

Dispositivi di I-O

Index

- [Categorie di dispositivi I/O](#)
 - [Dispositivi leggibili dall'utente](#)
 - [Dispositivi leggibili dalla macchina](#)
 - [Dispositivi per la comunicazione](#)
 - [Funzionamento \(semplificato\) dei dispositivi I/O](#)
 - [Differenze tra dispositivi di I/O](#)
 - [Data rate](#)
 - [Applicazioni](#)
 - [Complessità del controllo](#)
 - [Unità di trasferimento dati](#)
 - [Rappresentazione dei dati](#)
 - [Condizioni di errore](#)
-

Categorie di dispositivi I/O

L'I/O è un'area assai problematica nella progettazione di sistemi operativi in quanto ci sta una grande varietà di dispositivi I/O e anche una grande varietà di applicazioni che li usano, risulta quindi difficile scrivere un SO che tanga conto di tutto

Esistono tre categorie di dispositivi:

- leggibili dall'utente
- leggibili dalla macchina
- dispositivi di comunicazione

Dispositivi leggibili dall'utente

Sono tutti quei dispositivi usati per la comunicazione diretta con l'utente, come ad esempio:

- stampanti

- “terminali” → monitor, tastiera, mouse, ecc. (da non confondere con i terminali software)
- joystick
- ecc.

Dispositivi leggibili dalla macchina

Sono quei dispositivi usati per la comunicazione con materiale elettronico, come ad esempio:

- dischi
- chiavi USB
- snesopri
- controllori
- attuatori

Dispositivi per la comunicazione

Dispositivi usati per la comunicazione con dispositivi remoti, come ad esempio:

- modem
- schede Ethernet
- Wi-Fi

Funzionamento (semplificato) dei dispositivi I/O

Un dispositivo di *input* prevede di essere **interrogato sul valore di una certa grandezza fisica al suo interno** (eg. per il mouse si tratta delle coordinate dell'ultimo spostamento effettuato, per la tastiera il codice Unicode dei tasti premuti, per il disco il valore dei bit che si trovano in una certa posizione al suo interno).

L'idea è che se un processo effettua una syscall `read` su un dispositivo del genere, vuole conoscere questo dato (per poterlo elaborare e fare altro)

Un dispositivo di *output* prevede di poter **cambiare il valore di una certa grandezza fisica al suo interno** (eg. per un monitor il valore RGB dei pixel, per una stampante il PDF di un file da stampare, per il disco il valore dei bit che devono sovrascrivere quelli che si trovano in una certa posizione al suo interno)

L'idea è che se un processo effettua una syscall `write` su un dispositivo del genere, vuole cambiare qualcosa (spesso l'effetto è direttamente visibile all'utente, ma in altri casi è visibile solo usando altre funzionalità di lettura, come per il disco)ù

Ci sono quindi almeno due syscall `read` e `write` (tra i dati che prendono in input queste syscall ci sta un identificativo del dispositivo su cui si vuole eseguire l'operazione, in Linux sono i *file descriptor*)

Dunque quando arriva una di queste due syscall, il kernel comanda l'inizializzazione del trasferimento di informazioni, mette il processo in blocked e passa ad altro; la parte del kernel che gestisce un particolare dispositivo I/O è chiamata *driver* e spesso il trasferimento è fatto con DMA (viene fatto direttamente tra la RAM e il dispositivo) A trasferimento completato, arriva l'interrupt, si termina l'operazione e il processo ritorna ready.

