

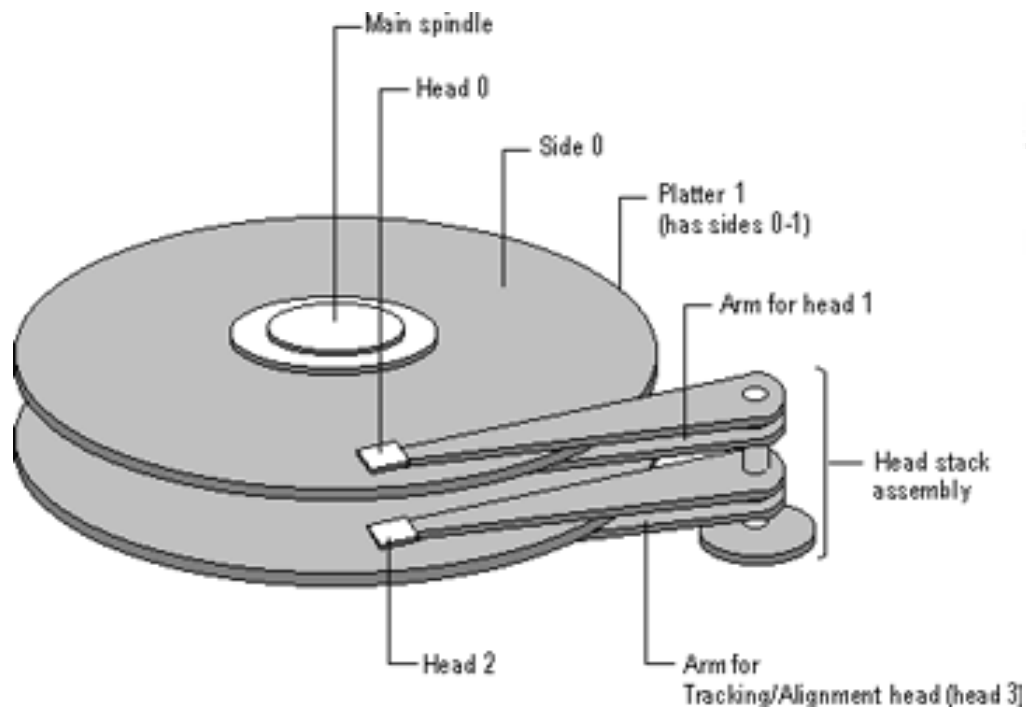
4. Gaia: Fitxategi-sistema eta S/I

1. Sarrera
2. Izen-eremua, katalogoak
3. Sistema erabiltzaileanitzak. Babesa.
Fitxategiak konpartitzea. Erabiltzaileen
kudeaketa
4. Dispositiboekiko independentzia. *Buffering*
5. S/I-ko sistema-deiak

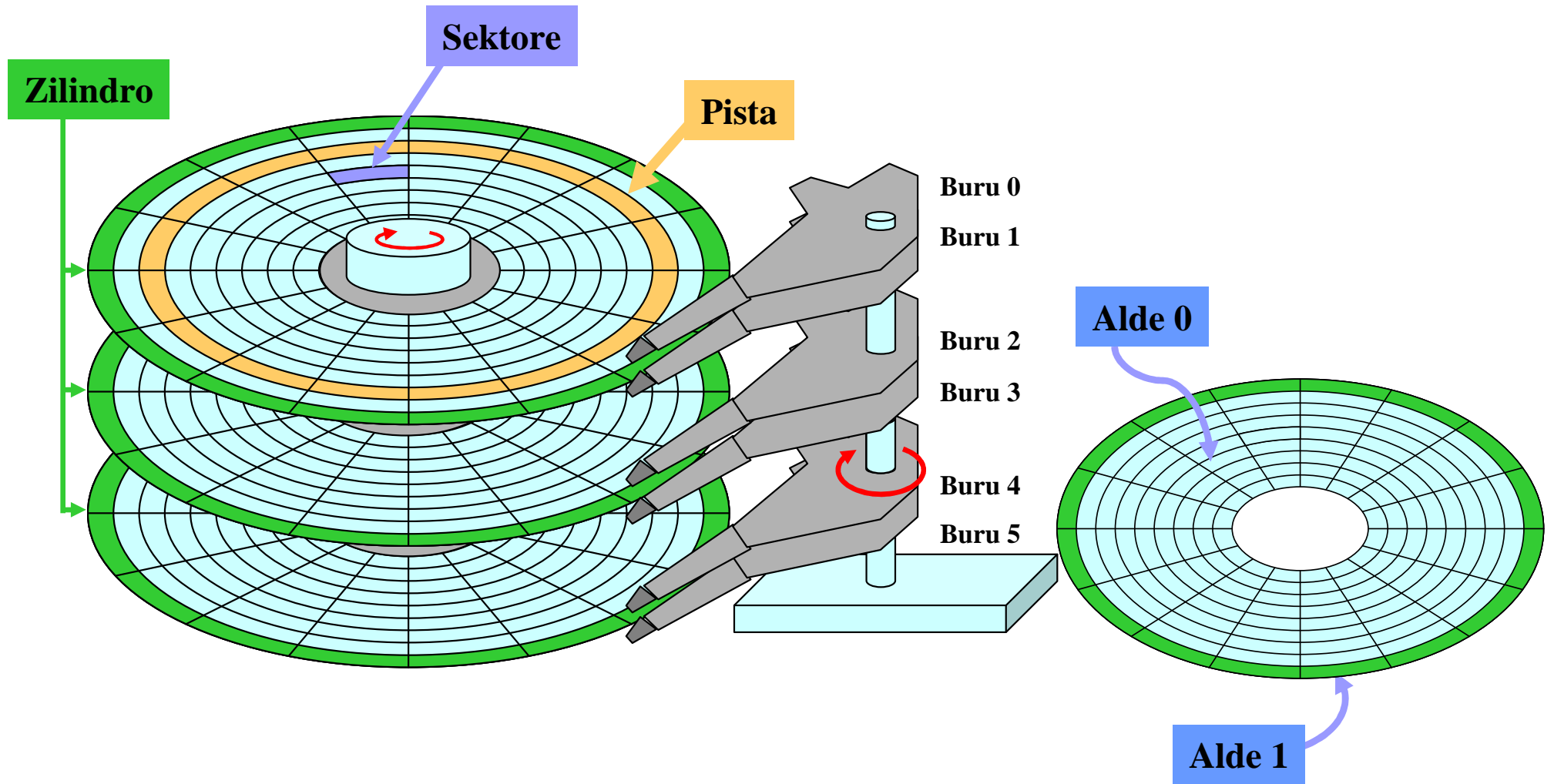
Zergatia

- Fitxategi-sistema = diskoaren abstrakzioa
 - Informazioa gorde: kantitatea, denbora luzez, atzipen konkurrentea
 - Dispositiboak, izaera desberdinekoak: zinta, IDE/SCSI, NFS
- Erabiltzaileen ikuspuntua:
 - Ez axola berezitasunak
 - Garrantzia: interfazea (“nola atzitu daiteke”)
- SEaren diseinatzailearen ikuspuntua:
 - Implementazioa (eraginkortasuna)

Diskoak



Diskoak



Fitxategi-sistemak. Kontzeptuak

- *Fitxategia* – datuak gordetzeko
- *Katalogoak* – fitxategiak antolatzeko
- *Partizioak* – katalogoen multzoa
(ere ezagunak “bolumen” bezala)
 - Katalogoen informazio guztia partizioan
 - mount: fitxategi-sistemarekin lan egiteko
- *Segurtasuna* – bideratu/mugatu atzipena fitxategi, katalogo eta partizioei

Fitxategi sistema bat

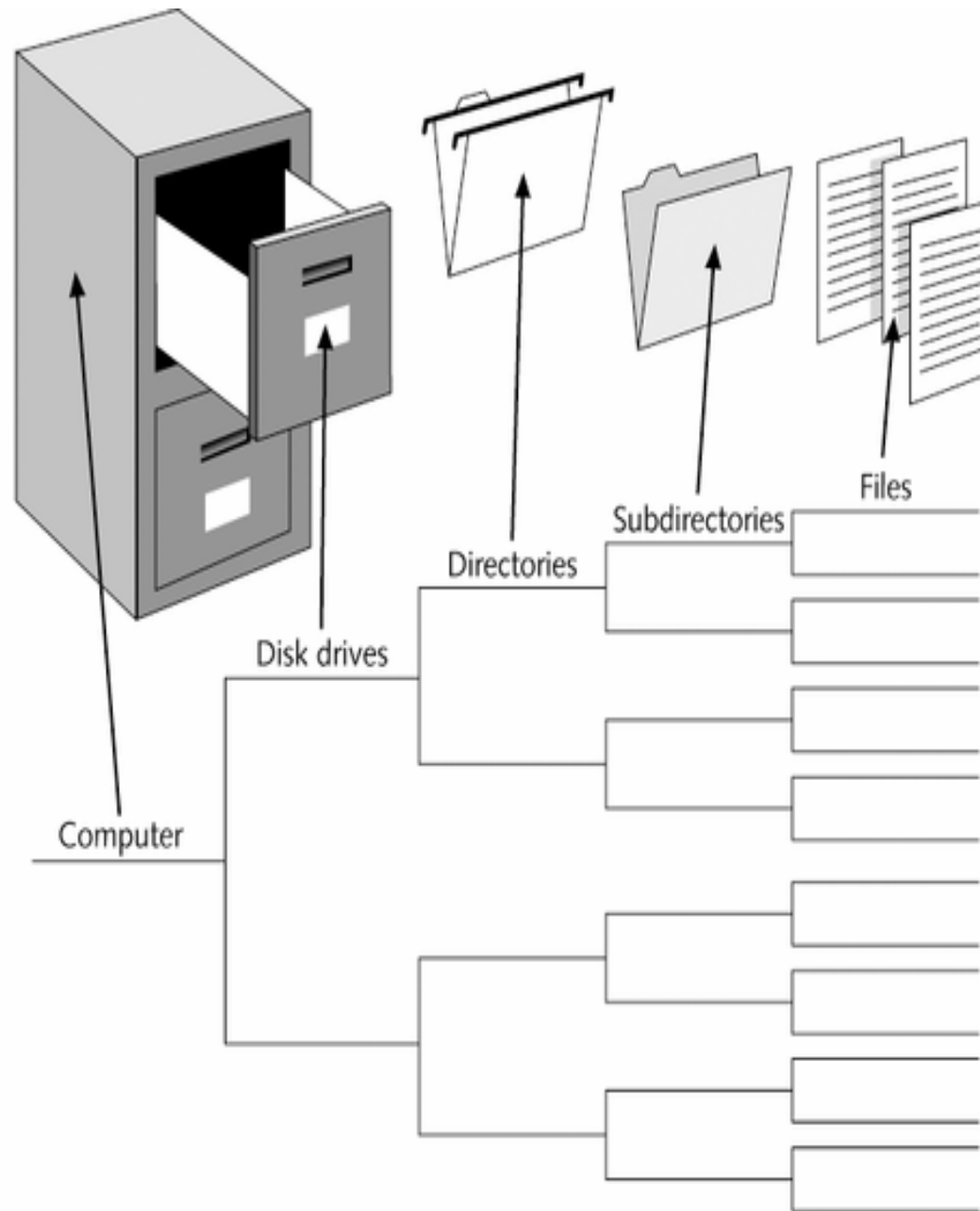
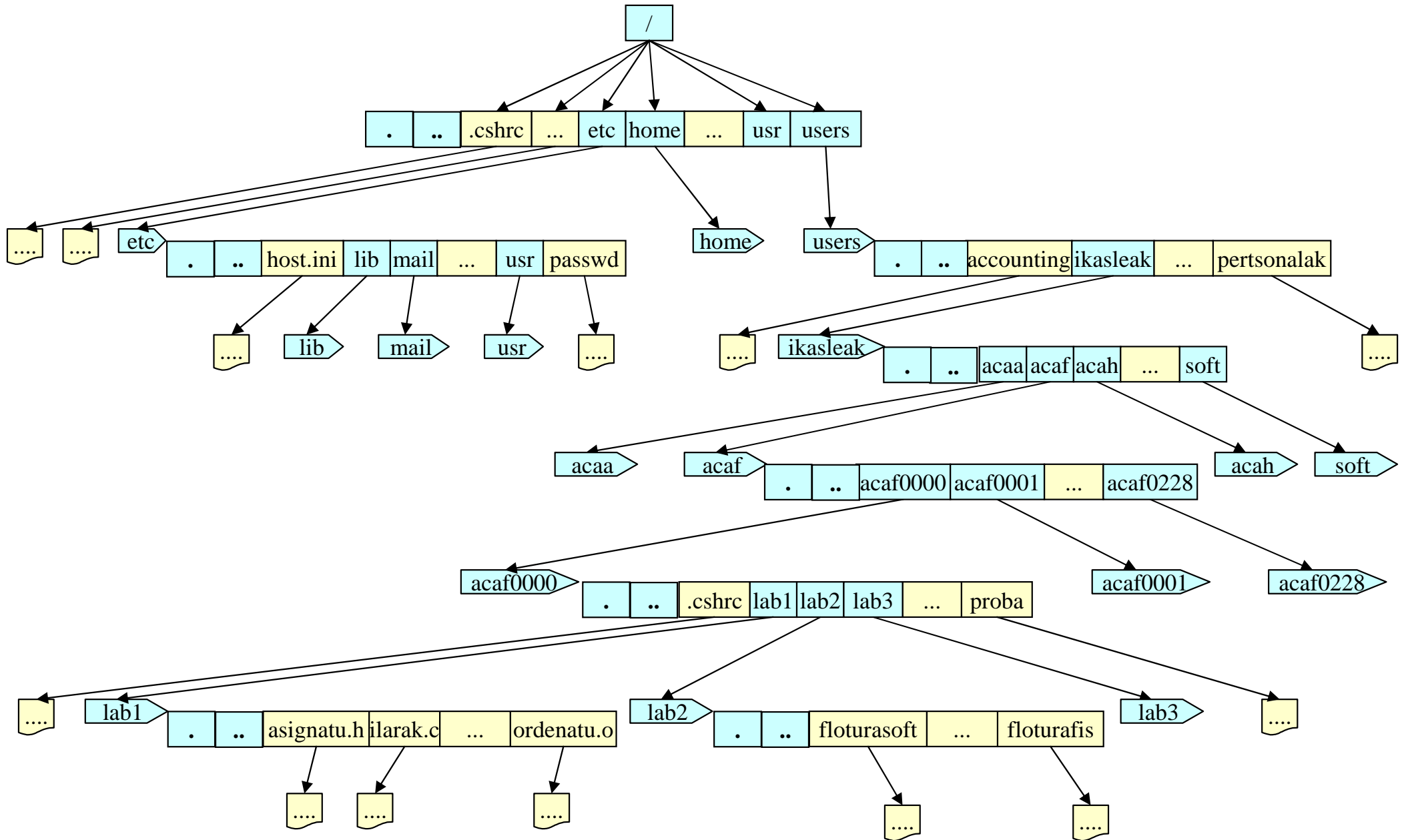


