Bir program neden bir bilgisayarda çalışırken diğerinde çalışmayabilir?



www.yzokul.com.tr

Bir program neden bir bilgisayarda çalışırken diğerinde çalışmayabilir?



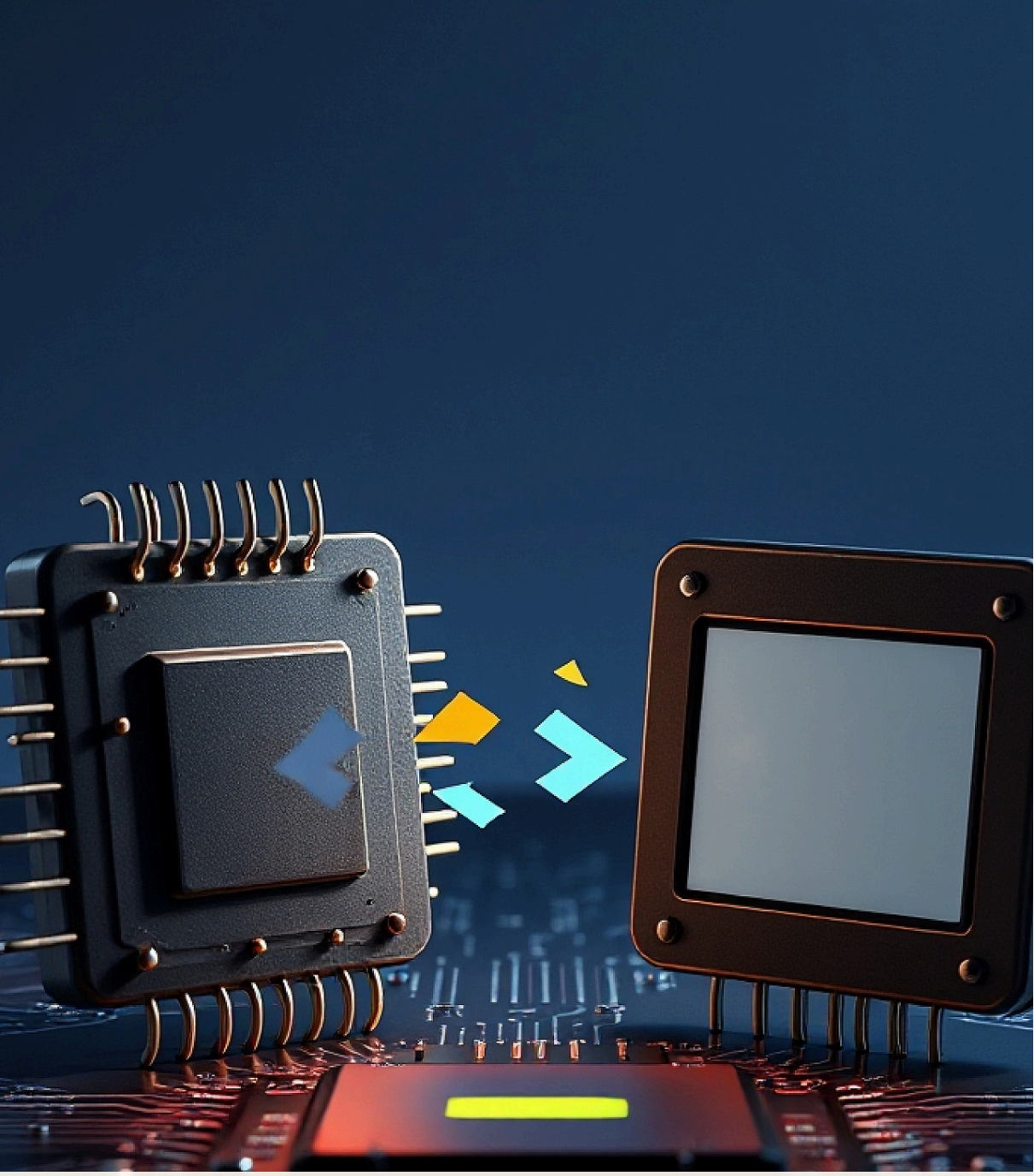
Bir programın bir bilgisayarda çalıştığını ama diğerinde çalışmadığını görebilirsin. Kısa yanıt, iki bilgisayarın aynı şekilde kurulmamış olmasıdır. Bir program, ortamındaki işletim sistemi, işlemci ve diğer dosyalar gibi birçok şeye bağlıdır. Bunlardan herhangi biri farklıysa veya eksikse program çalışmayabilir.



