Bölüm 16: Linux

İşletim Sistemleri

Unix Tabanlı İşletim Sistemleri Evrimi

- **Unics**: 1960 başları, *Bell Laboratuarı*, çok kullanıcılı, zaman paylaşımlı.
- **Unix**: 1960 sonları, çok görevli, çok kullanıcılı.
- **PDP-11**: 1970, *DEC* tarafından geliştirilen mini bilgisayar.
- **BSD**: 1970 sonları, *Berkeley California Üniversitesi*, geliştirilen Unix türevi.
- **Minix**: 1987, *Tanenbaum*, Unix benzeri eğitimsel işletim sistemi.
- **Linux**: 1991, *Linus Torvalds*, ücretsiz ve açık kaynaklı.

Unix Tarihçe

- 1969'da *Bell* Laboratuvarında *Ken Thompson* ve *Dennis Ritchie* geliştirdi.
- Mini bilgisayarlar için çok kullanıcılı, çok görevli bir işletim sistemi.
- Orijinal *UNIX* işletim sistemi, Assembly dilinde yazılmıştır.
- 1972'de *UNIX*, taşınabilirlik için *C* dilinde yeniden yazıldı.
- 1984'te *AT&T*, ticari *UNIX* sistemleri için *System V*'i piyasaya sürdü.
- 1990 başlarında, açık kaynak hareketi sonucu,
  - *Linux* ve *BSD* gibi *UNIX* tabanlı işletim sistemleri geliştirildi.

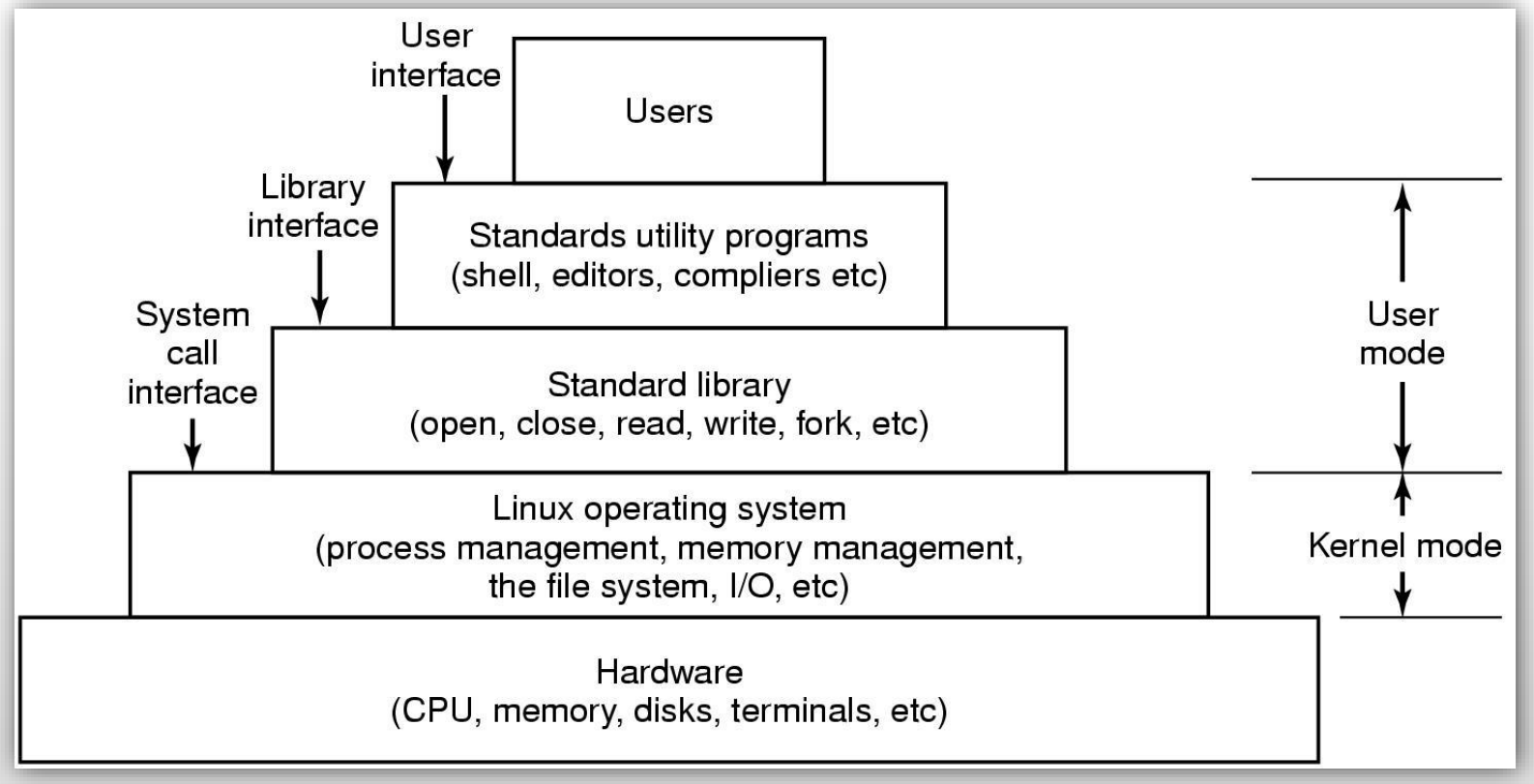
Linux Tarihçe

- 1991: Linus Torvalds tarafından bir hobi projesi olarak oluşturuldu.
- 1992: Linux sürüm 0.12 yayınlandı.
- 1994: Linux'un ilk ticari sürümü (*SLS Linux*).
- 1998: *Red Hat* tarafından ilk Linux dağıtımı.
- 2000: Linux halka açık bir şirket haline geldi (*Red Hat*).
- 2002: İlk yerleşik Linux cihazı piyasaya sürüldü (*Sharp Zaurus PDA*).
- 2005: Linux çalıştıran ilk *Android* cihazının lansmanı (*HTC Dream*).
- 2008: Linux, sunuculardaki en popüler işletim sistemi oldu.

Hedefler

- Programcılar tarafından programcılar için tasarlandı. ☺
  - Basit (*simple*)
  - Şık (*elegant*)
  - Tutarlı (*consistent*)
  - Güçlü (*powerful*)
  - Esnek (*flexible*)

UNIX Katmanlar



UNIX Komutlar

- **ls** - dosya ve dizinleri listeler.
- **cd** - mevcut dizini değiştirir.
- **mkdir** - yeni bir dizin oluşturur.
- **rm** - dosyaları veya dizinleri kaldırır.
- **touch** - yeni bir dosya oluşturur.
- **cp** - dosyaları veya dizinleri kopyalar.
- **mv** - dosyaları veya dizinleri taşır veya yeniden adlandırır.
- **cat** - bir dosyanın içeriğini gösterir.
- **echo** - verilen metni ekranda gösterir.
- **less** - bir dosyanın içeriğini sayfa sayfa görüntüler.

UNIX Komutlar

- **head** - bir dosyanın ilk birkaç satırını görüntüler.
- **tail** - bir dosyanın son birkaç satırını görüntüler.
- **grep** - bir dosyada verilen metni arar.
- **find** - dosyaları veya dizinleri arar.
- **tar** - bir tar arşivi oluşturur veya arşivden çıkartır.
- **ps** - süreç bilgilerini gösterir.
- **proc** - sistem bilgilerini ve kaynak kullanımını görüntüler.
- **chmod** - dosya veya dizin izinlerini değiştirir.
- **chown** - bir dosyanın veya dizinin sahibini veya grubunu değiştirir.
- **ssh** - uzak bir makineye terminal üzerinden güvenli bağlanır.

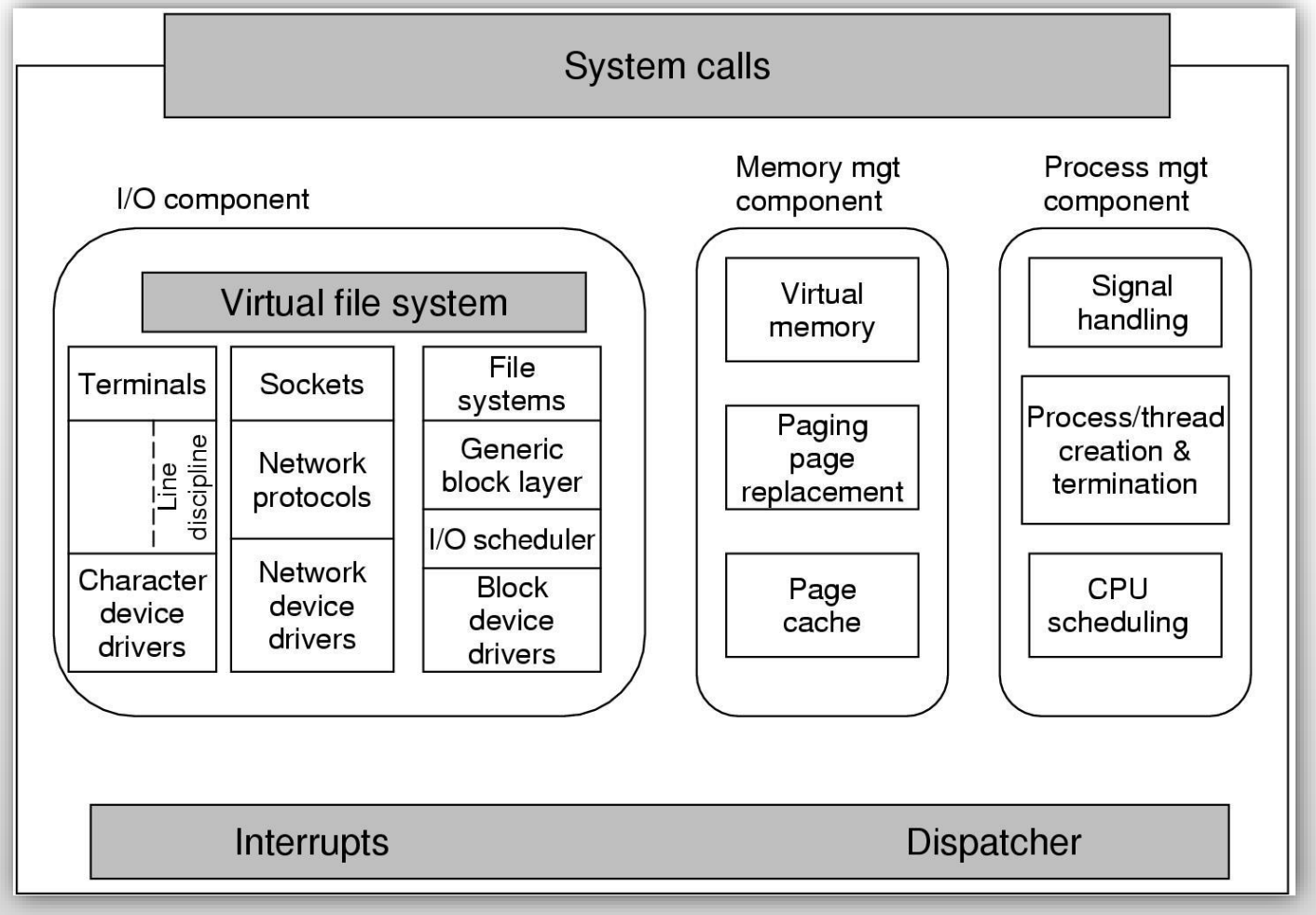
UNIX Komutlar

- **scp** - makineler arasında güvenli dosya kopyalar.
- **sync** - dosyaları makineler arasında senkronize eder.
- **ping** - ağ bağlantısını test eder.
- **ifconfig** - ağ yapılandırma bilgilerini görüntüler.
- **route** - yönlendirme bilgilerini görüntüler veya ayarlar.
- **traceroute** - bir ana bilgisayara giden ağ yolunu görüntüler.
- **netstat** - ağ durumu bilgilerini görüntüler.
- **iptables** - güvenlik duvarı kurallarını yapılandırır.
- **curl** - bir URL'den veri aktarır.
- **wget** - web'den dosya indirir.

UNIX Çekirdek

<table>
<tr><td colspan="5">System calls</td><td colspan="3">Interrupts and traps</td></tr>
<tr><td colspan="2">Terminal handing</td><td>Sockets</td><td>File naming</td><td>Map-ping</td><td>Page faults</td><td rowspan="2">Signal handling</td><td rowspan="2">Process creation and termination</td></tr>
<tr><td rowspan="2">Raw tty</td><td>Cooked tty</td><td>Network protocols</td><td>File systems</td><td colspan="2">Virtual memory</td></tr>
<tr><td>Line disciplines</td><td>Routing</td><td>Buffer cache</td><td colspan="2">Page cache</td><td colspan="2">Process scheduling</td></tr>
<tr><td colspan="2">Character devices</td><td>Network device drivers</td><td colspan="2">Disk device drivers</td><td colspan="2">Process dispatching</td></tr>
<tr><td colspan="8">Hardware</td></tr>
</table>

UNIX Çekirdek



POSIX Sinyaller

- Süreçler arası iletişim (*IPC*) yöntemidir.
- Bir olay sürecini bildirmek veya yürütme akışını kesmek için kullanılır.
- Senkron veya asenkron olabilir.
- Sürecin sonlandırılması gibi önceden tanımlanmış değerler kümesi var.
- Sinyal işleyici,
  - kaynakları serbest bırakmak, veya
  - bilgileri günlüğe (log) kaydetmek gibi gerekli işlemleri gerçekleştirir.

