



# **Bölüm 11: Çok İşlemcili Sistemler**

## **İşletim Sistemleri**



# Çok İşlemcili Sistemler

- Bellek, iletişim ve G/Ç kaynaklarını paylaşan çok işlemcili bir bilgisayar sistemi.
- Çok İşlemcili Sistem Türleri:
  - **Simetrik Çoklu İşlemci (SMP):** Her işlemci, paylaşılan kaynaklara eşit erişime sahiptir.
  - **Asimetrik Çoklu İşlemci (AMP):** Bir işlemci ana işlemci, diğerleri yardımcı işlemci görevi görür.
  - **Küme Sistemleri:** Tek bir sistem olarak birlikte çalışan, birbirine bağlı bağımsız sistemler kümesi.



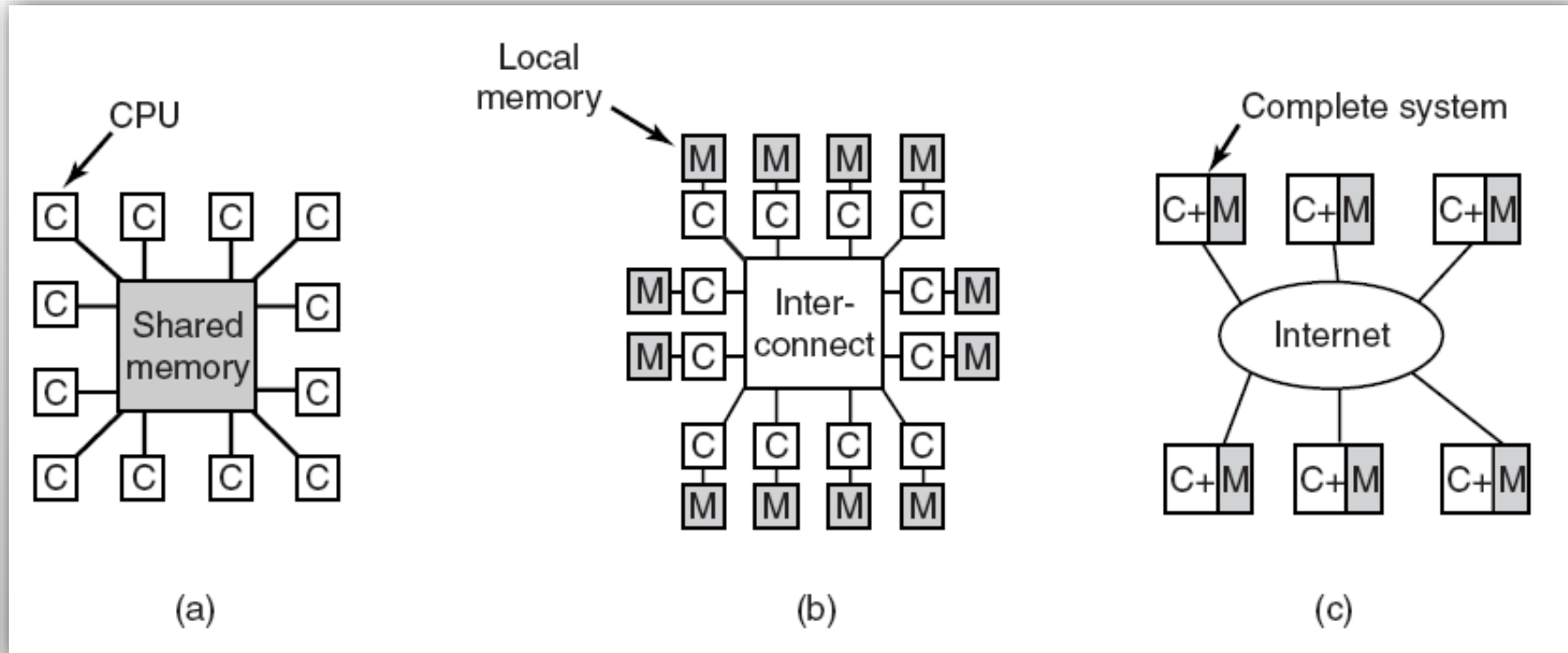
# Çok İşlemcili Sistemler

- İşlemcileri ve paylaşılan kaynakları bağlamak için ağ ve veri yolu kullanılır.
- İşletim sistemi, işlemci tahsisini, senkronizasyonu ve işlemler arasındaki iletişimi yönetmelidir.
- Daha fazla işlemci eklenerek performansı artırma yeteneği.
- Optimum kullanımı sağlamak için görevler işlemciler arasında dağıtılır.
- Paylaşılan verilerin güncel ve tutarlı olmasının sağlanması.
- Performansı artırmak için aynı anda birden çok süreç yürütebilmeli.



# Çok İşlemcili Sistemler

- (a) Paylaşımlı bellek çok işlemcisi. (b) Mesaj ileten çoklu bilgisayar. (c) Geniş alan dağıtılmış bir sistem.





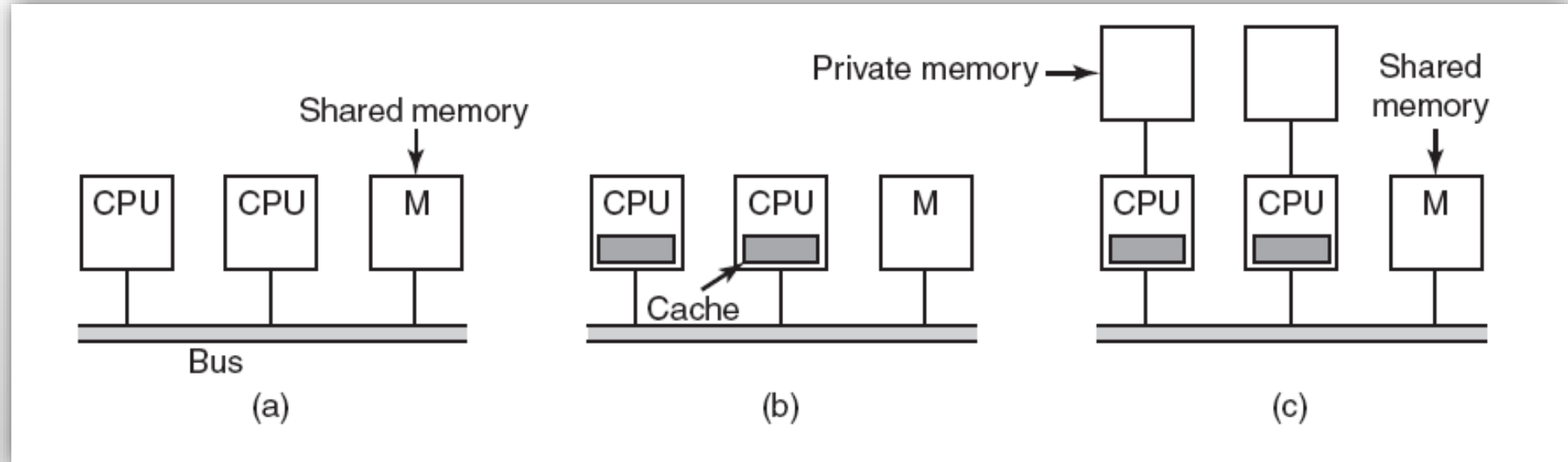
# Veri Yolu Tabanlı Mimariler

- **Tekdüzen Bellek Erişimi (UMA):** tüm işlemciler paylaşılan belleğe aynı erişim süresine sahiptir. Paralel algoritmaların programlanmasını basitleştirir ve tüm işlemcilerin eşit derecede iyi performans göstermesini sağlar.
- **Önbellek:** işlemciye yakın yerleştirilmiş küçük ve hızlı bir bellektir.
- **Paylaşılan Bellek:** birden çok işlemcinin aynı belleğe erişimi vardır, bu da onların verileri paylaşmasına ve birbirleriyle iletişim kurmasını sağlar.
- **Özel Bellek:** sistemdeki her işlemciye atanır ve onlara yerel değişkenlerin ve verilerin depolanması için ayrılmış bir alan sağlar.



# Veri Yolu Tabanlı Mimariler

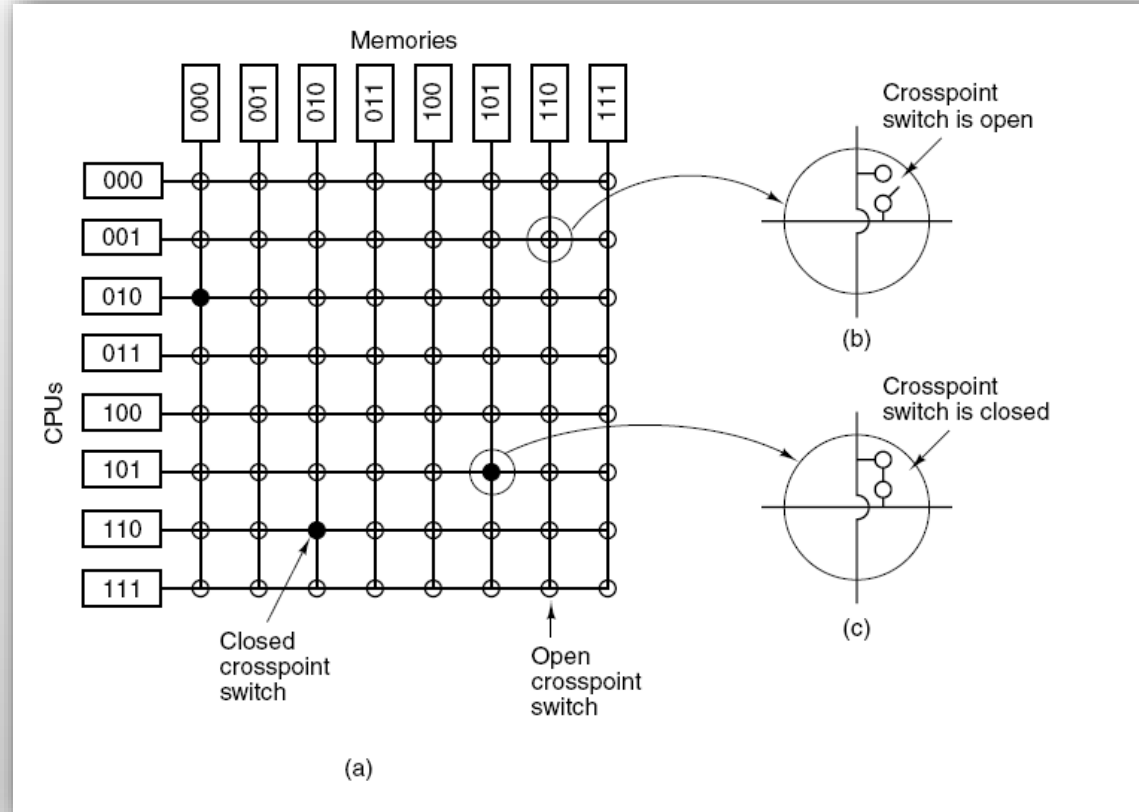
- Üç çeşit veri yolu tabanlı UMA (Uniform Memory Access) çoklu işlemci sistem. (a) paylaşımlı bellek. (b) önbellek. (c) önbellek ve özel bellek bir arada.





# Çapraz Çubuk Anahtarları Kullanarak Bellek Erişimi

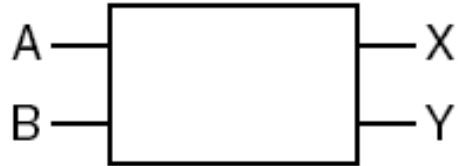
- (a) 8 × 8 çapraz çubuk anahtarı. (b) Açık bir kesişme noktası. (c) Kapalı bir kesişme noktası.



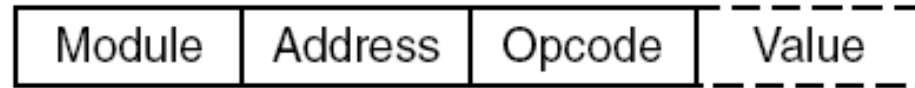
# Çok Aşamalı Anahtarlama Ağlarının Kullanımı



- (a) A ve B olmak üzere iki giriş hattına ve X ve Y olmak üzere iki çıkış hattına sahip bir  $2 \times 2$  anahtarı. (b) Bir mesaj formatı.



(a)

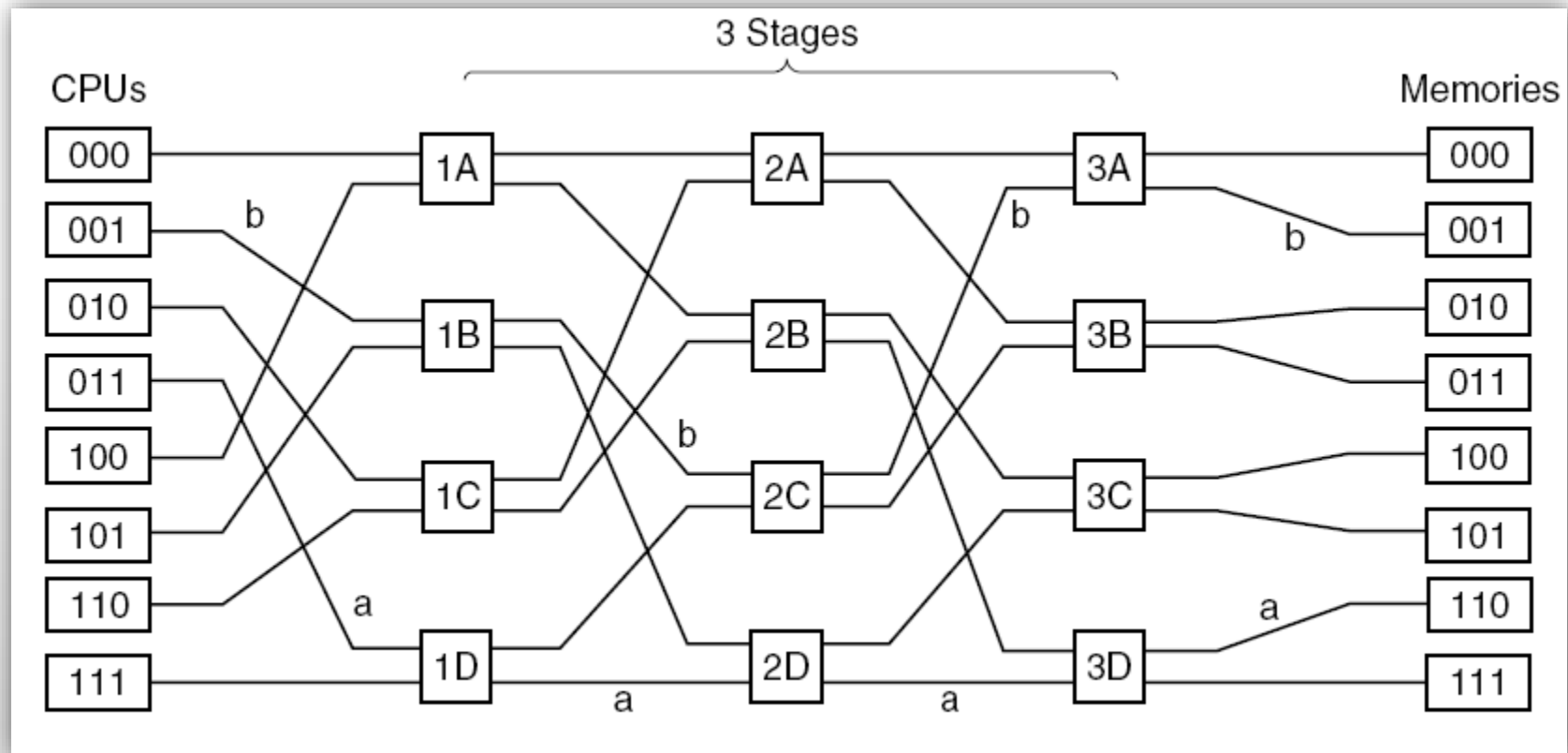


(b)





# Bir Omega Anahtarlama Ağı





# Bir Omega Anahtarlama Ağı

- Çok işlemcili bir sistemde işlemciler ve paylaşılan bellek arasında ara bağlantı sağlayan bir bilgisayar ağ mimarisidir.
- Anahtar düğümleri ve düğümler arasındaki ara bağlantıları olan ağ benzeri bir yapıya sahiptir.
- İşlemciler arasında eşzamanlı iletişime izin verir.
- Tasarımı karmaşık ve uygulaması zor.
- Çok sayıda anahtar düğümü nedeniyle güç tüketimi yüksek.



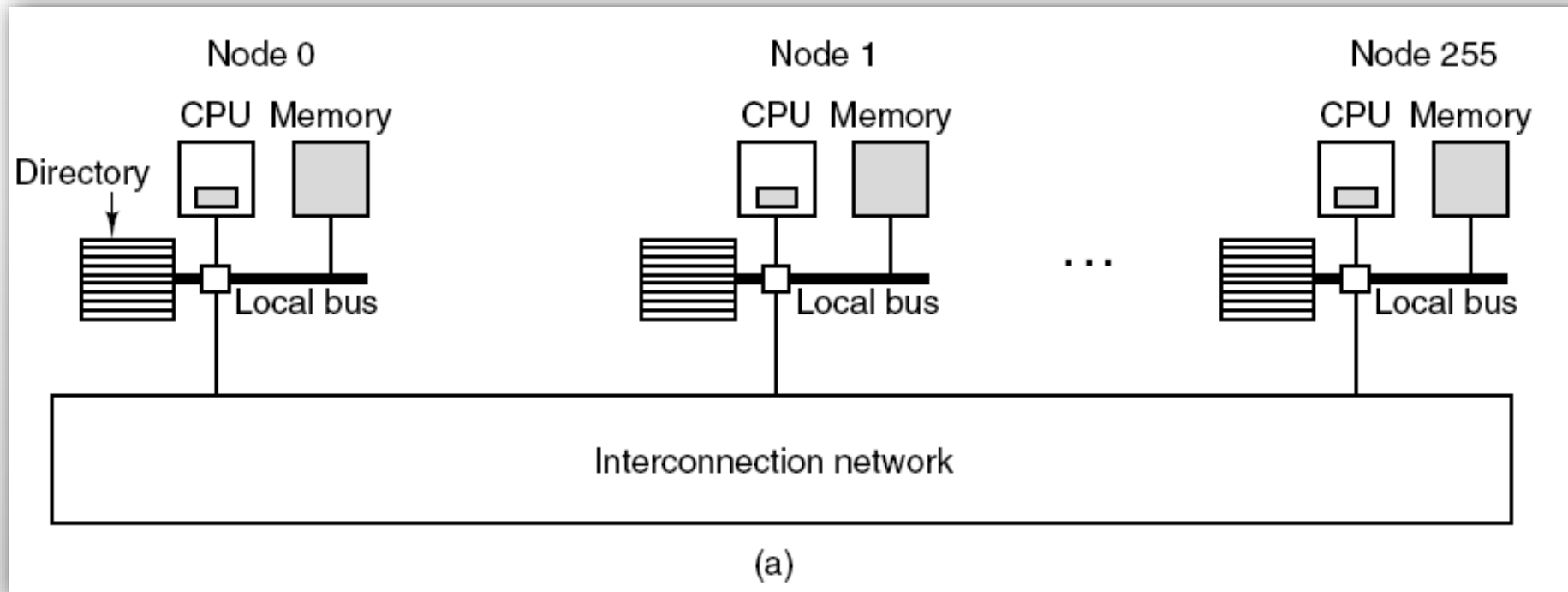
# NUMA Çok İşlemcileri

- NUMA makinelerinin özellikleri:
  - Tüm CPU'lar tarafından görülebilen tek bir adres alanı vardır.
  - Uzak belleğe erişim LOAD ve STORE komutları ile yapılır.
  - Uzak belleğe erişim, yerel belleğe erişimden daha yavaştır.



