

İşletim Sistemlerine Giriş

Girdi/Çıktı
(I/O)

Girdi/Çıktı (I/O)

İşletim sisteminin temel fonksiyonlarından biride bilgisayardaki tüm I/O aygıtlarını kontrol etmesidir.

Aygıtlara komutlar göndermeli, kesmeleri yakalamalı ve hatalar ile ilgilenmelidir.

Donanım ile sistemin kalan kısmı arasında bir arayüz oluşturmaktadır.

Kullanılan aygıtlar için arayüz mümkün olduğu kadar aynı olmalıdır.

I/O aygıtları kullanım için gerekli olan kod işletim sisteminin büyük bir kısmını oluşturur.

Girdi/Çıktı aygıtları

I/O aygıtları iki kategoriye ayrılabilir: **blok aygıtları** ve **karakter aygıtları**.

Bir blok aygıtı(block device) sabit boyutlu bloklar şeklinde veri saklar. Her bloğun kendi adresi vardır. Blok boyutları 512 byte-32768 byte arasında değişir. Farklı bloklar birbirinden bağımsız olarak okunup yazılabilir.

Diskler en temel blok aygıtıdır. Okuma kafasının bulunduğu kolun nerede olduğu önemli değildir. İstenilen silindir aranabilir ve okuma kafası istenilen verinin bulunduğu yere konumlandırılabilir.

Girdi/Çıktı aygıtları

Karakter aygıtları(character device) karakter akımlarını (stream) kabul eder yada dağıtırlar.

Adreslenemezler ve bu aygıtlarda arama işlemi gerçekleştirilemez.

Yazıcılar, ağ aygıtları, mouse,... bu tip aygıtlardır.

İşletim sisteminin dosya sistemi bu aygıtların hepsi ile soyut aygıtlar olarak ilgilenir. Aygıtların arka planında aygıtı özel olan olaylar ile ilgilenmezler.

Girdi/Çıktı aygıtları farklı hızlardadır

Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
Fast Ethernet	12.5 MB/sec
ISA bus	16.7 MB/sec
EIDE (ATA-2) disk	16.7 MB/sec
FireWire (IEEE 1394)	50 MB/sec
XGA Monitor	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec
Sun Gigaplane XB backplane	20 GB/sec

Aygıt Kontrolcöleri (Device Controller)

I/O aygıtları, mekanik ve elektronik bileşenlerden meydana gelir.

Elektronik bileşene aygıt **kontrolcüsü veya uyumlayıcı** (adapter) adı verilir.

IEEE,ISO,ANSI standartları ile aynı tip aygıtların uymaları gerekli olan kurallar belirlenmiştir. Aygıtların kontrolcöleri de bu standartlar temelinde hazırlanırlar.

Temel görevleri; veri okumak, yazmak, hata kontrolü yapmaktır. Okunan verilerin belleğe kopyalanması işleminide gerçekleştirirler.

Bellek Eşlemeli G/Ç (Memory Mapped I/O)

Her kontrolcü içerisinde belirli sayıda yazmaç vardır. Bu yazmaçlar işlemci ile iletişimde kullanılırlar.

İşlemci bu yazmaçlara komut ve bilgi yazarak, kontrolcünün veri getirmesini, yazmasını, aygıtı kapatıp açmasını,... gibi işlemleri gerçekleştirir. Ayrıca aygıtın mevcut durumu hakkındaki bilgilerde bu yazmaçlardan elde edilir.

Kontrol yazmaçlarından başka aygıtlarda veri tamponları (data buffers) bulunmaktadır. Bu tamponlardaki verileri işletim sistemi okuyabilir ve onlara veri yazabilir.

Örneğin; ekran kartlarında bulunan tampon bellek alanına işletim sistemi veri yazar, bu veri görüntülenir.

Bellek Eşlemeli G/Ç (Memory Mapped I/O)

İşlemcinin kontrolcünün yazmaçlarını okuyup buralara veri yazabilmesi için iki seçeneği vardır.

Birinci olarak, her kontrol yazmacının bir port(kapı) numarası vardır. I/O işlemleri için özel assembly komutları kullanılır.

IN REG,PORT
OUT PORT,REG

gibi.