Figure 3-1 A file system

Fitxategi sistemaren ikuspegia



Katalogoen egituraren antolaketa

- Katalogoen antolaketa egiterakoan ondokoak kontsideratu behar dira:
 - Sistema Eragilearen fitxategiak
 - Software aplikazioak
 - Lan fitxategiak, hots, dokumentuak, grafikoak, kalkulu orriak eta datu-baseak
 - Sarean zehar konpartitu beharreko fitxategiak
 - Utilitateen fitxategiak
 - Behin-behineko fitxategiak

Katalogoen antolaketa - Unix

- UNIX sistemetan, katalogoen antolaketa arrunta ondokoa da:
 - *bin* erabiltzaileen programa eta utilitateetarako (*binary*)
 - *lib* liburategietarako
 - *usr* erabiltzaileen fitxategi eta programentzako
 - *var* edukin aldakorreko edo behin-behineko fitxategientzat
 - *tmp* behin-behineko fitxategientzat
 - *dev* dispositiboen fitxategientzat
 - *mnt* diskete, CD-ROM unitate, eta oro har muntatu daitekeen edozein beste dispositiboentzako
 - *etc* sistemaren eta konfigurazio fitxategientzako
 - *sbin* erabiltzaileen programa eta utilitateetarako (*system binary files*)
 - *home* erabiltzaileen kontuetarako
 - *proc* sistemako baliabideen kudeaketarako

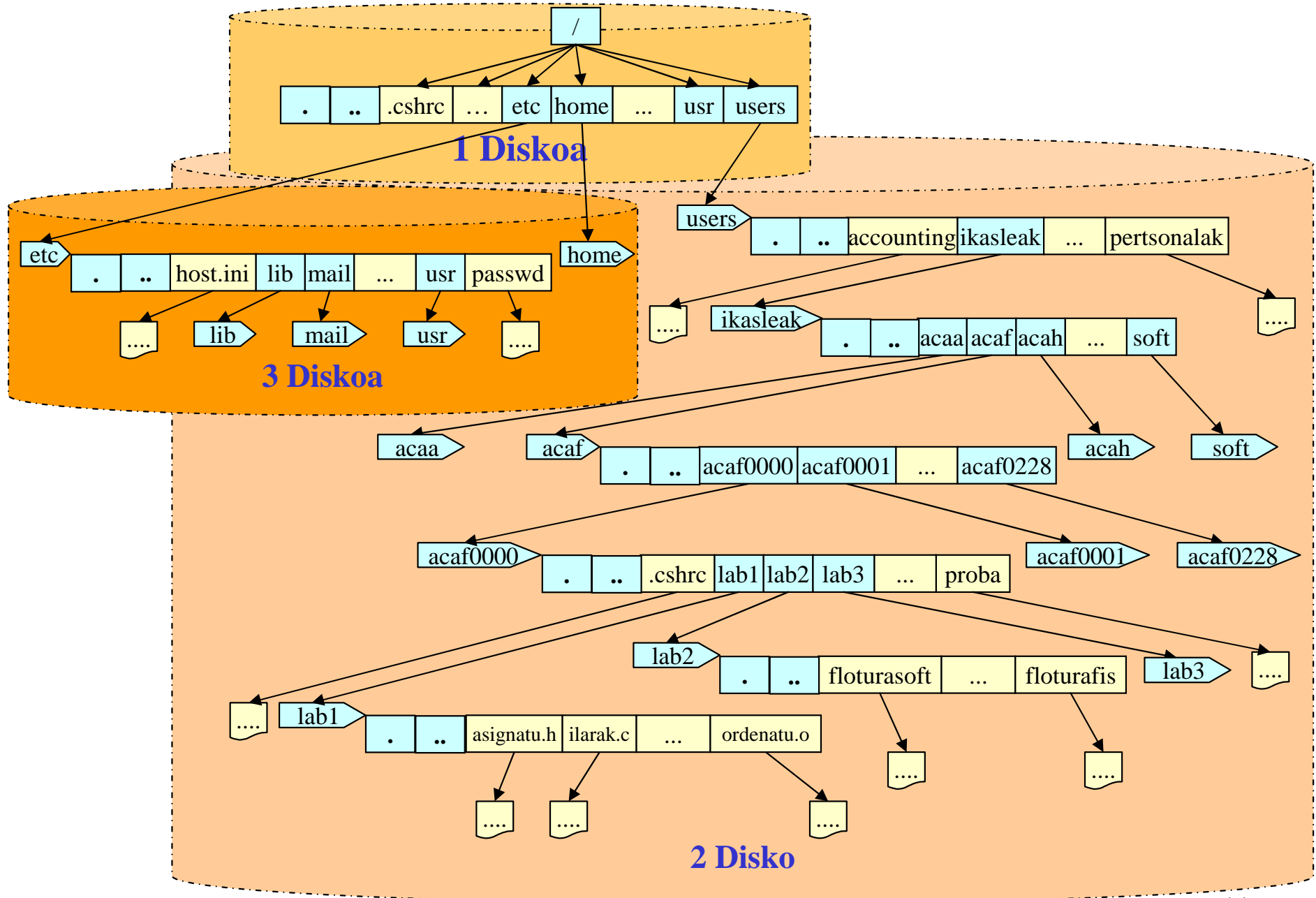
Katalogoen antolaketa - NT

DOS eta Windows9X-ekin bateragarritasuna mantendu behar du

- C:\temp behin-behineko fitxategiak
- C:\winnt NT sistemaren fitxategiak, boot eta erabiltzaileen konfigurazio fitxategiak
- C:\winnt\system NT sistemaren fitxategiak, driverren fitxategiak, estekatze dinamikorako liburutegiak, konfigurazio fitxategiak
- C:\winnt\system32 sistemaren fitxategiak, liburutegi konpartituak, programen fitxategiak, konfigurazio fitxategiak
- C:\winnt\system32\drivers driver gehiago
- C:\winnt\system32\drivers\etc konfigurazio fitxategi gehiago

Sistema errobakarra

Dispositiboak (diskoak/partizioak, disketeak, ...) *muntatuak* zuhaitz bakarraren puntu batean (UNIX)



Katalogoen Izenak

- / – erroa Unix-en
- ./ – uneko katalogoa
- ../ – guraso-katalogoa
- ~ – erabiltzailearen erro-katalogoa
(adibidez: ~acaf0251/)

Fitxategi baterako erreferentzia (*izena*)

[*Bidea*] *Izena*

❑ ***Bidea (PATH)***: fitxategia zuhaitzean kokatzen duen ibilbidea.

➤ ***Absolutua***: ibilbidea zuhaitzaren errotik hasita (/).

➤ ***Erlatiboa***: ibilbidea uneko katalogotik hasita.

(deia egiten duen prozesua aurkitzen den katalogoa)

Bidea hutsik uzten bada fitxategia uneko katalogoan dagoela suposatzen da.

❑ ***Izena***: fitxategiari dagokion katalogoko izena.

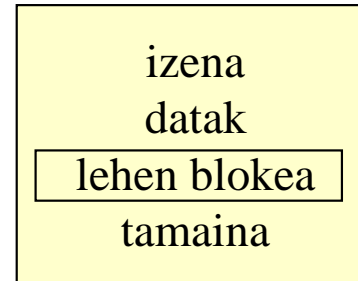
Adibideak:

```
lab1/adibidea
./lab1/adibidea
../acaf0003/lab1/adibidea
/users/alumnos/acaf/acaf0003/lab1/adibidea
```

Fitxategi baten informazioa

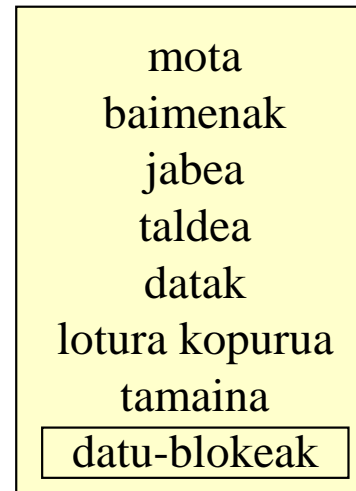
- Kontrol informazioa
 - **Katalogoan** (FAT32)

*Katalogoko
sarrera*



- **Fitxategi/inode**-an bertan
(UNIX, NTFS, ...)