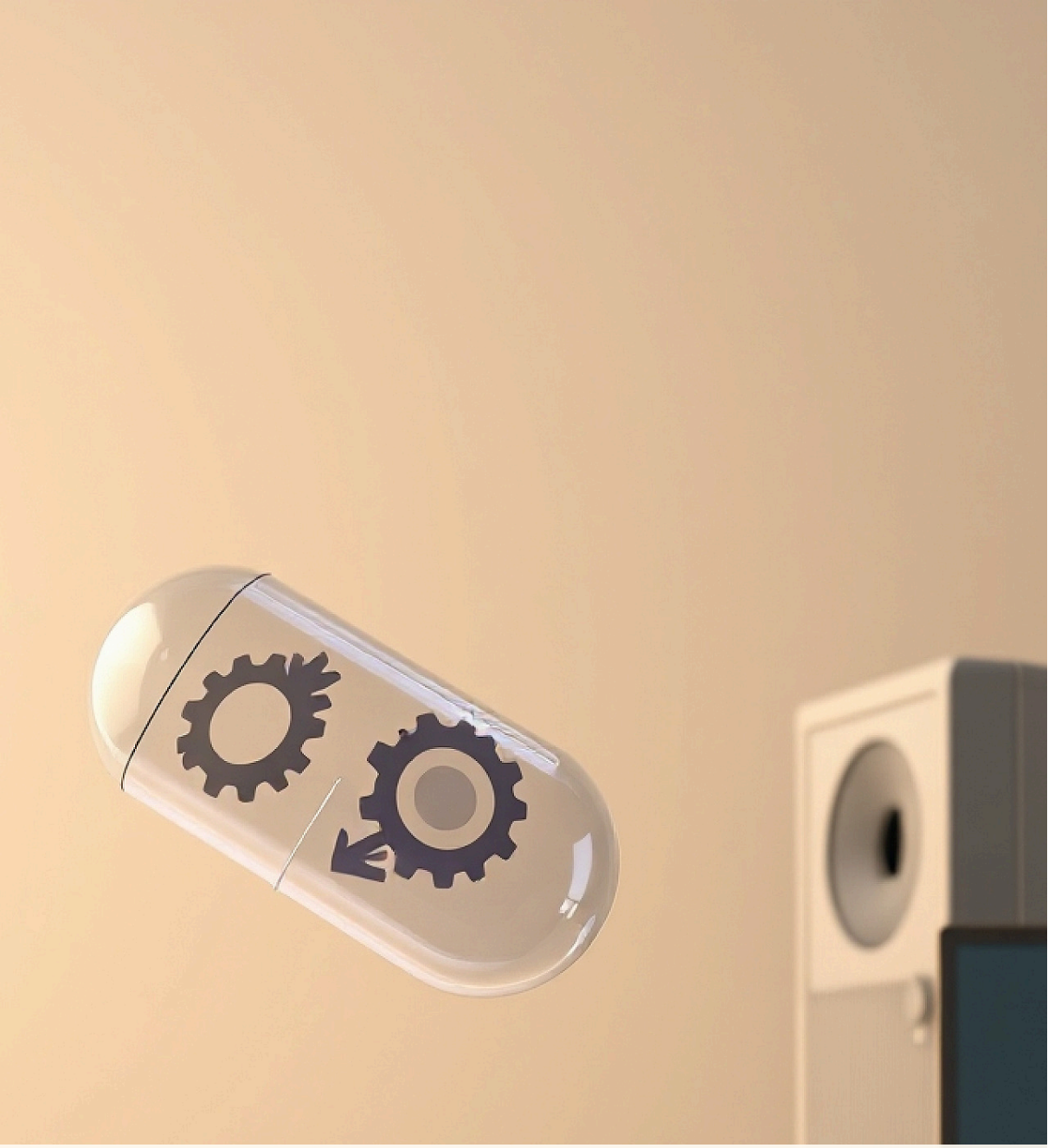
Bir program yalnız değildir. Çalışmak için bir bilgisayara, bir işletim sistemine ve çoğu zaman başka dosyalara ihtiyaç duyar. Bunlar birlikte programın ortamını oluşturur. Bu ortamın herhangi bir parçası ikinci bilgisayarda farklıysa veya eksikse program başlamayabilir ya da çalışmayı durdurabilir. Bir programın çalışmak için eksiksiz bir araç setine ihtiyaç duyduğunu düşün.