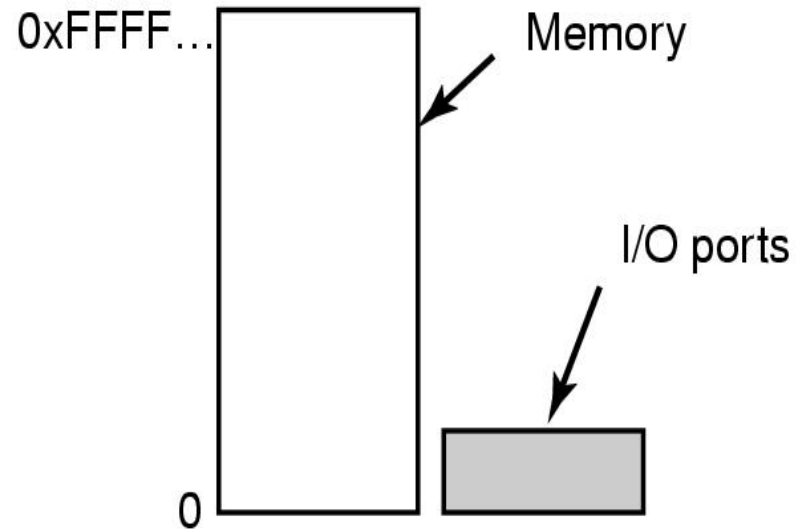
Bu sistemde bellek ve I/O için kullanılan adres uzayı farklıdır.

Bellek Eşlemeli G/Ç (Memory Mapped I/O)

Two address

One address space

Two address spaces



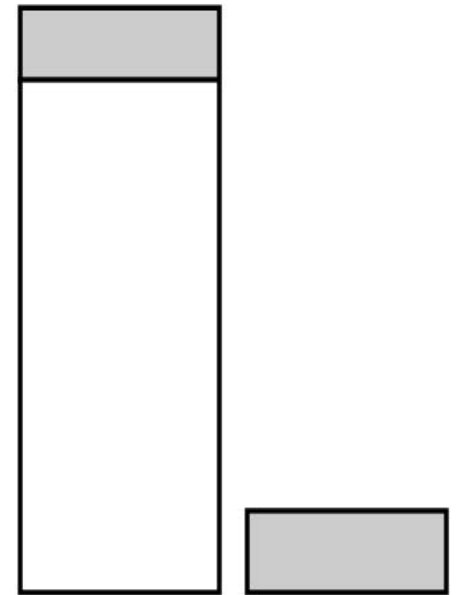
(a)

Birinci yöntem



(b)

İkinci yöntem



(c)

Bellek Eşlemeli G/Ç (Memory Mapped I/O)

İkinci yaklaşımda, tüm kontrol yazmaçları bellekteki belli bir alan ile eşleştirilirler.

Her kontrol yazmacı belirli bir adrese atanır.

Bu sisteme bellek eşlemeli G/Ç (memory mapped I/O) denilir.

Bu şekilde işletim sistemi kullanıcı süreçleri ile I/O işlemleri arasındaki korumayı kolaylıkla gerçekleştirir. yazmaçların bulunduğu bellek alanını kullanıcının sanal bellek uzayına koymayarak kullanıcının bu alanları direkt kullanmasına engel olur.

Doğrudan Bellek Erişimi (DMA-Direct Memory Access)

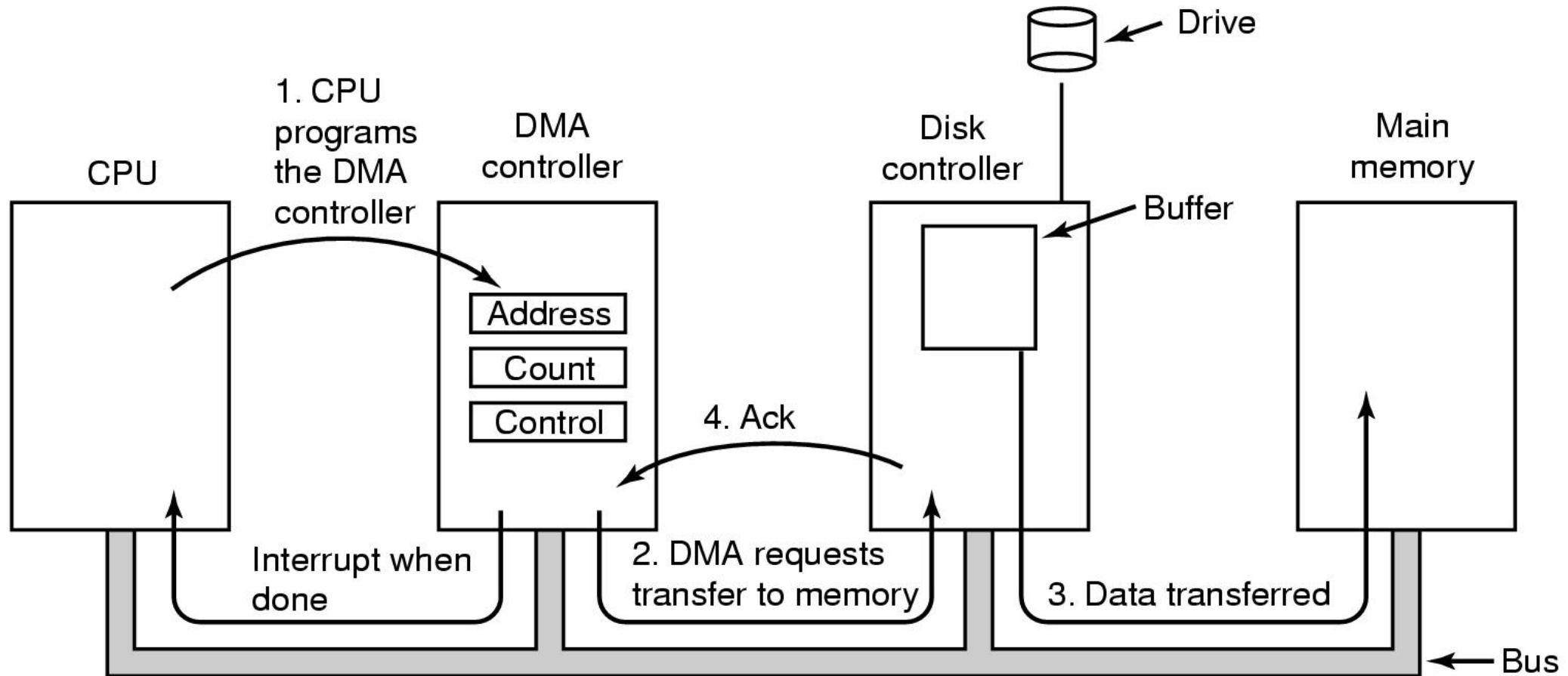
İşlemci doğrudan aygıt sürücüsünü kullanırsa her seferinde I/O verisini alacak ve örneğin belleğe koyacaktır. İşlemci bu işlemlerde verinin içeriğini hiç kullanmaz. Taşıma için işlemci zamanı israf edilmiş olur.