Inode



- Edukina edo datuak

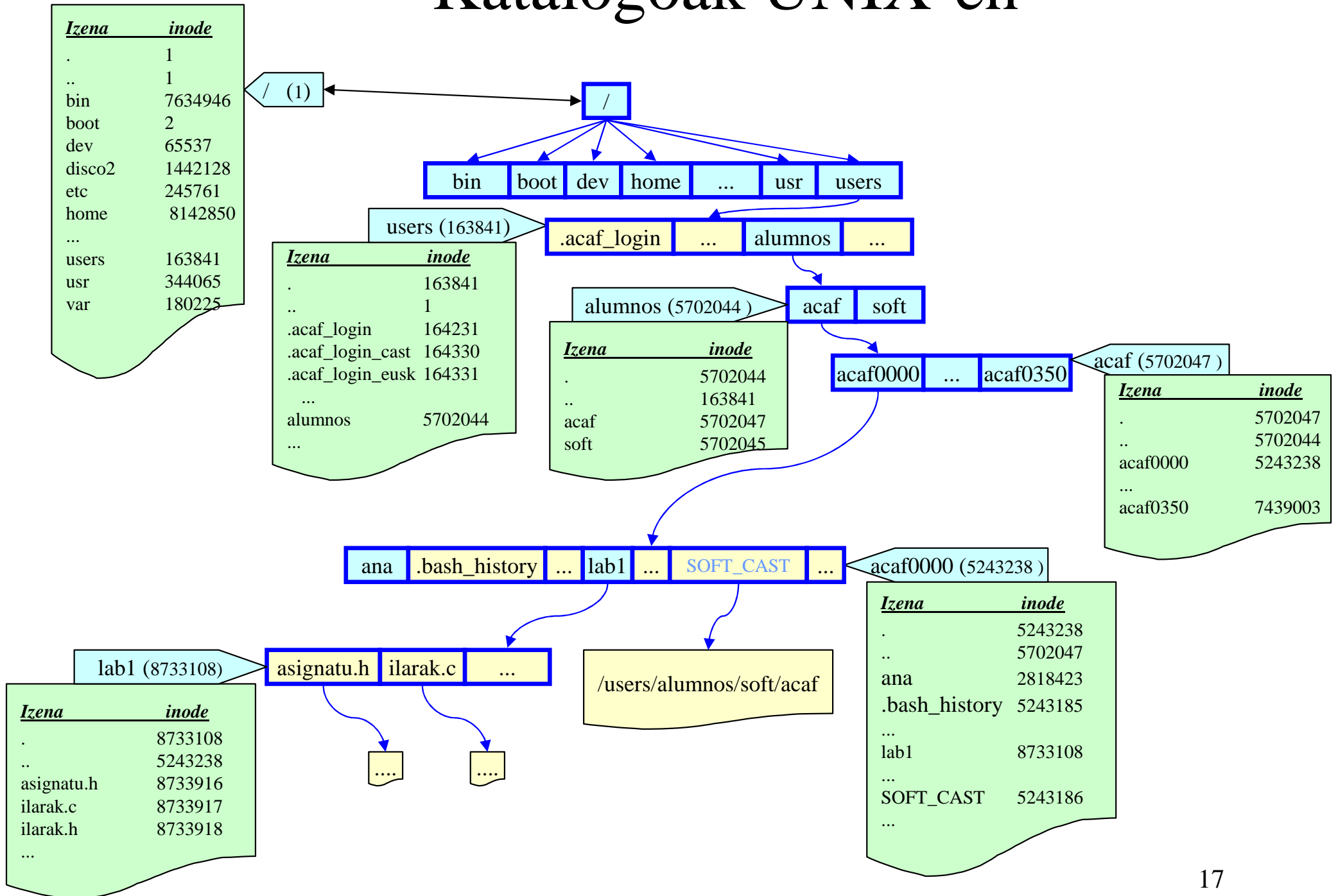
Katalogoak UNIX-en

- Zuhaitz hierarkiko bat bezala antolaturik daude.
- Fitxategiak zein azpi-katalogoak izan ditzateke.
- Fitxategi berezi bezala inplementatzen dira:
 - Sarrerak dituen fitxategi berezia
 - Sarrera bakoitzak inode zenbaki bat eta fitxategi izen bat ditu

```
struct dirent {  
    long d_ino;           // inode zenbakia  
    off_t d_off;  
    unsigned short int d_reclen;  
    char d_name[NAME_LEN]; // sarreraren izena  
};
```

<u>Izena</u>	<u>inode</u>
.	8733108
..	5243238
asignatu.h	8733916
ilarak.c	8733917
ilarak.h	8733918
...	

Katalogoak UNIX-en

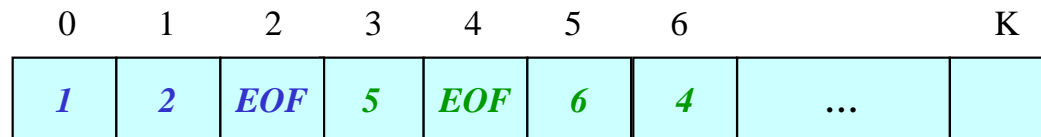


Fitxategi baten informazioa (edukia edo datuak)

□ Edukia edo datuak

➤ Lista kateatua.

adib. FAT: blokeen indizeen lista



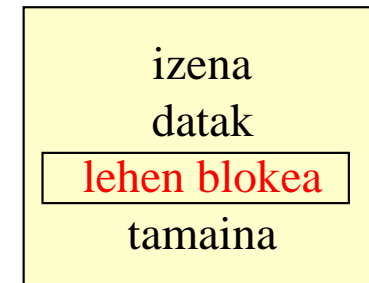
0 -> 1 -> 2 -> EOF

3 -> 5 -> 6 -> 4 -> EOF

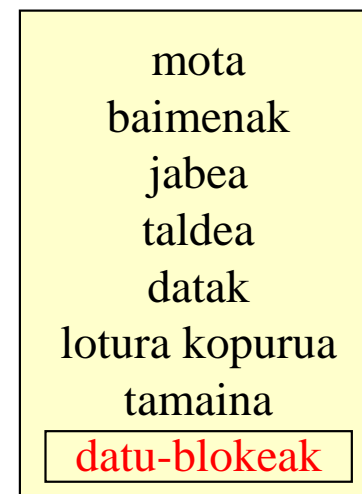
➤ Blokeen indizeen bektorea

adib. UNIX

*Katalogoaren
sarrera*

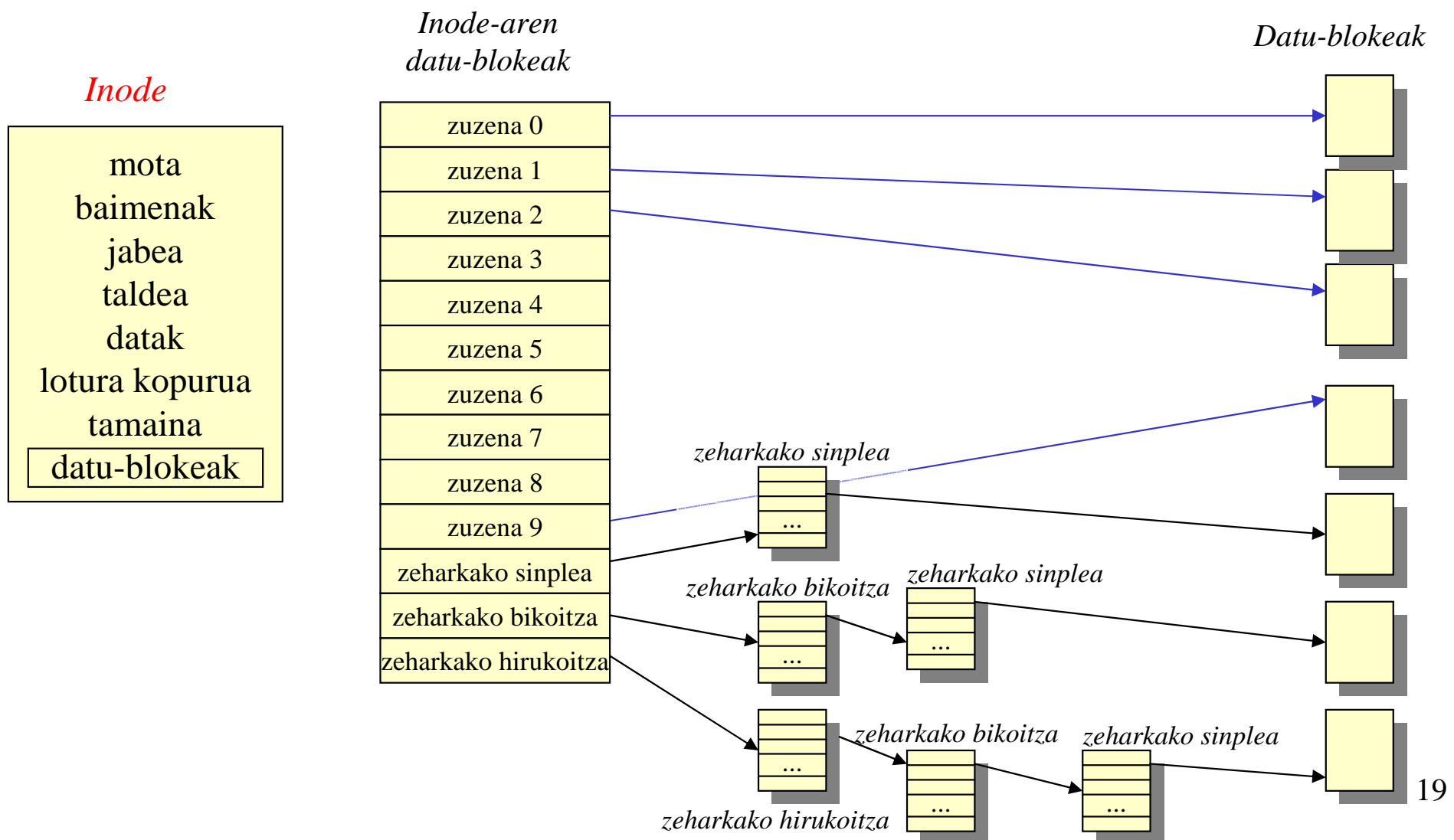


Inode



Fitxategi baten informazioa (edukia edo datuak)

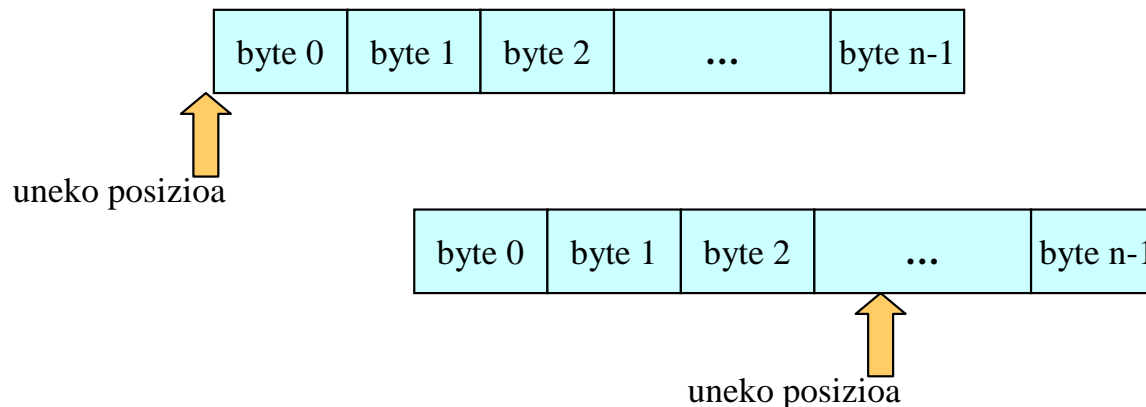
➤ Blokeen indizeen bektorea
adib. UNIX



Informazioa atzitzeko modua

- Atzipen **SEKUENTZIALA**
 - Fitxategia byte-sekuentzia bat bezala ikusten da.
 - k . byte-a atzitzeko aurreko $k-1$ byte-ak atzitu behar dira.
 - n byte atzitu ondoren, uneko posizioa n balioz aurreratzen da.
- Atzipen **ZUZENA** (Ausazkoa)
 - Fitxategiaren edozein byte-a atzitu daiteke aurrekoak atzitu behar izan gabe.
 - UNIXen hau bideratzen da fitxategiaren edozein byte-an kokatzen duen funtzio baten bitartez, eta ondoren atzipen sekuentziala eginez.

