

# Bölüm 11: Linux

## İşletim Sistemleri

# Tarihçe

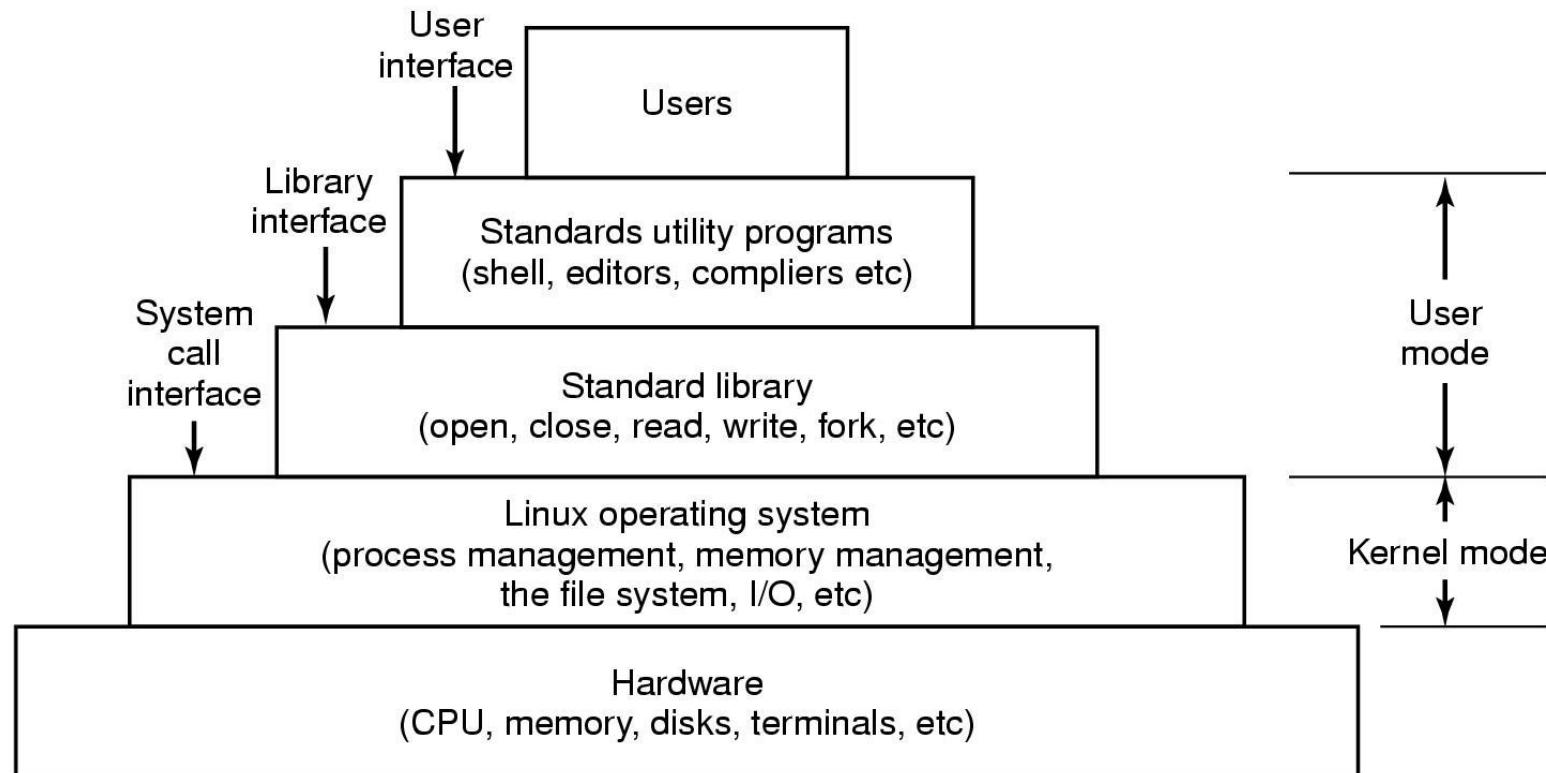
- UNICS
- PDP-11 UNIX
- Portable UNIX
- Berkeley UNIX
- Standard UNIX
- MINIX
- Linux

# Hedefler

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

# UNIX

- .
- .



# UNIX Yardımcı Programları

- .

Program	Typical use
cat	Concatenate multiple files to standard output
chmod	Change file protection mode
cp	Copy one or more files
cut	Cut columns of text from a file
grep	Search a file for some pattern
head	Extract the first lines of a file
ls	List directory
make	Compile files to build a binary
mkdir	Make a directory
od	Octal dump a file
paste	Paste columns of text into a file
pr	Format a file for printing
rm	Remove one or more files
rmdir	Remove a directory
sort	Sort a file of lines alphabetically
tail	Extract the last lines of a file
tr	Translate between character sets

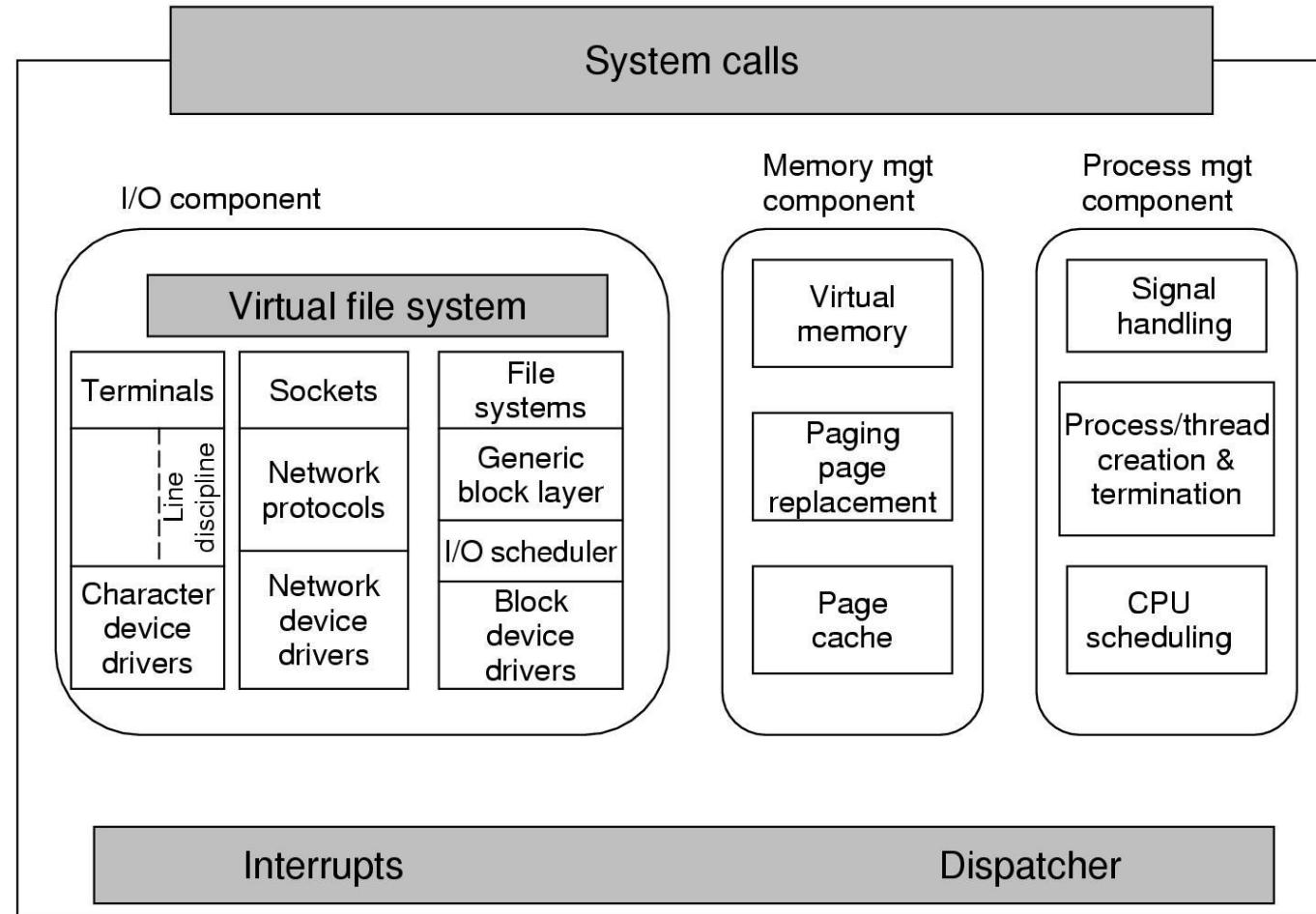
# UNIX Çekirdek

- .
- .

System calls					Interrupts and traps					
Terminal handling		Sockets	File naming	Map-ping	Page faults	Signal handling	Process creation and termination			
Raw tty	Cooked tty	Network protocols	File systems	Virtual memory						
	Line disciplines	Routing	Buffer cache	Page cache		Process scheduling				
Character devices		Network device drivers	Disk device drivers			Process dispatching				
Hardware										

# UNIX Çekirdek

- .
- .



# POSIX Sinyaller

• .

Signal	Cause
SIGABRT	Sent to abort a process and force a core dump
SIGALRM	The alarm clock has gone off
SIGFPE	A floating-point error has occurred (e.g., division by 0)
SIGHUP	The phone line the process was using has been hung up
SIGILL	The user has hit the DEL key to interrupt the process
SIGQUIT	The user has hit the key requesting a core dump
SIGKILL	Sent to kill a process (cannot be caught or ignored)
SIGPIPE	The process has written to a pipe which has no readers
SIGSEGV	The process has referenced an invalid memory address
SIGTERM	Used to request that a process terminate gracefully
SIGUSR1	Available for application-defined purposes
SIGUSR2	Available for application-defined purposes

# Süreç Yönetimi için Sistem Çağrıları

• .

System call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, opts)	Wait for a child to terminate
s = execve(name, argv, envp)	Replace a process' core image
exit(status)	Terminate process execution and return status
s = sigaction(sig, &act, &oldact)	Define action to take on signals
s = sigreturn(&context)	Return from a signal
s = sigprocmask(how, &set, &old)	Examine or change the signal mask
s = sigpending(set)	Get the set of blocked signals
s = sigsuspend(sigmask)	Replace the signal mask and suspend the process
s = kill(pid, sig)	Send a signal to a process
residual = alarm(seconds)	Set the alarm clock
s = pause( )	Suspend the caller until the next signal

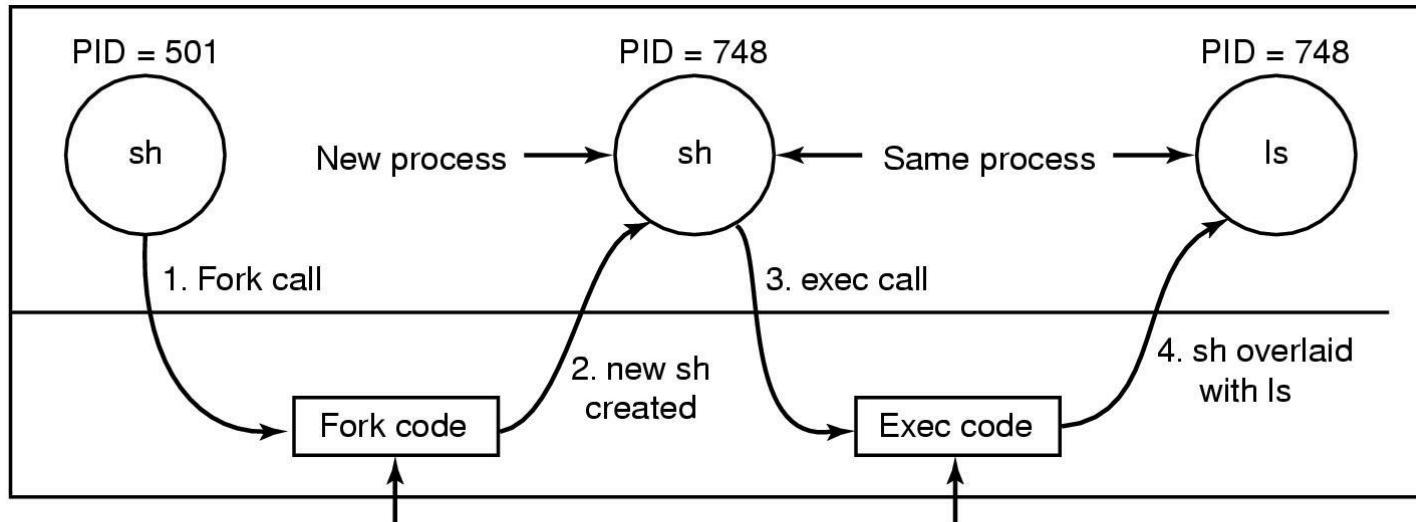
# İş Parçacıkları

- 
- 

Thread call	Description
pthread_create	Create a new thread in the caller's address space
pthread_exit	Terminate the calling thread
pthread_join	Wait for a thread to terminate
pthread_mutex_init	Create a new mutex
pthread_mutex_destroy	Destroy a mutex
pthread_mutex_lock	Lock a mutex
pthread_mutex_unlock	Unlock a mutex
pthread_cond_init	Create a condition variable
pthread_cond_destroy	Destroy a condition variable
pthread_cond_wait	Wait on a condition variable
pthread_cond_signal	Release one thread waiting on a condition variable

# ls (list) Komutu

- .
- .



