

HARDWARE I/O > 12.2

La comunicazione con l'I/O avviene tramite i controller.

In particolare vengono impartiti segnali di controllo e modificati registri dei dati del controller.

I metodi principali di interazione con l'I/O controller

• Direct I/O Instructions:

Si usano istruzioni per trasferire word/byte nelle I/O ports.

• I/O Memory Mapped: (12.2.1)

I registri di controllo dell'I/O sono mappati in memoria consentendo di usare le stesse operazioni di lettura/scritt.

indirizzi per I/O (in esadecimale)	dispositivo
000-00F	controllore DMA
020-021	controllore delle interruzioni
040-043	timer
200-20F	controllore dei giochi
2F8-2FF	porta seriale (secondaria)
320-32F	controllore del disco
378-37F	porta parallela
3D0-3DF	controllore della grafica
3F0-3F7	controllore dell'unità a dischetti
3F8-3FF	porta seriale (principale)

Figura 12.2 Indirizzi delle porte dei dispositivi di I/O nel pc (elenco parziale).

Gli I/O dispongono dei seguenti registri:

- data_in:
- status: Status I/O e istruzioni.
- data_out:
- control: Per mandare istruzioni.

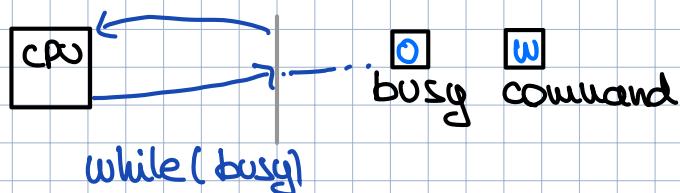
INTERAZIONI >

Le interazioni fra I/O e sistema sono gestite con

POLLING: CPU aspetta evento I/O (12.2.2)

La CPU "aspetta" l'I/O tramite l'attesa attiva.

Ciò è realizzabile tramite un meccanismo di handshake che consente la sincronizzazione delle operazioni:



La CPU aspetta con letture continue. Quando pronto, il controller lascia passare.

Può essere onerosa ex over cons.

Interrupt: L'I/O notifica la CPU (12.2.3)

«Guardare LO!» Nei sistemi con ≥ 1 CPU è

Presente un meccanismo di scheduling per l'assign.

DMA: I/O scrive direttamente su memoria (12.2.4)

Si usa un DMA Controller [HW] per consentire una comunicazione diretta senza CPU.

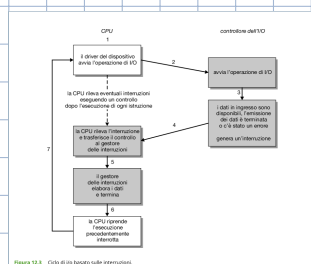
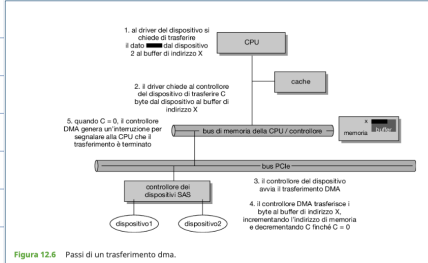


Figura 12.3 Ciclo di interruzione.

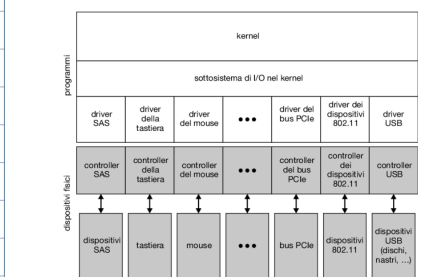
La CPU comunica l'istruzione da eseguire al DMA Controller, che andrà a scrivere / leggere / Trasferire dati in maniera indirip. Interrupt
Per poi segnalare il completamento



APPLICATION I/O INTERFACE > 12.3

Il S.O. standardizza l'interazione fra I/O e processi. Questo avviene con vari layer:

- Driver: Astrae la logica di gestione dei vari controller



I devices a loro volta possono essere classificati in:

- Block / Char. Flux I/O : tr. 1 blocco di byte / 1 byte
- Seq. / Direct Access: Accesso seq. / Accesso a qualunque loc.
- Sync / Async: Response Time preved. / Imprev.
- Condivisibili / Dedicati: Ai thread.
- Per Veloc. • Lettura / Scrittura

Molti SO hanno il comando `ioctl()`; per mandare comandi.