POSIX Sinyaller

- **SIGABRT**: Sürecin hata sonucu anormal şekilde sonlandırılması için üretilir.
- **SIGALRM**: *alarm()* ile ayarlanan süre dolunca üretilir.
- **SIGFPE**: Sıfıra bölme gibi geçersiz işlem gerçekleştiğinde oluşturulur.
- **SIGHUP**: Sistem bir bağlantı kaybı algıladığında üretir.
- **SIGILL**: Geçersiz bir makine talimatıyla karşılaştığında oluşturulur.
- **SIGQUIT**: Çıkış tuşu (CTRL-) tarafından üretilir.
- **SIGKILL**: *kill* komutu tarafından üretilir.
- **SIGPIPE**: Kapatılmış bir boruya yazmaya çalışıldığında üretilir.
- **SIGSEGV**: Geçersiz bir bellek adresine erişildiğinde oluşturulur.
- **SIGTERM**: *kill* komutu tarafından üretilir.
- **SIGUSR1**: Kullanıcı tanımlı sinyal.

Süreç Yönetimi için Sistem Çağrıları

- Linux çekirdeği tarafından sağlanır.
- Kullanıcı düzeyinde programın çekirdekle etkileşim kurmasını sağlar.
- Programın işletim sisteminden hizmet istemesi için birincil mekanizmadır.
- Örnekler: *open(), read(), write(), close(), fork(), exec(), wait().*
- Linux'ta dosya ve aygıt G/Ç, süreç yönetimi ve süreçler arası iletişim (*IPC*) dahil olmak üzere birçok sistem çağrısı türü vardır.
- Sistem çağrıları, Linux işletim sisteminin kritik bir bileşenidir.

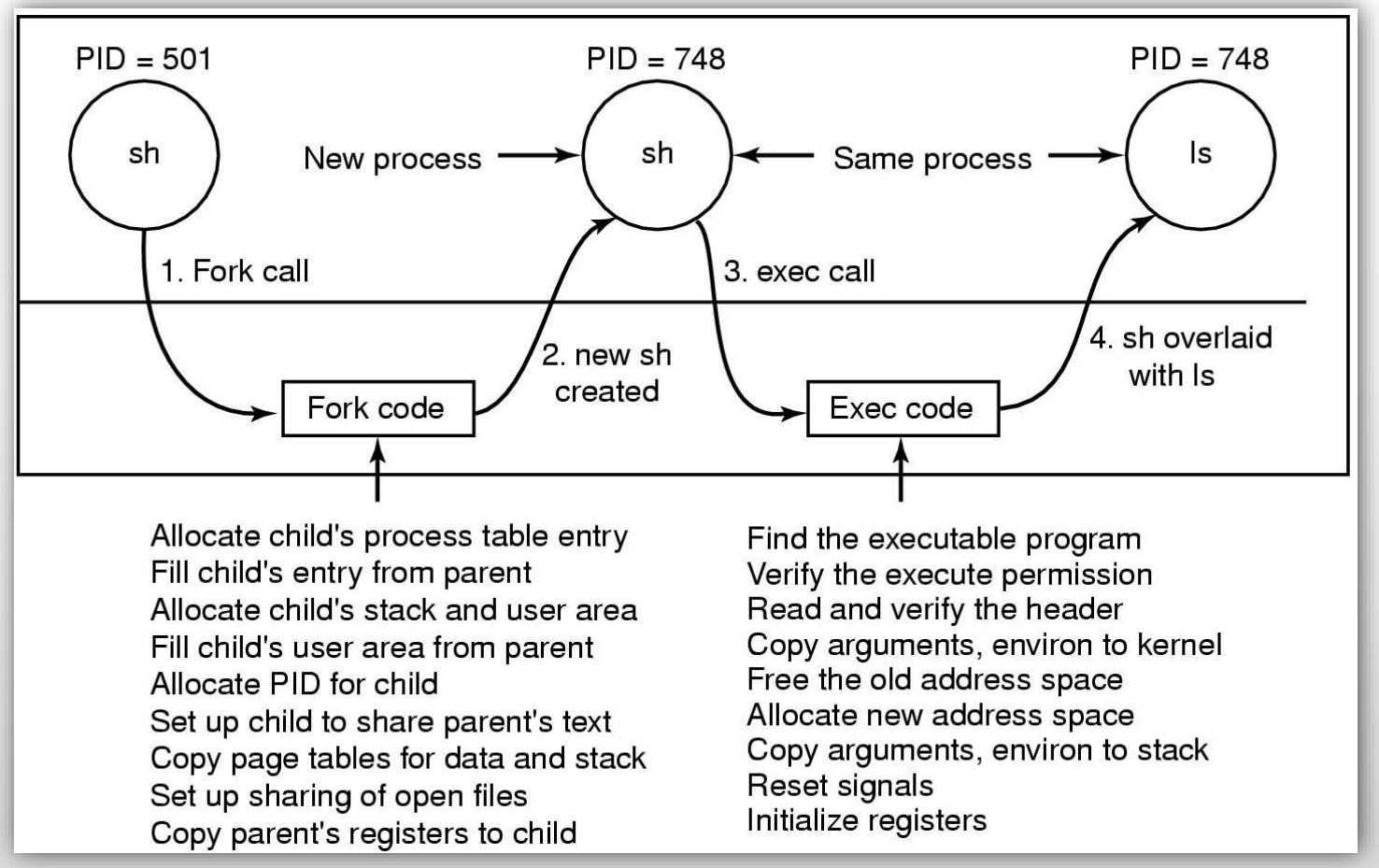
Süreç Yönetimi için Sistem Çağrıları

- **fork**(): süreci çoğaltarak yeni bir süreç oluşturur.
- **waitpid**(): çocuk sürecin sonlanmasını bekler.
- **execve**(): mevcut süreç görüntüsünü yenisiyle değiştirir.
- **exit** (): çağıran süreci sonlandırır.
- **sigaction**(): belirtilen sinyal için yürütülecek işlevi atar.
- **sigreturn**(): kontrolü sinyal işleyiciye bırakır.
- **sigprocmask**(): sürecin sinyal maskesini atar.
- **sigpending**(): sürecin bloke olan sinyallerini döndürür.
- **sigsuspend**(): sürecin sinyal maskesini geçici olarak değiştirir.
- **kill** (): sürecin sonlanması için bir sinyal gönderir.
- **alarm**(): verilen saniye değeri için bir çalar saat ayarlar.
- **pause**(): süreci bir sinyal alınana kadar uyutur.

pthread İşlevleri

- **pthread_create**: yeni iş parçacığı oluşturur.
- **pthread_exit**: iş parçacığını sonlandırır.
- **pthread_join**: bir iş parçacığının sonlanmasını bekler.
- **pthread_mutex_init**: *muteks* nesnesi başlatır.
- **pthread_mutex_destroy**: *muteks* nesnesini yok eder.
- **pthread_mutex_lock**: *muteks* nesnesinin sahipliğini alır.
- **pthread_mutex_unlock**: *muteks* nesnesinin sahipliğini bırakır.
- **pthread_cond_init**: koşul değişkeni nesnesini başlatır.
- **pthread_cond_destroy**: koşul değişkeni nesnesini yok eder.
- **pthread_cond_wait**: koşul değişkenine bağlı iş parçacığını bloke eder.
- **pthread_cond_signal**: iş parçacığının blokesini kaldırır.

ls (list) Komutu

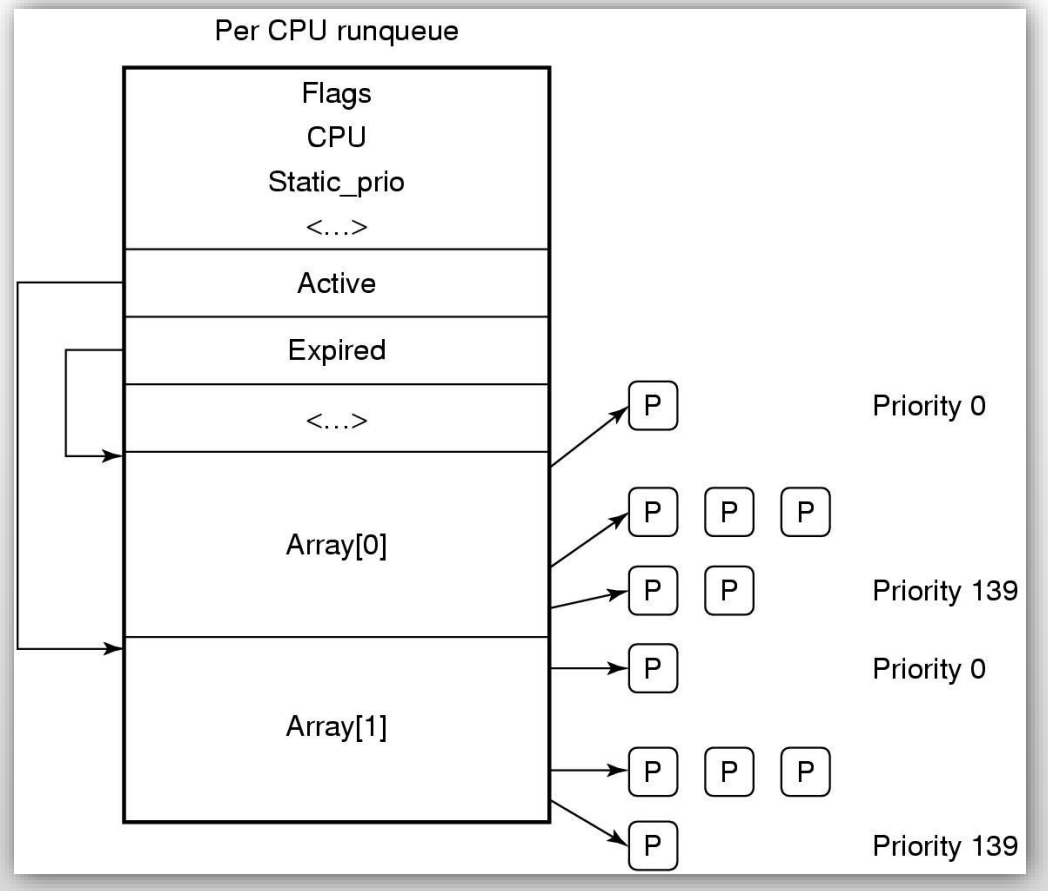


Linux Çizelgeleme

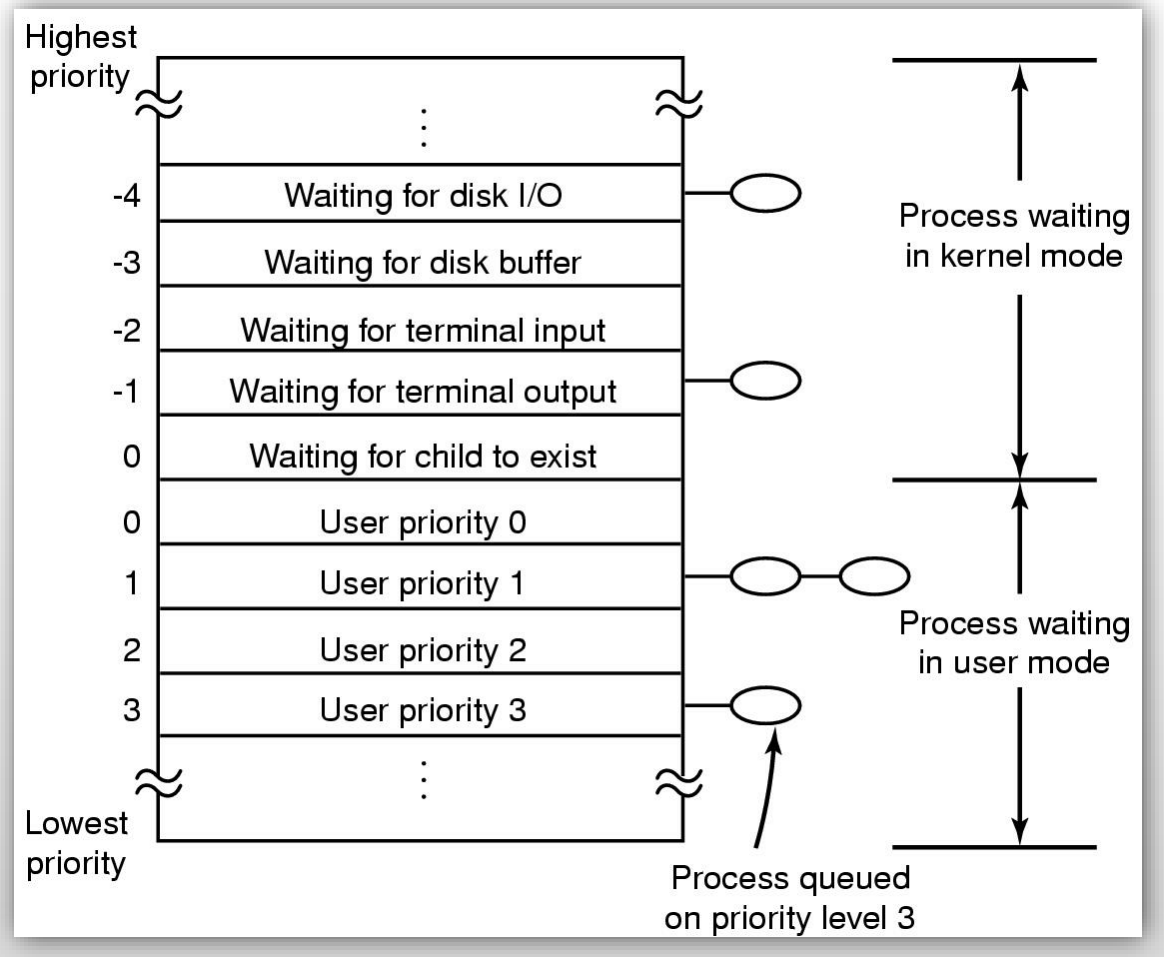
- Gerçek zamanlı ilk-giren ilk-çıkar (*realtime FIFO*).
- Gerçek zamanlı sıralı (*realtime round robin*).
- Zaman paylaşımlı (t*imesharing*).

