

Bölüm 1: Giriş

İşletim Sistemleri

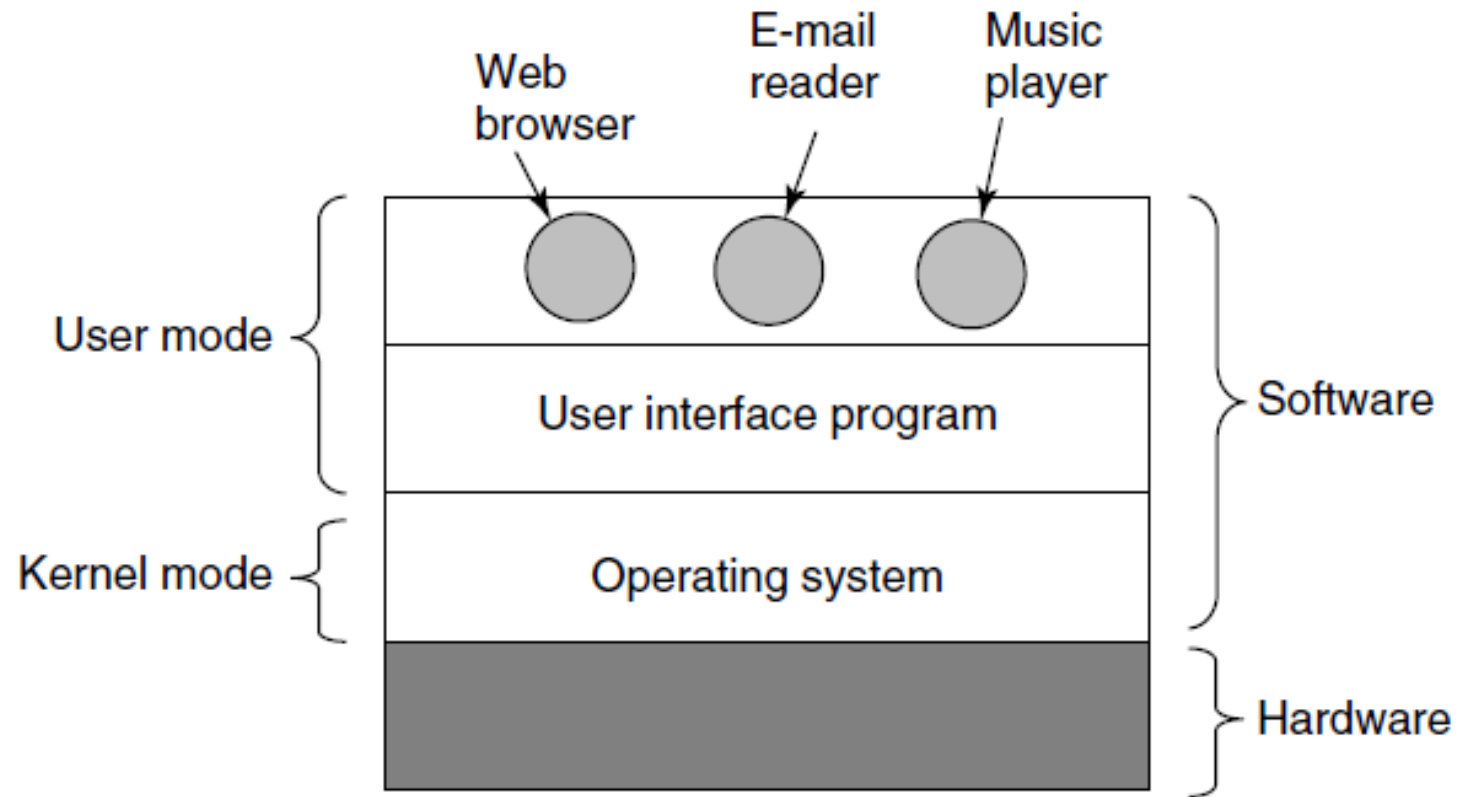
Modern Bilgisayarın Bileşenleri:

- Bir veya daha fazla işlemci
- Ana bellek
- Diskler
- Yazıcılar
- Klavye
- Fare
- Ekran
- Ağ arayüzleri
- G/Ç cihazları

İşletim Sistemi Nerede Yer Alır

- Bilgisayar donanımı ve yazılım arasında bir arayüzdür.
- Donanım, fiziksel olarak mevcut olan bileşenleri (örneğin CPU, RAM, diskler) temsil eder.
- Yazılım ise, bilgisayarın yapabileceği işlemleri yürütmek için yazılmış kodları içerir.
- Donanımın yazılım tarafından nasıl kullanılacağını yönetir.
- İşletim sistemi, yazılımın donanımı kullanmasını kontrol ederken, aynı zamanda donanımın kullanımını optimize eder ve sistemin güvenliğini sağlar.

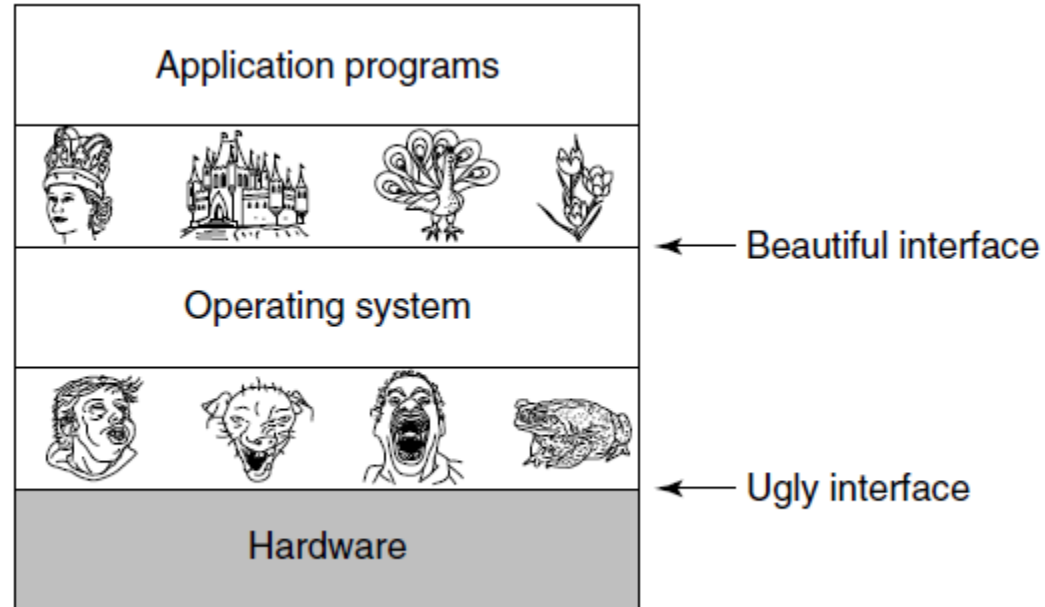
İşletim Sistemi Nerede Yer Alır



Genişletilmiş Makine Olarak İşletim Sistemi

- Donanımın üstüne inşa edilmiş bir yazılımdır.
- Bilgisayar donanımını kullanmayı kolaylaştırır.
- Donanımın özelliklerini ve yeteneklerini kullanılabilir hale getirir.
- Donanımın özelliklerini gizler ve direk kullanmasını engeller.
- İşletim sistemi arayüzünü kullanmak daha kolaydır.
- İşletim sistemleri çirkin donanımları güzel soyutlamalara dönüştürür.

Genişletilmiş Makine Olarak İşletim Sistemi



Kaynak Yöneticisi Olarak İşletim Sistemi

- İşletim sistemi, bilgisayar donanımının kaynaklarını etkili bir şekilde yönetir. Kaynak kullanımını optimize eder. Kaynakların uygulamalar arasında adil bir şekilde dağıtımını sağlar.
- Üstten aşağıya bakış açısı:
 - Uygulama programları için soyutlamalar sağlar
- Aşağıdan yukarıya bakış açısı:
 - Karmaşık sistemin parçalarını yönetir
- Alternatif bakış açısı:
 - Kaynakların düzenli ve kontrollü dağıtımını sağlar.

İşletim Sistemlerinin Tarihi

- İlk jenerasyon (1945-55)
 - vakum tüpleri
- İkinci jenerasyon (1955-65)
 - transistörler ve batch sistemleri
- Üçüncü jenerasyon (1965-1980)
 - IC'ler ve çoklu programlama
- Dördüncü jenerasyon (1980-günümüz)
 - kişisel bilgisayarlar
- Beşinci jenerasyon (1990-günümüz)
 - mobil bilgisayarlar

Transistörler ve Batch Sistemleri

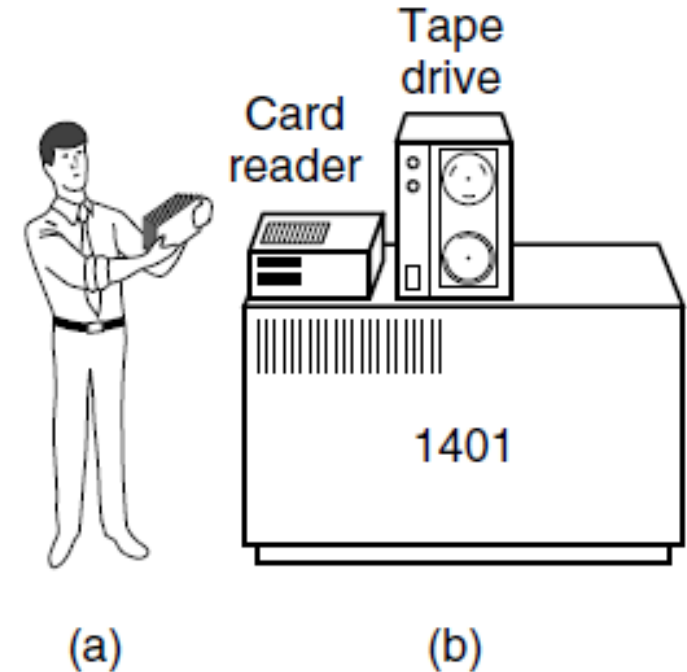
- Transistörlerin icadı ile birlikte ikinci jenerasyon işletim sistemleri ortaya çıktı. Transistörler, vakum tüplerin yerini aldılar
 - daha küçük, daha güvenilir ve daha enerji verimli
- Batch sistemler, işlemlerin toplu olarak yürütülmesini sağlar. İşlemler işlem kuyruğuna eklenir ve işletim sistemi sırayla yürütür.
 - İşlemlerin paralel olarak yürütülmesini engeller.
 - İşlemlerin manuel olarak yürütülmesini gerektirir.
 - Veri işleme, hesaplama ve raporlama gibi işlemler için kullanılır
 - Gerçek zamanlı işlemler için uygun değildir.

