

PIPE E FIFO

Pipe

Possono essere intesi come file sequenziali e sono condivisi tra processi per la comunicazione.

La comunicazione avviene in modo unidirezionale, pipe in inglese significa tubo.

Se delle informazioni sono lette dalla PIPE esse spariscono, a meno che non vengano poi riscritte.

Le PIPE a livello di OS non sono altro che buffer di dimensione solitamente intorno ai 4kb.

Le PIPE vengono utilizzate nella comunicazione di processi relazionati, cioè conseguenti ad una fork, altrimenti si usano le FIFO (named PIPE).

Per scriverci si usano i descrittori a mo' di semplici file.

La creazione avviene attraverso la syscall `pipe(int fd[2])`, dove `fd` è un array di due interi, all'indice 0 avremo il descrittore di lettura, all'1 quello di scrittura. Pipe ritorna -1 in caso di fallimento altrimenti 0.

Per scrivere su Pipe si usano semplicemente la `read` e la `write` a cui vengono passati i descrittori aperti con `pipe()`;

Le PIPE sono dispositivi logici e pertanto la "chiusura del file" per convenzione avviene quando tutti coloro che condividevano il descrittore di scrittura `fd[1]` lo hanno chiuso. Così facendo la `read` restituirà 0, notifica del fatto che tutti gli scrittori hanno terminato il loro lavoro. Allo stesso modo uno scrittore che tenti di scrivere su una PIPE quando tutte le copie di `fd[0]` sono state chiuse riceverà un segnale `SIGPIPE` (broken PIPE).

Per fare in modo che tutto funzioni e non si verifichino deadlock, ogni processo DEVE chiudere il descrittore che non gli compete assicurando quindi una modularità e una correttezza di debug da parte della `read` e della `write`, altrimenti si andrebbe come detto in deadlock, ossia un lettore o uno scrittore aspetterebbe invano una risorsa da un partner che semplicemente non c'è.

Fifo

Come detto le FIFO si utilizzano per comunicazione tra processi non relazionati. Una FIFO (file speciale) viene creato con `mkfifo()` passando il nome e la modalità di creazione, ritorna -1 in caso di fallimento, altrimenti 0. RICORDA di fare `unlink()` per eliminare presenza della FIFO nell'OS.

Una volta creata va aperta con la `open()` specificando se in lettura o scrittura. L'apertura però è bloccante, se la si tenta di aprire in lettura si rimane bloccati finché qualcuno non la apre in scrittura. Utilizzando `O_NONBLOCK` questo comportamento viene inibito. Scrivere su una FIFO che non ha lettori si riceve `SIGPIPE` dall'OS.