Linux Çizelgeleme – Öncelik Kuyrukları

- .

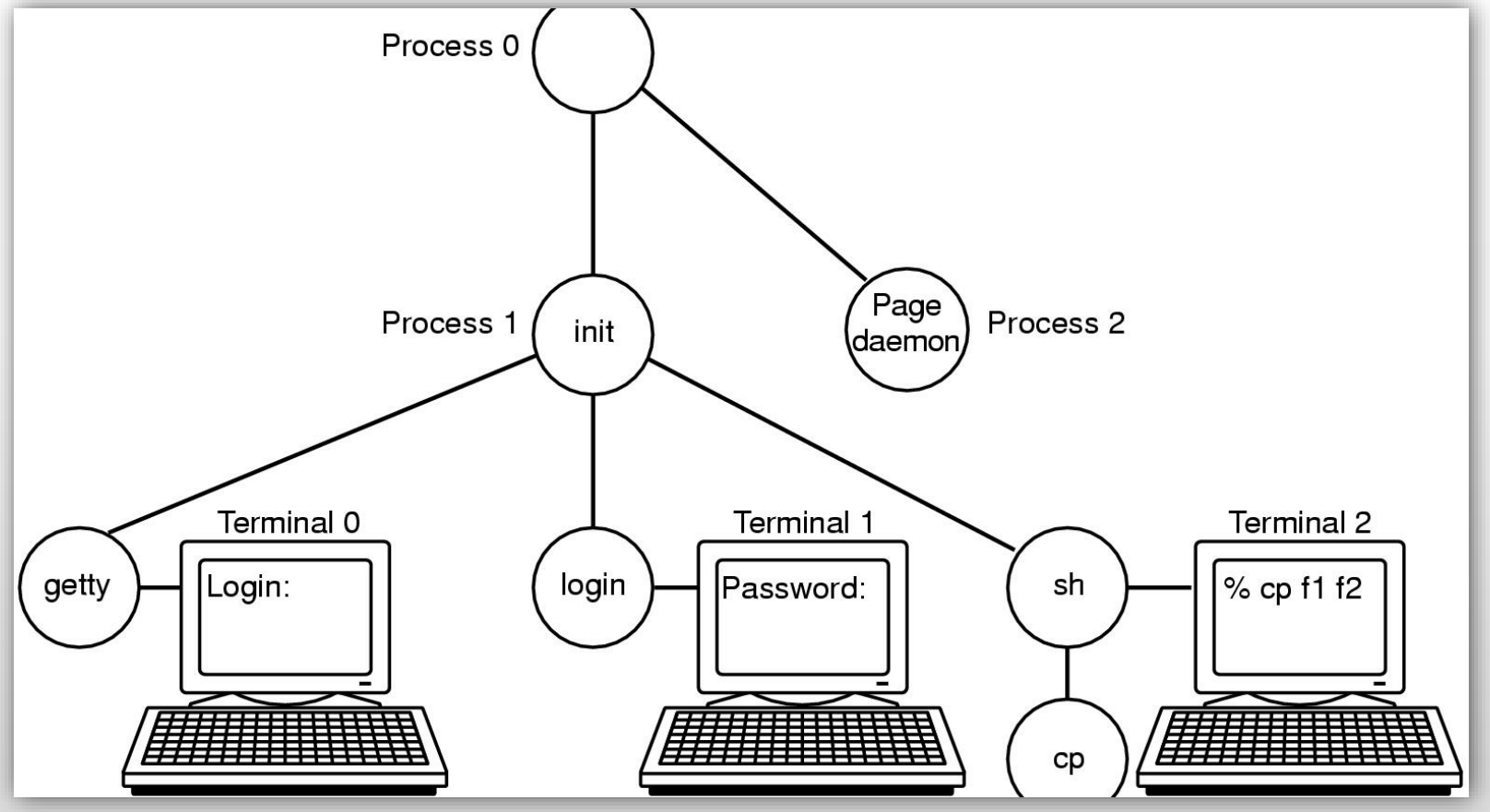


Linux Çizelgeleyici



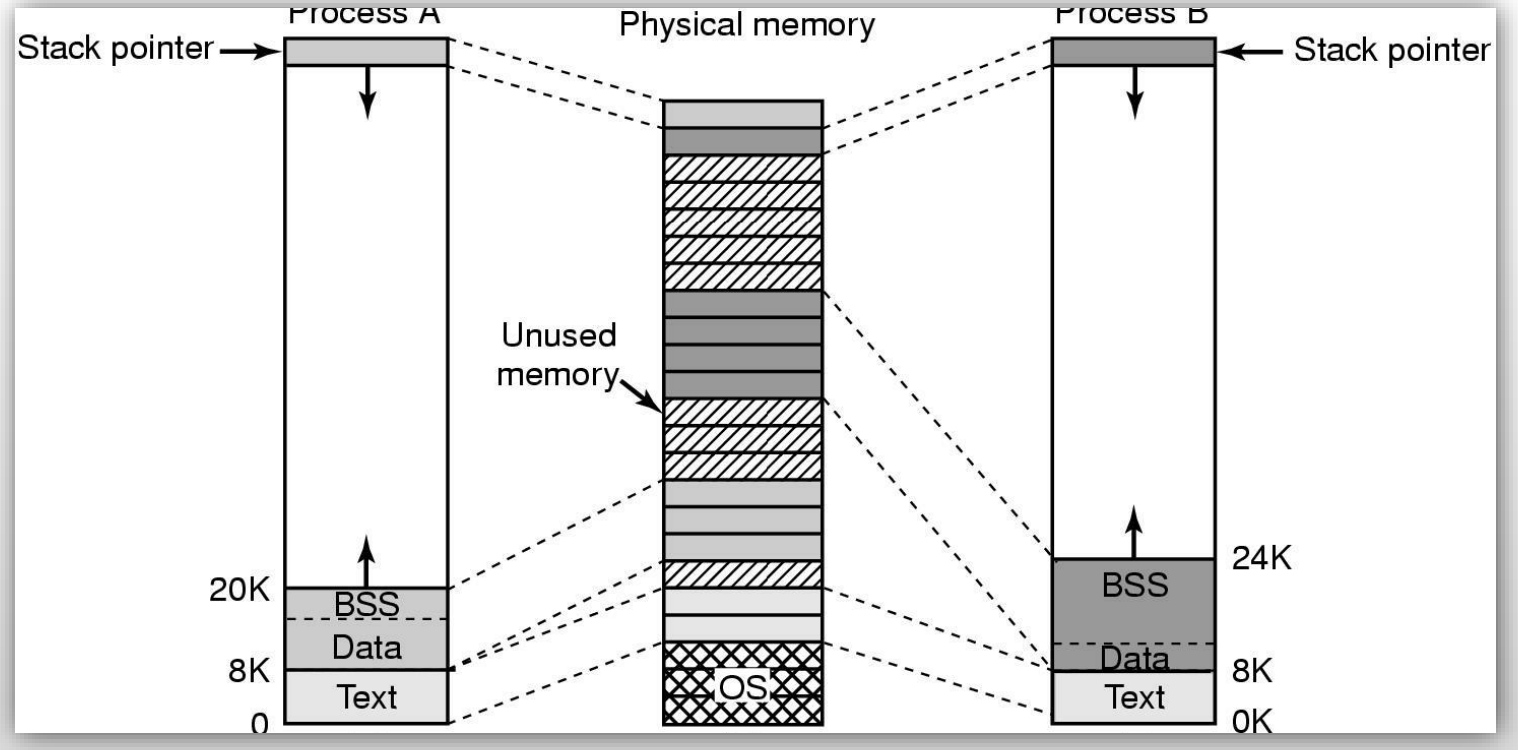
Önyükleme (boot)

- .

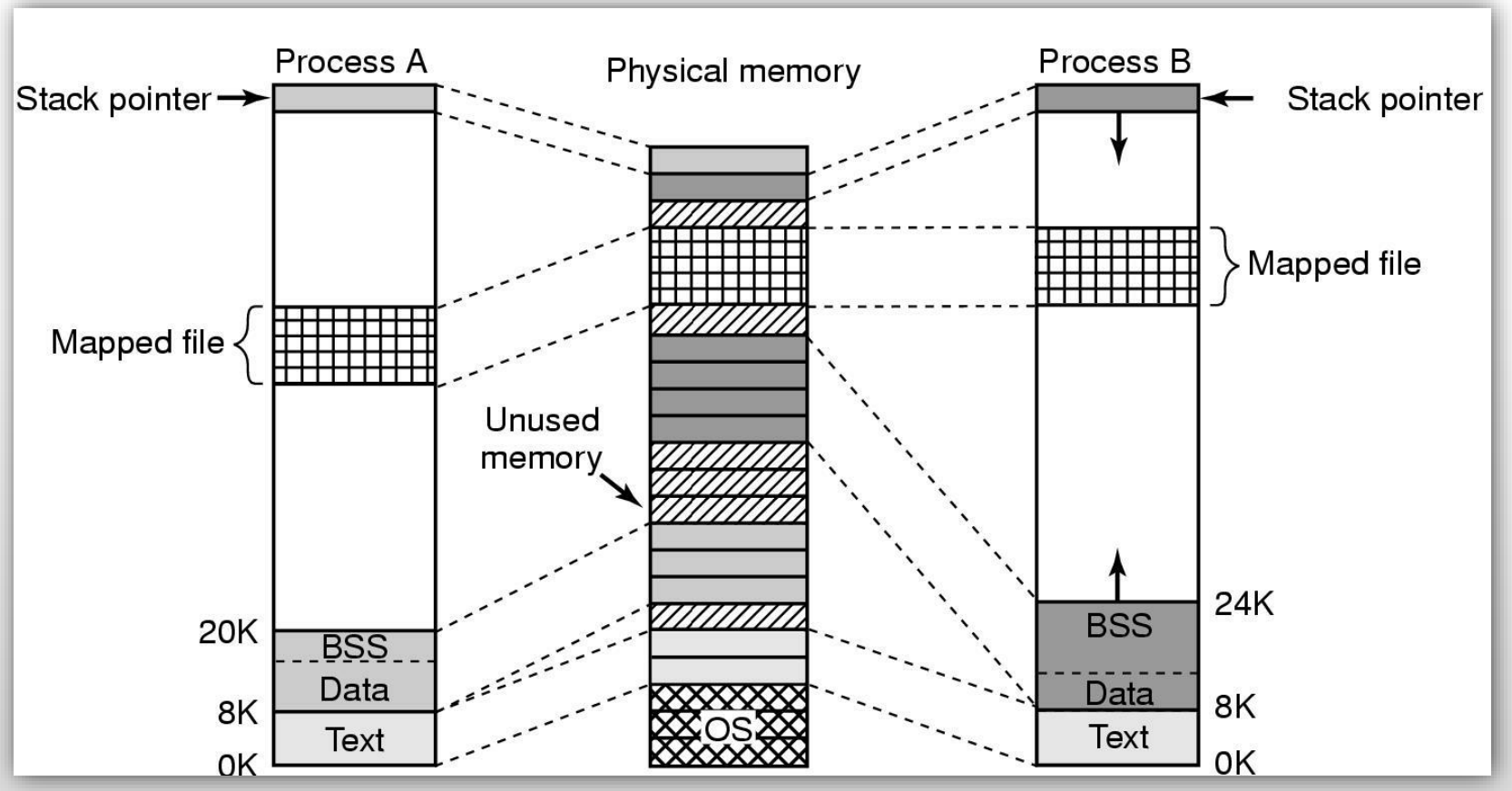


Bellek Yönetimi

- Süreç *A* ve *B*'nin sanal adres alanı ve fiziksel bellek.