Adibideak:



Sistema erabiltzaileanitzak

Accounting: Kontabilitatea

Informazioaren konfidentziasuna

Informazioaren segurtasuna/fidagarritasuna

Baimenen kudeaketa

Atzipenen kontrola/konprobaketa

Baliabideen erabilpenaren kontabilitatea

Baliabideen atzipen murrizpenak (kuotak)

Erabiltzaileen taldeak, jabea, ...

Fitxategien konpartizioa (loturak)

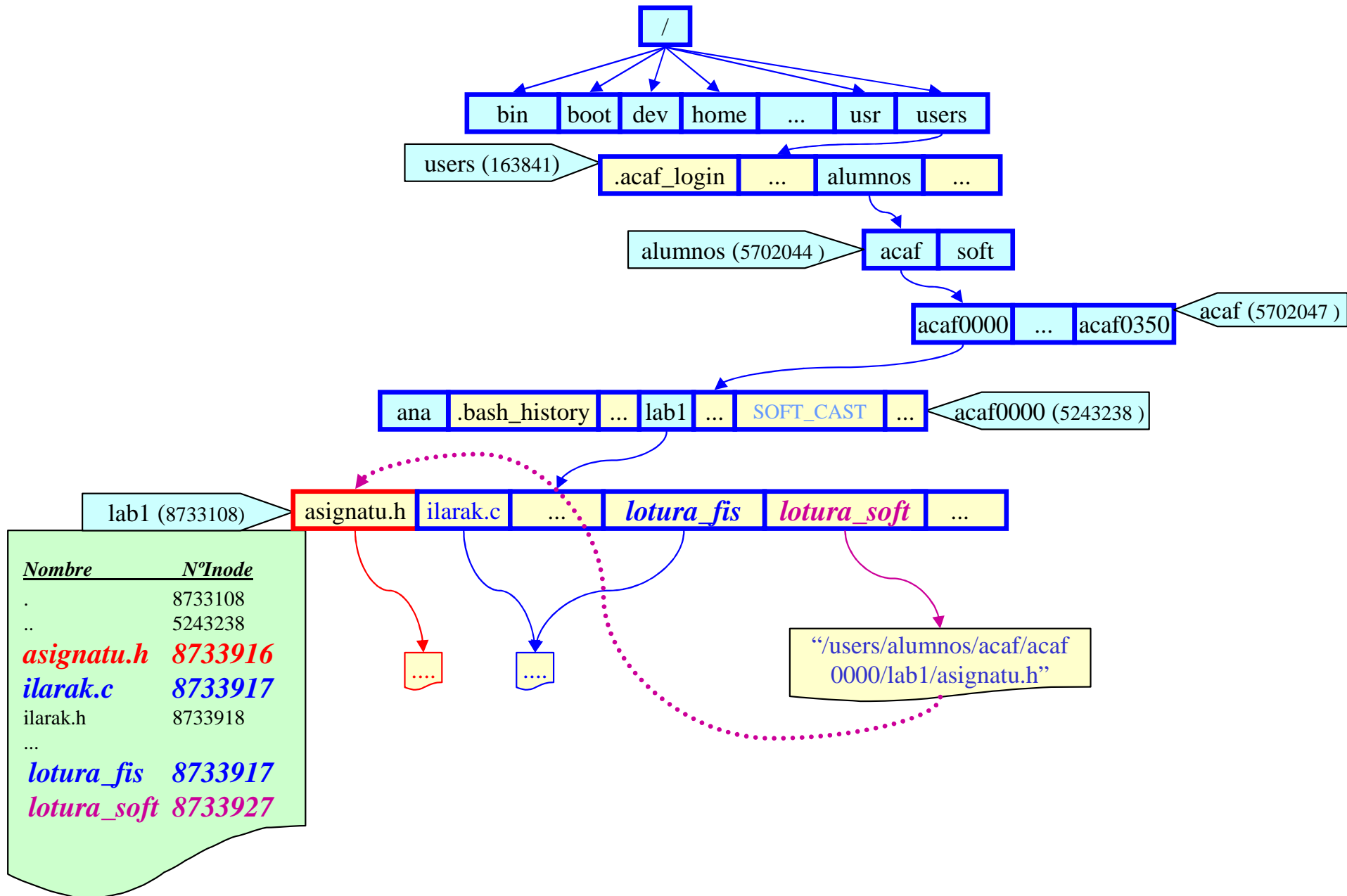
- Fitxategi bereziak, katalogoak: lotura *izena* bidez (SOFT)
- Fitxategi arruntak: lotura *inode zenbaki* bidez (HARD)
- Katalogoak (UNIX): katalogoko sarrerei buruzko informazioa duen fitxategia:

- *Izena*.
- *inode zenbakia*.

<i>Izena</i>	<i>inode</i>
.	8733108
..	5243238
asignatu.h	8733916
ilarak.c	8733917
ilarak.h	8733918
...	

jabea	taldea	besteak
RWX	RWX	RWX

Loturak: *izena* – *inode bidezkoak*



Software lotura

```
[acaf0000@acpt00 acaf0000]$ ls -l
```

```
total 3204
```

```
-rw-r--r--  1 acpcoayj profes    803 may 16  2005 2003Maiatza1.dat
```

```
drwxr-xr-x 13 acpcoayj profes   4096 dic 22  2004 acaf
```

```
drwxr-xr-x  3 acpcoayj profes   4096 jul 11  2005 azterketa_probak
```

```
-rw-----  1 acpcoayj profes  1780 mar 23  2004 esaerak.txt
```

```
-rw-r--r--  1 acpcoayj profes    37 may  8  2005 jon
```

```
-rw-----  1 acpcoayj profes  1780 ene 21  2005 mezu.txt
```

```
drwxr-xr-x 10 acpcoayj profes   4096 may 16  2005 proba05
```

```
drwxr-xr-x 10 acpcoayj profes   4096 mar  9 12:46 proba06
```

```
-rw-----  1 acpcoayj profes  2042 mar 23  2004 refranes.txt
```

```
lrwxrwxrwx  1 acpcoayj profes    24 feb 23 16:25 SOFT_CAST -> /users/alumnos/soft/acaf
```

```
lrwxrwxrwx  1 acpcoayj profes    29 mar  9 12:35 SOFT_EUSK -> /users/alumnos/soft/acaf/eusk
```

```
-rw-----  1 acpcoayj profes 1615081 dic 29  2004 tc.pdf
```

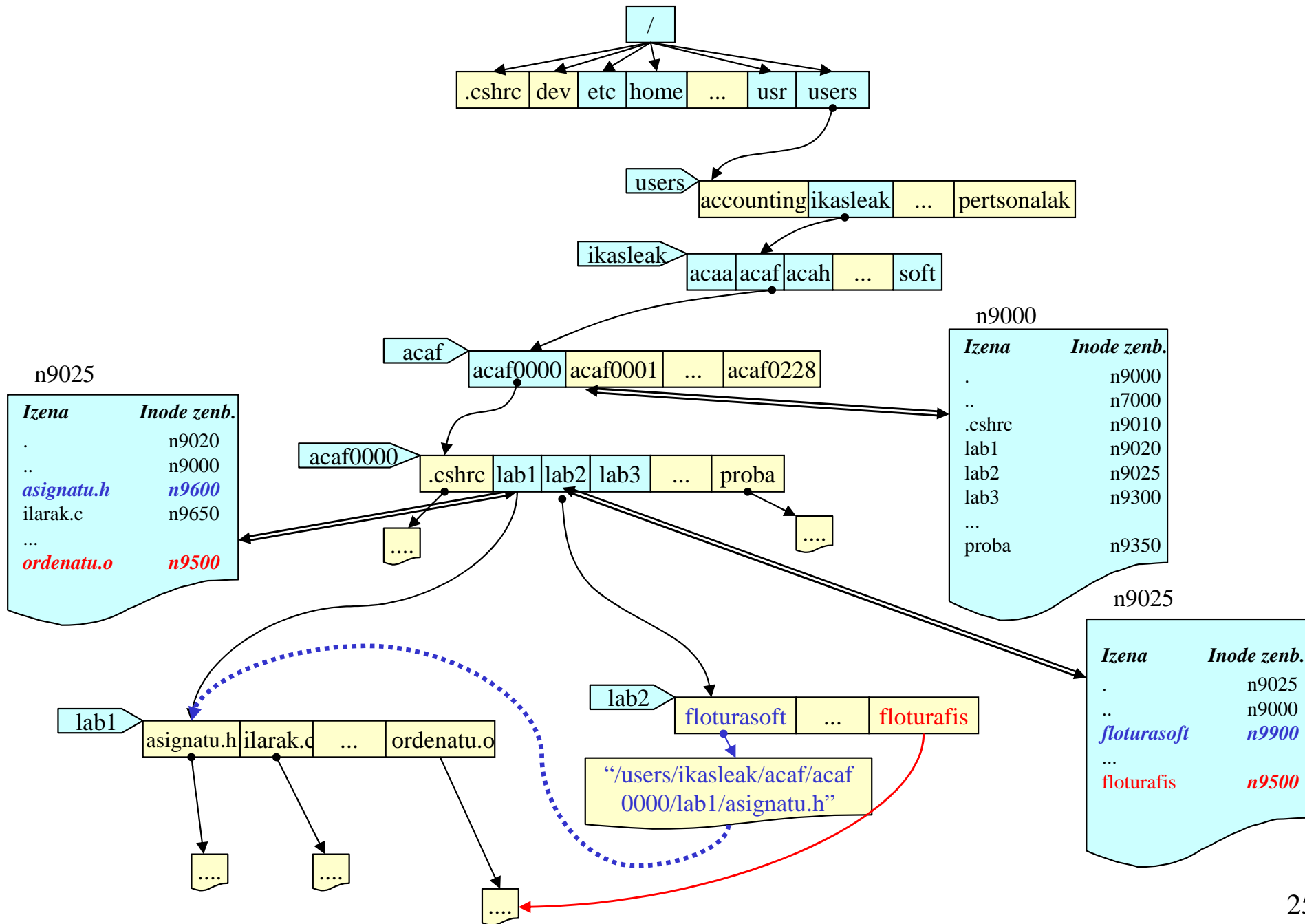
```
-rw-----  1 acpcoayj profes 1615081 dic 29  2004 tc.ps
```

```
[acaf0000 @acpt00 acaf0000]$
```