Warning

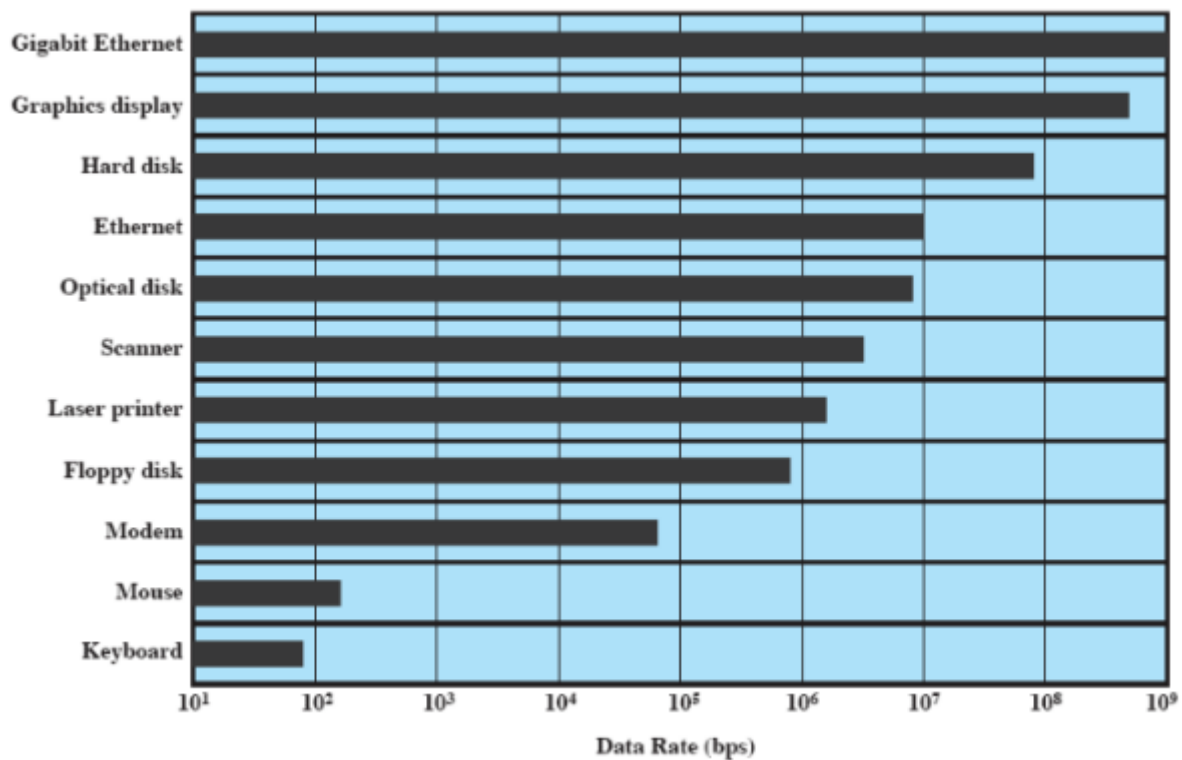
Questa normale esecuzione delle operazioni però potrebbe variare ad esempio in caso di fallimento dell'operazione (bad block su disco...) oppure potrebbe essere necessario fare ulteriori trasferimenti, per esempio dalla RAM dedicata al DMA a quella del processo

Differenze tra dispositivi di I/O

I dispositivi di I/O possono differire sotto molti aspetti:

- data rate (frequenza di accettazione/emissione dati)
- applicazione
- difficoltà nel controllo (tastiera vs. disco...)
- unità di trasferimento dati (caratteri vs. blocchi)
- rappresentazione dei dati
- condizioni di errore

Data rate



Applicazioni

Ogni dispositivo di I/O ha una diversa applicazione ed uso

I dischi sono usati per memorizzare files, a tale scopo richiedono un software per poterli gestire. I dischi però sono anche usati per la memoria virtuale, per la quale serve un altro software apposito (nonché hardware).

Un terminale usato da un amministratore di sistema dovrebbe avere una priorità più alta

Complessità del controllo

Per quanto riguarda la quantità di controlli necessari per i dispositivi I/O, una tastiera o un mouse richiedono un controllo molto semplice.

Ad esempio per quanto riguarda una stampante, questa è più articolata; richiedono infatti di ricevere un PDF o un PS (poi la traduzione da PDF ad azioni della stampante è ben più complessa)

Il disco è tra le cose più complicate da controllare, fortunatamente non viene fatto tutto da software, e molte cose vengono controllate da hardware dedicato

Unità di trasferimento dati

Per trasferire i dati dal dispositivo che li genera alla memoria e viceversa ci sono due possibilità:

- trasferirli in **blocchi di byte** di lunghezza fissa → usato per la memoria secondaria (dischi, chiavi USB, CD/DVD, ...)
- trasferirli come **flusso (*stream*) di byte** (o caratteri) → qualsiasi cosa che non sia memoria secondaria (terminali, stampanti, schede audio, dispositivi di rete...)

Rappresentazione dei dati

I dati sono rappresentati secondo codifiche diverse a seconda del dispositivo che li genera/accetta (una vecchia tastiera potrebbe usare ASCII puro, mentre una moderna l'UNICODE)

Possono essere diversi anche i controlli di parità

Condizioni di errore

Gli errori possono essere riparabili o non, e la loro natura varia di molto a seconda del dispositivo. Possono cambiare: nel modo in cui vengono notificati, sulle loro conseguenze (fatali, ignorabili), come possono essere gestiti