La gestione dell'I/O
(*Cap. 5, Tanenbaum*)

Prestazioni e generalità
Modelli di funzionamento
Gestione software
Supporti su disco
Orologi

Prestazioni e generalità
Misura delle prestazioni

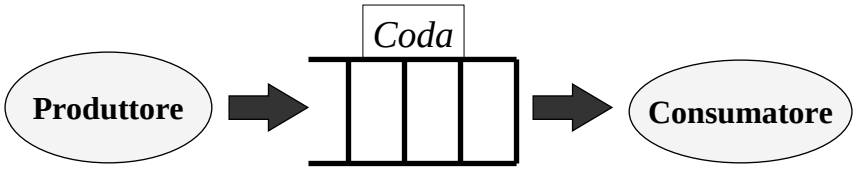
- Le prestazioni di I/O si misurano in termini di
  - **Tempo di risposta** (o *di latenza*)
    - L'intervallo di tempo tra l'emissione di una richiesta di I/O ed il suo soddisfacimento
  - **Banda passante** (o *throughput*)
    - Il numero medio di operazioni di I/O completabili per unità di tempo (equivalente alla quantità totale di dati trasmessi)

Prestazioni e generalità
I/O come collo di bottiglia

- **CPU**: la velocità raddoppia ogni 18 mesi
- **DRAM**: la capacità raddoppia ogni 2-3 anni
- **Disco rigido**: la capacità raddoppia ogni 3 anni
- **I/O**: le prestazioni migliorano del 10% all'anno
  - Limiti derivanti da seri problemi *meccanici*
- Le prestazioni generali degli elaboratori aumentano poco per via del *collo di bottiglia* causato dall'I/O

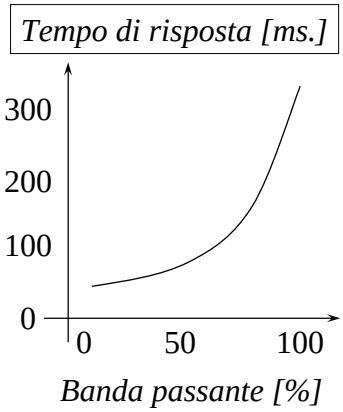
Modelli di funzionamento
Modello di coda - 1

Un sistema di I/O può essere rappresentato con un modello produttore-consumatore dotato di una coda dei prodotti per compensare le diverse velocità relative



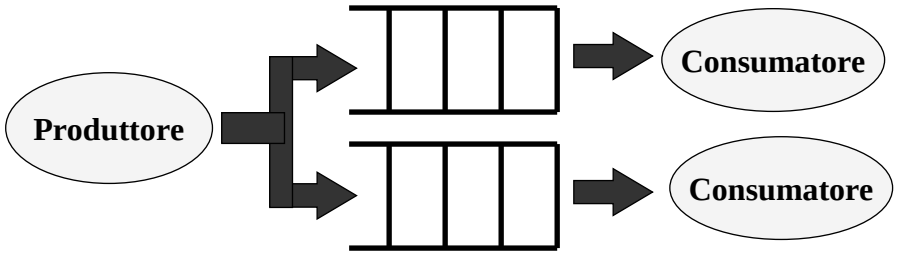
Modelli di funzionamento
Modello di coda - 2

- Si ha basso tempo di risposta con coda *vuota* e consumatore *libero*
- Si ha elevata banda passante con coda *mai vuota* e consumatore *mai inoperoso*



Modelli di funzionamento
Modello di coda - 3

- Il tempo di risposta non è facilmente migliorabile
- La banda passante può essere aumentata moltiplicando i produttori e/o i consumatori



Modelli di funzionamento
Classificazione dei dispositivi di I/O

- **A blocchi**
  - Il dispositivo emette e riceve dati strutturati in *blocchi* di dimensione fissa (512B-32kB), ciascuno con un indirizzo proprio
    - Disco, nastro
  - I blocchi possono essere indirizzati (per lettura o scrittura) *indipendentemente*
- **A caratteri**
  - Il dispositivo emette e riceve sequenze *non strutturate* di caratteri
    - Stampante, interfaccia di rete, *mouse*
  - Non consente indirizzamento dei dati

