

Acquisizione e memorizzazione delle informazioni

Davide Quaglia

Sommario

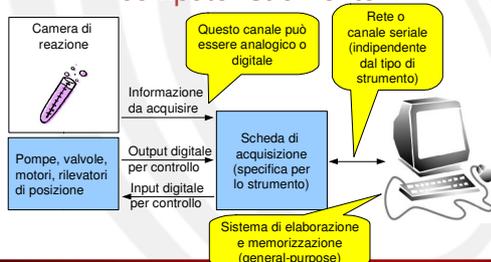
- Acquisizione da sensori
- Cenni sulla compressione
- Memorizzazione su supporti di massa
- Alcuni formati di memorizzazione

Acquisizione da sensori

Introduzione

- Molti strumenti di laboratorio sono oggi automatizzati
- Il fenomeno bio-fisico non viene più osservato da un essere umano ma viene trasformato in numeri per essere elaborato da un calcolatore
 - Acquisizione
 - Memorizzazione

Struttura di una interfaccia computer-strumento



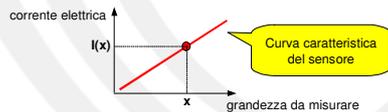
Informazione da acquisire

- L'informazione da acquisire è una grandezza che può essere misurata attraverso
 - una misura di intensità luminosa
 - una misura di concentrazione ionica
 - una misura di tensione/corrente proveniente da altro tipo di sensore
 - Temperatura
 - Pressione
 - Ecc...

Informazione da acquisire (2)

- Se conosco la legge che lega la grandezza da misurare (ad es. la concentrazione di una sostanza) con una grandezza che riesco a misurare (ad es. una corrente elettrica) allora posso automatizzare la misura

• Es.



7

Misura di intensità luminosa

- I misuratori di intensità luminosa convertono la luce in un segnale elettrico
- Spesso le caratteristiche costruttive di tali misuratori li rendono sensibili ad una specifica frequenza dello spettro luminoso
- Tale selettività si può ottenere anche antepponendo un filtro ottico davanti al misuratore

8

Misura di intensità luminosa (2)

- Certi atomi o molecole, se eccitati da energia sotto forma di luce o calore possono rilasciare tale energia sotto forma di luce di una lunghezza d'onda ben precisa (luminescenza, fluorescenza, fosforescenza)
- La misura di intensità luminosa può determinare la concentrazione di tali sostanze

9

Misura di concentrazione ionica

- Uno ione è una molecola dotata di carica elettrica
- Il movimento di uno ione e il suo urto genera
 - una corrente elettrica (misurazione di intensità)
 - un impulso elettrico (conteggio)
 - una emissione di luce (misurazione di intensità)
- Intensità di corrente, numero di impulsi di corrente oppure intensità luminosa sono proporzionali alla concentrazione della sostanza da analizzare
- I dispositivi di misura possono avere un meccanismo di moltiplicazione del flusso di ioni per amplificare l'effetto e facilitarne la rilevazione

10

Altri tipi di sensori

- Termocoppia: giunzione di due metalli che assume una differenza di potenziale (tensione) proporzionale alla temperatura
- Termoresistenza: solido metallico la cui resistenza elettrica è proporzionale alla temperatura
 - Es. PT100 = sonda di platino
- Sensore piezoelettrico per misure di pressione: la pressione meccanica su un cristallo crea una tensione tra certe sue facce

11

Segnale analogico e segnale digitale

- Un **segnale analogico** consiste di **valori continui** esprimibili solo mediante **numeri reali**
- Un **segnale digitale** consiste in una **sequenza discreta** di numeri interi con **precisione finita**
 - i calcolatori sanno elaborare solo segnali digitali e, in particolare, rappresentati in base 2

12

Natura dell'informazione acquisita

- La rilevazione del fenomeno fisico può consistere nel conteggio di eventi
 - l'informazione acquisita è già in formato digitale (sequenza di numeri)
 - Es. contatore Geiger per la radioattività
- La rilevazione del fenomeno può generare un segnale elettrico con tensione/corrente proporzionale all'intensità della grandezza misurata
 - l'informazione è un segnale continuo e quindi necessita di una conversione analogico/digitale

13

Calibrazione

- La curva caratteristica dei sensori può subire variazioni dovute a fattori ambientali o usura
- Spesso la curva subisce solo una traslazione o una variazione di pendenza o di scala
- Calibrazione
 - si verifica la presenza di una variazione misurando un campione di valore noto
 - si compensano le variazioni della curva nel sistema di misurazione
 - si lancia un segnale di allarme in caso di compensazione impossibile

14

Conversione analogico/digitale

- La conversione analogico/digitale consiste in due azioni **indipendenti**:
 - campionamento
 - quantizzazione

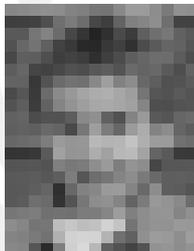
15

Campionamento di un segnale

- Per catturare la variazione di un segnale nel suo dominio (es. tempo o spazio) occorre ripetere la **misura** in punti ad intervalli regolari di tale dominio (es. istanti di tempo o punti dello spazio).
- Intuitivamente più il segnale varia velocemente (ad es. nel tempo o nello spazio) e maggiormente vicini devono essere tali punti cioè maggiore deve essere la **frequenza di campionamento**.