Allocate child's process table entry  
Fill child's entry from parent  
Allocate child's stack and user area  
Fill child's user area from parent  
Allocate PID for child  
Set up child to share parent's text  
Copy page tables for data and stack  
Set up sharing of open files  
Copy parent's registers to child

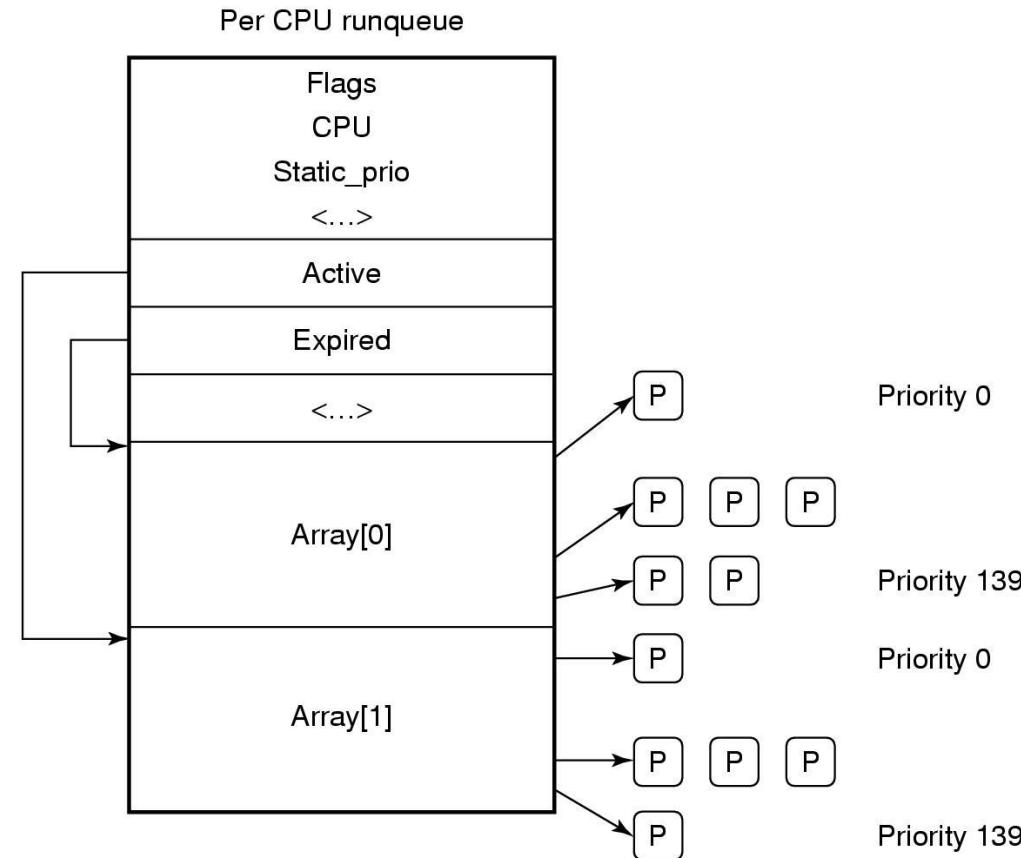
Find the executable program  
Verify the execute permission  
Read and verify the header  
Copy arguments, environ to kernel  
Free the old address space  
Allocate new address space  
Copy arguments, environ to stack  
Reset signals  
Initialize registers

# Linux Çizelgeleme

- Çizelgeleme amaçlı üç çeşit:
- Gerçek zamanlı ilk-giren ilk-çıkar (Real-time FIFO).
- Gerçek zamanlı sıralı (Real-time round robin).
- Zaman paylaşımı (Timesharing).

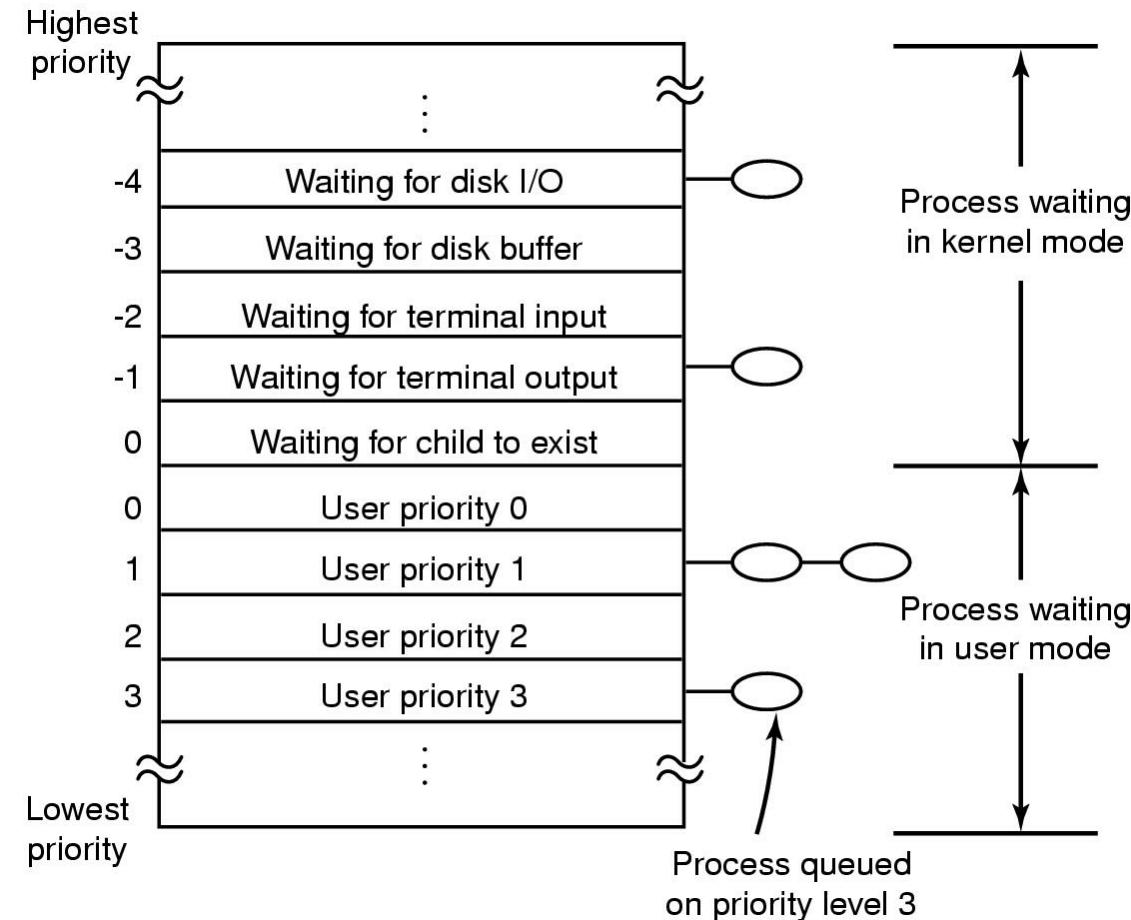
# Linux Çizelgeleme – Öncelik Dizileri

- 
- 



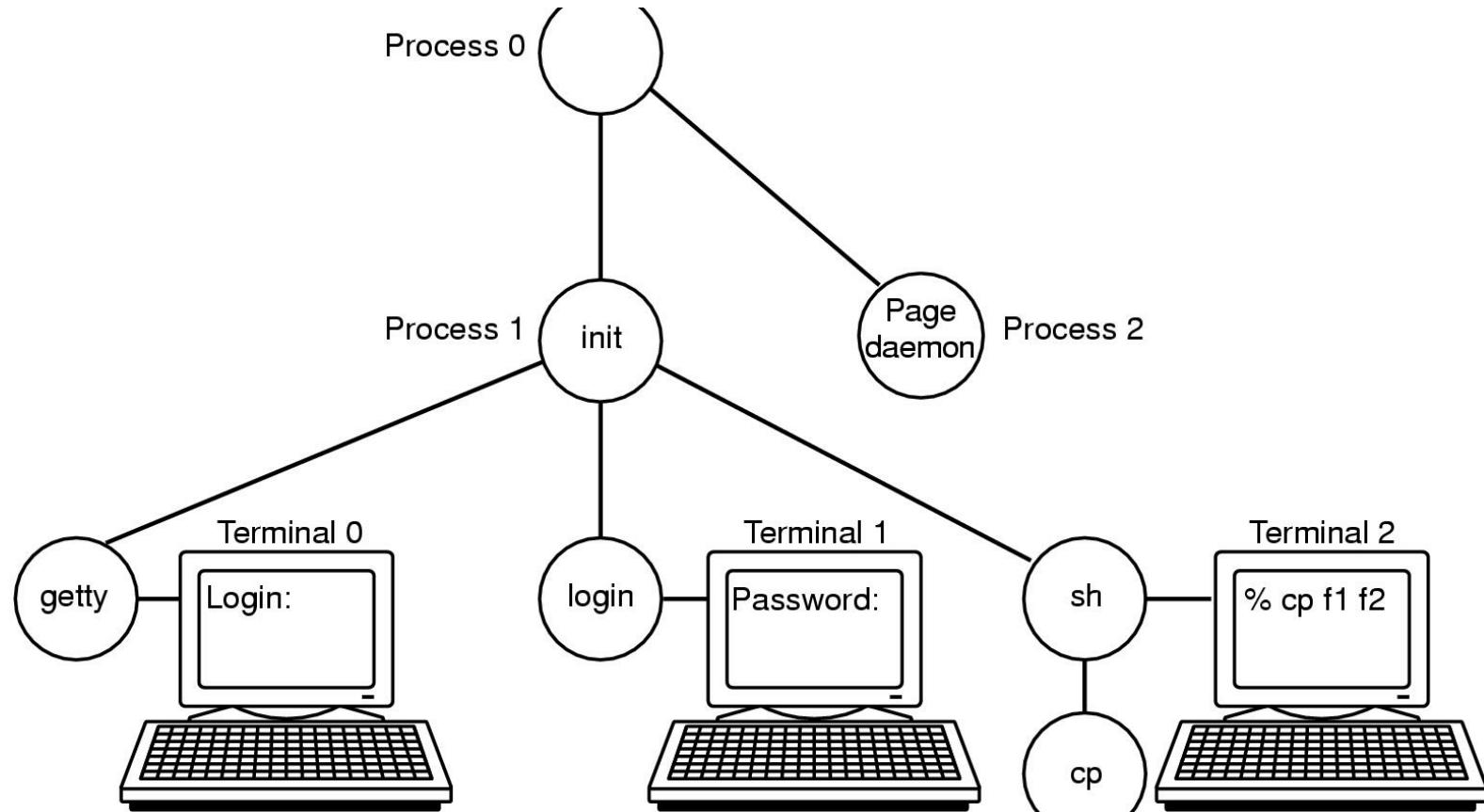
# UNIX Çizelgeleyici

- ..



