

## Bölüm 3: Prosesler



BİL 304 İşletim Sistemleri

3.1

Doç.Dr. Ahmet Zengin

## Bölüm 3: Prosesler

- Proses Kavramları
- Proses Planlama
- Proses Üzerinde İşlemler
- Prosesler Arası İletişim
- IPC (Prosesler Arası İletişim) Örnekleri
- İstemci-Sunucu Sistemleri Arasındaki İletişim

BİL 304 İşletim Sistemleri

3.2

Doç.Dr. Ahmet Zengin

## Bölüm Hedefleri

- Proses mantığını tanıtmak – çalışan bir program
- Proseslerin yapısını anlatmak (Proseslerin zamanlanması, oluşturulması ve sonlandırılması, iletişimi vb.)
- İstemci – sunucu sistemlerin iletişimini tarif etmek

BİL 304 İşletim Sistemleri

3.3

Doç.Dr. Ahmet Zengin

## Proses Kavramları

- Bir işletim sistemi çeşitli programlar yürütür:
  - Batch sistemi – görevler (jobs)
  - Zaman paylaşımli sistemler – kullanıcı programları yada görevleri.
- Ders kitaplarında görev/süreç (job / proses) terimleri neredeyse birbirlerinin yerine kullanılır.
- Proses, yürütülen bir programdır denilebilir ve prosesler sıralı bir biçimde yürütülmelidir.
- Bir proses aşağıdakileri alanları içerir:
  - Program Sayacı
  - Yığın (stack)
  - Veri bölümü (data section)

BİL 304 İşletim Sistemleri

3.4

Doç.Dr. Ahmet Zengin

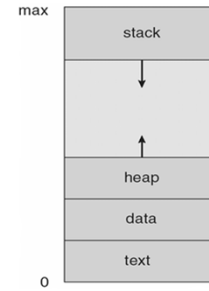


## Proses

- Bölümleri,
  - **program kodu** (text section),
  - **Mevcut işlem**, program sayacı ve diğer kaydediciler tarafından tutulur.
  - **Yığın** (stack) geçici veriyi içerir.
    - Fonksiyon parametreleri, geri dönüş değerleri, yerel değişkenler
  - **Veri bölümü** (data section) global değişkenleri içerir.
  - **Bellek yığını** (heap) çalışma zamanında ayrılmış bilgileri içerir.
- Program pasif, Proses ise aktiftir
  - Program çalıştırılabilir halde belleğe yüklendiğinde proses halini alır.
- Programın çalıştırılması komut satırından komutun girilmesi, grafik arayüzde program ikonu üzerine tıklanması vb. ile başlar.
- Bir program birden fazla proses içerebilir.
  - Örneğin birden fazla kullanıcının aynı programı çalıştırması.



## Bellekteki Bir Proses

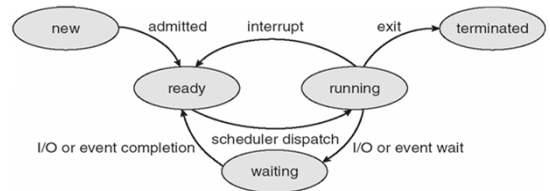


## Proses Durumları

- Çalışma anında proseslerin durumları değişir.
  - **yeni**: Yeni bir proses oluşturuluyor.
  - **çalışıyor**: İşlemler gerçekleştiriliyor.
  - **bekleme**: Proses bazı olayların gerçekleşmesini beklemektedir.
  - **hazır**: Proses, işlemciye aktarılmayı beklemektedir.
  - **sonlandırılmış**: Prosesin yürütülmesi tamamlandığını belirtir.



## Proses Durum Diyagramı





## Proses Kontrol Bloğu (PCB)

Her proses aşağıdaki bilgileri içerir :

- Proses durumu
- Program Sayacı
- CPU kayıtları
- CPU planlama bilgisi
- Bellek yönetim bilgisi
- Hesaplama bilgisi
- I/O durum bilgisi