# NUMA Çok İşlemcileri

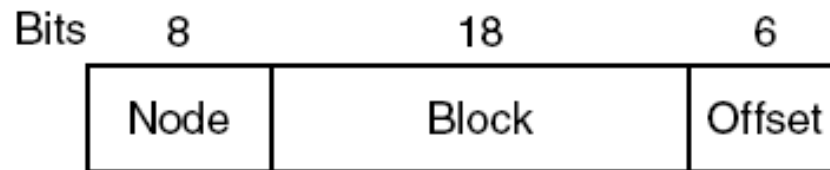
- (a) 256 düğümlü dizin tabanlı çok işlemci.



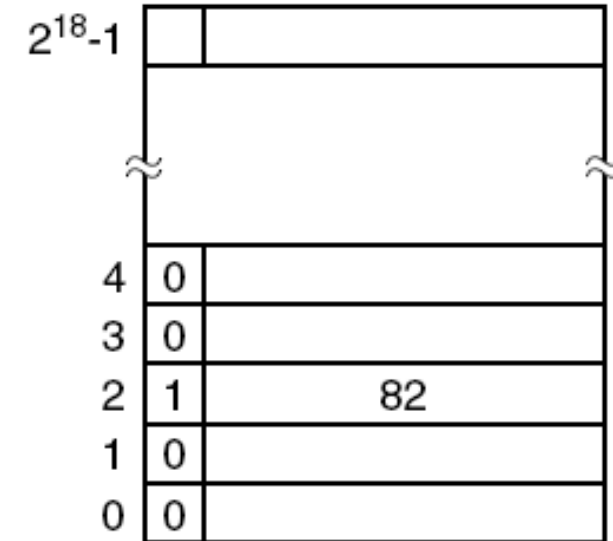


# NUMA Çok İşlemcileri

- (b) 32 bitlik bir bellek adresinin alanlara bölünmesi. (c) 36. düğümdeki dizin.



(b)

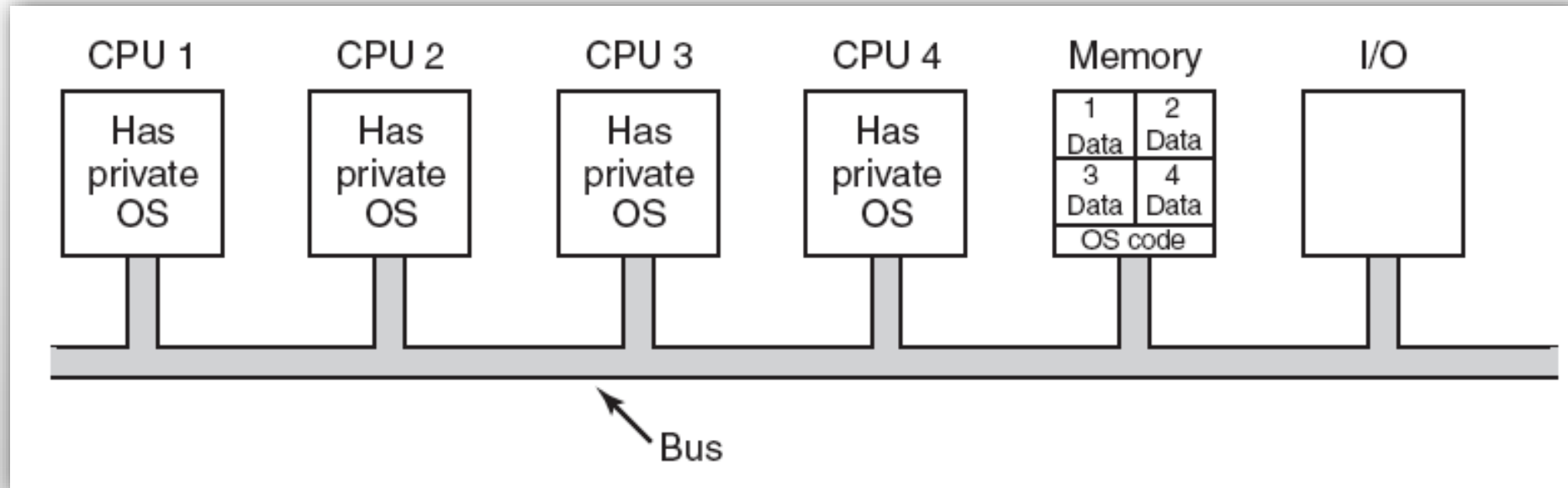


(c)



# Her CPU'nun Kendi İşletim Sistemi Vardır

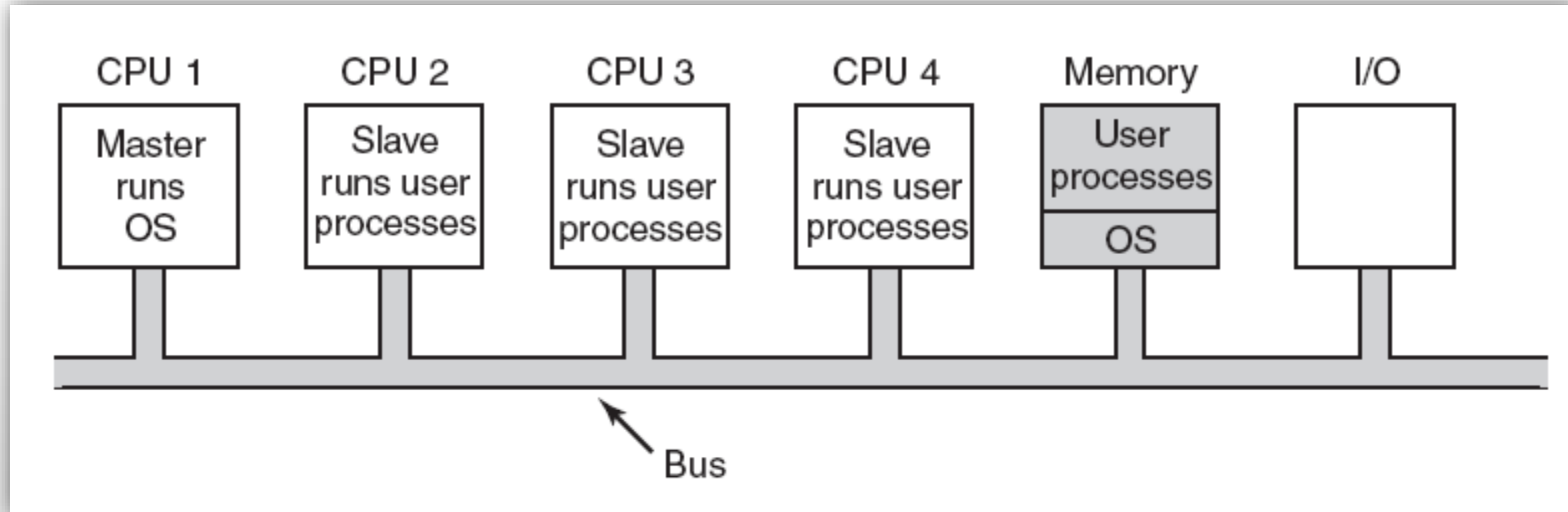
- Belleği dört CPU arasında bölünür, ancak işletim sistemi kodunun tek bir kopyası paylaşılır. Veri olarak işaretlenen kutular, işletim sisteminin her CPU için özel verileridir.





# Master-Slave Çok İşlemcili Model

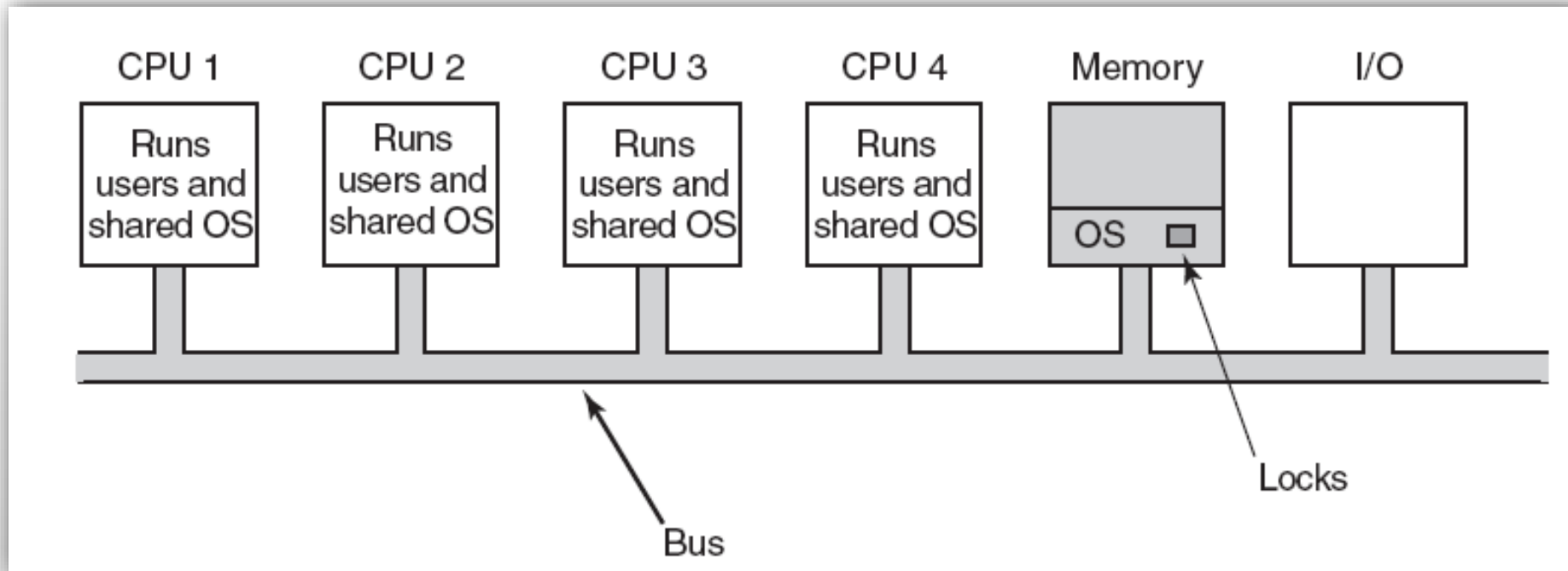
■ .





# Simetrik Çoklu İşlemciler

- SMP çok işlemcili model

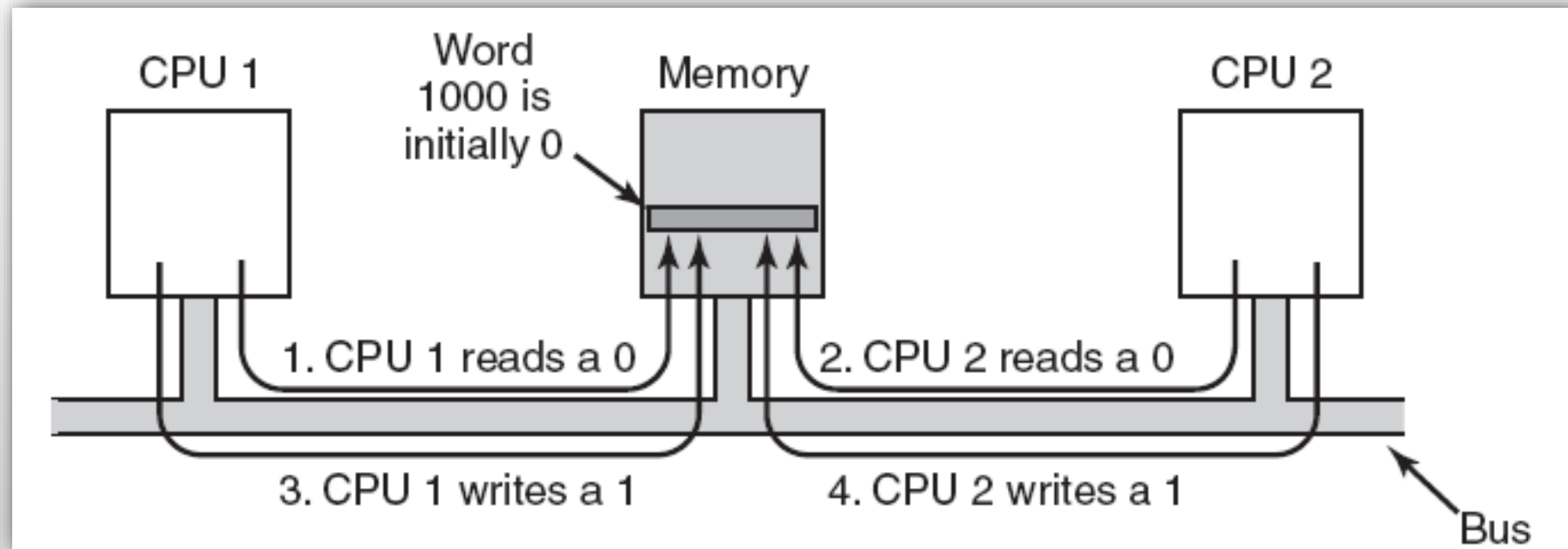






# Çok İşlemcili Senkronizasyon

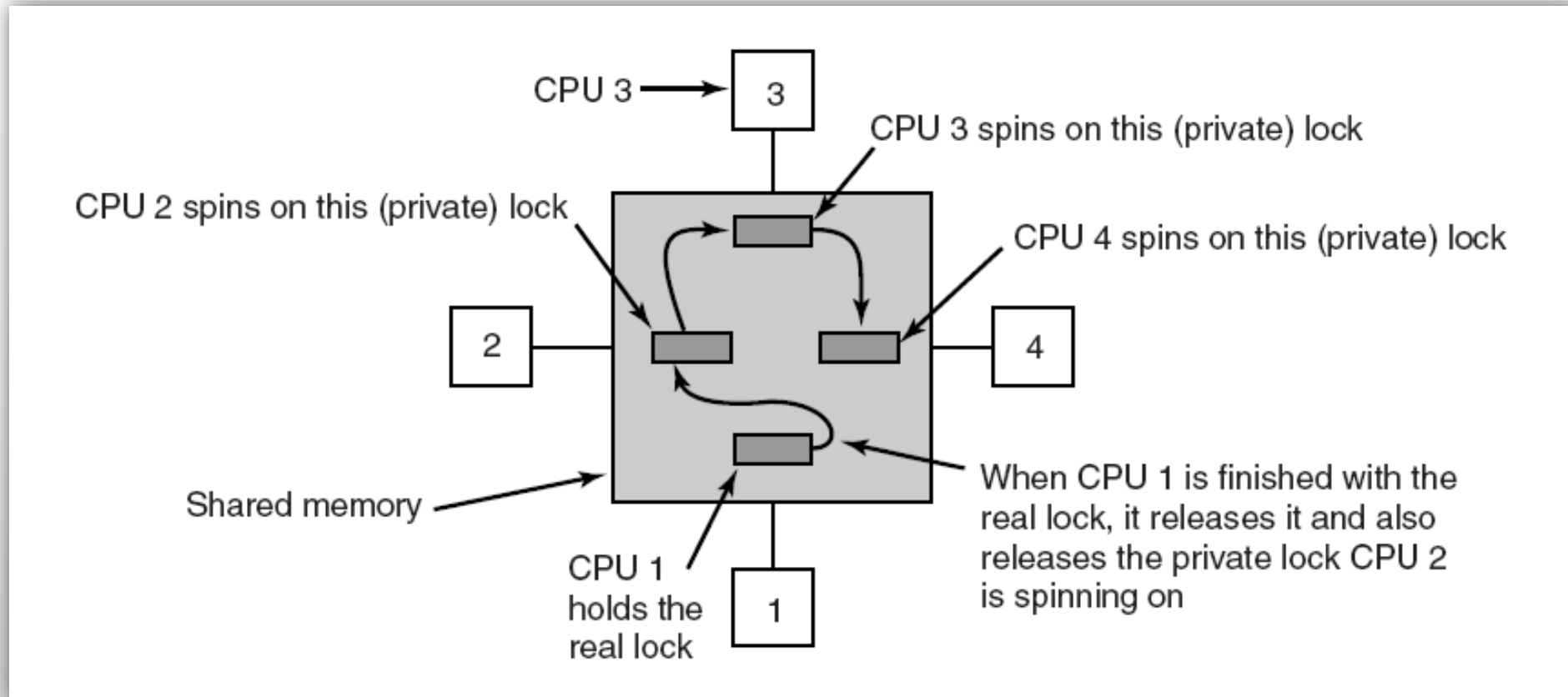
- Veri yolu kilitlenemezse TSL komutu başarısız olabilir. Bu dört adım, başarısızlığın gösterildiği bir olaylar dizisini gösterir.





# Çok İşlemcili Senkronizasyon

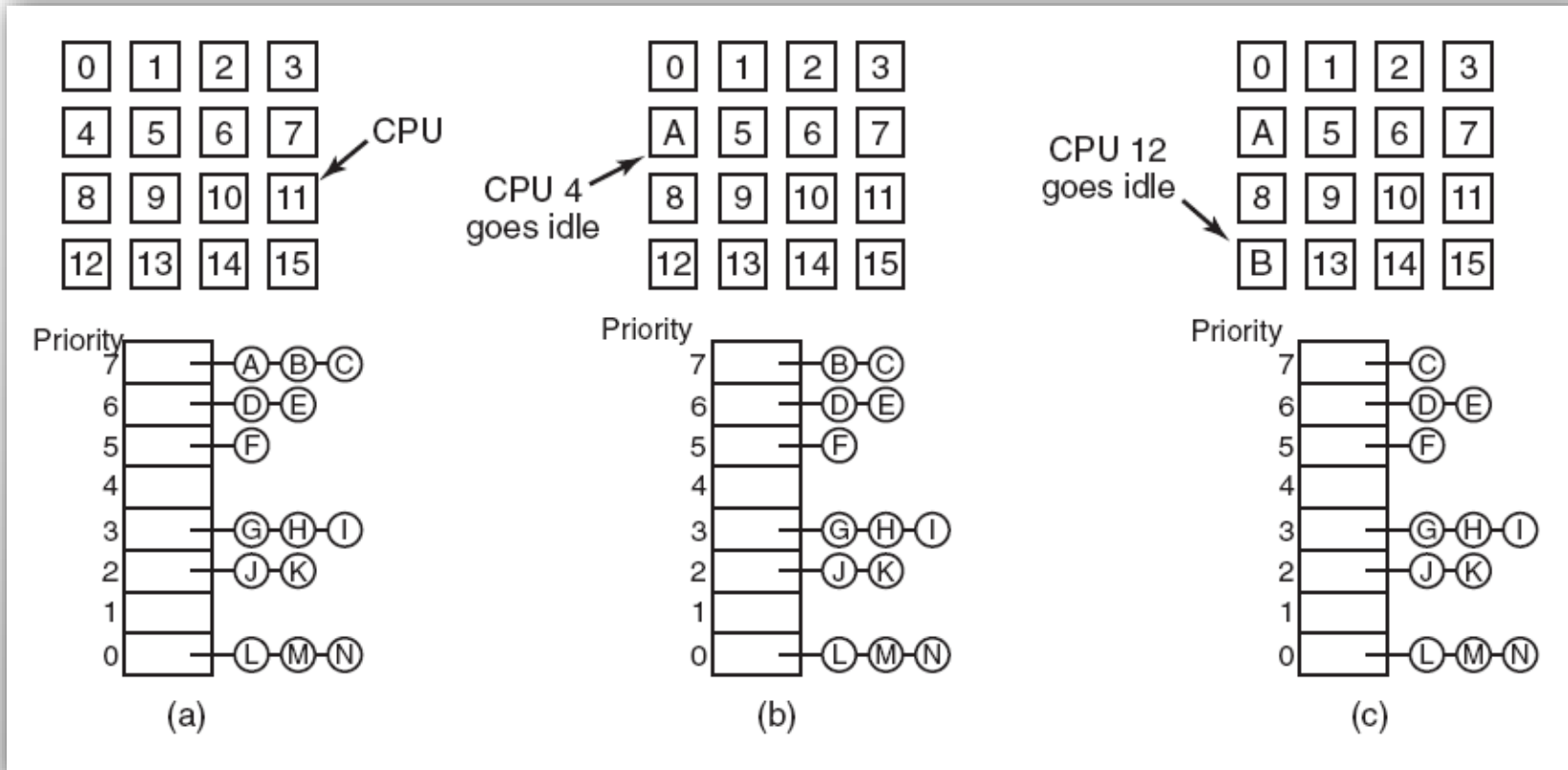
- Önbelleğin çöp olmasını önlemek için çoklu kilit kullanımı