Modelli di funzionamento
Struttura dei dispositivi di I/O - 1

- **Parte meccanica**
  - Il dispositivo vero e proprio
- **Parte elettronica**
  - Chiamata *device controller* (o adattatore)
  - Un singolo *controller* può gestire più dispositivi
- **L'interfaccia** tra dispositivo e *controller* è spesso di basso livello (primitivo)
  - La funzione del *controller* è precisamente quella di offrire al S/O un' interfaccia meno rozza verso il dispositivo

Modelli di funzionamento
Struttura dei dispositivi di I/O - 2

- I *controller* utilizzano registri e banchi di memoria per comunicare con la CPU
  - I registri contengono comandi
    - Di configurazione, di controllo, di esecuzione
  - I banchi di memoria (buffer) contengono i dati in emissione o in ricezione
  - Il S/O ha accesso ad entrambi

Modelli di funzionamento
Struttura dei dispositivi di I/O - 3

- Varie tecniche permettono di includere i registri del *controller* nello spazio di indirizzamento della CPU
  - Ciascun registro viene visto come una **porta speciale di I/O** indirizzata da istruzioni speciali
  - Tutti i registri vengono visti e trattati come parte della memoria (***memory-mapped I/O***)
    - La corrispondente area non viene allocata liberamente
  - Soluzioni ibride (Pentium)
    - Banchi di memoria del controller come *memory-mapped I/O*
    - Registri di controllo come porte speciali di I/O

Modelli di funzionamento
Struttura dei dispositivi di I/O - 4

- La tecnica *memory-mapped I/O* ha diversi vantaggi
  - Non necessita di istruzioni speciali
    - Le istruzioni che accedono memoria ‘normale’ accedono anche le aree di I/O
    - Il software di controllo di dispositivo può essere scritto interamente in linguaggi ad alto livello
  - Consente una più agevole protezione
    - è sufficiente nascondere le aree di I/O allo spazio di indirizzamento di utente (*privilegi*)
    - Con la segmentazione, *più* aree di I/O possono mappare sul *medesimo* spazio di indirizzamento fisico

Modelli di funzionamento
Struttura dei dispositivi di I/O - 5

- La tecnica *memory-mapped I/O* presenta anche alcuni svantaggi
  - Non si presta all’uso di cache
    - Il dato rilevante è *sempre e solo* nella memoria del dispositivo
    - Occorre disabilitare selettivamente la cache
  - Non è compatibile con architetture a ‘bus multipli’
    - I dispositivi di I/O non possono rispondere ad indirizzi emessi su bus non connessi
    - Occorre filtrare gli indirizzi emessi dalla CPU ed instradarli sul bus appropriato
      - Filtraggio a sorgente piuttosto che a destinazione

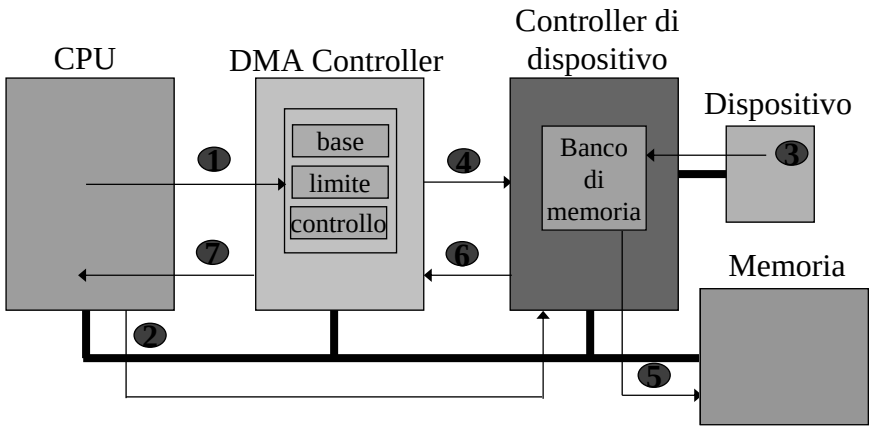
Modelli di funzionamento
Accesso ai dispositivi di I/O - 1

