Lezione 30

I/O: tecniche di controllo

Memorie di massa

Proff. A. Borghese, F. Pedersini

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

Tecniche di controllo dell'I/O

- Tecniche di controllo I/O
  - A controllo di programma diretto
  - A controllo di programma con polling
  - Ad interruzione
  - Ad accesso diretto alla memoria (DMA)
- Memorie di massa: tecnologie

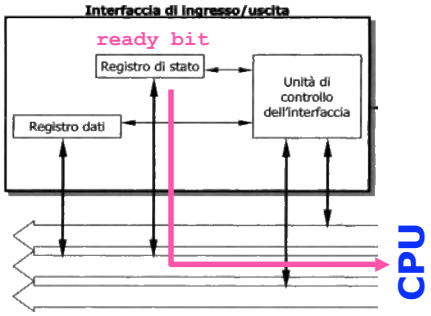
I/O a **controllo di programma**

I/O a controllo di programma:

- La periferica ha un ruolo **passivo.**
  La CPU si occupa sia del controllo, sia del trasferimento dati.
  - La CPU, dopo avere predisposto il controller all'esecuzione dell'I/O, **si ferma** e si mette ad **interrogare il registro di stato** della periferica **in attesa** che il **ready bit** assuma un determinato valore.

- Svantaggio: CPU bloccata in stato di: ***busy waiting*** o ***spin lock***

```
begin
   ... // Predisponi i registri del controller
   ... // ad effettuare una operazione di
lettura
   while (ready-bit == 0) do { }; // spin lock
   ... // Carica il dato acquisito
end;
```



I/O a controllo di programma – costo

- Esempio: tastiera a controllo di programma
  - Tastiera gestita a controllo di programma che opera a **0,01kByte/s**.
  - Frequenza di clock: **50MHz**
  - Determinare il tempo in cui verrebbe effettivamente utilizzata la CPU per trasferire **1 parola**, tenendo conto che ci vogliono **20 cicli di clock per ogni byte**

Tempo di trasferimento:

T = 4 byte/parola / **0,01** kByte/sec **= 0,4** sec/parola

#cicli_clock = $50 \cdot 10^6$ * 0,4 = $\mathbf{2 \cdot 10^7}$ cicli_clock/parola

In realtà utilizza solo:

20·4 = 80 cicli_clock per trasferire la parola

→ sfruttamento CPU: $(80 / 2 \cdot 10^7) = 4 \cdot 10^{-6} = 0{,}0004$ %

- **Prestazioni I/O a controllo di programma:**
  - \+ Risposta molto veloce (bassa latenza)
  - – Alta probabilità di spin lock

Sommario

- Tecniche di controllo I/O
  - A controllo di programma diretto
  - A controllo di programma con polling
  - Ad interruzione (interrupt)
  - Ad accesso diretto alla memoria (DMA)
- Memorie di massa: tecnologie

Polling

- **POLLING**: interrogazione **ciclica** (non bloccante) dello stato di tutte le periferiche
  - Interrogazione del **registro di stato** della periferica
  - In presenza di uno stato di **busy–waiting** su una periferica, si esegue il **polling sulle altre** periferiche di I/O
  - Se la periferica interrogata necessita di intervento:
    - si soddisfa la richiesta
    - quindi si prosegue il ciclo di polling sulle altre periferiche

```
// Leggi dato da perif_x
begin
    Predisponi i registri controller per una read;
b:  if (ready_bit (perif_1) == 1) servi perif_1;
    if (ready_bit (perif_2) == 1) servi perif_2;
        …
    if (ready_bit (perif_n) == 1) servi perif_n;
    goto b;
end;
```

I/O con polling: valutazione prestazioni

- Polling dura più del Controllo da programma, ma non è bloccante!
  1. trasferimento del controllo alla procedura di polling,
  2. accesso alla periferica,
  3. ritorno al programma utente

  *Typ: **400 cicli_clock***

- **Prestazioni I/O a polling:**
  - \+ Risolve il problema dello spin lock
  - – Risposta meno veloce che nel controllo di programma (maggiore latenza)