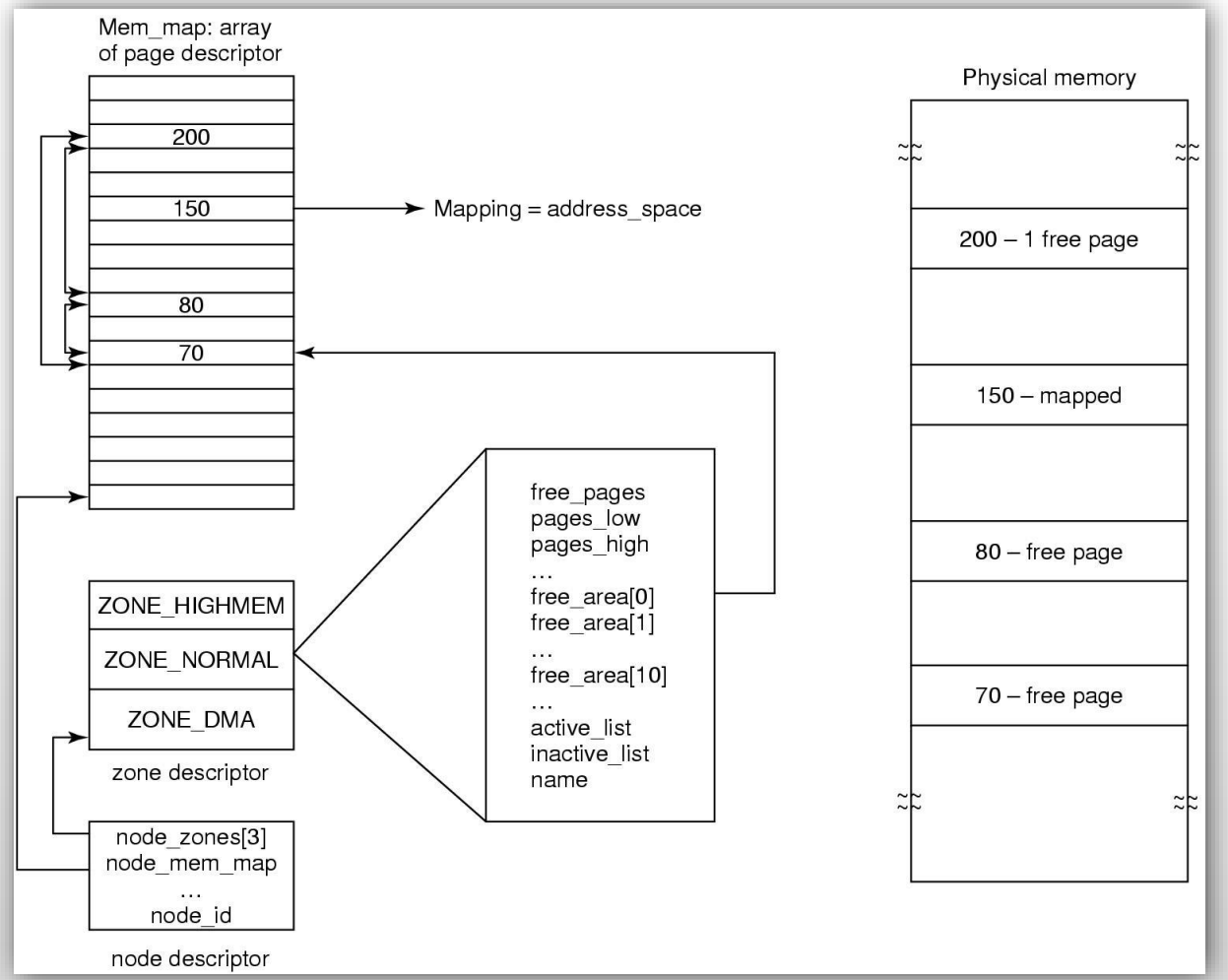
Paylaşımlı Dosya (Mapped File)



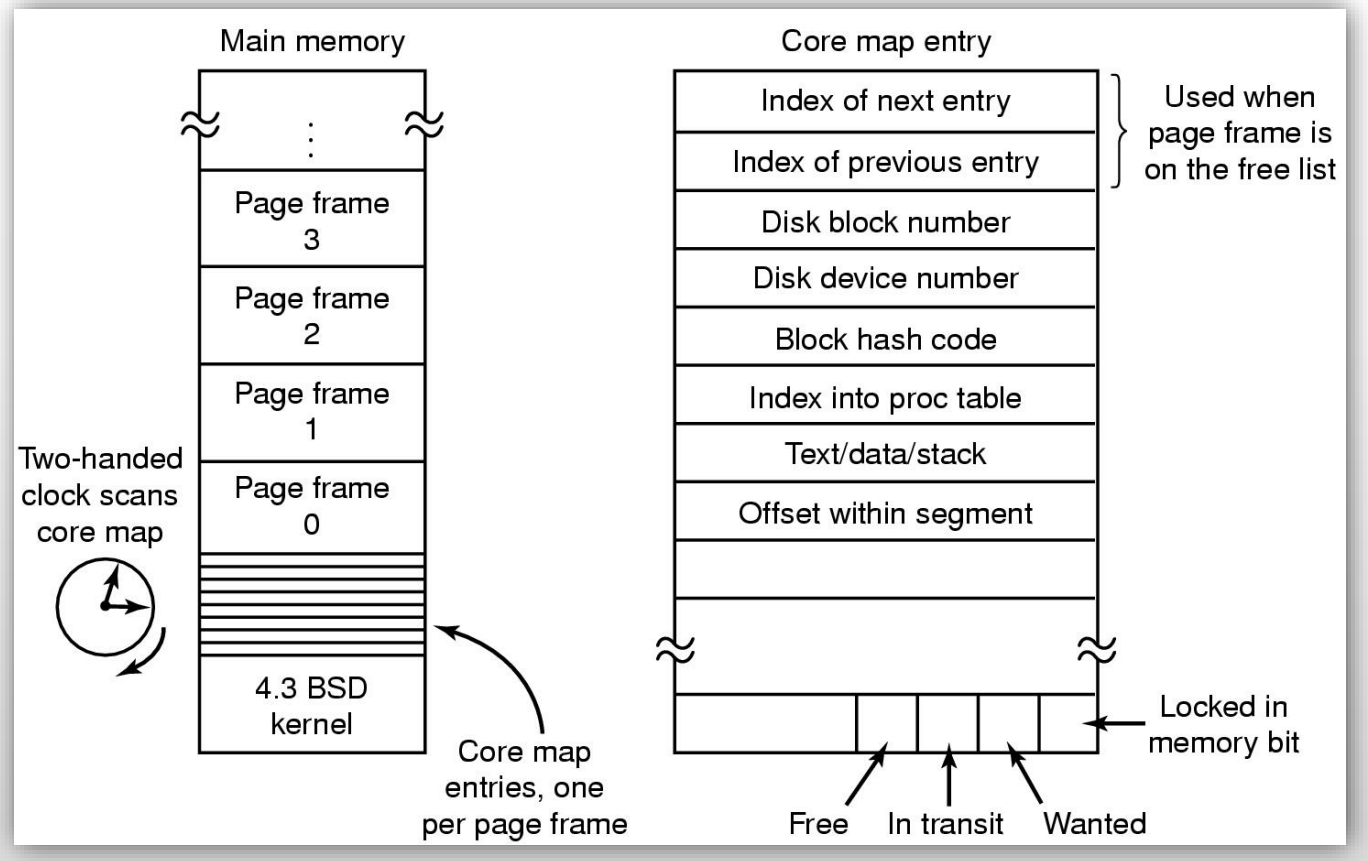
Bellek Yönetimi Sistem Çağrıları

- **brk**:
  - veri (*data*) kesiminin boyutunu artırmak veya azaltmak için kullanılır.
- **mmap**:
  - bir dosyayı veya aygıtı belleğe eşler.
  - doğrudan erişim için bir dosyayı belleğe eşlemek için kullanılır.
- **unmap**:
  - mmap kullanılarak ayrılan belleği serbest bırakır.
  - artık ihtiyaç duyulmayan belleği sisteme geri döndürür.

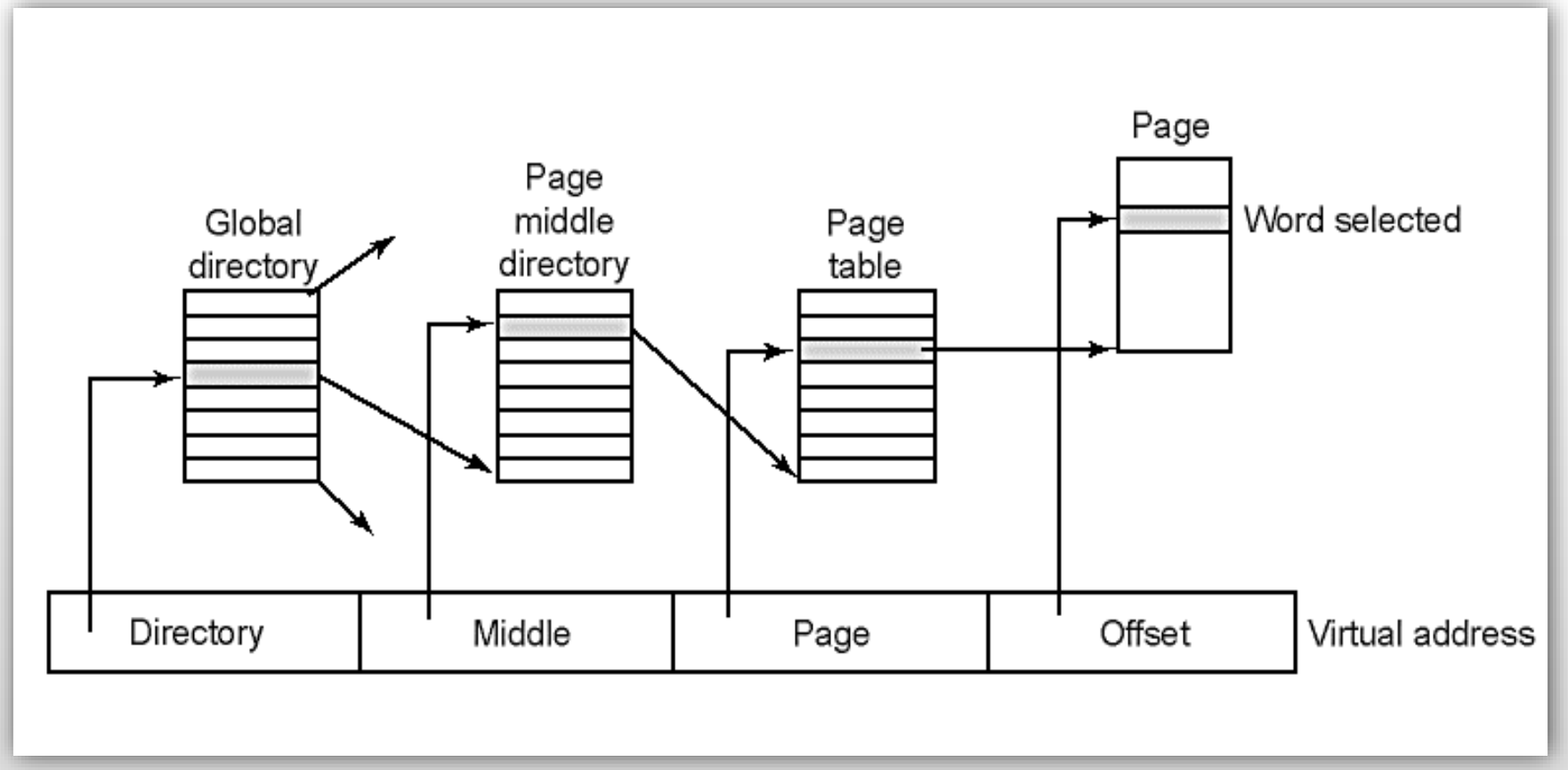
Linux Ana Bellek Gösterimi



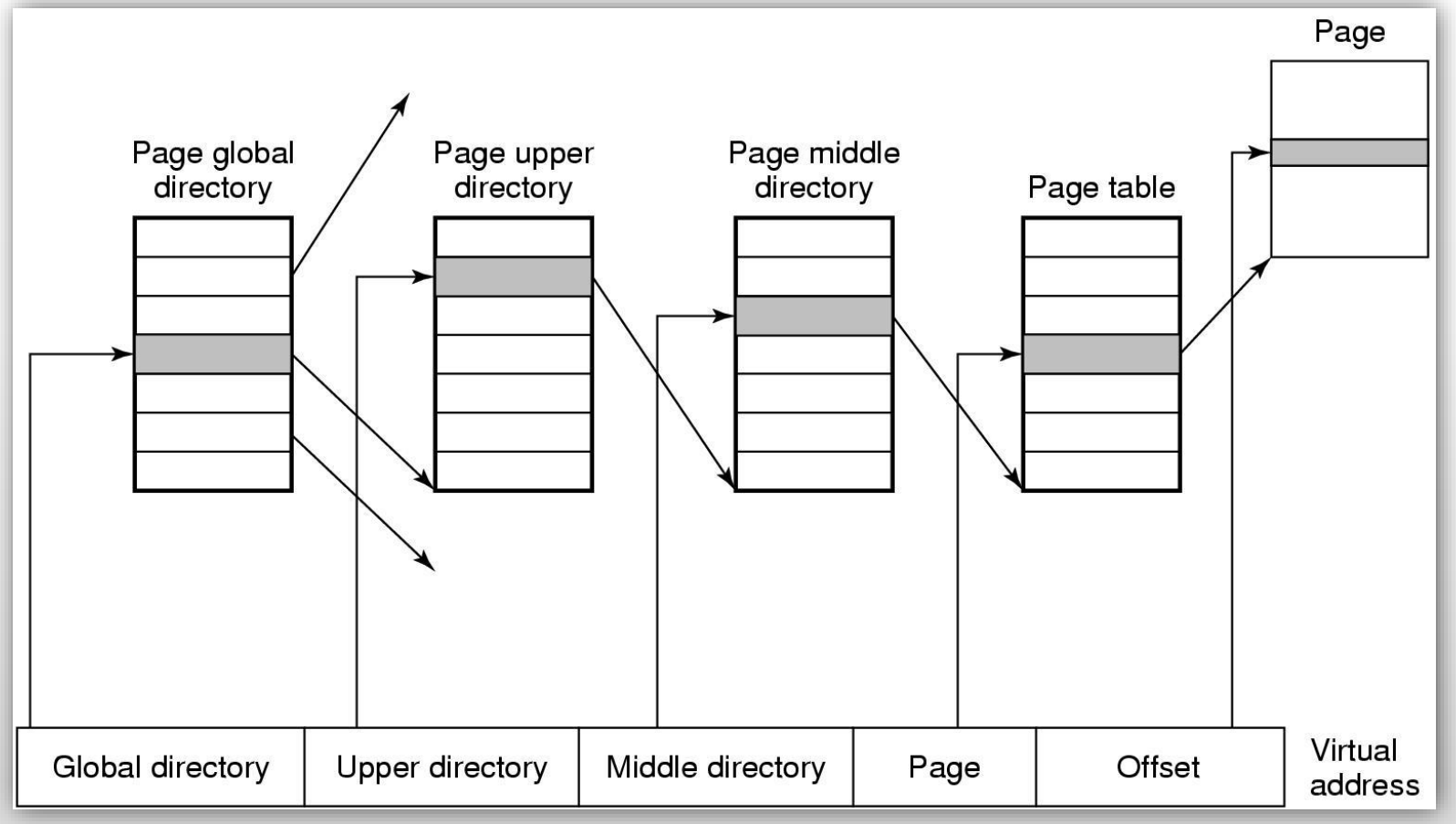
Sayfalama (Paging)