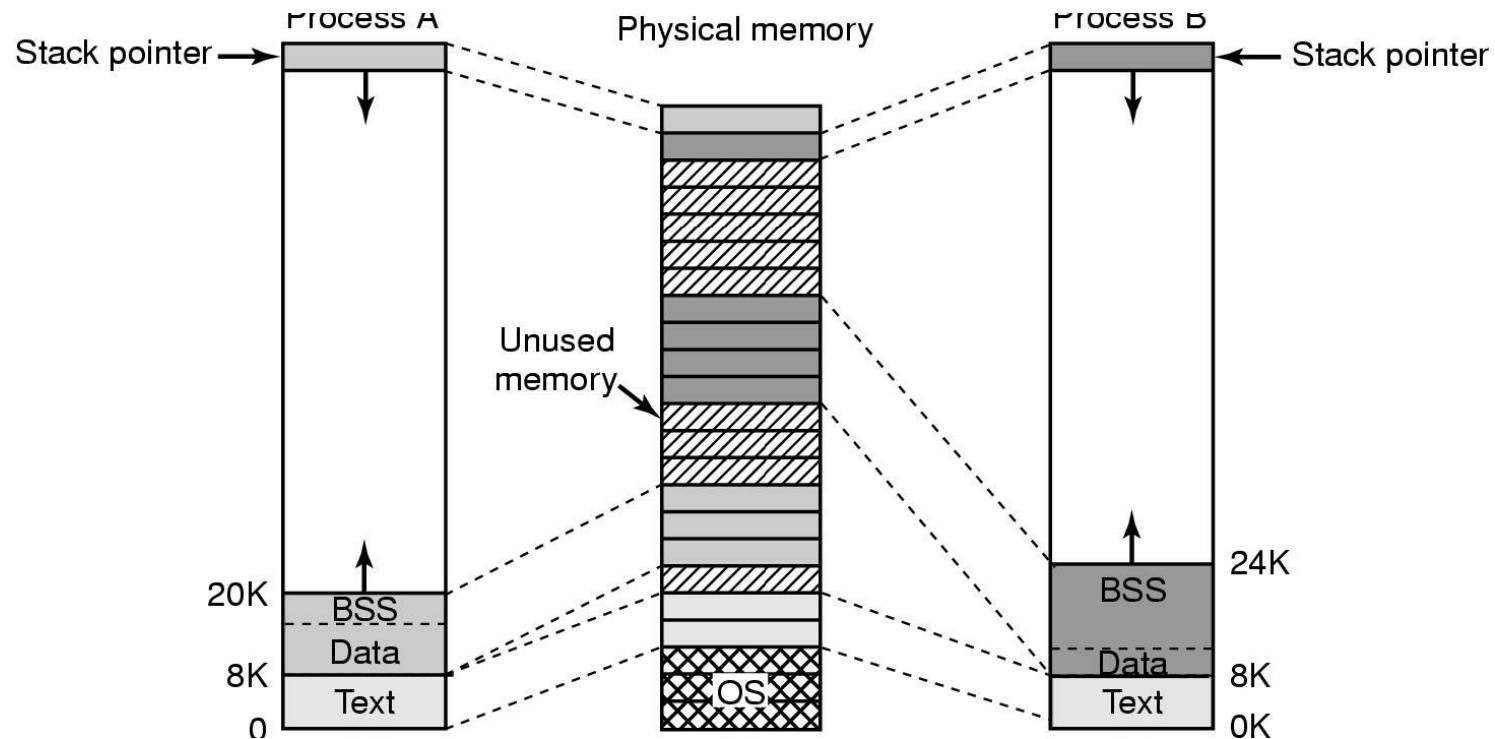
# UNIX'i Önyükleme (boot)

- ..

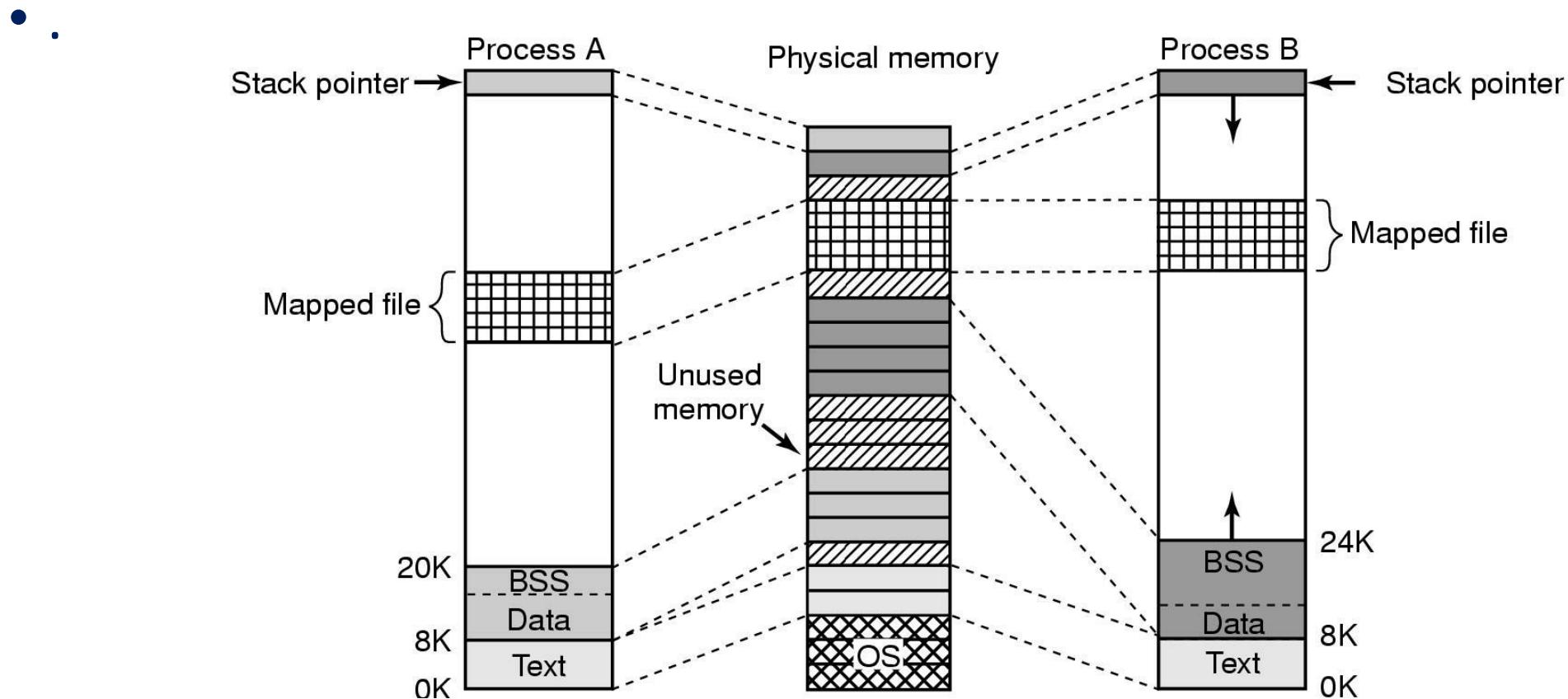


# Bellek Yönetimi

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



# Dosya Paylaşımı



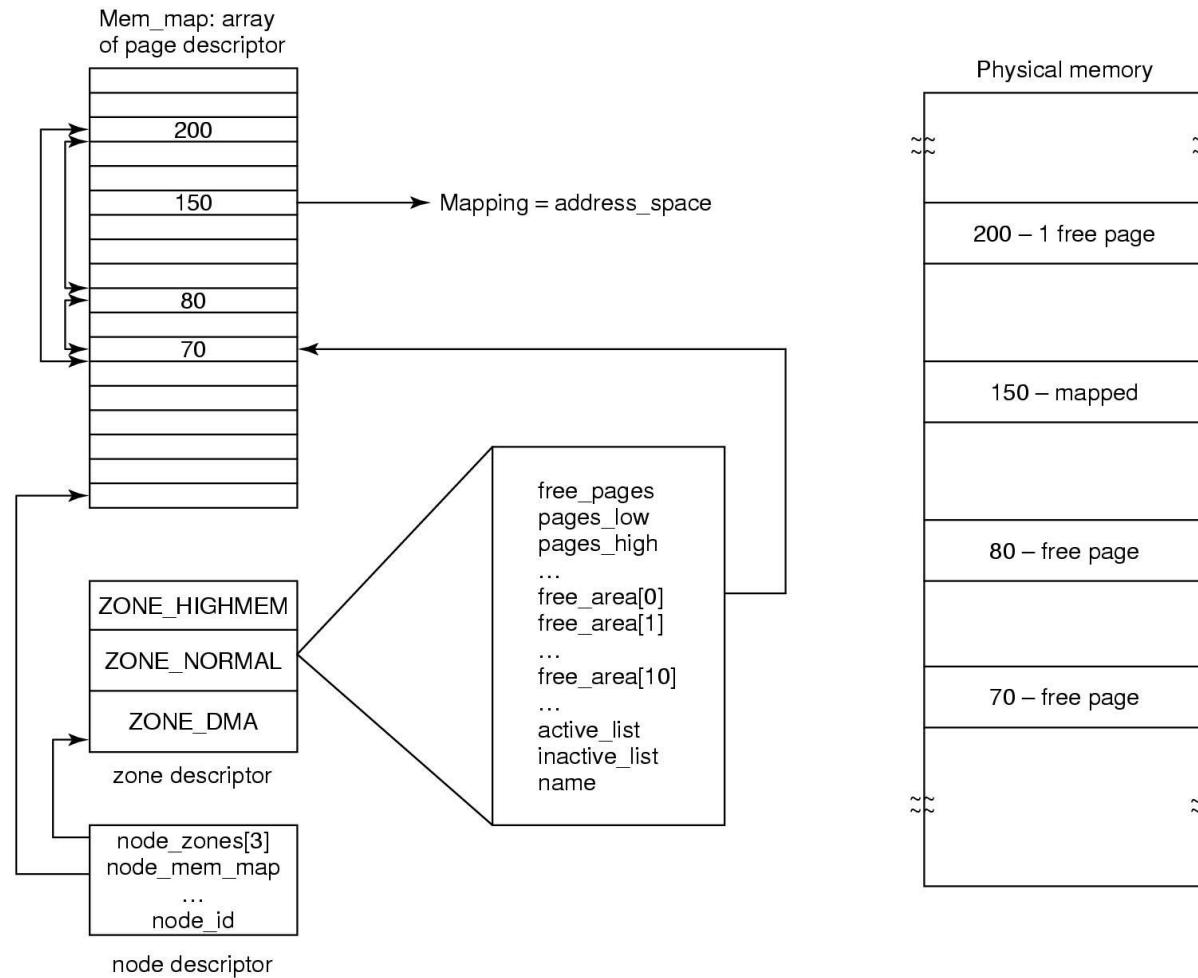
# Bellek Yönetimi Sistem Çağrıları

..

