

Library functions vs System Calls

Una introduzione minima

Library functions

Finora abbiamo sempre usato library functions: `fprintf`, `fread`, `malloc`, `strdup`, etc.

- Fanno parte della libreria C
- Forniscono un meccanismo “user friendly” per interagire con il sistema operativo (esempio `fopen`)
- oppure forniscono comode funzionalità (esempio: `strlen` non interagisce con il sistema operativo, ma è comoda)

System calls (per noi)

- Funzioni nelle librerie C che offrono funzionalità “simili” alle Library functions
- Più difficili da usare, ma offrono un maggiore controllo
- Perché esistono, perché questo nome?

Notazione

Nel seguito del corso quando scriviamo ad esempio

`fread(3)`

indichiamo che ci riferiamo alla funzione `fread` nella sezione **3** del comando `man`

Le sezioni del manuale: man man

The table below shows the section numbers of the manual followed by the types of pages they contain.

- 1 Executable programs or shell commands
- 2 System calls (functions provided by the kernel)
- 3 Library calls (functions within program libraries)
- 4 Special files (usually found in /dev)
- 5 File formats and conventions eg /etc/passwd
- 6 Games
- 7 Miscellaneous (including macro packages and conventions), e.g. **man(7)**, **groff(7)**
- 8 System administration commands (usually only for root)
- 9 Kernel routines [Non standard]

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

read – read a line from standard input

SYNOPSIS

read [-r] var...

NAME

read - read from a file descriptor

SYNOPSIS

#include <unistd.h>

ssize_t read(int fd, void *buf, size_t count);

Terminologia

Il Kernel fornisce le proprie funzionalità agli utenti attraverso delle funzioni che chiameremo **kernel system call**

Le **kernel system call** sono indipendenti dal C, ad esempio le possiamo chiamare dall'assembler ARM con **svc**

In C le **system call** sono funzioni di libreria che hanno una corrispondenza 1-1 con le **kernel system call**

Riassumendo

- Le system call sono l'interfaccia con il kernel
- Ognuna ha un singolo compito “semplice”
- Le library functions sono più sofisticate ma sono comunque basate sulle system call
- Vediamo qualche esempio....

Lettura e scrittura di file

Library functions:

`fopen, fprintf, fread, fwrite`

System calls:

`open, read, write`

Vediamo le differenze facendo riferimento alle pagine man

Gestione della memoria

Library functions:

`malloc, free, realloc, calloc`

System calls:

`brk, sbrk`

Vediamo le differenze facendo riferimento alle pagine man

Perché studiamo le system call?

Perché alcune funzionalità le richiedono, ad esempio le pipe sono basati sui **file descriptor** quindi per leggere e scrivere si deve usare `read/write`

In tutti gli altri casi, usatele solo se sapete cosa state facendo...