Üç Düzeyli Sayfa Tablosu (Page Table)



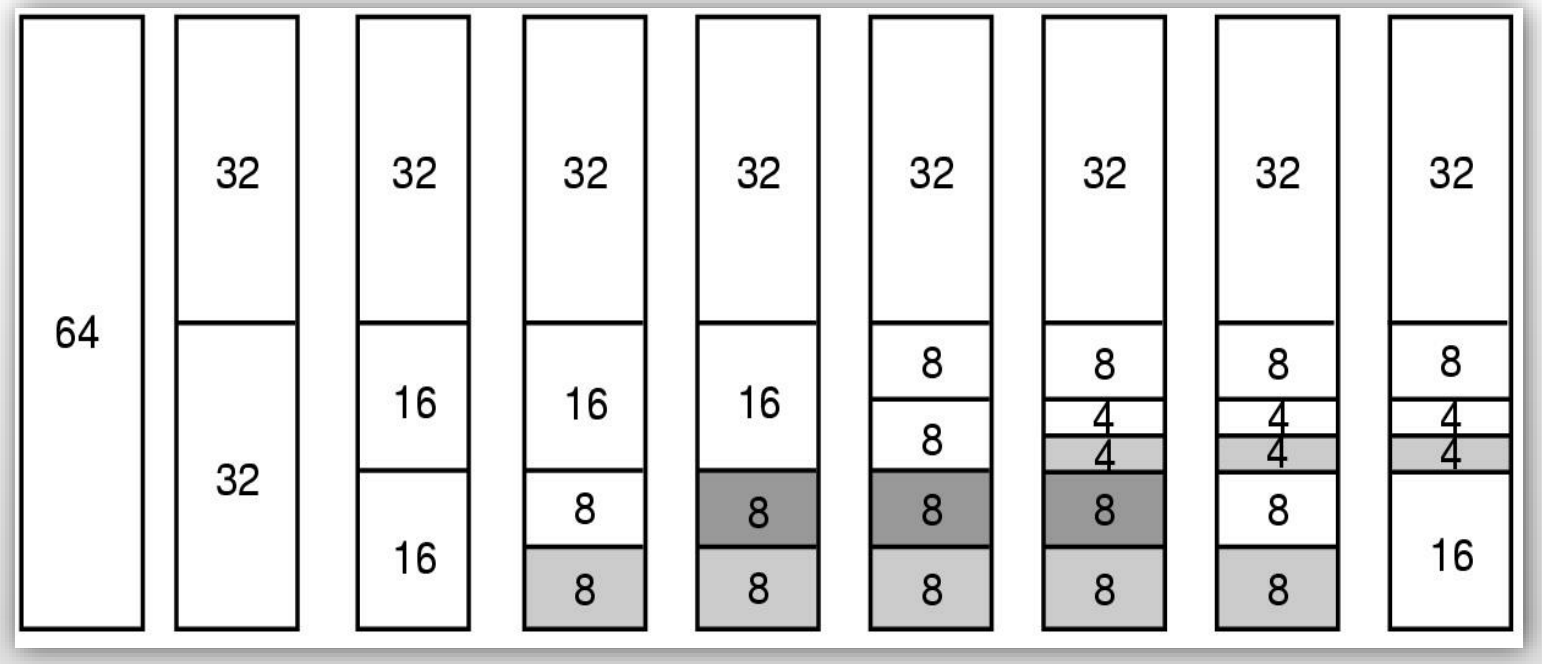
Dört Düzeyli Sayfa Tablosu (Page Table)



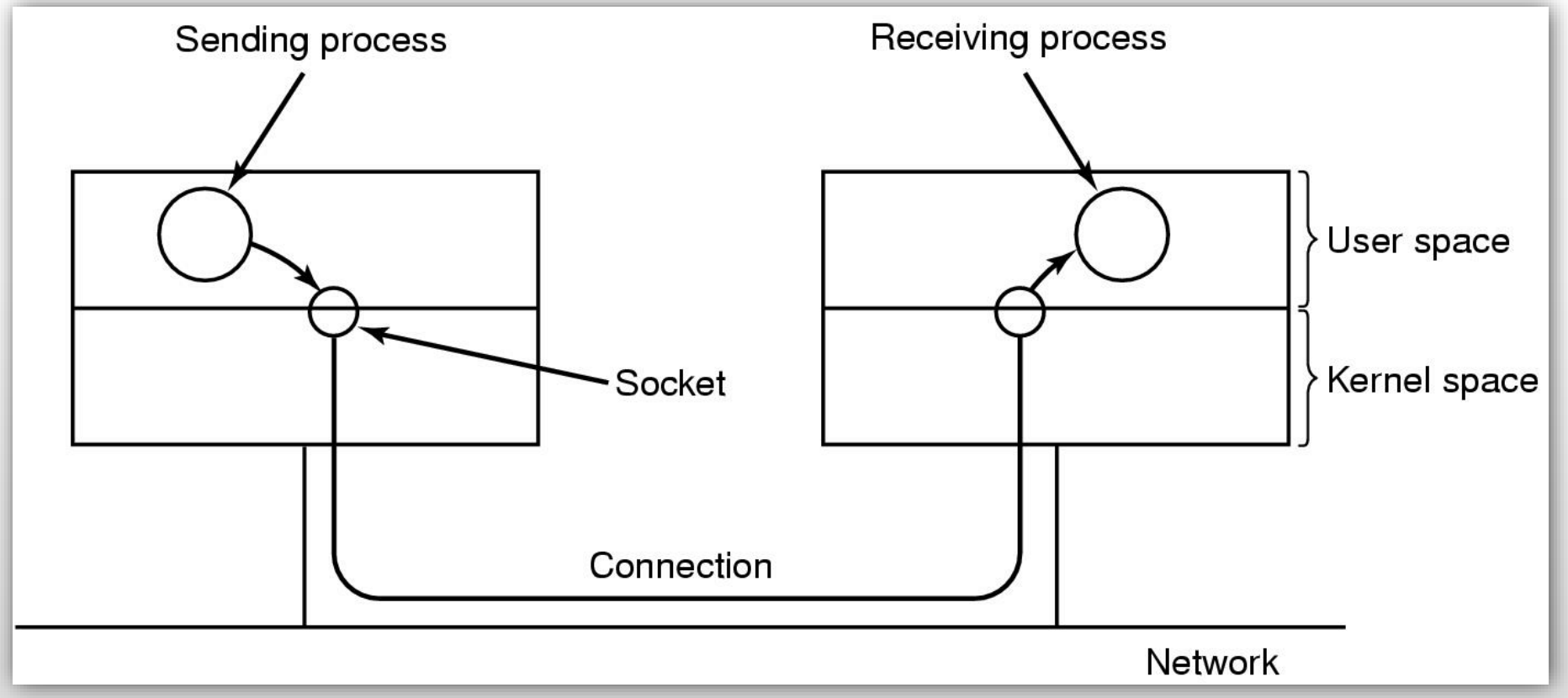
Buddy Algoritması

- Belleği verimli yönetebilmek için dinamik bellek tahsis stratejisi.
- Belleği eşit boyutlu bloklara böler,
  - ikili ağaç yapısında her düğüm, bir bellek bloğunu temsil eder.
- Bellek tahsis edileceğinde, en küçük blok aranır ve iki parçaya bölünür.
- Bellek serbest bırakıldığında,
  - bitişik bloklar kontrol edilir, daha büyük bir blok oluşturulur.
- Bloklar birleştirilerek, harici parçalanma azaltılır.
- İkili ağaç yapısı kullanılır, en küçük bellek bloğu hızlıca bulunur.

Buddy Algoritması



Ağ Haberleşmesi – Socket Kullanımı



Terminal G/Ç Yönetimi

- **cfsetospeed**: çıkış baud hızını ayarlar. (*output baud rate*)
- **cfsetispeed**: giriş baud hızını ayarlar. (*input baud rate*)
- **cfgetospeed**: çıkış baud hızını alır.
- **cfgetispeed**: giriş baud hızını alır.
- **tcsetattr**: Bir terminal ile ilişkili parametreleri değiştirir.
- **tcgetattr**: Bir terminal ile ilişkili parametreleri alır.

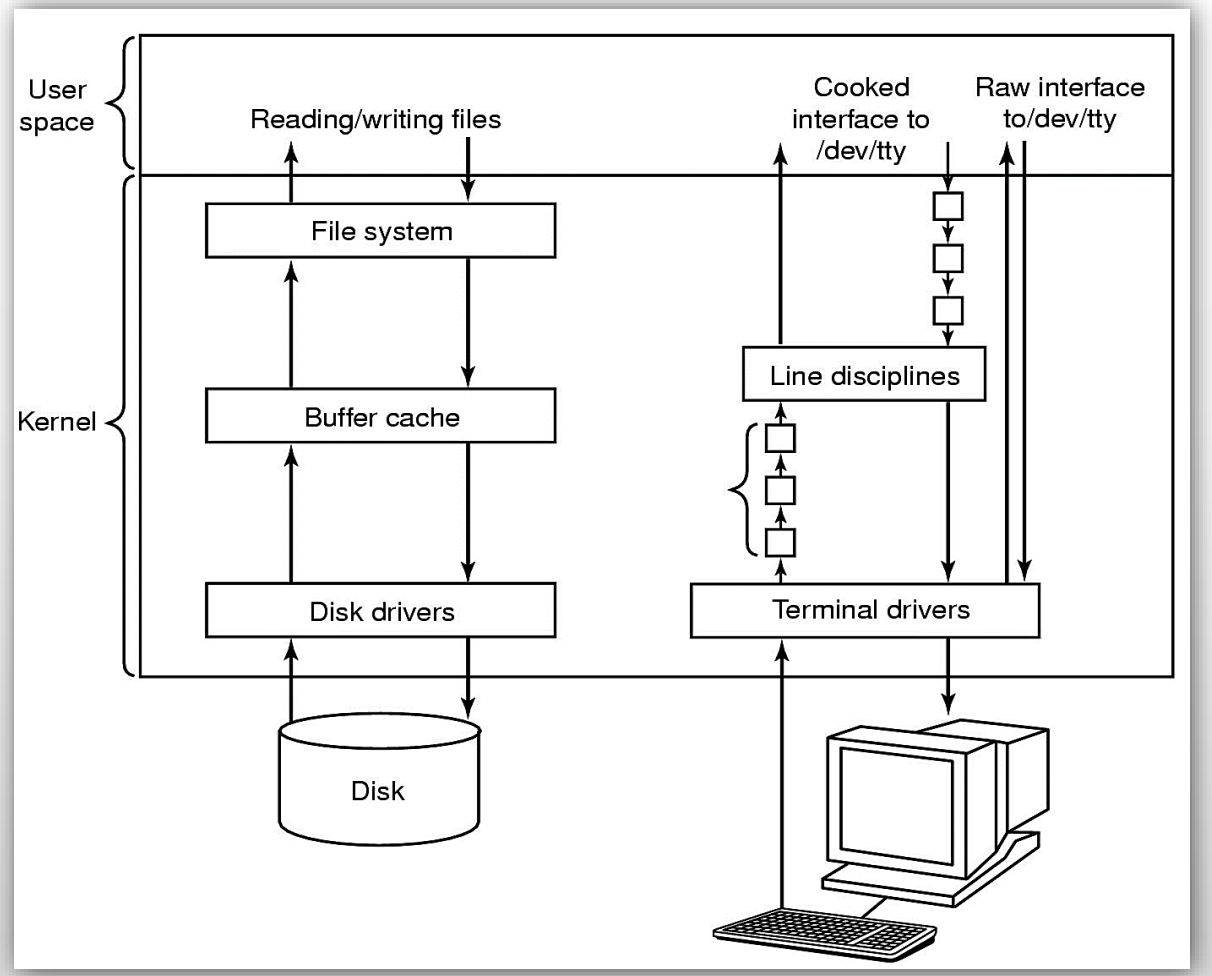
UNIX G/Ç

- **mem_read, mem_write** - bellekten okur, belleğe yazar.
- **k_open** - klavye aygıtını açar.
- **k_close** - klavye aygıtını kapatır.
- **k_read** - klavyeden veri okur.
- **k_ioctl** - klavyenin davranışını kontrol eder.
- **tty_open** - *tty* aygıtı açar.
- **tty_close** - *tty* aygıtını kapatır.
- **tty_read** - *tty* aygıtından veri okur.
- **tty_write** - *tty* aygıtına veri yazar.
- **tty_ioctl** - tty aygıtının davranışını kontrol eder.