- Valutazione efficienza del polling: *data una CPU con clock f_C=500 MHz, determinare l'impatto del polling per 3 dispositivi diversi:*
  - Mouse: deve essere interrogato almeno 30 volte al secondo per non perdere alcun movimento dell'utente.
  - Floppy disk: trasferisce dati al processore in parole da 16 bit ad una velocità di 50 kByte/s
  - Hard disk: trasferisce dati al processore in blocchi di 4 parole alla velocità di 4 MByte/s

Efficienza I/O mediante Polling

- Efficienza: % utilizzo della CPU per gestire il trasferimento I/O
  - Grazie al polling non bloccante, nel tempo restante la CPU può fare altro
- Mouse:
  - 30 x 400 = 12,000 cicli_clock/sec
  - 12,000 / 500,000,000 = 0,000024 sec/sec → 0,0024%
  - *Piccolo impatto sulle prestazioni*
- Floppy disk:
  - Per ogni accesso possiamo trasferire **2 Byte**
  - A 50 kB/sec → occorrono **25 k accessi/sec**
  - In termini di cicli di clock: 25k x 400 = **10 Mcicli_clock/sec**
  - (ignorando discrepanza tra 25k e 25.000) 10 M / 500 M → 2%
  - *Medio impatto sulle prestazioni*
- Hard disk:
  - Per ogni accesso possiamo trasferire **16 Byte**
  - a 4 MB/sec → occorrono **250 k accessi/sec**
  - In termini di cicli di clock: 250k x 400 = **100 Mcicli_clock/s**
  - (ignorando differenza tra 250k e 250.000) 100M/500M → 20%
  - *Alto impatto sulle prestazioni*

Polling e Controllo diretto: limiti

- Limiti di polling e controllo di programma diretto:
  - **Controllo diretto**: alta probabilità di **spin lock**
  - **Polling**: durata talvolta notevole della procedura di gestione (overhead)

In entrambi i casi...

- con periferiche lente:
  - eccessivo spreco del tempo di CPU, che per la maggior parte del tempo rimane occupata nel ciclo di **busy waiting / polling**
- con periferiche veloci:
  - il lavoro svolto dalla CPU è quasi interamente dovuto all'effettivo trasferimento dei dati

Sommario

- Tecniche di controllo I/O
  - A controllo di programma diretto
  - A controllo di programma con polling
  - Ad interruzione (interrupt)
  - Ad accesso diretto alla memoria (DMA)
- Memorie di massa: tecnologie

Interrupt

La periferica segnala alla CPU di avere bisogno di attenzione mediante un apposito segnale (bus di controllo)

→ **`interrupt:`** interrompe il normale funzionamento del processore

- Funzionamento:
  - La periferica che necessita di attenzione avvisa la CPU, generando un segnale **`interrupt request`**
  - Quando il processore "se ne accorge" (fase di fetch), lo riconosce producendo un segnale di **`interrupt acknowledge`**
  - LA CPU abbandona il programma corrente e passa ad eseguire la procedura di risposta all'interrupt
  - Il programma utente deve in seguito poter procedere dal punto in cui è stato interrotto
    → **Salvataggio del contesto**

Esempio: comando *print* con periferica *busy*

1. Invio del comando: **`print`**
2. Se la periferica è in stato **`busy`**, CPU torna alla sua attività, scaricando sul registro di controllo la richiesta di output
3. Quando la periferica diventa **`ready`**, viene inviato un interrupt → **`interrupt request`**
4. Il programma di risposta all'interrupt, provvederà a trasferire alla periferica il dato che su vuole stampare
5. Terminato di "servire" l'interrupt, riprende l'esecuzione del programma originario, dal punto di interruzione
   → necessità di salvataggio del contesto del programma originario

I/O ad interrupt - Valutazione efficienza

- Esempio: Hard disk
  - Frequenza di clock è **500Mhz**
  - Trasferimento di blocchi di **4 parole** (1 word alla volta)
  - Trasferimento a **4 Mbyte/s**
  - Il disco sta trasferendo dati solamente per il **5%** del suo tempo
  - Il costo di ogni interruzione è **500 cicli di clock**
- Trasferimento:
  4 MB = 1Mword/s → 250 k interrupts/sec
- Costo dell'interruzione:
  *250 k int/s * 500 cicli_clock = 125 Mcicli_clock/sec*
- Frazione di utilizzo del processore per il trasferimento (interrupt):
  125 M / 500 M = 25 %
- Frazione del processore utilizzata in realtà:
  *125M / 500M * 0.05 = 1,25 %*

Interrupt **gerarchici**

- Accodamento degli interrupt → **FIFO**
- Annidamento degli interrupt → **LIFO** (cf. stack)

Occorre definire una priorità !