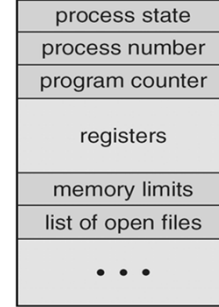
BİL 304 İşletim Sistemleri

3.9

Doç.Dr. Ahmet Zengin



## Proses Kontrol Bloğu (PCB)



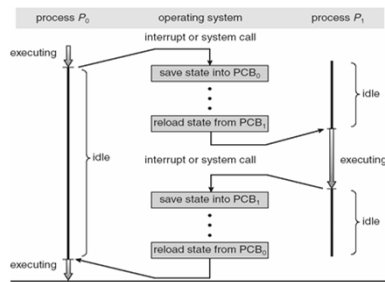
BİL 304 İşletim Sistemleri

3.10

Doç.Dr. Ahmet Zengin



## CPU'da Proses Değişimi



BİL 304 İşletim Sistemleri

3.11

Doç.Dr. Ahmet Zengin



## Proses Planlama

- İşlemcinin maksimum kullanımı adına, prosesler arası geçiş ve zaman paylaşımı çok hızlı gerçekleşir.
- **Proses planlayıcı**, işlenecek prosesi işleme hazır prosesler arasından seçer.
- Prosesler aşağıdaki **sıralama kuyruklarında** tutulur.
  - **İş kuyruğu**– sistemdeki tüm prosesler kümesi
  - **Hazır kuyruğu**– ana bellekte bulunan, hazır ve işleme girmeyi bekleyen prosesler kümesi
  - **Aygıt kuyrukları**– I/O aygıtlarından gelecek mesajı bekleyen prosesler kümesi
  - Prosesler kuyrukları arasında geçiş yapılabilir.



BİL 304 İşletim Sistemleri

3.12

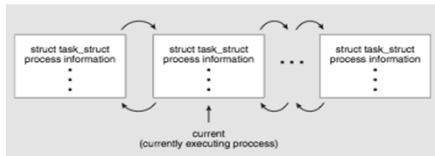
Doç.Dr. Ahmet Zengin



## Proseslerin Linux'ta Temsili

- Represented by the C structure `task_struct`  

```
pid_t pid; /* Proses tanımlayıcı */
long state; /* Prosesin durumu */
unsigned int time_slice; /* planlama bilgisi */
struct task_struct *parent; /* bu ebeveyn Proses */
struct list_head children; /* bu çocuk Proses */
struct files_struct *files; /* açık dosyaların listesi */
struct mm_struct *mm; /* address space of this process */
```



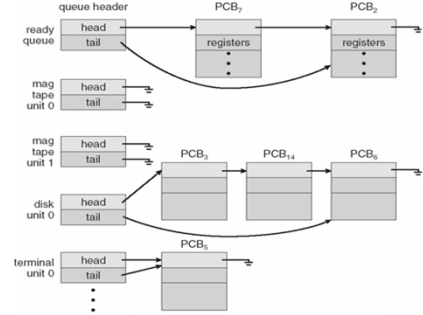
BİL 304 İşletim Sistemleri

3.13

Doç.Dr. Ahmet Zengin



## Hazır Kuyruğu ve Çeşitli I/O Aygıt Kuyrukları



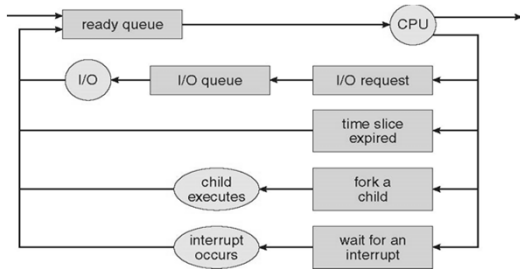
BİL 304 İşletim Sistemleri

3.14

Doç.Dr. Ahmet Zengin



## Proses Sıralama Diyagramı



BİL 304 İşletim Sistemleri

3.15

Doç.Dr. Ahmet Zengin



## Sıralayıcılar

- Uzun vadeli sıralayıcılar / iş planlayıcı** – Hazır kuyruğundan getirilen prosesler arasından seçim yapar.
- Kısa vadeli sıralayıcılar / CPU planlayıcı** – Prosesin bir sonraki işlemini ve işlem için ayrılan alanın tahsisini seçer.
  - Bazen sistemde sadece bir sıralayıcı vardır



BİL 304 İşletim Sistemleri

3.16

Doç.Dr. Ahmet Zengin



## Sıralayıcılar(Devam)

- Kısa vadeli sıralayıcılar çok sık çağırılırlar.
  - (milisaniyeler düzeyinde)  $\Rightarrow$  hızlı olmalı
- Uzun vadeli sıralayıcılar çok seyrek çağırılırlar
  - (saniyeler, dakikalar düzeyinde)  $\Rightarrow$  yavaş olabilir
- Uzun vadeli sıralayıcılar, *çoklu programlama* derecesini kontrol eder.
- Prosesler her iki biçimde de tanımlanmış olabilir:
  - **I/O-bağımlı Proses** –giriş/çıkış hesaplaması için daha fazla zaman harcar, çok az CPU sarfıyatı
  - **CPU-bağımlı Proses** – Hesaplamalar için zaman harcar, çok fazla CPU sarfıyatı