<b>System call</b>	<b>Description</b>
<code>s = brk(addr)</code>	Change data segment size
<code>a = mmap(addr, len, prot, flags, fd, offset)</code>	Map a file in
<code>s = unmap(addr, len)</code>	Unmap a file

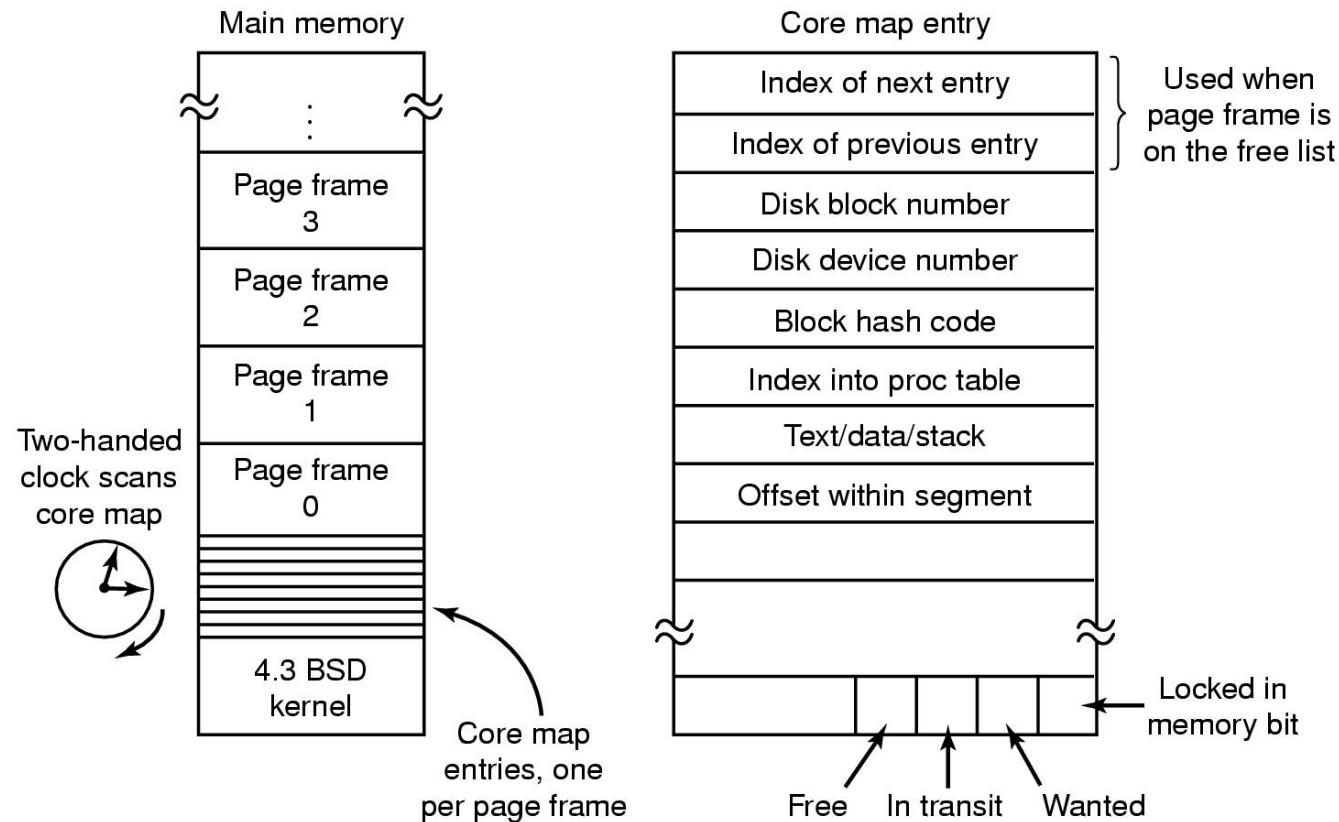
# Linux Ana Bellek Gösterimi

- 
- 



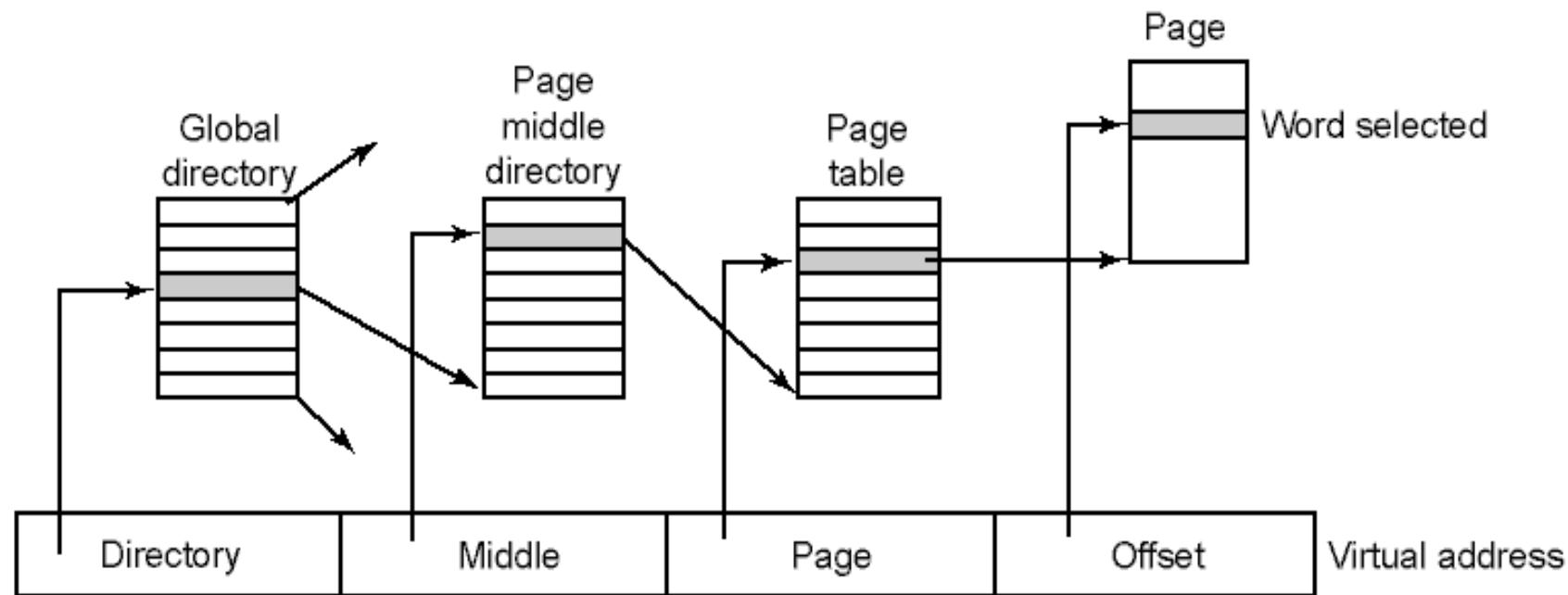
# UNIX Sayfalama

- .
- .



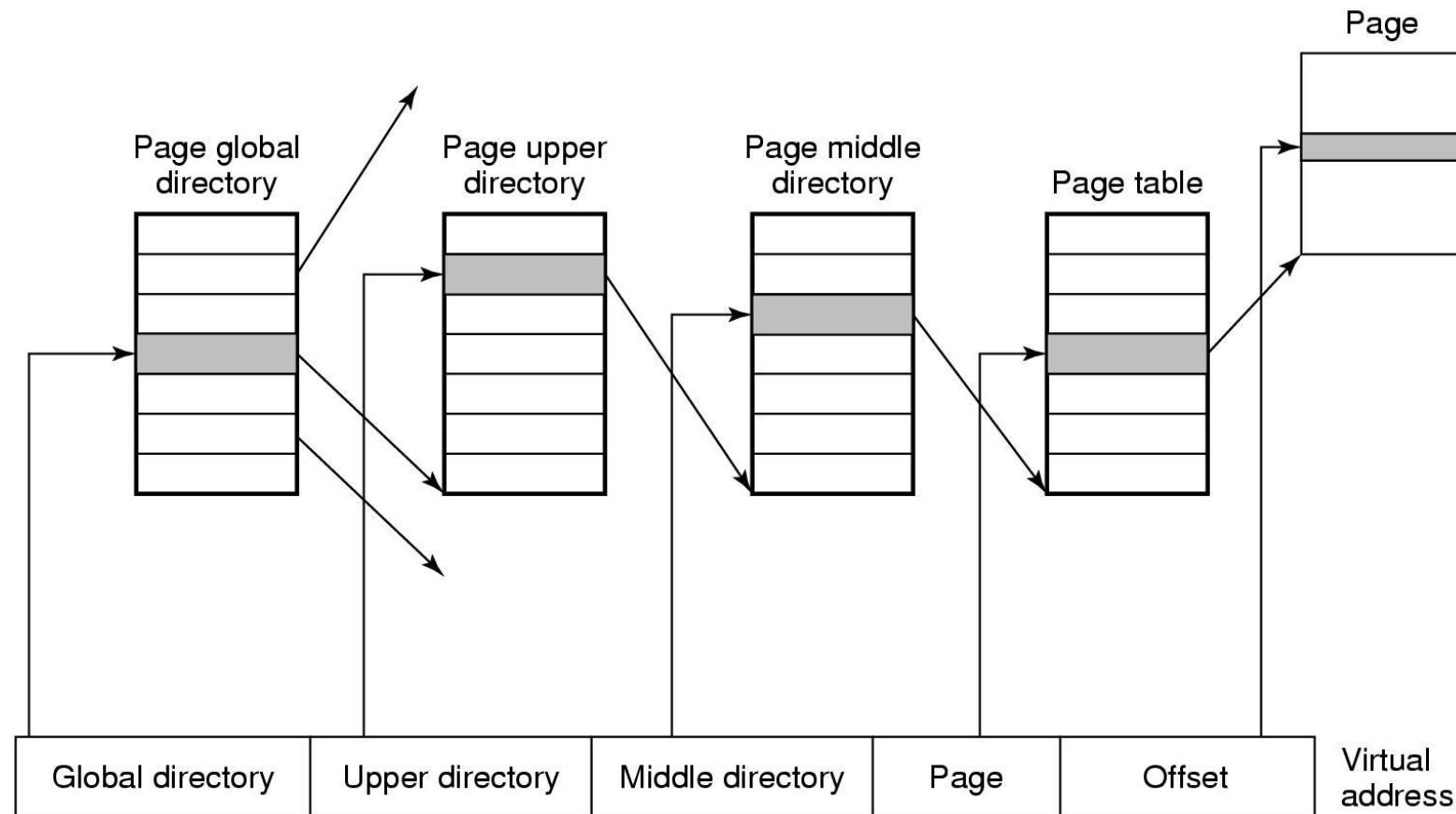
# Üç Düzeyli Sayfa Tablosu

- .
- .



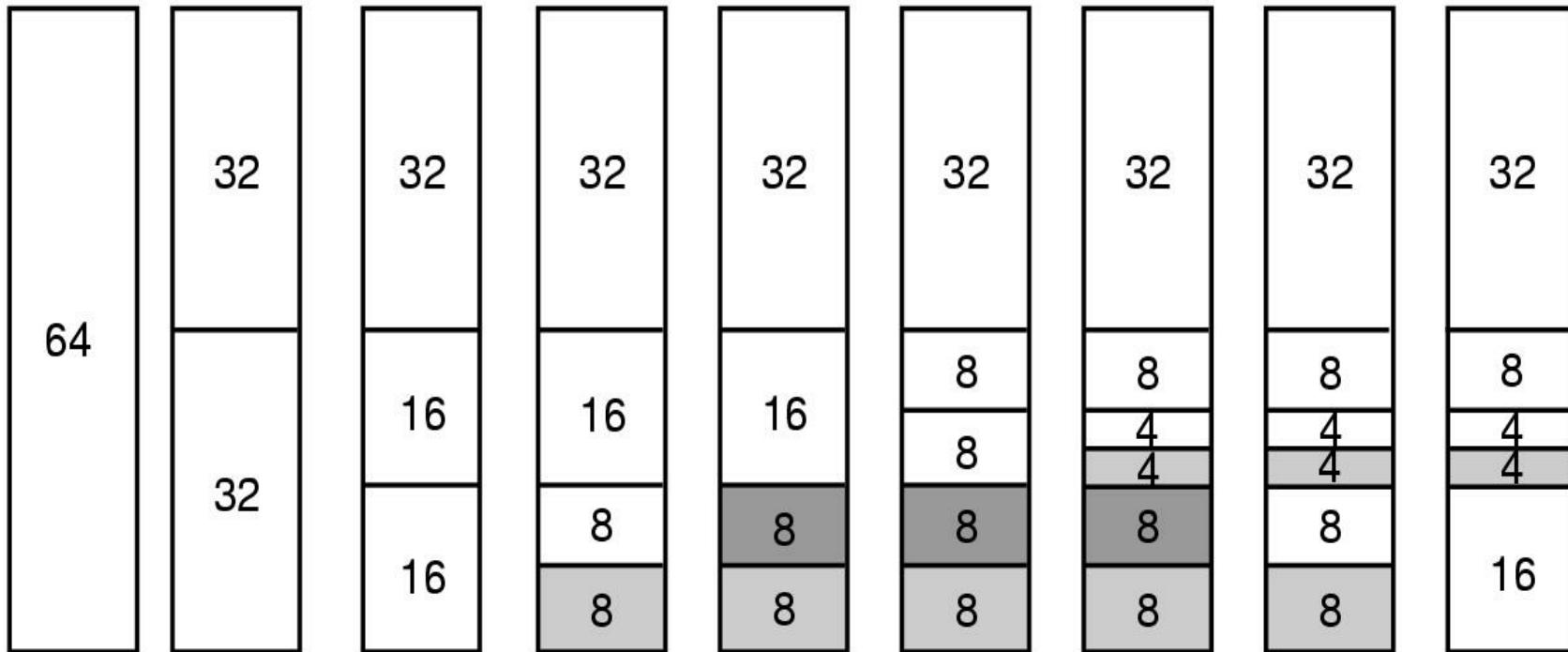
# Dört Düzeyli Sayfa Tablosu

- .
- .