16

Quanti campioni ?



16 x 21 pixel

17

Quanti campioni ?



64 x 85 pixel

18

ESD

Quanti campioni ?



128 x 171 pixel

19

ESD

Quanti campioni ?



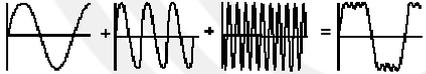
512 x 683 pixel

20

ESD

Teorema del campionamento

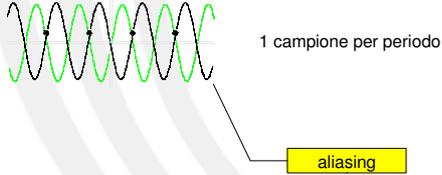
- Un segnale può sempre essere rappresentato come **somma di sinusoidi** (rappresentazione in frequenza).
- Per ricreare fedelmente un segnale occorre ricreare fedelmente **tutte** le sinusoidi che lo compongono.



21

ESD

Teorema del campionamento (2)



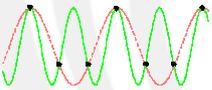
1 campione per periodo

aliasing

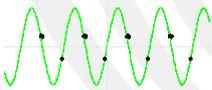
22

ESD

Teorema del campionamento (3)



1.5 campioni per periodo



2 campioni per periodo

23

ESD

Teorema del campionamento (4)

- Per riprodurre fedelmente una sinusoida occorrono 2 campioni per periodo.
- La frequenza di campionamento deve essere doppia della frequenza della sinusoida.

24

Teorema del campionamento (5)

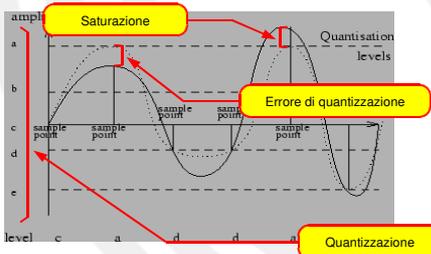
- Se il segnale da campionare ha **larghezza di banda f** allora la sua componente sinusoidale a frequenza più alta ha frequenza f .
- Quindi la frequenza di campionamento di tale segnale deve essere doppia della larghezza di banda del segnale (**Teorema di Nyquist**).

Quantizzazione di una misura

Assegnazione di una misura a precisione infinita ad un numero avente precisione finita.

- 1) Determinazione di un range di variazione.
- 2) Individuazione di un insieme di punti **finito** all'interno di tale range.
- 3) Assegnamento di un **numero intero** a ciascun punto.
- 4) Misurazione del segnale analogico e rappresentazione della misura con il numero corrispondente al punto più vicino.

Quantizzazione: problemi



Effetto della quantizzazione sulle immagini



256 livelli (8 bit)

8 livelli (3 bit)

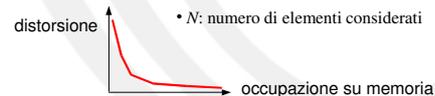
Effetto della quantizzazione

- La quantizzazione implica sempre un errore di arrotondamento detto di **errore di quantizzazione**.
- Maggiore è il numero di bit su cui si rappresenta la misura e minore è l'errore di quantizzazione.
- Tale errore è irreversibile.
 - Quando un procedimento di acquisizione/memorizzazione dati prevede uno o più fasi di quantizzazione allora non è possibile ritornare alla rappresentazione originale.
- **Codifica dell'informazione con perdita (lossy)**

Distorsione di quantizzazione

- Una possibile misura dell'errore di quantizzazione è detta **distorsione**
- Una possibile misura della distorsione su una sequenza $\{v_i\}$ è data da:
 - v_i : simbolo prima della quantizzazione
 - v'_i : simbolo dopo la decodifica
 - N : numero di elementi considerati

$$MSE = \frac{\sum_{i=1}^N (v_i - v'_i)^2}{N}$$



Cenni sulla compressione delle informazioni

Premessa

- Quanto si dirà in questi lucidi si applica a sorgenti di informazioni che producono **serie di dati discreti a precisione finita**:
 - dati digitali su file (es., eseguibili, database)
 - sorgenti naturali dopo campionamento e quantizzazione
- Le tecniche che verranno presentate **non** provocano una perdita irreversibile di informazione.

Definizioni

- Una sorgente di informazione emette una **serie ordinata di dati**
- Tale serie di dati si può rappresentare come una **sequenza di simboli** dove ciascun simbolo appartiene ad certo **alfabeto** dipendente dal tipo di applicazione
 - Es: campioni di temperatura quantizzati su 16 bit con segno
 - l'alfabeto è costituito dai numeri interi nell'intervallo [-32768, 32767]
 - Es: mappa di fluorescenza con pixel su 256 livelli di grigio
 - l'alfabeto è costituito dai numeri interi nell'intervallo [0,255]

Cosa vuol dire compressione ?

