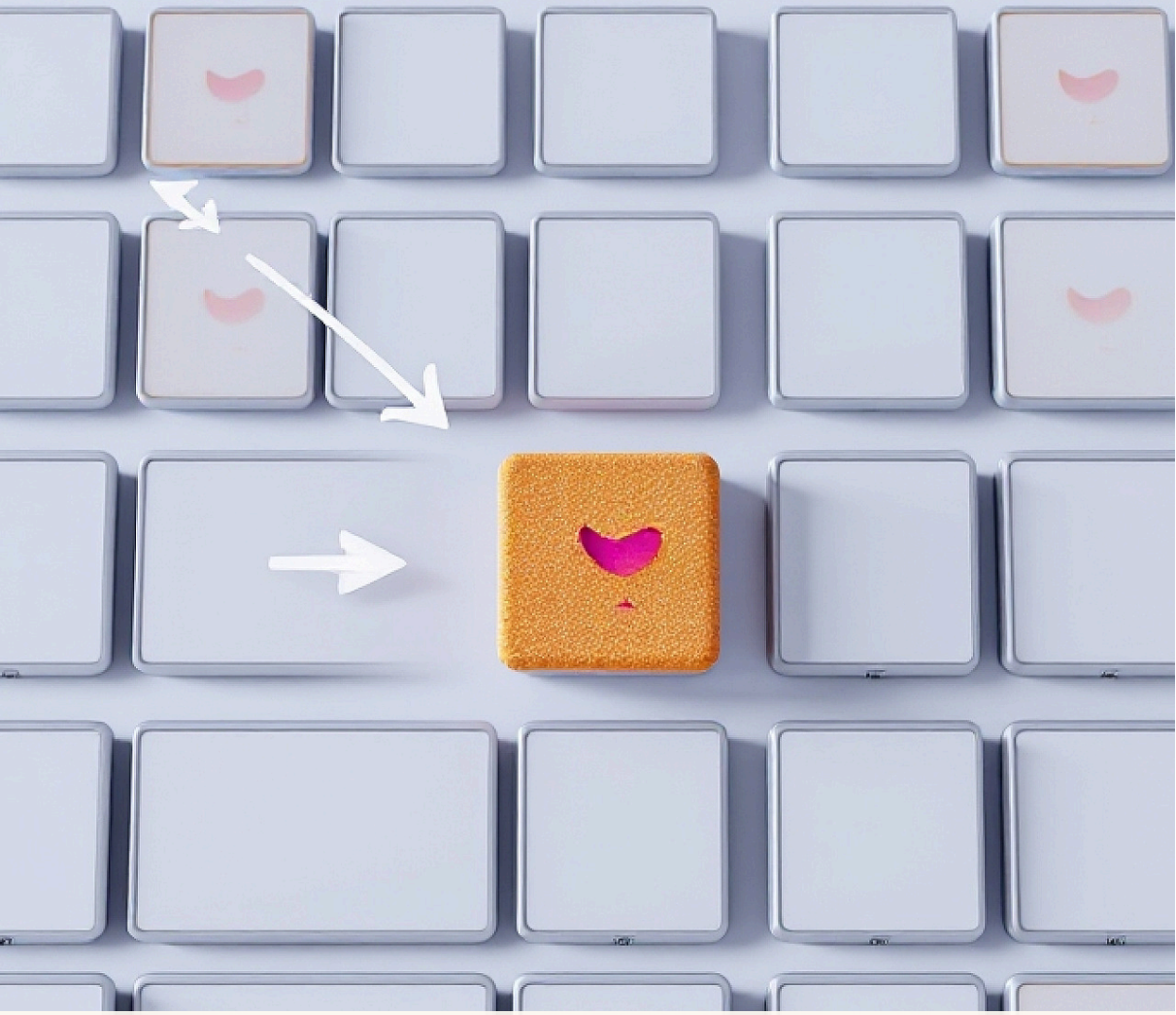
Bellek, programın çalışırken kullandığı çalışma alanıdır. Bunu, programın o anda kullandığı şeyleri koyduğu bir masa gibi düşün. Masada ne kadar çok şey varsa o kadar çok bellek gerekir. Çok sayıda program açtığında veya büyük dosyalar yüklediğinde masa kalabalıklaşır. Bunu anlamak, bellek kullanımının neden arttığını görmene yardımcı olur.



Çok fazla veriyi aynı anda tutmak çok fazla bellek alır. Devasa bir liste veya dosya yükleyen bir program hepsini aynı anda bellekte tutar. Örneğin, çok büyük bir metin dosyasını bir listeye okumak her satırı bellekte saklar. Dosya devasaysa bellek kullanımı da devasa olur. Büyük veri saklamak, yüksek bellek kullanımının doğrudan nedenidir.



