

Corso di Informatica

Modulo T1

C1-Organizzazione sequenziale con indici

M. Malatesta C1-Organizzazione sequenziale con indici-17

1
25/11/2011

Prerequisiti

E' necessario conoscere:

- organizzazione sequenziale
- organizzazione relativa
- struttura dati tabella
- chiave primaria
- accesso diretto
- cenni sulla struttura di un hard-disk

M. Malatesta C1-Organizzazione sequenziale con indici-17

2
25/11/2011

Informazioni generali

In questa Unità si discute un nuovo tipo di organizzazione sequenziale, detto **organizzazione sequenziale con indici**. Questo tipo di organizzazione è un'evoluzione dell'organizzazione sequenziale e presenta, come vedremo, vantaggi e svantaggi.

N.B. Nel descrivere gli algoritmi di trattamento dei file, indichiamo con:

Tipo r;

un generico record (un oggetto di classe **Tipo**) e con l'intero k il suo attributo usato come chiave primaria, tralasciando per semplicità eventuali restanti attributi.

M. Malatesta C1-Organizzazione sequenziale con indici-17

3
25/11/2011

Introduzione

Dovendo gestire un elenco di dati ad esempio anagrafici, si potrebbe pensare allo schema seguente, basato su due file:

- si accede al primo file, fornendo il valore di una chiave alfanumerica, *chiave primaria*, ad es. "Assini";
- alla chiave trovata è associata la posizione logica del record cercato. Accedendo al file di dati a detta posizione, otteniamo i dati completi.

Chiave	Posizione
Assini	4
Botti	1
Ceri	3
Giusti	2

Cognome	Nome	Indirizzo	Provincia	Saldo
Botti	Anna	Napoli	NA	2000
Giusti	Ennio	Roma	RM	1450
Ceri	Luigi	Perugia	PG	1100
Assini	Carlo	Pisa	PI	800

M. Malatesta C1-Organizzazione sequenziale con indici-17

4
25/11/2011

Organizzazione *indexed*

L'organizzazione sequenziale con indice (o **index sequential o indexed file**) prevede due file:

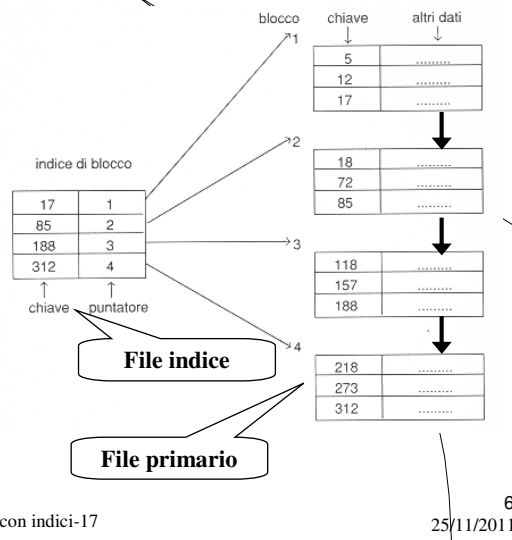
- **file di indici (index file, dizionario)**, nel quale:
 - sono presenti tutte le chiavi, ordinate (in genere, in ordine crescente);
 - ogni chiave è associata alla posizione logica del record corrispondente
 - è possibile accesso diretto
- **file primario (work file)**, in cui sono presenti i dati effettivi e completi:
 - organizzazione sequenziale o relativa
 - con accesso diretto
 - a **blocchi** (pagine)
 - ordinati in base alla chiave

Organizzazione *indexed*

In generale, considerando le chiavi come numeri, un file con organizzazione ad indici può essere rappresentato come indicato.

Il file indice è ordinato in base alle chiavi e ogni record contiene:

- la *chiave massima* presente nel blocco;
- l'*indice del blocco* associato.