- Maschere di interrupt
  - Ad ogni interrupt viene associato un livello
  - Il **livello corrente** è associato allo stato del sistema
  - Interrupts **allo stesso livello** sono accodati **FIFO**
  - La maschera dell'interrupt viene memorizzata nello **status register** (registro di stato del sistema – 8 livelli in MIPS)
  - La maschera degli interrupt pending viene caricata nel cause register. Per ogni bit a 1, c'è un interrupt pending per quel livello
  - Codifica di priorità

Sommario

- Tecniche di controllo I/O
  - A controllo di programma diretto
  - A controllo di programma con polling
  - Ad interruzione (interrupt)
  - Ad accesso diretto alla memoria (DMA)
- Memorie di massa: tecnologie

DMA: Direct Memory Access

- Limiti della gestione interrupt-driven:
  - La gestione mediante interrupt non svincola la CPU dal dovere eseguire le operazioni di trasferimento
- Per periferiche veloci, le operazioni di trasferimento dati occupano un tempo preponderante rispetto al tempo speso in spin-lock
- Per evitare l'intervento della CPU nella fase di trasferimento dati, è stato introdotto il protocollo di trasferimento: Direct Memory Access (DMA)

Caratteristiche della DMA

- DMA controller: processore specializzato nel trasferimento dati tra dispositivi di I/O e memoria centrale
- Per attivare il trasferimento, viene la CPU spedisce al DMA controller le seguenti informazioni:
  - il tipo di operazione richiesta
  - l'indirizzo da cui iniziare a leggere/scrivere i dati
  - il numero di bytes riservati in memoria centrale
- Il DMA controller gestisce il trasferimento dei dati: la CPU si svincola completamente dall'operazione di I/O
- Dopo avere trasferito tutti i dati, il DMA controller invia un `interrupt request` alla CPU per segnalare il completamento del trasferimento.

I/O mediante DMA – efficienza

- Esempio: Hard disk
  - Frequenza di clock è 500 Mhz
  - Trasferimento di blocchi di 8 kbyte per ogni DMA
  - Trasferimento a 4 Mbyte/s
  - Il costo dell'inizializzazione del DMA è di 1000 cicli di clock
  - Il costo dell' interruzione al termine del DMA è di 500 cicli di clock
- Per ciascun trasferimento DMA occorre:
  - 1000 + 500 cicli di clock (tempo di inizio + tempo di fine)
  - Numero di DMA/sec: 4 Mbyte/s / 8 kbyte = 500 DMA/sec
  - Numero cicli_clock CPU richiesti: 1500*500 = 750 kCC/sec
  - Frazione del processore utilizzata: 750k / 500M = 0,2 %

Sommario

- Tecniche di controllo I/O
  - A controllo di programma diretto
  - A controllo di programma con polling
  - Ad interruzione (interrupt)
  - Ad accesso diretto alla memoria (DMA)
- Memorie di massa: tecnologie

Memorie di massa: tecnologie

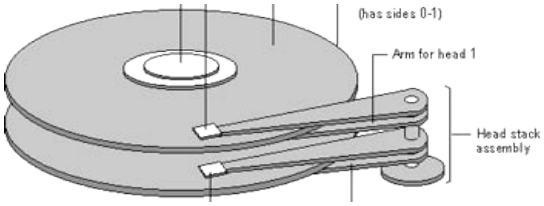
Memorie di massa:
memorie atte a memorizzare dati in modo non volatile

Tecnologie:

- Memorizzazione MAGNETICA
  - I dati sono letti/scritti mediante una testina magnetica
  - DISCHI (hard disk, floppy disk)
  - NASTRI
- Memorizzazione OTTICA
  - CD–ROM
  - DVD
- Memorizzazione a SEMICONDUTTORE
  - FLASH memories

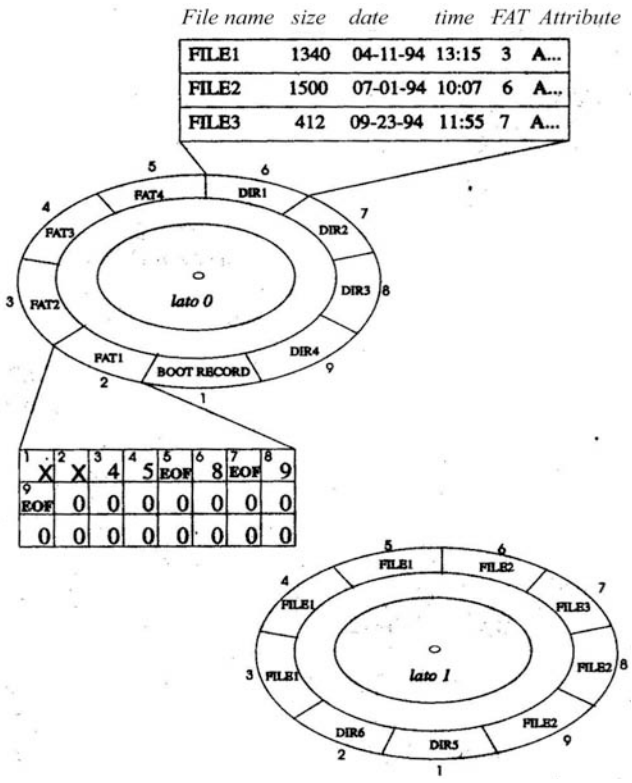
Hard disk – disco rigido

- Costituiti da piatti rotanti (da 1 fino a 25) ognuno con due facce
  - Esiste una testina per ogni faccia
  - Le testine di facce diverse sono collegate tra loro e si muovono contemporaneamente, in modo solidale tra loro
  - La pila dei piatti viene fatta ruotare alla stessa velocità: (4,200 – 10,000 rpm)
- Ogni faccia è divisa in circonferenze concentriche chiamate tracce
  - N. tracce: 1000 ÷ 5000
- L'insieme delle tracce di ugual posto su piatti diversi è chiamato cilindro
  - Solitamente, ogni traccia contiene la stessa quantità di bit
  - le tracce più esterne memorizzano informazione con densità minore
  - La densità dipende dalla qualità del disco
- Ogni traccia è suddivisa in settori
  - Il settore è la più piccola unità che può essere letta/scritta su disco (512B ÷ 8kB)



Hard disk – struttura

- L'indice del contenuto del disco è solitamente scritto nella traccia 0
  - File Allocation Table (FAT)



Hard disk – lettura / scrittura

- Per leggere/scrivere informazioni sono necessari tre passi:
  - la testina deve essere posizionata sulla traccia corretta;
  - il settore corretto deve passare sotto la testina;
  - i dati devono essere letti/scritti

- **Tempo di seek (ricerca)**
  - tempo per muovere la testina sulla traccia corretta.
- **Tempo di rotazione**
  - tempo medio per raggiungere il settore da trasferire (tempo per 1/2 rotazione).
- **Tempo di trasferimento**
  - tempo per trasferire l'informazione.
- A questi tempi va aggiunto il tempo per le operazioni del controller.