Transistörler ve Batch Sistemleri

- Eski bir batch sistemi.

(a) Programcılar 1401'e kartlar getirir.

(b) 1401, iş kartlarını tape'e okur.



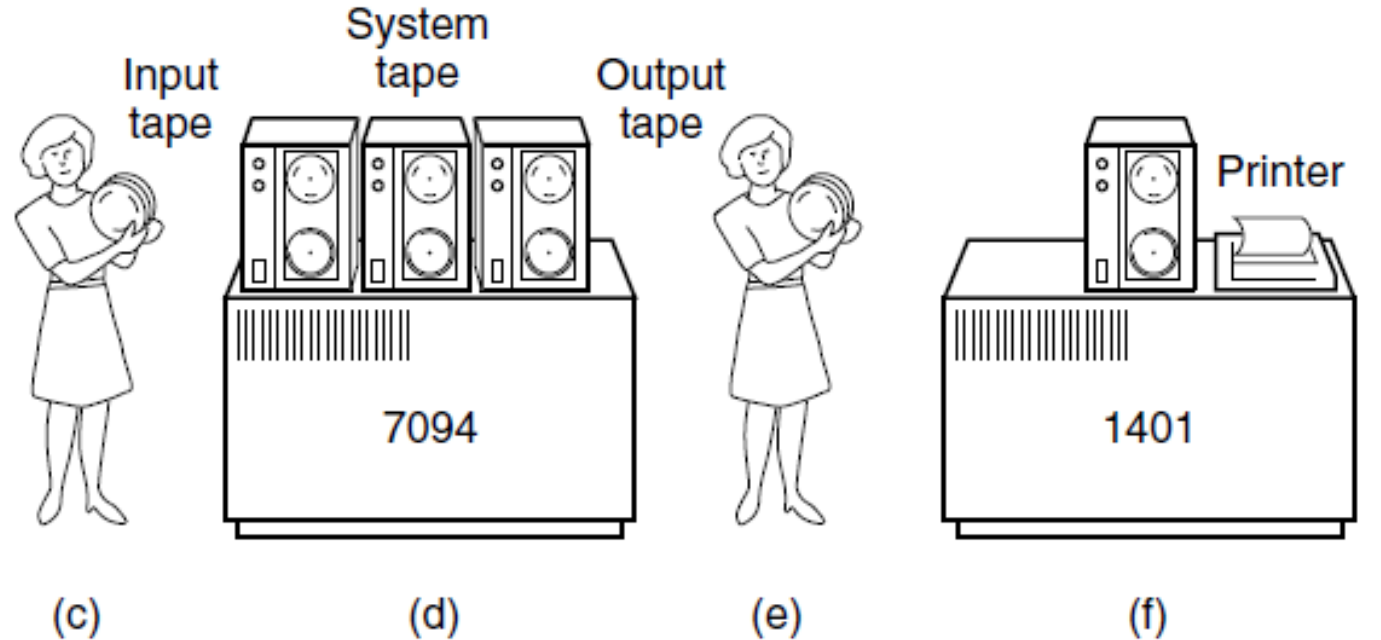
Transistörler ve Batch Sistemleri

(c) Girdi bandının 7094'e taşınması.

(d) 7094 hesaplamaları yapar.

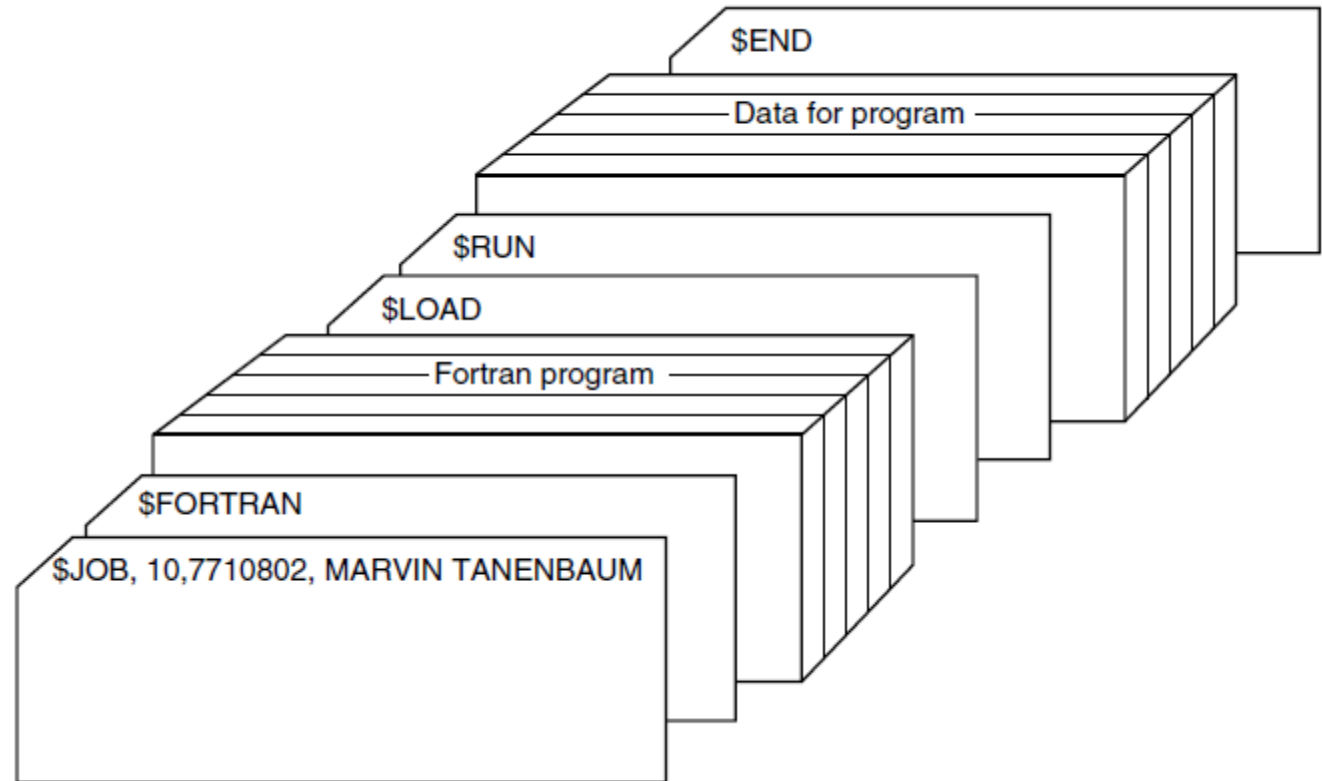
(e) Çıktı bandının 1401'e taşınması.

(f) 1401 çıktıyı yazdırır.



Tipik bir FMS işinin yapısı

- FMS (Flexible manufacturing system)