Her bilgisayar Windows, macOS veya Linux gibi bir işletim sistemi çalıştırır. Bu sistemler bilgisayarın kaynaklarını yönetir ve programlara hizmetler sunar. Bir işletim sistemi için oluşturulmuş bir program, sistemler farklı şekillerde çalıştığı için başka birinde çalışmayabilir. Örneğin, Windows için yapılmış bir program özel bir yardım olmadan Mac'te açılmayabilir.



İşlemciler farklı komut kümelerini, yani farklı dilleri anlar. Bir program genellikle derlenir, yani belirli bir işlemci türü için komutlara çevrilir. Bir işlemci türü için derlenmiş bir program, farklı bir işlemcisi olan bir bilgisayarda açılmayabilir. Bunun nedeni, komutların diğer işlemcinin anlamadığı bir biçimde olmasıdır.



Bir sistem için derlenmiş bir dosyayı düşün. Bu, diğer bilgisayarın okuyamadığı bir dilde yazılmış bir mektup gibidir. Bu yüzden başka bir sistemde doğrudan açılmayabilir. Dosya, yalnızca özgün sistemin işlemcisinin ve işletim sisteminin anlayabileceği bir biçimde komutlar içerir. Başka bir yerde çalıştırmak için o diğer sistem için yapılmış bir sürüme ihtiyacın olur.



