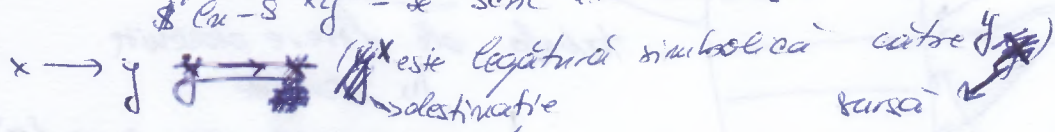


• Legătură simbolică

\$ln -s xy -x sene în director:



- dezavantaj:

- \$rmv y (persta legătura: - file not found)
- \$ln x z (nu se vede că x și z aparțin aceluiași fișier fizic)

! legătura fizică nu merge pe directoare, doar pe fișiere

• Caracteristici ale modului:

- nr de octeți (dimensiunea în octeți)
- data ultimei modificări
- alte date (data creării fișierului, data ultimei deschideri--)

! Dacă am drept de scriere în director pot face și chiar dacă nu am drepturi pe fișier

• Apeluri de sistem

- int stat (char *fs, struct stat *st) - nu deschide modul legăturii simbolice // deschide efectiv modul.

nume
introdu în director
(fișierului)

informațiile
modului

struct stat st;

stat ("abcde", &st)

> examen!

- st_mode - memorează pe 16 biți cele 9 drepturi de acces + 3 drepturi speciale + pe 4 biți memorează tipul fișierului

st_mode & S_IFDIR != 0 → director

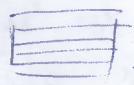
Macro: S_ISDIR(st.st_mode) != 0 → director

MANSTAT - da toate constantele

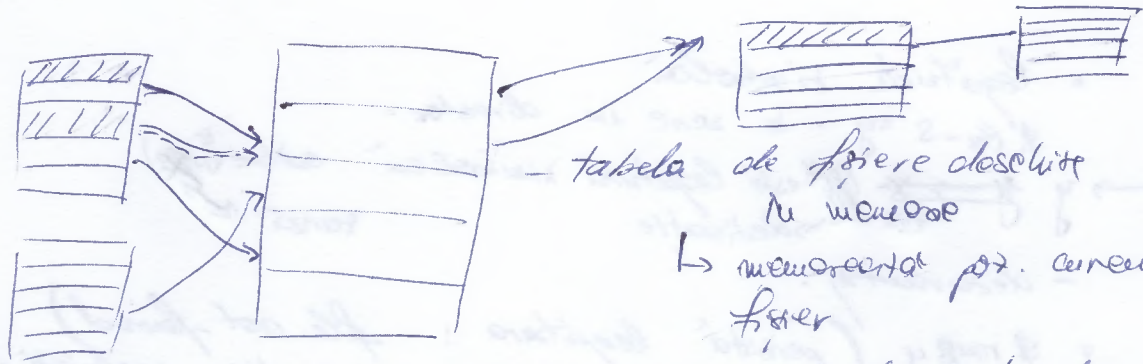
- int lstat (char *fs, struct stat *st) - deschide modul legăturii simbolice

• Exploatare fișier

Struct. fișierului
în memorie



Încare proces posedă o
tabelă de descriptor asupra fișierelor
deschise



- o bijectie
deschideri separate → intrări separate în tabela de fișiere deschise

- tabela de intrări - încarcarea a singurii date în tabel în tabelă (nu există copii)
- un intrat poate să apară o dată în tabelă:
tabela de blocuri → fiecare proces poate să pună blocuri.

• Apeluri sistem:

• Deschiderea sistemului fișierului

open - este apel de sist., face apel de sistem
- biblioteca C standard

int open (char *nume, int mod, int dr)

↓
poate să creeze
un fișier nou
dr - drepturi de la stmode

se creează prin disp
bit ce bit este în
constante (def în 'header')

≥ 0 - proces
nu ocupă în
tabela de
descrieri

- 1 - eroare

- O_CREAT - dacă fișierul nu există este creat.
- drepturi pe director
- O_EXCL - exclusiv creare (se "supera" dacă există)
- O_TRUNC - dacă fișierul există, conținutul se va trunchia

Beschwerden

- "r" O-ONLY

- "a" o ONLY 10-CREAT 10-ATTENDS
Acord cu deschis au fost n' vreau sa lucrez cu bibl. sistem?
- descriptiv

Jurers :

FILE* fopen (const char*, const char*) - ret null → mod nu este
compatibil cu
O_RDONLY, ... etc.

Opera NU returneata cel mai mic descriptie liber.

• Active:

int read (int desc, void *p, int lg)

descriptor | address max. len. octets

alt. file sig. | alt. file sig. | alt. file sig.

-1 encode - descriptor
invalid
EINVA

Success - nr de seturi
care au fost celti
(lg sau cu ceto, lg)

At urm. real va da \bullet ~~se ret.~~ \bullet val $< lq$
 \rightarrow a ajuns la finalul f_3

• Senere!

```
int write (int desc, void *p, int sz)
```

-1 Error

forces - no de seters
can feel some /

- cool return < log
- error : disk full

equivalent: fronte

• Apeluri de inchidere: (examen!)

int close (int fd)

- `int dup(int desc)` - duplicarea descriptorului
 ↳ cel mai mic descriptor liber

↳ el mio mio descriptor liber

Ex

myErr

int d;

d = open("myErr", O_WRONLY | O_CREAT | O_TRUNC); // drepturi
implicit

close(2); // inchid stderr

dup(d); // dup x va duplica in c.m.m. desc. liber = 2

close(d); // scrie prin 2 in myErr

Blocare asupra anumit zone de fișier

int fcntl (int desc, int op, ...)

operatiune (data prin defunc) L in func.
de op poate sa
mai urmeze un argument

op = ...

F_GETFD - nu e necesar al terlea f
- info pe bit - caract. desc.

FD_CLOEXEC

F_SETFD - modife

int fd;

fd = fcntl(d, F_GETFD);

fd |= FD_CLOEXEC

fcntl(d, F_SETFD, fd);

F_GETFL - caract pe bit de la 0-...

F_SETFL - modife

→ O_APPEND

F_DUPFD - face duplicare fara al terlea arg,