UNIX G/Ç

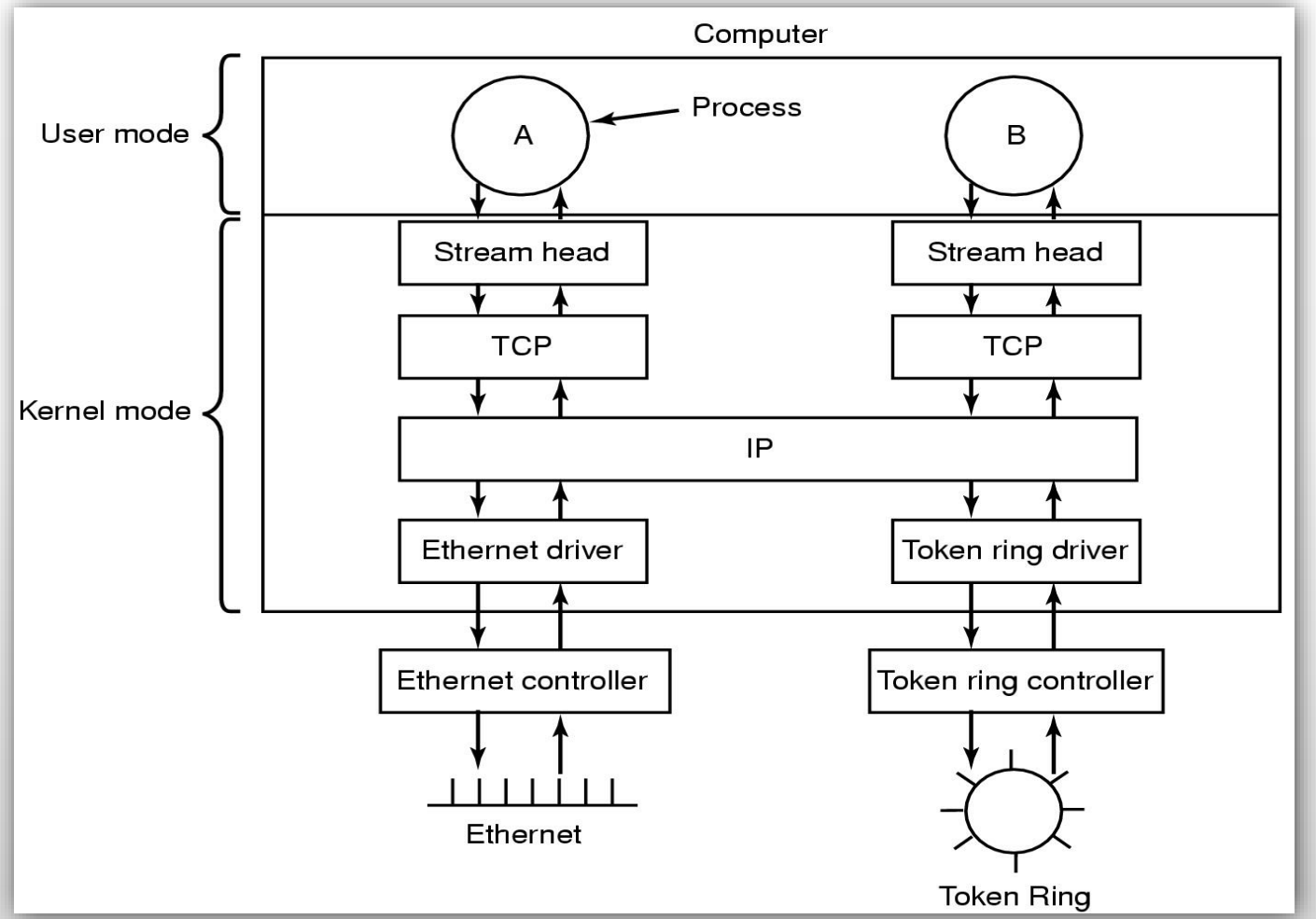
- **pr_open** - yazıcı aygıtını açar.
- **pr_close** - yazıcı aygıtını kapatır.
- **pr_write** - yazıcı aygıtına veri yazar.
- **pr_ioctl** - yazıcı aygıtının davranışını kontrol eder.
- **ip_open** - IP aygıtı açar.
- **ip_close** - IP aygıtını kapatır.
- **ip_write** - IP aygıtına veri yazar.
- **ip_ioctl** - IP aygıtının davranışını kontrol eder.

UNIX G/Ç Sistemi – BSD

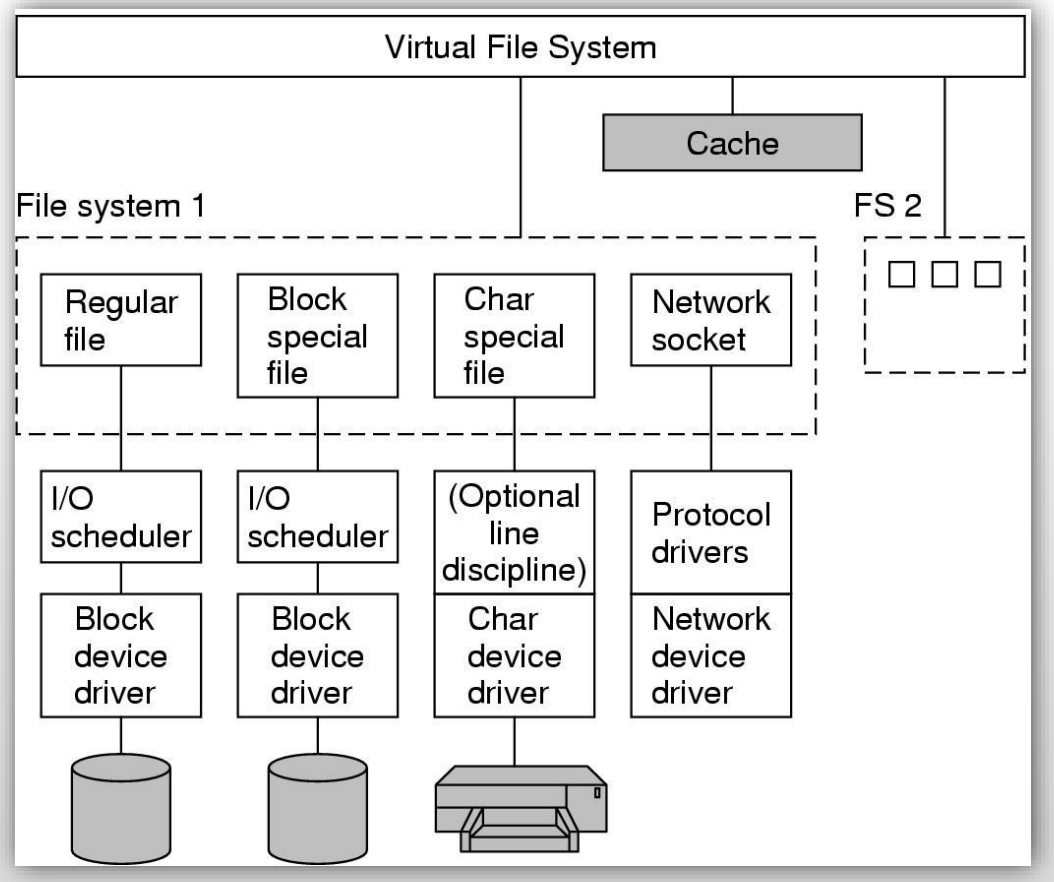


Akışlar (Streams) – System V

- .



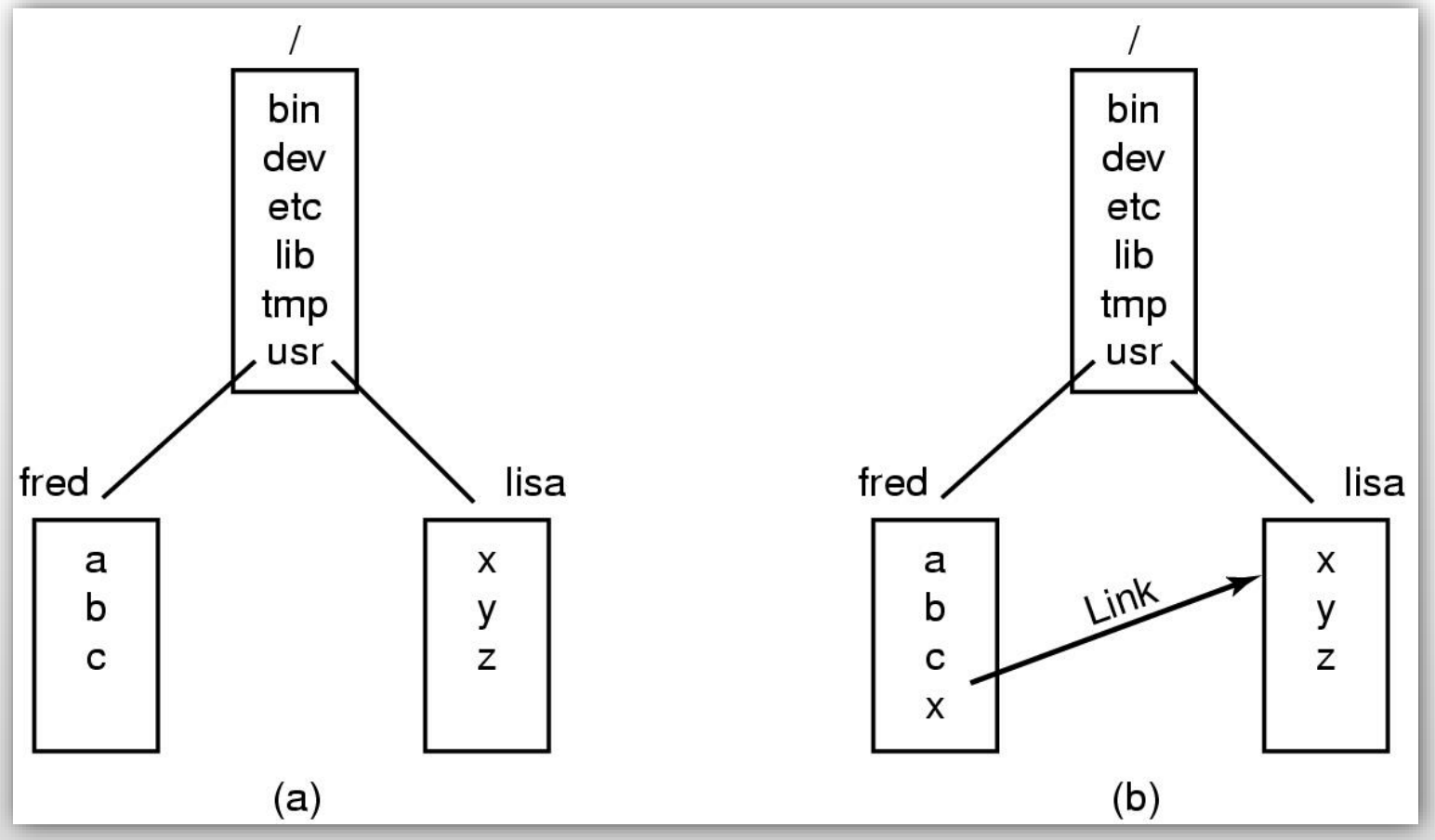
Linux G/Ç Sistemi



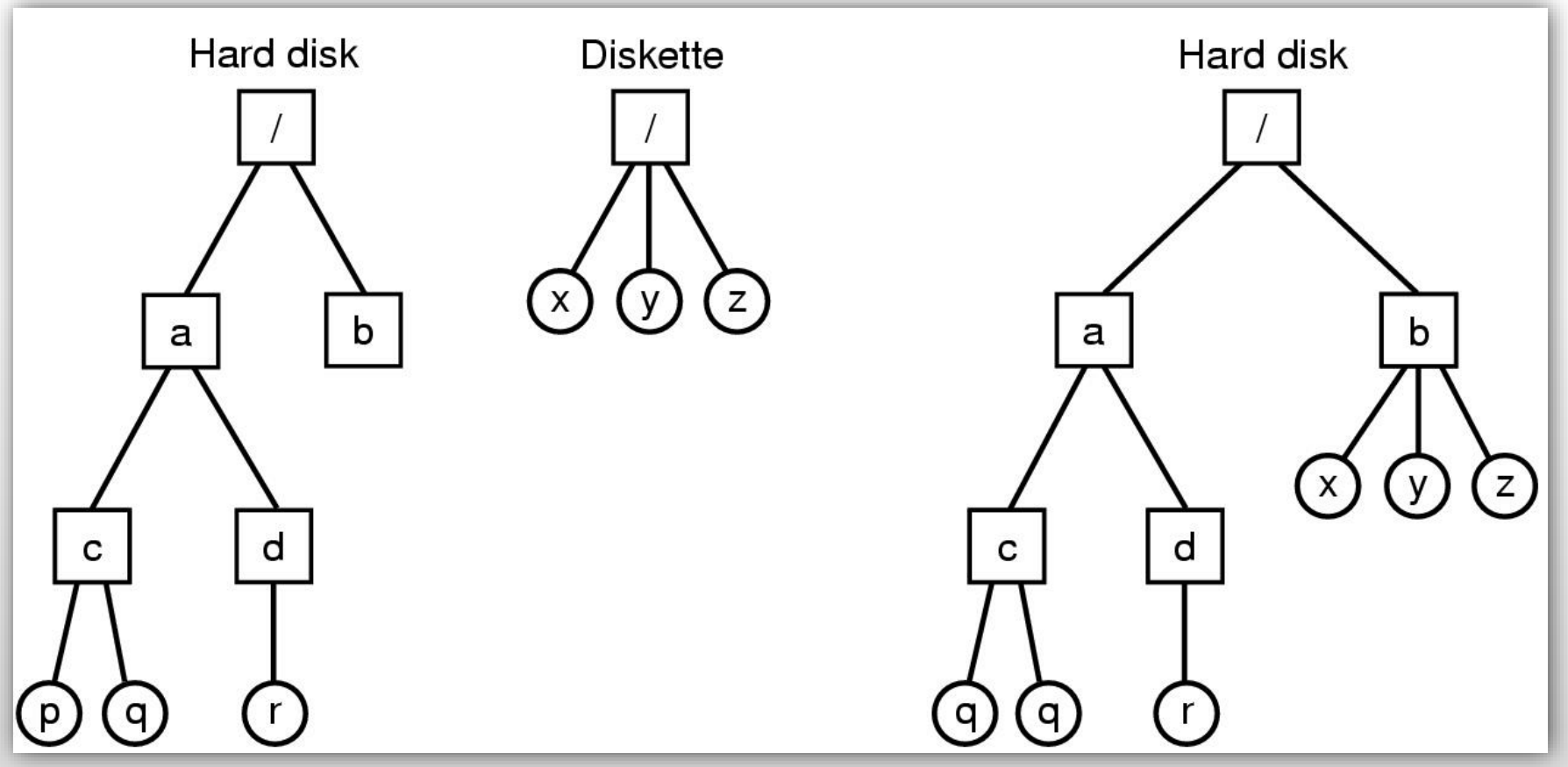
Önemli Dizinler (Klasör)

- **bin**: temel görevleri gerçekleştiren yürütülebilir dosyaları içerir.
- **dev**: donanım aygıtlarını temsil eden aygıt dosyalarını içerir.
- **etc**: uygulamalar için yapılandırma dosyalarını içerir.
- **lib**: yürütülebilir dosyaların kullandığı kütüphane dosyalarını içerir.
- **usr**: kullanıcı ile ilgili yürütülebilir dosyaları, kütüphane dosyalarını ve paylaşılan dosyaları içeren *bin*, *lib* ve *share* dizinleri içerir.