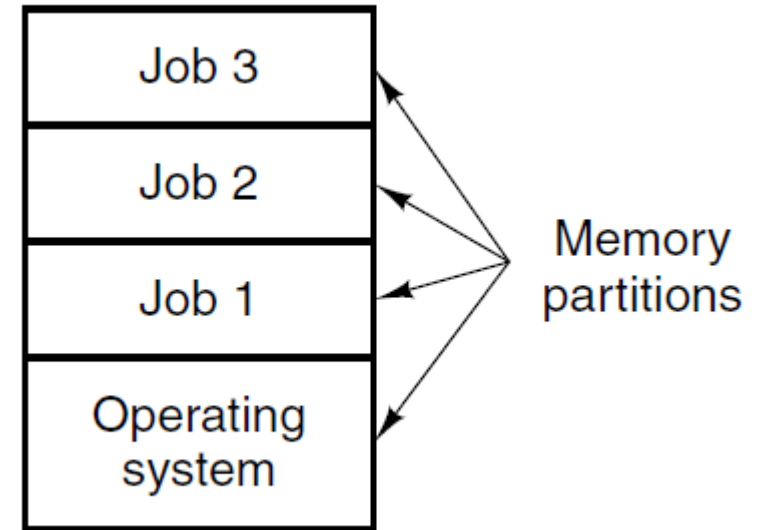


Bütünleşik Devreler ve Çoklu Programlama

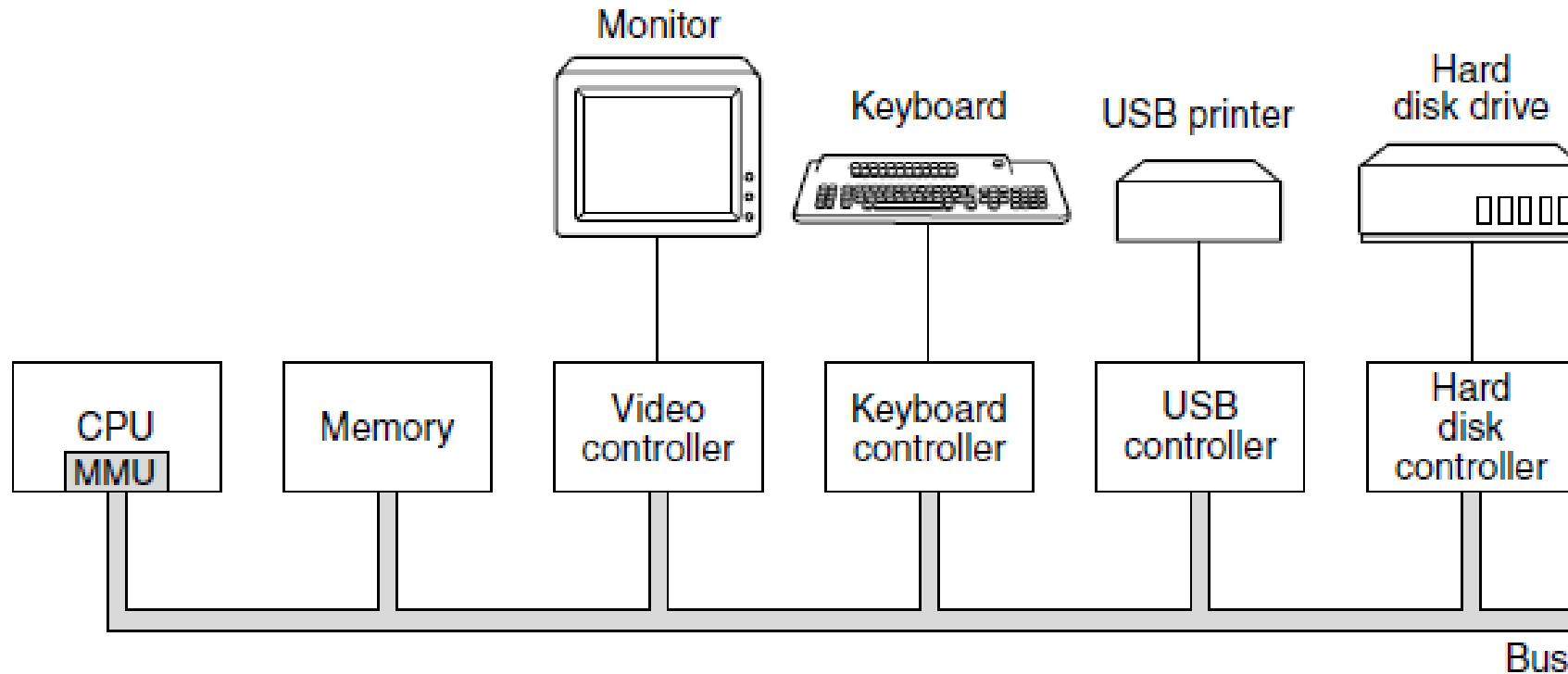
- Bütünleşik devrelerin (IC) icadı ile birlikte üçüncü jenerasyon işletim sistemleri ortaya çıktı. IC'ler transistörlerin yerini aldı.
 - daha küçük, daha güvenilir ve daha enerji verimli
- Çoklu programlama, birden fazla işlemi aynı anda yürütmek için kullanılır.
 - Dinamik olarak işlemlerin ağırlıklarının ayarlanmasını sağlar.
 - İşlemler arasında eşitliği sağlar ve işlemlerin paralel olarak yürütülmesini sağlar.
 - Gerçek zamanlı işlemler için uygun.

Bütünleşik Devreler ve Çoklu Programlama

- Bellekte üç işi olan bir çoklu programlama sistemi.



Kişisel Bilgisayarın Bazı Bileşenleri



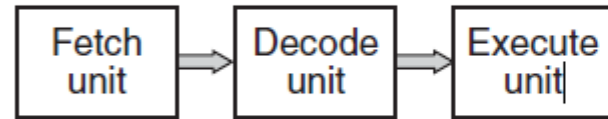
İşlemciler

- Bilgisayarın en önemli bileşenidir ve tüm işlemleri yürütmek için kullanılır.
- İşlemci,
 - bilgisayar kodunu anlar
 - kodu yürütmek için gerekli olan işlemleri gerçekleştirir.
 - çok çekirdekli yapıda olabilir ve birden fazla işlemi aynı anda yürütebilir.
 - hız, çekirdek sayısı, önbellek boyutu, veri yolu genişliği ve diğer özellikler açısından değişebilir.

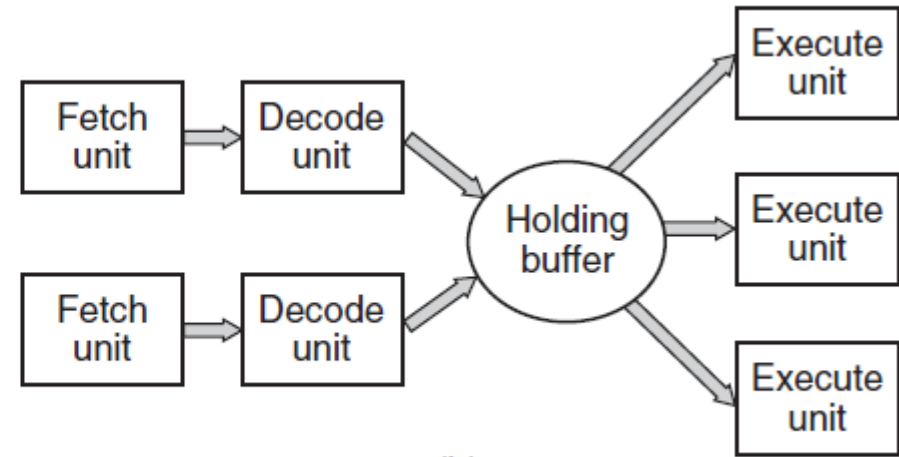
İşlemciler

(a) Üç aşamalı bir boru hattı (pipeline).

(b) Bir superscalar CPU



(a)



(b)

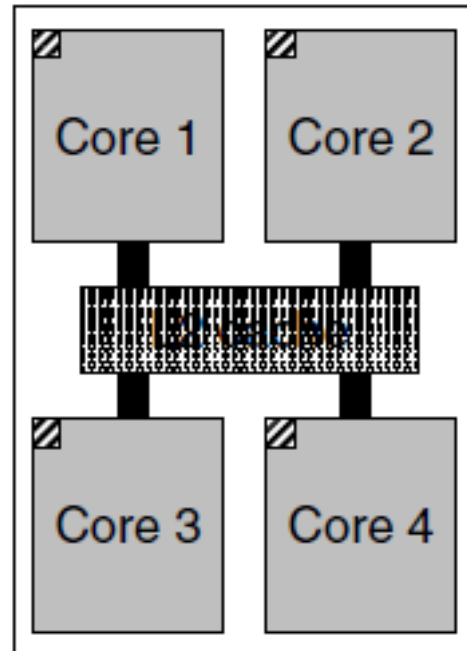
Bellek

- İşlemciler tarafından okunabilecek ve yazılabilecek verileri geçici olarak saklamak için kullanılan bileşendir.
- RAM (Random Access Memory) olarak da adlandırılır.
- Bellek boyutu, bilgisayarın performansını ve kullanılabilirliğini etkiler.
- Bellek, işlemler arasında verileri paylaşmayı ve hızlı erişimi sağlar.

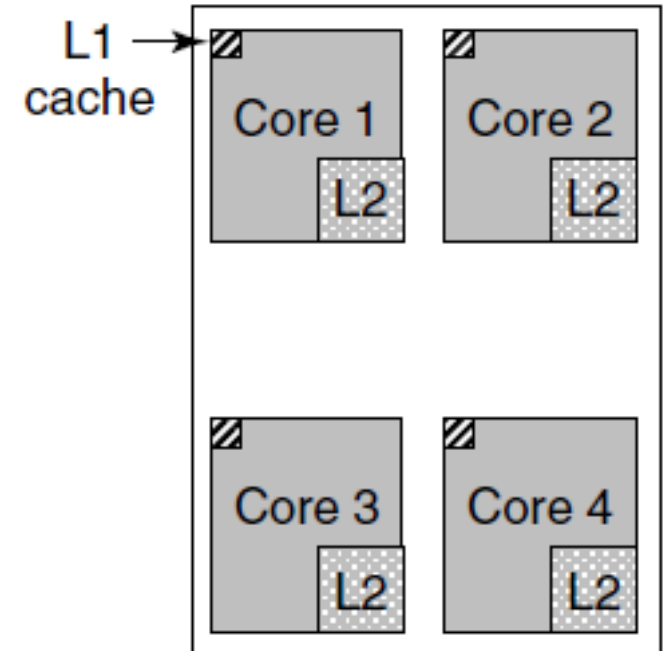
Bellek

(a) Paylaşımlı L2 önbellekli bir dört çekirdekli chip.

(b) Ayrı L2 önbellekli dört çekirdekli chip.



(a)



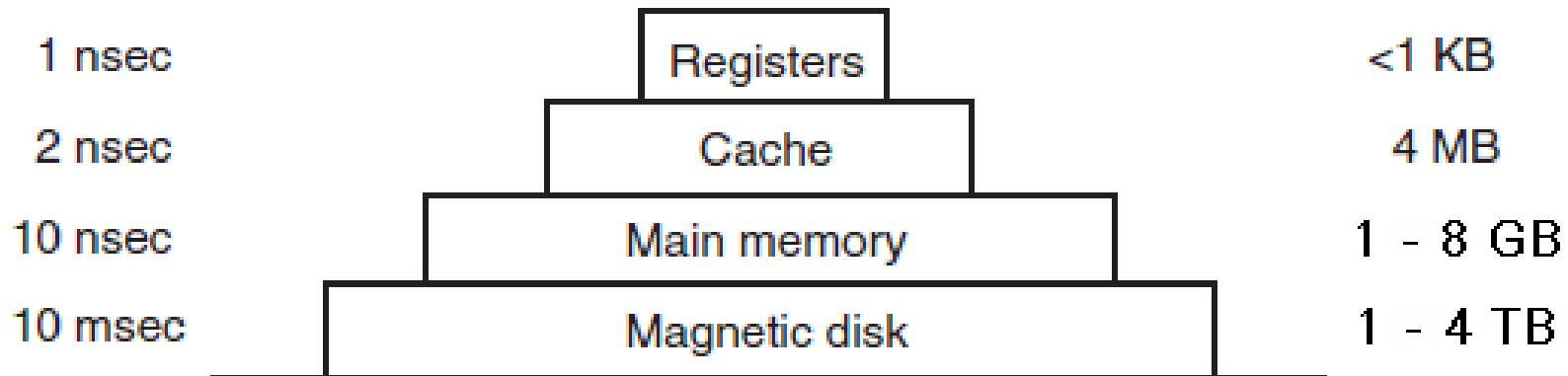
(b)

Bellek

Tipik bir bellek hiyerarşisi. Numaralar çok yaklaşık tahminlerdir.