Binlerce yüksek çözünürlüklü fotoğrafı aynı anda açmayı düşün. Her fotoğraf bellekte yer kaplar, bu yüzden binlercesi birlikte çok büyük miktarda bellek gerektirir. Tüm fotoğrafları aynı anda yükleyen bir fotoğraf görüntüleyici, bunları tek tek yükleyenden çok daha fazla bellek kullanır. Bu örnek, büyük verinin ne kadar hızlı biriktiğini gösterir.



Aynı bilgiyi tekrar tekrar kopyalamak belleği boşa harcar. Program sonra bir yerine birkaç özdeş kopya tutar. Örneğin, bir program ihtiyaç duyan her işlev için büyük bir listeyi kopyalarsa bellek kopyalarla dolar. Her kopya kendi yerini kaplar. Gereksiz kopyalardan kaçınmak, bellek tasarrufunun basit bir yoludur.



Bir program artık kullanmadığı nesneleri tutmaya devam edebilir. Bu unutulmuş nesneler bırakılana kadar hâlâ bellek kaplar. Örneğin, bir program eski sonuçları hiç temizlenmeyen bir listede tutabilir. Sonuçlara ihtiyaç olmasa bile bellekte kalırlar. Kullanılmayan nesneleri bırakmak, o alanı başka işler için serbest bırakır.



Verilerin düzenlenme biçimi, ne kadar bellek gerektiğini değiştirir. Bazı veri yapıları ek bilgi saklar ve diğerlerinden daha fazla yer kaplar. Örneğin, bir sayı listesi genellikle aynı sayıları anahtar olarak içeren bir sözlükten daha az bellek kullanır. Basit ihtiyaçlar için basit bir yapı seçmek bellek kullanımını düşük tutar. Bellek için veri yapıları önemlidir.



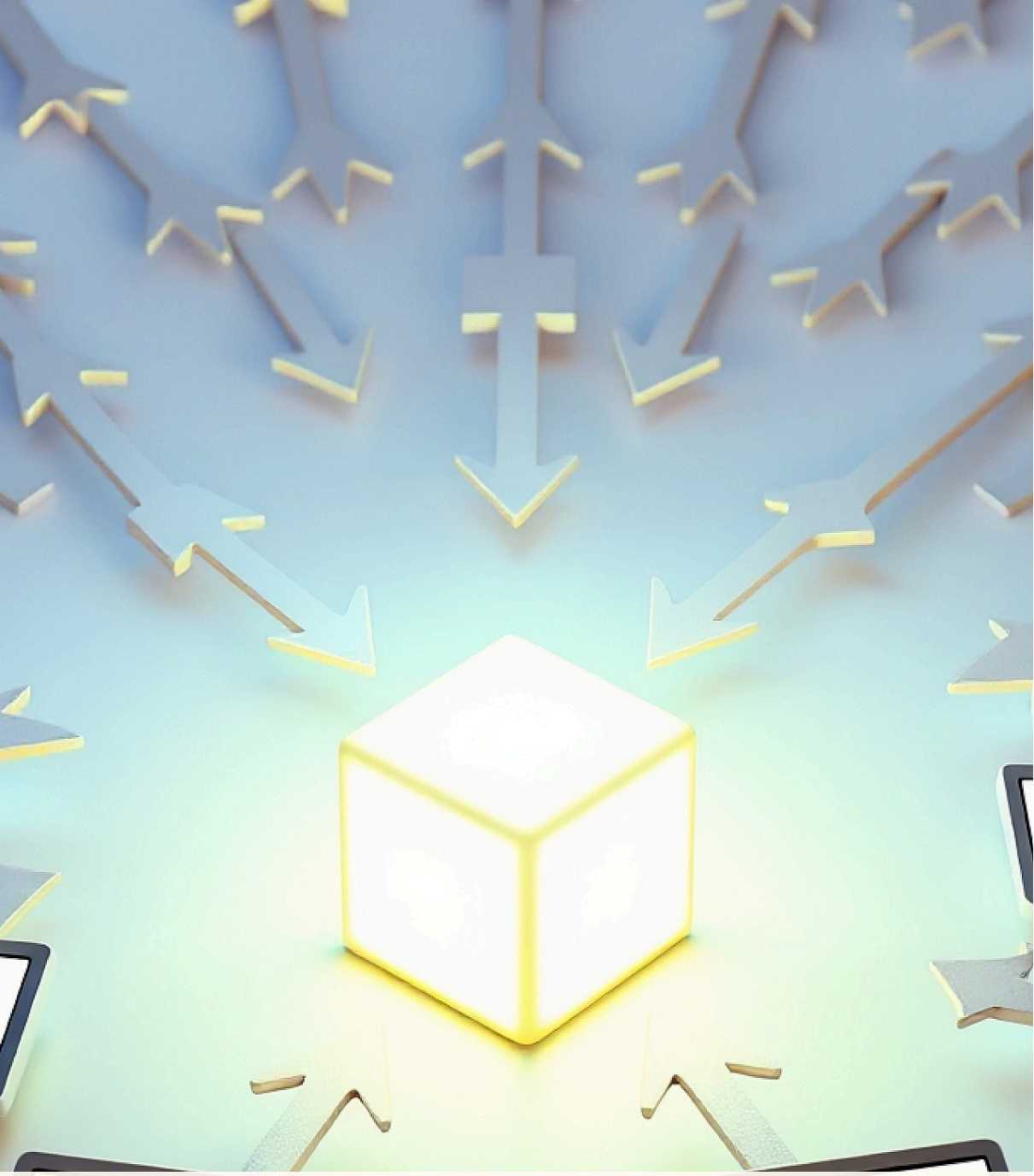
Aynı bilgiyi tutmanın iki yolunu karşılaştır. Bir yol diğerinden çok daha fazla bellek kullanabilir. Örneğin, bir milyon sayıyı küçük tam sayılardan oluşan bir listede saklamak, bunları metin dizeleri olarak saklamaktan daha az bellek kullanır. Veri aynıdır ama maliyet farklıdır. Doğru yapıyı seçmek bellek kullanımını düşük tutar.