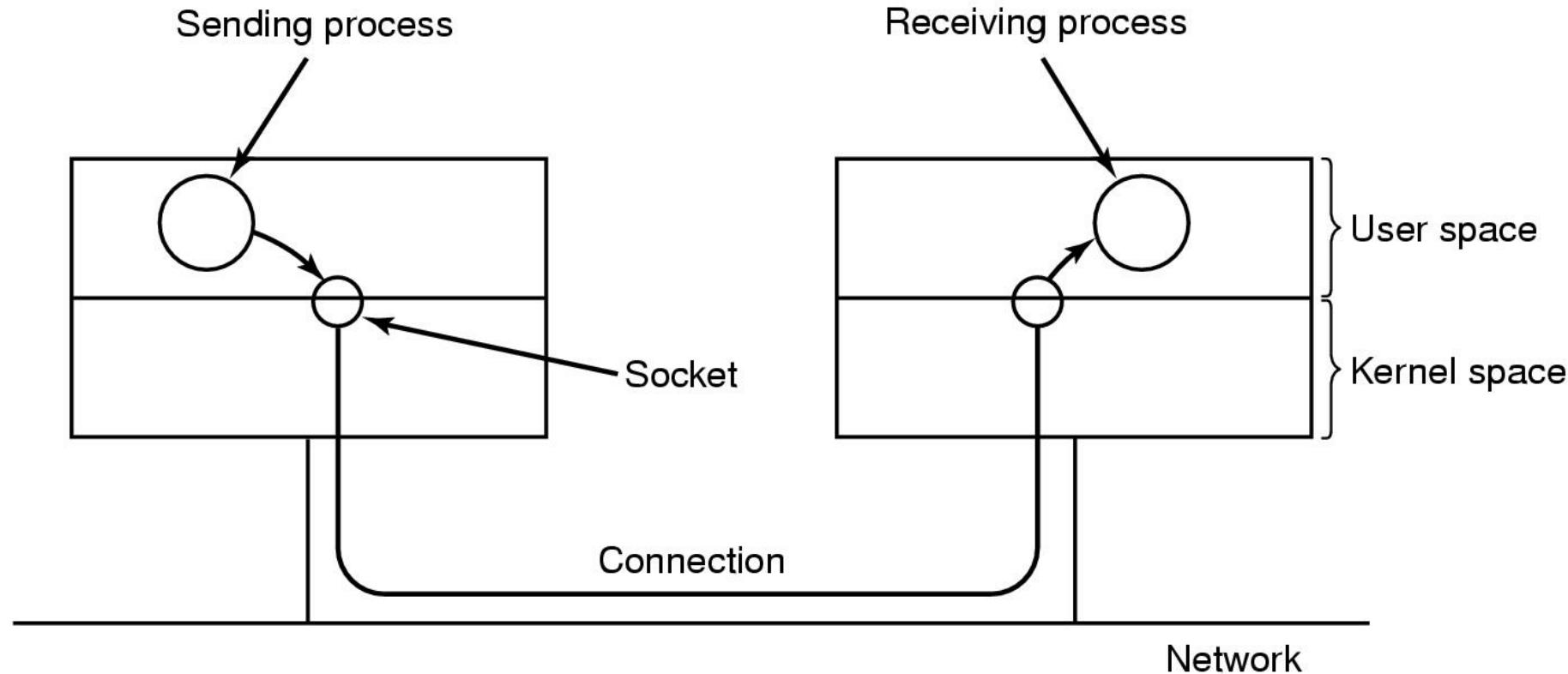
# Buddy Algoritması

- .
- .



# Ağ Haberleşmesi – Socket Kullanımı

- .
- .



# Terminal Yönetimi

..

Function call	Description
<code>s = cfsetospeed(&amp;termios, speed)</code>	Set the output speed
<code>s = cfsetispeed(&amp;termios, speed)</code>	Set the input speed
<code>s = cfgetospeed(&amp;termios, speed)</code>	Get the output speed
<code>s = cfgetispeed(&amp;termios, speed)</code>	Get the input speed
<code>s = tcsetattr(fd, opt, &amp;termios)</code>	Set the attributes
<code>s = tcgetattr(fd, &amp;termios)</code>	Get the attributes

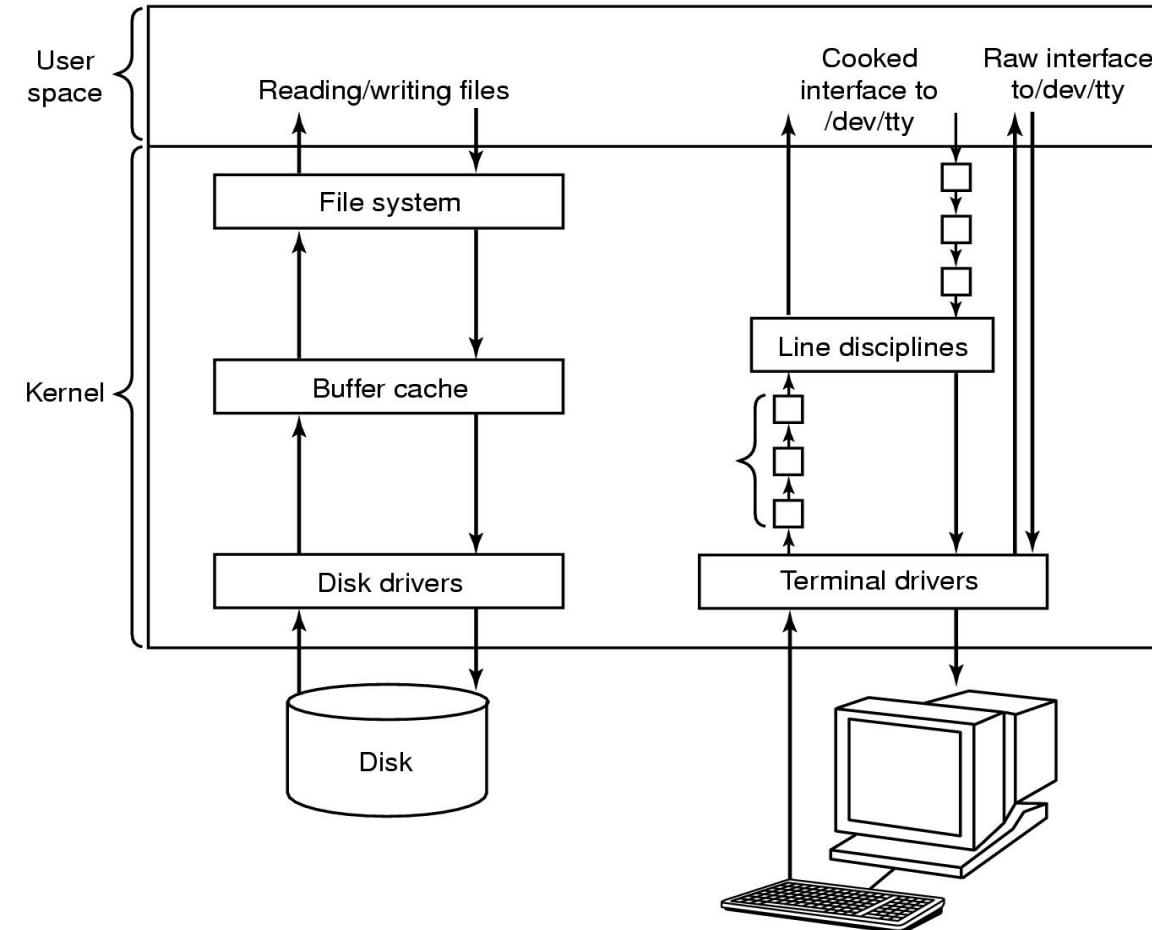
# UNIX G/Ç

• .

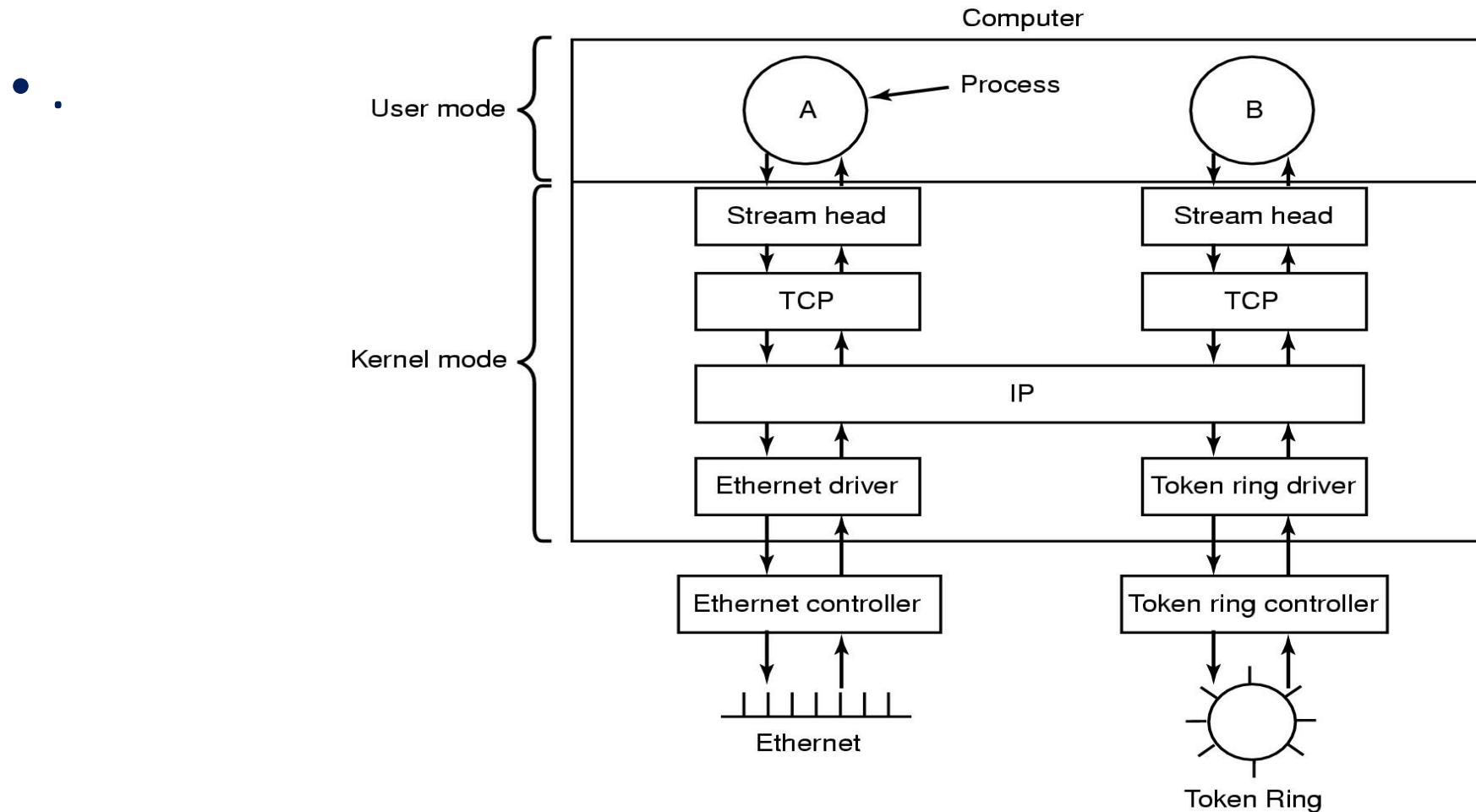
Device	Open	Close	Read	Write	ioctl	Other
Null	null	null	null	null	null	...
Memory	null	null	mem_read	mem_write	null	...
Keyboard	k_open	k_close	k_read	error	k_ioctl	...
Tty	tty_open	tty_close	tty_read	tty_write	tty_ioctl	...
Printer	lp_open	lp_close	error	lp_write	lp_ioctl	...