# Zaman Paylaşımı

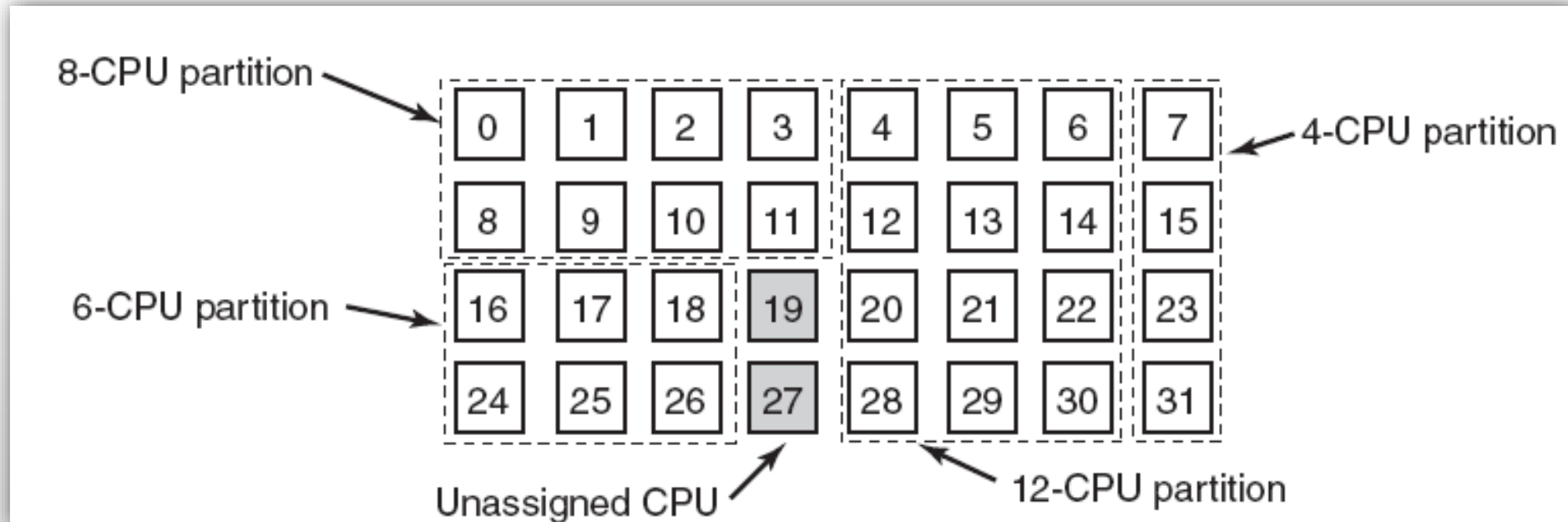
- Çok işlemcili zamanlama için tek bir veri yapısı kullanma.





# Alan Paylaşımı

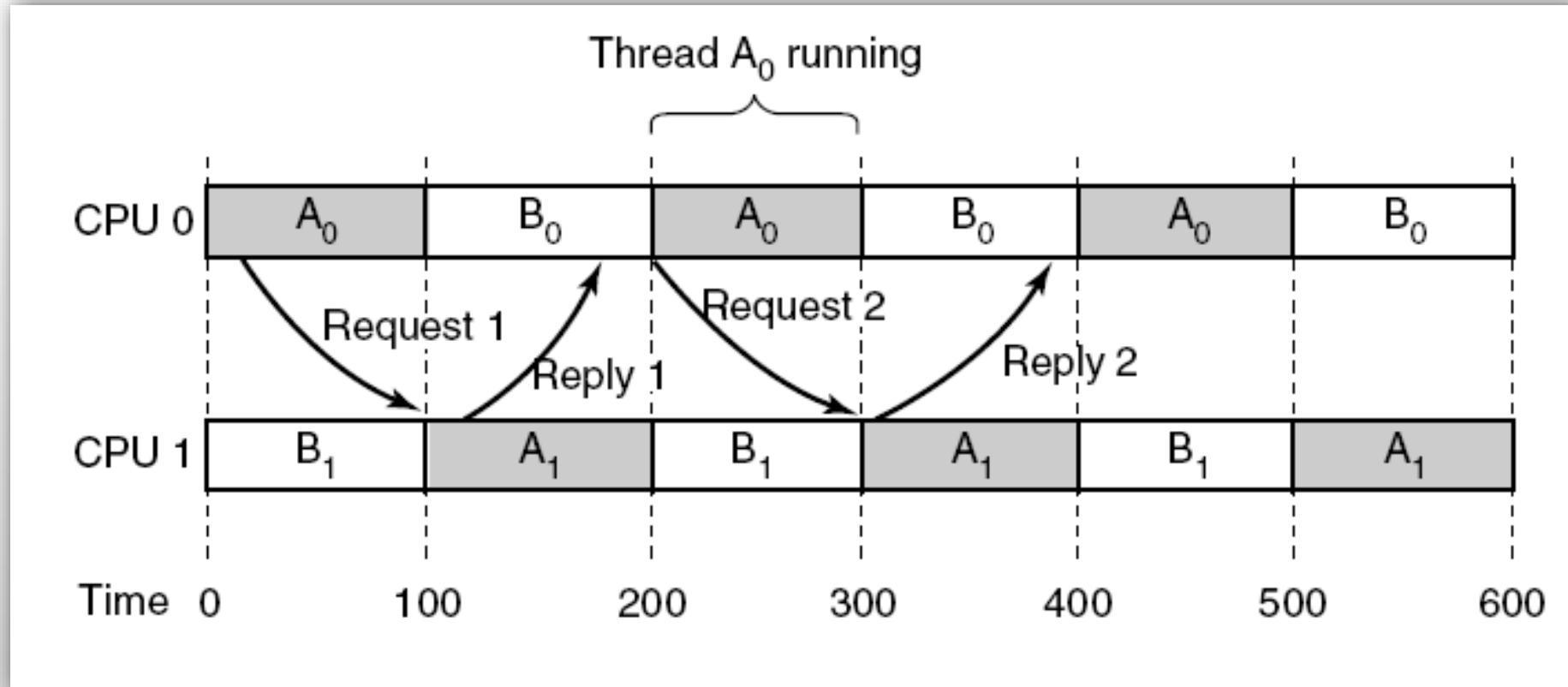
- 32 CPU kümesi, iki CPU boşta olmak üzere dört bölüme ayrılmıştır.





# Gang Çizelgeleme

- Aşaması biten A iş parçacığına ait iki iş parçacığı arasındaki iletişim.





# Gang Çizelgeleme

- Çete çizelgelemesinin üç bölümü:
  - İlgili iş parçacığı grupları bir birim, bir çete olarak planlanır.
  - Tüm üyeler, farklı zaman-paylaşımlı CPU'larda aynı anda çalışır.
  - Tüm üyeler zaman dilimlerini birlikte başlatır ve bitirir.



# Gang Çizelgeleme

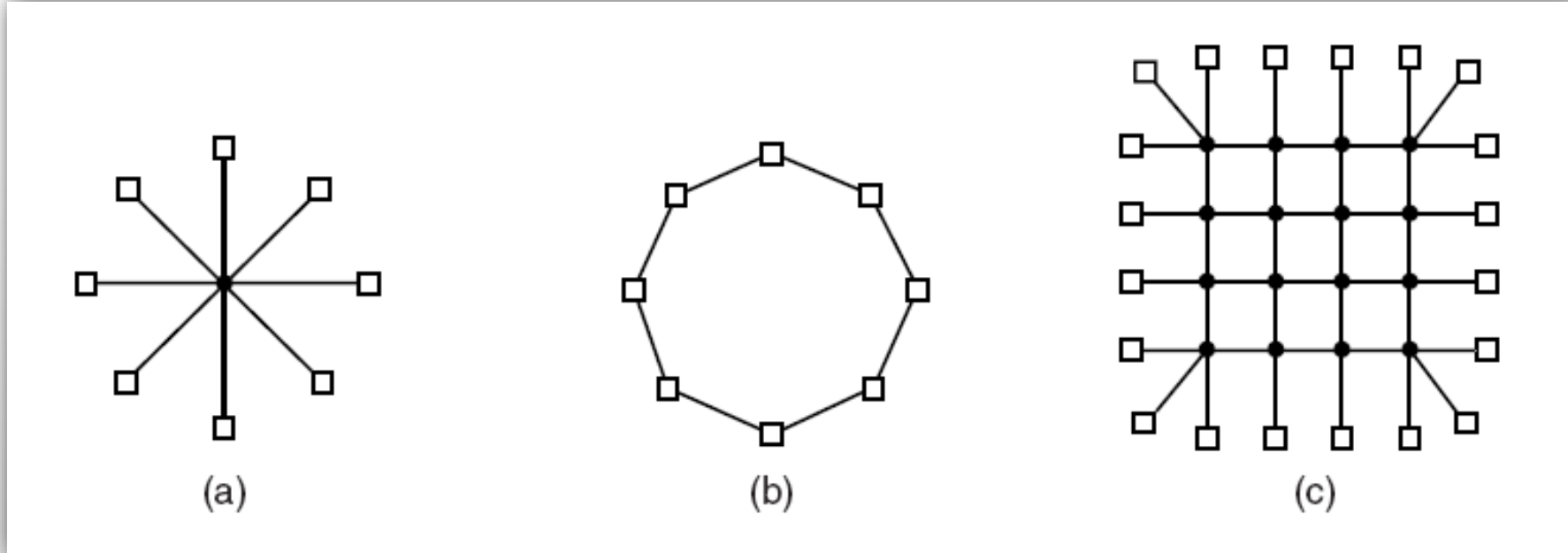
■ .

		CPU					
		0	1	2	3	4	5
Time slot	0	$A_0$	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$
	1	$B_0$	$B_1$	$B_2$	$C_0$	$C_1$	$C_2$
	2	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$E_0$
	3	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$
	4	$A_0$	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$
	5	$B_0$	$B_1$	$B_2$	$C_0$	$C_1$	$C_2$
	6	$D_0$	$D_1$	$D_2$	$D_3$	$D_4$	$E_0$
	7	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$



# Arabađlantı Teknolojisi

- Çeşitli ara bađlantı topolojileri. (a) Tek anahtar. (b) yüzük. (c) ızgara.

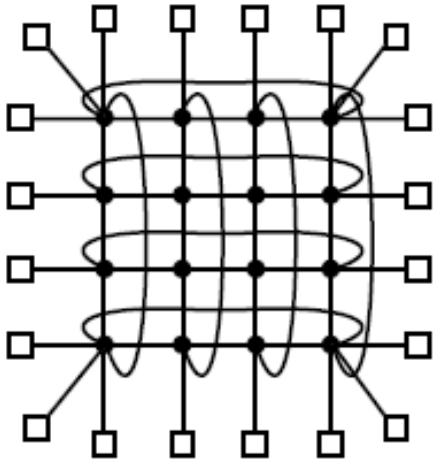




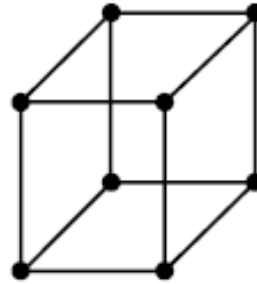


# Arabağlantı Teknolojisi

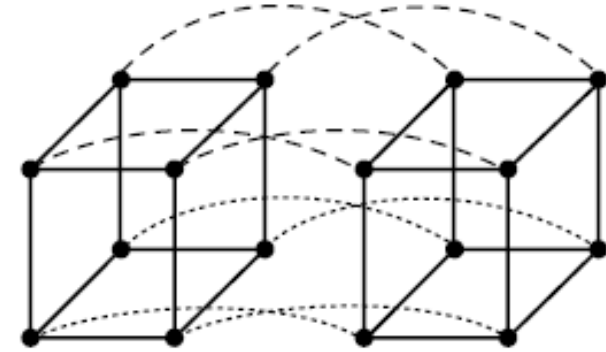
- Ara bağlantı topolojileri. (d) Çift simit (torus). (e) küp. (f) 4D hiperküp.



(d)



(e)

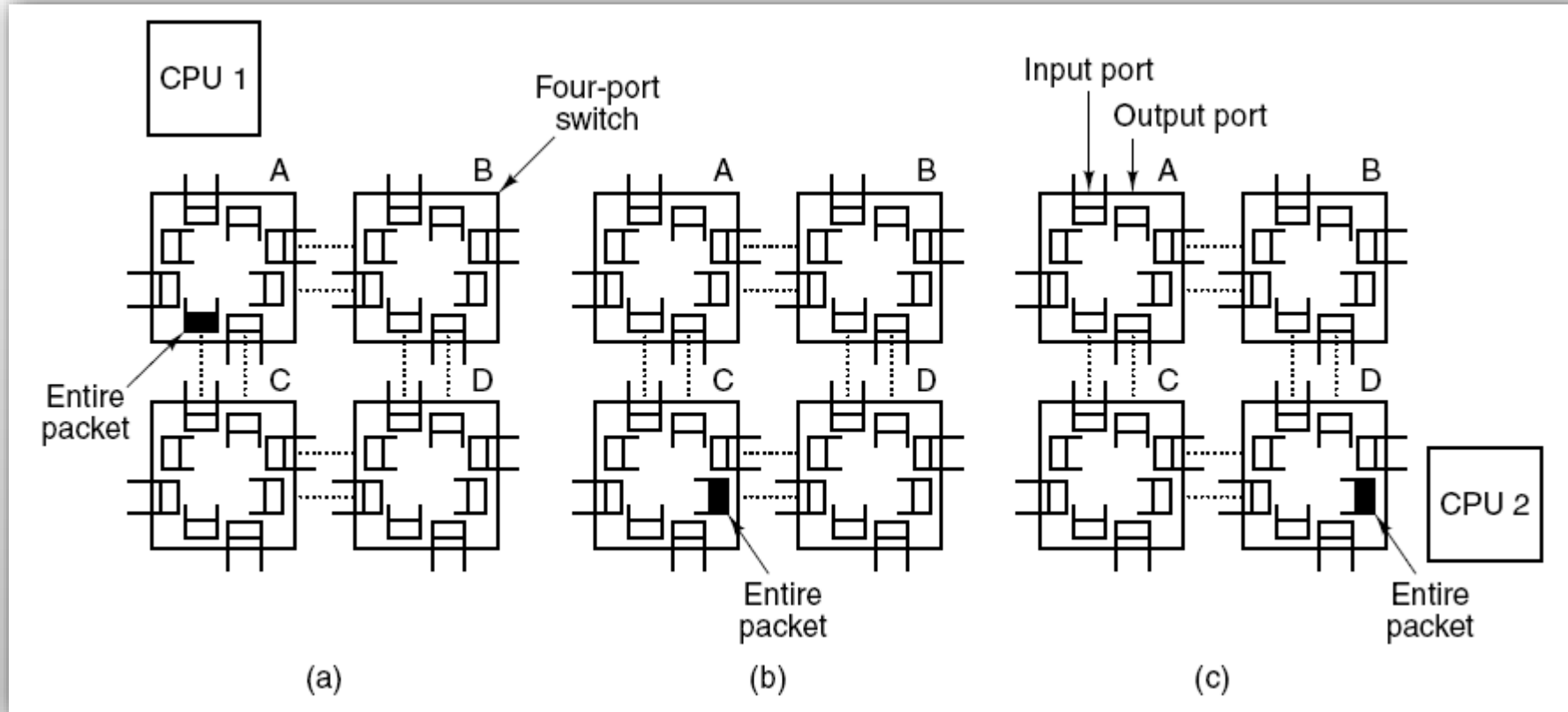


(f)



# Arabağlantı Teknolojisi

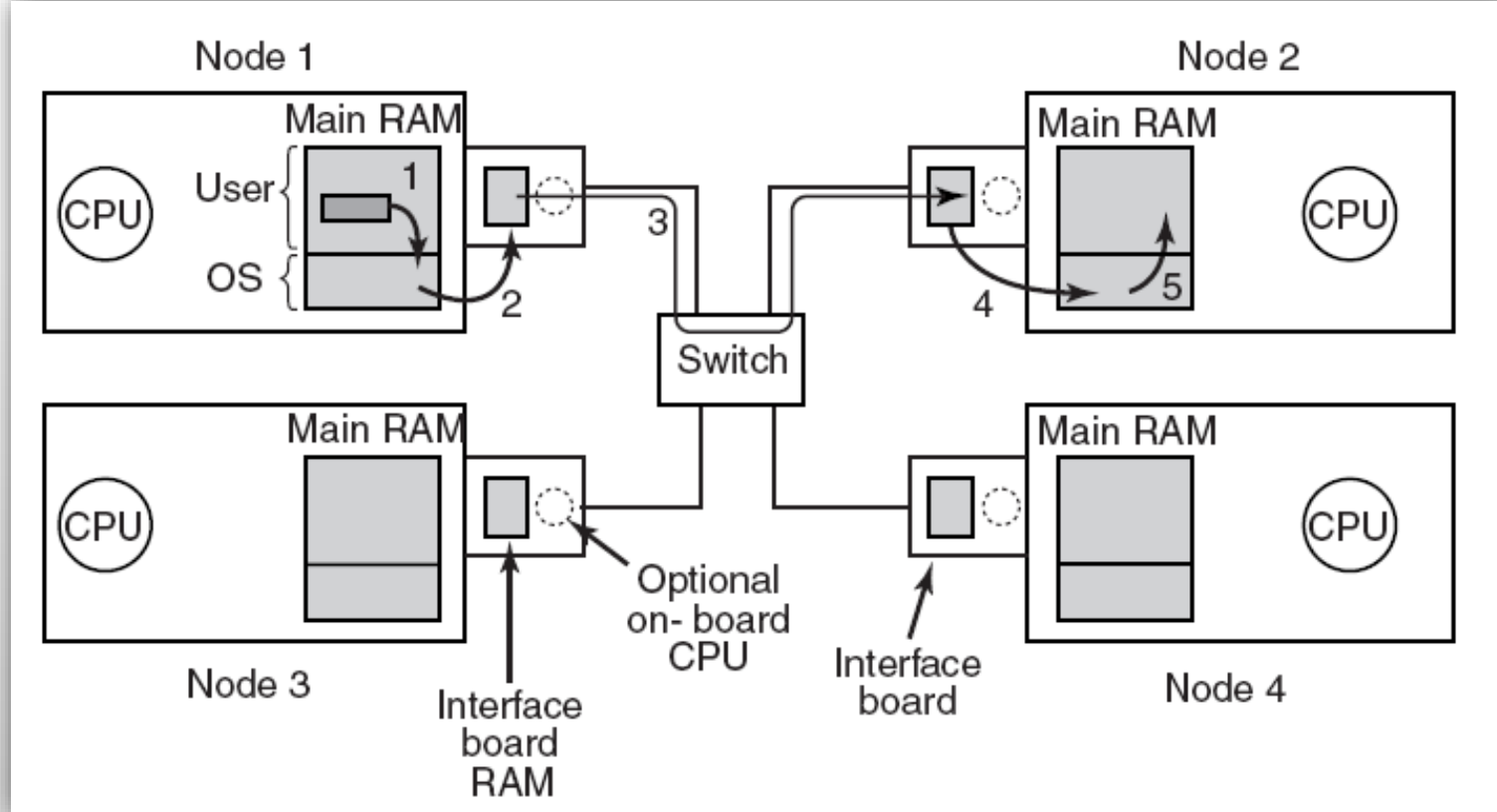
- Depola ve ilet (store and forward) paket anahtarlama.