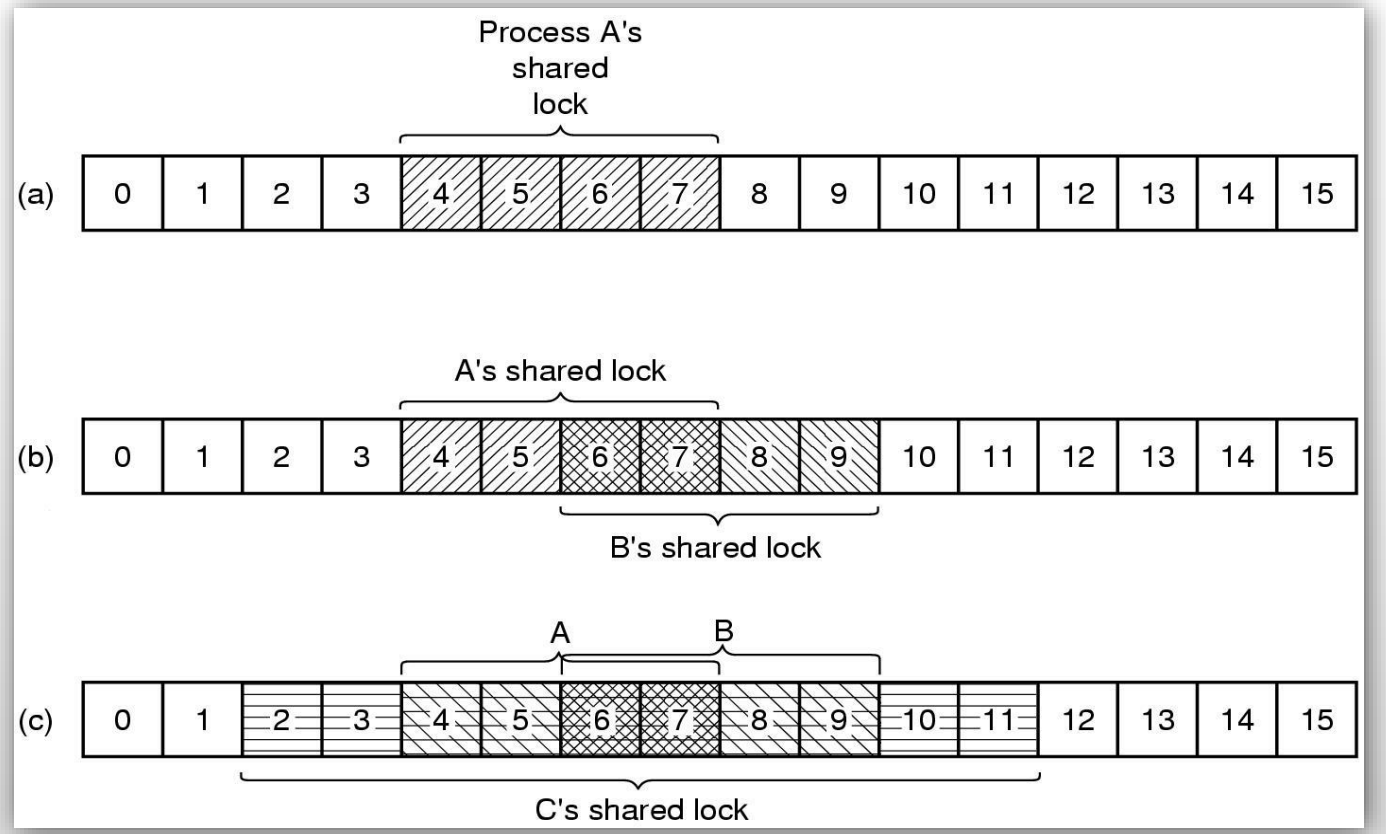
Dosya Sistemi



Dosya Sistemi



Dosyaların Kitlenmesi (lock)



Dosya Yönetimi Sistem Çağrıları

System call	Description
fd = creat(name, mode)	One way to create a new file
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
s = fstat(fd, &buf)	Get a file's status information
s = pipe(&fd[0])	Create a pipe
s = fcntl(fd, cmd, ...)	File locking and other operations

Istat Sistem Çağrısı Dönüş Değerleri

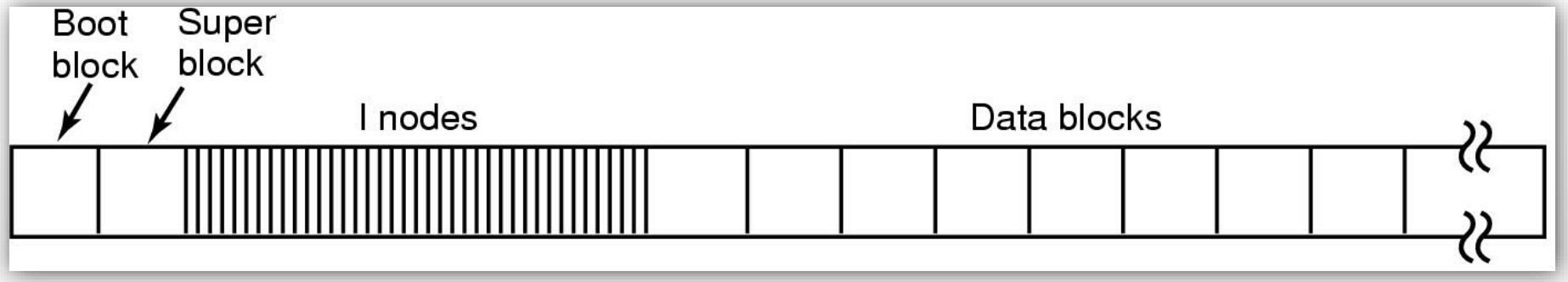
- .

Device the file is on
I-node number (which file on the device)
File mode (includes protection information)
Number of links to the file
Identity of the file's owner
Group the file belongs to
File size (in bytes)
Creation time
Time of last access
Time of last modification

Dizin Yönetimi Sistem Çağrıları

System call	Description
s = mkdir(path, mode)	Create a new directory
s = rmdir(path)	Remove a directory
s = link(oldpath, newpath)	Create a link to an existing file
s = unlink(path)	Unlink a file
s = chdir(path)	Change the working directory
dir = opendir(path)	Open a directory for reading
s = closedir(dir)	Close a directory
dirent = readdir(dir)	Read one directory entry
rewinddir(dir)	Rewind a directory so it can be reread