Hardware lotura vs Software lotura

```
[acpcoayj@acpt00 acpcoayj]$ ll
total 3224
-rw-r--r--      1 acpcoayj profes      803 may 16  2005 2003Maiatzal.dat
drwxr-xr-x    13 acpcoayj profes    4096 mar  5  2007 acaf
drwxr-xr-x     3 acpcoayj profes    4096 jul 11  2005 azterketa_probak
-rw-r--r--      1 acpcoayj profes      11 feb 21 10:04 dat.txt
-rw-r--r--      1 acpcoayj profes      16 feb 26  2007 epa
-rw-----      1 acpcoayj profes    1780 mar 23  2004 esaerak.txt
-rw-----      1 acpcoayj profes    1780 ene 21  2005 mezu.txt
drwxr-xr-x    10 acpcoayj profes    4096 may 16  2005 proba05
drwxr-xr-x    11 acpcoayj profes    4096 abr 20  2007 proba06
-rw-----      1 acpcoayj profes    2042 mar 23  2004 refranes.txt
-rwxr-xr-x     1 acpcoayj profes   12038 feb 21 10:03 scanf
-rw-r--r--      1 acpcoayj profes      305 feb 21 10:02 scanf.c
lrwxrwxrwx     1 acpcoayj profes      24 feb 23  2006 SOFT_CAST -> /users/alumnos/soft/acaf
lrwxrwxrwx     1 acpcoayj profes      29 mar  9  2006 SOFT_EUSK -> /users/alumnos/soft/acaf/eusk
-rw-----      1 acpcoayj profes  1615081 dic 29  2004 tc.pdf
-rw-----      1 acpcoayj profes  1615081 dic 29  2004 tc.ps
[acpcoayj@acpt00 acpcoayj]$ ln epa epa2
[acpcoayj@acpt00 acpcoayj]$ ll
total 3228
-rw-r--r--      1 acpcoayj profes      803 may 16  2005 2003Maiatzal.dat
drwxr-xr-x    13 acpcoayj profes    4096 mar  5  2007 acaf
drwxr-xr-x     3 acpcoayj profes    4096 jul 11  2005 azterketa_probak
-rw-r--r--      1 acpcoayj profes      11 feb 21 10:04 dat.txt
-rw-r--r--      2 acpcoayj profes      16 feb 26  2007 epa
-rw-r--r--      2 acpcoayj profes      16 feb 26  2007 epa2
-rw-----      1 acpcoayj profes    1780 mar 23  2004 esaerak.txt
-rw-----      1 acpcoayj profes    1780 ene 21  2005 mezu.txt
drwxr-xr-x    10 acpcoayj profes    4096 may 16  2005 proba05
drwxr-xr-x    11 acpcoayj profes    4096 abr 20  2007 proba06
-rw-----      1 acpcoayj profes    2042 mar 23  2004 refranes.txt
-rwxr-xr-x     1 acpcoayj profes   12038 feb 21 10:03 scanf
-rw-r--r--      1 acpcoayj profes      305 feb 21 10:02 scanf.c
lrwxrwxrwx     1 acpcoayj profes      24 feb 23  2006 SOFT_CAST -> /users/alumnos/soft/acaf
lrwxrwxrwx     1 acpcoayj profes      29 mar  9  2006 SOFT_EUSK -> /users/alumnos/soft/acaf/eusk
-rw-----      1 acpcoayj profes  1615081 dic 29  2004 tc.pdf
-rw-----      1 acpcoayj profes  1615081 dic 29  2004 tc.ps
[acpcoayj@acpt00 acpcoayj]$ █
```

Loturak: izena – *inode* bidezkoak



UNIX-eko baimenak: `ls -l`

- **rwxrwxrwx**
Jabea Taldea Besteak

Fitxategi mota:

- fitxategi arruntak
- d katalogoak (*directory*)
- l loturak (*link*)
- c karakteretako fitxategi bereziak (gailu motelak)
- b bloketako fitxategi bereziak (gailu azkarrak)
- p izena duten *pipe*-ak edo FIFOak (komunikazioarako)

Unix-eko baimenak

Baimenak	Binary	Octal
- - -	0 0 0	0
- - x	0 0 1	1
- w -	0 1 0	2
- w x	0 1 1	3
r - -	1 0 0	4
r - x	1 0 1	5
r w -	1 1 0	6
r w x	1 1 1	7

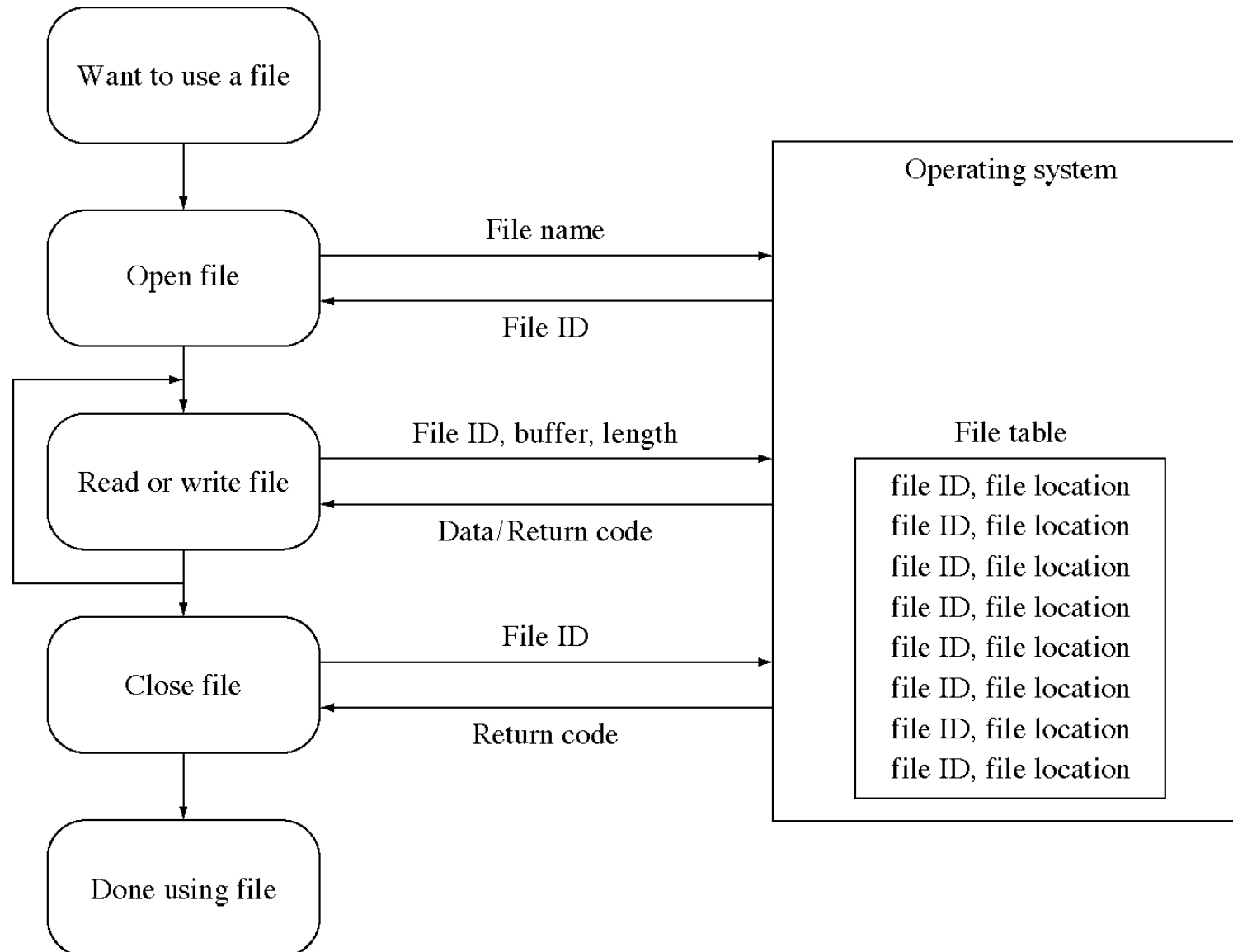
chmod komandoa

- Aurreko zenbaki zortzitarren adierazpena erabiliz, `chmod` komandoaren bitartez fitxategien baimenak alda daitezke
- Sintaxia:
`chmod BaimenenMaskara fitxategia`
- Adibidea:
`chmod 755 public_html`
- Oro har, katalogo eta exekutagarriek “755” baimenak izan ohi dituzte, aldiz, beste fitxategiek “644” baimenak

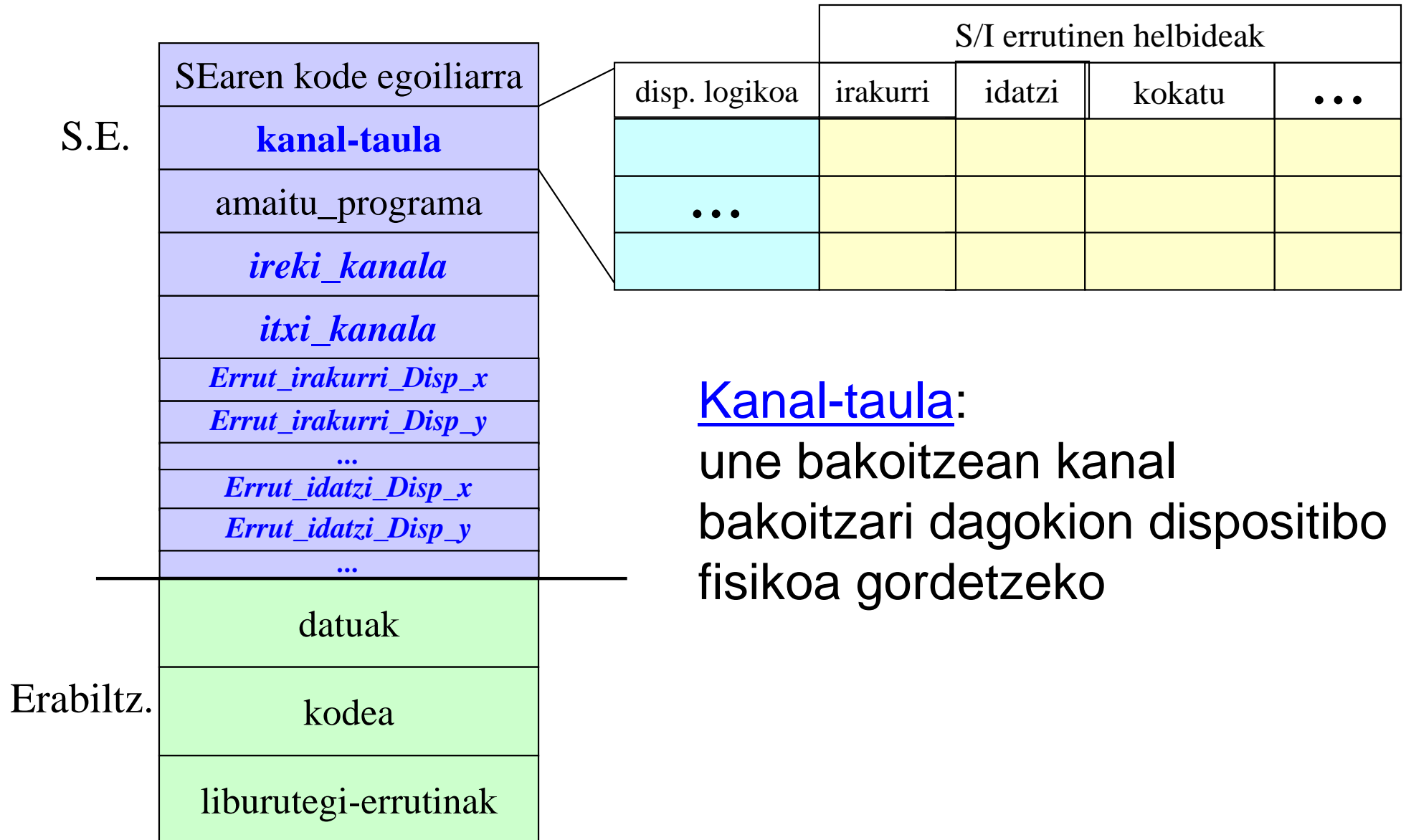
Dispositiboekiko independentzia

- **Arazoa:** dispositibo bat erabiltzerakoan:
⇒ dispositiboaren errutinak esplizituki erreferentziatu
⇒ dispositiboz aldatzean programak ere aldatu egin behar ziren ⇒ programak berriro konpilatzeko beharra
- **Aurrerapena:** dispositibo logikoen erabilpena, kanalak
- Kanalen bidez, **dispositiboekiko independentzia** lortzen da
- Kanal-taula: une bakoitzean kanal bakoitzari dagokion dispositibo fisikoa gordetzeko