# UNIX G/Ç Sistemi – BSD

- 
- 

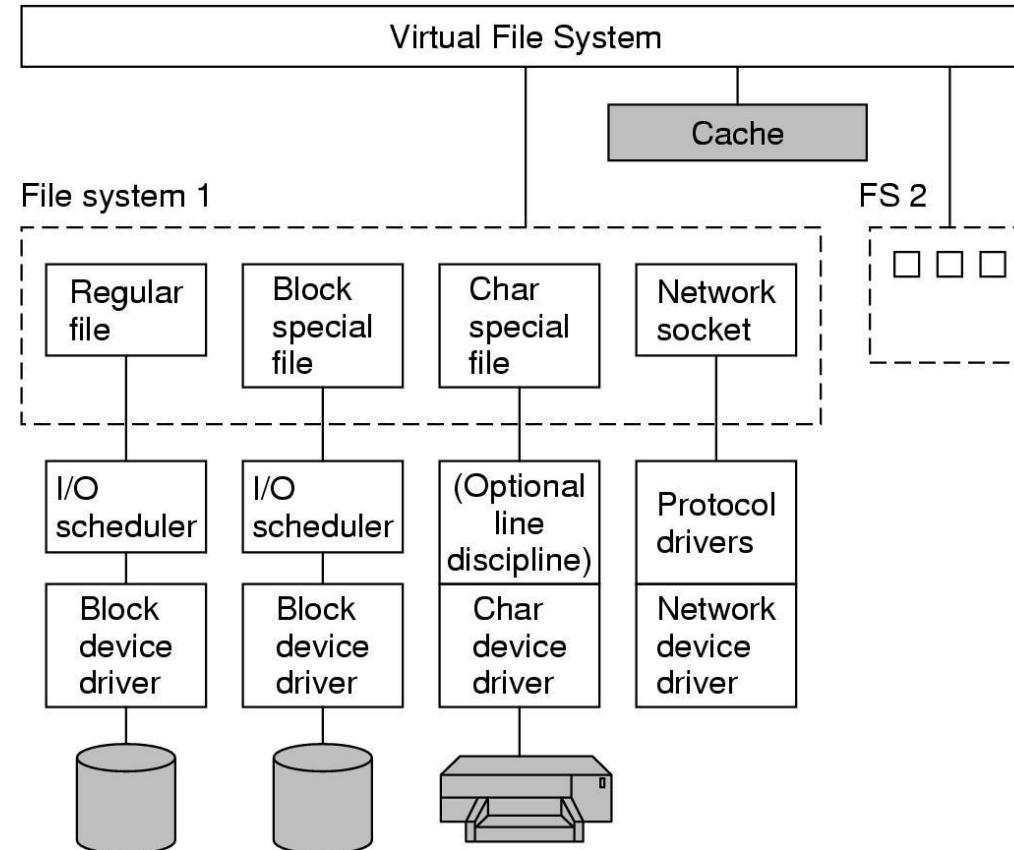


# Akışlar (streams) – System V



# Linux G/Ç Sistemi

- .
- .



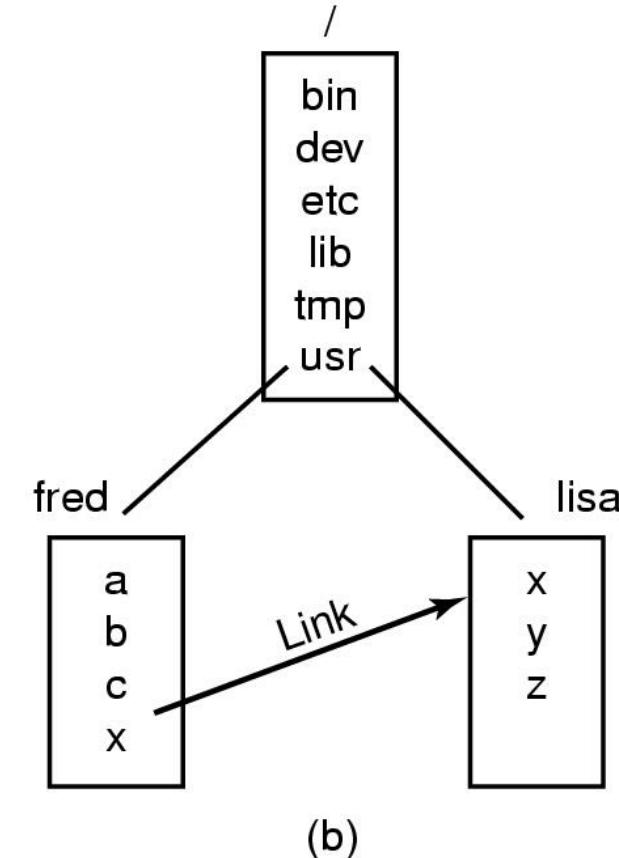
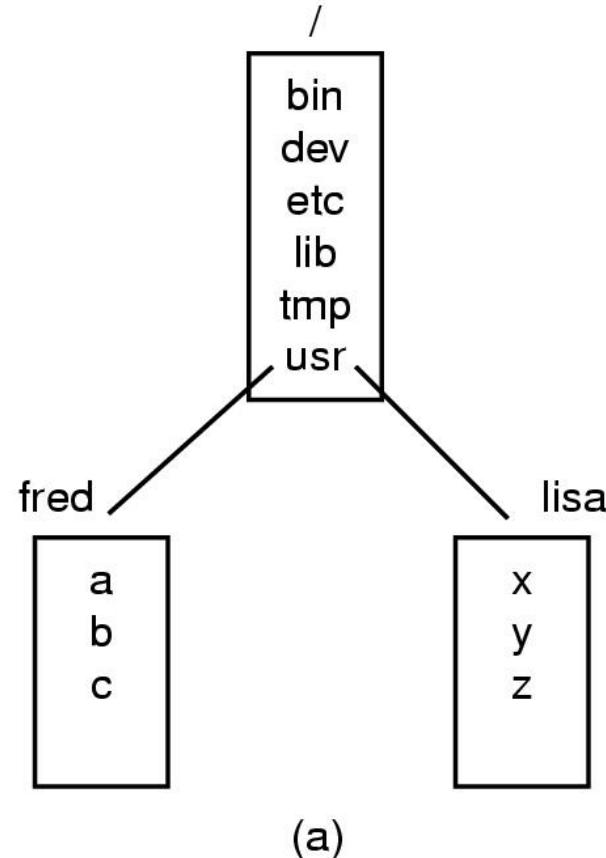
# Önemli Dizinler

..

Directory	Contents
bin	Binary (executable) programs
dev	Special files for I/O devices
etc	Miscellaneous system files
lib	Libraries
usr	User directories

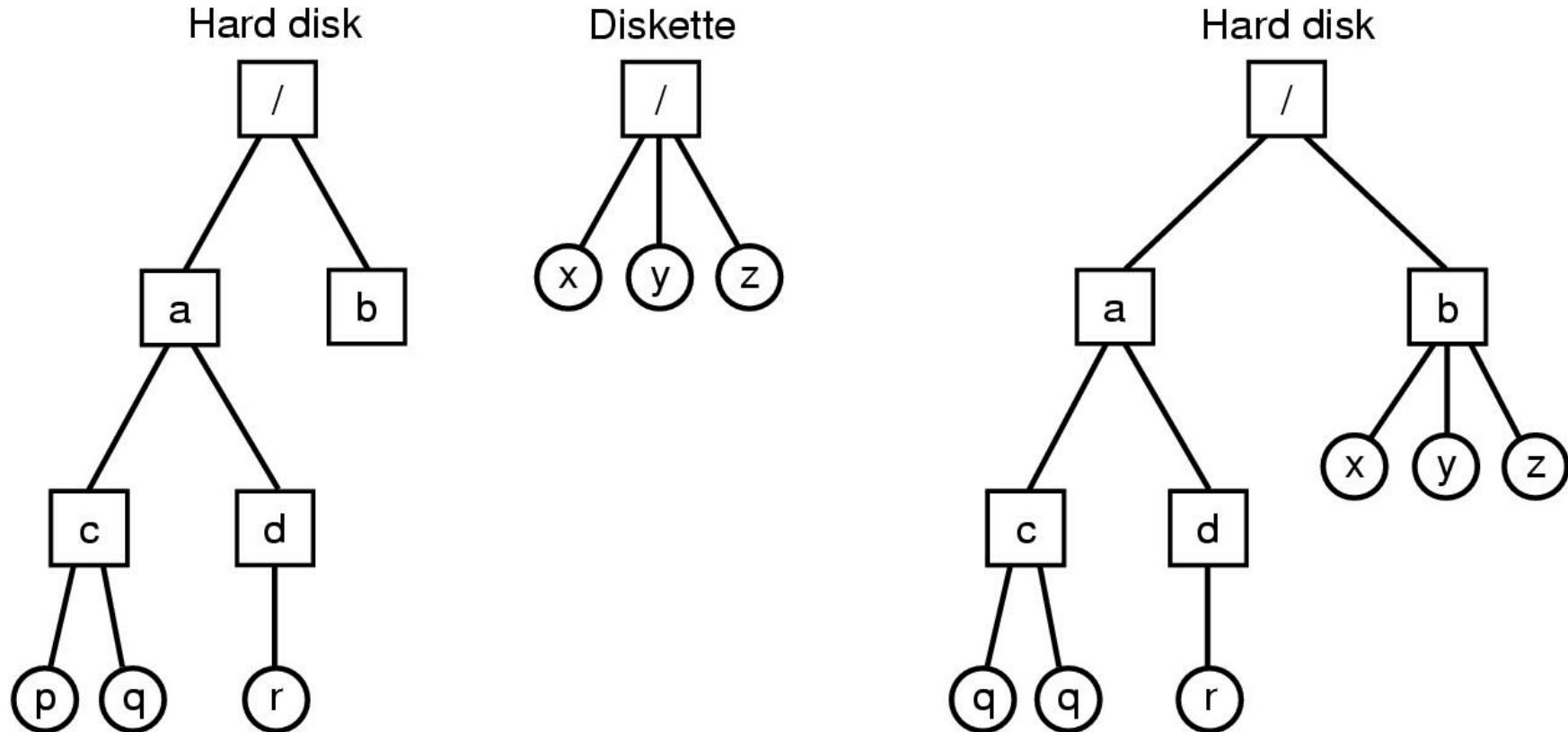
# UNIX Dosya Sistemi

- ..