# Ağ Arayüzleri

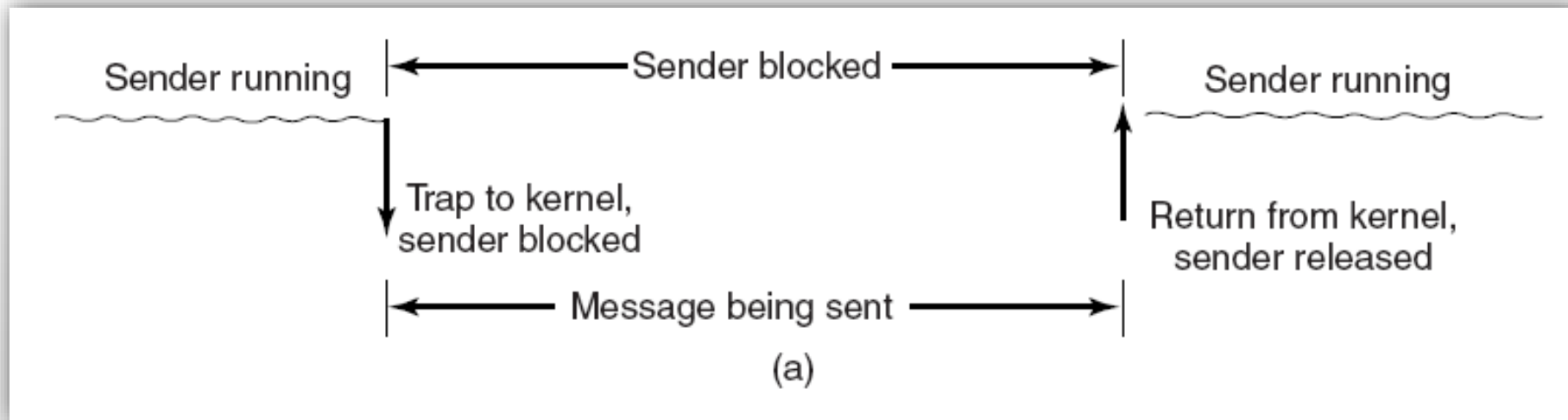
- Bir çoklu bilgisayarda ağ arayüz kartlarının konumu.





# Bloke Eden ve Etmeyen Çağrılar

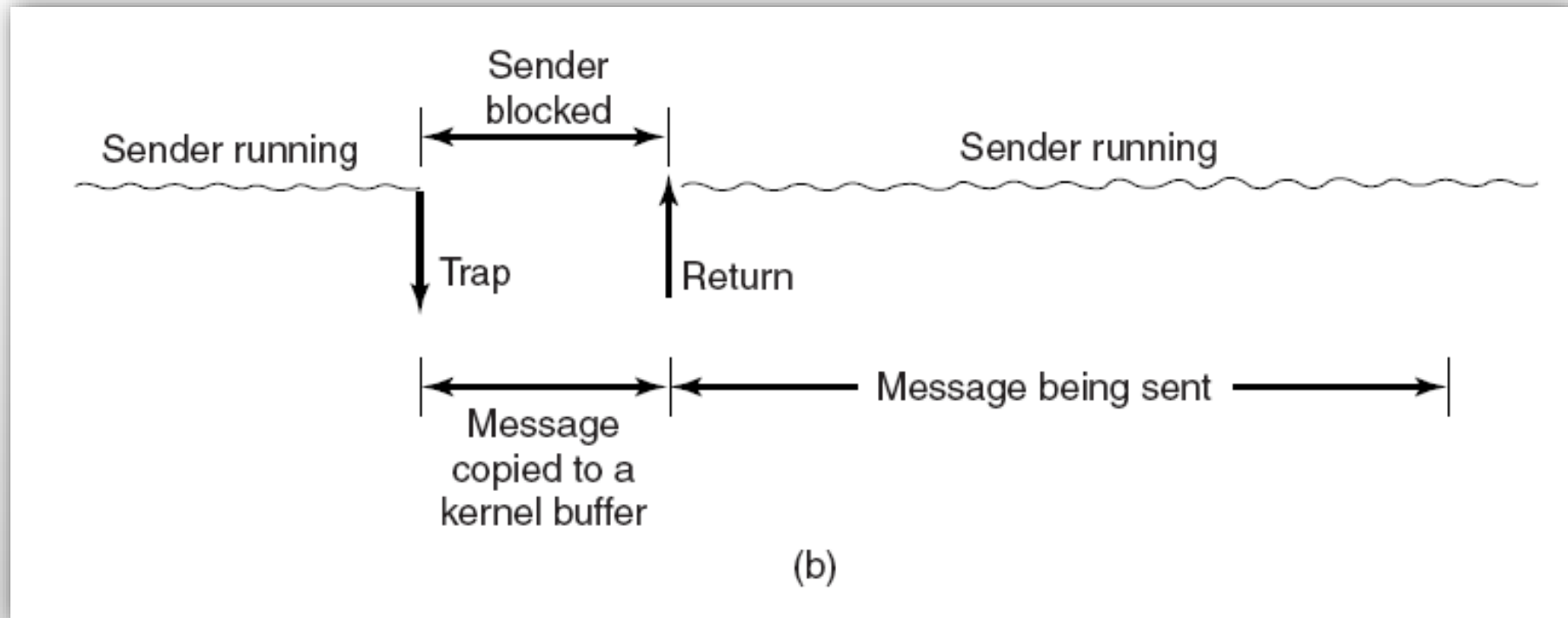
- (a) Bloke eden (blocking) gönder çağrısı.





# Bloke Eden ve Etmeyen Çağrılar

- (b) Bloke etmeyen (non-blocking) gönder çağrısı.





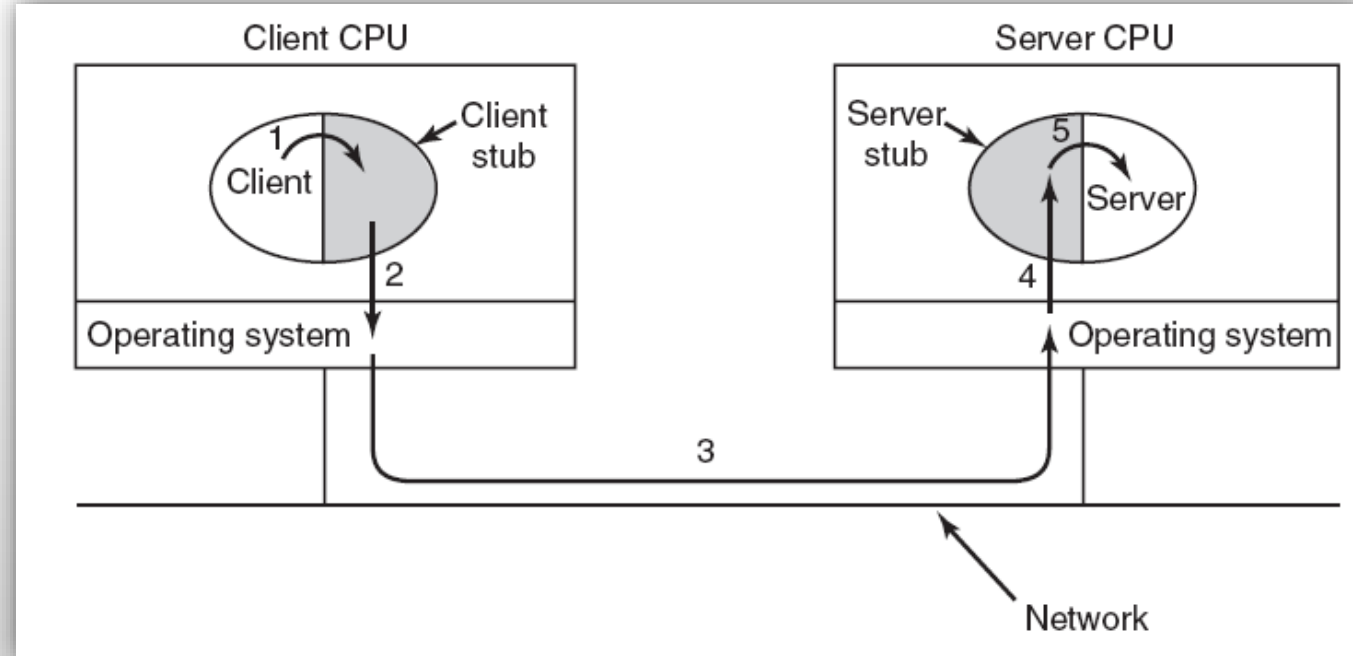
# Bloke Eden ve Etmeyen Çağrılar

- Gönderen taraftaki seçenekler:
- Bloke eden (blocking) gönderim (mesaj iletimi sırasında CPU boşta bekler).
- Bloke etmeyen (non-blocking) kopya ile gönderim (fazladan kopya için CPU zamanı boşa harcanır).
- Kesilme ile bloke etmeyen gönderim (programlamayı zorlaştırır).
- Yazarken kopyalama (sonunda muhtemelen fazladan kopya gerekir).



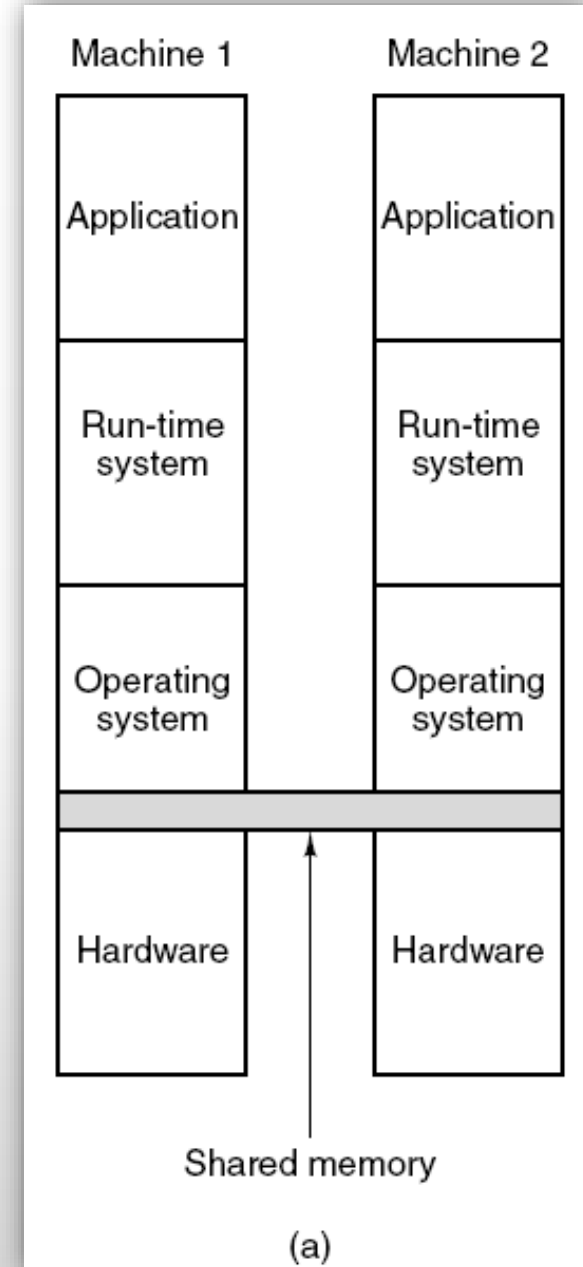
# Uzaktan Prosedür Çağrısı

- Uzaktan prosedür çağrısı (remote procedur call) yapma adımları. Stub: Program geliştirme sürecinde daha sonra aslıyla değiştirilmek üzere geçici olarak tutulan yürütülebilir program.



# Dağıtık Paylaşımlı Bellek

- Paylaşılan belleğin uygulanabileceği çeşitli katmanlar. (a) Donanım.

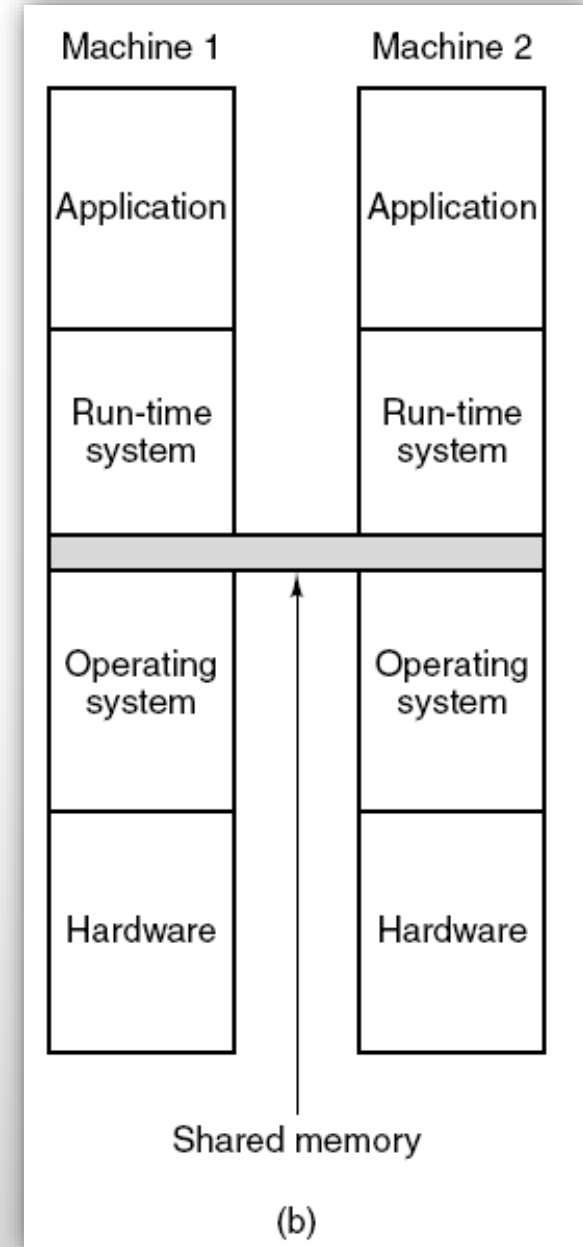






# Dağıtık Paylaşımlı Bellek

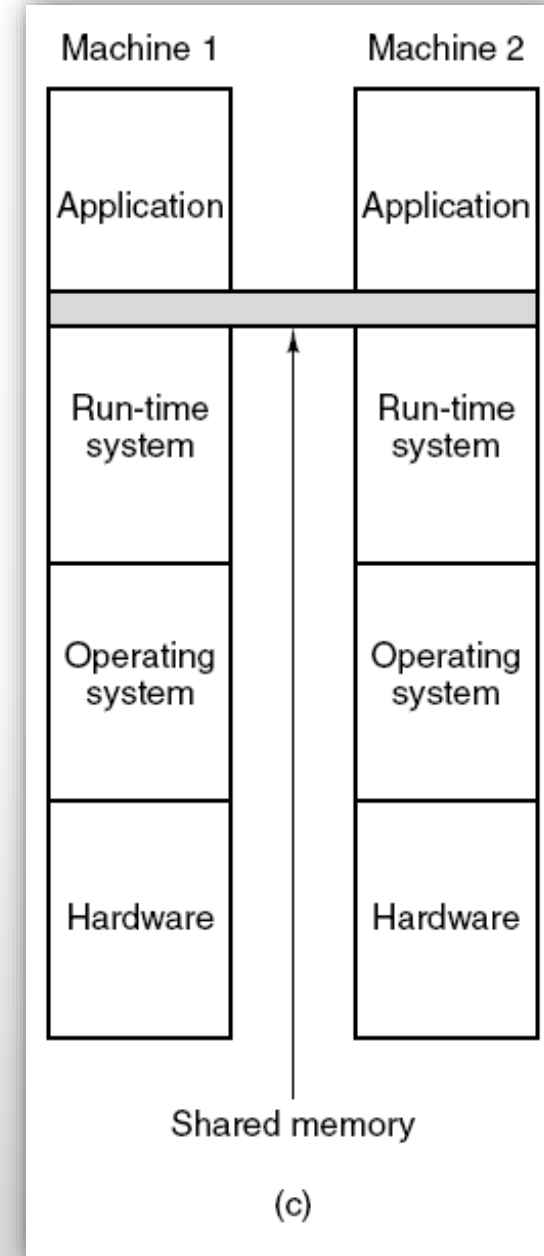
- Paylaşılan belleğin uygulanabileceği çeşitli katmanlar. (b) İşletim sistemi.





# Dağıtık Paylaşımlı Bellek

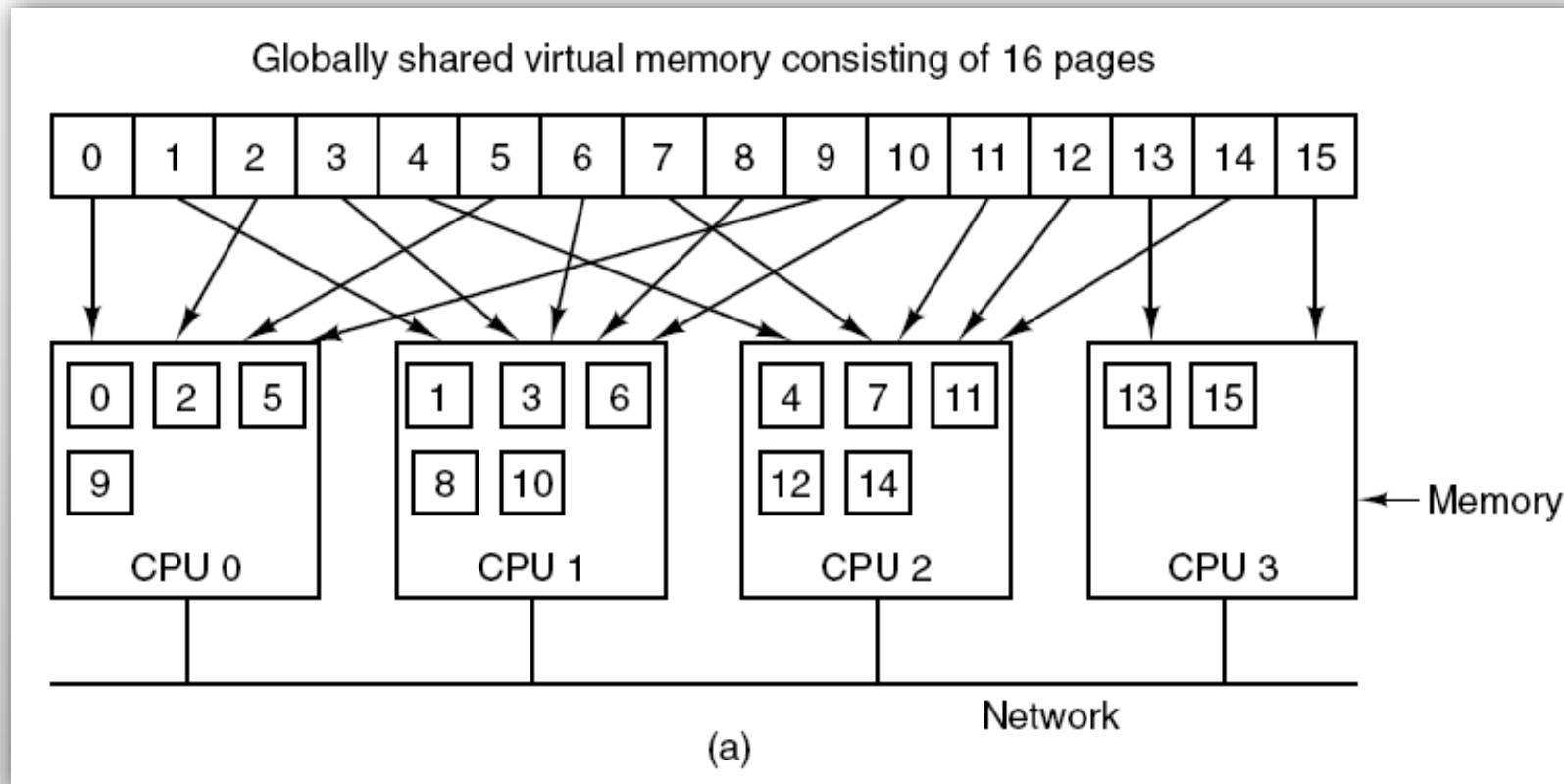
- Paylaşılan belleğin uygulanabileceği çeşitli katmanlar. (c) Kullanıcı yazılımı.