- Premesso che ogni simbolo deve essere codificato con dei bit per essere elaborato o trasmesso
- Definizione teorica di compressione

*Cambio **reversibile** di alfabeto tale che la rappresentazione binaria della sequenza di simboli, tradotta nei simboli del nuovo alfabeto, richiede meno bit per essere memorizzata.*

Eliminazione della correlazione statistica

Correlazione statistica

- Sia data una sequenza ordinata di simboli s_0, s_1, \dots, s_n di un certo alfabeto emessa da una sorgente di informazione.
- C'è correlazione quando la probabilità che s_i assuma un certo valore dipende dal valore assunto dagli elementi precedenti della sequenza.
- Si dice che l'informazione contenuta nella sequenza è **ridondante** e si può cambiare alfabeto in modo da usare meno bit.

Esempio di correlazione nei testi

- 13% di lettere eliminate
- quante parole si riesce a ricostruire ?

Le sp zie s no aleate pre iose nella
p epa azi ne di suc ulenti pia ti. Es e
s no ben con sci te fin dall'an ichi à;
at or o ad es e vi er no ve e e pro rie
vie com erci li e per esse si
c mba terono du e bat agl e.

Esempio di correlazione nei testi

- 6% di lettere eliminate
- quante parole si riesce a ricostruire ?

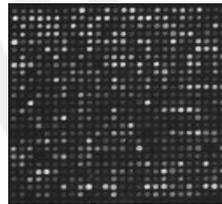
Le sp zie sono aleate pre iose nella
prepa azione di suc ulenti pia ti. Es e
sono ben conosciute fin dall'antichità;
at orno ad es e vi erano vere e proprie
vie com erciali e per esse si
comba terono dure bat aglie.

Esempio di correlazione nei testi

Le spezie sono alleate preziose nella
preparazione di succulenti piatti. Esse
sono ben conosciute fin dall'antichità;
attorno ad esse vi erano vere e proprie
vie commerciali e per esse si
combattono dure battaglie.

Esempio di correlazione nelle immagini

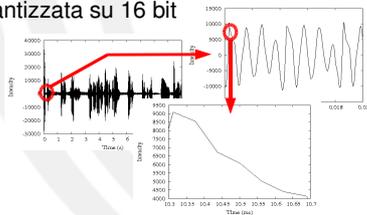
Immagine b/n da scansione di micro array



← Spazi tra un
pozzetto e l'altro

Esempio di correlazione in una sequenza temporale

- Andamento nel tempo della temperatura
quantizzata su 16 bit



Eliminazione della correlazione

- **Si cambia alfabeto di rappresentazione**
- Codifica run-length
 - Es. in una immagine b/n a due livelli
1,1,1,1,0,0,0,0,0,1,1,1,1,1 → (1,4),(0,6),(1,5)
- Codifica differenziale
 - Es. della temperatura (vedi lucido precedente)
 $s_0, s_1, s_2, \dots, s_n \rightarrow (s_1 - s_0), (s_2 - s_1), \dots, (s_n - s_{n-1})$
- Trasformata
 - Es. della temperatura (vedi lucido precedente)
Sviluppo in serie di Fourier: ampiezza delle componenti sinusoidali
 $s_0, s_1, s_2, \dots, s_n \rightarrow a_0, a_1, \dots, a_n$

ESD

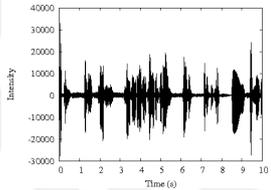
Codifica entropica

43

ESD

Frequenza dei simboli: esempio

- Sequenza temporale su 16 bit: alfabeto = interi su [-32768, 32767]



44

ESD

Quanti bit/simbolo ?

- E' meglio rappresentare con meno bit i simboli più frequenti e con più bit quelli meno frequenti.

45

ESD

Quanti bit/simbolo ? (2)

- Numero minimo teorico di bit/simbolo per l'alfabeto s:

$$H(s) = \sum_k p_k \log_2 \frac{1}{p_k}$$

- p_k è la frequenza di apparizione del simbolo k -esimo
- entropia della sorgente di simboli (Shannon)**
- tale formula fornisce un valore medio minimo ma non dice come assegnare i bit ai simboli per ottenerlo !

46

ESD

Esempio di calcolo dell'entropia

- Una sorgente emette simboli appartenenti ad un alfabeto di 4 simboli a, b, c, d
- Frequenze di apparizione dei 4 simboli sono:
 - 1/2, 1/4, 1/8, 1/8
 - somma delle frequenze uguale a 1
- Entropia della sorgente:

$$H(s) = \frac{1}{2} \log_2(2) + \frac{1}{4} \log_2(4) + \frac{1}{8} \log_2(8) + \frac{1}{8} \log_2(8)$$

$$= \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8} = 1.75 \text{ bit / simbolo} \leq \lceil \log(4) \rceil \text{ bit / simbolo}$$

47

ESD

Codifica entropica

- Per ottenere una lunghezza media dell'alfabeto vicina al valore dell'entropia occorre rappresentare i simboli su un numero variabile di bit



- Problema: come si fa a sapere quando inizia/finisce un simbolo ?

