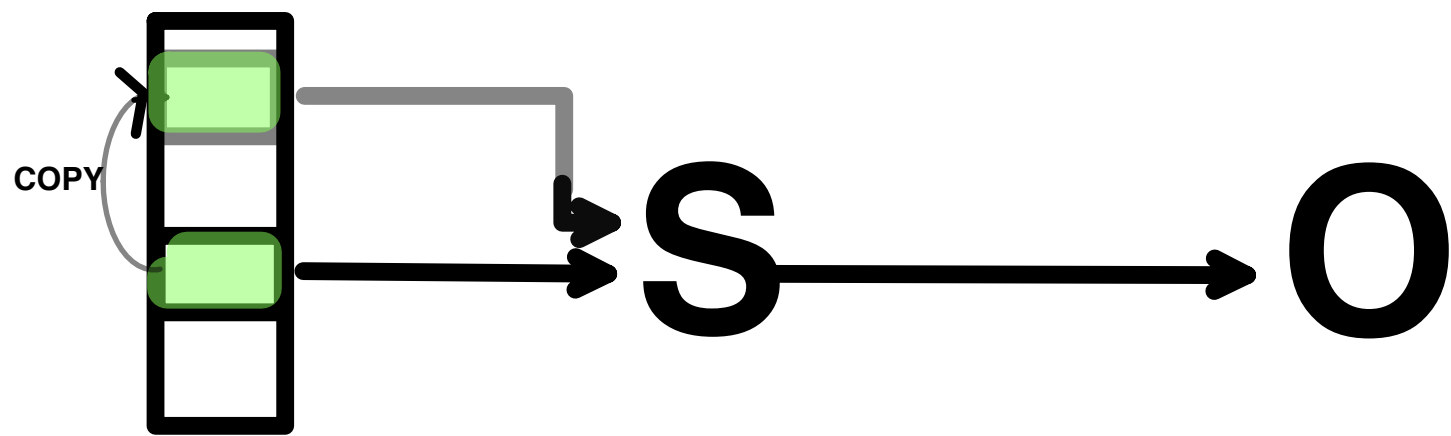


Duplicazione di descrittori e redirezione

Quando noi lavoriamo con i canali di I/O su un VFS, supponiamo di avere una tabella dei descrittori di un certo processo attivo, e una entry di questa tabella ci porta verso una sessione che ci porta verso uno specifico oggetto di I/O O. Quindi ci porta sull'i-node associato a questo oggetto. È possibile copiare il valore che è presente in questa entry in un'altra entry? E quindi avere anche un'altra entry che ci porta sulla stessa sessione?



Questa operazione si chiama "DUPLICAZIONE DI CANALE".
Questa operazione si può fare e quello che viene proposto dai VFS UNIX è l'API dup().

int dup(int descriptor)