# Dağıtık Paylaşımlı Bellek

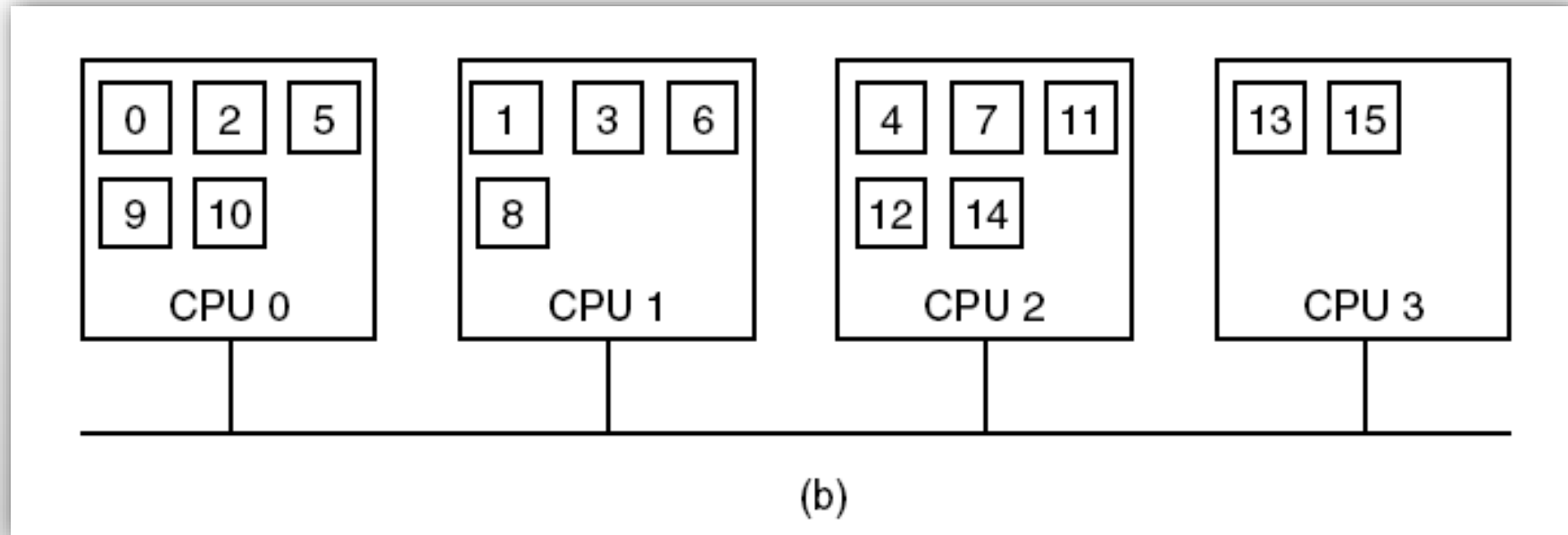
- (a) Dört makine arasında dağıtılan adres alanının sayfaları.





# Dağıtık Paylaşımlı Bellek

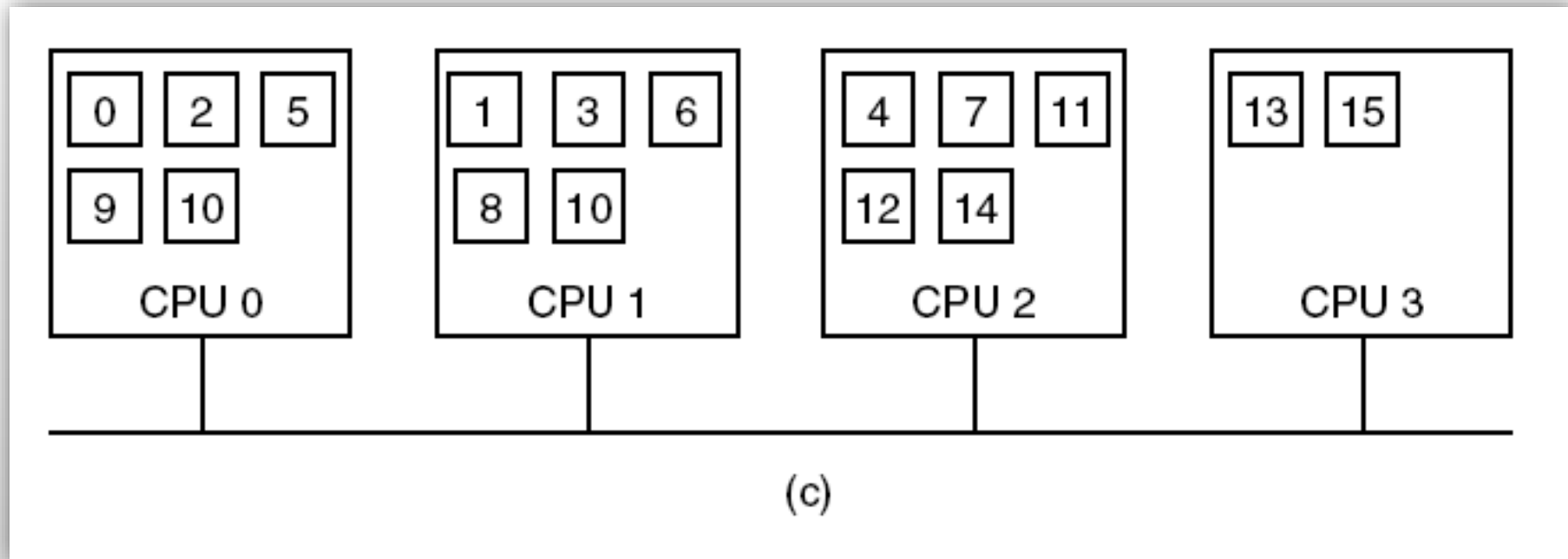
- (b) CPU 1 sayfa 10'a referans verdikten ve sayfa oraya taşındıktan sonraki durum.





# Dağıtık Paylaşımlı Bellek

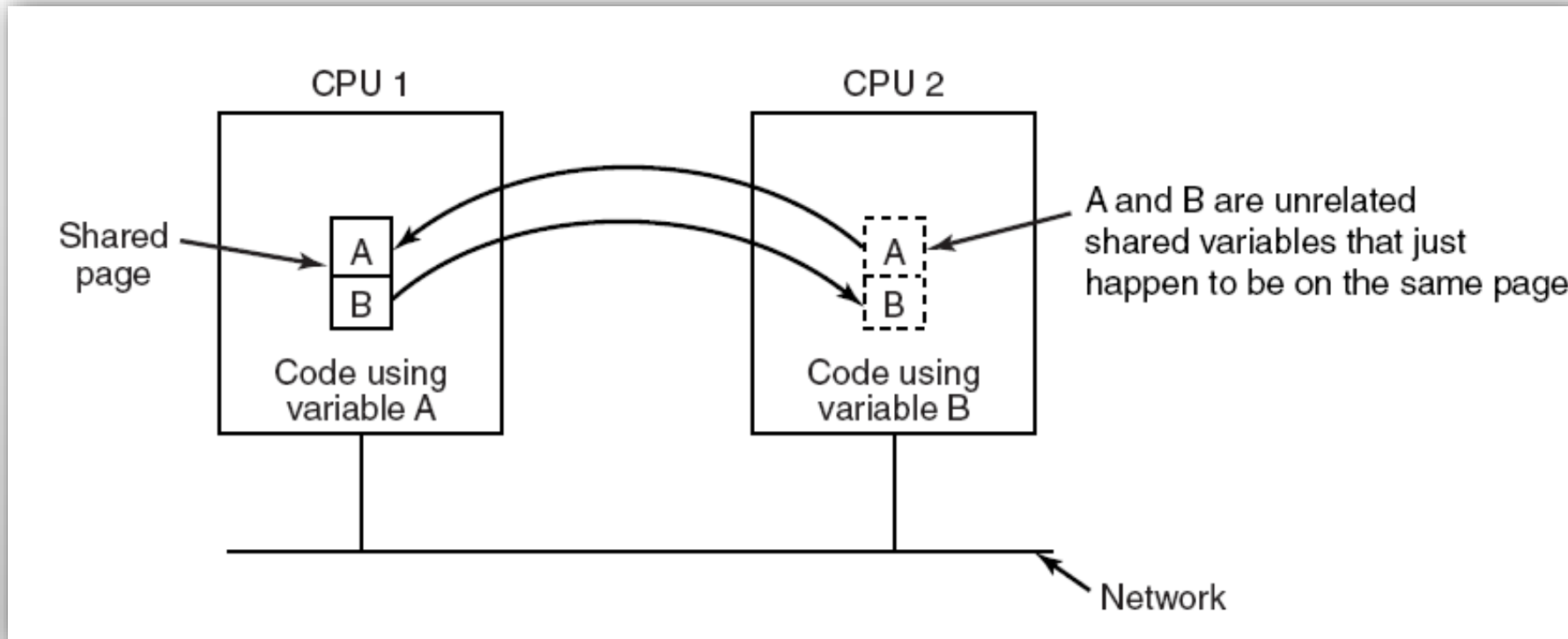
- (c) Sayfa 10 salt okunursa ve kopyalayarak çoğaltma (replication) kullanılıyorsa durum.





# Yanlış Paylaşım (false sharing)

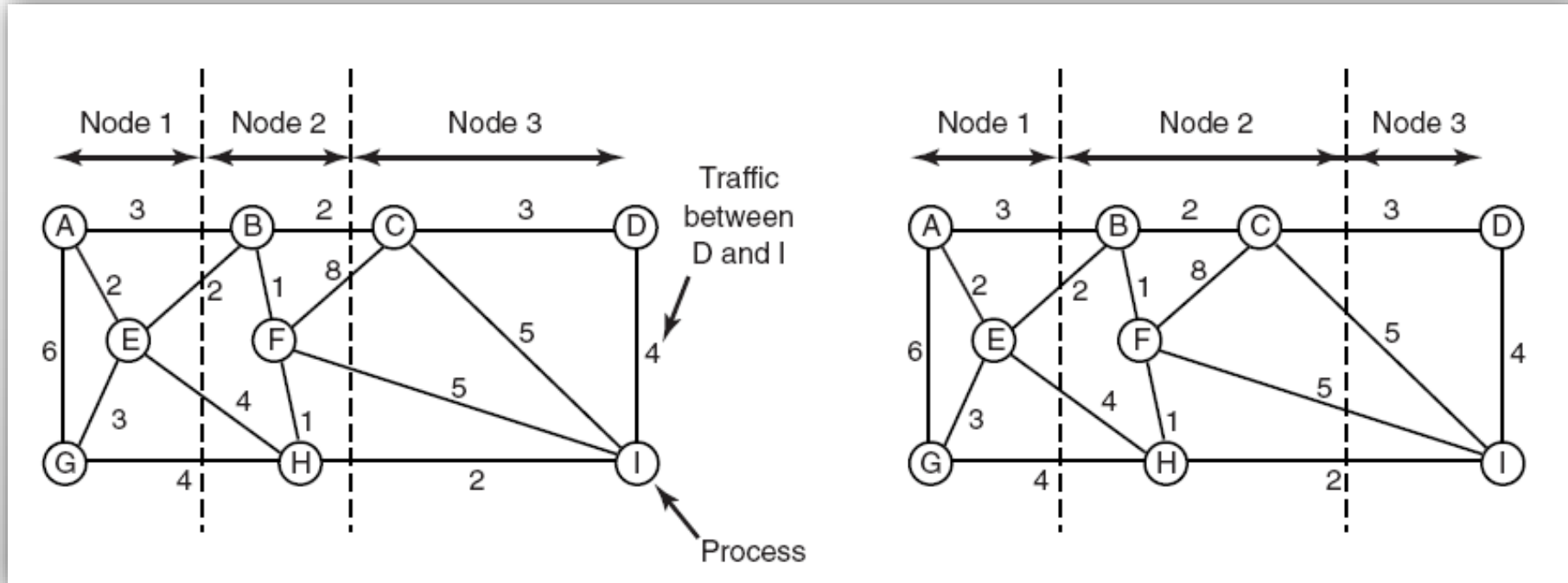
- İki alakasız değişken içeren bir sayfanın yanlış paylaşımı.





# Bir Çizge-Teorik Deterministik Algoritma

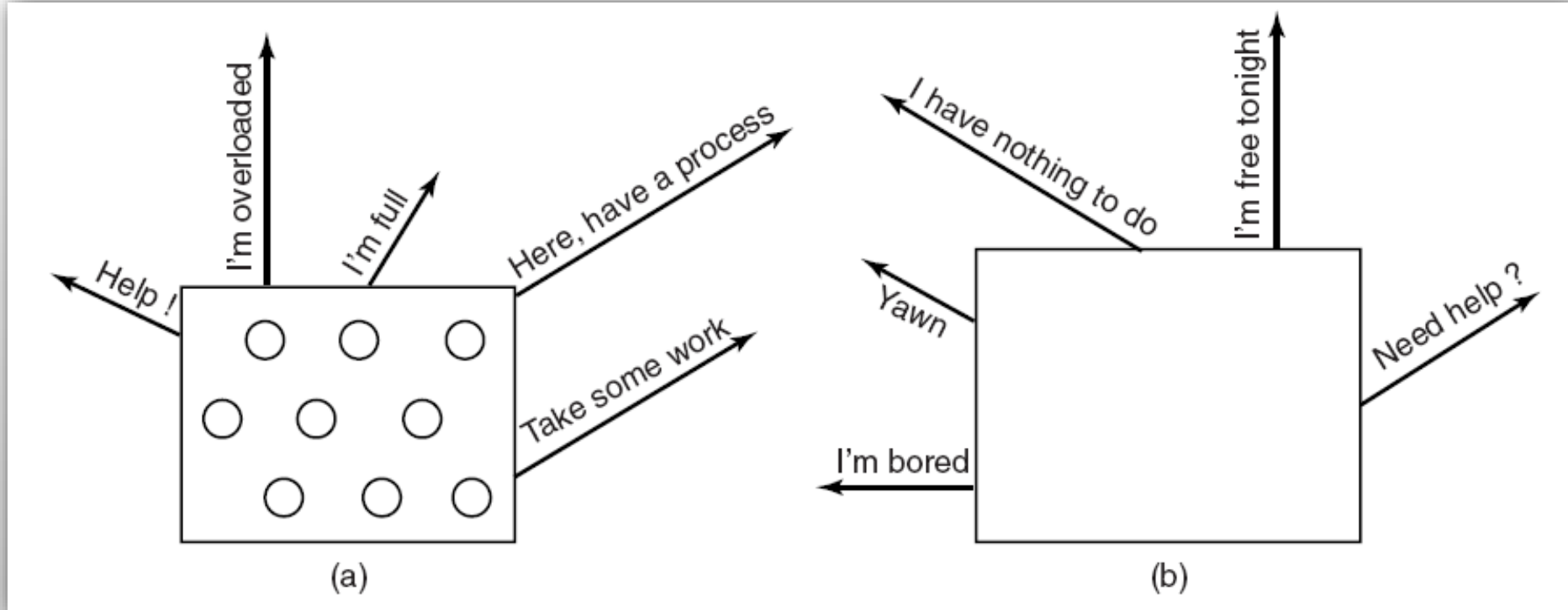
- Dokuz süreci üç düğüme atamanın iki yolu.





# Gönderici Tarafından Başlatılan Dağıtık Sezgisel Algoritma

- (a) Süreçleri devretmek için hafif yüklü bir düğüm arayan aşırı yüklenmiş düğüm. (b) Yapacak iş arayan boş bir düğüm.

