

Dischi magnetici

La geometria:

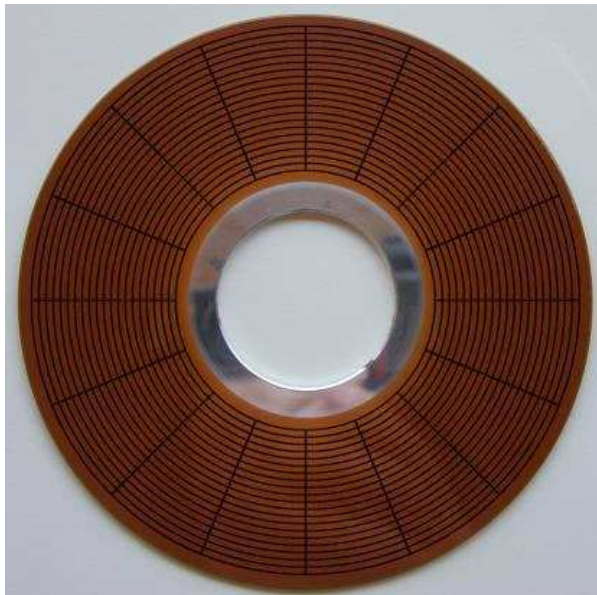
- heads
- cylinders
- sectors

Ogni settore può essere identificato dalla terna (head, cylinder, sector)

1



2

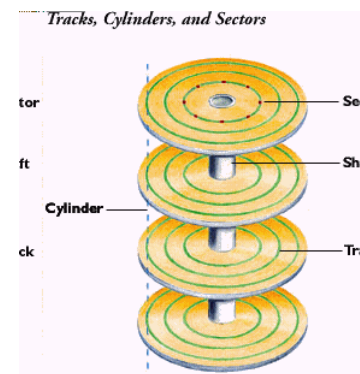


3

La differenza fra traccia e cilindro

L'immagine mostra quattro piatti, con disegnate tre tracce

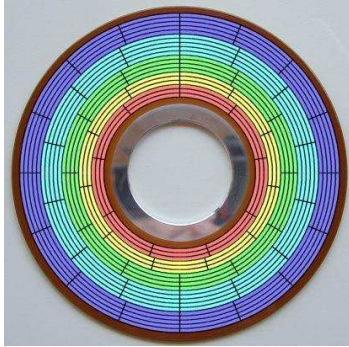
Il cilindro indicato è composto da otto tracce (due per superficie)



4

Zoned Bit Recording

20 tracce divise in 5 zone



5

Il BIOS non “capisce” i dischi formattati con ZBR

Questi dischi presentano al BIOS una geometria virtuale (es. 63 settori per traccia)

Il controller del disco esegue una traduzione da virtuale a fisico

Limite del BIOS:

- 16 bit per specificare il # di settori
 - 4 bit per specificare il # di testine
 - 6 bit per specificare il # di tracce
- ⇒ max # settori = $(2^{16} - 1) * 2^4 * (2^6 - 1) = 66,059,280$
⇒ max capacità = $66,059,280 * 512 = 31.499$ GB

Per superare questo limite si usa il *Logical Block Addressing*:
i settori sono numerati linearmente

6

La formattazione

0. Formattazione di basso livello

- marca la struttura fisica (tracce, settori) sulle superfici
- eseguita dal fabbricante
- richiede di conoscere con precisione la geometria fisica del disco

1. Partizionamento

- divide il disco in “pezzi” logici *basati su una convenzione*
- è una funzione del sistema operativo
- tutti i SO su PC rispettano la convenzione del PC IBM

2. Formattazione di alto livello

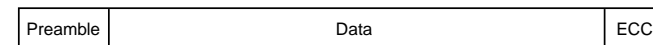
- è una funzione del sistema operativo
- (es. comando “format” di MS-DOS, comando mkfs(8) di Unix)
- costruisce un *filesystem* in una partizione

7

Formattazione di basso livello

- marca la struttura fisica (tracce, settori) sulle superfici

Un settore:



- eventuali blocchi difettosi vengono “rimpiazzati” con blocchi di scorta
- il rimpiazzo è fatto in maniera trasparente dal controller