ISTRUZIONI BLOCK / CHAR I/O 12.3.1

- `get()`: Lettura buffer
- `put()`: Scrittura buffer
- `write()`: Prox Block.
- `read()`:

Char I/O

Block

ISTRUZIONI PER NETWORK DEVICES 12.3.2

- `select()`: System call usata dai processi per mettersi in attesa di risposte eventuali dai socket.

CLOCK / TIMER 12.3.3

Consentono di segnare l'ora corrente, il tempo trascorso, timer.

TIPOLOGIA DI I/O SYS CALLS > 12.3.4

BLOCKING: Sospeso fino a fine I/O (Intero processo)

NON BLOCKING: "Restituisce quello che è riuscito a fare"

Consente di andare avanti nell'esecuzione delle istruzioni.

Necessita un meccanismo di "wake-up" per reattive quando

arriva la risposta. Ciò è implementabile con:

- Multi Thread
- Select
- Sospensione

ASync :

Simili a non-bloccanti ma torna immediatamente senza fare niente.

VECTORED I/O > 12.3.5

Consente l'esecuzione di operazioni multiple su array di I/O.

KERNEL I/O SUBSYSTEM > 12.4

L'ILIO Subsystem mette a disposizione diverse policy.

SCHEDULING I/O 12.4.1

Effettua lo scheduling delle istruzioni imponendo un ordine.

BUFFERING 12.4.2

Si impiega un'area di mem. fra I/O e device.
Questo consente di migliorare le prestazioni.
e diminuire la frequenza di i/o sys calls.

Si possono incrementare le performance del kernel tramite:

- Caching (12.4.3)
- Code di Spooling: Riserva I/O. (12.4.4)
- Bypass SO: Da spazio user a mem. del controller.

FILE SYSTEM INTERFACE > 13.

L'S.O. rappresenta tutti gli elementi come file. Un file è un'unità di memoria logica contigua.

Ogui file header: name, extension, location, checksum, time data.

OPERAZIONI 13.1.2

- create / delete / truncate
- write: *Necessita di un puntatore di scrittura*
- read: *// // // // lettura*
- seek: *Riposiziona il puntatore (se random)*

Le operazioni richiedono la open del file e il S.O.

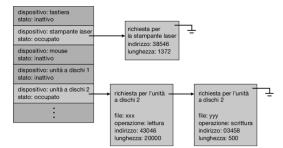


Figura 12.10 Tabella dello stato dei dispositivi.

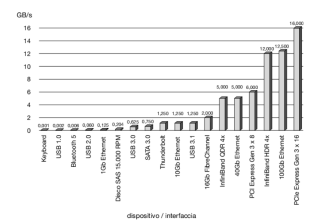


Figura 12.11 Dispositivi di I/O utilizzati in pc e data center e velocità dell'interfaccia.

censisce una tabella dei file aperti (1 vproc e una gen).

LOCKING

- Locked: accesso esclusivo al file
- shared lock: + Processi accedono in contemp. (lettura).

STRUTTURA INTERNA 13.1.5

- Semplice: Seq di word arbitrarie (csv)
- Complesse: JSON, fastq, xml...

ACCESSO 13.2

- Seq: Lettura / Scrittura in sequenza
- Diretto: file organizzato a blocchi in cui si può accedere in maniera diretta.
- Index Based: Si organizzano gli elementi in blocchi raggruppati per indice. (indice + ricerca)