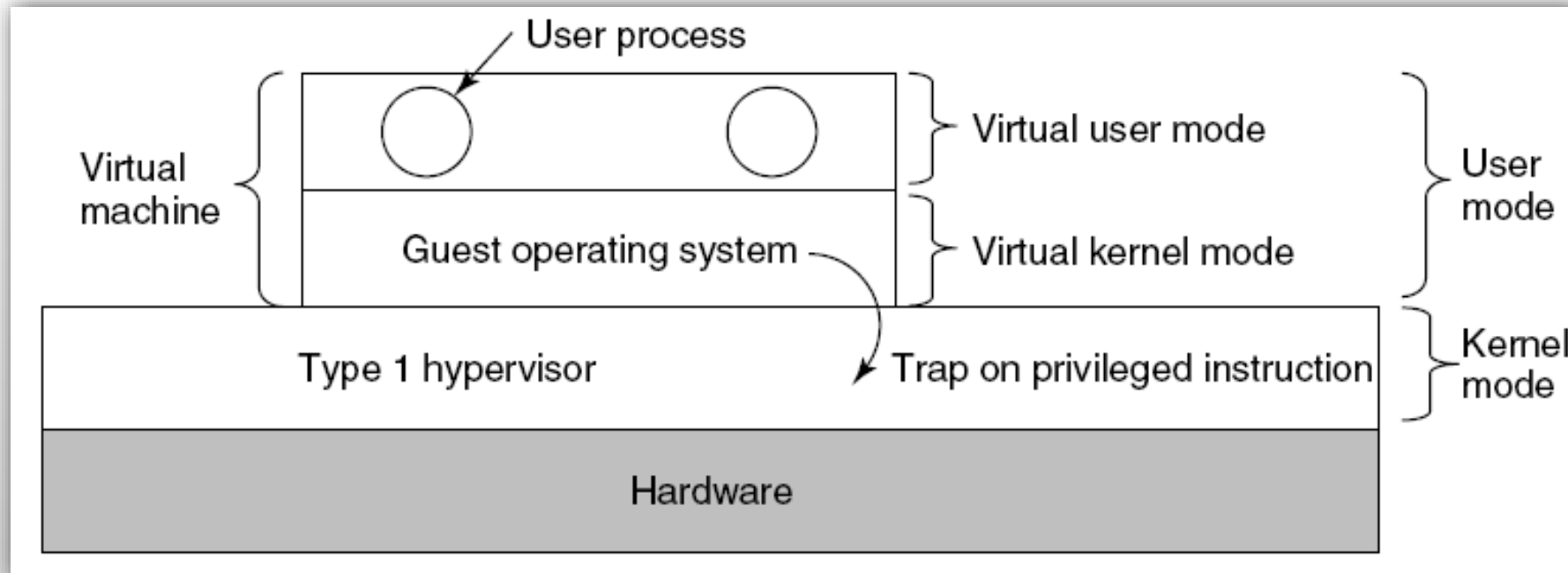






# Tip 1 Hipervizörler

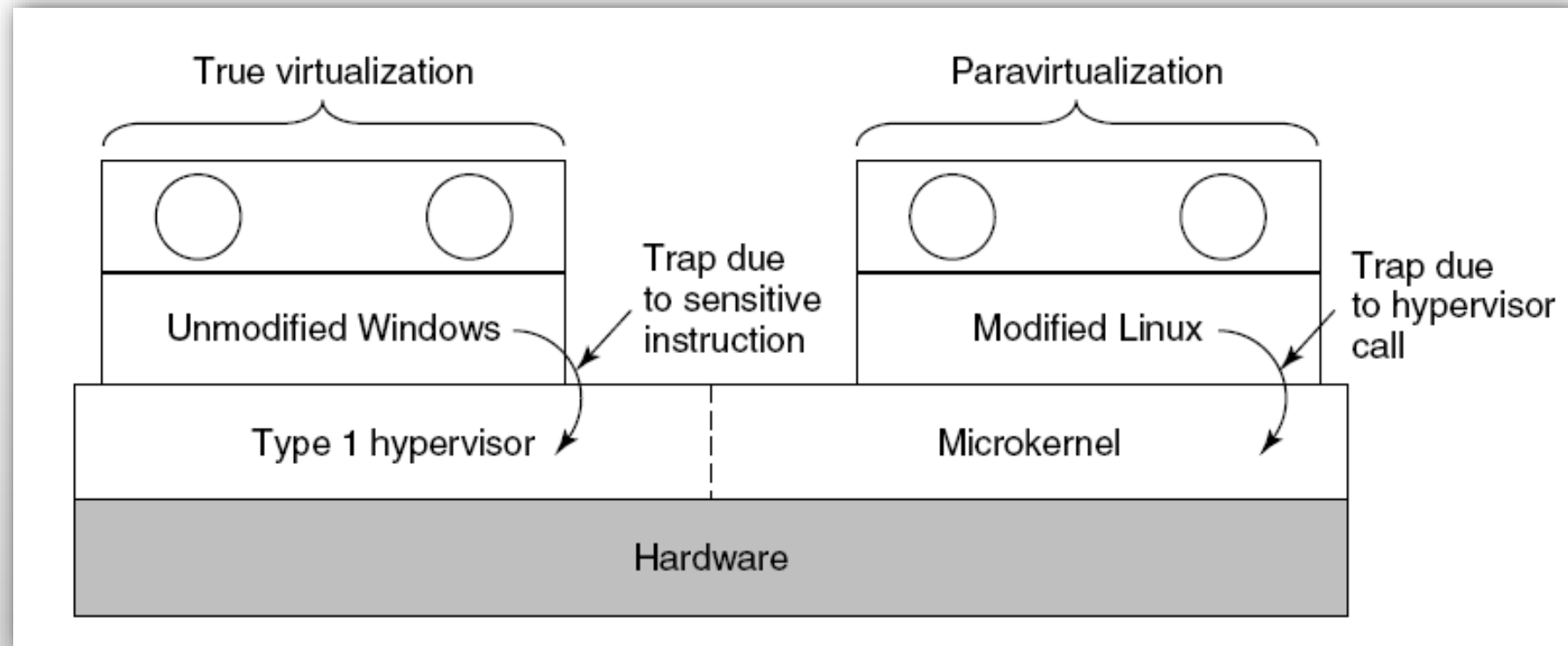
- Sanal makinedeki işletim sistemi çekirdeğe yönelik bir komut yürüttüğünde, sanallaştırma teknolojisi mevcutsa hiper yöneticiye iletir (trap).





# Yarı Sanallaştırma (paravirtualization)

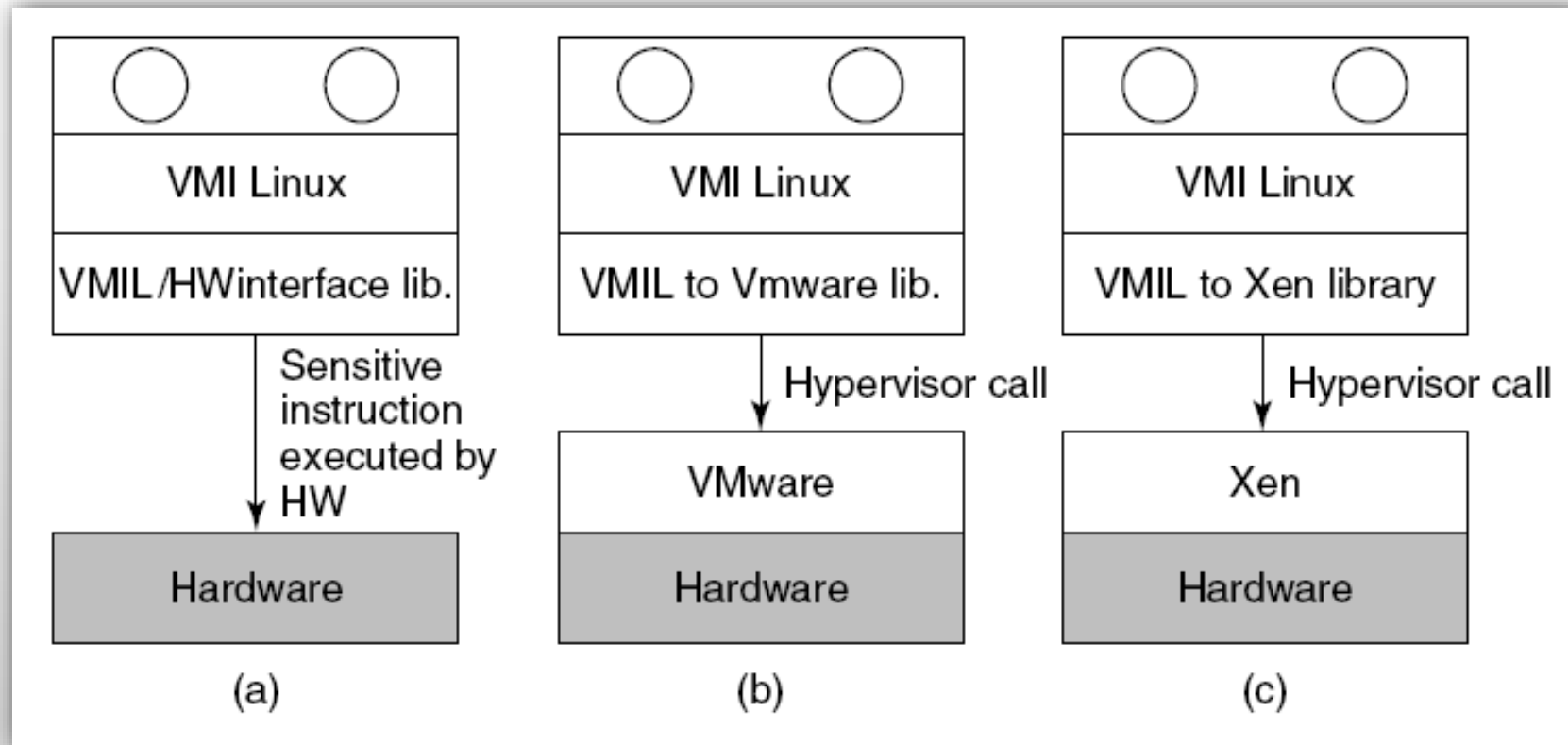
- Hem gerçek sanallaştırmayı hem de sanallaştırmayı destekleyen bir hipervizör.





# Yarı Sanallaştırma (paravirtualization)

- (a) çıplak donanım (b) VMware (c) Xen üzerinde çalışan VMI Linux.





# Çoklu Bilgisayar

- Bir ağ ile birbirine bağlı bağımsız bilgisayardan oluşan bir sistem.
- Her bilgisayar kendi işletim sistemine sahip ve bağımsız olarak çalışır.
- Bilgisayarlar arasında iletişim sınırlıdır.
- Büyük ölçekli simülasyonlar ve paralel bilgi işlem görevleri için uygundur.



# Çoklu İşlemci

- Ortak bir veri yolu veya çapraz çubuk anahtarıyla bağlanan birden fazla işlemciye sahip bir sistem.
- Tüm işlemciler ortak bir hafızayı paylaşır ve işlemciler arası iletişim yoluyla birbirleriyle iletişim kurar.
- Görevlerin işlemcilere tahsisini kontrol eden tek bir işletim sistemine sahiptir.
- Büyük işlem gücü gerektiren zorlu uygulamalar için uygundur.



# Dağıtık Sistem

- Bir ağ aracılığıyla iletişim kuran ve eylemlerini koordine eden otonom bilgisayarlar topluluğu.
- Her bilgisayar kendi işletim sistemini çalıştırır ve kendi belleğine ve kaynaklarına sahiptir.
- Kaynak tahsisi ve görev dağıtımı, dağıtılmış bir işletim sistemi tarafından yönetilir.
- Yüksek düzeyde esneklik ve ölçeklenebilirlik gerektiren karmaşık ve büyük ölçekli uygulamalar için uygundur.



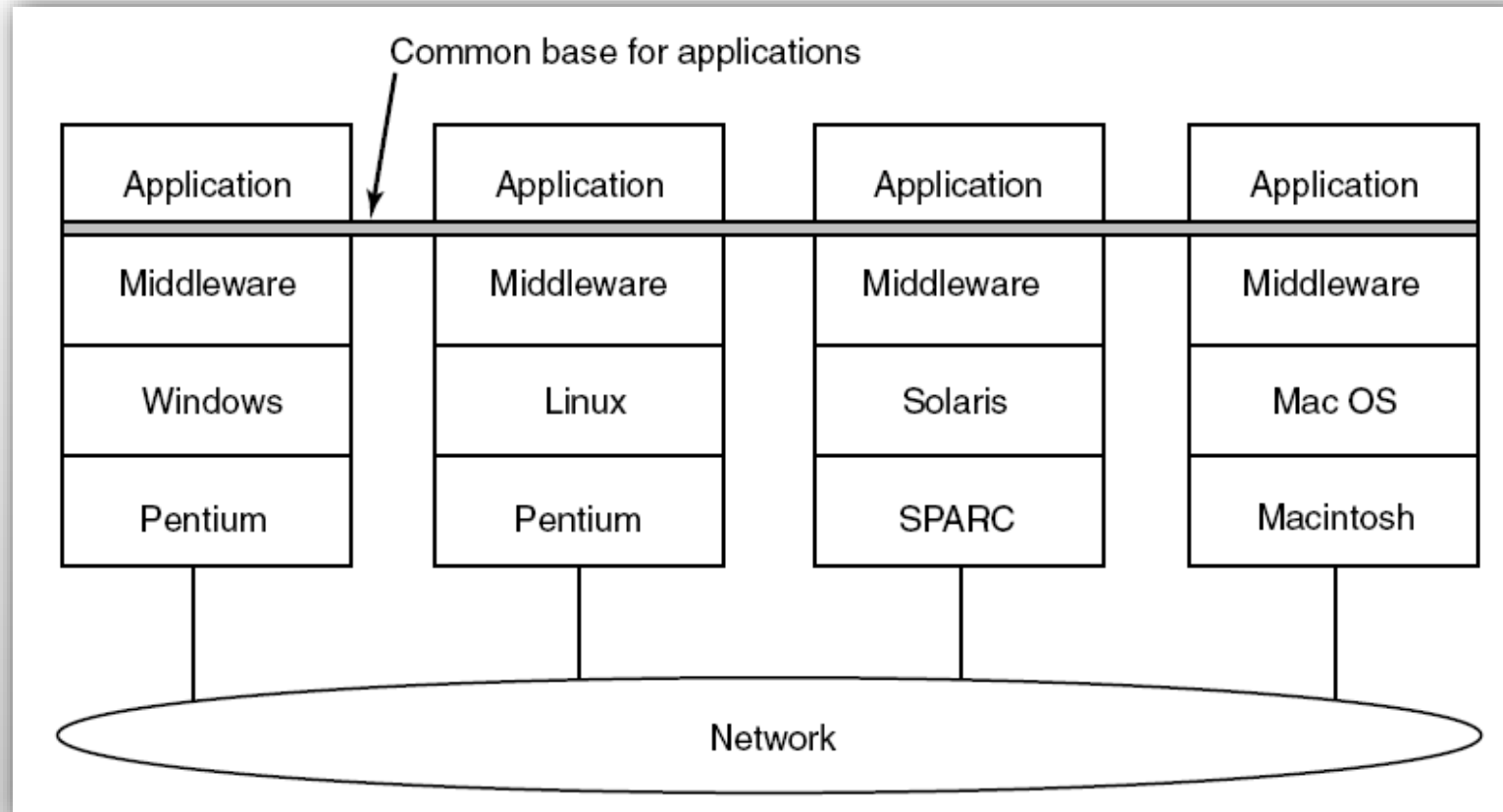
# Karşılaştırma

Öge	Çoklu İşlemci	Çoklu Bilgisayar	Dağıtık Sistem
Düğüm yapılandırması	CPU	CPU, RAM, ağ arabirimi	Tüm Bilgisayar
Düğüm çevre birimleri	Hepsi paylaşıldı	Paylaşılan hariç belki disk	Düğüm başına tam set
Konum	Aynı raf	Aynı oda	Dünya çapında
Düğümler arası iletişim	Paylaşılan RAM	Özel ara bağlantı	Geleneksel ağ
İşletim sistemleri	Bir, paylaşılan	Çoklu, aynı	Hepsi farklı
Dosya sistemleri	Bir, paylaşılan	Bir, paylaşılan	Her düğümün kendine ait
Yönetim	Tek	Tek	Çoklu



# Dağıtık Sistemler

- Dağıtılmış bir sistemde ara yazılımın konumlandırılması.

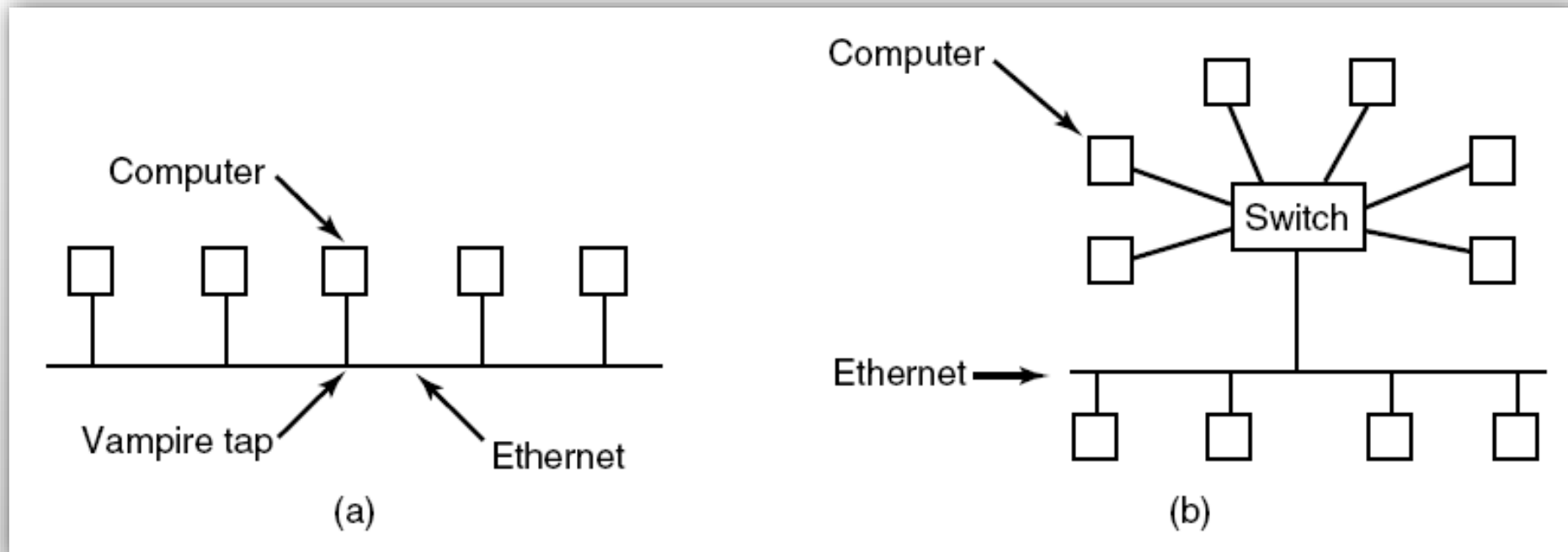






# Ethernet

- (a) Klasik Ethernet. (b) Anahtarlmalı Ethernet.





# Ağ Protokolleri

- **İletim Kontrol Protokolü (TCP):**
  - Hata algılama ve düzeltme yetenekleriyle güvenilir, bağlantı yönelimli veri aktarımı sağlar.
- **Kullanıcı Datagram Protokolü (UDP):**
  - Hata algılama veya düzeltme olmaksızın güvenilir olmayan, bağlantısız veri aktarımı sağlar.
- **İnternet Protokolü (IP):**
  - Ağlar arasında veri iletimi için yönlendirme ve adresleme işlevi sağlar.
- **Basit Ağ Yönetimi Protokolü (SNMP):**
  - Ağ cihazlarını yönetmek ve izlemek için kullanılır.

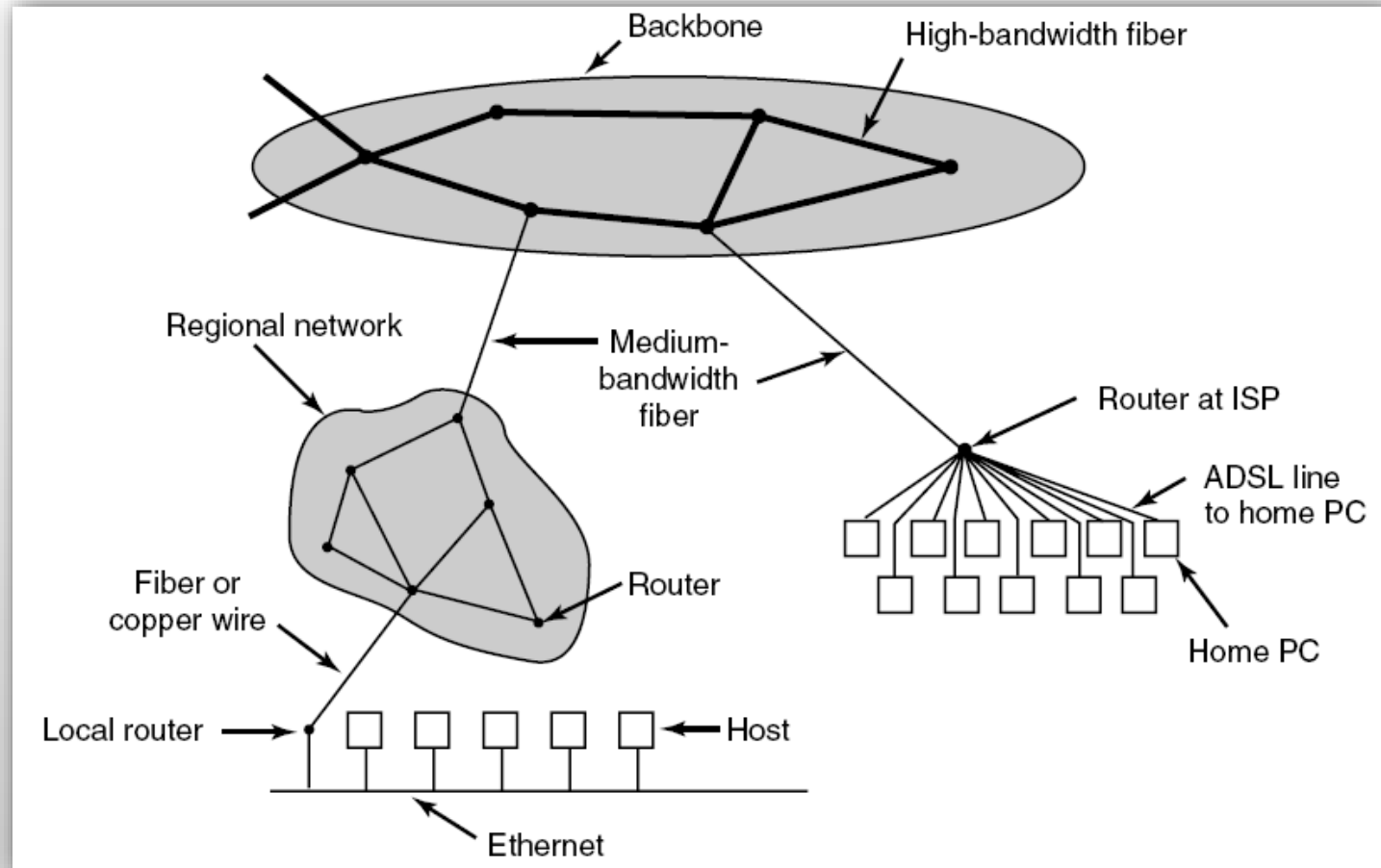


# Ağ Protokolleri

- **Etki Alanı Adı Sistemi (DNS):**
  - Etki alanı adlarını IP adreslerine çözümler.
- **Dosya Aktarım Protokolü (FTP):**
  - İnternet üzerinden dosya aktarımı için kullanılır.
- **Köprü Metni Aktarım Protokolü (HTTP):**
  - World Wide Web'de bilgi alışverişi için kullanılır.
- **Basit Posta Aktarım Protokolü (SMTP):**
  - E-posta mesajları göndermek için kullanılır.