Steps in using a file



Dispositiboekiko independentzia



Dispositiboekiko independentzia

liburutegi-errutinak

```
sar_lib_errut (int disp_logikoa, char*bektore,  
               int kop, int asink_sink)  
{ par0 = disp_logikoa;  
  par1 = SARRERA;  
  par2 = bektore;  
  par3 = kop;  
  par4 = asink_sink;  
  sistema_deia(SARRERA_IRTEERA);  
}
```

```
irt_lib_errut (int disp_logikoa, char *bektore,  
               int kop, int asink_sink)  
{ par0 = disp_logikoa;  
  par1 = IRTEERA;  
  par2 = bektore;  
  par3 = kop;  
  par4 = asink_sink;  
  sistema_deia(SARRERA_IRTEERA);  
}
```

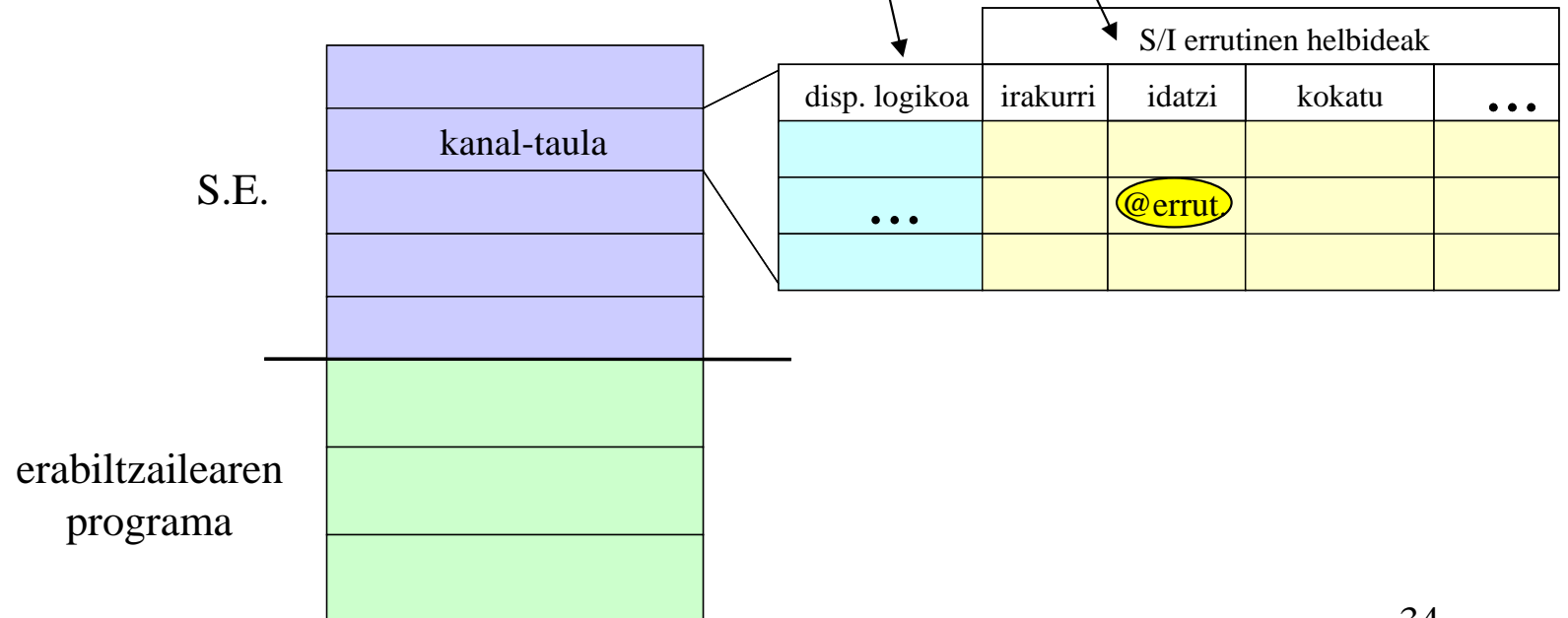
erabiltzailearen programa (sink.)

```
main ()  
{ int f1, f2;  
  char bekt[80];  
  ireki(f1, "sarrera_dispositiboa");  
  ireki(f2, "irteera_dispositiboa");  
  while (TRUE)  
  {  
    sar_lib_errut(f1, bekt, 80, SINKRO);  
    irt_lib_errut(f2, bekt, 80, SINKRO);  
  }  
  itxi(f1);  
  itxi(f2);  
}
```

Dispositiboekiko independentzia

sakabanatze-errutina

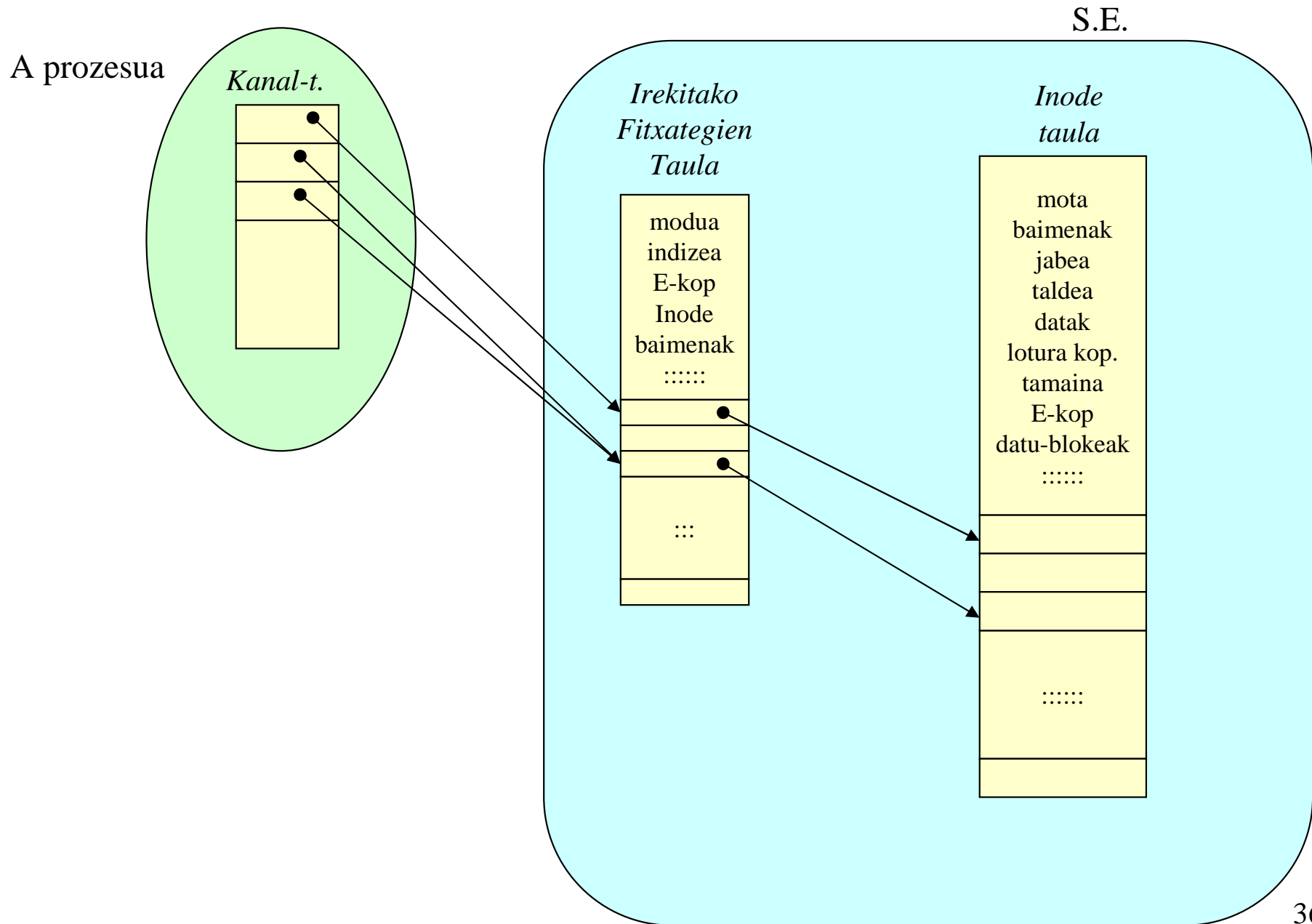
```
sakabanatze_errutina ()  
{ int (*errut_edo_errore)();  
  errut_edo_errore = lortu_errutina(par0, par1);  
  if (!erroreak(DISP, &errut_edo_errore))  
    (*errut_edo_errore)();  
}
```



Kanal lehenetsiak (estandararak)

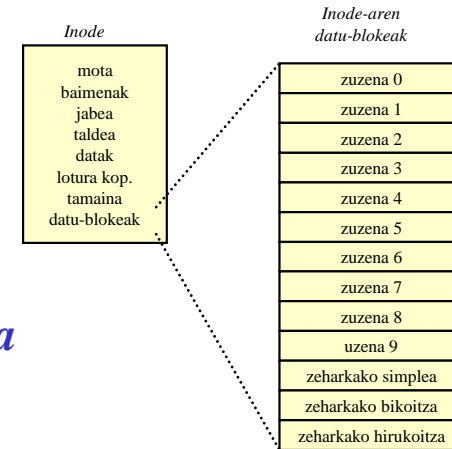
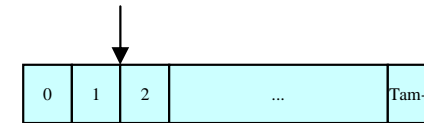
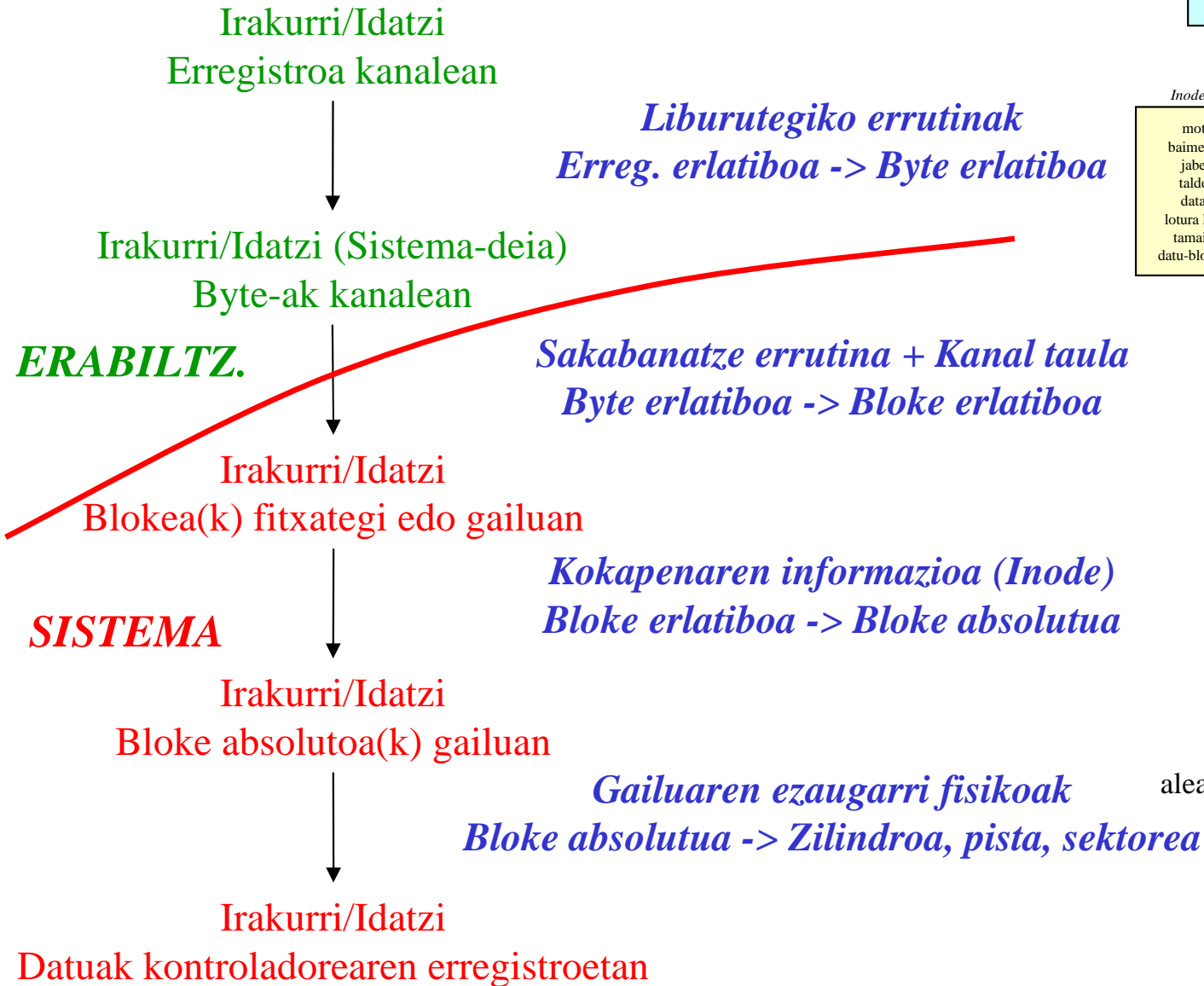
- Kanal estandararak = dispositibo lehenetsiak (sarrera, irteera, errore). Unixen: 0, 1, 2 (*stdin, stdout, stderr*)
- KI-ak mekanismoak ditu kanaletan eragiteko:
 - **kanalak berbideratzeko** mekanismoak (< > >&)
(<< >> >>&)
 - **pipeak** (|)
- **guraso-ume** herentzia (KI-erabiltzailearen programa)