8

Tempo necessario per leggere o scrivere un blocco

0. seek time
1. rotational delay
2. data transfer time

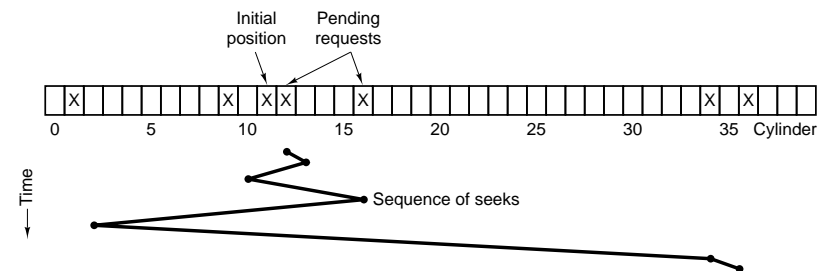
Il tempo di seek domina gli altri due

Se le richieste di I/O sono servite in modo First Come, First Served (FCFS) abbiamo pessima performance

9

Algoritmi per ottimizzare il tempo di seek (i)

Shortest Seek First (SSF)



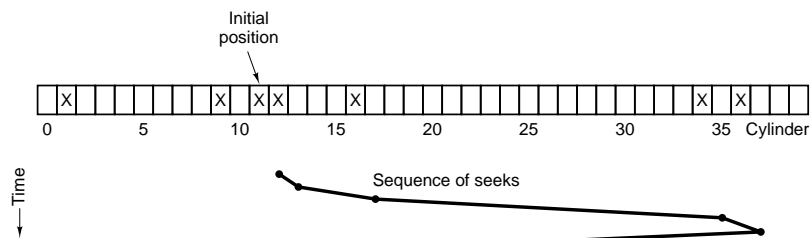
Mediamente è il doppio più veloce di FCFS

Ma ha un problema di *fairness*

10

Algoritmi per ottimizzare il tempo di seek (ii)

Algoritmo dell'ascensore



11

Ma se il disco usa una geometria virtuale...

... diversa dalla geometria fisica?

Il SO non può sapere se il cilindro 39 è più vicino al cilindro 40 o al cilindro 200

Comunque questi algoritmi vengono usati dal controller

Il sistema operativo può comunque *ordinare* le richieste di I/O

12

I/O in unix: buffer cache

Quando il SO riceve una richiesta di lettura, prima va a vedere se il blocco è presente nella cache

Quando il blocco viene letto, viene conservato nella cache

Quando il SO riceve una richiesta di scrittura, la modifica viene fatta sulla cache

A intervalli regolari, i blocchi *sporchi* vengono copiati dalla cache al disco

13

Clocks

Ogni PC ha due orologi:

Real Time Clock (RTC) (o "CMOS Clock")

- mantiene la data e l'ora quando il PC è spento
- non viene usato quando il PC è acceso

System clock (o "kernel clock" o "software clock")

- lancia interrupt a intervalli regolari
- la routine di interrupt viene usata per vari scopi

14

Clock interrupt handler

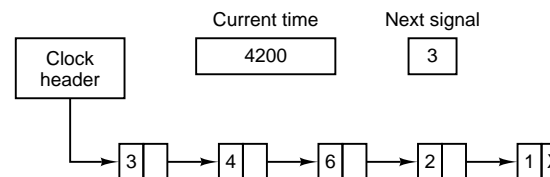
- mantiene l'ora corrente (sotto forma di "numero di secondi passati da 1/1/1970 0:00" ovvero "Unix timestamp")
- implementa lo scheduling con prelazione
- mantiene i contatori di CPU Usage dei processi
- gestisce le syscall `alarm(2)`, `sleep(2)`
- gestisce vari timer interni al kernel
- mantiene statistiche, profilazione, ecc.