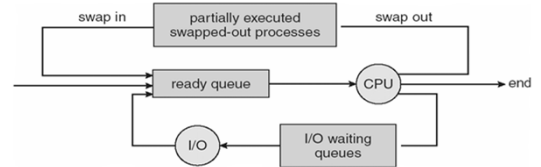
BİL 304 İşletim Sistemleri

3.17

Doç.Dr. Ahmet Zengin



## Orta vadeli sıralayıcıların eklenmesi



BİL 304 İşletim Sistemleri

3.18

Doç.Dr. Ahmet Zengin



## Bağlam Değiştirme (Context Switch)

- CPU diğer prosese geçtiği zaman, sistem mutlaka eski prosesin durumunu kaydetmeli ve yeni prosesin daha önce kaydedilmiş durumunu yüklemeli
- **Bağlam PCB'dir.**
- Aşırı sayıda bağlam değiştirme; sistem geçişler sırasında kullanışlı olmaz
  - Daha karmaşık OS ve PCB  $\rightarrow$  daha uzun bağlam değişimi
- Donanım desteği zamana bağlıdır.
  - Bazı donanımlar CPU başına birden fazla kaydedici sağlar  $\rightarrow$  birden fazla bağlam bir kerede yüklenir



BİL 304 İşletim Sistemleri

3.19

Doç.Dr. Ahmet Zengin



## Proses Oluşturulması

- Ebeveyn Proses, çocuk prosesleri oluşturur. Bu şekilde ağaç yapısı meydana gelir.
- Genelde prosesler, **bir proses kimlik numarası** (Proses identifier - pid) ile tanımlanır ve yönetilir.
- Kaynak Paylaşımı türleri:
  - Ebeveyn ve çocuk prosesler tüm kaynakları paylaşır.
  - Çocuk prosesler ebeveyn prosesin kaynaklarını kullanır.
  - Ebeveyn ve çocuk hiçbir kaynağı paylaşmaz.
- Uygulama:
  - Ebeveyn ve çocuk proses eşzamanlı çalışır.
  - Ebeveyn proses, çocuk proses sonlanana kadar bekler.



BİL 304 İşletim Sistemleri

3.20

Doç.Dr. Ahmet Zengin



## Proses Oluşturulması (Devam)

- Adres alanı
  - Çocuk proses, ebeveyn prosesin alanını kopyalar .
  - Çocuk prosese bir program yüklenmiş olur.
- UNIX örnekleri :
  - **fork** işlevi yeni proses oluşturulması için sisteme çağrıda bulunur.
  - **exec** sistem çağrısı Prosesin bellek alanını yeni bir program ile değiştirmek için bir fork sonrası çağrılır.



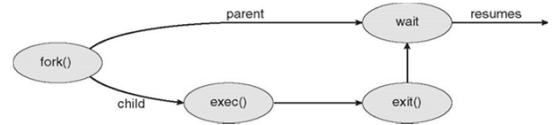
BİL 304 İşletim Sistemleri

3.21

Doç.Dr. Ahmet Zengin



## Proses Oluşturulması



BİL 304 İşletim Sistemleri

3.22

Doç.Dr. Ahmet Zengin



## Fork İşlemi Yapan C Programı

```

#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
int main()
{
    pid_t pid;
    /* fork another Proses */
    pid = fork();
    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child Proses */
        execlp("/bin/ls", "ls", NULL);
    }
    else { /* parent Proses */
        /* parent will wait for the child */
        wait (NULL);
        printf ("Child Complete");
    }
    return 0;
}
  
