


Gestione dell'Input / Output

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it

Università degli Studi di Milano

Riferimento Patterson: 6.3, 6.4 e 6.6


Sommario

Funzionamento dei device driver:

- A controllo da programma diretto
- A controllo da programma con polling
- Ad interrupt
- Ad accesso diretto alla memoria (DMA)

I dischi

Funzionamento dei device driver

Funzione dei driver:
- Controllano l'operato dei device controller.
- Gestiscono lo scambio dei dati dal controller (registro dati) e la memoria.

I/O a controllo da programma.
I/O a controllo da programma con polling.
I/O mediante interrupt.
DMA.

I/O a controllo di programma

E.g. chiamata syscall per la stampa di una stringa.

La periferica ha un ruolo passivo. Il processore esegue tutto il lavoro.

Svantaggio: 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. Stato *busy waiting* o *spin lock*.

```
begin
        1. Predisponi i registri del controller ad effettuare una
        operazione di lettura.
        2. While (ready-bit = = 0) do;          // spin lock (o busy waiting)
        3. Carica il dato acquisito;
end;
```

Esempio: Receiver

```
# NB i dispositivi vengono indirizzati tramite gli indirizzi "alti".

.text
.globl main
main:
        li $t0, 0x1 0000 0000       # indirizzo del receiver control register (4Gbyte)
        li $t2, 0x1 0000 0004        # indirizzo del receiver data register

# Ciclo di lettura di un carattere
ciclo:  lw $t1, 0($t0)              # Contenuto del registro di controllo
        rem $t1, $t1, 2             # if $t1 = = 1 (resto = 1) esci
        beqz $t1, ciclo
        lw $a0, 0($t2)

        li $v0, 10                  # exit
        syscall
```


I/O a gestione di programma - costo

Ipotesi:
1) Tastiera gestita a controllo di programma che opera a 0,01Kbyte/s.
2) Frequenza di clock: 50Mhz.

Determinare il tempo in cui verrebbe effettivamente utilizzata la CPU per trasferire 1 word, tenendo conto che ci vogliono 20 cicli di clock per trasferire ogni byte.

T = 4 [byte] / 10 [byte] / [s] = 0,4s

#cicli_clock = $50*10^6$ [#cicli] / [s] * 0,4 [s] = $20,0 * 10^6$ [#cicli]

Invece ne utilizza solo 20 * 4 = 80 per trasferire i dati.

%Sfruttamento della CPU è: $(80/20,0 * 10^6) *100 = 0,0004\%$

I/O a gestione di programma - costo

Ipotesi:

1) Hard-disk1 che opera a 50Kbyte/s.
2) Hard-disk2 che lavora a 2MByte /s.
3) Frequenza di clock: 50Mhz.

Determinare la percentuale di tempo in cui verrebbe effettivamente utilizzata la CPU per trasferire 1 word, tenendo conto che ci vogliono 20 cicli di clock per ogni byte (approssimare 1k = 1,000).

Hard-disk1: 4 byte / 50kbyte/s = 80µs fase attiva =>
$50x10^6$/s (#cicli_clock/tempo) * $80x10^{-6}$s (tempo_trasferimento in #cicli) =
4000#cicli_clock => %sfruttamento = (20*4) / $4x10^3$ = 2%

Hard-disk2: 4 byte / 2Mbyte /s = 2µs fase attiva =>
$50x10^6$/s (#cicli_clock/tempo) * $2x10^{-6}$ (tempo_trasferimento in #cicli) =
100 cicli_clock => %sfruttamento = (20*4) / 100 = 80%.

Spin-lock è un tempo dedicato all'attesa.

Sommario

Funzionamento dei device driver:

- A controllo da programma diretto
- A controllo da programma con polling
- Ad interrupt
- Ad accesso diretto alla memoria (DMA)

I dischi

Polling

Interrogazione del registro di stato della periferica.

Ciclo di polling: durante un ciclo di busy-waiting su un dispositivo si esegue il **polling** sugli altri dispositivi di I/O.