Programlar çoğu zaman paylaşılan kütüphanelere dayanır; bunlar bilgisayarda zaten kurulu kod koleksiyonlarıdır. Bu kütüphaneler birçok programın kullandığı ortak işlevleri sağlar. Diğer bilgisayarda gerekli kütüphaneler yoksa program başlayamaz. Örneğin, bir program görüntüleri işlemek için bir kütüphaneye ihtiyaç duyabilir ve bu kütüphane eksikse program başarısız olur.



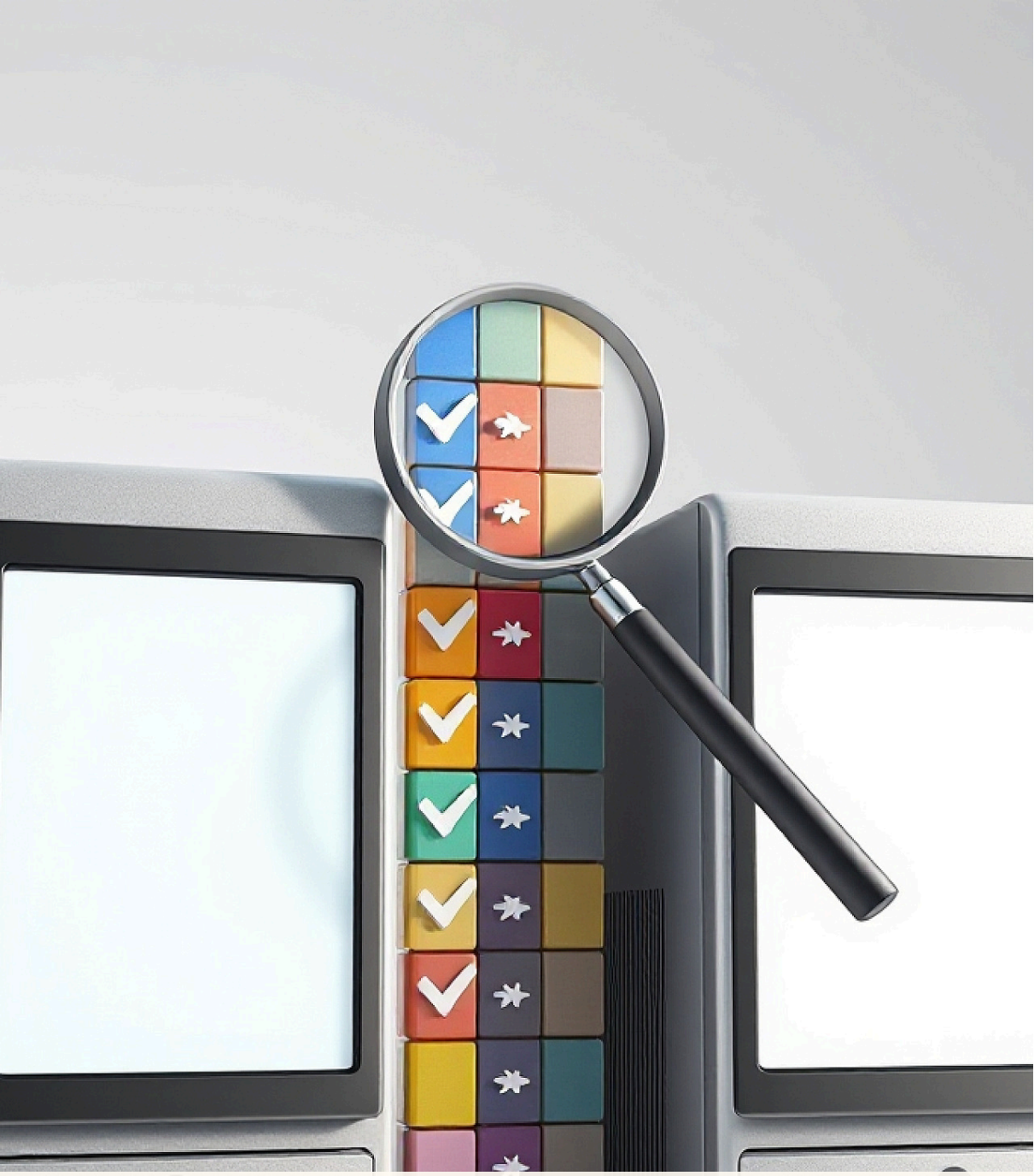
Bir programın çalışması için veri dosyaları veya yapılandırma dosyaları gibi ek dosyalara ihtiyacı olabilir. Bu dosyalar programın kendisinin parçası değildir ama doğru çalışması için gereklidir. Bu dosyalar ikinci bilgisayarda eksikse program bozulabilir veya hatalar gösterebilir. Örneğin, bir oyunun seviye bilgilerini içeren bir veri dosyasına ihtiyacı olabilir ve bu olmadan oyun yüklenemez.