Tipi di accesso

Nei file organizzati ad indici possiamo avere:

- **accesso sequenziale:** è sempre possibile la scansione sequenziale dei record, come in tutte le altre organizzazioni. La freccia che unisce i blocchi del file primario nella figura precedente, indica appunto che è possibile visitare i record uno dopo l'altro con chiave logica crescente (ad es. in ordine alfabetico).
- **accesso a chiave:** avviene secondo la procedura seguente:
 - si immette la chiave k
 - si ricerca k nel dizionario (ad esempio con una ricerca binaria)
 - si preleva la posizione associata p
 - si preleva il record in posizione p del *work file* mediante accesso diretto.

File *multi-index*

L'organizzazione *index sequential* è nata con l'introduzione delle unità a disco (anni 60).

Le prime realizzazioni utilizzavano un *work file*, ordinato fisicamente sulla chiave primaria, e permettevano di associarlo ad un unico *index file*.

Le realizzazioni più moderne utilizzano un *work file* non ordinato su alcun campo e permettono per uno stesso *work file*, che sia costruito più di un *index file* in modo che sia possibile accedere attraverso più chiavi logiche (**multi-Index file**).

Si può utilizzare una chiave:

- **primaria**, quando occorre individuare uno ed un solo record (univoca)
- **secondaria** (o alternativa), anche non univoca, per individuare più di un record, procedendo sequenzialmente dal primo in poi.

File multi-index

Chiave	Posizione
Assi	4
Bott	1
Ceri	3
Gius	2

Metodo a **chiave parziale**

In alcuni casi, per ottimizzare tempi e occupazione di memoria si può:

- usare una **chiave parziale**
- mantenere contenute le dimensioni del file indice, così da registrarlo in memoria centrale sotto forma di:
 - tabella (garantisce velocità, come la ricerca binaria)
 - lista (garantisce ottimizzazione dello spazio di memoria)

Provincia	Posizione
RM	2
NA	1
PG	3
PI	4

Cognome	Nome	Indirizzo	Provincia	Saldo
Botticelli	Anna	Napoli	NA	2000
Giustini	Ennio	Roma	RM	1450
Cerini	Luigi	Perugia	PG	1100
Assini	Carlo	Pisa	PI	800

M. Malatesta C1-Organizzazione sequenziale con indici-17

9
25/11/2011

Operazioni su file indexed

Le operazioni sono quelle consuete:

- **ricerca di un record di chiave k**
- **aggiornamento in base al valore di k**

Supponiamo di operare con un'organizzazione con indici a più livelli.

M. Malatesta C1-Organizzazione sequenziale con indici-17

10
25/11/2011

Operazioni su file indexed

- ricerca su file indexed

La ricerca di un record di chiave k procede secondo il seguente algoritmo:

Tipo Ricerca (Intero k)

Flusso work-f, index-f;

Inizio

Ricerca nel file index-f la prima chiave $k1 \geq k$;

Acquisisci l'indirizzo p della pagina associata a $k1$;

Accedi nel file primario alla pagina p ;

Ricerca nella pagina p la chiave k ;

Se (record presente) **Ritorna** r ;

Altrimenti Ritorna null;

Fine

M. Malatesta C1-Organizzazione sequenziale con indici-17

11
25/11/2011

Operazioni su file indexed

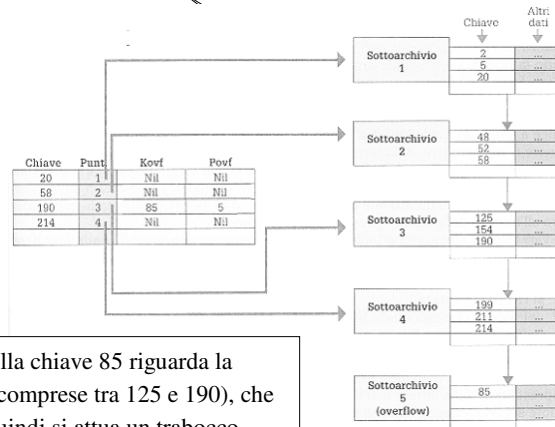
- inserimento su file indexed

L'inserimento di un record

di chiave k si intuisce che espone al rischio di dover riorganizzare l'archivio.