```



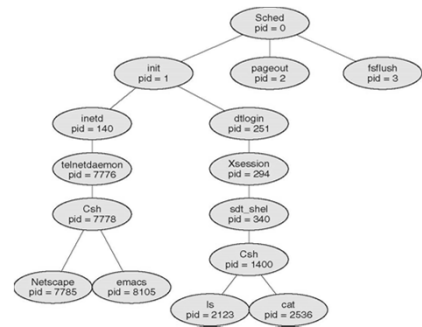
BİL 304 İşletim Sistemleri

3.23

Doç.Dr. Ahmet Zengin



## Solaris'te Proses Ağaç Yapısı



BİL 304 İşletim Sistemleri

3.24

Doç.Dr. Ahmet Zengin



## Proses'lerin Sonlanması

- Proses son kod ifadesinin çalıştırır ve işletim sistemine silinip silinmeyeceğini sorar (çıkış)
  - Çıkış verisi çocuktan ebeveyn procese (**bekleyerek**)
  - Prosese ayrılan alan işletim sistemi tarafından sonlandırılır.
- Ebeveyn Proses çocuk Prosesin çalışmasını sonlandırabilir (**durdurmak**)
  - Çocuk Proses tahsis edilmiş kaynakların dışına çıkarılır.
  - Artık çocuk Proses görev tayin etmek gerekmez.
  - Eğer ebeveyn Proses sonlandırılırsa
    - › Bazı işletim sistemi ebeveyn proses sonlandırıldıktan sonra çocuk prosesin çalışmasına izin vermez
    - › Tüm çocuk Prosesler sonlandırılır. - **basamaklı sonlandırma**



BİL 304 İşletim Sistemleri

3.25

Doç.Dr. Ahmet Zengin



## Prosesler Arası İletişim

- Prosesler işletim sistemi içerisinde bağımsız ya da işbirliği içinde çalışabilirler.
- İşbirliği içerisindeki prosesler veri paylaşımı da dahil olmak üzere diğer prosesleri etkileyebilir ya da diğer proseslerden etkilenebilirler.
- Proseslerin işbirliği yapma nedenleri:
  - Bilgi paylaşımı
  - Daha hızlı hesaplama
  - Modülerlik
  - Rahatlık
- İşbirliği içindeki prosesler prosesler arası haberleşmeye (Interproses communication - IPC) ihtiyaç duyarlar.
- 2 temel IPC modeli mevcuttur:
  - Paylaşılmış bellek
  - Mesajlaşma



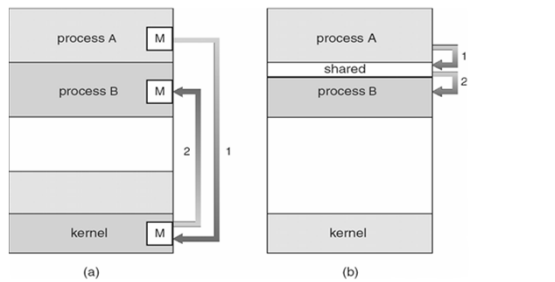
BİL 304 İşletim Sistemleri

3.26

Doç.Dr. Ahmet Zengin



## Haberleşme Modelleri



BİL 304 İşletim Sistemleri

3.27

Doç.Dr. Ahmet Zengin



## İşbirliği içerisindeki Prosesler

- **Bağımsız** prosesler, diğer proseslerin çalışmasından etkilenebilir ve diğer prosesleri etkileyemezler.
- **İşbirliği yapan** prosesler, diğer proseslerin çalışmasından etkilenebilir ve diğer prosesleri etkileyebilirler.
- Prosesler arası işbirliğinin avantajları :
  - Bilgi paylaşımı
  - Daha hızlı hesaplama
  - Modülerlik
  - Rahatlık



BİL 304 İşletim Sistemleri

3.28

Doç.Dr. Ahmet Zengin



## Üretici-Tüketici Problemi

- İşbirliği içindeki proseslere ilişkin bir paradigma: üretici proses tüketici proses tarafından kullanılmak üzere bilgi üretir.
  - *Sınırlandırılmamış tampon*: tampon için limit konulmamıştır
  - *Sınırlandırılmış tampon*: sabit bir tampon boyutu mevcuttur.



BİL 304 İşletim Sistemleri

3.29

Doç.Dr. Ahmet Zengin



## Sınırlı-Tamponlu- Paylaşımlı-Bellek Çözümü

- Paylaşılmış veri

```
#define BUFFER_SIZE 10
typedef struct {
    ...
} item;

item buffer[BUFFER_SIZE];
int in = 0;
int out = 0;
```
- Çözüm doğru, ancak sadece BUFFER\_SIZE-1 eleman kullanılabilir.