48

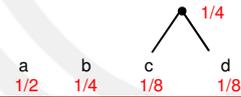
Algoritmo di Huffman

- Metodo di assegnamento bit ai simboli che garantisce la corretta decodifica
- Si costruisce un albero binario dove le foglie rappresentano i simboli
- Ad ogni biforcazione la somma delle frequenze dei simboli del sotto-albero di destra deve essere il più possibile vicina alla somma delle frequenze dei simboli del sotto-albero di sinistra.
- Ciascun arco di una biforcazione è etichettato con 0 e 1 rispettivamente

Creazione codice di Huffman

- Alfabeto di 4 simboli a, b, c, d con frequenze rispettivamente di $1/2, 1/4, 1/8, 1/8$
- Primo passo: a, b, c, d

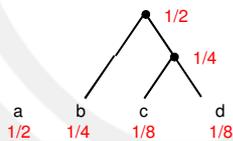
$1/2 \quad 1/4 \quad 1/8 \quad 1/8$



Creazione codice di Huffman

- Alfabeto di 4 simboli a, b, c, d con frequenze rispettivamente di $1/2, 1/4, 1/8, 1/8$
- Secondo passo: a, b, c, d

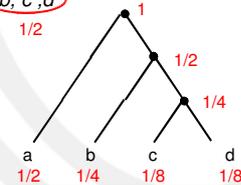
$1/2 \quad 1/4 \quad 1/4$



Creazione codice di Huffman

- Alfabeto di 4 simboli a, b, c, d con frequenze rispettivamente di $1/2, 1/4, 1/8, 1/8$
- Terzo passo: a, b, c, d

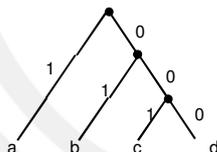
$1/2 \quad 1/2$



Creazione codice di Huffman

- Alfabeto di 4 simboli a, b, c, d con frequenze rispettivamente di $1/2, 1/4, 1/8, 1/8$
- Creazione del codice:

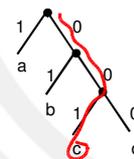
$a \rightarrow 1$
 $b \rightarrow 01$
 $c \rightarrow 001$
 $d \rightarrow 000$



Decodifica codice di Huffman

- In ricezione per decodificare i simboli si visita l'albero binario in base ai bit in arrivo

Es: 001 \rightarrow c



Huffman: considerazioni

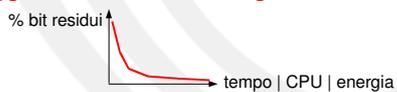
- L'algoritmo di Huffman è ottimo (rispetto all'entropia) se le frequenze sono potenze di 2.
- Per distribuzioni di frequenze più complicate esistono algoritmi più sofisticati
 - Codifica aritmetica
 - Codici di Golomb

Codifica entropica: altri problemi

- Stima delle frequenze p_k
 - migliore è la stima e maggiore è la compressione
 - ho bisogno di fare ipotesi vincolanti sulla sorgente oppure considerare un elevato numero di esempi

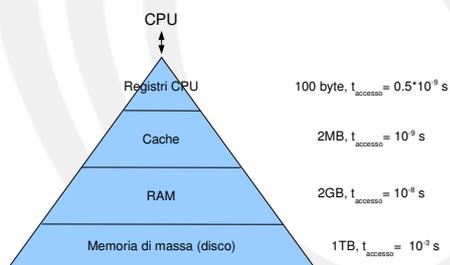
Considerazioni sulla complessità

In generale se si vuole comprimere di più una sequenza di simboli si devono usare **algoritmi più complessi** che impiegano più tempo oppure necessitano di una **CPU più potente** e comunque sempre portano ad un **maggiore consumo di energia**.



Memorizzazione su supporti di massa

Gerarchia di memoria



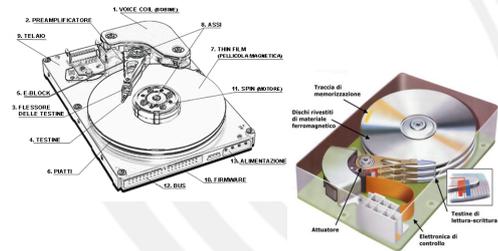
Prestazioni

- Capacità: $1 \text{ K} = 2^{10}$ $1 \text{ M} = 2^{20}$ $1 \text{ G} = 2^{30}$
- Tempo di accesso: tempo che intercorre tra la richiesta di un dato e l'arrivo del dato alla CPU

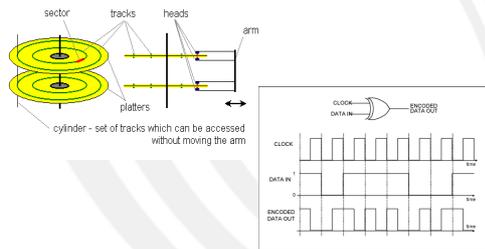
Memoria di massa

- Funzione:
 - ospitare dati e programmi in modo permanente
 - serbatoio per la memoria virtuale
- Tecnologie
 - disco magnetico: il più diffuso
 - disco ottico
 - memoria flash: si sta diffondendo

