UNIX Türevi Sistemlerde Sistem Fonksiyonları, POSIX Fonksiyonları ve Standart C Fonksiyonları

Kaan Aslan-Sebahat Ersoy
3 Mart 2009

İşletim sistemlerini fonksiyonlardan oluşan kontrol yazılımları olarak ele alabiliriz. Gerçekten de işletim sistemlerinin çekirdekleri tipik olarak fonksiyonların birbirlerini çağırması biçiminde prosedürel teknikler kullanılarak yazılmaktadır. İşte çekirdeğe yönelik temel işlemleri yapan bir grup fonksiyon *kullanıcı modunda (user mode)* çalışan uygulama programları tarafından da çağrılabilir. Bu tür fonksiyonlara *sistem fonksiyonları (system call)* ya da *API (Application Programming Interface)* denilmektedir. Sistem fonksiyonları o sisteme özgüdür. Bu nedenle bunların sayıları, isimleri ve parametrik yapıları sistemden sisteme farklılıklar gösterebilir. Örneğin, *Windows* sistemlerinde dosya açmak için *CreateFile* isimli sistem fonksiyonu kullanılırken, *Linux* sistemlerinde *open (sys_open)* isimli bir fonksiyon kullanılmaktadır. Bunların parametrik yapıları tamamen birbirinden farklıdır. *UNIX* türevi iki sistem arasında bile sistem fonksiyonları bakımından önemli farklılıkların söz konusu olabileceğini belirtelim.

POSIX fonksiyonları *POSIX* standartlarında belirtilmiş arayüz fonksiyonlardır. Eğer bir sistem *POSIX* uyumluysa (aslında sistemlerin belirli derecede uyumlu olması mümkündür), bu fonksiyonlar o sistemde aynı isimde ve parametrik yapıda bulunmak zorundadır. Böylece programcı bu fonksiyonları standart *UNIX* fonksiyonları olarak kullanabilir. Şüphesiz *POSIX* fonksiyonları da sistemle ilgili işlemler söz konusu olduğunda kendi içlerinde sistem fonksiyonlarını çağırırlar. Bazı *POSIX* fonksiyonları çalıştığınız sistemde tam olarak bir sistem fonksiyonuna karşılık gelir. Örneğin, *Linux* sistemlerinde *chmod POSIX* fonksiyonu bire bir *sys_chmod* isimli sistem fonksiyonuna karşılık gelmektedir. Yani *chmod* fonksiyonunun yaptığı tek şey *sys_chmod* fonksiyonunu çağırmaktır. Bazı *POSIX* fonksiyonları kendi içlerinde birden fazla sistem fonksiyonunu çağırıyor olabilir. Bazı *POSIX* fonksiyonları da hiçbir sistem fonksiyonunu çağırmadan tamamen kullanıcı modunda işlem yapıyor olabilir.

UNIX türevi sistemlerde programcının öncelikle *POSIX* fonksiyonlarını kullanmasını tavsiye ederiz. Böylece programınızın *POSIX* uyumlu sistemler için taşınabilirliğini sağlamış olursunuz. Peki sistem fonksiyonlarını doğrudan çağırmanın gerektiği durumlar söz konusu olabilir mi? Bazen evet. Çalıştığınız sistem başka sistemlerde bulunmayan bazı özelliklere ve olanaklara sahip olabilir. Diğer standartlarda olduğu gibi *POSIX* de ortak özellikler temel alınarak oluşturulmuştur. Bazı durumlarda o sistemin *POSIX* fonksiyonlarıyla erişemediğiniz olanaklarını kullanmak isteyebilirsiniz. Bu durumda o sisteme özgü sistem fonksiyonlarını doğrudan çağırmak gerekebilir. Örneğin çalıştığınız sistemde *POSIX*'in *open* fonksiyonunda olmayan özel bir dosya açış modu bulunuyor olabilir. *Linux* sistemlerinde *POSIX* karşılığı olmayan birçok sistem fonksiyonu vardır.

Standart *C* fonksiyonları hangi sistem söz konusu olursa olsun *C* derleyicilerinde bulunması gereken fonksiyonlardır. Bu fonksiyonlar yalnızca *UNIX* türevi sistemlerde değil diğer sistemlerdeki *C* derleyicilerinde de vardır. Şüphesiz en taşınabilir olanı

standart *C* fonksiyonlarını kullanmaktır. Fakat standart *C* fonksiyonları çok temel işlemleri yapan fonksiyonlardır. Büyük bir uygulamanın yalnızca bu fonksiyonları kullanarak yazılması söz konusu olamaz.[1]

Bunların dışında *C* derleyicilerinin ek birtakım fonksiyonları da söz konusu olabilir. Derleyicilerin standartlarda belirtilenlere ek olarak sahip olduğu bu tür fonksiyonlara (ve genel olarak özelliklere) *eklentiler (extensions)* denilmektedir. Bazı eklentiler çok yaygın biçimde pek çok *C* derleyicisinde bulunur. Örneğin, *strupr* bir standart *C* fonksiyonu değildir fakat pek çok *C* derleyicisinin bu fonksiyonu barındırdığını görüyoruz.

---

[1] Standart *C* fonksiyonları aynı zamanda birer *POSIX* fonksiyonudur ve *POSIX.1* standartlarında listelenmiştir.