

# **BİLGİSAYAR İŞLETİM SİSTEMLERİ**

## **ÖDEV III**

**Hazırlayan : Beycan Kahraman**  
**No : 040020337**  
**Teslim Tarihi : 05.05.2006**  
**Öğretim Görevlisi : Yrd. Doç. Dr. A. Şima Etaner - Uyar**

### AMAÇ:

Verilen bu son ödevin amacı, semafor ve ortak bellek kullanımının etkin olarak kullanılabilmesidir. Ödevde bir adet sayma semaforu, iki adet sırayla çalışmayı sağlayacak semafor ve bir adet ortak bellek değişkeni kullanılmıştır.

### GELİŞTİRME:

Hazırladığım programı geliştirirken TUBİTAK ve UEKAE tarafından geliştirilen Linux işletim sisteminde, GCC 3.4.4 derleyicisini kullandım.

### FONKSİYONLAR:

Tüm semafor ve ortak bellek kullanımında yararlandığımız kütüphane ve fonksiyonların yanı sıra uygulamalarda verilen **sem\_wait(semafor, no)** ve **sem\_signal(semafor, no)** fonksiyonlarından yararlanılmıştır. Bunların dışında başka fonksiyon ya da yordamlara başvurulmamıştır.

### GEREKLİ TANIMLAMALAR:

Programımı hazırlarken yukarıda açıklandığı üzere üç semafor yapısı kullandım. Bunlardan ilki catidaki prosesin(anne), çocukları beklemesini sağlayan **sem\_cati(0)** semaforudur. BU semafor yardımı ile catidaki görevli aşağıdaki işçilerin taşıyıcıyı doldurmalarını bekleyecektir.

Kullandığım diğer iki proses ise **sem\_isci[i]** semaforlarıdır. Bu semaforlar yardımı ile çalışan işçilerin çalışma sıraları düzene sokulmuş ve taşıdıkları kiremit sayısı eşitlenmeye çalışılmıştır.

Kullanılan semaforların dışında, toplam taşınan kiremit sayısını tutan **sayi** ortak bellek değişkeni kullanılmıştır.

### PROGRAM:

Program yazılırken aşağıdaki pseudocode gözönünde tutulmuştur.

Çatıdaki (Anne Proses):

```
Gerekli prosesleri ve ortak bellek alanını hazırla;  
ilk çalışacak işçiyi seç. (bir tanesini durdur)  
while ( tasinan_kiremit_sayisi < toplam_gerekli_kiremit_sayisi )  
    Çocukları uyandır;  
    Çocukların sonlanmalarını bekle (ilk sayma semaforu ile);  
    Gelen taşıyıcıyı boşalt ve geri gönder;  
Semafor ve ortak bellek alanını sisteme iade et.
```

İşçiler (Çocuk Prosesler):

```
Semafor ve ortak bellek alanlarıyla gerekli bağlantıları kur;  
while( tasinan_kiremit_sayisi < toplam_gerekli_kiremit_sayisi )  
    kendi iznini kapat;  
    kiremit_ekle();  
    diğer_işçiye izin ver;  
    if( taşıyıcı şu anki adımda veya bir sonraki adımda dolacak )  
        catidakin_i_uyar();
```

**SONUÇ:** Sonuçta büyük oranda istenen sonuca yaklaşıldıysa da, program çıktısında %100 oranında doğru sonuç elde edilememiştir. Ara sıra, beklenenin tersine ölümcül kilitlenmeye rastlanmaktadır. Tüm uğraşlarıma rağmen bunun nedenini bulamadım.