Typical access time

Typical capacity



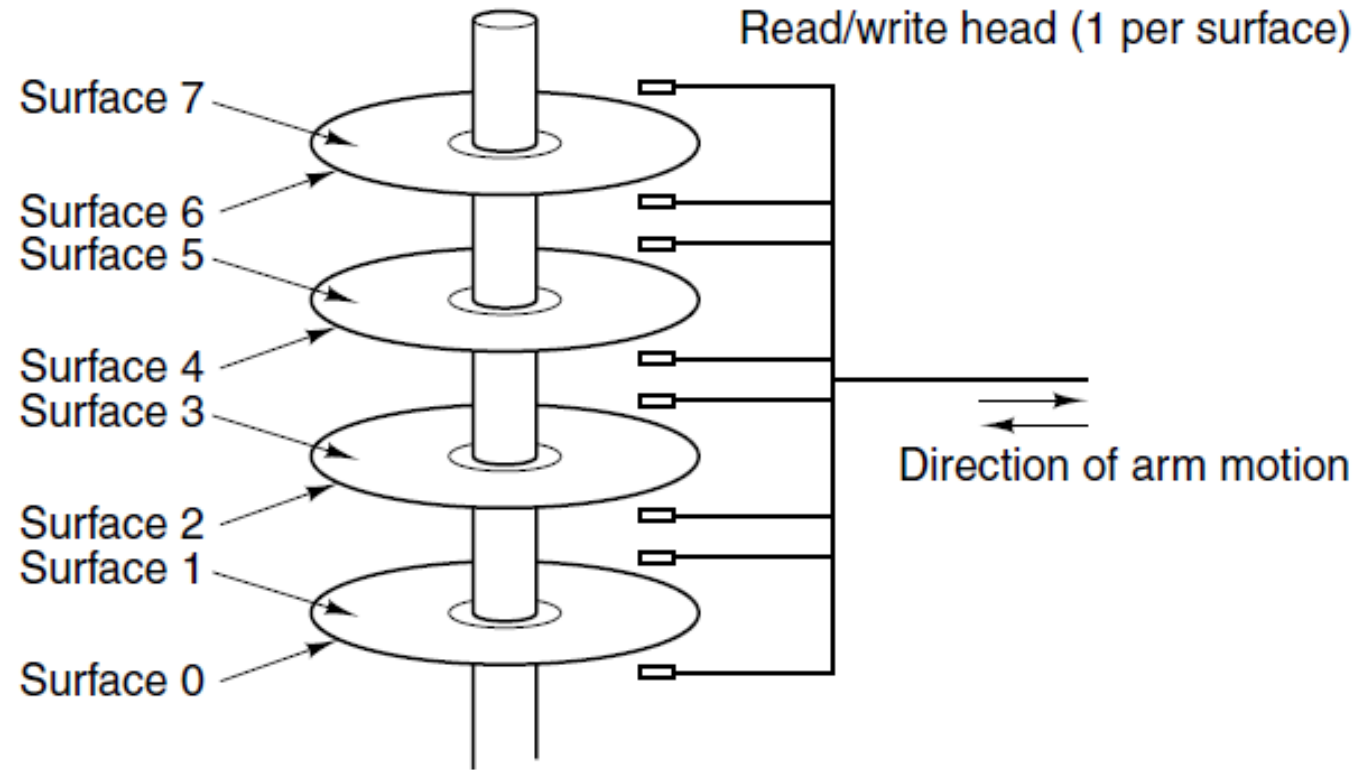
Önbellekleme sistemi sorunları

- Yeni bir öge önbelleğe ne zaman yerleştirilmeli?
- Yeni öge hangi önbellek satırına koyulmalı?
- Yer açmak için önbellekten hangi öge çıkarılmalı?
- Çıkarılan öge bellekte nereye yerleştirilmeli?

Disk

- Verileri uzun vadeli saklamak için kullanılan cihazlardır.
- Okuma ve yazma işlemleri için veriler disk plakaları üzerinde saklanır.
- Disk sürücüleri, farklı boyutlarda ve kapasitelerde olabilir.
- Disk sürücüsü yapısı, disk plakası, okuyucu/yazıcı kafası, motor ve kontrol elemanlarından oluşur.
 - Disk plakası, verileri saklamak için kullanılan alandır.
 - Okuyucu/yazıcı kafası, verileri okuma ve yazma işlemleri için kullanılır.
 - Motor, disk plakasını döndürür ve okuyucu/yazıcı kafasını hareket ettirir.
 - Kontrol elemanları, disk sürücüsünün işlemlerini yönetir.

Disk Sürücüsünün Yapısı

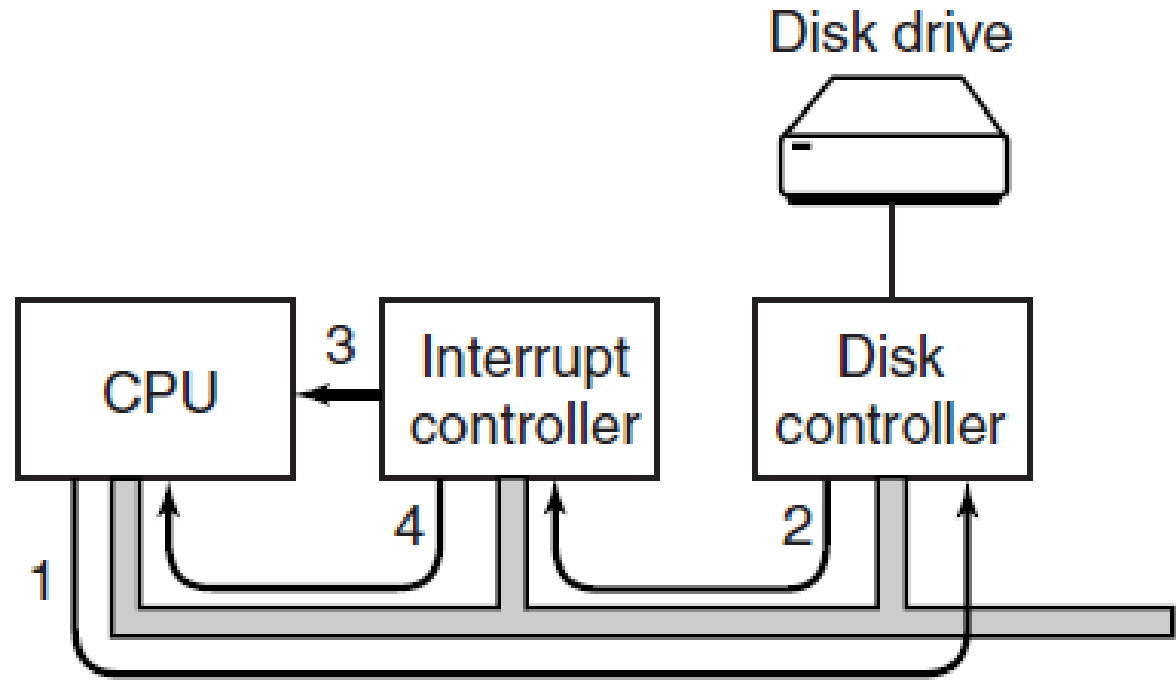


G/Ç Cihazları

- Bilgisayarın veri alma ve veri gönderme işlemlerini gerçekleştirmek için kullandığı cihazlardır.
- Dış dünya ile bilgisayar arasındaki veri transferini sağlar.
- Çeşitli tipte olabilir: Klavye, fare, ekran, yazıcı, tarayıcı, ses kartı, kameralar, vb.
- İşletim sistemi tarafından yönetilir ve kullanıcının cihazları kullanmasına izin verir.
- Bilgisayarın performansını ve kullanılabilirliğini etkiler.