- Tutti i dispositivi si affacciano sul canale indirizzi controllato dalla CPU
- Per ogni indirizzo emesso sul canale, la CPU specifica se per lettura o scrittura
- Il *tipo* di indirizzo emesso individua univocamente I'unità chiamata a rispondere
  - Blocchi diversi di indirizzi sono assegnati ad unità diverse
    - Nessuna sovrapposizione, nessun conflitto

Modelli di funzionamento
Accesso ai dispositivi di I/O - 2

- Quando CPU e dispositivo hanno velocità relative molto diverse non conviene che la più veloce debba attendere quella più lenta
- Il meccanismo detto ***Direct Memory Access*** (**DMA**) consente tale disaccoppiamento
- Questo richiede l'uso di un *DMA controller* capace di prendere occasionalmente possesso dei canali indirizzi e dati

Modelli di funzionamento
Accesso ai dispositivi di I/O - 3



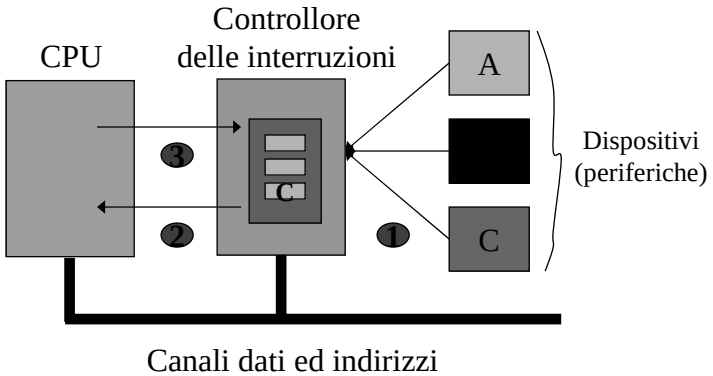
Modelli di funzionamento
Accesso ai dispositivi di I/O - 4

1. La CPU programma il *DMA controller* informandolo della destinazione e dell'ampiezza del trasferimento da effettuare
2. La CPU comanda al *controller* di dispositivo di acquisire dati dal dispositivo
3. Il *controller* di dispositivo salva i dati acquisiti nella sua memoria interna
4. Il *DMA controller* comanda al *controller* di dispositivo di trasferire i dati acquisiti in memoria principale
5. Il *controller* di dispositivo effettua il trasferimento
6. Il *controller* di dispositivo informa il *DMA controller* l'esito del trasferimento
7. Il *DMA controller* informa la CPU dell'esito del trasferimento

Modelli di funzionamento
Accesso ai dispositivi di I/O - 5

- Il *DMA controller* può accedere al canale dati in uno di due modi
  - **Una parola alla volta**, sottraendo di tanto in tanto alla CPU il controllo sul canale (*cycle stealing*)
  - **Per blocchi**, prendendo possesso del canale per una serie di trasferimenti (*burst mode*)
  - La CPU è bloccata in entrambi i casi, ma il *burst mode* è **più efficace** perché l'acquisizione del canale è onerosa
- La CPU può fare a meno del DMA se essa è molto più veloce del *DMA controller*

Gestione software
Il meccanismo delle interruzioni - 1



Gestione software
Il meccanismo delle interruzioni - 2

1. Il dispositivo **C** segnala che ha bisogno di attenzione (p.es. ha completato un trasferimento)
2. Il controllore delle interruzioni segnala alla CPU che il dispositivo **C** chiede attenzione
3. La CPU attiva il servizio di gestione dell'interruzione richiesta e ne notifica l'inizio al controllore delle interruzioni
   - Solo *dopo* tale notifica il controllore delle interruzioni si dispone ad accogliere altre richieste dai dispositivi