Quando una periferica necessita di un qualche intervento, si soddisfa la richiesta e si prosegue il ciclo di polling sugli altri I/O.

```
// Leggi dato da perif_x
begin
a.   Predisponi i registri dei controller ad eseguire una read;
b.   if(ready_bit (perif_1) = = 1) servi perif_1;    #Esempio: Mouse
     if (ready_bit (perif_2) = = 1) servi perif_2;   #Esempio: Hard disk1
     if (ready_bit (perif_3) = = 1) servi perif_3;   #Esempio: Hard disk2

          ….
     if (ready_bit (perif_n) = = 1) servi perif_n;
     UpdateFunctions;                                #Programma di gestione in funzione della
                                                     #situazione delle periferiche (sistemi di controllo)
     goto b;
end;
```


I/O a gestione di programma – costo polling

Ipotesi:

1) Costo del polling (# cicli di clock per un'operazione di polling, costituita da trasferimento del controllo alla procedura di polling, accesso al dispositivo di I/O, trasferimento dati e ritorno al programma utente): 400 cicli di clock.
2) Frequenza di clock: 500Mhz
3) Parola di 4 byte.

Determinare l'impatto del polling per 3 dispositivi diversi:

A) Mouse. Deve essere interrogato almeno 30 volte al secondo per non perdere alcun movimento dell'utente.
B) Hard disk. Trasferisce dati al processore in parole da 16 bit ad una velocità di 50 Kbyte/s.
C) Hard disk. Trasferisce dati al processore in blocchi di 4 parole e può trasferire 4 Mbyte/s.

Supponiamo che il costo del trasferimento per la CPU sia dovuto principalmente alle operazioni di preparazione e di richiesta di bus, mentre il costo per la CPU dovuto al trasferimento tramite il bus sia trascurabile (ad esempio perchè viene utilizzata la modalità *burst* delle DRAM).

I/O a gestione di programma - costo

Mouse:

(Per ogni accesso trasferisco 2 byte: x, y)
Occorrono quindi 30 accessi/s.
In termini di cicli di clock: 30 x 400 = 12,000 cicli_clock/s
12,000 / 500,000,000 = 0,000024s => 0,0024%
Piccolo impatto sulle prestazioni.

Hard disk1:

Per ogni accesso possiamo trasferire (half word).
Occorrono quindi 25k accessi/s.
In termini di cicli di clock: 25k x 400 = 10Mcicli_clock/s => 2%
Medio impatto sulle prestazioni.

Hard disk2:

Per ogni accesso possiamo trasferire 16byte.
Occorrono quindi 250k accessi/s
In termini di cicli di clock: 250k x 400 = 100Mcicli_clock/s => 20%
Alto impatto sulle prestazioni.

Ciclo di polling 22,0024% utilizzo CPU

I/O a controllo di programma

- I miglioramenti del polling rispetto al controllo di programma sono molto limitati.

- I problemi principali del polling (e dell'I/O a controllo di programma) sono:
  - Con periferiche lente, un eccessivo spreco del tempo di CPU, che per la maggior parte del tempo rimane occupata nel ciclo di busy waiting.
  - Con periferiche veloci, il lavoro svolto dalla CPU è quasi interamente dovuto all'effettivo trasferimento dei dati.

Il polling funziona bene per i sistemi embedded.

Nei dischi, si potrebbe attivare il polling quando ne è richiesto l'utilizzo.
Posizionamento delle testine (=> spin lock)

Sommario

Funzionamento dei device driver:

- A controllo da programma diretto
- A controllo da programma con polling
- Ad interrupt
- Ad accesso diretto alla memoria (DMA)

I dischi

Interrupt

E' la periferica a segnalare al processore (su una linea del bus dedicata) di avere bisogno di attenzione.

La segnalazione viene chiamata ***interrupt*** perché interrompe il normale funzionamento del processore (***interrupt request***).

Quando il processore "se ne accorge" (fase di fetch), riceve un segnale di ***interrupt acknowledge***.

Viene eseguita una procedura speciale, chiamata ***procedura di risposta all'interrupt***.