# İnternet Omurgası





# Reliable ve Unreliable Ağ Servisleri

- Güvenilir Ağ Hizmetleri:
  - Verilerin teslimi garanti edilir.
  - Alındı onayı göndericiye geri gönderilir.
  - TCP gibi güvenilir taşıma protokollerini kullanır.
- Güvenilir Olmayan Ağ Hizmetleri:
  - Verilerin teslimi konusunda garanti yoktur.
  - Alındı makbuzu geri gönderilmez.
  - UDP gibi güvenilmez aktarım protokolleri kullanır.



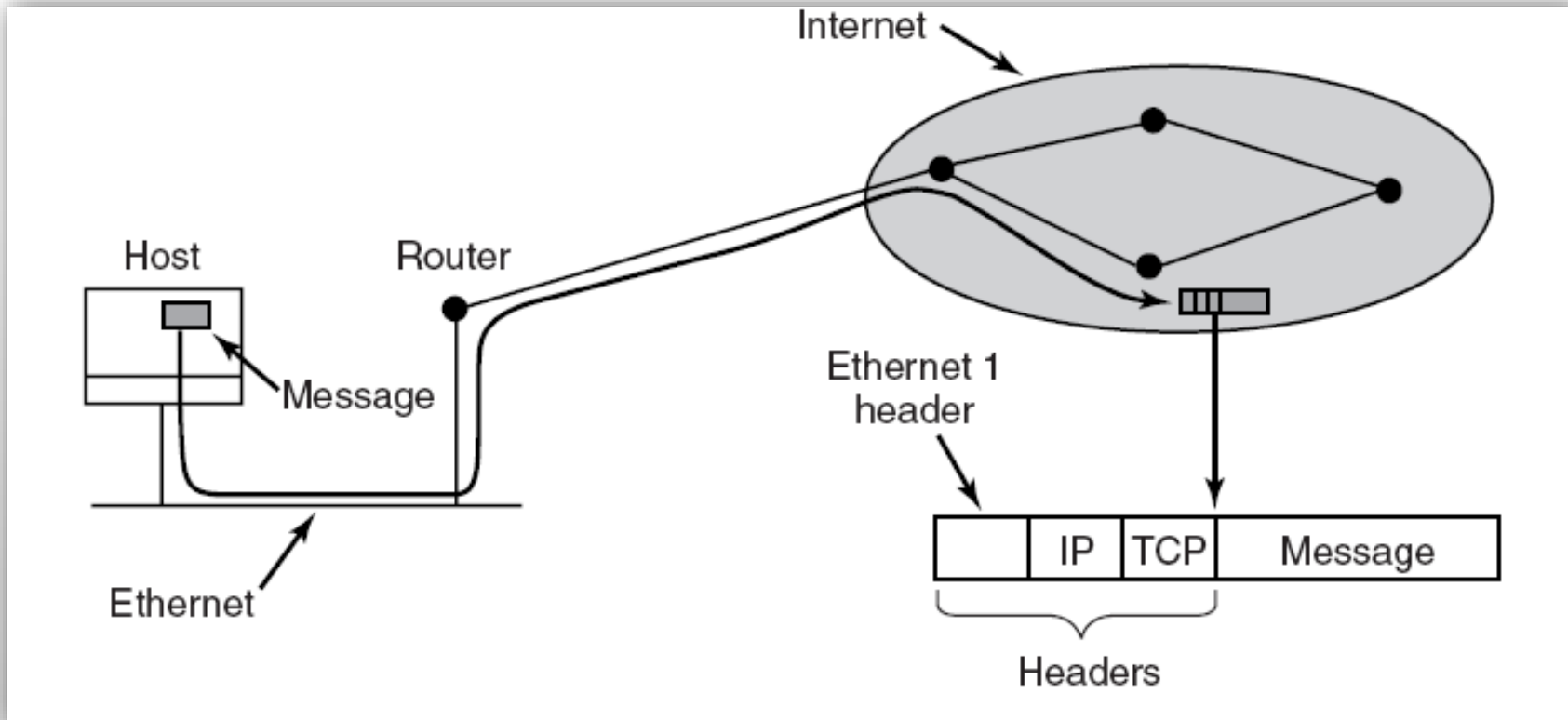
# Connection Oriented ve Connectionless

- Bağlantı Odaklı Hizmetler: İki düğüm arasında güvenilir bir bağlantı kurar
  - Akış kontrolü, hata düzeltme ve yeniden iletim mekanizmaları var
  - Garantili veri teslimatı, veri bütünlüğü
  - Örnekler: TCP, SCTP
- Bağlantısız Hizmetler: Kalıcı bir bağlantı kurmaz
  - Teslimatlarını onaylamadan paketleri gönder
  - Bağlantı kurma ve sonlandırma maliyeti yok
  - Daha yüksek verim, düşük gecikme süresi
  - Örnekler: UDP, IP



# Ağ Protokolleri

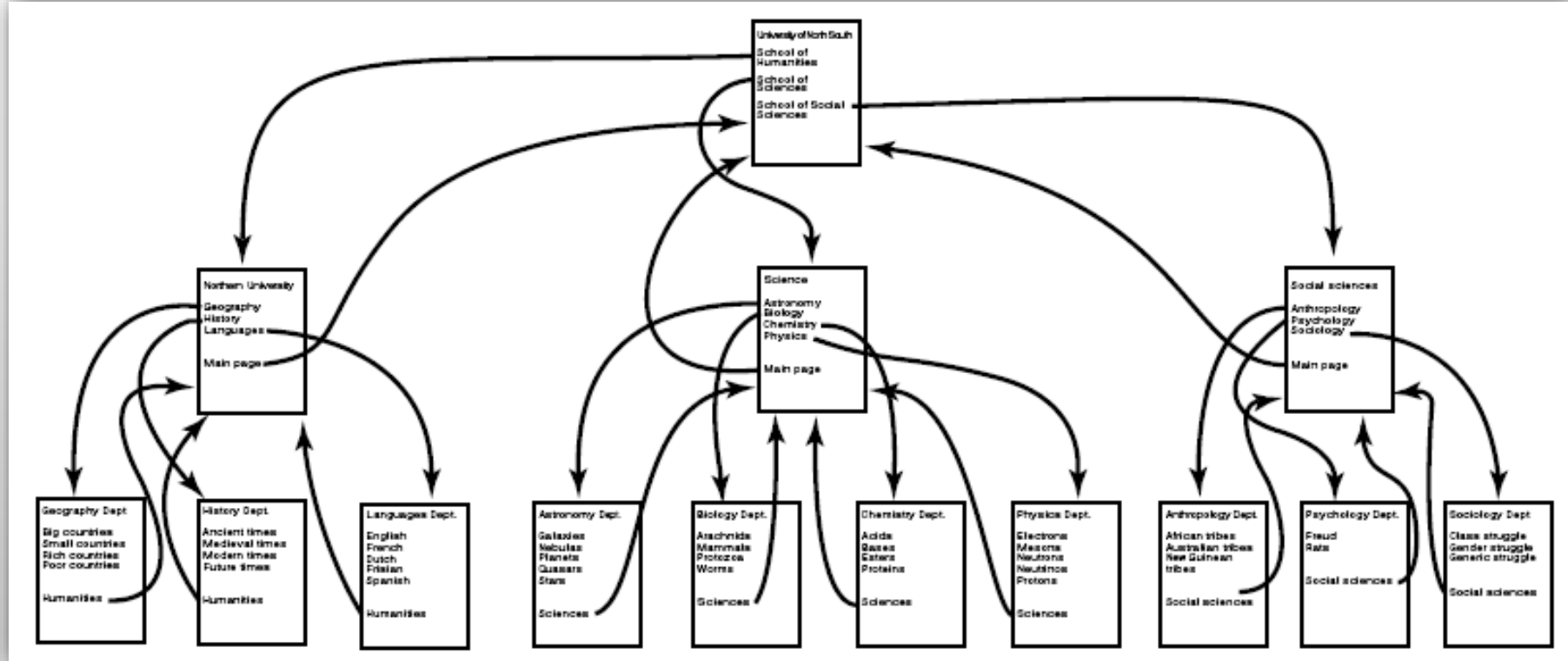
- Paket başlıklarının birikmesi.





# Belge Tabanlı Ara Yazılım

- Web, belgelerden oluşan yönlendirilmiş büyük bir çizgedir.







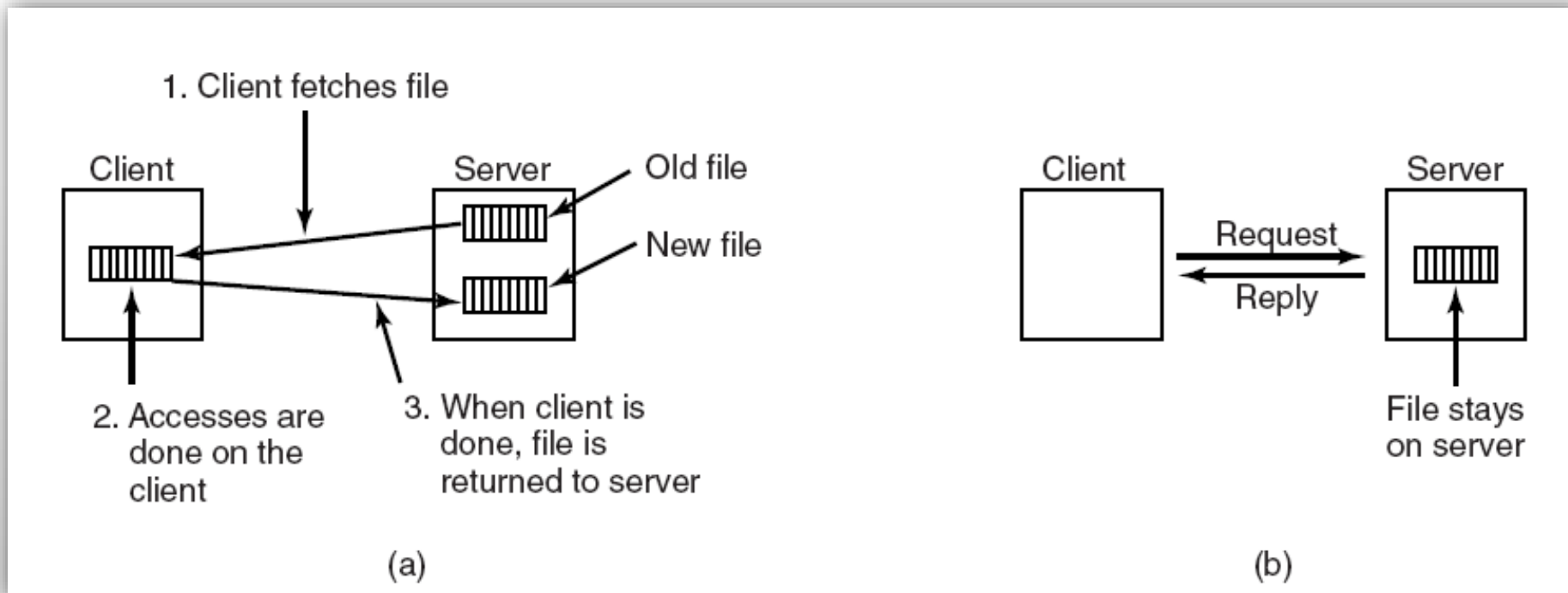
# Tarayıcı Sayfayı Nasıl Getirir

- Tarayıcı DNS'den `www.xxx.org`'un IP adresini ister.
- DNS, `XX.XXX.XX.XXX` ile yanıt verir.
- Tarayıcı, `XX.XXX.XX.XXX`'de 80 nolu porta bir TCP bağlantısı kurar.
- Daha sonra `xxx.html` dosyasını isteyen bir istek gönderir.
- `www.xxx.org` sunucusu `xxx.html` dosyasını gönderir.
- TCP bağlantısı serbest bırakılır.
- Tarayıcı `xxx.html` içindeki tüm metni görüntüler.
- Tarayıcı `xxx.html` içindeki tüm resimleri alır ve görüntüler.



# Dosya Sistemi Tabanlı Ara Yazılım

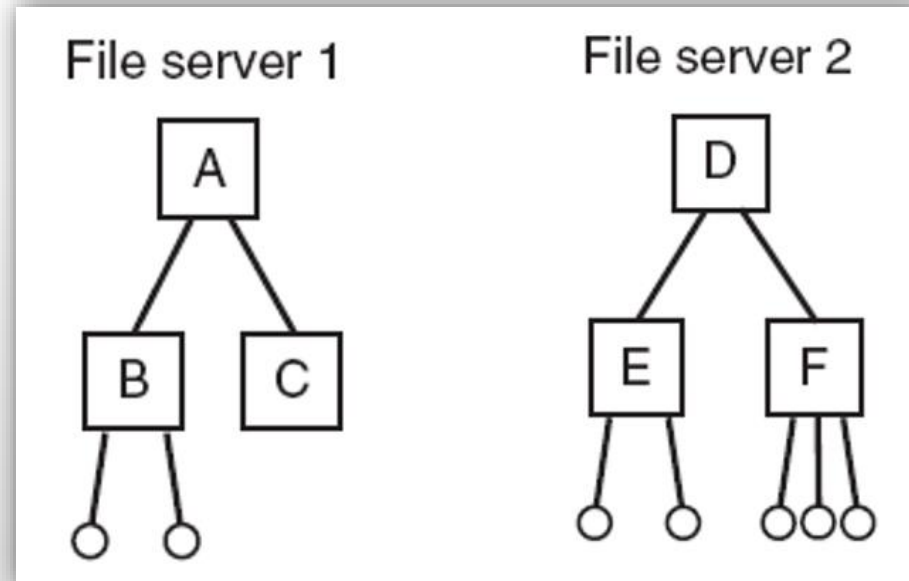
- (a) Yükleme/indirme modeli. (b) Uzaktan erişim modeli.





# Dizin Hiyerarşisi

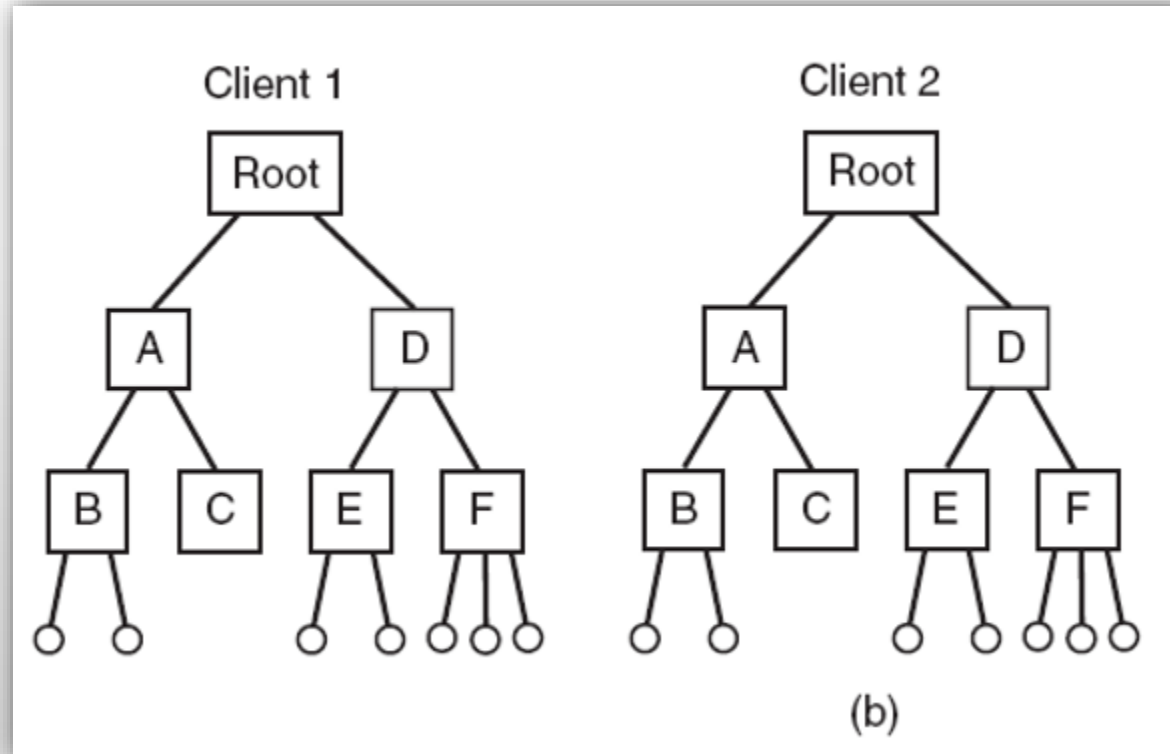
- (a) İki dosya sunucusu. Kareler dizinlerdir ve daireler dosyalardır.





# Dizin Hiyerarşisi

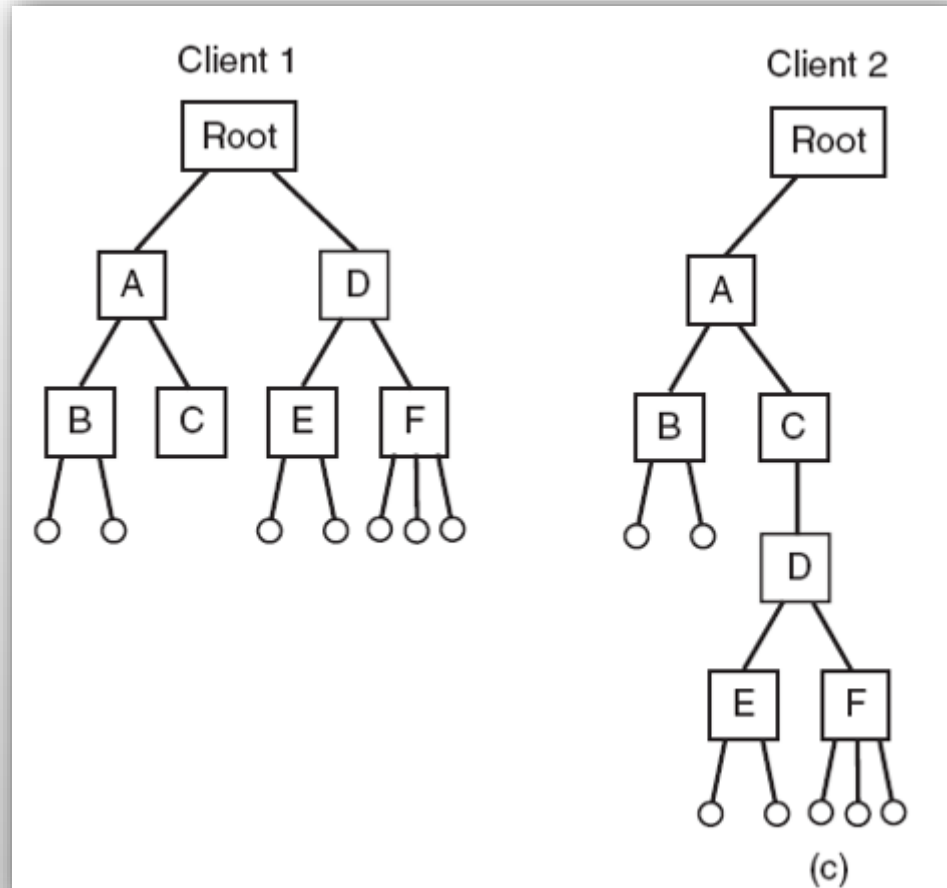
- (b) Tüm istemcilerin aynı dosya sistemi görünümüne sahip olduğu bir sistem.