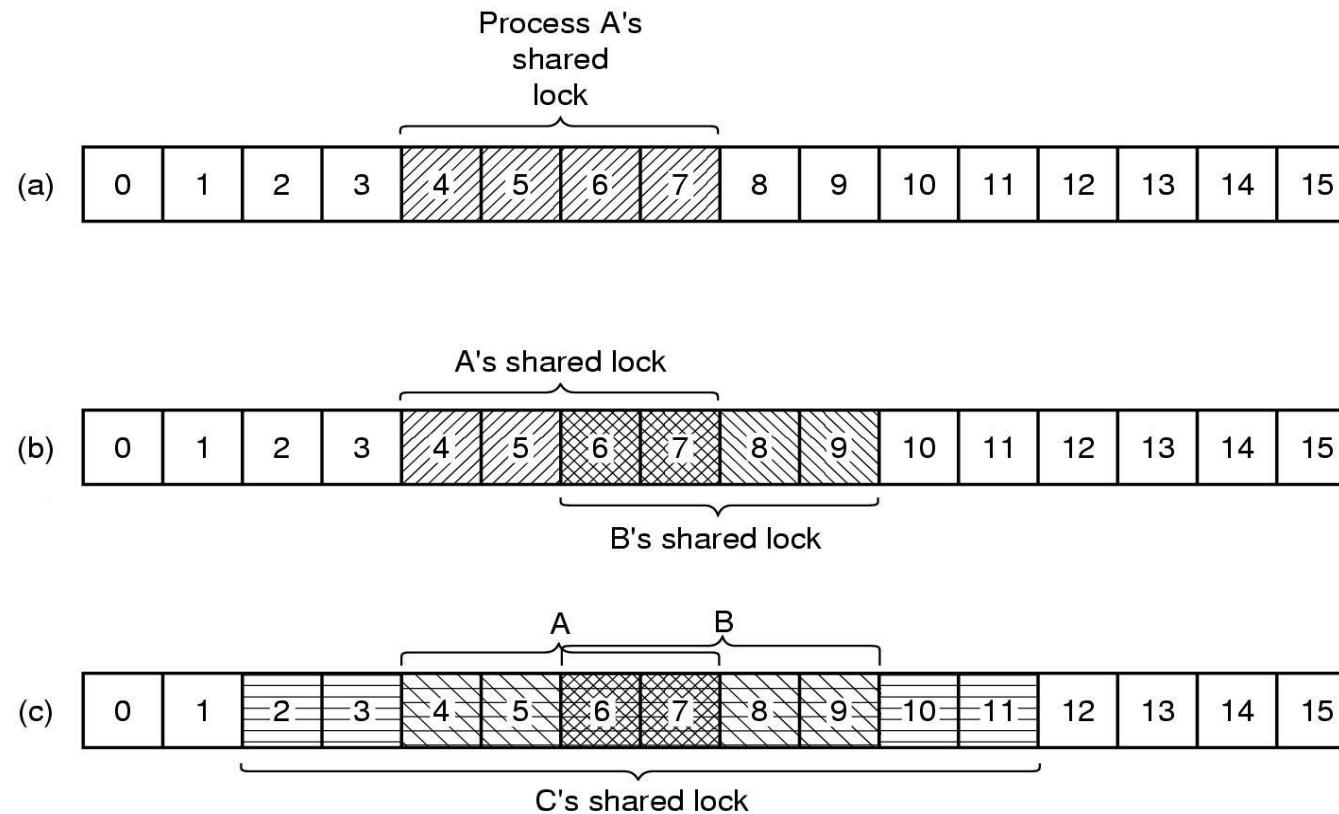
# UNIX Dosya Sistemi

• .



# Dosyaların Kitlenmesi (lock)

•



# Dosya Yönetimi Sistem Çağrıları

..

<b>System call</b>	<b>Description</b>
<code>fd = creat(name, mode)</code>	One way to create a new file
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &amp;buf)</code>	Get a file's status information
<code>s = fstat(fd, &amp;buf)</code>	Get a file's status information
<code>s = pipe(&amp;fd[0])</code>	Create a pipe
<code>s = fcntl(fd, cmd, ...)</code>	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

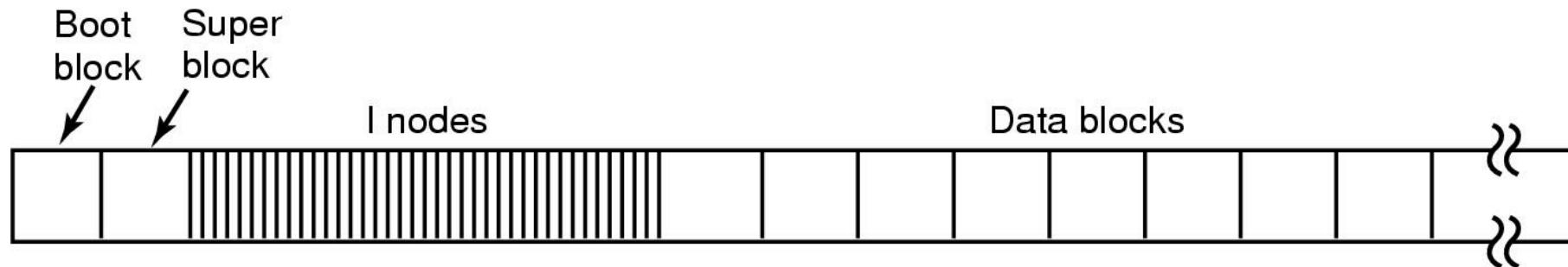
# Dizin Yönetimi Sistem Çağrıları

..

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

# UNIX'te Disk Düzeni

- .
- .

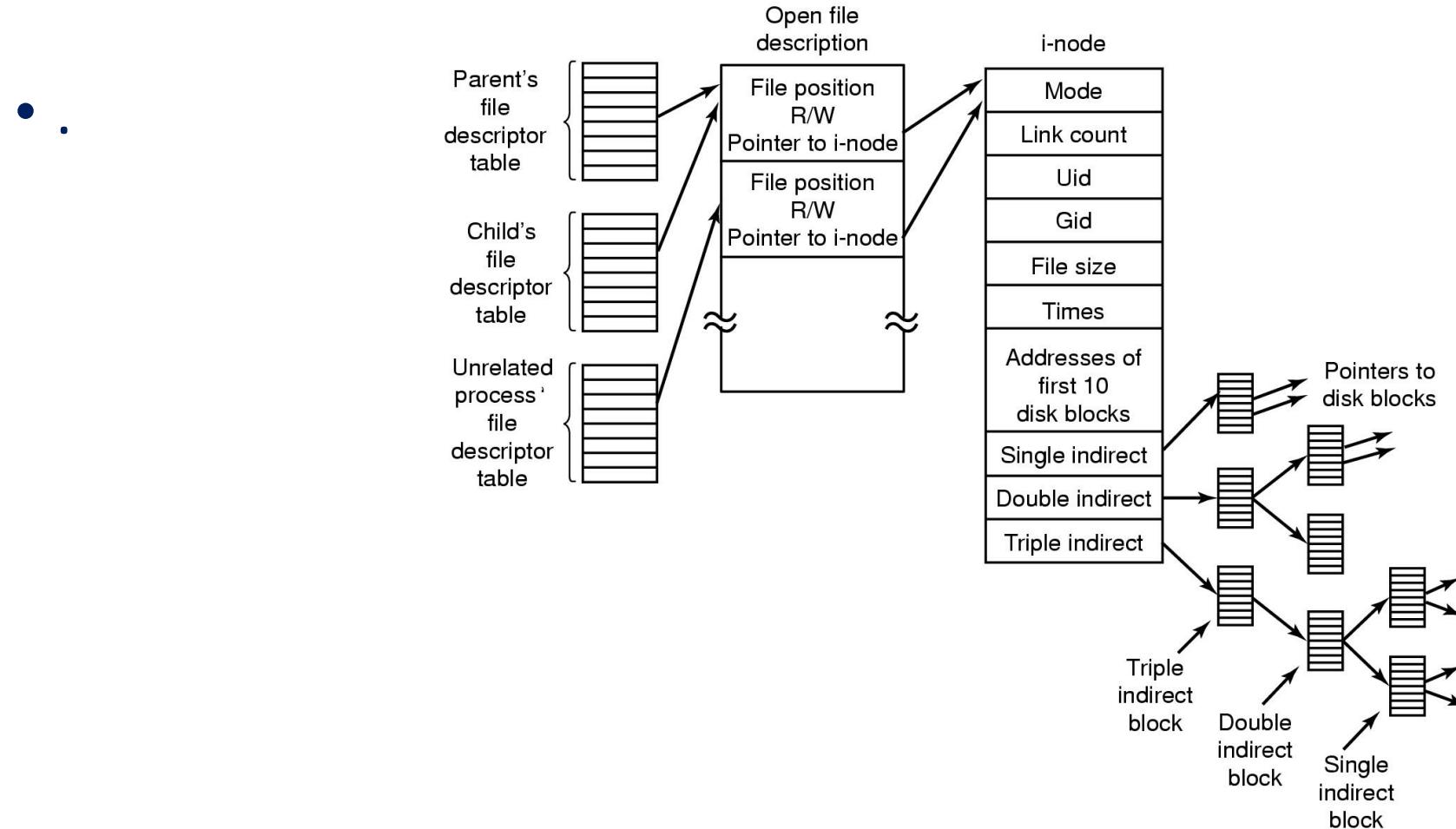


# I-node Yapısı

- .
- .

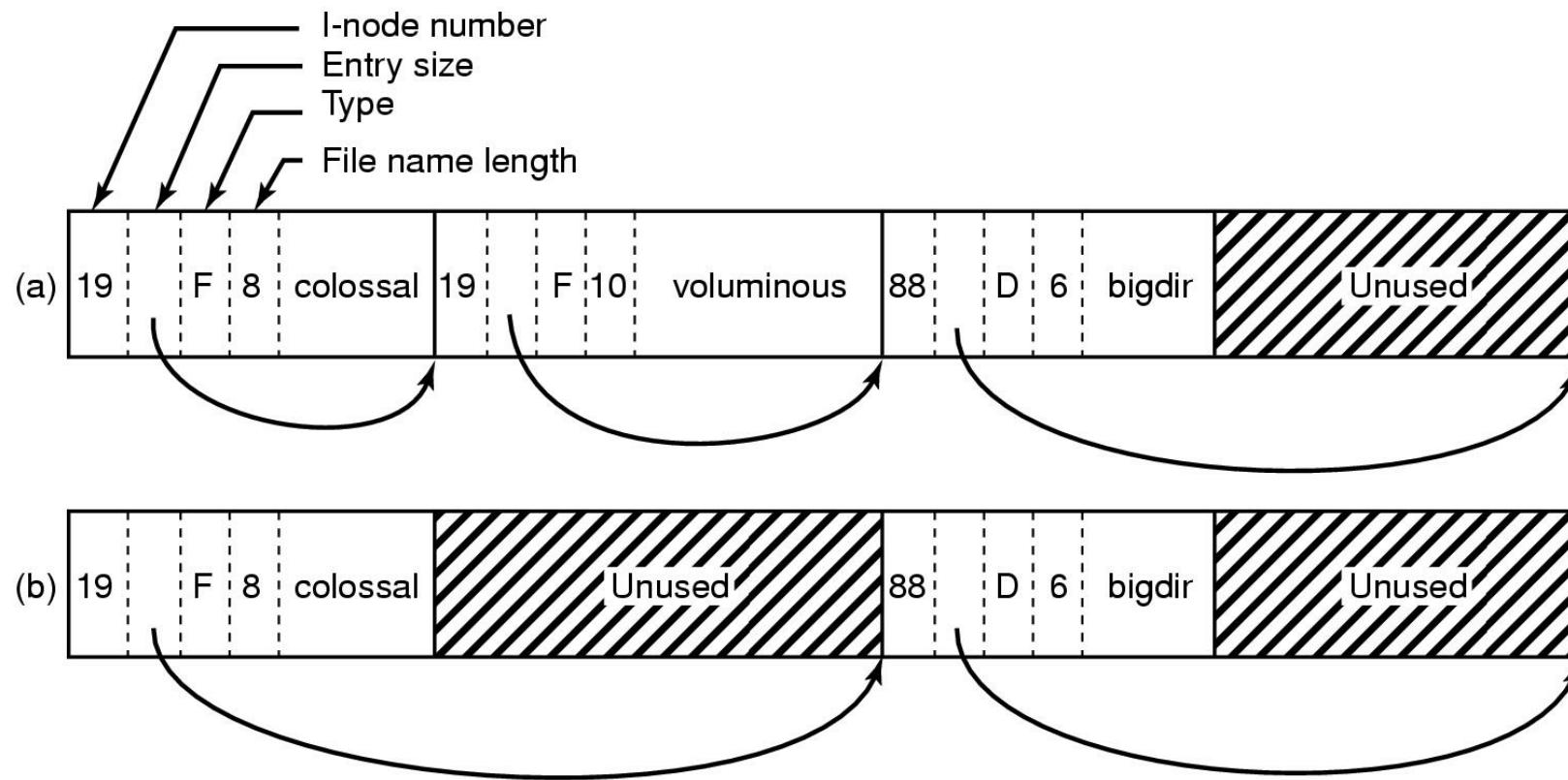
Field	Bytes	Description
Mode	2	File type, protection bits, setuid, setgid bits
Nlinks	2	Number of directory entries pointing to this i-node
Uid	2	UID of the file owner
Gid	2	GID of the file owner
Size	4	File size in bytes
Addr	39	Address of first 10 disk blocks, then 3 indirect blocks
Gen	1	Generation 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 Tablosu, Açık Dosya Açıklaması