Disco magnetico



Principio di funzionamento



Organizzazione dei dati

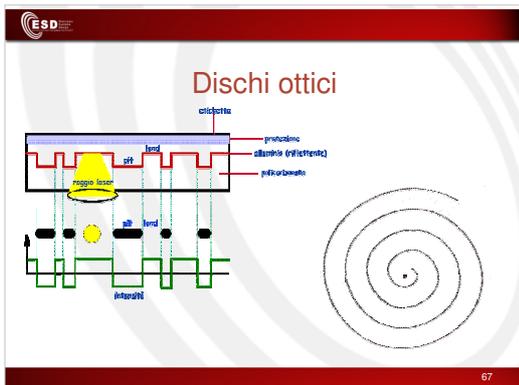
- Il settore è l'unità minima di lettura/scrittura
 - identificato univocamente da
 - numero di faccia
 - numero di traccia
 - numero di settore all'interno della traccia
 - contiene 512 o 1024 byte
- Il sistema operativo si occupa di creare la *metafora* del File System (file, cartelle, link) mentre comanda al disco lettura e scrittura a quantità intere di settori

Prestazioni

- Tempo di seek = tempo per posizionare il braccio sulla traccia (un media 5-10 ms)
- Latenza rotazionale = tempo di attesa prima che il settore scelto passi sotto le testine
 - In media $1/2 * 1/V_{rotazione}$
 - A 10000 giri/minuto $1/2 * 60/10000 = 3 \text{ ms}$
- Tempo di trasferimento del settore
 - Se la velocità di trasferimento è 20-40 MB/s per trasferire un settore di 512 byte ci vogliono $512/V_{transfer} = 12-24 \mu\text{s}$

Evoluzione dei dischi magnetici





- Dischi ottici**
- CD
 - 700 MB
 - DVD
 - Singolo lato, singolo strato (4.7 GB)
 - Singolo lato, doppio strato (8.5 GB)
 - Doppio lato, singolo strato (9.4 GB)
 - Doppio lato, doppio strato (17 GB)
 - Blu-ray
 - Minore lunghezza d'onda del laser e quindi maggiore densità
 - 25 GB per lato
- 68

- Memoria flash**
- Memoria a semiconduttore come la RAM
 - I bit vengono memorizzati sotto forma di carica elettrica in microscopici condensatori
 - Non perde l'informazione quando si toglie l'alimentazione
 - Usata per chiavette USB, memorie per fotocamere, dischi a stato solido, sistemi embedded
- 69

- Dischi a stato solido**
- Vantaggi
 - rumorosità assente
 - minore possibilità di rottura
 - minori consumi
 - tempo di accesso ridotto: decimi di ms
 - maggiore resistenza agli urti
 - minore produzione di calore
 - Svantaggi
 - un maggiore costo per bit rispetto a quelli magnetici
 - minore durata del disco a causa del limite di riscritture delle memorie flash.
- 70

- Redundant Array of Inexpensive/Independent Disks (RAID)**
- Motivazione
 - CPU veloce ma dispositivi di I/O lenti
 - Il costo dei dischi è basso
 - Si può aumentare la velocità di un sistema parallelizzando l'I/O
 - Si può aumentare l'affidabilità di un sistema introducendo ridondanza nella memorizzazione
 - Si può pensare di utilizzare in maniera coordinata un insieme di dischi fissi che però vengono visti dalla CPU come un unico disco
- 71

- RAID: architettura**
- Insieme dei dischi
 - Un controllore RAID viene installato tra la CPU e il sistema di dischi
 - Il controllore può essere implementato
 - in HW (maggiori prestazioni)
 - In SW come driver del sistema operativo
- 72

RAID: possibili configurazioni

- In gergo RAID le configurazioni sono chiamate *livelli*
- Un insieme di settori e' chiamato *stripe*
- Livello 0
 - Suddivisione delle stripe su piu' dischi in round-robin
- Livello 1
 - Suddivisione dell'insieme in due meta'
 - Ciascuna meta' e' gestita come Liv0
 - La seconda meta' e' una copia della prima (mirror)

73

RAID: possibili configurazioni (2)

- Livello 2
 - ogni byte di dato viene diviso in gruppi di 4 bit
 - vengono aggiunti k bit di controllo
 - 4 bit di dato + k bit di controllo vengono scritti su $4+k$ dischi, un bit per disco
- Livello 3
 - Come Liv2 ma con $k = 1$ (bit di parita')
- Le configurazioni 2 e 3 garantiscono alte prestazioni ma richiedono una sincronizzazione tra i dischi e quindi sono difficili da implementare

74

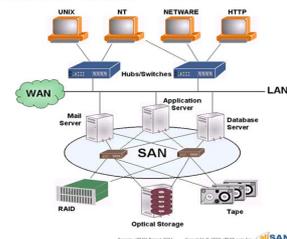
RAID: possibili configurazioni (3)

- Livello 4
 - insieme di M dischi gestito come Liv0 + un disco con una stripe di parità per M stripe
 - unisce il vantaggio della ridondanza al fatto di usare meno spazio per il mirroring
 - in caso di guasto ad un disco viene fatto accesso al disco di parità che diventa il collo di bottiglia
- Livello 5
 - come Liv4 ma stripe di dato e stripe di parità sono distribuite su tutti gli $M+1$ dischi per evitare il problema del collo di bottiglia

75

Storage Area Network (SAN)

Storage Area Networks



76

Concetto di file

- Il sistema operativo astrae dalle caratteristiche fisiche dei supporti di memorizzazione fornendo una visione logica
- File = spazio di indirizzamento logico e contiguo (insieme di informazioni correlate identificate da un nome)
 - Tipi di file
 - Dati
 - Numerici
 - Caratteri
 - Binari
 - Programmi