# Dizin Hiyerarşisi

- (c) Farklı istemcilerin farklı dosya sistemi görünümüne sahip olabildiği bir sistem.





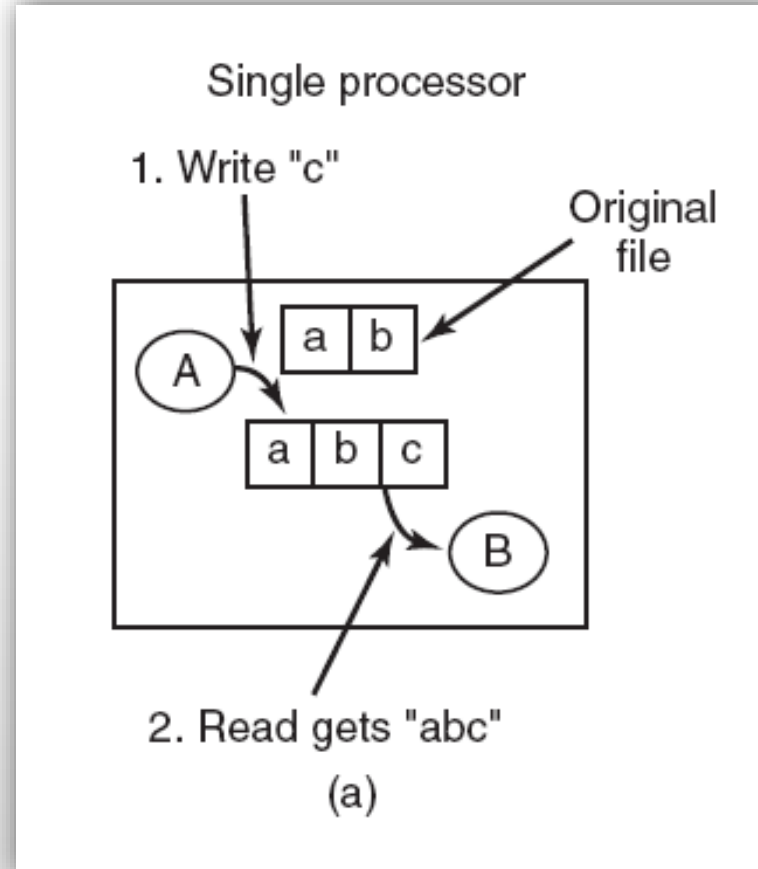
# Adlandırma Şeffaflığı

- Dağıtılmış sistemde dosya/dizin adlandırmaya yönelik üç yaygın yaklaşım:
  - Makine + yol adlandırma, /makine/yol veya makine:yol gibi
  - Uzak dosya sistemlerini yerel dosya hiyerarşisine bağlama
  - Tüm makinelerde aynı görünen tek bir ad alanı. (name space)



# Dosya Paylaşımının Semantiği

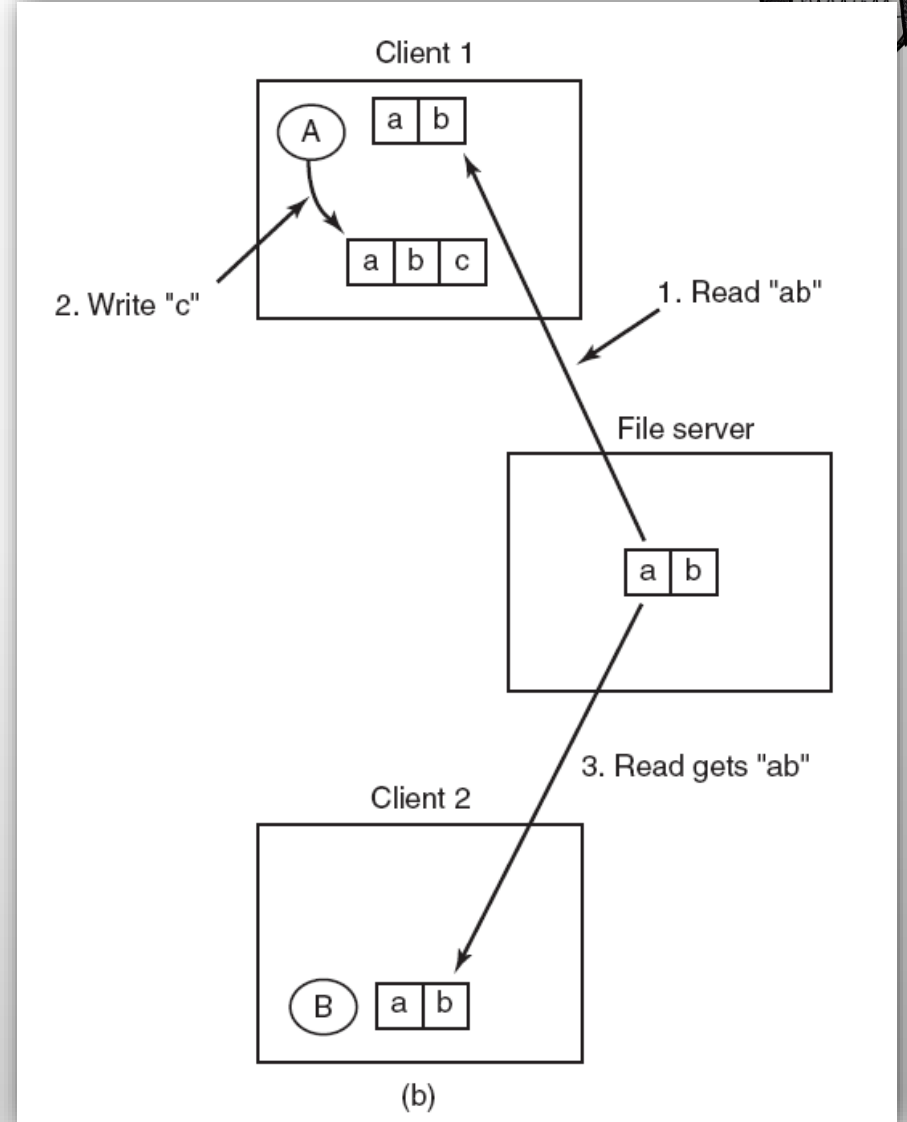
- (a) Sıralı tutarlılık.





# Dosya Paylaşımının Semantiği

- (b) Önbelleğe alma özelliğine sahip dağıtık bir sistemde, bir dosyayı okumak geçersiz bir değer döndürebilir.







# CORBA (Common Object Request Broker Architecture)

- Dağıtık yazılım bileşenleri arasında birlikte çalışabilirliği sağlayan bir ara katman teknolojisidir.
- **Nesne yönelimli:** CORBA, nesneler, sınıflar ve arayüzler gibi nesne yönelimli kavramlara dayalıdır.
- **Platformdan bağımsız:** nesne ve arabirimleri tanımlamak için farklı platformlarda ve işletim sistemlerinde çalışan bileşenler arasında iletişime izin veren standart bir arabirim tanımlama dili (IDL) kullanır.
- **Ağ saydamlığı:** CORBA, nesnelerin bir ağ üzerindeki konumlarından bağımsız olarak birbirleriyle iletişim kurmasına izin vererek, ağ iletişimi üzerinden bir soyutlama düzeyi sağlar.



# CORBA (Common Object Request Broker Architecture)

- **Nesne İstek Aracısı (ORB):** ORB, istemci ve sunucu nesneleri arasında aracı görevi görür ve aralarındaki iletişimi yönetir.
- **Arayüz Tanımlama Dili (IDL):** nesneleri, arayüzleri ve yöntemleri tanımlamak için kullanılır ve istemcilerin ve sunucuların birbirleriyle iletişim kurmasına olanak tanır.
- **Dinamik Çağırma Arayüzü (DII):** istemcilerin nesnelerin yöntemlerini dinamik olarak keşfetmesine ve çağırmasına olanak tanır.
- **Statik Çağırma Arabirimi (SII):** DII'nin önceden derlenmiş bir sürümüdür, performansın artmasına neden olabilir.



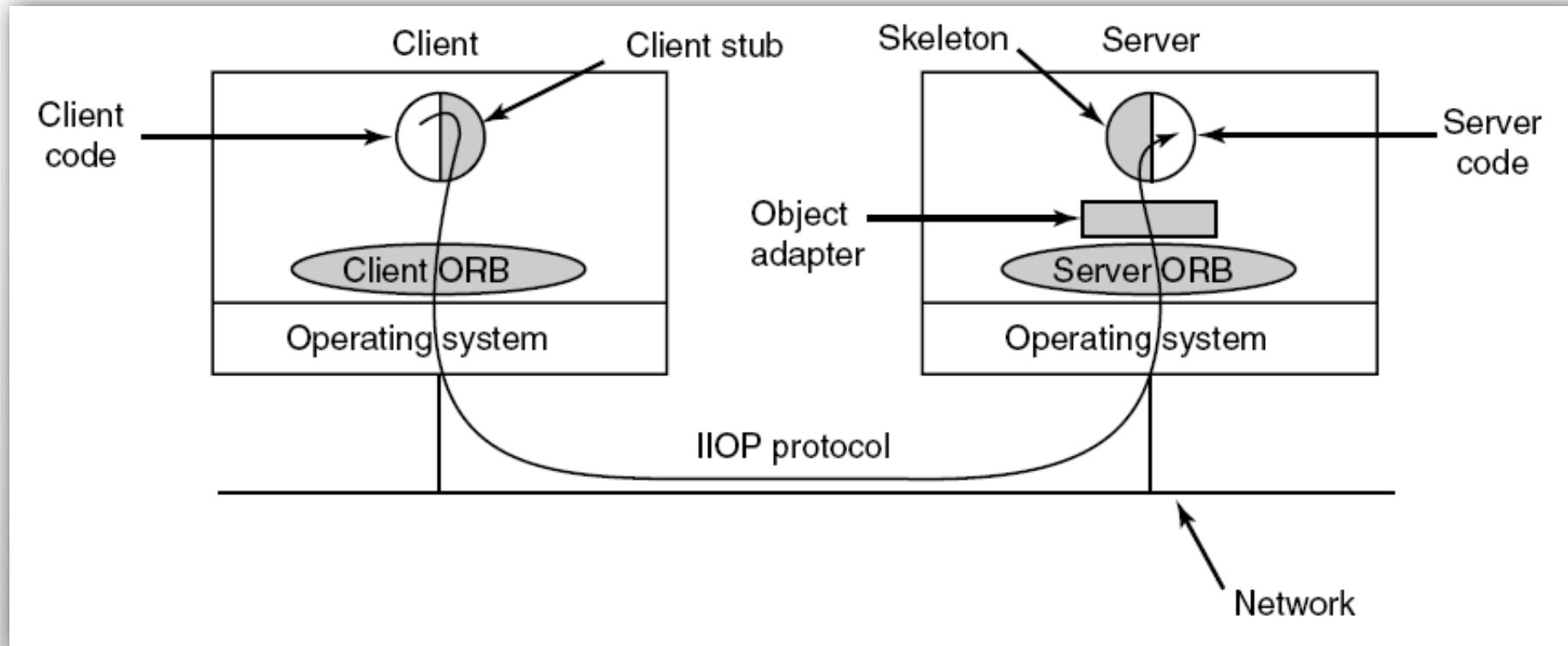
# CORBA (Common Object Request Broker Architecture)

- Birlikte çalışabilirlik, ölçeklenebilirlik ve yeniden kullanılabilirlik sağlar.
- CORBA kullanan nesneler arasındaki iletişimin ek yükü, tek bir işlemdeki doğrudan yöntem çağrılarına kıyasla yavaş performansa neden olabilir.
- CORBA mimarisi, geliştiriciler için dik bir öğrenme eğrisi ile karmaşık olabilir.
- Birçok faydasına rağmen, CORBA geniş çapta benimsenmemiştir ve artık aktif olarak sürdürülmemektedir.



# CORBA (Common Object Request Broker Architecture)

- CORBA'ya dayalı dağıtık sistemin ana unsurları. CORBA parçaları gri renkte.





# Linda: Coordination-Based Middleware

- Dağıtık bilgi işlem için koordinasyon tabanlı bir ara yazılım sistemidir.
- Verilerin içeriğine göre depolanabileceği, kaldırılabilirliği veya sorgulanabileceği bir "demet alanı" kavramını kullanır.
- Demetler, dağıtılmış bir veritabanı gibi davranan bir veri yapısındadır.
- Linda, bu demetleri kullanarak süreçler arasında senkronizasyon ve iletişim hizmetleri sağlar.
- Linda, paralel işleme, bilimsel simülasyonlar ve işbirliğine dayalı çalışma ortamları gibi süreçlerin koordinasyon ve iletişim gerektirdiği uygulamalarda kullanılır.



# Linda: Coordination-Based Middleware

- İletişim ve senkronizasyon için bir sistem
- Bağımsız süreçler, soyut bir demet alanı (tuple) aracılığıyla iletişim kurar
- Tuple, her biri temel dil tarafından desteklenen bir tip değer olan bir veya daha fazla alandan oluşan bir yapıdır.

```
("abc", 2, 5)  
("matrix-1", 1, 6, 3.14)  
("family", "is-sister", "Stephany", "Roberta")
```



# Eşleşen Demetler (tuples)

- Aşağıdaki üç koşul karşılanırsa bir eşleşme oluşur
  - Şablon ve demet aynı sayıda alana sahiptir.
  - Karşılık gelen alanların tipleri eşittir.
  - Şablondaki her sabit veya değişken, demet alanıyla eşleşir.



# Publish-Subscribe Dağıtık Sistemler

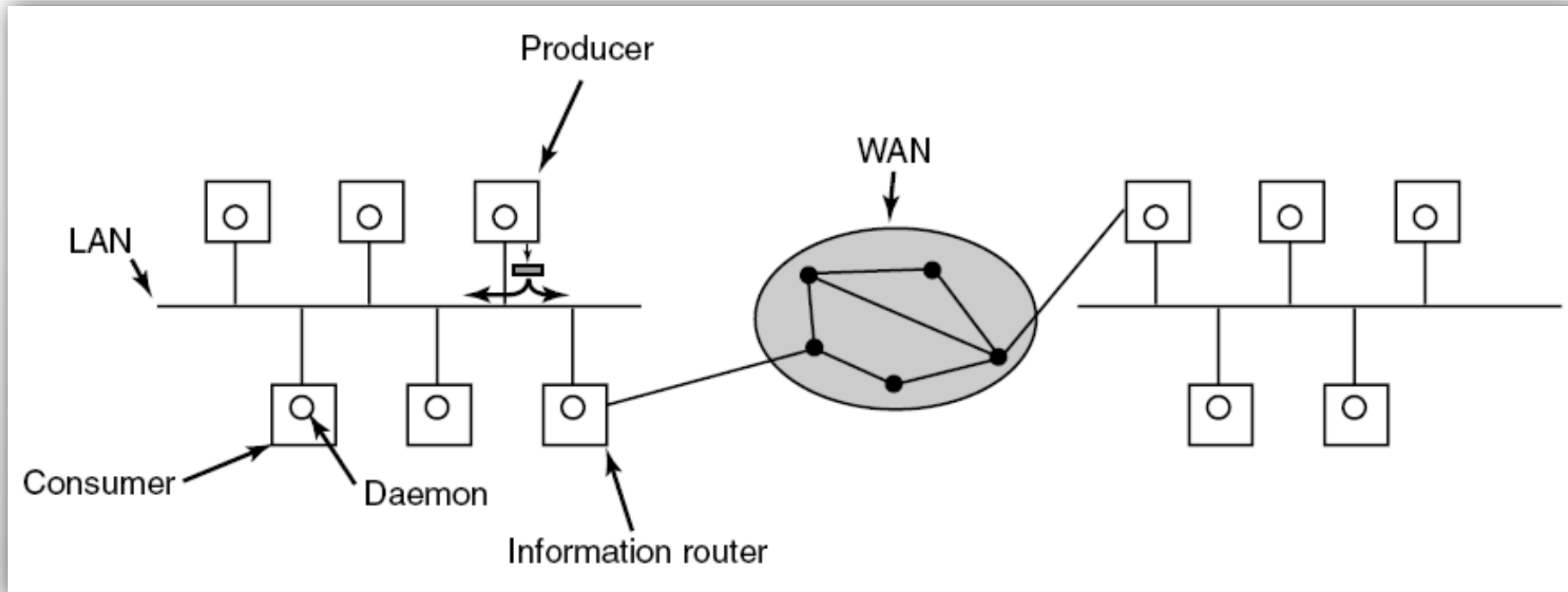
- Yayınla-Abone Ol (Pub-Sub), üreticilerin bir konuya mesaj gönderdiği ve tüketicilerin ilgilendikleri konulara abone oldukları bir mesajlaşma modelidir.
- Süreçlerin kimliklerini veya iletişim protokollerinin özelliklerini bilmeden iletişim kurmalarına olanak tanır.
- Aracı, üreticilerden gelen mesajları almaktan ve aboneliklerine göre abonelere dağıtmaktan sorumludur.
- Göndericileri ve alıcıları birbirinden ayırarak sistemdeki değişikliklerin etkisini azaltır.
- Aboneler dinamik olarak eklenebilir ve kaldırılabilir. (ölçeklenebilir)
- Yeni konular eklenebilir ve bunlara abone olunabilir. (esnek)
- Mesajlar abonelere teslim edilene kadar aracıda saklanır. (güvenilir)





# Yayınla/Abone Ol

- Yayınlama/abone olma mimarisi. (publish/subscribe)



# Jini



- Jini, Sun Microsystems tarafından geliştirilmiştir.
- Cihazların ve hizmetlerin önceden yapılandırma olmaksızın bir ağ üzerinden dinamik olarak birbirlerini keşfetmesine, birleştirmesine ve kullanmasına izin verir.
- Birlikte çalışabilirliği sağlamak için iletişim protokolü olarak Java RMI'yi (Uzaktan Yöntem Çağırma) kullanır.
- Dağıtık sistemler için esnek, ölçeklenebilir ve güvenli bir altyapı sağlar.
- Jini, genel sistemi etkilemeden cihazların ve hizmetlerin eklenebildiği, kaldırılabilirdiği ve yeniden yapılandırılabilirdiği, yüksek düzeyde merkezi olmayan ve dinamik uygulamaların geliştirilmesini sağlar.



# Jini

- Jini istemcileri ve hizmetleri, JavaSpaces kullanarak iletişim kurar ve senkronize olur.
- Bir JavaSpace'te tanımlanan yöntemler:
  - **Yaz:** JavaSpace'e yeni bir girdi koyar.
  - **Oku:** JavaSpace'ten şablonla eşleşen bir girdiyi kopyalar.
  - **Al:** şablonla eşleşen bir girdiyi kopyalar ve JavaSpace'ten kaldırır.
  - **Bildir:** Eşleşen bir girdi yazıldığında çağıranı bilgilendirir.



SON