Descrizione invoca la duplicazione di un descrittore

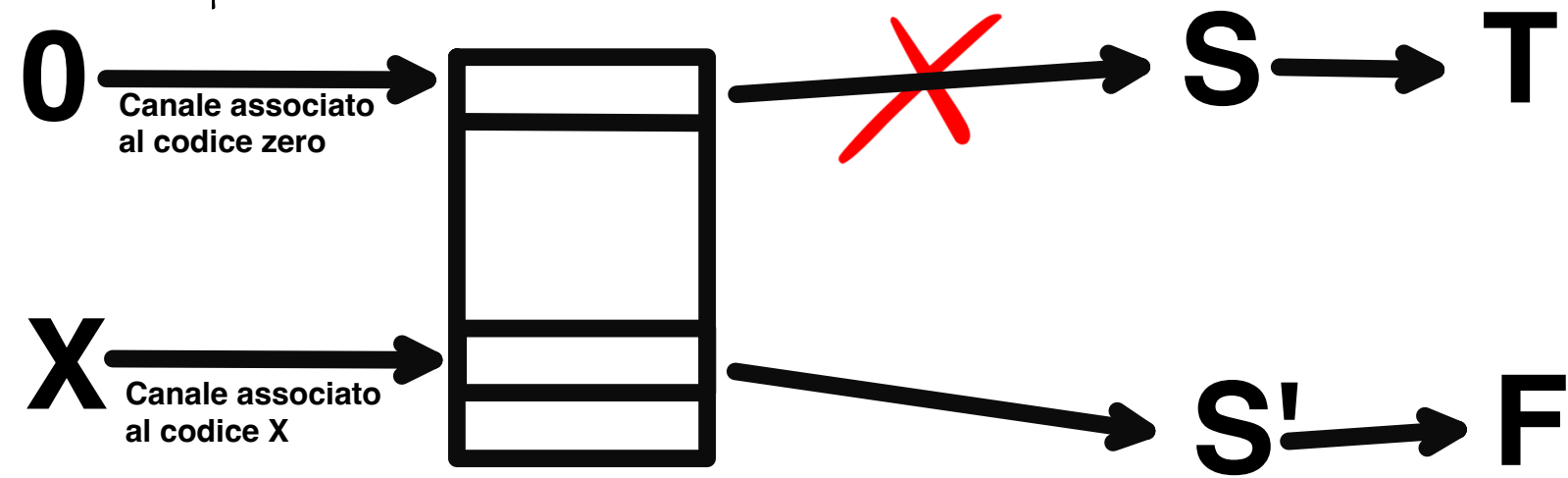
Argomenti descriptor: descrittore da duplicare

Restituzione -1 in caso di fallimento, altrimenti un nuovo descrittore

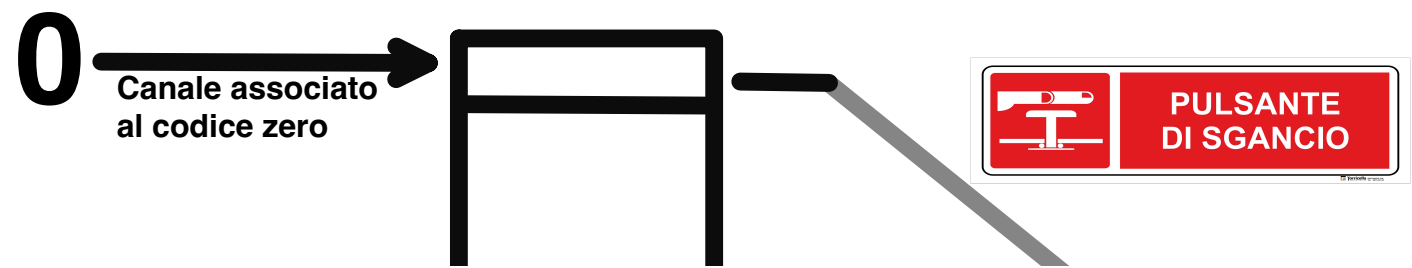
Come parametro passiamo il codice numerico del canale che vogliamo "duplicare": se la duplicazione avviene significa che noi abbiamo due canali per arrivare sulla stessa identica sessione. Questo servizio ci ritorna un intero, che è esattamente il canale duplicato.

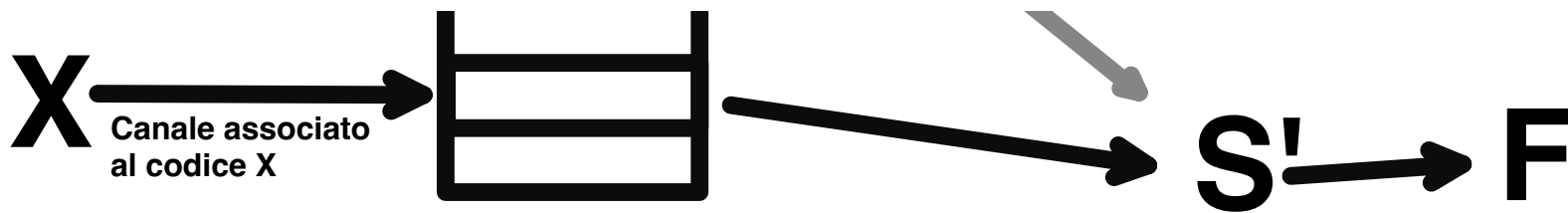
Il descrittore restituito è il primo libero nella tabella dei descrittori del processo (a partire dall'inizio della tabella)

Supponiamo di avere una situazione di questo tipo:
il canale associato al codice zero ci porta su una sessione dove noi andiamo ad interagire con una tastiera.
Mentre invece X ci porta verso un file F.



