

UNIX Türevi Sistemlerde Dizinlerin Erişim Hakları

2-Ekim-2009

Bana çok sorulan sorulardan biri de budur. Diyorlar ki, “yahu neden global değişkenlere varsayılan bir değer (tipik olarak sıfır değeri) atanıyor da yerel değişkenlere atanmıyor? Bu hak mı, adalet mi?”. Özellikle *C#* ve *Java* gibi içerisine değer atanmamış değişkenlerin kullanılmasına izin vermeyen dillerde çalışanlar yerel değişkenlerin otomatik olarak sıfırlanmamasından memnun değiller. Örneğin:

```
{
    int i, total;
    //...
    for (i = 0; i < 100; ++i)
        total += i;    /* C ve C++'ta çöp değer,
                       C# ve Java'da error! */
    .....
}
```

Sizce de yerel değişkenler derleyici tarafından sıfırlansaydı daha iyi olmaz mıydı?..

Aslında bu sorunun oldukça basit bir açıklaması var. Genel olarak statik ömürlü nesnelere (*C* ve *C++*'taki global ya da *static* yerel nesnelere, sınıfların *static* veri elemanlarına vs.) derleme aşamasında derleyici tarafından ilkdeğerleri verilebiliyor.^[1] Bu işlem derleme aşamasında yapıldığı için bir etkinlik sorununa yol açmıyor. Halbuki yerel değişkenler programın akışı değişkenin tanımlandığı yere geldiğinde (tipik olarak akış değişkenin tanımlandığı bloğa girdiğinde) yaratılıyorlar ve bloktan çıktığında yok ediliyorlar.^[2] İşte derleyicilerin yerel değişkenleri sıfırlaması bu nedenle bir verimlilik sorunu oluşturabilmektedir. Yani örneğin:

```
{
    int a;

    ...
    a = 10;
    ...
}
```

Burada derleyici yerel değişkeni yaratıp içerisine sıfır değerini makina komutlarıyla atamak zorunda. Fakat gördüğümüz gibi bu örnekte derleyicinin atadığı sıfır değeri daha sonra değiştirilmiş. O halde derleyici değişkeni sıfırlamış olsaydı burada gereksiz bir atama yapılmış olacaktı. Tabi siz de şimdi diyeceksiniz ki “öyleyse derleyici de her zaman değil, biz değişkene atama yapmamışsak onu sıfırlasın.” Ehm...Haklısınız ne diyeyim? Fakat benim de buna karşılık söyleyeceğim bir çift sözüm var:

1. Derleyicinin bunu yapması yine de verimsiz olur. Çünkü derleyici bazı durumlarda değişkeni farklı yerlerde defalarca sıfırlamak zorunda kalabilir. (Örneğin bir yerel değişkene *if* deyiminin doğru kısmında atama yapılmış, yanlış kısmında atama yapılmamış olsun. Böyle bir sürü durumun peşi sıra geldiğini düşünün.)

2. Ayrıca bunun gerçek anlamda programcıya bir faydası yok gibi gözüküyor. Programcının değer atamadığı bir yerel değişkeni yanlışlıkla kullandığını düşünelim. Bu durumda sıfırlamanın programcıya ne faydası olacak? Halbuki böylesi durumlarda derleyicinin *error* (C# ve Java'da) ya da uyarı mesajlarıyla (C ve C++'ta) programcuyu durumdan haberdar etmesi daha uygun değil midir? (Dışsal bağlama (*external linkage*) özelliğine sahip statik ömürlü nesnelere için böyle bir uyarı mekanizmasını ayarlamak mümkün gözüküyor.)

^[1] Bazı dillerde statik ömürlü nesnelere verilen ilkdeğerlerin sabit ifadesi olması zorunlu değil. Bu tür dillerde ilkdeğerleme işlemi derleyicilerin başlangıç kodları (startup code) tarafından yapılıyor.

^[2] Semantik olarak böyle olsa da pek çok derleyici (çeşitli, nedenlerle) tüm yerel değişkenleri akış fonksiyona girdiğinde yaratıyor, çıktığında yok ediyor.