77

Attributi di un file

- Attributi sono memorizzati su disco nella struttura della directory
 - **Nome**
 - Unica informazione in formato "leggibile"
 - **Tipo**
 - **Posizione**
 - Puntatore allo spazio fisico sul dispositivo
 - **Dimensione**
 - **Protezione**
 - Controllo su chi può leggere, scrivere, eseguire
 - **Tempo, data e identificazione dell'utente**

78

Operazioni su file

- **Creazione**
 - Cercare spazio su disco, nuovo elemento su directory per attributi
- **Scrittura**
 - System call che specifica nome file e dati da scrivere
 - Necessario puntatore alla locazione della prossima scrittura
- **Letture**
 - System call che specifica nome file e dove mettere dati letti in memoria
 - Necessario puntatore alla locazione della prossima lettura (lo stesso della scrittura)
- **Riposizionamento all'interno di file**
 - Aggiornamento puntatore posizione corrente
- **Cancellazione**
 - Libera spazio associato al file e l'elemento corrispondente nella directory
- **Troncamento**
 - Mantiene inalterati gli attributi ma cancella contenuto del file

79

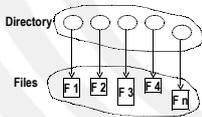
Operazioni su file (2)

- **Apertura**
 - Ricerca del file nella struttura della directory su disco, copia del file in memoria e inserimento di un riferimento nella tabella dei file aperti
 - In sistemi multiutente ci sono 2 tabelle, per gestire i file aperti
 - Una tabella per ogni processo contiene riferimenti per file aperti relativi al processo (es.: puntatore alla locazione di lettura/scrittura)
 - Una tabella per tutti i file aperti da tutti i processi contiene i dati indipendenti dal processo (es.: posizione sul disco, dimensione file, data accessi, n° di processi che hanno aperto il file)
- **Chiusura**
 - Copia del file in memoria su disco

80

Struttura delle directory

- Una collezione di nodi contenenti informazioni sui file
 - Entrambi residenti su disco
- Determinano la struttura del file system



81

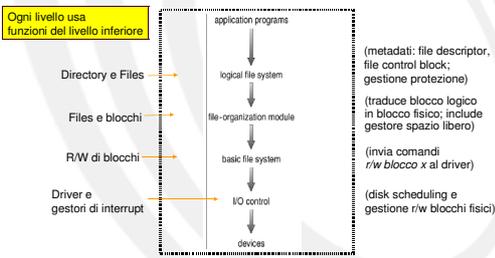
Introduzione

- Implementazione del file system
- Metodi allocazione dello spazio su disco
- Gestione dello spazio libero
- Efficienza e prestazioni
- Recupero delle informazioni

82

File System a livelli

Ogni livello usa funzioni del livello inferiore



83

Implementazione del file system

- Per gestire un file system si usano diverse strutture dati
 - Parte su disco
 - Parte in memoria
- Caratteristiche fortemente dipendenti dal sistema operativo e dal tipo di file system
 - Esistono caratteristiche comuni a tutte le tipologie

84

Implementazione del file system (2)

- **Blocco di boot**
 - Informazioni necessarie per l'avviamento del S.O.
- **Blocco di controllo delle partizioni**
 - Dettagli riguardanti la partizione
 - Numero e dimensione dei blocchi, lista blocchi liberi, lista descrittori liberi, ...
- **Strutture di directory**
 - Descrivono l'organizzazione dei file
- **Descrittori di file (inode)**
 - Vari dettagli sui file e puntatori ai blocchi dati

85

Implementazione del file system (3)

- **Tabella delle partizioni**
 - Informazioni sulle partizioni *montate*
- **Strutture di directory**
 - Copia in memoria delle directory a cui si è fatto accesso di recente
- **Tabella globale dei file aperti**
 - Copie dei descrittori di file
- **Tabella dei file aperti per ogni processo**
 - Puntatore alla tabella precedente ed informazioni di accesso

86

Allocazione dello spazio su disco

- Definisce come i blocchi su disco sono allocati ai file (o alle directory)
- Alternative
 - Allocazione contigua
 - Allocazione a lista concatenata (*linked*)
 - Allocazione indicizzata
- Obiettivi
 - Minimizzare tempi di accesso
 - Massimizzare utilizzo dello spazio

87

Allocazione contigua

- Ogni file occupa un insieme di blocchi contigui su disco
 - Entry della directory semplice
 - Contiene indirizzo blocco di partenza e lunghezza (numero di blocchi)
- Vantaggi
 - Accesso semplice
 - Accesso al blocco $b+1$ non richiede spostamento testina rispetto al blocco b
 - A meno che b non sia l'ultimo blocco di un cilindro (raro)
 - Supporta accesso sequenziale e casuale
 - Per leggere blocco i di un file che inizia al blocco b basta fare $b+i$

88

Allocazione contigua (2)

