



Sistemi Operativi

Esercizi File System

Docente: Claudio E. Palazzi
cpalazzi@math.unipd.it

Esercizio Blocchi Liberi - Bitmap

Quesito 3. Un *file system* utilizza un vettore di *bit* (*bitmap*), con posizioni numerate da sinistra verso destra, per indicare i blocchi liberi presenti in una data partizione di disco. La formattazione della partizione libera tutti i blocchi tranne il 1°, che viene assegnato alla *directory* radice. In tale sistema, l'assegnazione di blocchi liberi a *file* considera sempre prima i blocchi di indice minore, trascurando ogni considerazione di contiguità. Si mostri il contenuto della parte iniziale di tale vettore dopo ciascuna delle seguenti operazioni:

- Scrittura del *file* A, ampio 6 blocchi
- Scrittura del *file* B, ampio 5 blocchi
- Rimozione del *file* A
- Scrittura del *file* C, ampio 8 blocchi
- Rimozione del *file* B

Si mostri poi l'evoluzione del contenuto di tale vettore a fronte della medesima sequenza di operazioni nel caso in cui il sistema volesse assicurare la massima contiguità dei blocchi assegnati ad ogni *file*.

Soluzione

Soluzione 3 (punti 5). Seguiamo l'evoluzione del contenuto della maschera nel primo caso, concentrandoci sulle sue prime posizioni:

Dopo la formattazione:	10000000000000 ...
Scrittura del file A:	11111100000000 ...
Scrittura del file B:	1111111111000 ...
Rimozione del file A:	1000001111000 ...
Scrittura del file C:	1111111111110 ...
Rimozione del file B:	11111100000110 ...

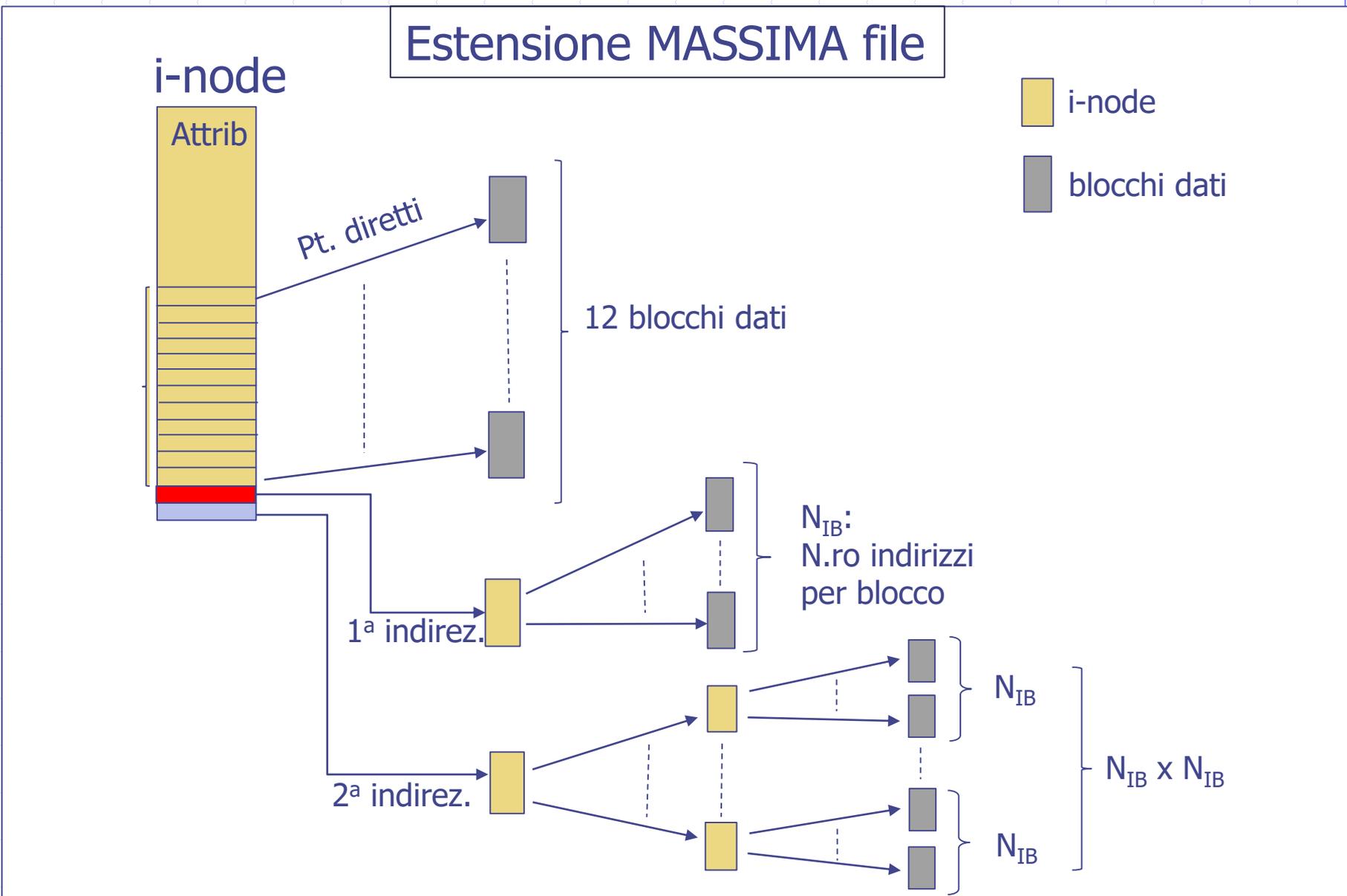
Vediamone ora l'evoluzione nel secondo caso, nel quale prevalgono considerazioni di contiguità dei blocchi assegnati ai file:

Dopo la formattazione:	10000000000000 ...	
Scrittura del file A:	11111100000000 ...	allocazione contigua senza frammentazione esterna
Scrittura del file B:	1111111111000 ...	allocazione contigua senza frammentazione esterna
Rimozione del file A:	1000001111000 ...	
Scrittura del file C:	1000001111111110 ...	allocazione contigua con frammentazione esterna
Rimozione del file B:	100000000001111110 ...	

Esercizio i-node

Quesito 4. Il progettista di un sistema operativo ha deciso di usare nodi indice (*i-node*) per la realizzazione del proprio *file system*, stabilendo che essi abbiano la stessa dimensione di un blocco, fissata a 512 *byte*. Il progettista ha poi deciso che un nodo indice primario contenga 12 campi di indirizzo di blocchi di disco e 2 campi puntatori a nodi indice di primo e secondo livello di indirizzazione rispettivamente. Sapendo che gli indirizzi di blocco sono espressi su 32 *bit*, si vuole allocare un *file* logicamente composto da 10.000 *record* da 80 *byte* ciascuno, imponendo che un *record* non possa essere suddiviso su due blocchi. Calcolare quanti blocchi verranno utilizzati per allocare il *file* dati e quanti per gestire la sua allocazione tramite nodi indice. Determinare inoltre l'occupazione totale in memoria secondaria risultante da tale strategia di allocazione.

Soluzione



Soluzione

Soluzione 4 (punti 5). Richiamiamo i dati del problema:

N_R = numero di *record* che compongono il *file* = 10.000

D_R = dimensione di un *record* = 80 *byte*

D_I = dimensione di un indirizzo = 4 *byte*

D_B = dimensione di un blocco = 512 *byte*

N_{RB} = numero di *record* per blocco = $\text{int}(D_B/D_R) = \text{int}(512/80) = \text{int}(6,4) = 6$

N_{BF} = numero di blocchi occupati dal *file* = $1 + \text{int}(N_R/N_{RB}) = 1 + \text{int}(10000/6) = 1.667$