Tuttavia, trattandosi di un archivio sequenziale, si può con vantaggio usare la tecnica delle **aree di overflow**.

L'inserimento della chiave 85 riguarda la pagina 3 (chiavi comprese tra 125 e 190), che risulta **saturo**. Quindi si attua un trabocco sulla pagina 5.



M. Malatesta C1-Organizzazione sequenziale con indici-17

12
25/11/2011

Operazioni su file indexed

- inserimento su file indexed

void Inserisci (**Tipo** r)

Inizio

Ricerca nel file indice la prima chiave $k_i \geq r.k$;

Se ($r.k > k_i$) *Inserisci ordinatamente r nell'area di overflow*

Altrimenti **Inizio** *Inserisci ordinatamente r nella pagina*

Se (*c'è trabocco*)

Inizio

Inserisci r in testa all'area di overflow

Se (*è il primo trabocco*)

Aggiorna Kovf, Povf, Chiave

Altrimenti *Aggiorna Chiave*

Fine

Fine

Fine

L'algoritmo per l'inserimento si presenta come segue:

Operazioni logiche

Le **operazioni logiche** principali per l'**organizzazione ad indici** sono le stesse dell'organizzazione sequenziale con, in aggiunta, le seguenti:

Operazione	Effetto
Leggi (Flusso f , Tipo rec , Stringa k)	Legge record rec con chiave k
Scrivi (Flusso f , Tipo rec , Stringa k)	Scrive record rec con chiave k
Cancella (Flusso f , Stringa k)	Cancella record di chiave k

Indici a più livelli

Quando il numero di blocchi diventa molto grande, il file indice cresce e, di conseguenza, la sua gestione può diventare pesante.

Si può allora pensare il file indice stesso con un'organizzazione ad indici, per riportare i tempi di elaborazione a valori accettabili.

L'**organizzazione a indici a più livelli** è un'evoluzione della gestione *indexed* descritta in precedenza.

In questa tecnica si struttura il file ad indici in più livelli di indici, in modo che ciascun livello contenga un gruppo di chiavi (ordinate) di cui è nota quella di valore massimo. In questo modo, data una chiave *k*, è possibile risalire al gruppo che la contiene; successivamente, si accede al file dati all'indirizzo corrispondente alla chiave *k*.

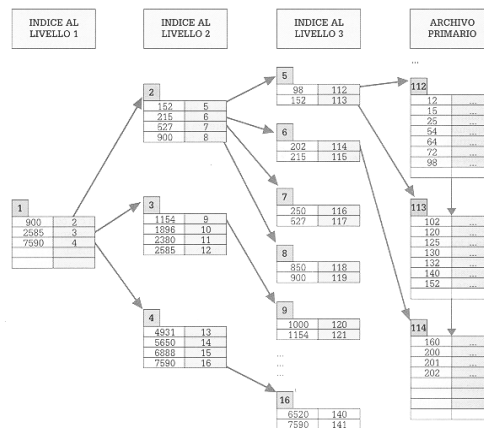
M. Malatesta C1-Organizzazione sequenziale con indici-17

15
25/11/2011

Indici a più livelli - ricerca di un record

Ad esempio, per cercare la chiave 200:

- si accede all'indice 1 a livello 1 e si trova 900 (indice 2)
- si accede all'indice 2 di livello 2 e si trova 215 (indice 6)
- si accede all'indice 6 di livello 3 e si trova 202 (indice 114)
- si accede al file primario alla pagina 114 e scorrendo sequenzialmente, si trova 200



M. Malatesta C1-Organizzazione sequenziale con indici-17

16
25/11/2011

Indici a più livelli - ricerca di un record

In generale, l'algoritmo per la ricerca di un record di chiave k in un file *indexed sequential a n livelli*, può essere descritto come segue:

1. cercare nell'indice di 1° livello la prima chiave di valore $\geq k$
2. accedere all'indice di 2° livello fino alla prima chiave $\geq k$
3. ...
- n. accedere all'indice di n° livello fino alla prima chiave $\geq k$
Accedere al file primario e, scorrendolo sequenzialmente, prelevare, se presente, l'informazione associata.

