

Sisteme de operare - lab11

Comunicatii între procese Unix: FIFO

Aspecte teoretice

Una dintre modalitățile de comunicare între procese în Unix este cea prin intermediul canalelor de comunicație (numite pipes, în limba engleza). Practic este vorba despre o "conductă" (un buffer) prin care pe la un capăt se scriu mesajele, iar pe la celălalt capăt se citesc - deci este vorba despre o structură de tip coadă, adică o listă FIFO (First-In,First-Out).

Aceste canale sunt de două categorii:

- pipe-uri interne: aceste "conducte" sunt create în memoria internă a sistemului Unix;
- pipe-uri externe: aceste "conducte" sunt fișiere de un tip special, numit fifo, deci sunt păstrate în sistemul de fișiere (aceste fișiere fifo se mai numesc și pipe-uri cu nume).

Canale externe.

Pipe-urile cu nume au fost introduse de la versiunea UNIX SYSTEM III, fiind o combinație între fișierele obișnuite și pipe-uri. Au un nume simbolic, fiind create într-un director, ca orice fișier obișnuit, dar păstrează toate caracteristicile fișierelor pipe. Fișierele **pipe cu nume** se mai numesc și **fișiere FIFO**.

Avantajul fișierelor pipe cu nume față de fișierele pipe este reprezentat de faptul că procesele care comunică între ele nu trebuie să își trimită descriptorii. Ca urmare comunicarea se poate face între procese care nu trebuie să provină din procesul care a deschis fișierul, cum este cazul la fișierele pipe. În acest caz, fiecare proces își poate deschide în citire sau scriere fișierul pipe cu nume, important fiind numele său simbolic.

Observație

Un fișier FIFO este vizibil cu comanda UNIX, `ls -l`, care afișează litera *p* în primul câmp al drepturilor de acces.

Crearea unui fișier FIFO

În versiunea SVR3, crearea unui fișier FIFO se realiza prin apelul funcției de sistem *mknod*. Funcția are rolul de a crea directoare și fișiere speciale. Interfața acestei funcții, particularizată pentru crearea fișierelor FIFO este:

```
#include <sys/types.h>
#include <sys/stat.h>
```

```
int mknod(char *nume, int tip_per_acces, 0)
unde:
```

- *nume* reprezintă fișierului FIFO.

- *tip_per_acces* este un întreg ce indică tipul fișierului *S_FIFO* (în cazul pipe-urilor cu nume) și drepturile de acces (r,w,x pentru proprietar, grup și ceilalți utilizatori). Parametrul se dă sub forma: *S_FIFO | drepturi*.

Observație

Crearea unui fișier FIFO este similară cu crearea unui fișier obișnuit dacă se folosește funcția *mkfifo* (care de fapt apelează tot *mknod*). Interfața acestei funcții este:

```
#include <sys/types.h>
#include <sys/stat.h>
```

```
int mkfifo(const char *nume, mode_t tip_perm_acces)
```

unde semnificația argumentelor este identică cu a celor de mai sus. Ultimul argument lipsește în acest caz.

Deschiderea unui fișier FIFO

Deschiderea unui fișier FIFO în citire sau scriere se face ca pentru orice fișier obișnuit folosind funcția *open*.

Citirea dintr-un fișier FIFO

Citirea din fișierele FIFO se realizează cu ajutorul funcției de sistem *read*, folosită în mod identic ca la fișierele obișnuite.

Scrierea într-un fișier FIFO

Scrierea într-un fișier FIFO se face cu ajutorul funcției sistem *write*, folosită în mod identic ca la fișierele ordinare. Dacă indicatorul *O_NDELAY* sau *O_NONBLOCK* este poziționat în momentul deschiderii și fișierul FIFO este plin sau nu are suficient spațiu pentru scrierea completă, procesul nu este blocat până la citirea datelor de către alt proces, ci funcția *write* returnează zero.

Dacă se încearcă scrierea într-un fișier FIFO care nu are atașat un proces care să citească din el, se generează semnalul *SIGPIPE*.

În cazul în care există mai multe procese care scriu într-un fișier FIFO, se consideră că scrierea este o operație atomică atâta timp cât volumul datelor scrise print-o operație de scriere nu depășește limita maximă precizată de constanta simbolică *PIPE_BUF*. În caz contrar operațiile de scriere de la mai multe procese pot fi mixate.

Închiderea unui fișier FIFO

Închiderea fișierului FIFO se face cu ajutorul funcției de sistem *close*, folosită ca pentru orice fișier obișnuit. Închiderea unui fișier FIFO implică sfârșit de fișier pentru procesul care citește din fișierul FIFO.

Ștergerea unui fișier FIFO

Spre deosebire de fișierele pipe, ștergerea fișierelor FIFO se realizează prin apelul funcției *unlink*. Fișierul se consideră șters când numărul său de legături devine zero.

Comunicarea între procese prin fișiere FIFO se face astfel:

1. Un proces crează fișierul FIFO, precizând numele său simbolic, apelând funcția de sistem *mknod* sau *mkfifo*.
2. Procesul care comunică date altora deschide fișierul cu funcția de sistem *open* și scrie datele respective cu funcția *write*.
3. Procesul care primește date deschide fișierul FIFO în citire cu funcția de sistem *open* și apoi citește datele din el cu funcția *read*.

Observație

Comunicarea se face într-un sens sau în ambele sensuri!