In Linux:

- incrementa la var globale `jiffies`
- incrementa la var globale `xtime`

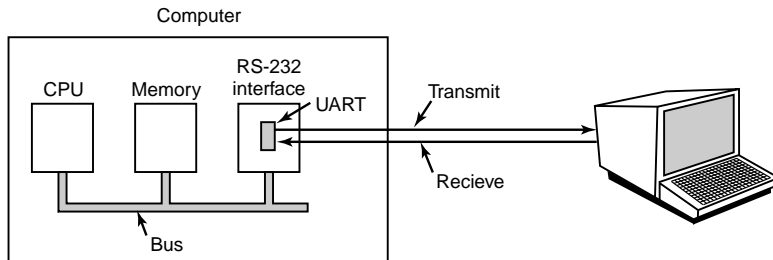
15

Implementare i soft timer



16

Character terminals



17

TTY input: crudo o cotto?

Il driver della tastiera converte i codici dei tasti in codici ASCII

Chi gestisce il tasto per cancellare?

- raw mode: gestito dall'applicazione
es. emacs, vi
l'applicazione riceve i caratteri appena sono digitati
- cooked mode: gestito dal driver
modo di default
l'applicazione riceve i caratteri una linea per volta

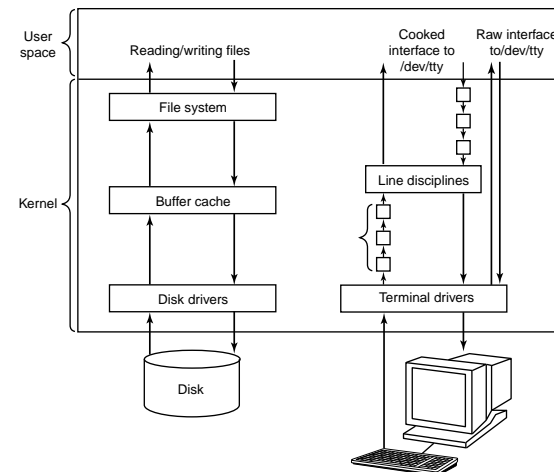
18

Caratteri di controllo

Character	POSIX name	Comment
CTRL-H	ERASE	Backspace one character
CTRL-U	KILL	Erase entire line being typed
CTRL-V	LNEXT	Interpret next character literally
CTRL-S	STOP	Stop output
CTRL-Q	START	Start output
DEL	INTR	Interrupt process (SIGINT)
CTRL-\	QUIT	Force core dump (SIGQUIT)
CTRL-D	EOF	End of file
CTRL-M	CR	Carriage return (unchangeable)
CTRL-J	NL	Linefeed (unchangeable)

19

I/O in Unix



20

TTY Output

Esistono centinaia di terminali diversi (veri e virtuali)

Ciascun terminale

- ha diverse *capacità*
- usa *sequenze di escape* diverse

Problema: realizzare applicazioni *portabili*

Soluzione: il database *terminfo*

21

Un frammento dal terminfo master database

```
dumb|80-column dumb tty,
    am,
    cols#80,
    bel=^G, cr=^M, cud1=^J, ind=^J,
unknown|unknown terminal type,
    gn, use=dumb,
lpr|printer|line printer,
    hc, os, 0Tbs,
    cols#132, lines#66,
    bel=^G, cr=^M, cub1=^H, cud1=^J, ff=^L, ind=^J,
glasstty|classic glass tty interpreting ASCII control characters,
    0Tbs, am,
    cols#80,
    bel=^G, clear=^L, cr=^M, cub1=^H, cud1=^J, ht=^I,
    .kbs=^H, kcub1=^H, kcu1=^J, nel=^M^J,
vanilla,
    0Tbs, bel=^G, cr=^M, cud1=^J, ind=^J,
```