Bir program daha fazla veri yüklerken bellek kullanımının arttığını izlemeyi hayal et. Bunun büyüdüğünü görmek, hangi bölümün en çok kullandığını fark etmene yardımcı olur. Belleği zaman içinde gösteren bir sistem aracı kullanabilirsin. Sayı sıçradığında, o bölümün ağır olduğunu anlarsın. Belleğin büyümesini izlemek, nedeni bulmanın pratik bir yoludur.



Belleği azaltmanın bir yolu, verileri hepsini birden değil parça parça işlemektir. Her parça kullanılır ve bir sonrakinden önce serbest bırakılır. Örneğin, büyük bir dosyayı tümünü yüklemek yerine satır satır oku ve her satırı işle. Parçalar hâlinde işlemek bellek kullanımını küçük ve sabit tutar.



Birçok kopya yerine tek bir paylaşılan veri parçası kullanmak bellek tasarrufu sağlar. Program bilgiyi çoğaltmak yerine aynı bilgiye işaret eder. Örneğin, bir programın birçok bölümü aynı büyük yapılandırmaya ihtiyaç duyuyorsa hepsi tek bir paylaşılan nesneye başvurabilir. Veriyi yeniden kullanmak israfı önler ve belleği düşük tutar.



Bir program artık ihtiyaç duymadığı nesneleri serbest bırakmalıdır. Bırakmak, belleği başka işler için serbest bırakır. Örneğin, büyük bir geçici listeyle işin bittiğinde onu temizle veya kapsam dışına çıkmasına izin ver. Birçok dilde bellek otomatik olarak geri kazanılır, ancak yine de başvuruları kaldırman gerekir. Kullanılmayanı bırakmak belleği sağlıklı tutar.



Yaygın bir hata, daha az belleğin her zaman daha hızlı olduğunu düşünmektir. Daha az bellek kullanmak her zaman daha hızlı çalışmak anlamına gelmez. Örneğin, verileri küçük parçalar hâlinde işlemek bellek tasarrufu sağlar ama tekrarlanan okumalar nedeniyle daha fazla zaman alabilir. Bazen daha fazla bellek kullanmak, ekstra işten kaçınarak işleri hızlandırır. Bellek ve hız ayrı konulardır.



Bir başka hata, çok fazla bellek kullanmanın her zaman yanlış olduğunu varsaymaktır. Bazen verileri bellekte tutmak basit ve hızlı seçimdir. Örneğin, küçük bir arama tablosunu bellekte tutan bir program, yeniden hesaplamadan soruları hızlıca yanıtlayabilir. Belleği akıllıca kullanmak, mümkün olduğunca az kullanmaktan daha iyidir. Daha fazla bellek her zaman kötü değildir.



Bellek kullanımı ve hız bir ödünleşimdir. Amaç, mümkün olduğunca az kullanmak değil, belleği akıllıca kullanmaktır. Örneğin, sonuçları bellekte önbelleğe almak tekrarlanan işi hızlandırır ama daha fazla bellek kullanır. İhtiyaçlarına göre karar ver: bellek darsa parçalar hâlinde işle; hız önemliyse veriyi el altında tut. Denge anahtardır.



Büyük veri, gereksiz kopyalar ve kullanılmayan nesneler yüksek bellek kullanımına neden olur. Verileri parçalar hâlinde işle ve ihtiyaç duymadığını bırak. Ama unutma: daha az bellek her zaman daha hızlı değildir. Belleği durumuna göre akıllıca kullan. Bu noktalar, bir programın neden çok fazla bellek kullandığını ve bunun nasıl yönetileceğini yanıtlar.