REENTRANT – INLINE

By Mustafa Onur Parlak

- **Reentrant**
  - Bir fonksiyon yürütmenin ortasında kesilebiliyor ve güvenli bir şekilde yeniden çağrılabiliyorsa **reentrant** olarak adlandırılır.
  - Kesintiye bir iç işlem, interrupt ya da sinyal gibi dış işlem sebep olabilir.
  - Reentred çağrılar tamamlandığında önceki çağrı devam eder.

Yanlış çalışan bir REENTRANT fonksiyonu

```
int32_t i;

/* Burada Fonksiyon1 REENTRANT değildir.
*  Çünkü global i değişkeni kullanılmıştır
*/
int32_t Fonksiyon1()
{
    return i * 5;
}

/* Burada Fonksiyon2 REENTRANT değildir.
*  Çünkü non-reentrant kullanılmıştır
*/
int32_t Fonksiyon2()
{
   return fun1() * 5;
}
```

Doğru çalışan bir REENTRANT fonksiyonu

```
/* Burada Fonksiyon1 REENTRANT örneğidir.
*  Çünkü Execution (yürütme) duraklatan ve kontrolü
*  Fonksiyon1'e kaydıran bir kesme kullanılmıştır
*  Burada Fonksiyon1 tamamlandıktan sonra
*  Kontrol tekrar Fonksiyon2'ye aktarılır ve tekrar
*  Çalışmaya başlar
*/
int fonksiyon1(int3_t i)
{
    return i * 5;
}

int fonksiyon2(int3_t i)
{
   return fonksiyon1(i) * 5;
}
```

- **Inline Fonksiyonu**
    - Fonksiyon çağrısının yükünü azaltmak için kullanılan bir optimizasyon yönetimidir.
    - Derleyiciye, fonksiyonun çağrıldığı yerde sadece fonksiyon gövdesini koymasını sağlar
- **Avantajları**
    - Fonksiyon çağırma yükünü kurtarır
    - Return çağrısının yükünden kurtarır
- **Dezavantajları**
    - Fonksiyon boyutunu arttırabilir, bu da **cache** kaybına sebep olur.
    - Eğer register kullanacak olan değişkenler sayısı artarsa, register değişken kaynak kullanımı üzerinde genel yük yaratabilir
    - Biri kodu değiştirdiğinde, çağrıldığı her yer yeniden derlenir
    - Header dosyasında kullanılırsa dosyanın boyutu büyük olur ve okunamaz hale gelebilir.
    - Gömülü sistemler için kullanışlı değildir, büyük boyutlu **binary'ler** tercih edilmez