- Svantaggi
 - Allocazione contigua dei blocchi presenta problemi simili a quelli dell'allocazione dinamica della memoria
 - Algoritmi best-fit, first-fit, worst-fit
 - Spreco di spazio (frammentazione esterna)
 - Richiede compattazione periodica dello spazio
 - Quanto spazio allocare quando viene creato un file?
 - I file non possono crescere dinamicamente di dimensione
 - Soluzioni
 - Se il file deve crescere e non c'è spazio → errore e terminazione del programma → ri-esecuzione
 - » Si tende a sovrastimare lo spazio necessario (spreco)
 - Trovare un buco + grande e ricoprire tutto il file (compresa la parte nuova) in quest'ultimo
 - » Trasparente x l'utente, ma rallenta il sistema

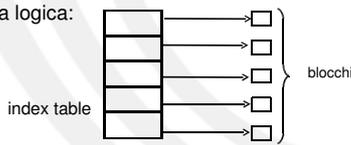
89

Allocazione a lista

- Ogni file è una **lista di blocchi**
- I blocchi possono essere sparsi ovunque nel disco
 - Directory contiene puntatori al primo e all'ultimo blocco
 - Ogni blocco contiene puntatore al blocco successivo
- Indirizzamento
 - X = indirizzo logico
 - N = dimensione del blocco
 - $X / (N-1)$ = numero del blocco nella lista
 - $X \% (N-1)$ = offset all'interno del blocco

90

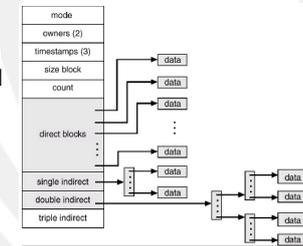
Allocazione indicizzata

- Ogni file ha un **blocco indice** (index block) contenente la **tabella degli indirizzi** (index table) dei blocchi fisici
- La directory contiene l'indirizzo del blocco indice
- Vista logica:
 

91

Allocazione indicizzata: UNIX

- esempio di struttura dati associata ad un file
- gestisce blocchi di 4KB



92

Gestione dello spazio libero

- Per tenere traccia dello spazio libero su disco si mantiene una lista dei blocchi liberi
 - Per creare un file si cercano blocchi liberi nella lista
 - Per rimuovere un file si aggiungono i suoi blocchi alla lista
- Alternative
 - Vettore di bit
 - Lista concatenata
 - Raggruppamento
 - Conteggio

93

Efficienza e prestazioni

- Disco è collo di bottiglia per le prestazioni
- Efficienza dipende da
 - Algoritmo di allocazione dello spazio su disco
 - Es.: in Unix si cerca di tenere i blocchi di un file vicini al suo i-node
 - Richiesta preallocazione i-node distribuiti sulla partizione
 - Tipo di dati contenuti nella directory
 - Es.: data di ultimo accesso di un file → lettura di un file richiede lettura e scrittura anche del blocco della directory

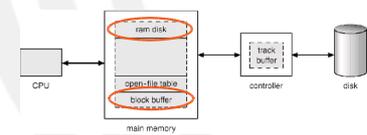
94

Efficienza e Prestazioni (2)

- Il controller del disco possiede una piccola cache che è in grado di contenere un'intera traccia ma...
- ... non basta per garantire prestazioni elevate, quindi
- Dischi virtuali (RAM disk)
- Cache del disco (detta anche buffer cache)

95

Efficienza e Prestazioni (3)



96

Dischi virtuali

- Parte della memoria gestita come se fosse un disco
- Il driver di un RAM disk accetta tutte le operazioni standard dei dischi eseguendole però in memoria
- Veloce
- Supporto solo per file temporanei, se spengo perdo tutto
- **Gestito dall'utente** che scrive sul RAM disk invece che sul disco vero e proprio (non è una cache!)

97

Cache del disco

- Porzione di memoria che memorizza blocchi usati di frequente
 - Simile alla cache tra memoria e CPU
 - Gestita dal S.O.
- Sfrutta principio della località
 - Spaziale
 - uso di dati "vicini" a quelli attualmente usati
 - Temporale
 - uso successivo degli stessi dati
- Trasferimento dati nella memoria del processo utente non richiede spostamento di byte

98

Cache del disco: problematiche

- Politica di rimpiazzamento
 - Cosa fare in caso di necessità di eliminare un settore?
 - LRU, LFU, RANDOM,
 - LRU poco efficiente per accesso sequenziale, meglio
 - rilascio indietro
 - lettura anticipata
- Politica di scrittura
 - Se l'operazione è una scrittura, come aggiornare il contenuto su disco?
 - Write-back: scrivo solo quando devo rimuovere il blocco dalla cache
 - Problemi di affidabilità in caso di crash
 - Write-through: scrivo sempre
 - Meno efficiente, la cache è solo in lettura

99

Problemi di file system in caso di guasto

- Crash del sistema: possibile inconsistenza tra disco e cache
- Guasto del disco: possibile inconsistenza tra i dati nella directory e i dati su disco relativi ai file
 - utilizzo di RAID
 - utilizzo di file system con log (journaled file system)
 - backup periodico del disco su memoria di massa ad alta capacità (nastri)
 - recupero di file persi tramite *restore* dei dati dai backup
 - si perde tutto ciò che è stato fatto dopo l'ultimo backup
 - utilizzo di tool di riparazione
 - scandisk (Windows), fsck (Unix)
 - potrebbe essere troppo tardi !