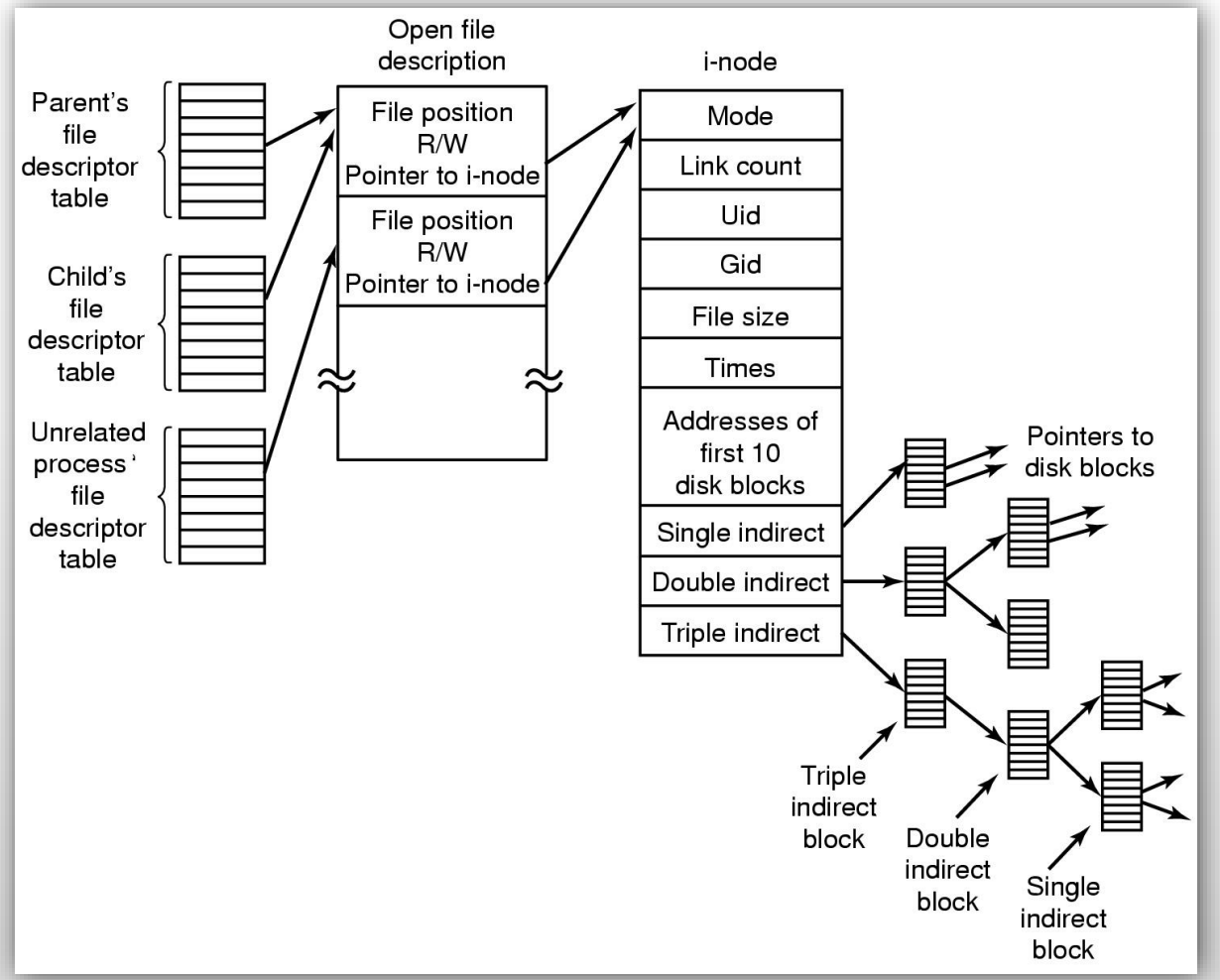
UNIX'te Disk Düzeni



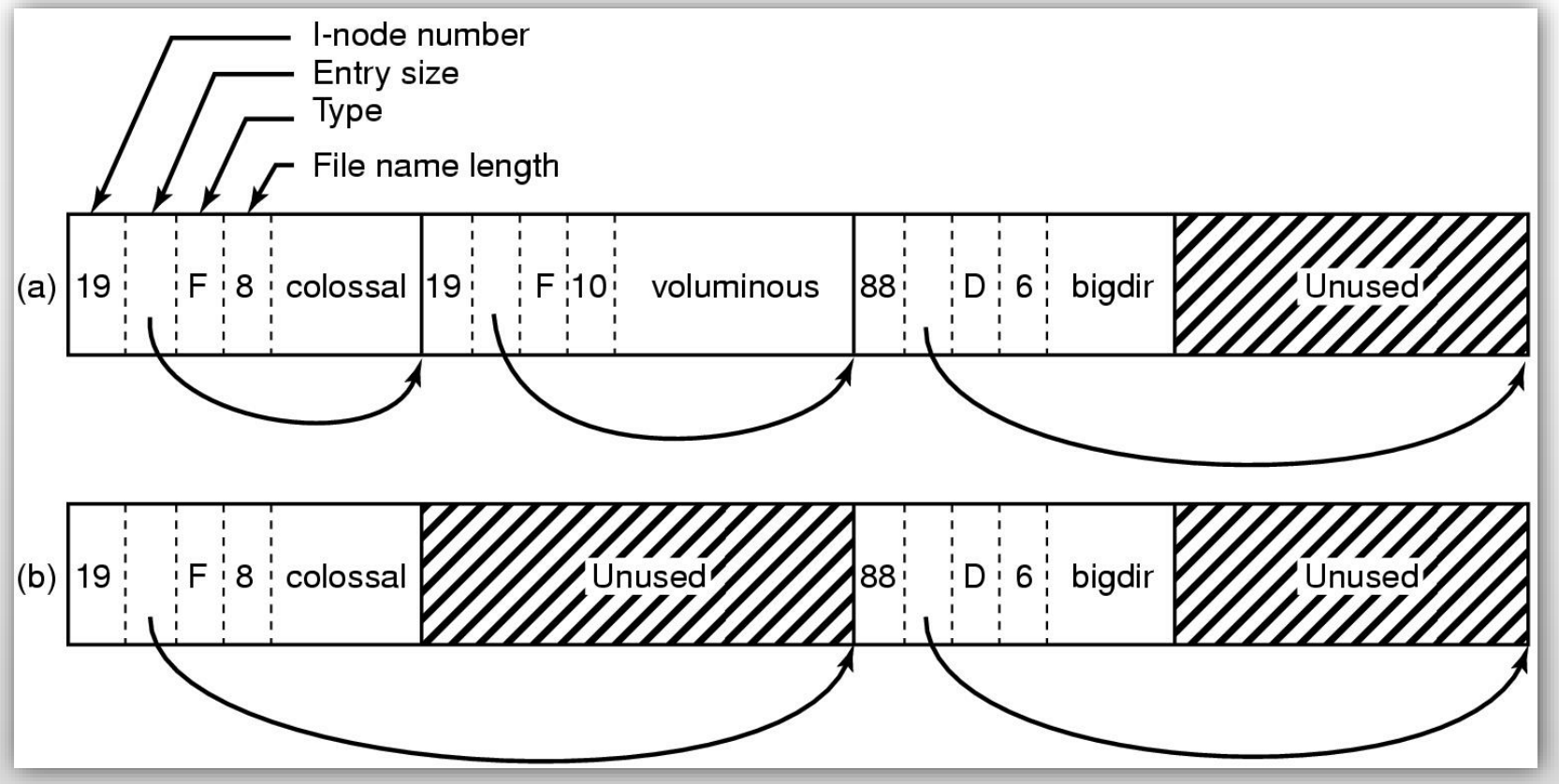
I-node Yapısı

Field	Bytes	Description
Mode	2	File type, protection bits, setuid, setgid bits
Nlinks	2	Number of directory entries pointing to this i-node
Uid	2	UID of the file owner
Gid	2	GID of the file owner
Size	4	File size in bytes
Addr	39	Address of first 10 disk blocks, then 3 indirect blocks
Gen	1	Generation number (incremented every time i-node is reused)
Atime	4	Time the file was last accessed
Mtime	4	Time the file was last modified
Ctime	4	Time the i-node was last changed (except the other times)

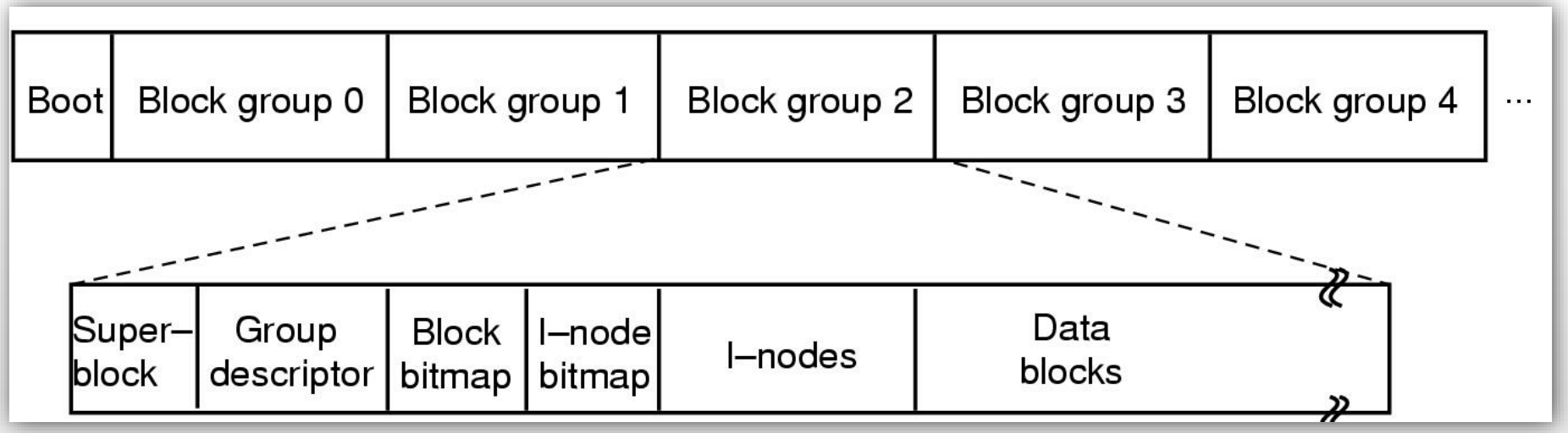
Dosya Tanımlayıcı ve Açık Dosyalar Tablosu



Üç Dosya İçeren BSD Dizini

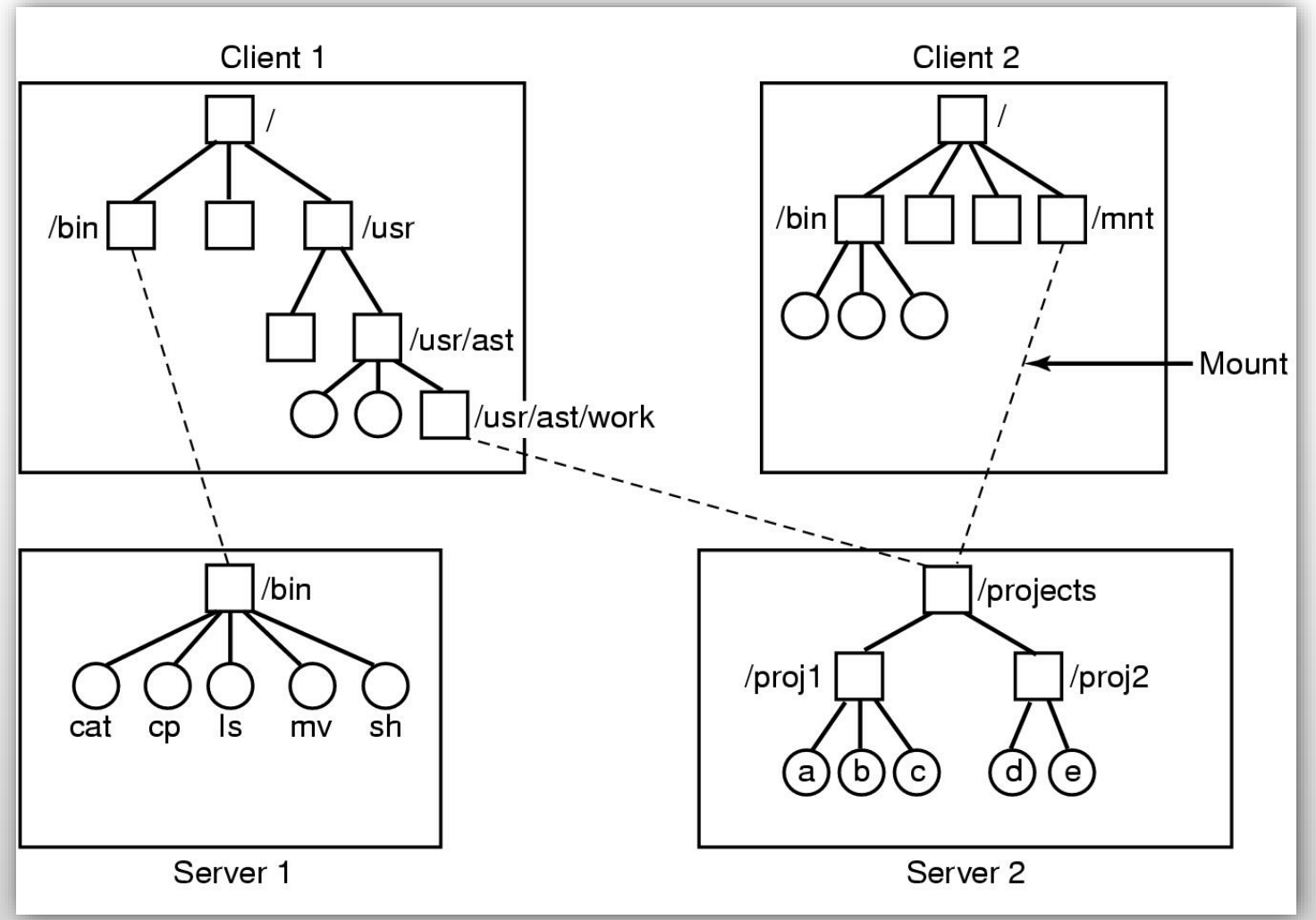


Linux Ext2 Dosya Sistemi Düzeni

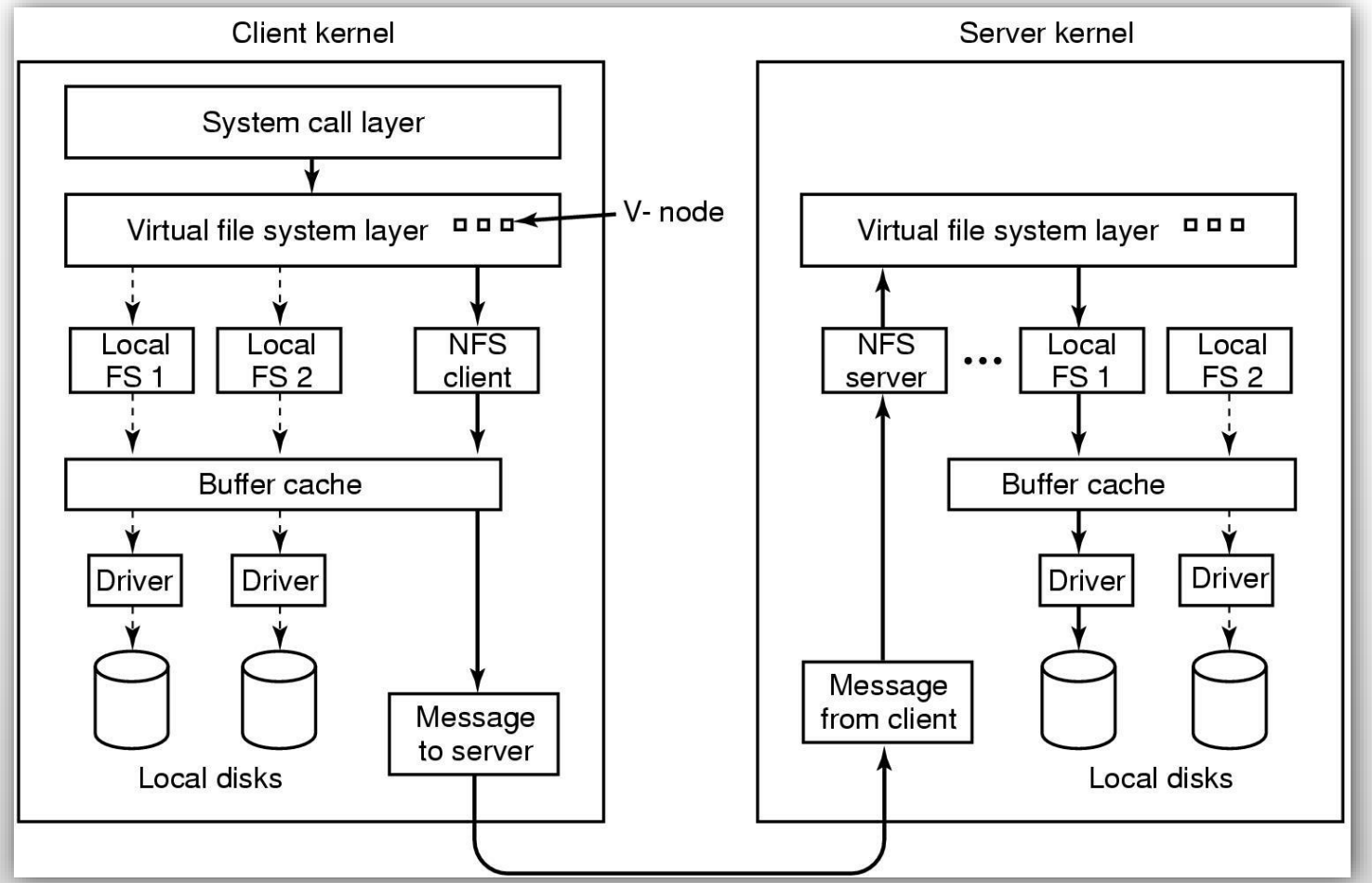


Ağ Dosya Sistemi (NFS)

- .



Ağ Dosya Sistemi (NFS) Katman Yapısı



Dosya Koruma İzinleri

- .

Binary	Symbolic	Allowed file accesses
111000000	rwx------	Owner can read, write, and execute
111111000	rwxrwx---	Owner and group can read, write, and execute
110100000	rw-r-----	Owner can read and write; group can read
110100100	rw-r--r--	Owner can read and write; all others can read
111101101	rwxr-xr-x	Owner can do everything, rest can read and execute
000000000	---------	Nobody has any access
000000111	------rwx	Only outsiders have access (strange, but legal)

Dosya Koruma Sistem Çağrıları

System call	Description
s = chmod(path, mode)	Change a file's protection mode
s = access(path, mode)	Check access using the real UID and GID
uid = getuid( )	Get the real UID
uid = geteuid( )	Get the effective UID
gid = getgid( )	Get the real GID
gid = getegid( )	Get the effective GID
s = chown(path, owner, group)	Change owner and group
s = setuid(uid)	Set the UID
s = setgid(gid)	Set the GID

SON