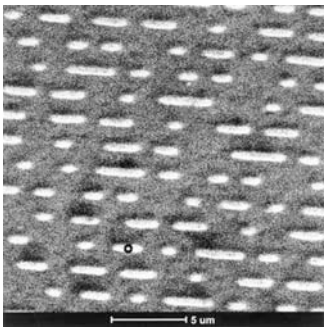
Hard Disk – prestazioni

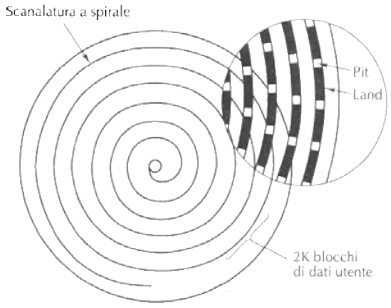
- **IBM/Hitachi DeskStar 7K250 (2004)**
- **Prestazioni**
  - **Capacità: 160 Gbyte**
  - Buffer (cache) Size: 8 MBytes
  - Spindle Speed 7,200 RPM
  - Internal Transfer Rate (max) 94.6 MBytes/sec
  - External (I/O) Transfer Rate (max): 150 MBytes/sec
  - Average Seek Time, Read-Write: 8.5–15 msec typical
  - Track-to-Track Seek, Read-Write: 1.1 msec typical
  - Average Latency: 4.17 msec
  - Probabilità d'errore (non recuperabile): 10^{-14}

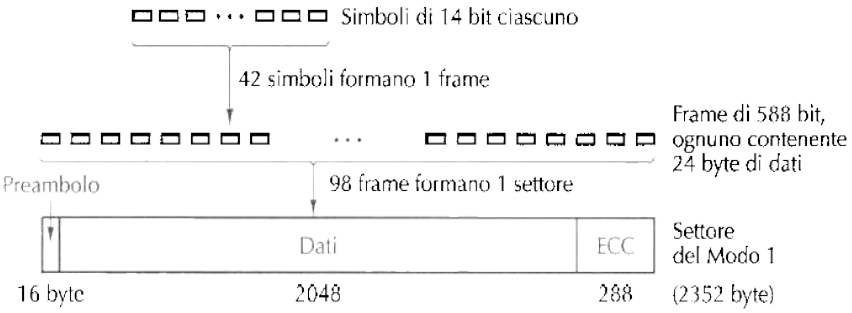
- **Caratteristiche fisiche:**
  - Number of Discs (physical): 3
  - Number of Heads (physical): 4
  - Total Cylinders: 16,383
  - Dimensioni: 101.6 x 146.1 x 25.4mm.
  - Peso: 0.6 kg

CD–ROM

- Struttura: superficie metallica protetta da uno strato di policarbonato
  - **LETTURA**: Un raggio laser infrarosso (λ=780 nm) colpisce la superficie del disco e viene riflesso in modo diverso a seconda dell'incisione
  - Capacità: **700 MByte**
  - Ampiezza pista: **1,6 μm**, lunghezza: **5,6 km**

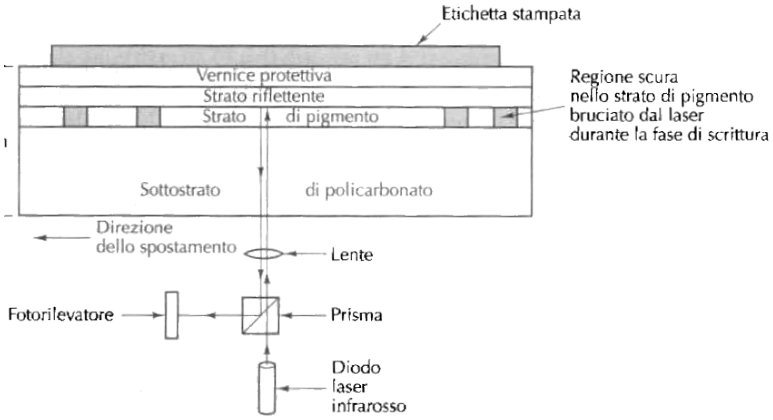






DVD

- **CD–R (recordable)**
  - SCRITTURA: la superficie metallica, riflettente, coperta da un pigmento che diviene opaco quando "bruciato" dal laser.



- DVD (Digital Video Disk)
  - Evoluzione tecnologia, sia per laser che per supporto: lettura/scrittura dati anche in profondità nel metallo (non solo in superficie)

	CD	DVD
largh. traccia:	1,6 μm	0,75 μm
lungh. pit	0,8 μm	0,4 μm
Laser	IR 0,78 μm	red: 0,65 μm
Capacità	0,7 GB	4,75 ÷ 17 GB

- **Blu-ray** DVD: utilizza un laser blu; capacità fino a 50 GB.