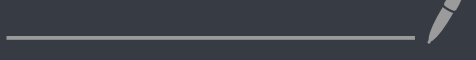

esercitazione 3

Quando un file viene messo in esecuzione, possiede almeno 3 file descriptor

0 : Standard Input

1 : Standard Output

2 : Standard error

alcune funzioni su file:

Write()

```
#include <unistd.h>
ssize write (int fd, const void *buf, size_t count);
```

i primi count byte di buf vengono scritti nel file che è stato associato al file descriptor fd. La chiamata restituisce il numero dei byte scritti oppure -1 se si è verificato un errore (in questo caso si può controllare la variabile errno)

Read()

```
#include <unistd.h>
ssize_t read (int fd, void *buf, size_t count);
```

I primi count byte di buf vengono letti dal file che è stato associato al file descriptor fd. La chiamata restituisce il numero di byte che sono stati letti.

open() → apre un file e restituisce un intero che ne rappresenta il file descriptor.

creat() → serve per creare un file specificando come deve chiamarsi e come deve essere aperto.

(le 2 chiamate sono equivalenti ma con flag differenti:)

close() → serve a chiudere un file specificando il file descriptor.

lseek() → imposta la posizione corrente di lettura e scrittura.

fstat() → restituisce informazioni sul file

dup(), dup2() → creano una copia del descrittore.
Il nuovo descrittore nel caso della dup(), è il descrittore non utilizzato con l'ordine più basso
Nel caso della dup2(), è il newfd dichiarato dall'utente

`link()` → Crea una hard link al file il cui nome è passato in ingresso

`unlink()` → rimuove un hard link