N_{IB} = numero di indirizzi in un blocco = $D_B/D_I = 512/4 = 128$

N_{ID} = numero di indirizzi in X blocchi a doppia indirazione = X^2

I blocchi da indirizzare sono $N_{BF} = 1.667$

- di questi, 12 possono essere indirizzati direttamente dal nodo indice principale
- dei rimanenti $1.667 - 12 = 1.655$, N_{IB} (cioè 128) sono indirizzabili ad indirazione singola tramite l'indirizzo ad indirazione singola del nodo indice principale
- dei rimanenti $1.655 - 128 = 1.527$, si possono utilizzare blocchi indiretti di 128 con indirazione doppia, come mostrato in figura.

$$1 + \text{int}(1527/128) = 12 \text{ blocchi}$$

Con le indicazioni riportate in figura possiamo concludere che:

- per allocare il *file* dati sono necessari NBF blocchi, cioè 1.667 blocchi
- per gestire l'allocazione del *file* sono necessari:
 - 1 blocco per il nodo indice principale
 - 1 blocco di indirizzi ad indirazione singola
 - $1 + 12$ blocchi per l'indirazione doppia

per un totale di $1 + 1 + 1 + 12 = 15$ blocchi

- l'occupazione totale in memoria secondaria vale $1.667 + 15 = 1.682$ blocchi da 512 *byte*, per un totale di $1.682 \cdot 512 = 861.184 = \text{byte} = 841 \text{ kB}$

Quesito

Si consideri un *file system* residente su una partizione di disco con dimensione dei blocchi logici e fisici di 512 B, dimensione dei *file* non superiori a 512 blocchi, e con tutte le informazioni su ciascun *file* già presenti in memoria principale.

Per ciascuno dei tre metodi di allocazione visti a lezione (**contigua, concatenata, indicizzata**):

1. si illustri come gli indirizzi logici vengono fatti corrispondere agli indirizzi fisici
2. assumendo che l'ultimo accesso sia stato fatto al blocco logico 10, si determini quanti blocchi fisici debbano essere letti dal disco per accedere al blocco logico 4.

Soluzione 1/3

Allocazione contigua: il *file* è denotato dall'indice del primo blocco fisico e dalla sua ampiezza in blocchi; vista la corrispondenza di ampiezza tra blocchi logici e fisici, ogni posizione interna al *file* (blocco logico e *offset* in esso) ha una corrispondenza diretta sul disco (blocco fisico e *offset*).

Allocazione concatenata: il *file* è denotato dagli indici del primo e dell'ultimo blocco fisico; una parte dei dati di ogni blocco contiene il puntatore al blocco successivo. La posizione interna al *file* espressa in (blocco logico i , *offset* 0) viene dunque tradotta mediante l'attraversamento di i posizioni nella lista concatenata a partire dalla testa.

Soluzione 2/3

Allocazione indicizzata: il *file* è denotato da un blocco speciale (detto appunto "indice"), che contiene gli indici dei blocchi fisici ove risiedono i dati. La posizione interna al *file* espressa in (blocco logico i , *offset* o) viene dunque tradotta localizzando il blocco fisico denotato dalla posizione i entro il blocco indice e la posizione o al suo interno. (Come noto, il blocco indice può essere realizzato come una tabella concatenata, tipo FAT, oppure come un blocco contiguo dedicato, tipo *i-node*.)

Soluzione 3/3

Blocchi fisici acceduti per procedere dal blocco 10 al blocco 4 :

Allocazione contigua: 1 (direttamente il blocco 4).

Allocazione concatenata: 4 (fino al blocco 4 a partire dalla testa della lista).

Allocazione indicizzata: 1 (direttamente il blocco 4, ma solo in virtù dell'ipotesi favorevole del quesito per la quale la dimensione massima del *file* sia interamente rappresentabile con un singolo blocco indice).