Indici a più livelli - inserimento di un record

In generale, l'inserimento su *file ordinati* a più livelli, prevede la gestione di aree di *overflow*, come negli archivi sequenziali.

Nel caso di *file non ordinati*, l'inserimento può essere fatto in coda o in posizioni cancellate in modo logico.

Applicazioni

L'organizzazione ad indici a 3 livelli, in particolare, è utilizzata nella progettazione dei file system, allo scopo di ottimizzare occupazione e tempi di ricerca delle informazioni.

I tre indici sono:

- **Indice di unità**, determina l'unità di memoria di massa su cui cercare l'informazione (se questa è suddivisa su più unità)
- **Indice di cilindro**, determina il numero di cilindro sui cui si trova l'informazione
- **Indice di traccia**, determina il numero di traccia sui cui si trova l'informazione (le tracce sono le pagine del file primario)

Valutazioni di efficienza

Possiamo osservare i seguenti vantaggi:

- questa tecnica, scomponendo il file ad indici in pagine, fa sì che il singolo gruppo di indici, di dimensioni più ridotte rispetto all'intero file ad indici, possa essere caricato in memoria centrale;
- la presenza di più livelli di indici riduce lo spazio di ricerca per gli indici.
- è comunque sempre possibile la scansione sequenziale dei record

L'organizzazione a più livelli di indici è consigliata quando l'archivio è **dinamico** (subisce frequenti aggiornamenti) poiché, pur prevedendo una notevole occupazione di memoria per gli indici, è sufficiente mantenere l'ordinamento degli indici in caso di inserimento o cancellazione.

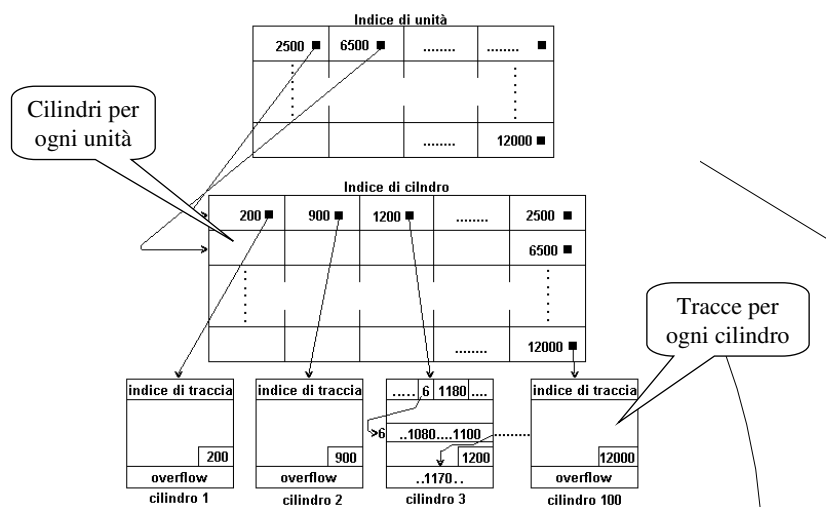
Tecniche di indicizzazione

Esistono vari metodi di gestione del file indice, più o meno potenti, più o meno flessibili, più o meno veloci.

Sostanzialmente possiamo avere:

- **tecnica ISAM** (*Indexed Sequential Access Method*), sviluppata originariamente da IBM, costituisce oggi la base per l'immagazzinamento dei dati in molti database, relazionali e non.
- **tecnica B-Tree** (basata sugli **ABR** o *Alberi Binari di Ricerca*, quali B-Albero, B*-Albero, B+_Albero), più attuale ed efficiente nella costruzioni degli indici (descritta in altre Unità).

Tecnica ISAM



Tecnica ISAM

Nel metodo **ISAM** i record e gli indici sono organizzati in maniera tale da riempire perfettamente le tracce dei cilindri e da minimizzare gli spostamenti delle testine di lettura/scrittura da un cilindro all'altro.