Dispositiboekiko independentzia UNIXen

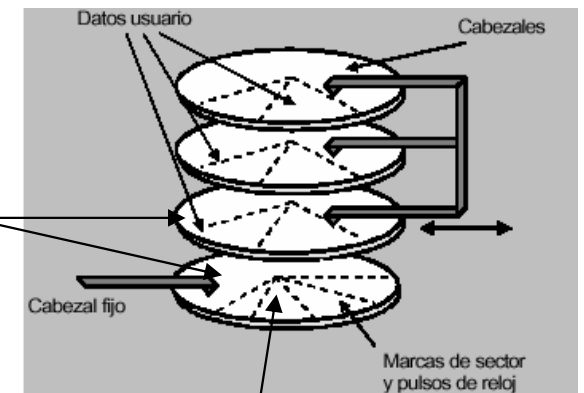
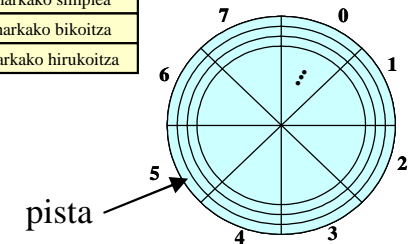
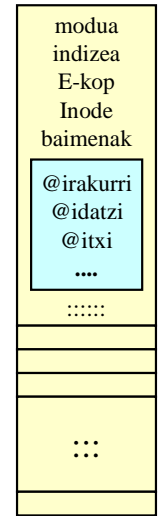


Helbideen itzulpenaren mailak

Itzulpen mailak

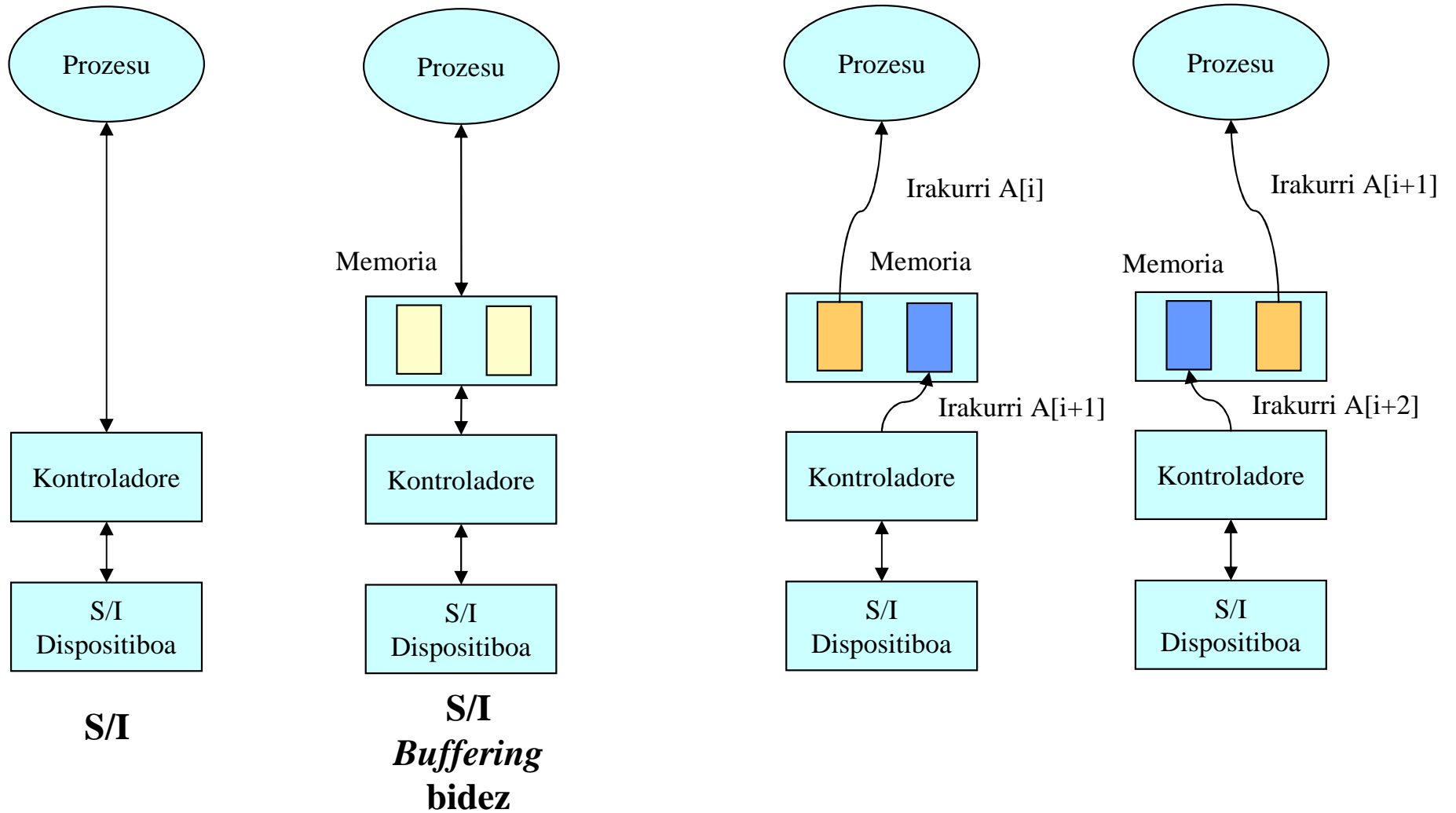


Irekitako Fitxategien Taula



sektorea

S/I Buffering bidez



UNIXeko S/Iko sistema-deiak

- **Irekiera / itxiera / irakurketa / idazketa / kokapena**

- int **open** (char ***path**, int **flags**, int **baim**);
- int **creat** (char ***path**, int **baim**);
- int **close** (int **fd**);
- int **read** (int **fd**, char ***buf**, unsigned **luz**);
- int **write** (int **fd**, char ***buf**, unsigned **luz**);
- long **lseek** (int **fd**, long **despl**, int **pos_kod**);
/* hasiera: 0, uneko_pos: 1, amaiera: 2 */
/* <stdio.h>: SEEK_SET, SEEK_CUR, SEEK_END */

Flags (aukerak):

O_RDONLY	O_WRONLY	O_RDWR	O_NDELAY
O_APPEND	O_DSYNC	O_RSYNC	O_SYNC
O_NOCTTY	O_CREAT	O_EXCL	O_TRUNC

UNIXeko S/Iko sistema-deiak (2)

- **Katalogoen kontrola**

- int **chdir** (char ***path**);
- int **mkdir** (char ***path**, int **baim**);
- int **rmdir** (char ***path**);
- int **link** (char ***path_jatorria**,
 char ***path_helburua**);
- int **symlink** (char ***path_jatorria**,
 char ***path_helburua**);
- int **unlink** (char ***path**);

UNIXeko S/Iko sistema-deiak (3)

- **Fitxategi eta dispositiboen kontrola**

- `int stat (char *path,
 struct stat *sbuf);`
- `int fstat (int fd,
 struct stat *sbuf);`
- `int fcntl (int fd, int komandoa,
 int argum);`
- `int ioctl (int fd, int komandoa,
 struct termio *buf);`

stat datu-egitura

- **st_dev:** fitxategiaren euskarri den unitatearen zenbakia (short)
- **st_ino:** inode zenbakia (ushort)
- **st_mode:** modua (short) eta baimen bitak
- **st_nlink:** lotura kopurua (short)
- **st_uid:** jabearen identifikatzailea (ushort)
- **st_gid:** taldearen identifikatzailea (ushort)
- **st_rdev:** fitxategi berezientzako unitate zenbakia (short)
- **st_size:** tamaina (long) (0 fitxategi berezientzako)
- **st_atime:** azken atzipenaren ordua (long)
- **st_mtime:** azken aldaketaren ordua (long)
- **st_ctime:** sorrera ordua (long)
- ...

UNIXeko S/Iko sistema-deiak (4)

termio datu-egiturarekin erlazionatutako komandoak

TCGETA	Lortu informazioa <i>termio</i> egituran.
TCSETA	Idatzi informazioa <i>termio</i> egituran oinarrituz.
TCSETAN	Aurreko TCSETA bezala, baina itxaron eginez martxan dagoen S/I ekintzaren amaiera arte.
TCSETAF	Aurreko TCSETA bezala, baina zain dauden S/I eragiketak ezabatuz.