Problema: Il programma utente deve potere procedere dal punto in cui è stato interrotto ➔ *Salvataggio del contesto.*

Interrupt – esempio – comando print

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.
4. Il programma di risposta all'interruzione, provvederà a trasferire alla periferica il dato che su vuole stampare.

I/O ad interrupt - costo

Frequenza di clock è 500Mhz
Il costo di ogni interruzione è 500 cicli di clock.

Hard disk:

Trasferimento di blocchi di 4 parole
Trasferimento a 4Mbyte/s
Il disco sta trasferendo dati solamente per il 5% del tempo.

•Trasferimento di 1Mword/s => Occorrono 250k interruzioni/s
•Costo dell'interruzione 250k int/s * 500 cicli_clock = 125M cicli_clock/s
•Frazione di utilizzo del processore per il trasferimento (interrupt) nel caso di trasferimento continuo da disco: 125M / 500M = 25%
•Frazione di utilizzo del processore, tenendo conto che il disco trasferisce solo per il 5% del tempo: 125M / 500M *0.05 = 1,25%

NB L'interrupt è più costoso del polling dal punto di vista dell'esecuzione in senso stretto, ma il costo si recupera perchè l'esecuzione della risposta all'interrupt è attiva solamente in concomitanza dell'interrupt.

Sommario

Funzionamento dei device driver:

- A controllo da programma diretto
- A controllo da programma con polling
- Ad interrupt
- Ad accesso diretto alla memoria (DMA)

I dischi

DMA

Tra il momento in cui termina l'invio del comando al controller ed il momento in cui il dato è disponibile sul controller, la CPU può fare altro (tipicamente l'esecuzione di un altro programma).

Il meccanismo interrupt driven non svincola la CPU dal dovere eseguire le operazioni di trasferimento dati.

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 in Direct Memory Access (DMA).

Viene disaccoppiato il colloquio processore-Memoria dal colloquio IO-Memoria. Questo è reso più facile dalla struttura a bus gerarchici.

Il device controller che gestisce il trasferimento diventa **bus master**.

I passi della DMA

Il DMA controller è un processore specializzato nel trasferimento dati tra dispositivo di I/O e memoria centrale.

Per attivare il trasferimento viene richiesto alla CPU:
1. Spedire al DMA controller il tipo di operazione richiesta
2. Spedire al DMA controller l'indirizzo da cui iniziare a leggere/scrivere i dati.
3. Spedire al DMA controller il numero di byte riservati in memoria.

Per attivare il trasferimento al controller viene richiesta la corretta lettura dello stato della memoria e l'aggiornamento dell'indirizzo a cui trasferire il dato. *E' il controller che gestisce il trasferimento del singolo dato.*

Caratteristiche della DMA

La CPU si svincola completamente dall'esecuzione dell'operazione di I/O.

Il controller avvia l'operazione richiesta e trasferisce i dati da/verso memoria mentre la CPU sta facendo altro.

Dopo avere trasferito tutti i dati, il DMA invia un interrupt alla CPU per segnalare il completamento del trasferimento.

La CPU perciò controlla il (device) controller.

I/O DMA - costo

Frequenza di clock è 500Mhz
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.

Hard disk:
Trasferimento di blocchi di 8kbyte per ogni DMA.
Trasferimento a 4Mbyte/s

Per ciascun trasferimento DMA occorre:
1000 + 500 cicli di clock = tempo di inizio + tempo di fine
Numero di DMA: 4(Mbyte/s) / 8kbyte = 500 / s
Numero di cicli di clock richiesti: 1500 * 500 = 750,000
Frazione del processore utilizzata: 750k / 500M = 0,15%

E' sottointeso che il tempo di trasferimento sia << al periodo di attivazione della DMA.
La DMA viene attivata ogni 2ms (500 DMA / s).
Il tempo richiesto per trasferire da I/O a memoria tramite bus 8Kbyte deve essere < 2ms.

Problema?

