İplikler (Threads)

Giriş

- geleneksel işletim sistemlerinde her prosesin
  - özel adres uzayı ve
  - tek akış kontrolü vardır
- bazı durumlarda, aynı adres uzayında birden fazla akış kontrolü gerekebilir
  - aynı adres uzayında çalışan paralel prosesler durumunda olduğu gibi

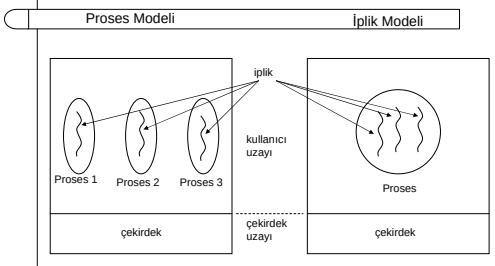
İplik Modeli

- iplik = hafif proses
- iplikler, aynı adres uzayını paylaşan ve çalışmalarını eş zamanlı yürüten proseslere benzetilebilir
- iplikler ile aynı proseste birden fazla işlem yürütme imkanı oluşur

İplik Modeli

- iplikler içinde yaratıldıkları prosesin tüm kaynaklarına erişebilir ve paylaşırlar:
  - adres uzayı, bellek, açık dosyalar, ...
- çoklu iplikli çalışma:
  - proses birden fazla ipliğe sahip
  - iplikler sıra ile yürütülürler

İplik Modeli



İplik Modeli

- iplikler prosesler gibi birbirinden bağımsız değil:
  - aynı adres uzayını paylaşırlar
    - global değişkenleri paylaşırlar
    - birbirlerinin yığınını değiştirebilir
    - koruma yok çünkü:
      - mümkün değil
      - gerek yok

İplik Modeli

- ipliklerin paylaştıkları:
  - adres uzayı
  - global değişkenler
  - açık dosyalar
  - çocuk prosesler
  - bekleyen sinyaller
  - sinyal işleyiciler
  - muhasebe bilgileri
- her bir ipliğe özel:
  - program sayacı
  - saklayıcılar
  - yığın
  - durum

İplik Modeli

- işler birbirinden büyük oranda bağımsız ise ⇒ proses modeli uygun
- işler birbirine çok bağlı ve birlikte yürütülüyorsa ⇒ iplik modeli uygun

İplik Modeli

- iplik durumları = proses durumları
  - koşuyor
  - askıda
    - bir dış olayı veya bir başka ipliği bekler (olay bekleme)
  - hazır

Yığın Kullanımı

- her ipliğin kendi yığını var
- yığında çağırılmış ama dönülmemiş yordamlarla ilgili kayıtlar ve yerel değişkenler yer alır
- her iplik farklı yordam çağrıları yapabilir
  - geri dönecekleri yerler farklı ⇒ ayrı yığın gerekli

İplikler Arası Etkileşim

- iplikler arasında
  - senkronizasyon ve
  - haberleşme olabilir

İpliklerin Gerçeklenmesindeki Sorunlar

- örn. UNIX'te `fork` sistem çağrısında
  - anne çok iplikli ise çocuk proseste de aynı iplikler bulunacak mı?
    - HAYIR ise program doğru çalışmayabilir
    - EVET ise,
      - örneğin annedeki iplik giriş bekliyorsa çocuktaki de mi beklesin?
      - giriş bilgisi hazır olunca her ikisine de mi yollansın?
  - benzer problem açık olan ağ bağlantıları için de var

İpliklerin Gerçeklenmesindeki Sorunlar

- bir iplik bir dosyayı kullanırken, bir diğer iplik dosyayı kapatırsa ne olur?
- bir iplik yetersiz bellek olduğunu farkedip bellek isteğinde bulunursa ne olur?
  - işlem tamamlanmadan bir başka iplik çalışır ve yeni iplik de belleğin yetersiz olduğunu farkedip istekte bulunursa ⇒ iki kere bellek alınabilir
- çözümler için iyi tasarım ve planlama gerekli

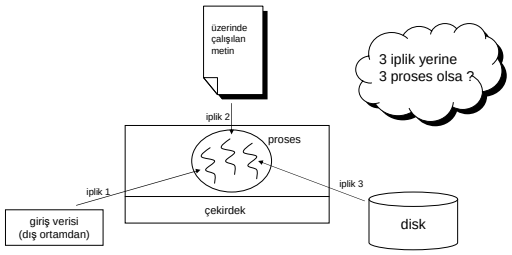
İplik Kullanımının Yararları

- bir proses birlikte yürütülebilecek olan birden fazla işlem içerebilir
  - işlemlerden bazıları bloke olursa diğerleri çalışabilir → ipliklere bölmek performansı arttırır
- ipliklerin kendilerine ait kaynakları yoktur → yaratılmaları / yokedilmeleri proseslere göre kolay ve hızlı

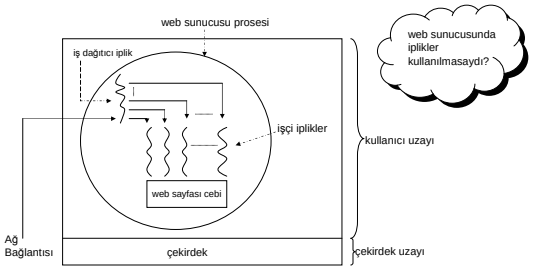
İplik Kullanımının Yararları

- ipliklerin bazıları işlemciye yönelik işlemler, bazıları giriş-çıkış işlemleri yapıyorsa performans artar
  - hepsi işlemciyi yoğun olarak kullanıyorsa performans artışı gözlenemez
- çok işlemcili sistemlere uygun → farklı işlemcilere farklı iplikler atanabilir (paralel çalışma)

İplik Kullanımına Örnek – 3 İplikli Kelime İşlemci Modeli



İplik Kullanımına Örnek – Web Sitesi Sunucusu



İplik Kullanımına Örnek – Web Sitesi Sunucusu

İş dağıtıcı iplik kodu

```
while TRUE {
    sıradaki_isteği_al(&tmp);
    işi_aktar(&tmp);
}
```

İşçi ipliklerin kodu

```
while TRUE {
    iş_bekle(&tmp);
    sayfayı_cepte_ara(&tmp,&sayfa);
    if (sayfa_cepte_yok(&sayfa)
         sayfayı_diskten_oku(&tmp,&sayfa);
    sayfayı_döndür(&sayfa);
}
```