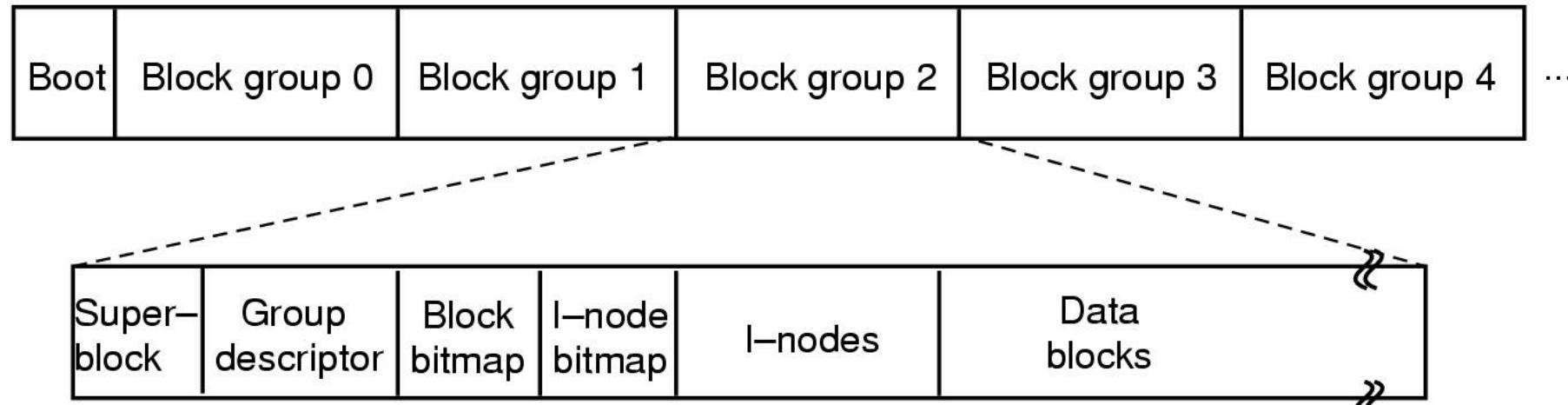
# Üç Dosyalı BSD Dizini

- .
- .

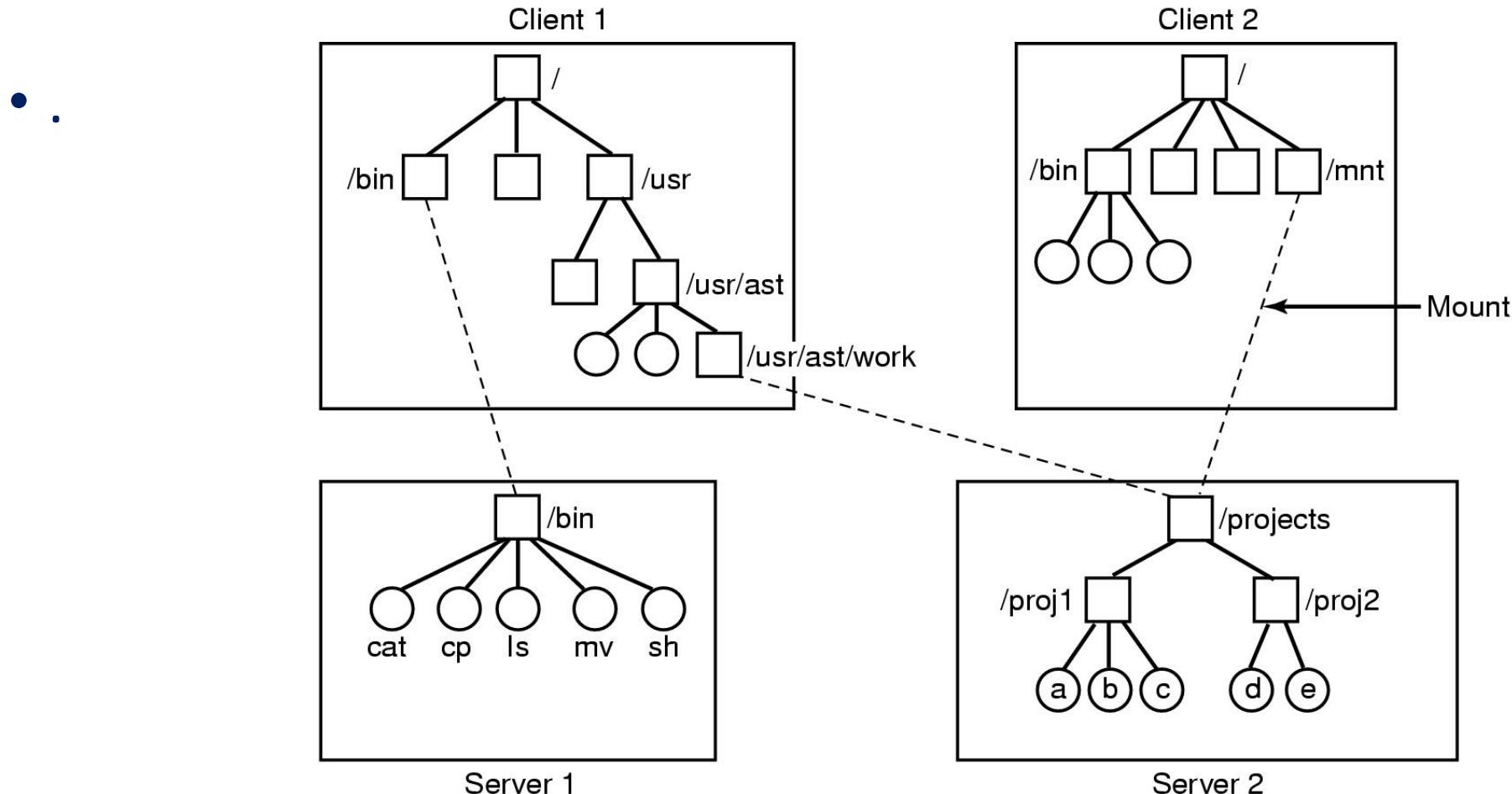


# Linux Ext2 Dosya Sistemi Düzeni

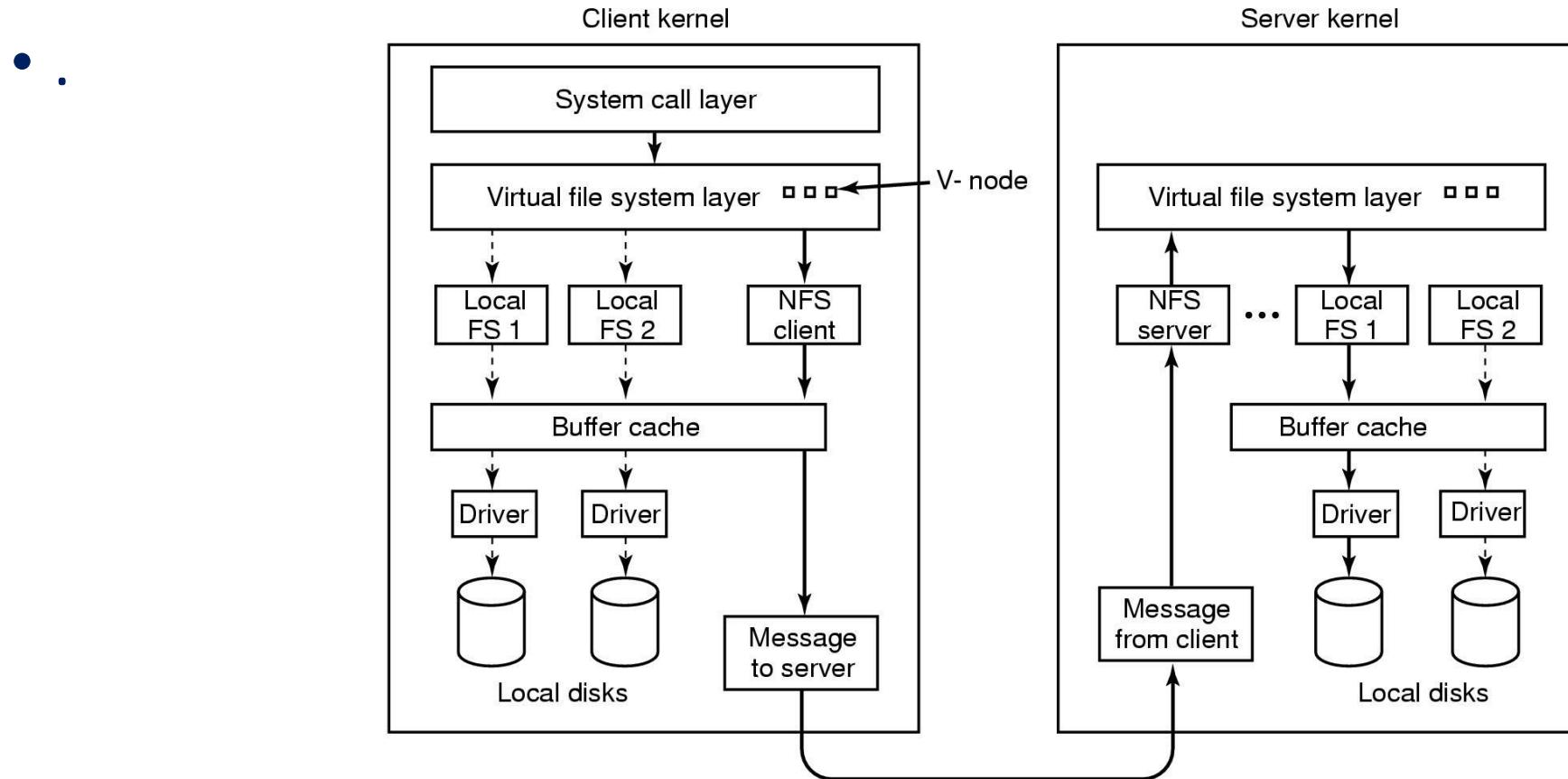
- .
- .



# Ağ Dosya Sistemi (NFS)



# NFS Katman Yapısı



# Dosya Koruma Durumları

..

<b>Binary</b>	<b>Symbolic</b>	<b>Allowed file accesses</b>
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 Koruması Sistem Çağrıları

..

<b>System call</b>	<b>Description</b>
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