Bu durumlarda işlemci eğer var ise, kontrolcü üzerindeki DMA'yı kullanır.

Her aygıt için farklı DMA olabileceği gibi, birden fazla aygıt için ortak bir DMA da olabilir.

DMA sistem yoluna(bus) işlemciden bağımsız olarak erişebilen bir donanımdır.

Doğrudan Bellek Erişimi (DMA-Direct Memory Access)



Doğrudan Bellek Erişimi (DMA-Direct Memory Access)

DMA'da işlemci tarafından okunup/ değiştirilebilen birkaç yazmaç vardır. Bir bellek yazmacı, bir sayaç yazmacı, bir veya daha fazla kontrol yazmacı vardır.

DMA olmadan yapılacak disk okuma işlemi şu şekilde gerçekleşir;

Kontrolcü aygıttan verileri okur, okuduğu verileri bit bit kendi içindeki tampon(buffer)'a koyar. Okunan veride hata olup olmadığını kontrol eder. Veri depolandıktan sonra işlemciye bir kesme(interrupt) gönderilir.

İşletim sistemi bu kesme sonucunda kontrolcünün iç yapısından bu veriyi okur ve belleğe yazar.

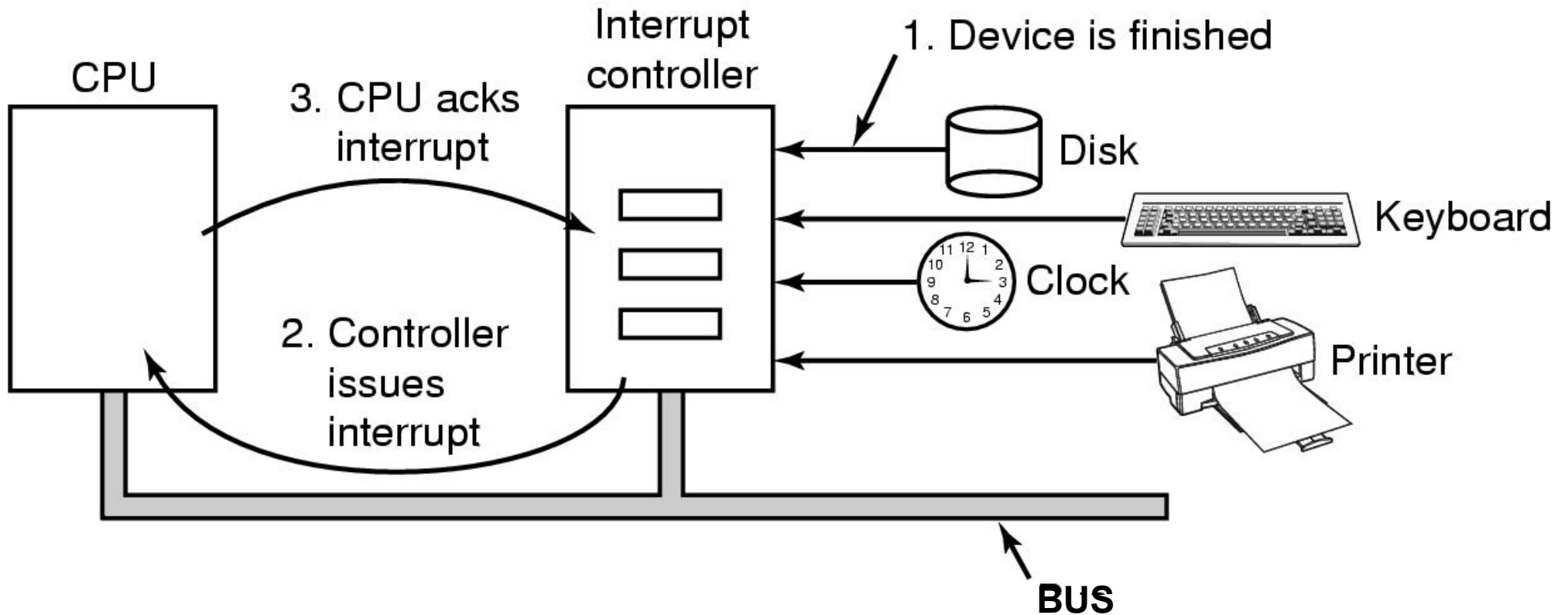
Doğrudan Bellek Erişimi (DMA-Direct Memory Access)