Sommario

Funzionamento dei device driver:

- A controllo da programma diretto
- A controllo da programma con polling
- Ad interrupt
- Ad accesso diretto alla memoria (DMA)

I dischi

Dischi magnetici

- Consentono di memorizzare dati in modo non volatile.
- I dati sono letti/scritti mediante una testina.
- I dischi magnetici sono di due tipi principali:
  - hard disk
  - floppy disk (messi in commercio da Apple e Tandy nel 1978). In precedenza esistevano solamente le cassette magnetiche, a loro volta evoluzione dei nastri magnetici immessi sul mercato nel 1934 in Germania dalla IG Farben, ora Basf, per un magnetofono AEG.

Hard disk

- Costituiti da un insieme di piatti rotanti (da 1 fino a 25) ognuno con due facce, di diametro che va da 2.5cm a 10cm.
- La pila dei piatti viene fatta ruotare alla stessa velocità (5,400 – 15,000 rpm = revolutions per minute).
- Ogni faccia è divisa in circonferenze concentriche chiamate **tracce** (10,000-50,000).
- Ogni traccia è suddivisa in **settori** (64 - 500).
- I settori sono suddivisi da **gap**
- Il settore è la più piccola unità che può essere letta/scritta da/su disco (tipicamente blocco da 512byte, ma c'è la spinta a portarli a 4,096byte).
- Esiste una testina per ogni faccia.
- Le testine di facce diverse sono collegate tra loro e si muovono solidalmente.

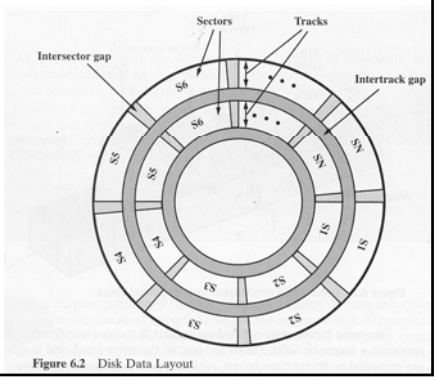
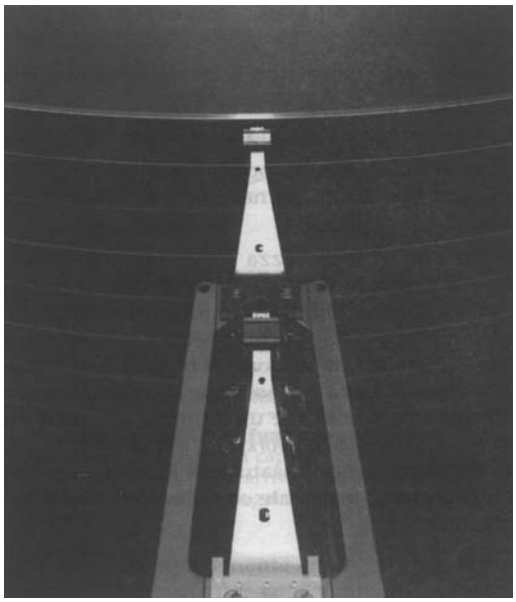


Figure 6.2 Disk Data Layout

Hard disk

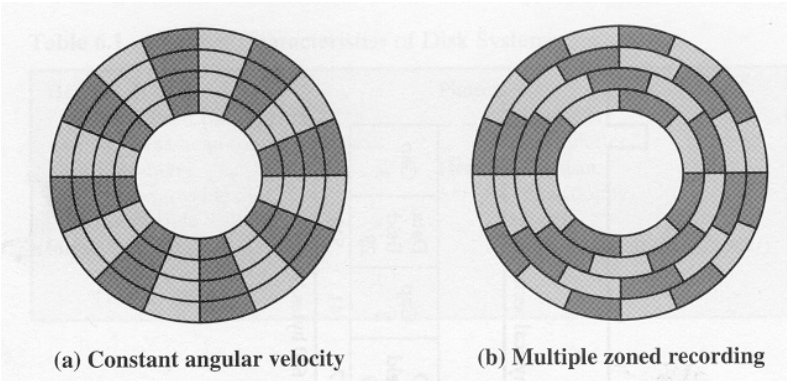
- Le testine si muovono in modo solidale.
- L'insieme delle tracce di ugual posto su piatti diversi è chiamato **cilindro**.
- La quantità di dati che possono essere memorizzati per traccia dipende dalla qualità del disco.
- Solitamente, ogni traccia di un disco contiene la stessa quantità di bit $\Rightarrow$ le tracce più esterne memorizzano informazione con densità minore.