Le principali caratteristiche dell'ISAM sono:

- si hanno tre livelli di indici;
- gli indici di livello più basso (gli indici di traccia) contengono il valore più grande delle chiavi in quella traccia;
- ogni cilindro contiene il valore più grande delle chiavi in quel cilindro;
- l'indice di cilindro è suddiviso in tracce;
- l'indice di unità è allocato permanentemente in memoria centrale;

Tecnica ISAM

- la prima traccia di ogni cilindro contiene l'indice di traccia per quel cilindro,
- l'ultima traccia di ogni cilindro contiene un'area di overflow comune alle tracce di quel cilindro;
- ad ogni area di overflow di cilindro può essere associata un'area di overflow indipendente su un altro cilindro allo scopo di ovviare agli inconvenienti dovuti ad un veloce riempimento della traccia di overflow di cilindro;
- è una tecnica che prevede che il file primario sia ordinato logicamente in base alla chiave

Ricerca su archivi ISAM

La ricerca in un archivio ISAM procede in questo modo:

- ricerca sull'Indice dei Cilindri del cilindro interessato
- movimento delle testine per il posizionamento sul cilindro
- lettura prima traccia per portare in memoria l'Indice di Traccia
- movimento delle testine per trovare la traccia interessata
- lettura della traccia interessata senz'altro spostamento delle testine
- individuazione in memoria del record interessato

Questo modo di organizzare i dati permetteva, con un solo movimento delle testine, di accedere a qualsiasi record e quindi era particolarmente efficiente.

M. Malatesta C1-Organizzazione sequenziale con indici-17

25
25/11/2011

Aggiornamenti su archivi ISAM

I problemi nascevano per gli inserimenti e le cancellazioni.

Per evitare di spostare i record in avanti (costo eccessivo) si adottavano diverse soluzioni:

- al momento della creazione del file, riempire le tracce solo al 70%.
- riservare le ultime due tracce di ogni cilindro come **Aree di Overflow Primario** in cui venivano registrati come lista ordinata i record. Nell'Indice di traccia era registrata:
 - la chiave massima contenuta in ogni traccia
 - la chiave più alta registrata nella catena di Overflow Primario per quella traccia.

In questo modo era possibile capire, senza altri movimenti della testina, se il record richiesto era nella traccia o in overflow.

M. Malatesta C1-Organizzazione sequenziale con indici-17

26
25/11/2011

Aggiornamenti su archivi ISAM

Quando anche l'Area di Overflow Primario si riempiva, si avevano due alternative:

- riorganizzare il file (copia su file sequenziale, cancellazione vecchio file e nuova creazione), cosa che richiedeva troppo tempo e impediva di usare nel frattempo il file e che veniva quindi svolta di notte.
- usare un'area detta **Area di Overflow Secondario** dove potevano espandersi le varie catene dell'Overflow Primario. Quest'ultima possibilità faceva abbassare notevolmente le prestazioni del metodo, poiché per la lettura ed anche per l'inserimento dei record in overflow, un unico movimento delle testine non era più sufficiente.

Argomenti

- Introduzione
- Organizzazione *indexed*
- Tipi di accesso
- File *multi-index*
- Operazioni su file indexed
 - ricerca su file indexed
 - inserimento su file indexed
- Operazioni logiche
- Indici a più livelli
 - ricerca di un record
 - inserimento di un record
- Applicazioni
- Valutazioni di efficienza
- Tecniche di indicizzazione
- Tecnica ISAM
- Ricerca su archivi ISAM
- Aggiornamenti su archivi ISAM

Altre fonti di informazione

- A.Lorenzi-D.Rossi, Le basi di dati e il linguaggio SQL – ed. ATLAS
- G.Callegarin – Corso di Informatica Generale 3 – CEDAM
- P.Gallo, F.Salerno – Informatica Generale - ed. Minerva Italica
- Cesarini, Pippolini, Soda – Informatica 3 – ed. CEDAM