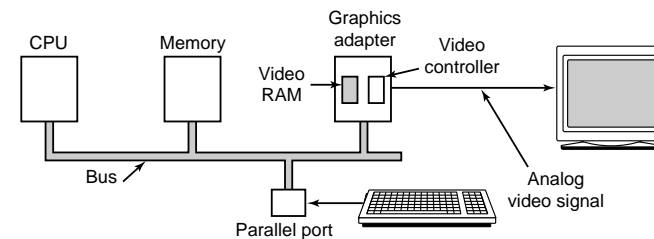
22

Le sequenze di escape ANSI

Escape sequence	Meaning
ESC [<i>n</i> A	Move up <i>n</i> lines
ESC [<i>n</i> B	Move down <i>n</i> lines
ESC [<i>n</i> C	Move right <i>n</i> spaces
ESC [<i>n</i> D	Move left <i>n</i> spaces
ESC [<i>m</i> ; <i>n</i> H	Move cursor to (<i>m</i> , <i>n</i>)
ESC [<i>s</i> J	Clear screen from cursor (0 to end, 1 from start, 2 all)
ESC [<i>s</i> K	Clear line from cursor (0 to end, 1 from start, 2 all)
ESC [<i>n</i> L	Insert <i>n</i> lines at cursor
ESC [<i>n</i> M	Delete <i>n</i> lines at cursor
ESC [<i>n</i> P	Delete <i>n</i> chars at cursor
ESC [<i>n</i> @	Insert <i>n</i> chars at cursor
ESC [<i>nm</i>	Enable rendition <i>n</i> (0=normal, 4=bold, 5=blinking, 7=reverse)
ESC M	Scroll the screen backward if the cursor is on the top line

23

Memory-mapped displays



24

I due modelli di output grafico

raster: basato su una matrice di punti (bitmap)

- es. immagini gif, jpeg

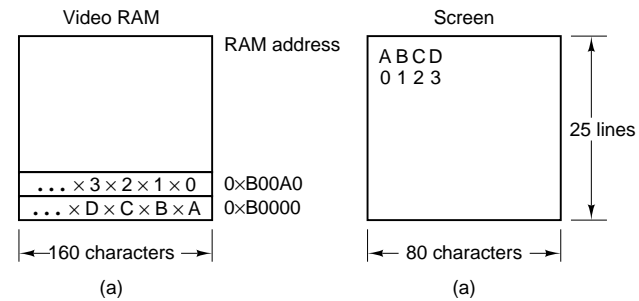
vector: basato su una lista di oggetti (segmenti, punti, spline)

- es. immagini Flash, Scalable Vector Graphics

25

Video controllers

- text mode
- graphics mode



26

Windowing software

I programmi basati su GUI sono centrati su un *event loop*

```
while (1) {
  get event
  if event is mouse button down then ...
  if event is mouse button up then ...
  if event is key down then ...
  if event is key up then ...
  ...
}
```

Gli eventi possono essere *preprocessati*

I sistemi di programmazione più "facili" nascondono l'event loop
(es. Visual Basic, TCL/Tk)

27

Esempio: Win32

L'elemento fondamentale è la *finestra*

La finestra ha una coda di eventi

Alla finestra viene associata una *WindowProc*

```
long MyWndProc(HWND hwnd, UINT message, WPARAM wParam,
               LPARAM lParam)
{
  switch (message) {
    case WM_PAINT: ... /* disegna la finestra */
    case WM_DESTROY: ... /* la finestra viene chiusa */
    ...
  }
  return DefWindowProc(hwnd, message, wParam, lParam);
}
```

28

Un'applicazione Windows inizia con WinMain

```
int WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
LPSTR lpCmdLine, int nCmdShow)
{
    HWND hwnd;

    hwnd = CreateWindowEx(
        // parameters
    );

    ShowWindow(hwnd, SW_SHOW);
    UpdateWindow(hwnd);

    while (GetMessage(&msg, NULL, 0, 0)) {
        DispatchMessage(&msg);
    }
    return msg.wParam;
}
```

29

Disegnare in una finestra

Il SO manda l'evento WM_PAINT quando una parte della finestra deve essere ridisegnata

```
switch (message)
{
    case WM_PAINT:
        hdc = BeginPaint(hwnd, &ps);
        TextOut(hdc, 0, 0, "Hello, Windows!", 15);
        EndPaint(hwnd, &ps);
        return 0L;
    ...
}
```

30

Rappresentare testo

I font possono essere rappresentati come bitmap

Ma i migliori risultati si ottengono con tecnologia vettoriale

Es. TrueType

20 pt: abcdefgh

53 pt: abcdefgh

81 pt: abcdefgh

31

The X Windows System

works in user mode

provides mechanism, not policy

- policy is left to Window Managers
- policy is left to widget libraries

works transparently over a network

32

The X Windows System