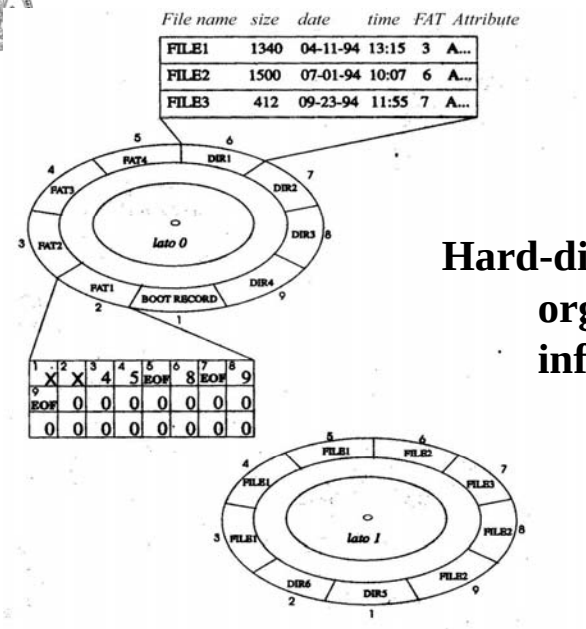


Memorizzazione dati su disco

La velocità di rotazione è costante.
Per ogni settore, il numero di bit per traccia è uguale.
Le tracce interne determinano la densità di bit.

Per aumentare l'impaccamento dell'informazione si utilizza la tecnologia del **multiple zone recording**.





Hard-disk: organizzazione delle informazioni

La traccia 0 contiene il contenuto del disco.

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 o 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). Misurato in rpm (rounds per minute = giri al **minuto**).
- **Tempo di trasferimento**: tempo per trasferire l'informazione.
- A questi tempi va aggiunto il tempo per le operazioni del **controller.**

Hard Disk - Seagate Ceetah 18XL

Dimensioni: 101.6 x 146.1 x 25.4mm.
Capacità: 18.2 Gbyte.

Forma: low-profile.
Default Buffer (cache) Size 4,096 Kbytes

Peso: 0.68kg.
Spindle Speed 10,000 RPM

Number of Discs (physical): 3
Total Cylinders: 14,384

Number of Heads (physical): 6
Bytes Per Sector: 512

Internal Transfer Rate (min-max): 284 Mbits/sec - 424 Mbits/sec
Formatted Int Transfer Rate (min-max) 26.6 MBytes/sec - 40.5 MBytes/sec
External (I/O) Transfer Rate (max): 200 MBytes/sec
Avg Formatted Transfer Rate: 35.5 MBytes/sec

Average Seek Time, Read-Write: 5.2-6msec typical
Track-to-Track Seek, Read-Write: 0.6-0.8msec typical
Average Latency: 2.99 msec

Typical Current (12VDC +/- 5%): 0.5 amps
Typical Current (5VDC +/- 5%): 0.8 amps
Idle Power (typ): 9.5 watts

Alcuni dischi

Caratteristiche	Seagate Barracuda 180	Seagate Cheetah X15-36LP	Seagate Barracuda 36ES	Toshiba HDD-1242	IBM Microdrive
Applicazione	High-capac Server	High-perf Server	Entry-level Desktop	Portable	Handheld device
Capacità	181.6Gbyte	36.7Gbyte	18.4Gbyte	5Gbyte	1Gbyte
Minimum track-to-track seek time	0.8ms	0.3ms	1.0ms	-	1.0ms
Average seek-time	7.4ms	3.6ms	9.5ms	15ms	12ms
Spindle speed	7,200 rpm	15k rpm	7,200 rpm	4,200 rpm	3,600 rpm
Average rotation delay	4.17ms	2ms	4.17ms	7.14ms	8.33ms
Maximum transfer rate	160Mbyte/s	522-709 Mbyte/s	25 Mbyte/s	66 Mbyte/s	13.3 Mbyte/s
Bytes per sector	512	512	512	512	512
Sector per track	793	485	600	63	-
Tracks per cylinder (number of platter surfaces)	24	8	2	2	2
Cylinders (number of tracks on one side of platter)	24,247	18,479	29,851	10,350	-

Hard disk - prestazioni

- Tempo medio di seek: da 8 a 20 ms (può diminuire di più del 75% se si usano delle ottimizzazioni).
- Tempo medio di rotazione: da 2.8 ms a 5.6 ms.
- Tempo medio di trasferimento: 2/15 MB per secondo e oltre con cache.
- Tempo di controllo (utilizzato dalla logica del controller).