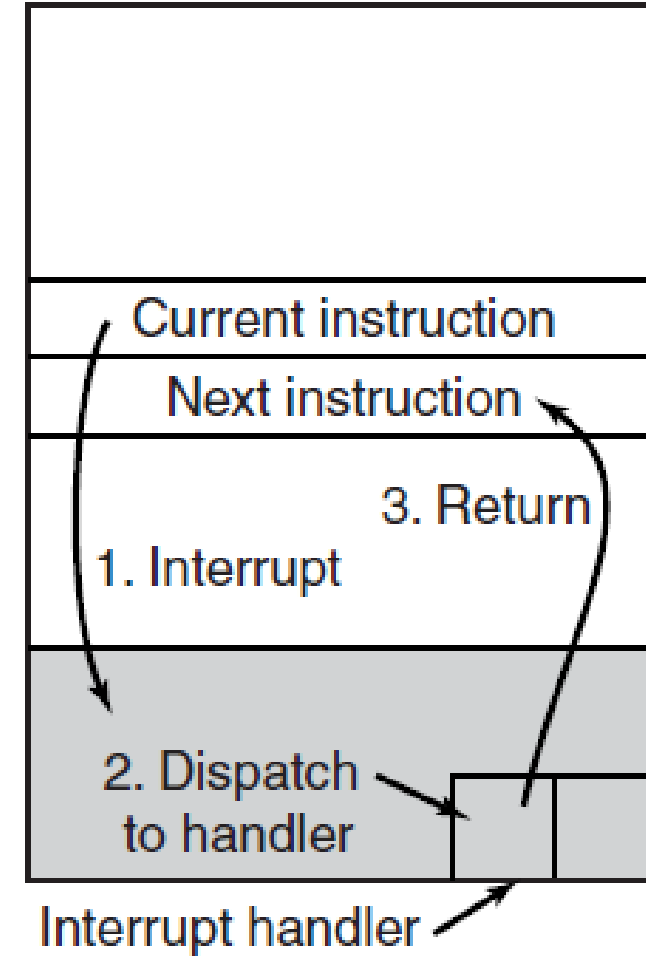
G/Ç Cihazları

G/Ç cihazını başlatma ve bir kesme alma adımları



G/Ç Cihazları

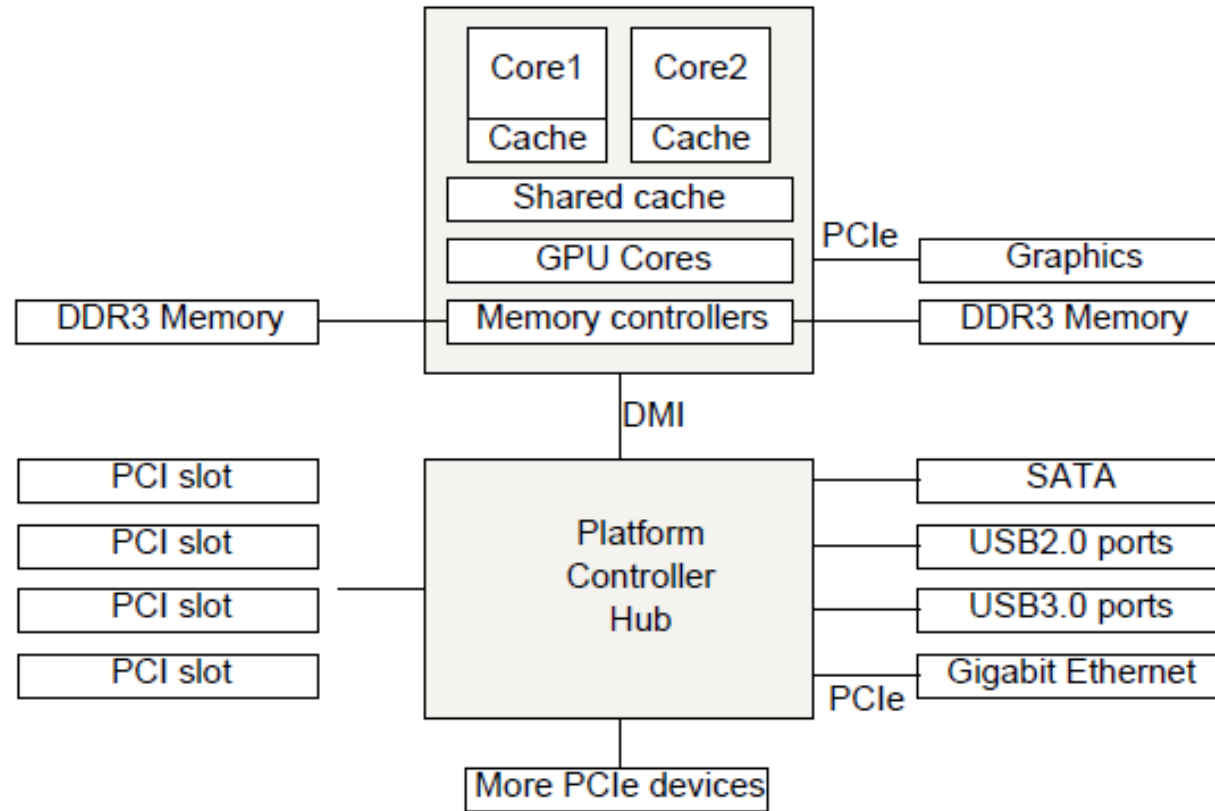
- Kesme işleme,
 - kesmeyi alma
 - kesme işleyicisini çalıştırma
 - kullanıcı programına dönme



Veriyolları

- Veriyolları, bilgisayar bileşenleri arasında veri ve sinyallerin taşınması için kullanılan yapılardır. Örneğin, işlemci, bellek, G/Ç cihazları arasında veri taşır.
- Genişliği ve hızı açısından değişebilir. Örneğin, PCI, PCI-Express, USB gibi.
- Veri taşınma yönetiminden işletim sistemi sorumludur.
- Bilgisayarın performansını ve kullanılabilirliğini etkiler.

X86 Sistem Yapısı



İşletim Sistemi Çeşitleri

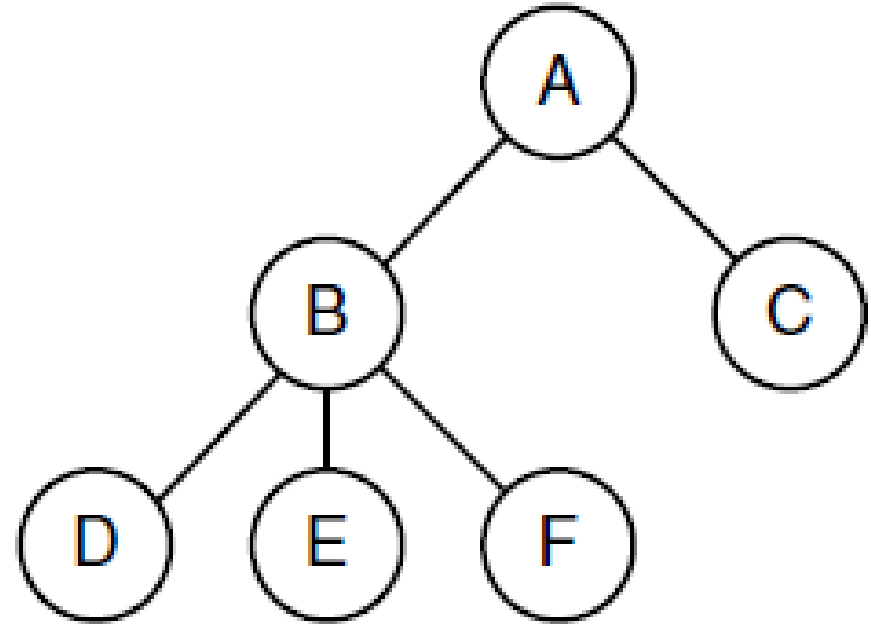
- Anaçatı (mainframe)
- Sunucu (server)
- Çokişlemcili (multiprocessor)
- Kişisel (personal)
- Mobil (handheld)
- Gömülü (embedded)
- Algılayıcı düğüm (sensor node)
- Gerçek zamanlı (real-time)
- Akıllı kart (smart card)

Süreçler

- İşletim sistemi tarafından yürütülen programlardır, işletim sistemi tarafından yönetilir.
- Bir programın çalıştırılabilmesi için gerekli tüm bilgiyi tutan konteyner olarak düşünülebilir.
- İşletim sistemi tarafından atanmış kaynaklar (örneğin bellek, CPU) ile ilişkilidir.
- Bellekte saklanır ve yürütülürler. Adres uzayı ile ilişkilidir.

Süreçler

Süreç ağacı. A süreci, B ve C olmak üzere 2 çocuk süreç başlatır. B süreci D, E ve F olmak üzere 3 çocuk süreç başlatır.

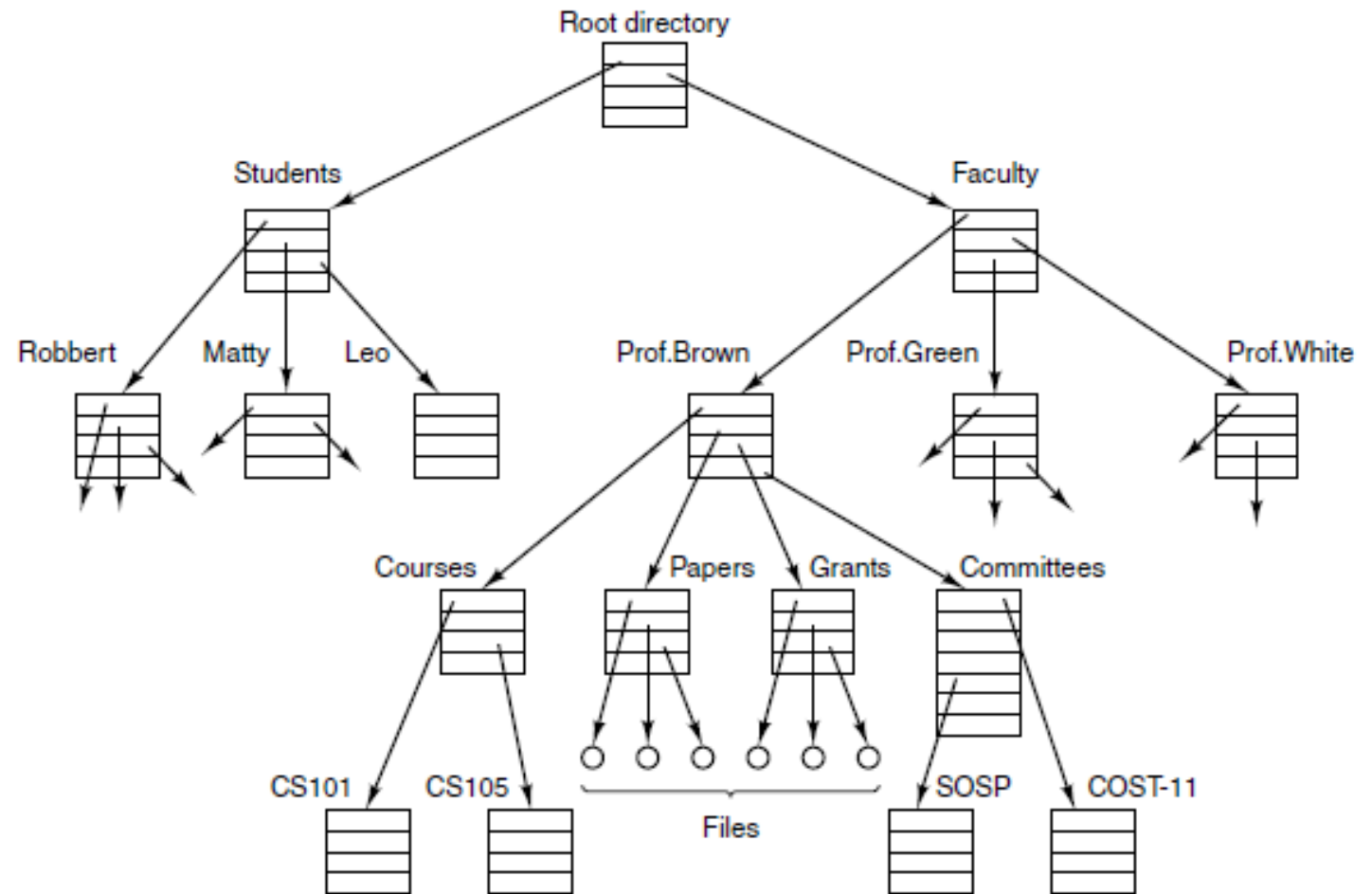


Dosyalar

- Veri depolama birimleridir.
- Dosyalar, veri, metin, resim, video, ses ve diğer türlerde olabilir.
- İşletim sistemi tarafından yönetilir ve veri depolama işlemleri gerçekleştirilir.
- İşletim sistemi tarafından belirlenen izin yapısına göre saklanır.
- Kullanıcılar tarafından erişilebilir, okunabilir, yazılabilir veya silinebilir.

