

Bilgisayar İşletim Sistemleri BLG 312

İplikler

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

Prosesler

- İşletim sistemi, prosesler aracılığıyla, G/Ç işlemleri ile hesaplama işlemlerini birarada yürüterek sistem etkinliğini artırır
- **Prosesler sisteme ek yük getirirler:**
 - Proses yaratma
 - Bağlam saklama/yükleme
 - Proses seçme, değiştirme
 - Bu işlemlerin tümü çekirdeğin etkin olmasını gerektirir

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

Prosesler

- 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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Modeli

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

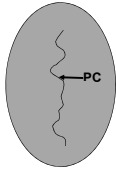
BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

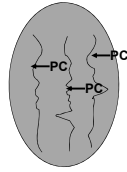
Doç. Dr. A. Şima Etaner-Uyar

İplikler

Proses kavramını genişlet:



Geleneksel proses



Çok iplikli proses

- Bir iplik kümesi aynı adres uzayını paylaşırlar

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Modeli

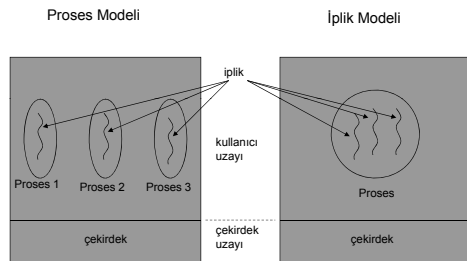
- iplikler içinde yaratıldıkları prosesin tüm kaynaklarına erişebilir ve paylaşırlar:
 - adres uzayı, bellek, açık dosyalar, ...
- çoklu iplikli çalışma:
 - proses birden fazla ipliğe sahip
 - iplikler sıra ile yürütülürler
 - Bağlam değiştirme daha düşük maliyetlidir
 - Bir iplik bloke olursa, bir diğeri devam eder

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Modeli



BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Modeli

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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Modeli

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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Modeli

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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

Yığın Kullanımı

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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Yaratılması

- prosesin başta tek ipliği var
- iplikler kütüphane yordamları ile yeni iplikler yaratılır
 - örn: `thread_create`
 - parametresi: koşturacağı yordamın adı
- yaratılan yeni iplik aynı adres uzayında koşar
- bazı sistemlerde iplikler arası anne – çocuk hiyerarşisi yer alır
 - çoğu sistemde tüm iplikler eşit

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Yokedilmesi

- işi biten iplikler kütüphane yordamı çağırısı ile sonlandırılır
 - örn: `thread_exit`
- zaman paylaşımı için zamanlayıcı yok ⇒ iplikler işlemciyi kendileri bırakır
 - örn: `thread_exit`

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplikler Arası Etkileşim

- iplikler arasında
 - senkronizasyon ve
 - haberleşme olabilir

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Gerçeklenmesindeki Sorunlar

- örn. UNIX'te `fork` sistem çağırısında
 - 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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Kullanımının Yararları

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

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Kullanımının Yararları

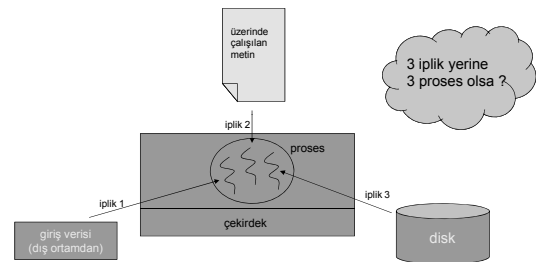
- ipliklerin bazıları işlemciye yönelik işlemler, bazıları giriş-çıkış işlemleri yapıyorsa performans artar
 - hepsi işlemciyi yoğun olarak kullanıyorsa performans artışı gözlenemez
- çok işlemcili sistemlere uygun → farklı işlemcilere farklı iplikler atanabilir (paralel çalışma)

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Kullanımına Örnek – 3 İplikli Kelime İşlemci Modeli

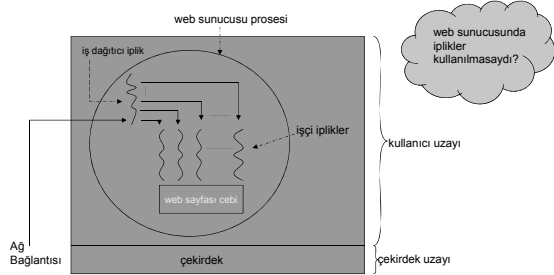


BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İplik Kullanımına Örnek – Web Sitesi Sunucusu



BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

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_cepce_ara(&tmp, &sayfa);
    if (sayfa_cepce_yok(&sayfa))
        sayfayı_diskten_oku(&tmp, &sayfa);
    sayfayı_dondür(&sayfa);
}
```

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Gerçeklenmesi

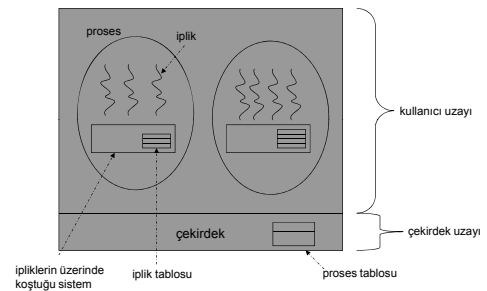
- iki türlü gerçekleştirilebilir
 - kullanıcı uzayında
 - çekirdek uzayında
- hibrid bir gerçekleştirilebilir

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesi



BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesi

- çekirdek ipliklerden haberdar değildir
- çoklu iplik yapısını desteklemeyen işletim sistemlerinde de gerçekleştirilebilir
- ipliklerin üzerinde koştuğu sistem uygun bir çalışma ortamı sunar
 - iplik yönetim yordamları
 - thread_create, thread_exit, thread_yield, thread_wait, ...
 - iplik tablosu
 - program sayacı, saklayıcılar, yığın işaretçisi, durum bilgisi, ...

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesi

- iplik askıya alınmasına neden olacak bir işlem yürütürse (örneğin bir başka ipliğin bir işi bitirmesini beklemek gibi..) çağrılan iplik yönetim yordamının yürüttüğü işlemler:
 - ipliğin durumunu "askıda" olarak değiştirir
 - ipliğin program sayacı ve saklayıcı içeriklerini iplik tablosuna saklar
 - sıradaki ipliğin bilgilerini tablodan alıp saklayıcılara yükler
 - sıradaki ipliği çalıştırır

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesinin Avantajları

- ipliklere ait ayrı bir iş sıralama algoritması bulunabilir
- çekirdekte iplik tablosu için alan ayırmaya gerek kalmaz
- tüm çağrılar yerel yordamlar \Rightarrow çekirdeğe çağrı (*sistem çağırısı*) yapmaktan daha hızlı ve maliyet düşük

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesindeki Problemler

- askıya alınmayla sonuçlanacak sistem çağrıları tüm iplikleri bloke eder
 - iplik doğrudan bu tür bir sistem çağırısı yürütemez çünkü işlem tüm ipliklerin askıya alınmalarına neden olur
 - çekirdek ipliği değil, onu içeren prosesi askıya alacaktır

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesinde Problemler

- **çözüm 1:** sistem çağrıları değiştirilebilir ancak
 - işletim sisteminin değiştirilmesi istenmez
 - kullanıcı programlarının da değişmesi gerekir
- **çözüm 2:** bazı sistemlerde, yapılan çağrının askıya alınmaya neden olup olmayacağı bilgisini döndüren sistem çağrıları var
 - sistem çağrılarına ara-birim (wrapper) yazılır
 - önce kontrol edilir, askıya alınma söz konusu olacaksa sistem çağrısı yapılmaz ve iplik bekletilir

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesinde Problemler

- **sayfa hataları**
 - programın yürütülmesi gereken kod parçasına ilişkin kısmı ana bellekte yoksa
 - sayfa hatası olur
 - proses bloke olur
 - gereken sayfa ana belleğe alınır
 - proses çalışabilir
- **sayfa hatasına iplik sebep olduysa**
 - çekirdek ipliklerden habersiz olduğundan tüm proses bloke edilir

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Kullanıcı Uzayında Gerçeklenmesinde Problemler

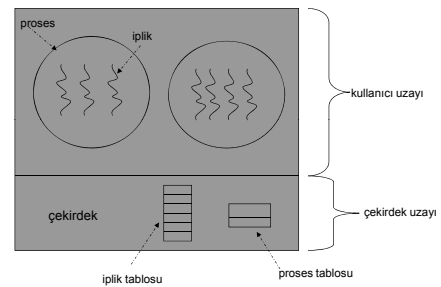
- **iş sıralama**
 - iplik kendisi çalışmayı bırakmazsa diğer iplikler çalışamaz
 - altta çalışan sistem belirli sıklıkta saat kesmesi isteyebilir
 - ipliklerin de saat kesmesi ile işi varsa karışıklık olabilir

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Çekirdek Uzayında Gerçeklenmesi



BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Çekirdek Uzayında Gerçeklenmesi

- çekirdek ipliklerden haberdardır
- iplik tablosu çekirdekte yer alır
- yeni iplik yaratmak için bir çekirdek sistem çağrısı yürütülür

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Çekirdek Uzayında Gerçeklenmesi

- ipliğin askıya alınmasına neden olabilecek tüm çağrılar çekirdek sistem çağrılarıdır
- işletim sistemi hangi ipliğin koşacağına karar verir
 - seçilen iplik aynı prosese ait olmayabilir

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Çekirdek Uzayında Gerçeklenmesi

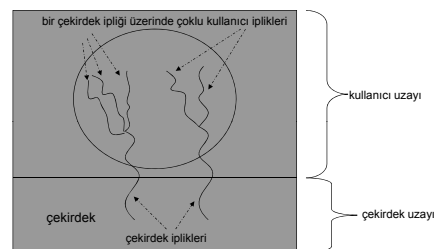
- ipliğin askıya alınmasına neden olabilecek sistem çağrılarının yeniden yazılması gerekmez
- sayfa hatası durumu da sorun yaratmaz
 - sayfa hatası olunca çekirdek aynı prosenin koşabilecek bir başka ipliği varsa, onu çalıştırır
- sistem çağrısı gerçekleştirme ve yürütme maliyetli
 - çok sık iplik yaratma, yoketme, ... işlemleri varsa vakit kaybı çok

BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Hibrit Yapıda Gerçeklenmesi



BLG 312 – Bilgisayar İşletim Sistemleri ©2006

Prof. Dr. Nadia Erdoğan

Doç. Dr. A. Şima Etaner-Uyar

İpliklerin Hibrit Yapıda Gerçeklenmesi

- çekirdek sadece çekirdek düzeyi ipliklerden haberdar olur
- bir çekirdek düzeyi iplik üzerinde birden fazla kullanıcı düzeyi iplik sıra ile çalışır
- kullanıcı düzeyi iplik işlemlerinde sözü edilen yarar ve sorunlar yine geçerlidir