Aynı kütüphane veya sistem farklı sürümlerde gelebilir. Bir sürüm için yapılmış bir program daha eski veya daha yeni bir sürümle çalışmayabilir. Bunun nedeni, sürümlerin işlerin nasıl çalıştığını değiştirebilmesi veya özellikleri kaldırabilmesidir. Örneğin, bir program bir kütüphanenin 2.0 sürümüne ihtiyaç duyabilir ama diğer bilgisayarda 1.0 sürümü vardır ve bu programın başarısız olmasına yol açar.



Bir bilgisayardaki izinler veya ortam ayarları gibi ayarlar, bir programın çalışıp çalışmayacağını etkiler. İzinler, kimin dosyalara erişebileceğini ve programları çalıştırabileceğini kontrol eder. Ortam ayarları programlara şeyleri nerede bulacaklarını söyler. Farklı ayarlar aynı programın başarılı veya başarısız olmasına yol açabilir. Örneğin, bir kullanıcının gerekli bir dosyayı okuma izni yoksa program başlayamaz.



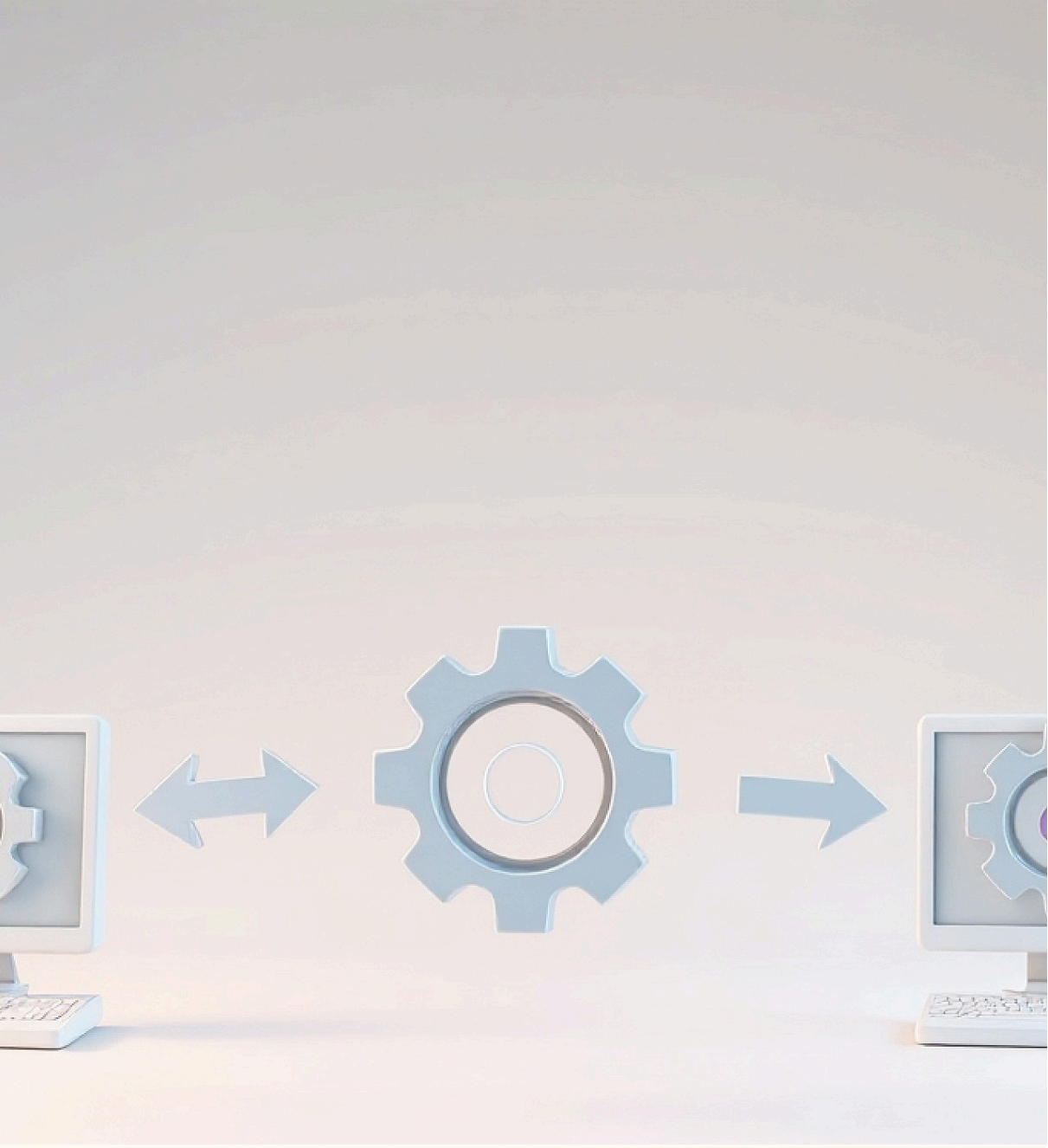
Aynı programı iki bilgisayarda yan yana karşılaştır. Farklı olanları listele: işletim sistemi, işlemci, kurulu kütüphaneler, mevcut dosyalar, kütüphanelerin sürümleri ve ayarlar. Bu farklılıklardan herhangi biri, programın neden bir bilgisayarda çalışıp diğerinde çalışmadığını açıklayabilir. Bu listeyi yapmak nedeni bulmana yardımcı olur.