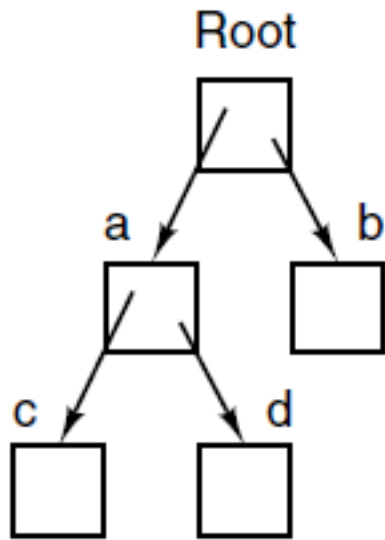
Dosyalar

- Örnek dosya sistemi.

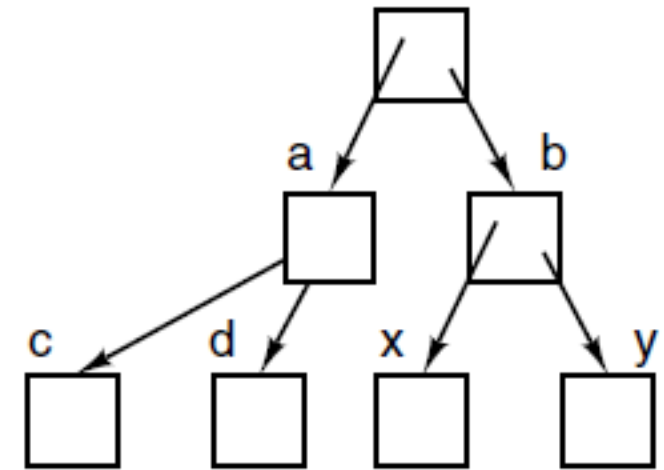
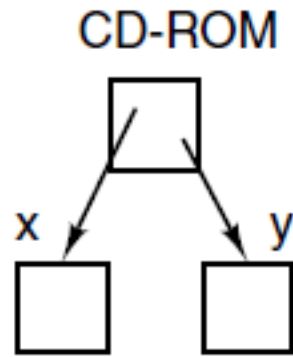


Dosyalar

- (a) Bağlamadan (mount) önce, CD-ROM'daki dosyalara erişilemez.
- (b) Bağlandıktan sonra, dosya hiyerarşisinin bir parçasıdırlar.



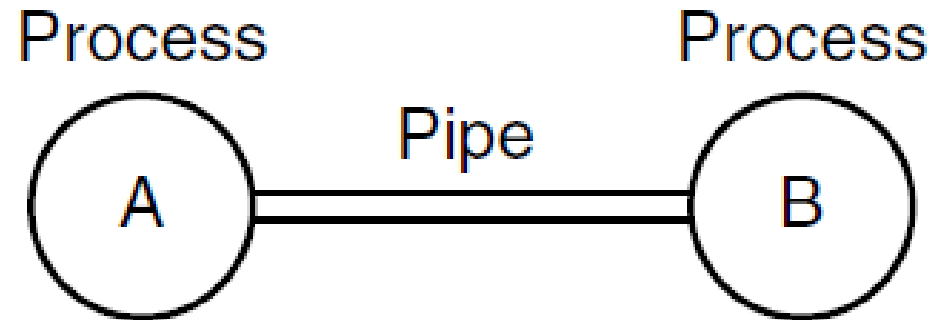
(a)



(b)

Dosyalar

- 2 süreç boru (pipe) ile bağlanmış

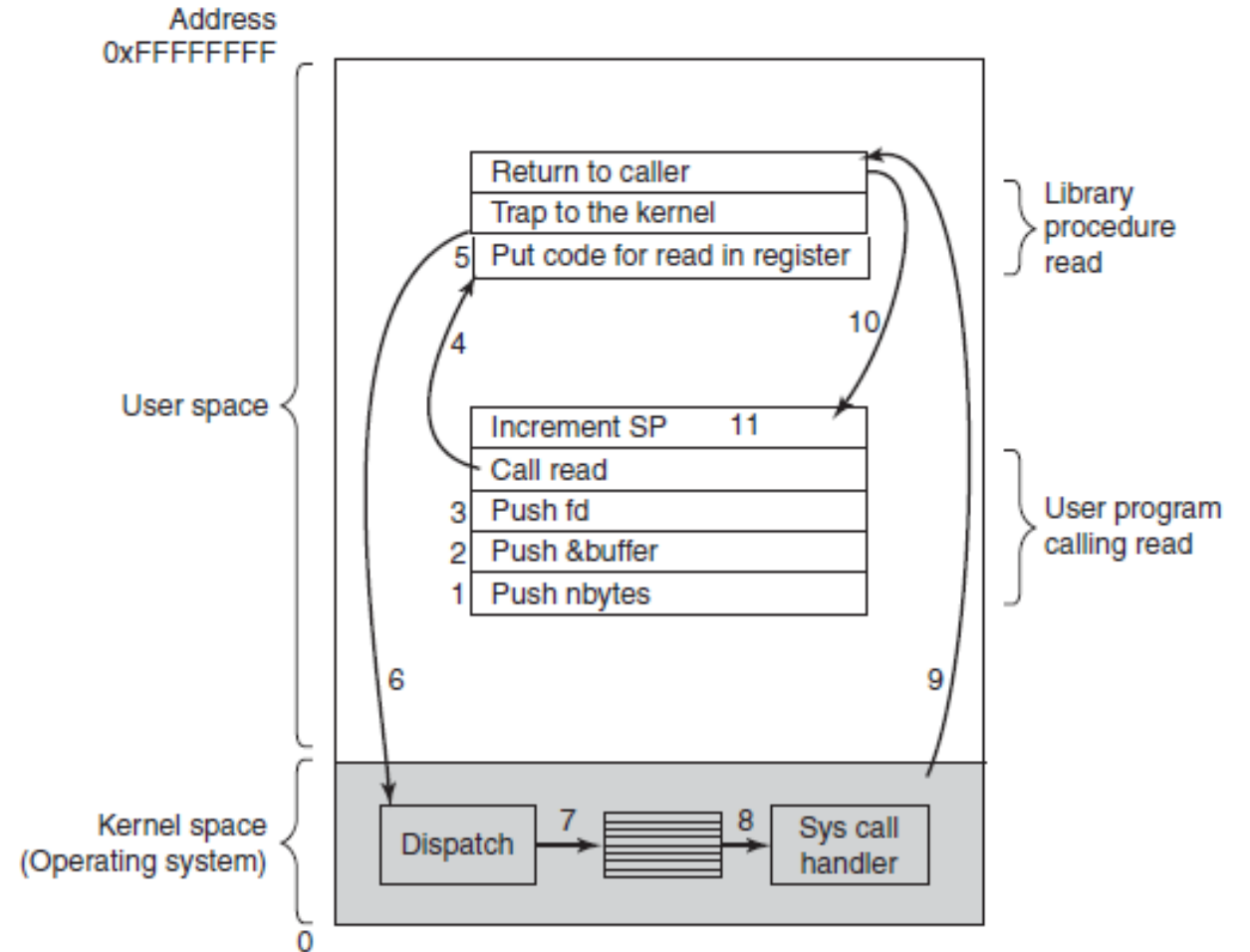


Sistem Çağrıları

- Sistem çağrıları, işletim sistemi tarafından sağlanan hizmetlere erişmek için kullanılır. Örneğin, dosya işlemleri, bellek yönetimi, zaman hizmetleri, vb.
- İşletim sistemi tarafından tanımlanmış bir arayüze göre gerçekleştirilir.
- Sistem çağrıları, uygulama programları tarafından kullanılır ve işletim sistemi tarafından yürütülür.

Sistem Çağrıları

read(fd, buffer, nbytes) sistem çağrısının adım adım gösterimi



Sistem Çağrıları

Başlıca POSIX sistem çağrıları. Hata durumunda -1 döner.

pid: işlem kimliği.

s: geri dönüş kodu

Process management

Call	Description
pid = fork()	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

Sistem Çağrıları

- fd: dosya tanıtıcı,
- n: bayt sayısı,
- position: dosya içinde görelî konum (offset).

File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing, or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

Sistem Çağrıları

Directory and file system management

Call	Description
<code>s = mkdir(name, mode)</code>	Create a new directory
<code>s = rmdir(name)</code>	Remove an empty directory
<code>s = link(name1, name2)</code>	Create a new entry, name2, pointing to name1
<code>s = unlink(name)</code>	Remove a directory entry
<code>s = mount(special, name, flag)</code>	Mount a file system
<code>s = umount(special)</code>	Unmount a file system

Sistem Çağrıları

seconds: geçen süre

Miscellaneous

Call	Description
<code>s = chdir(dirname)</code>	Change the working directory
<code>s = chmod(name, mode)</code>	Change a file's protection bits
<code>s = kill(pid, signal)</code>	Send a signal to a process
<code>seconds = time(&seconds)</code>	Get the elapsed time since Jan. 1, 1970

Süreç Yönetimi

```
#define TRUE 1
```

```
while (TRUE) {  
    type_prompt( );  
    read_command(command, parameters);  
  
    if (fork( ) != 0) {  
        /* Parent code. */  
        waitpid(-1, &status, 0);  
    } else {  
        /* Child code. */  
        execve(command, parameters, 0);  
    }  
}
```

/* repeat forever */
/* display prompt on the screen */
/* read input from terminal */

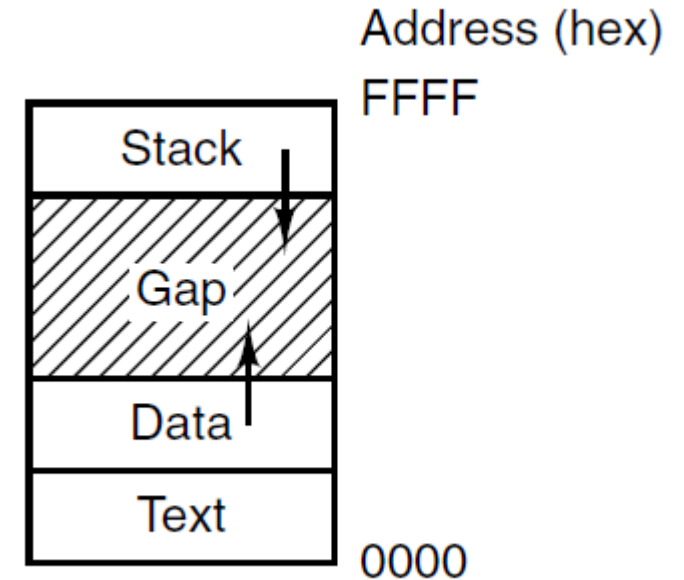
/* fork off child process */

/* wait for child to exit */

/* execute command */

Süreçlerin Bellek Yönetimi

- Süreçler 3 kesime sahiptir. metin, veri, yığıt



Dizin Yönetimi

(a) usr/jim/memo'yu ast'nin dizinine bağlamadan önce.

