

FAMILIA DE APELURI EXEC

- are ca prim argument un drac \* numele fișierului.

① `int execl (char *exe, char *arg0, char *arg1, ..., char *argn, NULL)`

ex: `ls -l`

`execl ("/bin/ls", "ls", "-l", NULL);`

② `int execlp (char *exe, char *arg0, char *arg1, ..., char *argn, NULL)`

Obs: nu mai scutim să dăm exe (executabilul) cu cale completă pt. că ține cont de variabilele de mediu PATH - listă de directoare cu cale completă separată prin : în UNIX (; pe windows)

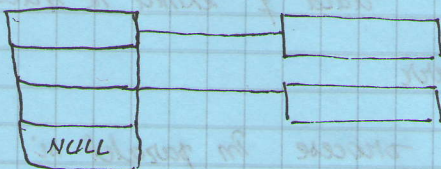
PATH = .: \$PATH - în PATH se adaugă directorul curent

`export PATH`

③ `int execl (char *exe, char *arg0, ..., char *argn, NULL, char **env)`

env - listă de adrese către grăuri de caractere, de care trebuie să aibă mintaxa variab = valoare.

ultimul parametru = NULL



④ `int execl (char *exe, char **argv)`



cale completă (cale care nu se termină în "p").

ultimul parametru NULL

⑤ `int execlp (char *exe, char **argv)`

⑥ `int execl (char *exe, char **argv, char **env)`



## COMUNICARE ÎNTRU PROCES

write(a);

...

write(b);

read(a);

read(b);

1. FIȘIERE DE TIP SPECIAL

2. IPC SYSTEM V

3. SEMNALE

### 1. FIȘIERE DE TIP SPECIAL

- se folosesc două tipuri de fișiere: pipe, socket

- sistemul de operare deschide automat trei fișiere pe terminal

- stdin → tastatură
- stdout
- stderr

} ecran

com < f - stdin nu mai e pe terminal, e pe fișierul f (citește din fișierul f)

com > f - redirectionare stdout (scrie în fișierul f) - w

com >> f - redirectionare stdout

diferența la > f dacă f există

la >> f dacă f există se face append

com 2 > f - redirectionează stderr

c1 | c2 | ... | cm - se lansează cele n procese în paralel și

stdout c1 = stdin c2 ...

pe ecran ? lui cm

1 pipe - comenzi filtru

- cum luată ? n-1 fișiere de tip special pipe

- pt. comunicare unidirecțională între procese



socket

- pt. comunicare bidirecțională

- pt. comunicarea între procese aflate pe margini diferite în rețea



## 2. IPC SYSTEM V

- cozi de mesaje (msg), vectori de semafoare (sem), zone de memorie (shm)
- Caracteristici:

- identificatorul extern (\*)  
intern

(\*) key-t ftok (char \* f, int nr)  
↓  
fixat, care tb să existe

- nu este apel sistem
- returnează numele unui identificator extern

msg

Pt fiecare IPC trebuie să avem un apel sistem care să recupereze identifi intern pornind de la cel extern

int msgget (key-t cheie, int opt)

-1 în caz de eșec

>0 în caz de succes: identificatorul intern al cozii de msg.

cheie ftok, IPC\_PRIVATE

opt IPC\_CREAT | IPC\_EXCL | drepturi de acces

int msgsnd (int id, void \*p, int lg, int opt)

↓  
adresa din memorie  
unde am construit msg

↓  
lungime msg

typedef struct myMes {

unsigned long tip;

}

opt: IPC\_NOWAIT



int msgrecv ( int id, void \*p, int lg, long tip, int opt )

- citește msg și îl pune la adresa p - msg se distruge  
(cel mai vedu msg de tipul tip)

tip > 0 cel mai vedu msg de tipul respectiv

= 0

- " -

dim coadă indiferent de tip

< 0

- " -

de tipul ? |tip|

opt : IPC\_NOWAIT, MSG\_NOERROR

↳ transformă msg la lg dacă e  
peea mare

int msgctl ( int id, int oper, struct msgid\_cls \*p )

oper - tipul operației dorite

IPC\_RMID șterge IPC complet din sistem.