Qual è il tempo di lettura/scrittura di un settore di 512byte in un disco che ha velocità di rotazione di 7,200 rpm? Il tempo medio di seek è di 12ms, la velocità di trasferimento di 10Mbyte/s ed il tempo aggiuntivo richiesto dal controllore è di 2ms.

12ms + (1/120 /2)*1000 ms + 0,5kbyte / 10Mbyte/s + 2ms = 12 + 4,2 + 0,05 + 2 = 18,25ms

Per un tempo di seek medio pari al 25% del tempo nominale (t_seek = 3ms)
3ms + 4,2ms + 0,5kbyte / 10Mbyte/s + 2ms = 9,25ms

RAID

RAID è un acronimo che sta per Redundant Array of Independent Disks (originariamente, 1988, stava per Redundant Array of Inexpensive Disks).

Ha queste caratteristiche:

1) RAID è un insieme, array, di dischi fisici visto dal sistema operativo come un drive logico singolo.
2) I dati vengono distribuiti attraverso i dispositivi fisici dell'array di dischi.
3) La capacità ridondante dei dischi viene utilizzata per memorizzare l'informazione di parità, che garantisce di potere recuperare i dati in casi di guasto (i guasti risultano più frequenti per la maggiore complessità dell'HW).


CD-ROM / DVD

- I CD-ROM sono basati sulla tecnologia laser per la memorizzazione delle informazioni. Vennero lanciati sul mercato nel 1982 da Philips e Sony per la registrzione di suoni.
- Memorizzano l'informazione codificata con incisioni di forme caratteristiche sulla superficie del disco.
- Un raggio laser colpisce la superficie del disco e viene da questa riflesso in modo diverso a seconda della forma della superficie colpita (superficie piatta, o rilievo rugoso che provoca scattering).
- Su CD-ROM è possibile immagazzinare informazione con una densità maggiore rispetto ai dischi magnetici.
- Un CD-ROM può memorizzare più di 650 MB di dati.

- Un DVD (Digital Video Disk) arriva a memorizzare 15.90 Gbyte (DVD-18). Esistono diversi dialetti e diversi formati HW: DVD-R, DVD+R.


Flash memory

- EEPROM (memoria cancellabile elettronicamente: tempi di lettura e scrittura molto diversi)
- Wear leveling (uniformità delle scritture)
- Used in portable devices and in hybrid systems

Characteristics	Kingston SecureDigital (SD) SD4/8 GB	Transend Type I CompactFlash TS16GCF133	RIDATA Solid State Disk 2.5 inch SATA
Formatted data capacity (GB)	8	16	32
Bytes per sector	512	512	512
Data transfer rate (read/write MB/sec)	4	20/18	68/50
Power operating/standby (W)	0.66/0.15	0.66/0.15	2.1/—
Size: height × width × depth (inches)	0.94 × 1.26 × 0.08	1.43 × 1.68 × 0.13	0.35 × 2.75 × 4.00
Weight in grams (454 grams/pound)	2.5	11.4	52
Mean time between failures (hours)	> 1,000,000	> 1,000,000	> 4,000,000
GB/cu. in., GB/watt	84 GB/cu.in., 12 GB/W	51 GB/cu.in., 24 GB/W	8 GB/cu.in., 16 GB/W
Best price (2008)	~ $30	~ $70	~ $300

Sommario

Funzionamento dei device driver:

- A controllo da programma diretto
- A controllo da programma con polling
- Ad interrupt
- Ad accesso diretto alla memoria (DMA)

I dischi