(b) bağlandıktan sonra.

/usr/ast		/usr/jim	
16	mail	31	bin
81	games	70	memo
40	test	59	f.c.
		38	prog1

(a)

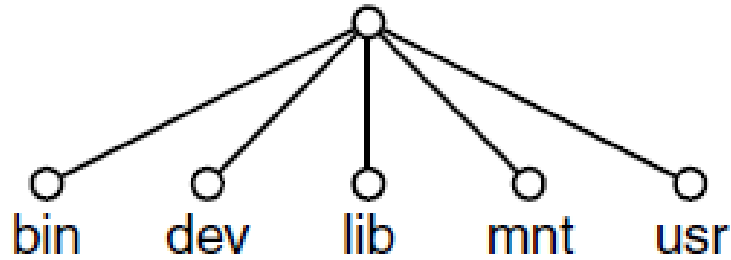
/usr/ast		/usr/jim	
16	mail	31	bin
81	games	70	memo
40	test	59	f.c.
70	note	38	prog1

(b)

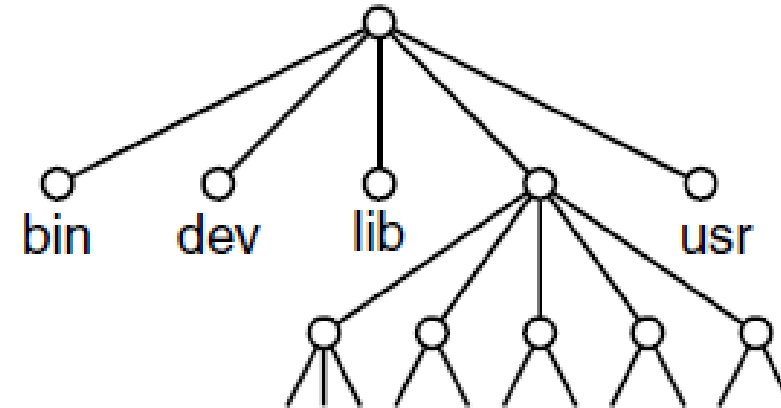
Dizin Yönetimi

(a) Bağlamadan önce dosya sistemi

(b) Bağlamadan sonra



(a)



(b)

The Windows Win32 API

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory

The Windows Win32 API

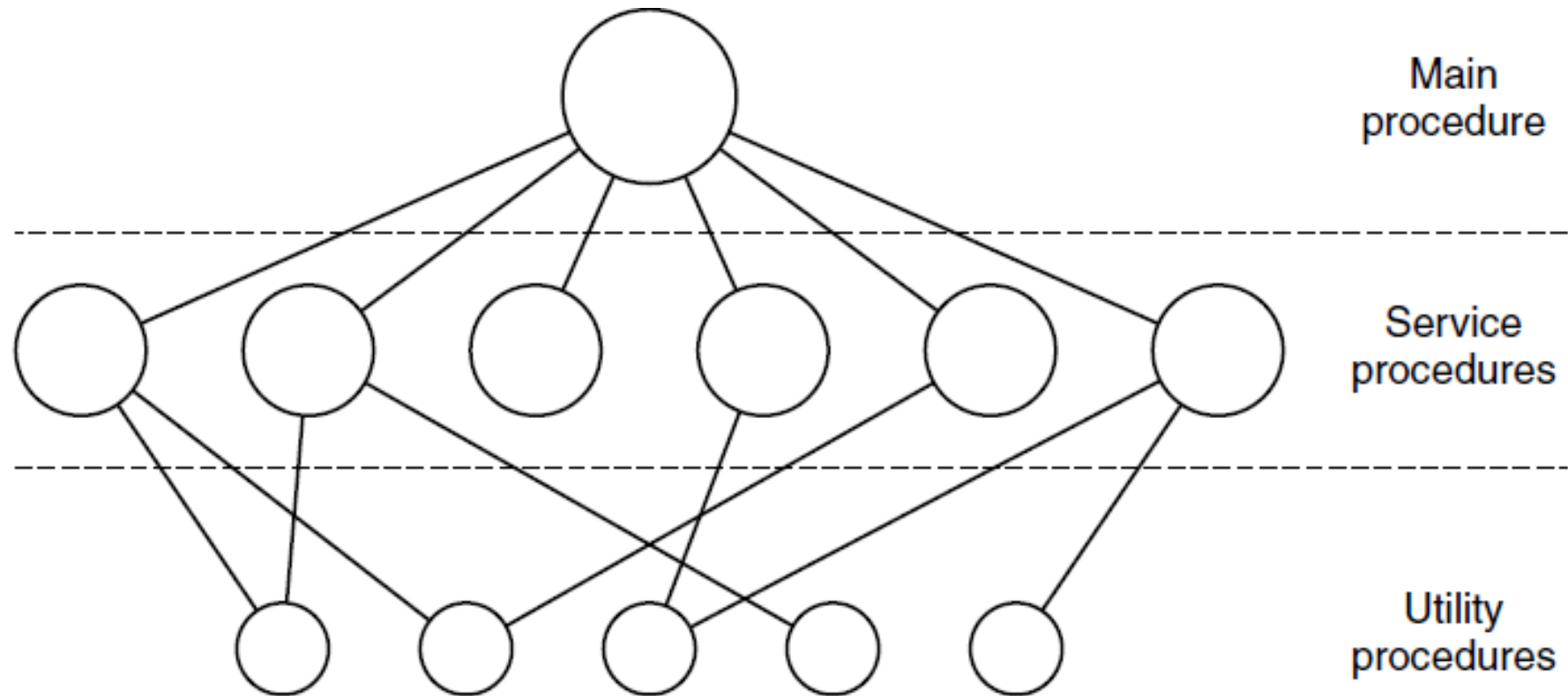
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Monolitik Sistem

İşletim sisteminin temel yapısı

- İstenen hizmet prosedürünü başlatan bir ana program.
- Sistem çağrılarını gerçekleştiren bir dizi hizmet prosedürü.
- Hizmet prosedürlerine yardımcı olan bir dizi yardımcı prosedür.