DMA kullanıldığında durum biraz farklıdır; işlemci DMA yı programlar. Bu program, DMA ya nereden hangi veriyi alacağını, nereye koyacağını belirtir.

İşlemci DMA ile birlikte okumak istediği aygıta da(örneğin disk) gerekli bilgileri gönderir, böylece aygıt okuduğu verileri iç tamponuna koyar.

DMA işlemci gibi davranarak aygıta (örn: disk kontrolcüsüne) okuma (read) komutunu gönderir. Disk kontrolcüsü veriyi belleğe koyar ve DMA yı bilgilendirir. DMA kontrolcüsü sayacı bir azaltır ve sonraki bellek bilgisini disk kontrolcüsüne gönderir. Sayaç=0 olduğunda DMA işlemciye kesme(interrupt) gönderir.

Kesmeler (Interrupts)



Kesmeler (Interrupts)

Bir I/O aygıtı kendisine verilen görevi bitirince bir kesme (interrupt) oluşturur. Yola (Bus) bir sinyal koyar. Interrupt kontrolcüsü bu sinyali görür ve yapması gerekli olan işi yapar.

Kesme kontrolcüsünde başka bir iş yoksa bu kesme hemen işlenir. Var ise bir kuyruğa konulur ve daha sonra işlenir. Yüksek öncelikli kesmeler öncelikle işlenirler.

Kesme kontrolcüsüne gelen bu kesme işlemciye gönderilir ve işlemcinin kabul (ACK) yani alındı bilgisi gelene kadar bu kesme sinyali işlemciye gönderilmeye devam eder.

Kesmeler (Interrupts)

İşlemciye gönderilen kesme numarasını işlemci kesme vektörü (interrupt vector) adı verilen tablodan kontrol eder ve bu kesmede gerçekleştirilecek olan kodların bulunduğu bir bellek bölgesinin (prosedür) başlangıç adresini elde eder.

Bu adresi Program Counter(PC) program sayacına koyar, bu şekilde bu prosedür işlenmeye başlar. Prosedür işlenmeye başlayınca ilk iş olarak kesme kontrolcüsü bilgilendirilir. İşlemci kabul mesajını göndermeyi biraz geciktirerek ölümcül kilitlenmeleri engeller.

Aygıt Sürücüler (Device Drivers)