UNIXeko S/Iko sistema-deiak (5)

- **Erabiltzaileanitzza**

- int **chmod** (char ***path**, int **modua**);
- int **chown** (char ***path**, int **jabea**,
int **taldea**);
- int **access** (char ***path**, int **modua**);
R_OK, W_OK, X_OK
- int **umask** (int **modua**);

C liburutegiko funtzioak

- **Katalogoekin erlazionatutako funtzioak**

- `DIR *opendir (char *path);`
- `int closedir (DIR *dir);`
- `struct dirent *readdir (DIR *dir);`

Erabilpen adibidea:

```
DIR *dir;
struct dirent *sarrera;

dir = opendir("dir1");
sarrera = readdir(dir);
write(1, sarrera->d_name, strlen(sarrera->d_name));
closedir(dir);
```

```
struct dirent
{
    ...
    char d_name[256];
}
```

C liburutegiko funtzioak

- **Irekiera / itxiera / irakurketa / idazketa**

- FILE ***fopen** (char ***path**, char ***mota**);
- int **fclose** (FILE ***fd**);
- int **fread** (void ***buf**, int **tamaina**,
int **kopurua**, FILE ***fd**);
- int **fwrite** (void ***buf**, int **tamaina**,
int **kopurua**, FILE ***fd**);

KARAKTERE KATEEKIN LAN EGITEKO FUNTZIOAK

- **strcpy (str1, str2)** → str2, str1-ean kopiatzen da
`char *strcpy(char *helburu, const char *jatorri);`
- **strcmp (str1, str2)** → bi karaktere kateak konparatzeko
`int strcmp(const char *str1, const char *str2);`
- **strcat (str1, str2)** → kateamendua: `str1 := str1 + str2`
`char *strcat(char *str1, const char *str2);`
- **strlen (str)** → karaktere katearen luzera itzuli
`int strlen(const char *str);`
- **atoi (str)** → string motatik integer motara (Ascii TO Integer)
`int atoi(const char *str);`
- **strstr (str1, str2)** → str2, str1-en azpikatea den edo ez esan
`char *strstr(const char *str1, const char *str2);`

Adibidea: *kopiatu*

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>

#define errore(a) {perror(a); exit(1);};
#define BUFSIZE 512

main(int argc, char *argv[]) /* kopiatu.c */
{
    int n;
    char buf[BUFSIZE];

    /* irakurketa eta idazketa zikloa */
    while ((n = read(0, buf, BUFSIZE)) > 0)
        if (write(1, buf, n) != n) errore("write");

    if (n == -1) errore("read");
}
```

Adibidea: *buztana*

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <fcntl.h>

#define errore(a) {perror(a); exit(1);};

main(int argc, char *argv[]) /* buztana.c */
{
    int fd, k;
    char buf[80];

    if (argc != 2) errore("argumentuak gaizki");
    if ((fd = open(argv[1], O_RDONLY, 0666)) == -1)
        errore("open");
    if (lseek(fd, -10, 2) == -1) errore("lseek");
    if ((k = read(fd, buf, 10)) != 10) errore("read");
    if (write(1, buf, k) != k) errore("write");
    if (close(fd) == -1) errore("close");
}
```

Adibidea: *oihartzuna_on_off*

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <termio.h>

#define errore(a) {perror(a); exit(1);};

main(int argc, char *argv[]) /* oihartzuna_on_off.c */
{
    struct termio tm;

    if (ioctl(0, TCGETA, &tm) == -1) errore("ioctl 1");
    if (tm.c_lflag & ECHO)
        tm.c_lflag = tm.c_lflag & ~ECHO;
    else
        tm.c_lflag = tm.c_lflag | ECHO;
    if (ioctl(0, TCSETA, &tm) == -1) errore("ioctl 2");
}
```

Adibidea: *kat*

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <sys/types.h>
#include <dirent.h>
#define errore(a) {perror(a); exit(1);};

main(int argc, char *argv[]) /* kat.c */
{
    DIR *dir;
    struct dirent *sarrera;

    if (argc != 2) errore("argumentuak gaizki");
    if ((dir = opendir(argv[1])) == NULL)
        errore("opendir");
    while ((sarrera = readdir(dir)) != NULL) {
        write(1, sarrera->d_name, strlen(sarrera->d_name));
        write(1, "\n", 1);
    }
    if (closedir(dir) == -1) errore("closedir");
}
```

Adibidea: *berrizendatu*

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>

#define errore(a) {perror(a); exit(1);};

main(int argc, char *argv[]) /* berrizendatu.c */
{
    if (argc != 3) errore("argumentuak gaizki");
    if (link(argv[1], argv[2]) == -1) errore("link");
    if (unlink(argv[1]) == -1) errore("unlink");
}
```

Adibidea: *kopiatu-kop*

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#define errore(a) {perror(a); exit(1);};
#define BUFSIZE 512

main(int argc, char *argv[]) /* kopiatu-kop.c */
{
    int n, kop = BUFSIZE;
    char buf[BUFSIZE];

    if (argc == 2) {
        kop = atoi(argv[1]);
        if ((kop < 1) || (kop > BUFSIZE)) errore("kop");
    }
    /* irakurketa eta idazketa zikloa */
    while ((n = read(0, buf, kop)) > 0)
        if (write(1, buf, n) != n) errore("write");
    if (n == -1) errore("read");
}
```

Adibidea: *fkopiatu-kop*

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#define errore(a) {perror(a); exit(1);};
#define BUFSIZE 512

main(int argc, char *argv[]) /* fkopiatu-kop.c */
{
    int n, kop = BUFSIZE;
    char buf[BUFSIZE];

    if (argc == 2) {
        kop = atoi(argv[1]);
        if ((kop < 1) || (kop > BUFSIZE)) errore("kop");
    }
    /* irakurketa eta idazketa zikloa */
    while ((n = fread(buf, sizeof(char), kop, stdin)) > 0)
        if (fwrite(buf, sizeof(char), n, stdout) != n)
            errore("fwrite");
    if (n == -1) errore("fread");
}
```

kopiatu-kop vs fkopiatu-kop

```
$ time kopiatu-kop 512 <f_100KB >/dev/null  
real    0m0.035s  
user    0m0.011s  
sys     0m0.020s
```

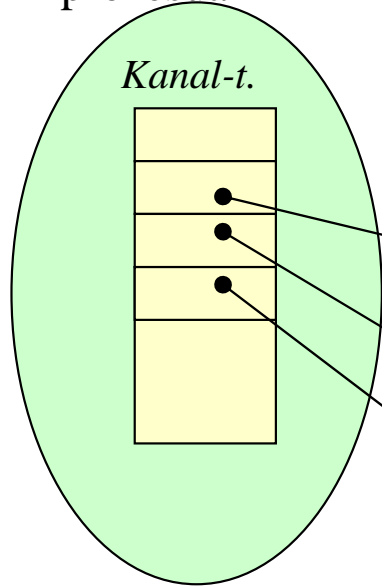
```
$ time fkopiatu-kop 512 <f_100KB >/dev/null  
real    0m0.020s  
user    0m0.006s  
sys     0m0.012s
```

```
$ time kopiatu-kop 1 <f_100KB >/dev/null  
real    0m8.666s  
user    0m2.817s  
sys     0m5.614s
```

```
$ time fkopiatu-kop 1 <f_100KB >/dev/null  
real    0m0.034s  
user    0m0.006s  
sys     0m0.025s
```

open, creat eta close sistema-deiak

A prozesua

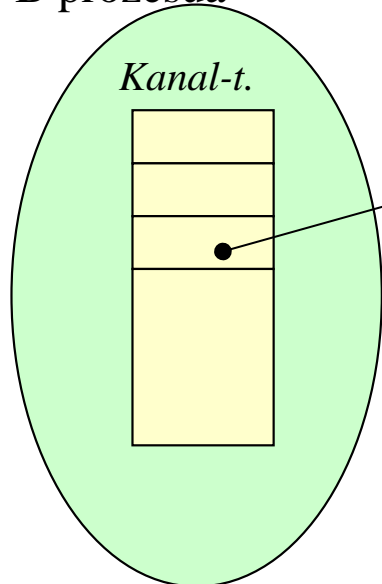


open / creat

close

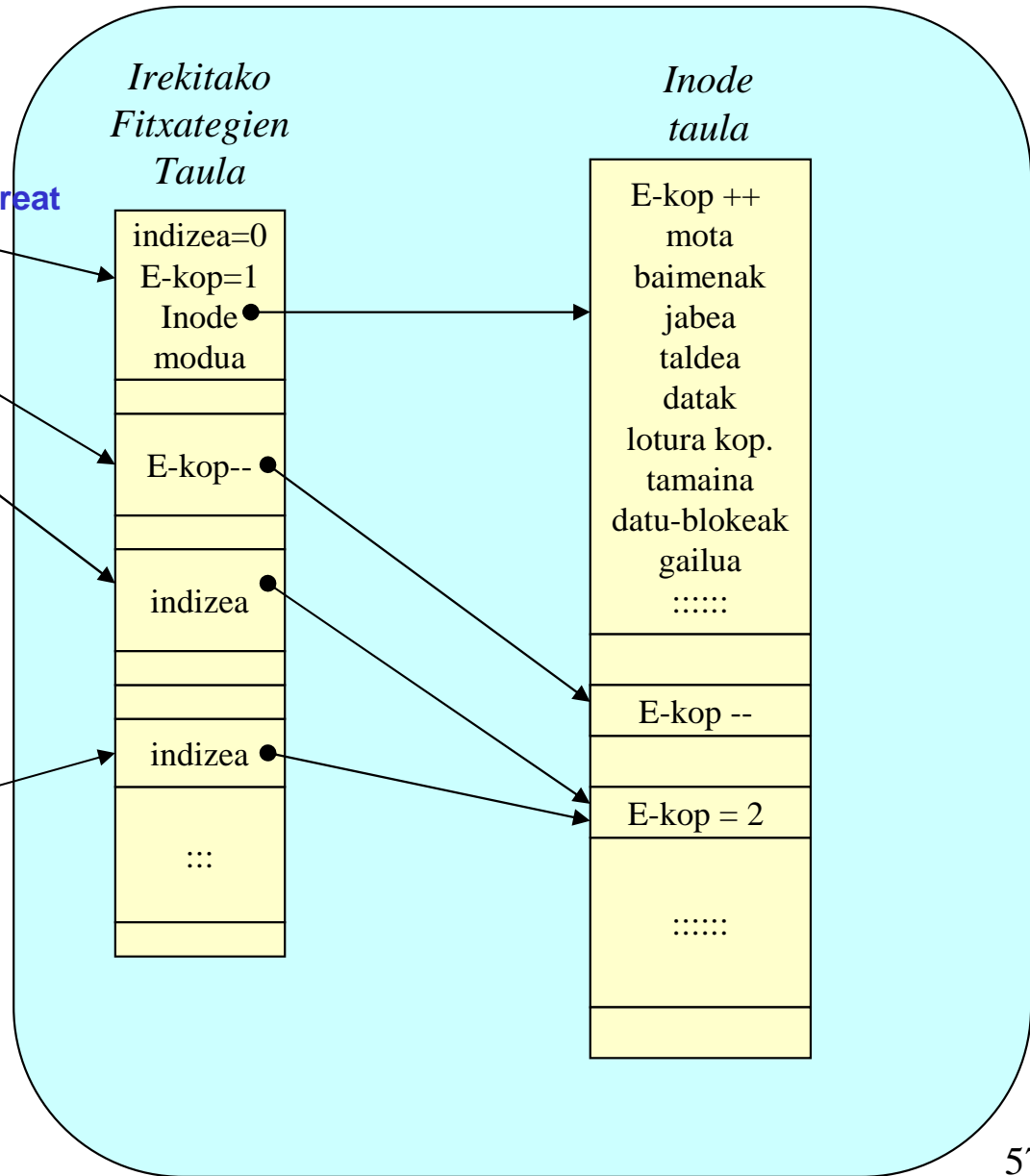
open

B prozesua



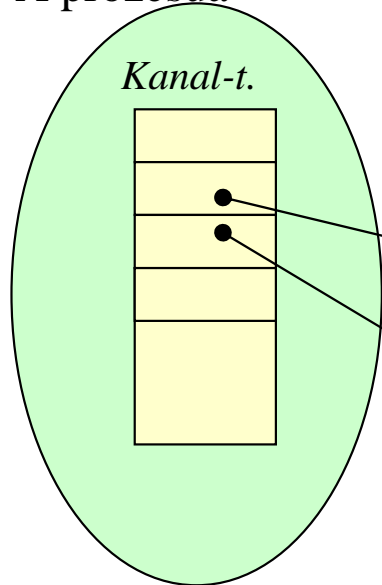
open

S.E.

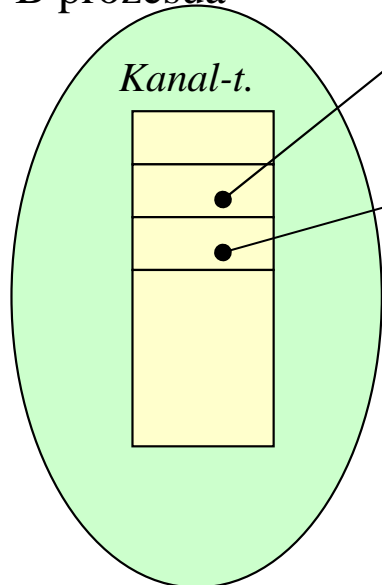


read, write eta lseek sistema-deiak

A prozesua



B prozesua



S.E.