100

Journaled file system

- Registra ogni cambiamento del file system come una *transazione*
 - Tutte le transazioni sono scritte su un *log*
 - Una transazione è considerata avvenuta quando viene scritta sul log
 - Anche se il file system può non essere aggiornato
 - Le transazioni sul log sono scritte in modo asincrono nel file system
 - Quando il file system è modificato, la transazione viene cancellata dal log
 - Se il sistema va in crash, le transazioni non avvenute sono quelle presenti sul log
- Vantaggio
 - Tolleranza ai guasti di sistema
 - Ottimizzazione dell'accesso al disco
- Esempio: file system ext3 di Linux

101

Alcuni formati di memorizzazione

03/16/09

Formati per immagini

- Modi standard di memorizzare e trasmettere immagini digitali
 - Entertainment
 - Immagini mediche
 - Risultati di esperimenti
- Tali formati possono essere
 - basati su pixel o vettoriali
 - lossy o lossless

103

Formati basati su pixel (raster o bitmap)

- Si basano su una descrizione dell'immagine come matrice bidimensionale di valori di colori (picture element = pixel)
- Risoluzione (spaziale) = numero di righe x numero di colonne
- Profondità di colore = numero di bit per ciascun pixel per rappresentare il suo colore:
 - Spazi di colore
 - RGB
 - YUV, YCbCr

104

Formati vettoriali

- L'immagine è descritta come composizione di elementi grafici di base (primitive grafiche)
 - Segmenti, curve
 - Punti
 - Cerchi, ellissi
 - Poligoni
- Per ogni elemento è definita posizione e colore

105

Formati raster vs. vettoriali



Basato su pixel

Vettoriale

106

Formati lossless e lossy

- Spesso la rappresentazione dell'immagine prevede una fase di compressione
- Compressione lossless
 - la fase di compressione non pregiudica la qualità dell'immagine (risoluzione, colore)
 - utilizza tecniche di rimozione della ridondanza statistica e codifica entropica
 - usata per archiviazione e per applicazioni mediche
- Compressione lossy
 - la fase di compressione pregiudica irreversibilmente la qualità dell'immagine (risoluzione, colore)
 - utilizza anche tecniche di rimozione di dettagli poco percepibili
 - usata per applicazione di entertainment
- A parità di immagine di partenza, le tecniche lossy comprimono di più di quelle lossless

107

Formati più comuni

- Raster
 - JPEG
 - TIFF
 - GIF
 - PNG
- Vettoriale
 - SVG
 - Postscript
 - PDF
 - SWF
 - Windows Metafile

108

Joint Photographic Experts Group (JPEG)

- Tecnica lossy basata sulla trasformata coseno discreta (DCT) per separare le frequenze spaziali
 - eliminazione della ridondanza statistica
 - eliminazione delle frequenze più alte
- Profondità di colore: YUV 8 bit per componente
- Se la qualità è scadente compare il fenomeno della blockingness (blocchettizzazione)
- Utilizzato in fotocamere digitali e per codificare le immagini nei documenti pdf

109

Tagged Image File Format (TIFF)

- Permette sia compressione lossless sia lossy
- Profondità di colore: 8 o 16 bit per componente
 - Scale di grigio
 - RGB
 - CMYK
 - CIE Lab
- Utilizzato
 - dagli scanner (es. fotografie di microarray)
 - frequentemente nel mondo tipografico
 - da alcune fotocamere per evitare il degrado del jpeg

110

Graphics Interchange Format (GIF)

- Lossless
- Palette indicizzata a 8 bit (256 colori)
 - In caso di sfumature si usa il dithering
- Formato adatto a memorizzare immagini sintetiche con larghe zone di colore uniforme come grafici, loghi, ecc...
- Memorizza più immagini permettendo animazioni

111

Portable Network Graphics (PNG)

- Nato da progetto open source come successore del GIF
- Lossless e lossy
- Profondità di colore
 - Indicizzata
 - Livelli di grigio
 - Colore a 24 bit (16 milioni di colori)
- I pixel sono memorizzati in modo da permettere una veloce preview senza aspettare di aver scaricato tutto il file

112

Scalable Vector Graphics (SVG)

- Formato vettoriale
- Sintassi XML eventualmente compressa con tecniche usate per file testuali (zip, gzip)
- Standard del World Wide Web Consortium

113

Formati per applicazioni mediche

- Health Level 7
- Clinical Document Architecture (CDA)
- CDISC
- DICOM

114

Health Level 7 (HL7)

- Formato per scambiare, memorizzare, condividere e recuperare informazioni mediche
- Alla base della fornitura, gestione e valutazione di servizi medici
- Il numero 7 ricorda che HL7 si pone a livello 7 della pila ISO/OSI

Ambito di impiego di HL7

- Gli ospedali hanno diversi sistemi automatici (hospital information systems)
 - Accettazione pazienti
 - Cartelle cliniche
 - Analisi di laboratorio
- HL7 fornisce interfacce standard, tipi di messaggi e linee guida per far parlare tra loro questi sistemi
- Reference Information Model (RIM) è una parte fondamentale di HL7 e uno standard ISO
 - Esprime il contenuto informativo dei processi ospedalieri
 - Connessioni semantiche e lessicali tra i messaggi previsti da HL7