Sistem Programlama

Ders 14

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

Sinyaller

- Sinyaller yazılımsal işkesme (interrupt) metodudur.
  - Sinyaller kullanılarak asenkron olaylar kontrol edilebilir.
    - Örneğin bir kullanıcı belirli bir tuşa basarak işkesme yaptığında.
    - Pipiline kullanıldığında bir sonraki program hatalı şekilde sonlanırsa.
- POSIX.1 sinyal kullanımını standart haline getirmiştir.
- Sinyallerin genel özellikleri ve kullanım şekillerini inceleyeceğiz.

Sinyaller

- Her sinyalin bir ismi vardır.
  - İsimler SIG ile başlar.
    - SIGABRT ==> bir işlem abort fonksiyonunu çağırarak anormal şekilde sonlandığında oluşturulan sinyal.
    - SIGALRM ==> alarm fonksiyonu ile bir zamanlayıcı ayarlandığında ve zaman dolduğunda oluşturulan sinyal.
- FreeBSD 5.2.1, MAC OS X 10.3 ve Linux 2.4.22 31 farklı sinyali destekler. Solaris 38 farklı sinyali destekler.
- Linux ve Solaris sistemlerinde uygulama tarafından yeni sinyaller tanımlanabilir.
- Sinyal isimleri pozitif tamsayı sabitleri ile tanımlanmıştır.
  - Bu sabitler sinyal numarasıdır ve signal.h dosyasında bulunur.
  - Hiçbir sinyal 0 numrasına sahip değildir.
    - kill fonksiyonu bazı durumlarda 0 numaralı sinyal oluşturur.
    - POSIX.1 bu sinyali null sinyal olarak tanımlar.

Sinyaller

- Birçok farklı durum sinyal oluşmasına sebep olabilir.
  - Terminal tarafından oluşturulan sinyaller kullanıcı belli tuşlara bastığında oluşur. Örneğin bir program çalışırken DELETE tuşuna basmak veya CTRL-C tuşlarına basmak işkesme (interrupt) sinyali olan SIGINT oluşmasına neden olur.
    - Bu sayede devam etmekte olan bir program durdurulur.
  - Donanımla ilgili hatalar sinyal oluşturur. Örneğin 0 ile bölünme işlemi yapıldığında, geçersiz hafıza okunması gibi.
    - Bu durumlarda genellikle donanım tarafından algılanır ve kernel uyarılır. Kernel bu durumla ilgili sinyali oluşturur.
    - Örneğin geçersiz hafıza okuması yapılırsa SIGSEGV sinyali oluşturulur.

Sinyaller

- Birçok farklı durum sinyal oluşmasına sebep olabilir (devam)
  - kill (2) fonksiyonu kullanılarak başka bir işleme veya işlem grubuna sinyal gönderilebilir.
    - Bu fonksiyonun kullanılabilmesi için bazı limitler vardır.
      - Sinyal gönderdiğimiz işlemin sahini olmalıyız yada yönetici olmalıyız.
  - kill (1) komutu başka bir sişleme sinyal göndermek için kullanılır.
    - Bu komutu tahmin edeceğiniz üzere kill fonksiyonunu çağırır.
    - Çoğunlukla arka planda çalışmakta olan kaçak işlemler için kullanılır.

Sinyaller

- Birçok farklı durum sinyal oluşmasına sebep olabilir (devam)
  - Yazılımla alakalı olarak işlem ile alakalı bir olay gerçekleştiğinde işleme haber vermek için sinyal oluşturulabilir.
    - Bunlar, 0 tarafından bölünme gibi, donanım kaynaklı olaylar değil yazılımla ilgili olaylardır.
  - Örneğin ağ üzerinde bant dışı veri geldiğinde SIGURG sinyali gönderilir.
  - Örneğin pipe kullanıldığında, pipe okuyan işlem sonlandığında ve pipe için yeni veri yazıldığında SIGPIPE gönderilir.
  - Örneğin bir alarm ayarlandığında ve alarm oluştuğunda SIGALRM gönderilir.

Sinyaller

- Sinyaller asenkron olaylardır.
  - Bir işlem sinyalin ne zaman kendisine gönderileceğini bilemez.
  - Hata yönetiminde gördüğümüz gibi errno değişkeni bir değeri kontrol ederek sinyalin oluşup oluşmadığı kontrol edilemez.
    - Tek yapılacak olan kernel'e sinyal oluşması durumunda ne yapılacağının söylenmesidir.
  - Daha önce bahsettiğiniz üzere bir sinyal alındığında üç farklı işlemden biri yapılmalıdır.
    - Bu sinyal ile alakalı aksiyonlar olarak adlandırılır.

Sinyaller

- Sinyal ile ilgili aksiyonlar
  - Sinyali görmezden gelme
    - Birçok sinyal alındığında görmezden gelmek mümkündür.
    - Ancak iki fonksiyon SIGKILL ve SIGSTOP görmezden gelinemez.
      - Bu sinyaller yönetici tarafından işlemleri durdurmak veya sonlandırmak için kullanılır.
    - Donanım hatası durumunda oluşan bir sinyal görmezden gelinirse, sonuç belirsizdir.

Sinyaller

- Sinyaller ile ilgili aksiyonlar
  - Sinyali yakalama
    - Bunu yapmak için sinyal alındığında çalışacak olan fonksiyonu kernel'e bildirmemiz gerekir.
      - SIGCHLD sinyali yakalanırsa bir alt işlem sonlanmış demektir.
        - Bu durumda yapılması gereken waitpid fonksiyonu ile alt işlemin sonlanma durumunun okunmasıdır.
      - İşleminiz geçici dosyalar ürettiyse ve SIGTERM sinyali yakalanırsa (bu sinyal kill komutu tarafından işlemi sonlandırmak için gönderilir) bu geçici dosyaları temizlemek isteyebilirsiniz.
    - SIGKILL ve SIGSTOP sinyalleri yakalanamaz.

Sinyaller

- Sinyaller ile ilgili aksiyonlar
  - Varsayılan aksiyonun yapılmasına izin verilir.
    - Her işlemin bir varsayılan aksiyonu vardır.
    - Dikkat edilmelidir ki birçok sinyalin varsayılan aksiyonu işlemin sonlandırılmasıdır.
  - Varsayılan işlem “terminate+core” ise işlemin mevcut çalışma klasöründe bir hafıza resmi dosyası oluşturulur.
    - Bu dosya kullanılarak işlemin neden sonlandığı incelenebilir.
    - Bazı durumlarda bu dosya yaratılmaz.
- Kitabınızın 293 numaralı sayfasında Unix sistemlerinde oluşturulan sinyallerin bir listesi bulunmaktadır.

signal fonksiyonu

- Unix sisteminde sinyal arayüzünü kullanmak için signal fonksiyonu kullanılabilir.

```
#include <signal.h>

void (*signal(int signo, void (*func)(int)))(int);
```

Dönüş: OK ise sinyal ile ilgili önceki aksiyon, hata ise SIG_ERR.

  - signo sinyalin ismidir.
  - func değişkenin değeri aşağıdakilerden biri olabilir.
    - SIG_IGN: sistem sinyali görmezden gelir.
    - SIF_DFL: sistem varsayılan aksiyonu yapar.
    - Sinyal yakalandığında çalıştırılacak olan fonksiyonun adresi.
  - Örnek8.
  - Örnek32.

signal fonksiyonu

```
$ ./sigusr &
$ kill -USR1 7216
received SIGUSR1
$ kill -USR2 7216
received SIGUSR2
$ kill 7216
[1]+ Terminated
```