BİL 304 İşletim Sistemleri

3.30

Doç.Dr. Ahmet Zengin



## Sınırlı-Tampon – Üretici

```
while (true) {
    /* data üretilir */
    while (((in == (in + 1) % BUFFER_SIZE count) == out)
        ; /* çalışmaz – serbest tampon yok */
    buffer[in] = item;
    in = (in + 1) % BUFFER_SIZE;
}
```



BİL 304 İşletim Sistemleri

3.31

Doç.Dr. Ahmet Zengin



## Sınırlı Tampon – Tüketici

```
while (true) {
    while (in == out)
        ; // çalışmaz -- nothing to consume

    // buffer tarafından veri silinir
    item = buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    return item;
}
```



BİL 304 İşletim Sistemleri

3.32

Doç.Dr. Ahmet Zengin





## Prosesler Arası İletişim - Mesajlaşma

- Proseslerin iletişim mekanizması diğer proseslerin işlemleriyle senkronizedir. Mesaj sistemi – prosesler birbiri ile, paylaşılan değişkenleri kullanmadan iletişim kurar.
- IPC iki işlemi destekler :
  - **send(message)** – gönderilecek mesaj boyutu, sabit ya da değişken olabilir.
  - **receive(message)**
- Eğer *P* ve *Q* prosesleri iletişim kurmak istiyorsa, şu işlemleri yapmaları gerekir :
  - Aralarında iletişim bağlantısı var olmalıdır.
  - send /receive yardımı ile mesaj alışverişi gerçekleştirmelidirler.
- İletişim bağlantısı oluşturulması
  - fiziksel (ör., paylaşılmış bellek, donanım veriyolu)
  - mantıksal (ör., mantıksal özellikler)



BİL 304 İletişim Sistemleri

3.33

Doç.Dr. Ahmet Zengin



## Doğrudan İletişim

- Proseslerin her biri gönderici ve alıcı olarak isimlendirilmelidir :
  - **send (P, message)** – P prosesine mesaj gönder
  - **receive(Q, message)** – Q prosesinden mesaj al
- İletişim bağlantısının özellikleri :
  - Bağlantılar otomatik olarak kurulur.
  - Her bir proses çifti arasında tam olarak bir bağlantı vardır.
  - Bir link 2 proses ile ilişkilendirilebilir.
  - Bağlantı tek yönlü olabilir, ancak genellikle iki yönlüdür.



BİL 304 İletişim Sistemleri

3.34

Doç.Dr. Ahmet Zengin



## Doğrudan Olmayan İletişim

- Mesajlar port veya posta kutularından alınır veya buralara gönderilir.
  - Her posta kutusu tek bir tanımlayıcıya sahiptir
  - Prosesler paylaşılmış bir posta kutusuna sahipse iletişim kurabilirler.
- İletişim bağlantısı özellikleri şunlardır :
  - Bağlantı, prosesler arası paylaşılmış bir posta kutusu var ise kurulur.
  - Bir bağlantı ikiden fazla proses ile ilişkilendirilebilir.
  - Her bir proses çifti birden fazla bağlantıya sahip olabilir.
  - Bağlantı tek yönlü ya da çift yönlü olabilir.



BİL 304 İletişim Sistemleri

3.35

Doç.Dr. Ahmet Zengin



## Doğrudan Olmayan İletişim

- İşlemler
  - Yeni bir posta kutusu oluştur,
  - Posta kutusu aracılığıyla mesaj gönder ve al.
  - posta kutusunu yok et.
- İletişim basitçe şu şekilde gerçekleşir:
  - **send(A, message)** – A'nın posta kutusuna bir mesaj gönder
  - **receive(A, message)** – A'nın posta kutusundan bir mesaj al.



BİL 304 İletişim Sistemleri

3.36

Doç.Dr. Ahmet Zengin



## Doğrudan Olmayan İletişim

- Posta kutusu paylaşımı
  - $P_1$ ,  $P_2$  and  $P_3$  Prosesleri A posta kutusunu paylaşıyor.
  - $P_1$  mesaj gönderiyor;  $P_2$  ve  $P_3$  mesajı alıyor.
  - Mesajı hangisi almıştır?
- Çözüm:
  - Bir bağlantının en fazla iki proses ile ilişkilendirilmesine izin verir.
  - Bir seferde yalnızca bir proses yürütmesine izin verir.
  - Sistemin rastgele bir alıcı seçimine izin verir. Gönderici, alıcının kim olduğunu bildirir.