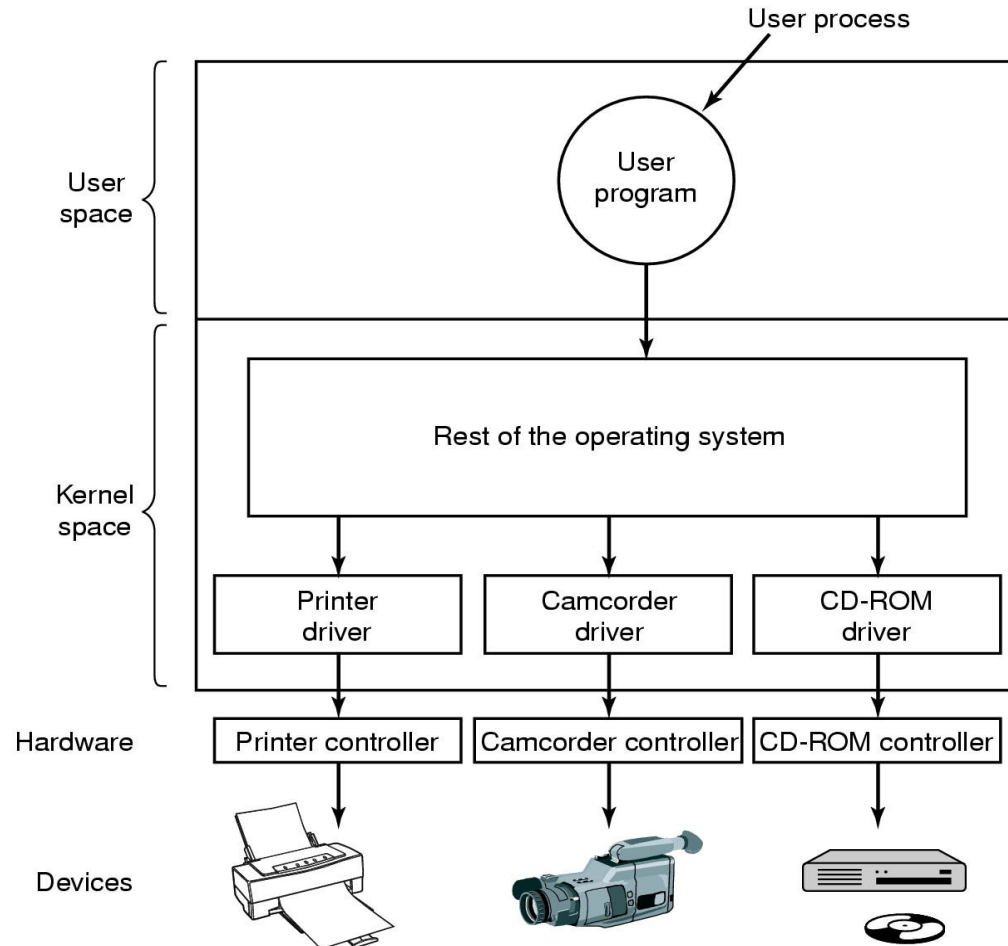
Her aygıt kontrolcüsü üzerinde aygıtı özel yazmaçlar bulunmaktadır. Bu yazmaçların sayısı aygıtı bağı olarak değişmektedir ve belirli aygıtı verilebilen komutlarda çeşitlilik göstermektedir.

Bilgisayara bağı olan her I/O aygıtı kullanabilmek için o aygıtı özel olan kodlar çalıştırılmalıdır. Aygıtı özel çalıştırılan ve aygıtta istediğimiz işleri gerçekleştiren koda **aygıt sürücüsü** denilmektedir.

Aygıt sürücüler genellikle donanımı üreten şirketlerce yazılmaktadır ve donanım ile birlikte verilmektedirler. Her işletim sistemi için aygıt sürücüler yazılır. (yaygın olanlar için)

Aygıt Sürücüler (Device Drivers)

Aygıt sürücülerini direkt çekirdek içerisinde olabileceği gibi kullanıcı uzayında da olabilir. Kullanıcı uzayında bulunan sürücülere sistem çağrıları kullanılarak erişilir.



Aygıt Sürücüler (Device Drivers)

İşletim sistemleri aygıtları blok ve karakter aygıtlar olarak sınıflandırır. İşletim sistemleri blok ve karakter aygıtlarını kullanabilmek için bu aygıtların sürücülerinin sağlaması gerekli olan arayüzü tanımlarlar.

Sürücüler tanımlanmış olan bu arayüzü gerçekleştirmek zorundadır. Bu arayüz içerisinde sürücüde bulunması gerekli olan prosedürler bulunmaktadır.

UNIX işletim sisteminde tüm sürücüler çekirdek içerisindeydi. UNIX işletim sistemi büyük bilgisayar merkezlerinde kullanıldığı için ve kullanılan donanımın değişmemesi nedeniyle bu yaklaşım sorun oluşturmuyordu.

Aygıt Sürücüler (Device Drivers)

UNIX makineye yeni bir donanım eklediğinizde bu donanıma ait olan sürücü işletim sisteminin koduna yönetici (administrator) tarafından eklenir ve işletim sisteminin çekirdeği tekrar derlenirdi. Bu şekilde oluşturulan yeni işletim sistemi çekirdeği kullanılırdı.

Kişisel bilgisayarların gelişmesi ile kullanılan donanım çeşitlendi. UNIX makinelerde kullanılan bu sisteo ihtiyaçlara cevap vermesine rağmen deneyimsiz ve bilgisiz kullanıcılar tarafından yapılamadığı için geçersiz oldu.