Possiamo chiudere il canale con codice pari a 0? In modo tale da eliminare il collegamento ed avere la prima entry di nuovo "free"?
Chiaramente Sì, possiamo usare close(0).
Però se dopo chiamo dup(X)? La duplicazione del canale X nella entry zerosima, perché è libera a questo punto. A quel punto la entry zerosima dirotterà: non ci porterà più sulla tastiera, ma sul file:





Quindi se noi andiamo ad usare il canale con codice numerico 0, ossia lo standard input, in realtà non stiamo più lavorando sulla tastiera, ma stiamo lavorando sul contenuto del file. Questo si riassume al simbolo **>** dei comandi linux su shell.

In particolare:

`./a.out > file.txt`

Significa che i dati che noi stiamo producendo sul canale standard di output non devono più essere trasmessi a ciò che era collegato allo stdout, ossia il terminale, bensì devono andare sul file. Abbiamo ridirezionato.

La shell figlio apre il file **f**, chiude lo standard output, e duplica la entry.

Lo fa la shell internamente (shell figlio per il discorso che si duplica).

Quindi quando chiameremo `printf()` l'output andrà su file!

Oppure l'output può essere ridirezionato verso una pipe:

`./a.out | ./file`

Il processo **P** ha il canale di standard pari ad 1, associato ad una PIPE, passando ovviamente tramite la sessione ovviamente, ma non è stato rappresentato per semplicità.



Poi abbiamo un programma **P'** che ha lo standard input che è collegato anche lui a questa PIPE.

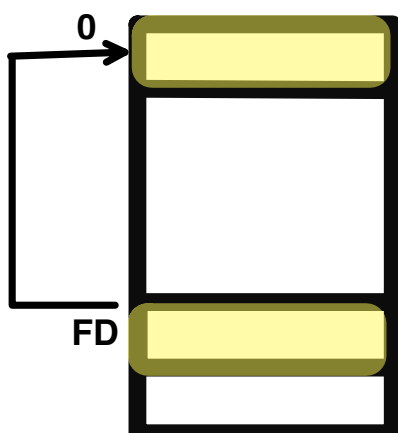


P utilizzando `printf` scrive, i dati vanno sulla PIPE e l'altro processo utilizzando `scanf` legge i dati prodotti da **P**. **P'** legge i dati dalla PIPE, i dati che ha scritto **P**.

```
#include ...
#define FNAME "info.txt"
#define STDIN 0

int main() {
    int fd;
    fd = open(FNAME,O_RDONLY); /* Apro il file in lettura */
    close(STDIN);             /* Chiudo lo standard input */
    dup(fd);                  /* Duplico il descrittore di file*/
    execlp("more","more",0); /* Eseguo 'more' con input ridirezionato */
}
```

La duplicazione di FD avviene nella prima entry, dato che ho chiuso STDIN, ossia il canale 0.



Poi viene lanciato `more`, che non è altro che un comando che legge i dati da standard input da canale 0, e riscriverli sul canale 1. Stiamo visualizzando il contenuto di `info.txt`.

La sessione ci porta a lavorare sul file. Tutte e due le entry ci permettono di lavorare sul file.

Più recenti standard

```
#include <unistd.h>
```

File Descriptor Table

```
int dup(int oldfd)
int dup2(int oldfd, int newfd)
```



Specifica esplicita della posizione della
“file-descriptor table” ove duplicare il canale originale