BİL 304 İletim Sistemleri

3.37

Doç.Dr. Ahmet Zengin



## Senkronizasyon

- Mesaj iletimi engelli ya da engelsiz olabilir.
- **Engelli**, senkron iletim olarak düşünülebilir.
  - **Engelli** gönderim, mesaj alınana kadar gönderici engellenir.
  - **Engelli** alım, mesaj hazır olana kadar alıcı engellenir.
- **Engelsiz**, asenkron iletim olarak düşünülebilir.
  - **Engelsiz** gönderim, mesaj yollar ve devam edilir.
  - **Engelsiz** alım, hazır mesaj varsa alır yoksa boş-null değer alır.



BİL 304 İletim Sistemleri

3.38

Doç.Dr. Ahmet Zengin



## Tamponlama

- Bağlantıyla ilişkilendirilmiş mesaj sırası, şu 3 yolla düzenlenir:
  1. Sıfır kapasite – 0 mesaj  
Gönderici, alıcıyı beklemelidir (Buluşma).
  2. Sınırlı kapasite –n adet mesajın sonlu bir uzunluğa sahip olması  
Gönderici, bağlantı dolu ise beklemelidir.
  3. Sınırsız kapasite – sonsuz uzunluk  
Gönderici hiçbir zaman beklemez.



BİL 304 İletim Sistemleri

3.39

Doç.Dr. Ahmet Zengin



## IPC Sistem Örnekleri - POSIX

- POSIX Paylaşılmış bellek
  - Proses öncelikle paylaşılmış bellek alanı oluşturur.

```
segment id = shmget(IPC_PRIVATE, size, S_IRUSR | S_IWUSR);
```

  - Proses paylaşılmış belleğe erişmek istemektedir.

```
shared memory = (char *) shmat(id, NULL, 0);
```

  - Şimdi, proses paylaşılan belleğe yazabilir.

```
sprintf(shared memory, "Writing to shared memory");
```

  - İşlem tamamlandığında önceden ayrılan bellek alanı geri alınabilir.

```
shmdt(shared memory);
```



BİL 304 İletim Sistemleri

3.40

Doç.Dr. Ahmet Zengin



## IPC Sistem Örnekleri - Mach

- Mach iletişimi mesaj tabanlıdır.
  - Hatta sistem çağrıları dahi birer mesajdır.
  - Her görev iki posta kutusu oluşturur – Kernel ve Notify
  - Sadece 3 sistem çağrısı mesaj transferine ihtiyaç duyar.  
`msg_send()`, `msg_receive()`, `msg_rpc()`
  - Mailboxların oluşturulma sebebi iletişimde mailboxlara ihtiyaç duyulmasıdır.  
`port_allocate()`



BİL 304 İşletim Sistemleri

3.41

Doç.Dr. Ahmet Zengin



## IPC Sistem Örnekleri – Windows XP

- Mesaj iletimi yerel prosedür çağrı (**local procedure call - LPC**) birimi aracılığıyla yönetilir
  - Yalnızca prosesler ve benzeri sistemler arasında çalışır.
  - İletişim kanalları kurmak ve sürdürmek için portları (posta kutuları gibi) kullanır.
  - Haberleşme aşağıdaki gibi çalışır:
    - › İstemci altsistemi bir bağlantı nesnesi açar.
    - › İstemci bağlantı isteği gönderir.
    - › Sunucu iki özel iletişim portu oluşturur ve bunlardan birini istemciye gönderir.
    - › İstemci ve sunucu mesajları göndermek, almak ve cevapları dinlemek amacıyla karşılıklı portlar kullanır.



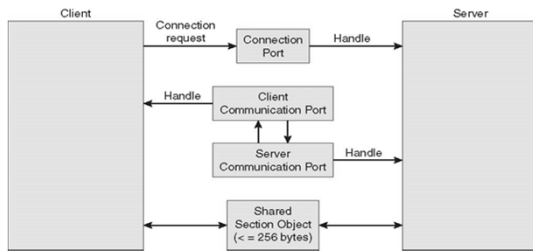
BİL 304 İşletim Sistemleri

3.42

Doç.Dr. Ahmet Zengin



## Windows XP'de Yerel İşlem Çağrısı



BİL 304 İşletim Sistemleri

3.43

Doç.Dr. Ahmet Zengin



## İstemci – Sunucu Sistemlerinde İletişim

- Soketler
- Uzaktan Prosedür Çağrıları
- Tüneller- pipes
- Uzaktan Metot Çağrıları (RMI - Java)



BİL 304 İşletim Sistemleri

3.44

Doç.Dr. Ahmet Zengin



## Soketler

- Soket, bir iletişimin bitiş noktası olarak tanımlanabilir.
- IP adresinin ve portun birleşimidir.
- **161.25.19.8:1625** soketi, **1625** portu ve **161.25.19.8** sunucusu demektir.
- İletişim, bir çift soket arasında meydana gelir.