Aygıt Sürücüler (Device Drivers)

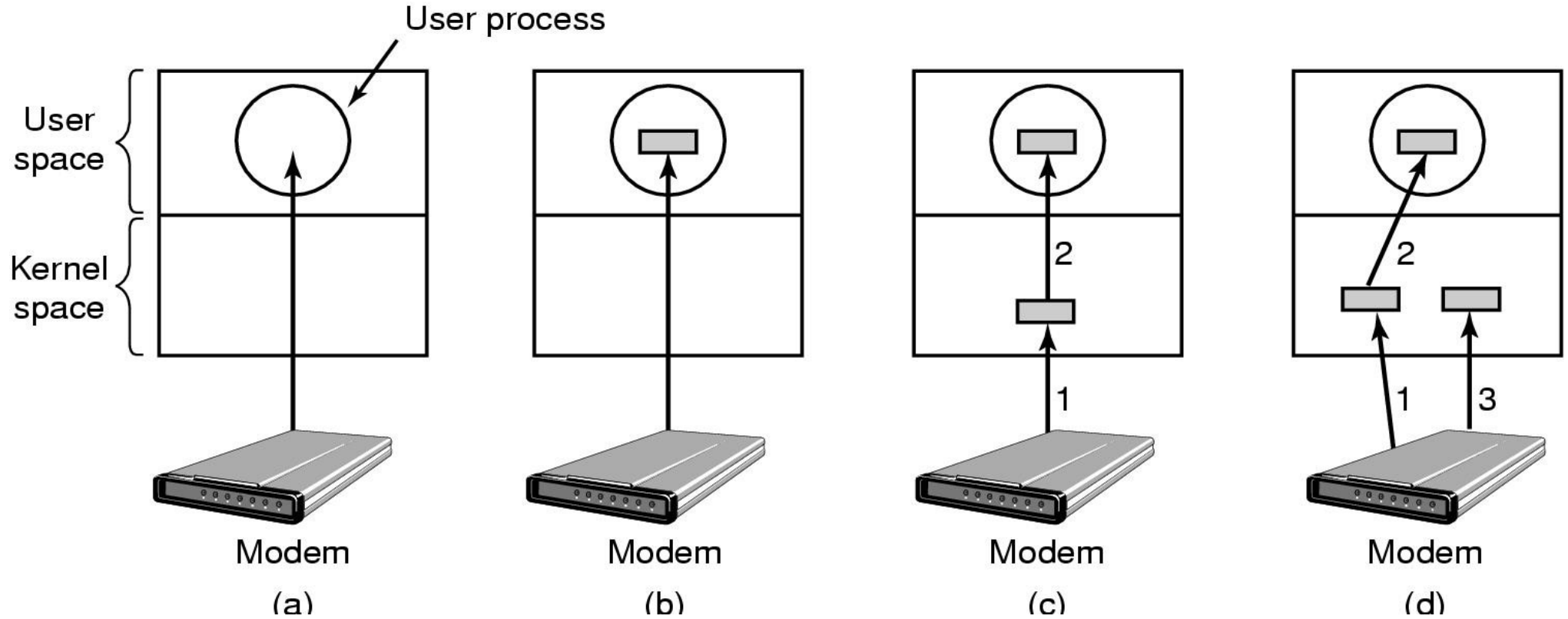
Sürücülerin çalışma zamanında, ihtiyaç anında dinamik olarak gerçekleştirilmesi fikri ortaya atılmıştır ve günümüzde bu yöntem tercih edilmektedir.

Aygıt sürücüsünde okuma, yazma, aygıtın açılıp kapatılması, hata kontrolü, günlükleme gibi işlemler olabilir.

Aygıt sürücüsü kendisine işletim sistemi tarafından verilen parametreleri kontrol eder. Bu parametreler hatasız ise işlemi yerine getirir. Hatalı ise işletim sistemini bilgilendirir.

