

# Sistem Programlama

## Ders 15

Dr. Öğr. Üyesi Mehmet Dinçer Erbaş  
Bolu Abant İzzet Baysal Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü

# Sinyaller

- kill ve raise fonksiyonları
  - kill fonksiyonu bir işleme veya bir işlem grubuna sinyal göndermek için kullanılır.
  - raise fonksiyonu ile bir fonksiyon kendine sinyal gönderebilir.

```
#include <signal.h>
```

```
int kill(pid_t pid, int signo);
```

```
int raise(int signo);
```

Dönüş: OK ise 0, hata ise -1.

- raise(signo) fonksiyonu kill(getpid(), signo) ile aynı işe yarar.

# kill ve raise fonksiyonları

- pid argümanının alabileceği değere göre dört farklı durum söz konusudur.
  - pid > 0
    - Sinyal işlem numarası pid olan işleme gönderilir.
  - pid == 0
    - Sinyal fonksiyonu çalıştıran işlemle aynı işlem grup numarasına sahip işlemlere gönderilir. Sinyalin gönderilmesi için gerekli iznin olması gerekir.
      - Sistemle ilgili işlemlere sinyal gönderilemez (örneğin init).
  - pid < 0
    - Sinyal işlem grup numarası pid değerinin mutlak değerine eşit olan bütün işlemlere gönderilir. Önceki durumda olduğu gibi sinyalin gönderilme izni olmalıdır ve sistem işlemleri ayrı tutulur.
  - pid == 1
    - Sinyal sistemdeki bütün işlemlere gönderilir. Sinyalin gönderilmesi için gerekli izin olmalıdır ve sistem işlemleri ayrı tutulur.

# kill ve raise fonksiyonları

- Bir işlem başka bir işleme sinyal gönderebilmek için gerekli izinleri olmalıdır.
  - Sistem yöneticisi istediği bütün işlemlere sinyal gönderebilir.
  - Öbür kullanıcılar için sinyali gönderenle sinyali alanın efektif kullanıcı numarası aynı olmalıdır.
  - SIGCONT sinyali her türlü işleme gönderilebilir.
  - POSIX.1 standardına göre \_POSIX\_SAVED\_IDS biti ayarlı ise efektif kullanıcı numarası yerine saved-set-used-ID kontrol edilir.
- POSIX.1 standardına göre sinyal numarası 0 null sinyaldir.
  - Bu sinyal gönderildiğinde sinyal gönderilmez. Hata denetimi ise yapılır.
    - Eğer bir işleme null sinyal gönderildi ve bu işlem yok ise, kill fonksiyonu -1 döner ve errno ESRCH olarak değiştirilir.
    - Ancak işlem numaraları tekrar kullanıldığı için sinyali gönderdiğiniz işlemin kontrol ettiğiniz işlem olup olmadığı konusuna dikkat edilmesi gerekir.
    - Ayrıca kill ile test yapma operasyonu atomik değildir.
      - Bu sebeple alınan cevaba fazla güvenmemek gereklidir.

# alarm ve pause fonksiyonları

- Alarm fonksiyonu belirtilen süre sonra sonlanacak bir zamanlayıcı ayarlamak için kullanılır.
  - Zaman dolduğunda SIGALRM sinyali oluşur.
  - Bu sinyal görmezden gelinirse veya yakalanmazsa, varsayılan aksiyon işlemin sonlanmasıdır.

```
#include <unistd.h>
```

```
unsigned int alarm(unsigned int seconds);
```

Dönüş: 0 veya önceki alarm için kalan süreyi döner.

- Her işlemin alarm için bir saati bulunur. Bu sebeple daha önce ayarlanmış ve sonlanmamış bir alarm var ise fonksiyon sonlanması için kalan süreyi döner.
- Eğer ayarlanmış ve sonlanmamış bir alarm var ve sinyal süresi olarak 0 verilirse önceki sinyal iptal edilir. Bu durumda fonksiyon kalan süreyi döner.

# alarm ve pause fonksiyonları

- Pause fonksiyonu çağırılan işlemi bir sinyal alıncaya kadar durdurur.

```
#include <signal.h>
```

```
int pause(void);
```

Dönüş: -1 ve errno EINTR olur.

- Pause fonksiyonu sadece bir sinyal yakalayıcı çalışır ve dönüş yaparsa döner.
- Örnek33.