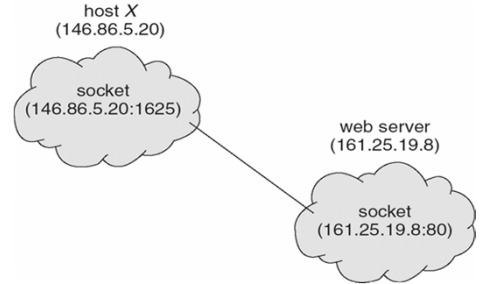
BİL 304 İşletim Sistemleri

3.45

Doç.Dr. Ahmet Zengin



## Soket İletişimi



BİL 304 İşletim Sistemleri

3.46

Doç.Dr. Ahmet Zengin



## Uzak Yordam Çağrısı

- Uzak prosedür çağrısı (Remote procedure call - RPC), yordam çağrılarını bağlı sistemler üzerindeki işlemlere ayırır.
- **Stub** – sunucu üzerindeki gerçek prosedür için istemci tarafındaki aracı
- İstemci tarafındaki stub, sunucunun yerini belirler ve parametreleri yönlendirir.
- Sunucu tarafındaki stub, mesajı alır, yönlendirilmiş parametreleri açar ve yordamı sunucu üzerinde uygular.



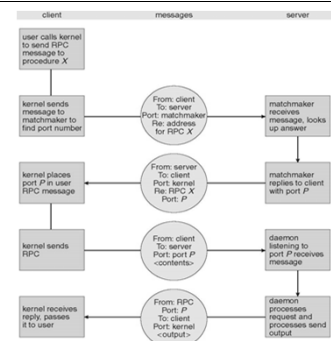
BİL 304 İşletim Sistemleri

3.47

Doç.Dr. Ahmet Zengin



## RPC'nin Çalışma Prensibi



BİL 304 İşletim Sistemleri

3.48

Doç.Dr. Ahmet Zengin



## Tüneller- Pipes

- İki proses arasında iletişime izin veren yapıdır.
- Sorunlar:
  - İletişim tek yönlü mü, çift yönlü müdür?
  - İletişim iki yönlü ise yarı dubleks mi çalışır, yoksa tam dubleks mi çalışır?
  - İletişim halindeki prosesler arasında bir ilişki (ebeveyn-çocuk) olmalı mıdır?
  - Tüneller ağ üzerinden kullanılabilir mi?



BİL 304 İşletim Sistemleri

3.49

Doç.Dr. Ahmet Zengin



## Sıradan Tüneller

- Sıradan tüneller, standart üretici-tüketici tipi iletişime izin verir.
- Üretici bir uçtan yazar (tünelin yazma ucu)
- Tüketici diğer ucundan okur (tünelin okuma ucu)
- Sıradan tüneller bu nedenle tek yönlü iletişim sağlar.
- Haberleşen prosesler arasında ebeveyn-çocuk ilişkisi gerekir.



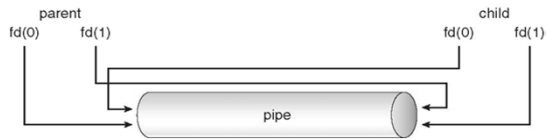
BİL 304 İşletim Sistemleri

3.50

Doç.Dr. Ahmet Zengin



## Sıradan Tüneller



BİL 304 İşletim Sistemleri

3.51

Doç.Dr. Ahmet Zengin



## Adlandırılmış Tüneller

- Adlandırılmış tüneller, sıradan olanlardan daha güçlüdür.
- İletişim çift yönlüdür.
- Haberleşen prosesler arasında ebeveyn-çocuk ilişkisi gerekli değildir.
- Birden fazla proses, kullanabilir.
- UNIX ve Windows işletim sistemlerince desteklenir.



BİL 304 İşletim Sistemleri

3.52

Doç.Dr. Ahmet Zengin

## Bölüm 3 - Son