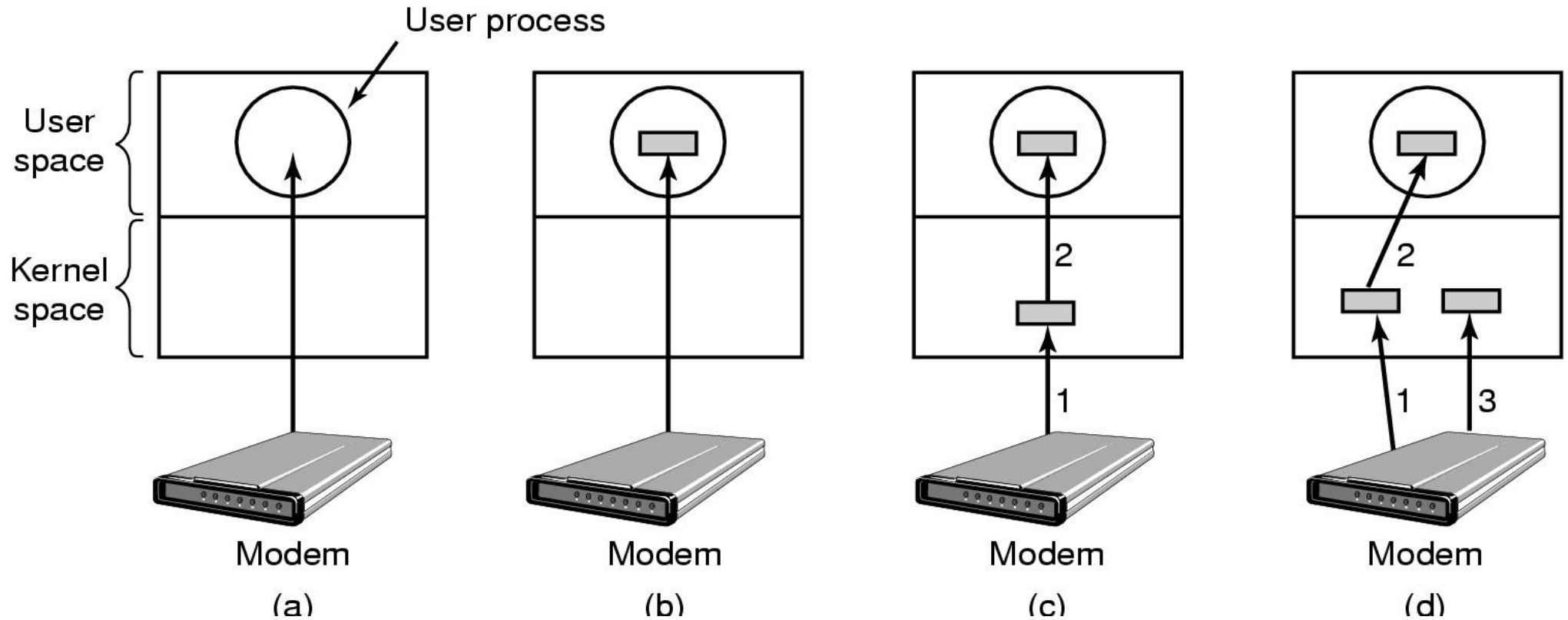
Aygıt sürücüler ve aygıtlar işletim sisteminde bir dosya gibi değerlendirilirler. Bu şekilde işletim sistemi dosyalar için uyguladığı güvenliği bu aygıtlara da uygular.

Tamponlama (Buffering)



Örneğin bir süreç modemden veri okumak istesin. İlk çözüm (a), her seferinde gelen karakteri okumak ve diğer karakter gelinceye kadar bloklanmaktır. Her yeni gelen karakter bir kesme oluşturur. Kesme işleyicisi karakteri sürece taşır ve süreci aktif hale getirir. Süreç diğer karakter için tekrar bloklanır.

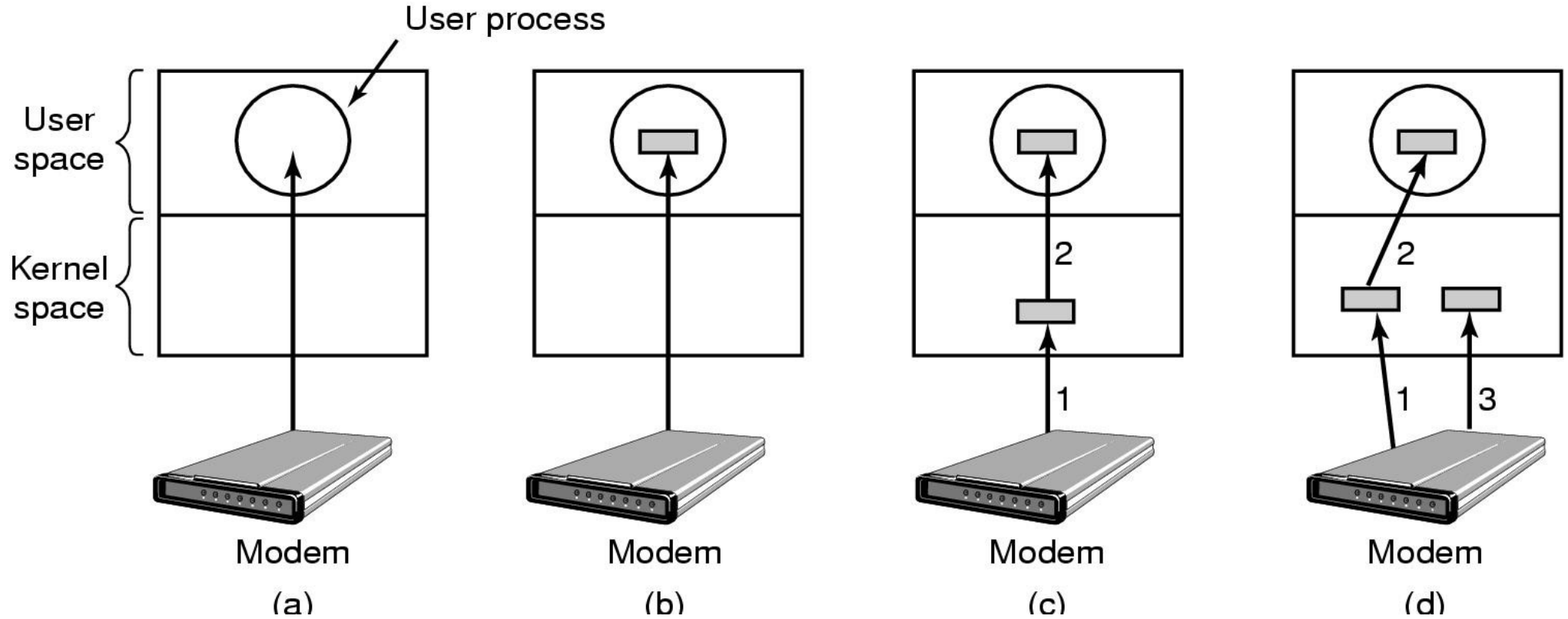
Tamponlama (Buffering)