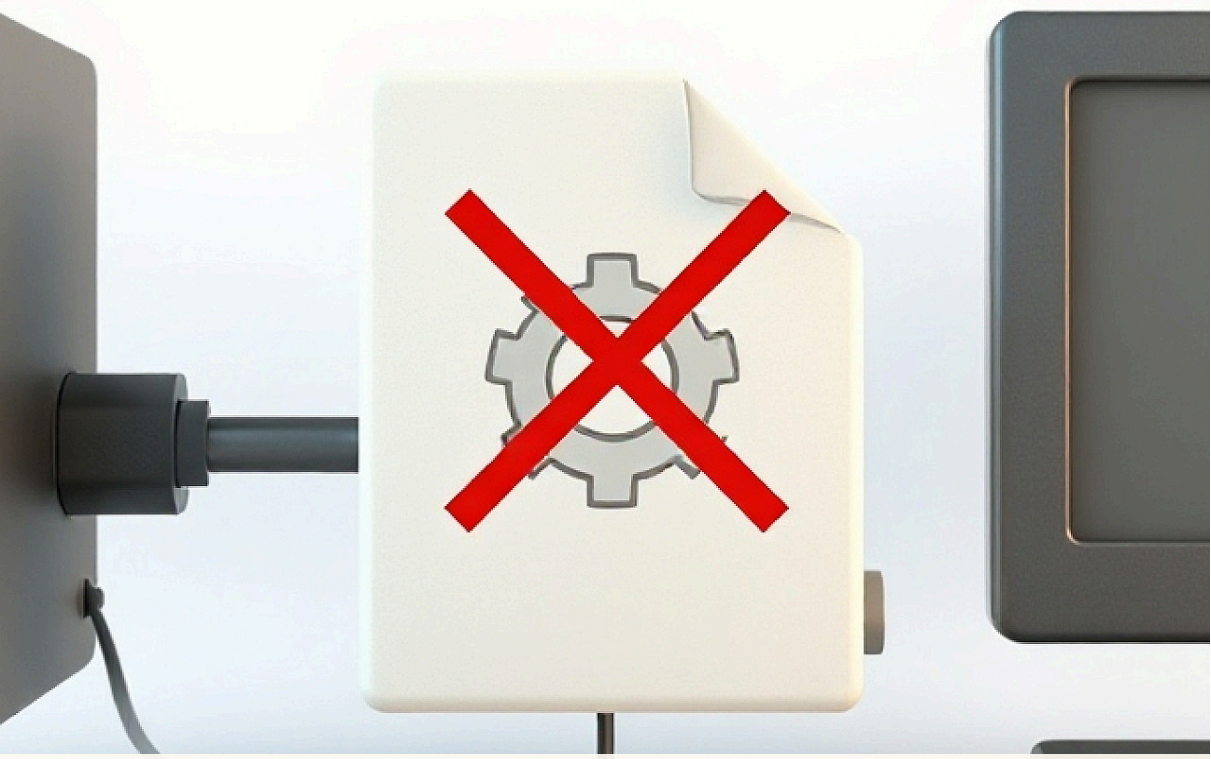
Yazılım, çalışmak için ihtiyaç duyduğu ortamı listeler. Bu gereksinimler, programın çalışması için bilgisayarda ne olması gerektiğini söyler. İşletim sistemini, işlemci türünü, belleği ve gerekli kütüphaneleri içerebilir. Örneğin, bir programın gereksinimleri Windows 10 ve en az 4 GB bellek gerektiğini söyleyebilir. Bu, bilgisayarının onu çalıştırıp çalıştıramayacağını bilmene yardımcı olur.



Bir programı kurmadan önce gereksinimlerini oku. Bunları bilgisayarının gerçekte sahip olduklarıyla karşılaştır. İşletim sistemi sürümünü, işlemci türünü ve gerekli kütüphanelere sahip olup olmadığını kontrol et. Bir şey eksikse program çalışmayabilir. Örneğin, program sahip olmadığın bir kütüphaneye ihtiyaç duyuyorsa önce onu kurabilirsin.



Taşınabilirlik, aynı yazılımın farklı ortamlara uygun bir biçimde çalışabilme yeteneğidir. Taşınabilir yazılım, birden fazla sisteme uyacak şekilde yapılır. Bu, farklı işletim sistemleri için farklı sürümler sağlamak veya sistemler arasında çalışan yöntemler kullanmak anlamına gelebilir. Örneğin, taşınabilir bir uygulama Windows, macOS ve Linux için sürümler halinde gelebilir.



Yaygın bir hata, bir dosyanın her bilgisayarda çalıştığını varsaymaktır. Aynı dosya farklı bir sistem için farklı bir biçime ihtiyaç duyabilir. Örneğin, Windows için derlenmiş bir program Linux için yapılmış bir sürüm olmadan Linux'ta çalışmaz. Dosyanın kullandığın bilgisayar için doğru biçimde olduğunu her zaman kontrol et.



Bir diğer yaygın hata, sürümleri ve ayarları göz ardı etmektir. Bunları önce kontrol etmek çoğu zaman bir programın neden bir bilgisayarda başarısız olduğunu açıklar. Kütüphanelerin ve işletim sisteminin sürümlerinin programın beklediğiyle eşleştiğinden emin ol. İzinler gibi ayarları da kontrol et. Örneğin, bir program kurulu olandan daha yeni bir kütüphane sürümüne ihtiyaç duyduğu için başarısız olabilir.



Programlar ortamlarına bağlıdır, bu yüzden sistem, işlemci, kütüphaneler, dosyalar, sürümler veya ayarlardaki farklılıklar onların başarısız olmasına yol açabilir. Kurmadan önce gereksinimleri kontrol etmeyi ve taşınabilirliği düşünmeyi unutma. Örneğin, bir program bir bilgisayarda çalışıp diğerinde çalışmıyorsa farkı bulmak için ortamlarını karşılaştır. Bu, sorunu anlamana ve çözmene yardımcı olur.