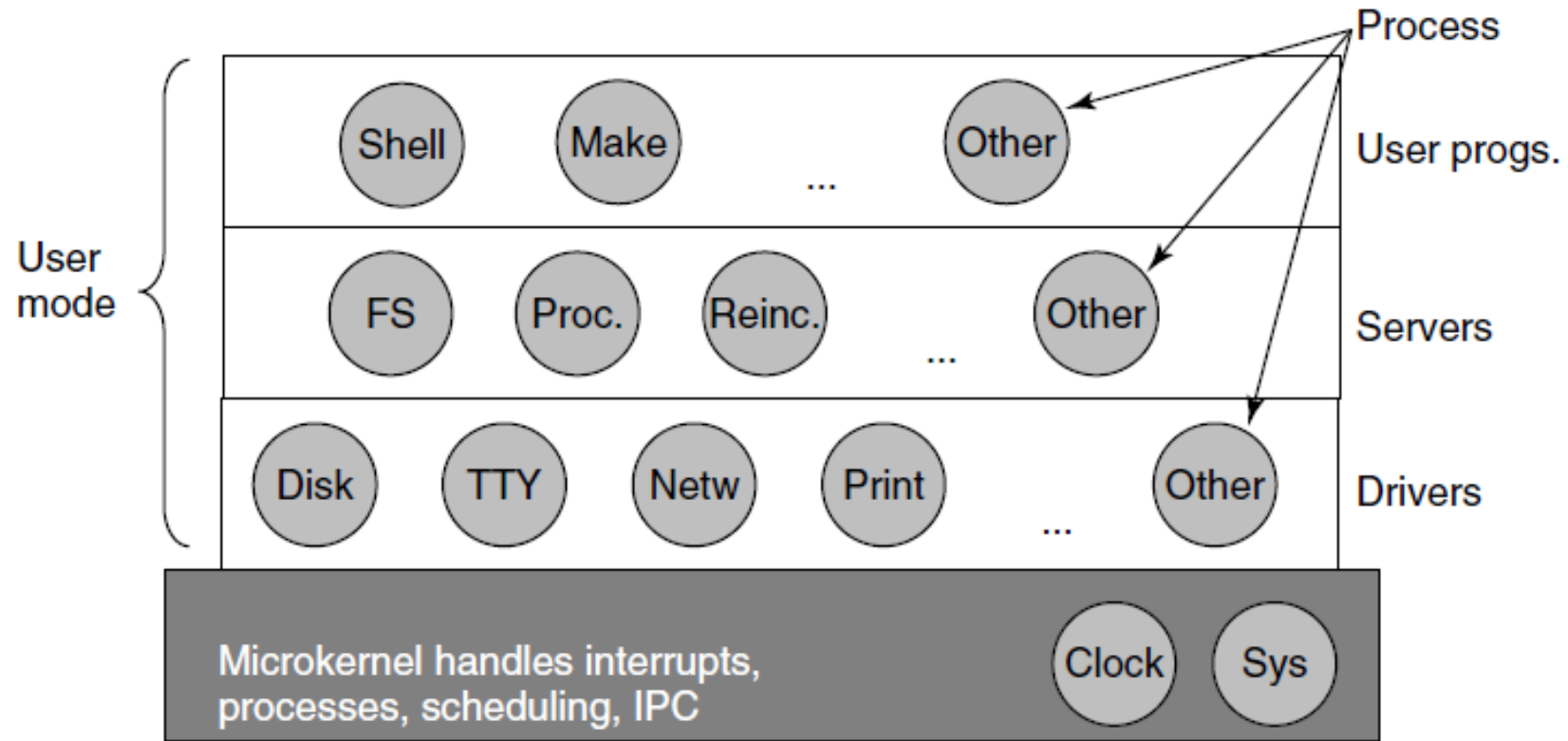
Monolitik Sistem Yapısı



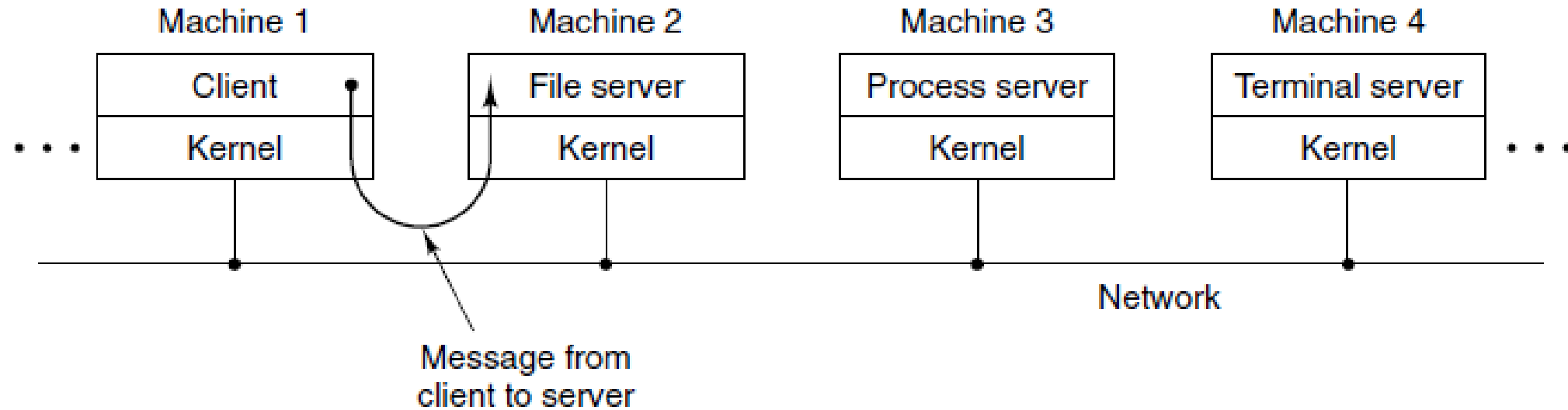
Katmanlı Sistem Yapısı

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

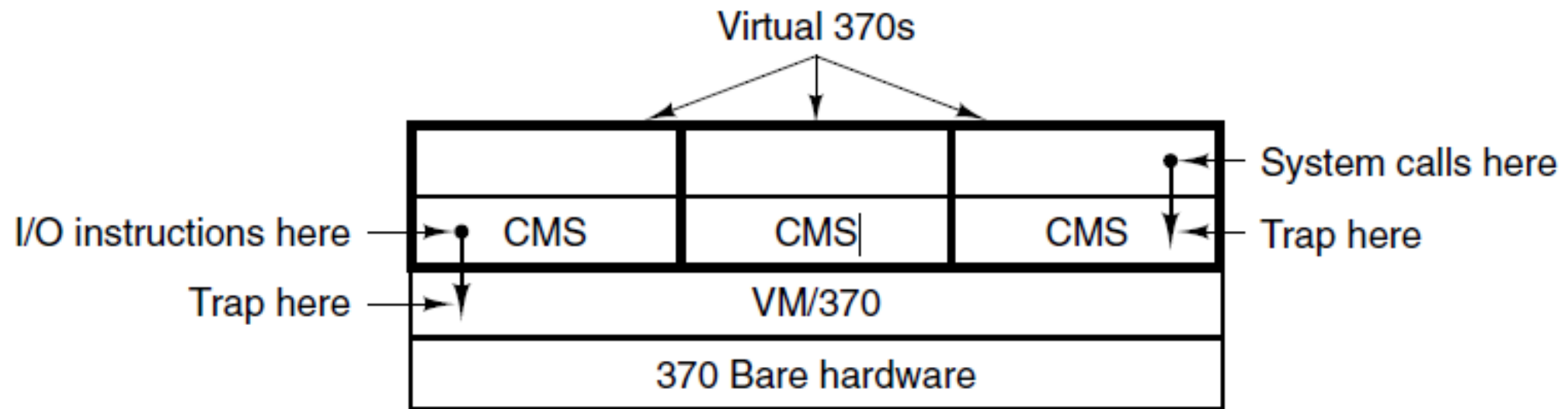
Mikrokernel Sistem Yapısı



İstemci Sunucu Modeli

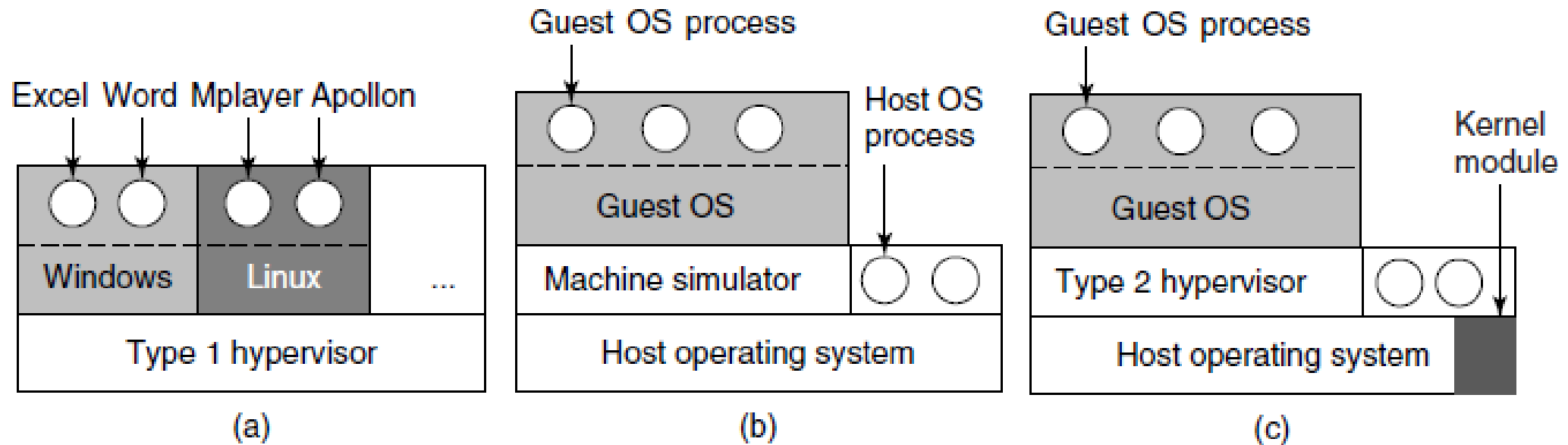


Sanal Makine Yapısı

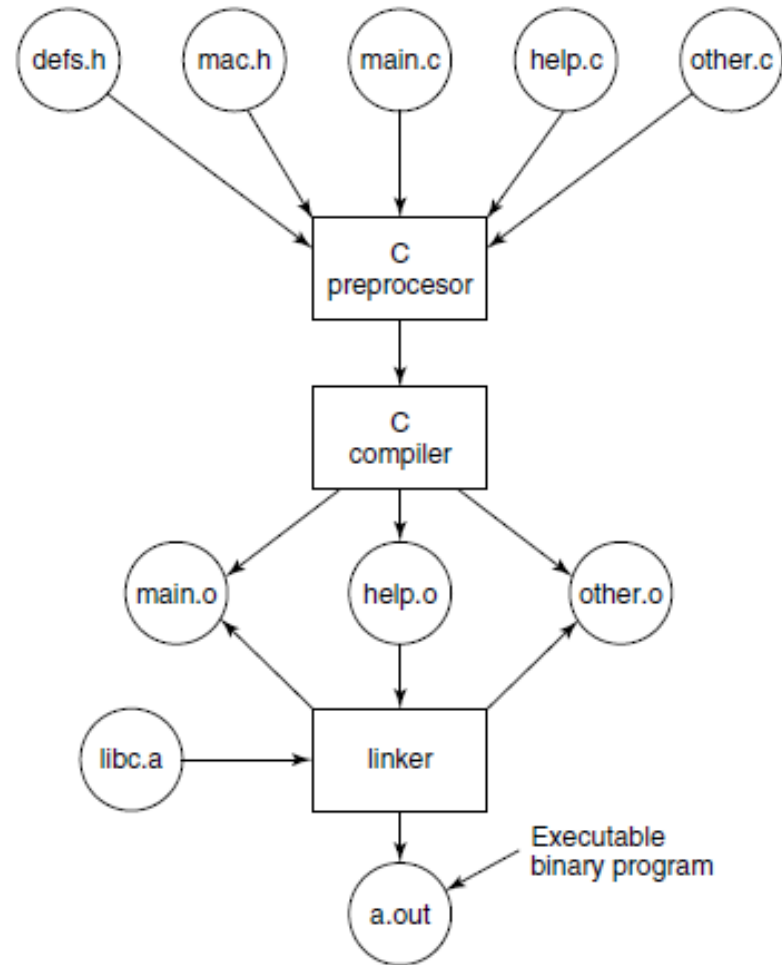


Sanal Makine Yapısı

(a) Tip 1 hipervizör. (b) Yalın tip 2 hipervizör. (c) Pratik tip 2 hipervizör.



Çalıştırılabilir Dosya Oluşturma



Metrik ve Birimleri

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^3	1,000	Kilo
10^{-6}	0.000001	micro	10^6	1,000,000	Mega
10^{-9}	0.000000001	nano	10^9	1,000,000,000	Giga
10^{-12}	0.0000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.0000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.0000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.0000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.0000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

SON