(b) Kullanıcı bir tampon alanı oluşturabilir. Gelen karakterler bu tampona yazılır. Bu tampon dolduğu zaman süreç uyandırılır.

? Tampon dolu iken bir karakter gelirse
? Her süreç tampon alanını kilitlerde

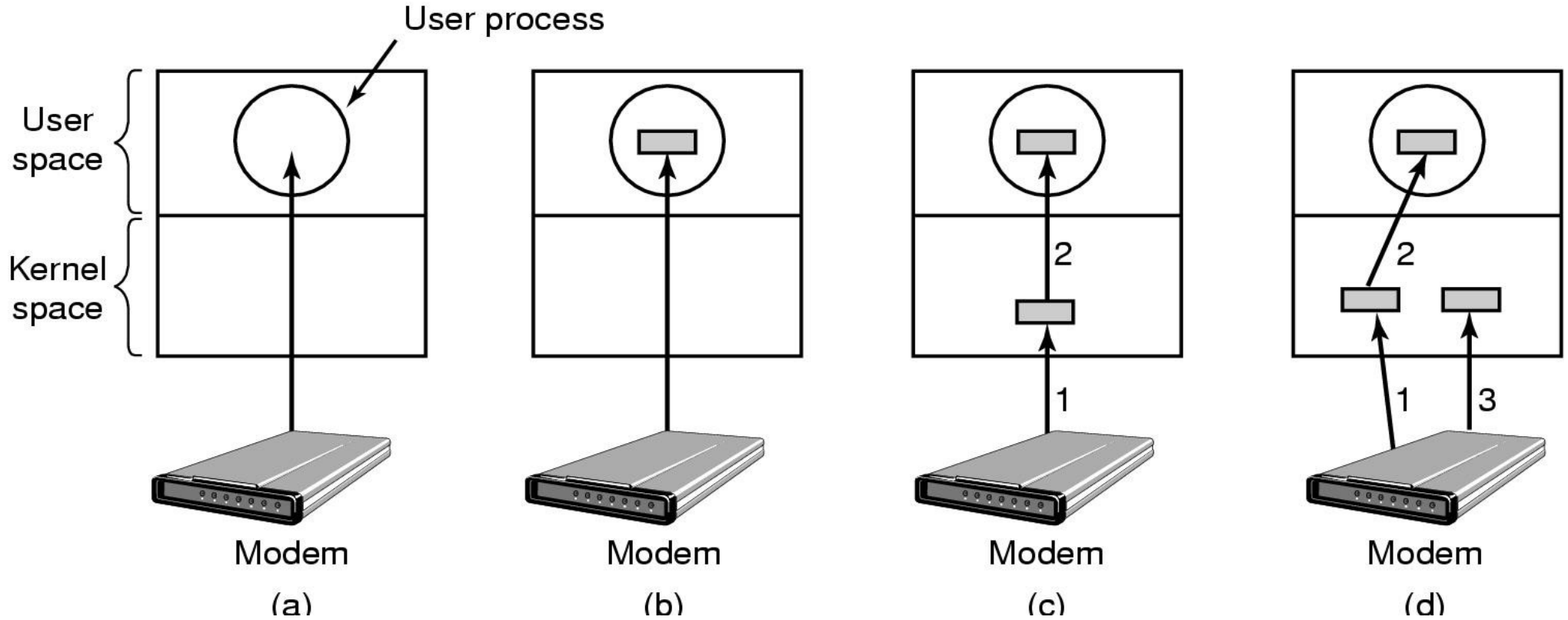
Tamponlama (Buffering)



(c) Kullanıcıda ve çekirdekte tampon bulunabilir. Çekirdekteki tampon dolduğu zaman sürecin tamponuna aktarılır.

? çekirdek tamponundaki veriler sürecin tamponuna aktarılırken yeni veri gelirse

Tamponlama (Buffering)



(d) Çekirdeğe ikinci bir tampon eklenir. Bu şekilde aktarma sırasında yeni veri gelirse ikinci tampon kullanılır. Aynı şekilde bu tamponda sürecin tamponuna aktarılırken diğer tampon yeni veriler için kullanılır.

İşletim Sistemlerine Giriş

Girdi/Çıktı
(I/O)