

Sistemi Operativi

ESERCITAZIONE

GESTIONE DEL FILE SYSTEM

Esercizio 1 (vale 6 punti)

Si consideri un file formato da 100 blocchi. Si assuma che il file control block (ed il blocco indice, nel caso di allocazione indicizzata) sia già in memoria principale. Calcolare quante operazioni di accesso al disco sono necessarie nel caso di allocazione contigua, concatenata, e indicizzata (su un solo livello) in ciascuna delle seguenti situazioni.

- a) Un blocco è aggiunto all'inizio.
- b) Un blocco è aggiunto in 51-esima posizione.
- c) Un blocco è aggiunto alla fine.
- d) Un blocco è rimosso dall'inizio.
- e) Un blocco è rimosso dalla 50-esima posizione.
- f) Un blocco è rimosso dalla fine.

Nel caso di allocazione contigua si assuma che non ci sia spazio per far crescere l'area assegnata al file verso l'inizio, ma che ce ne sia per farlo crescere verso la fine. Nel caso di aggiunta di un blocco, si assuma che il blocco di informazioni da aggiungere sia in memoria principale.

Motivare brevemente le risposte.

Soluzione

	Contigua	Concatenata	Indicizzata
a			
b			
c			
d			
e			
f			

Soluzione: allocazione contigua

	Contigua	Concatenata	Indicizzata
a			
b			
c			
d			
e			
f			

- Per posizionare correttamente un blocco in una posizione che non sia l'ultima, occorre spostare tutti i blocchi che dovranno seguirlo (a partire dall'ultimo), ed ogni spostamento richiede un'operazione di lettura ed una di scrittura
- C'è poi da considerare la scrittura del blocco stesso

Soluzione: allocazione contigua

	Contigua	Concatenata	Indicizzata
a	201		
b	101		
c	1		
d			
e			
f			

- Per posizionare correttamente un blocco in una posizione che non sia l'ultima, occorre spostare tutti i blocchi che dovranno seguirlo (a partire dall'ultimo), ed ogni spostamento richiede un'operazione di lettura ed una di scrittura
- C'è poi da considerare la scrittura del blocco stesso

Soluzione: allocazione contigua

	Contigua	Concatenata	Indicizzata
a	201		
b	101		
c	1		
d			
e			
f			

- Per posizionare correttamente un blocco in una posizione che non sia l'ultima, occorre spostare tutti i blocchi che dovranno seguirlo (a partire dall'ultimo), ed ogni spostamento richiede un'operazione di lettura ed una di scrittura
- C'è poi da considerare la scrittura del blocco stesso
- Operazioni simili vanno fatte per mantenere la contiguità anche quando un blocco viene rimosso, a meno che non sia l'ultimo o il primo (in tali casi si agisce solo sul FCB in memoria principale ed il costo è 0)

Soluzione: allocazione contigua

	Contigua	Concatenata	Indicizzata
a	201		
b	101		
c	1		
d	0		
e	98		
f	0		

- Per posizionare correttamente un blocco in una posizione che non sia l'ultima, occorre spostare tutti i blocchi che dovranno seguirlo (a partire dall'ultimo), ed ogni spostamento richiede un'operazione di lettura ed una di scrittura
- C'è poi da considerare la scrittura del blocco stesso
- Operazioni simili vanno fatte per mantenere la contiguità anche quando un blocco viene rimosso, a meno che non sia l'ultimo o il primo (in tali casi si agisce solo sul FCB in memoria principale ed il costo è 0)

Soluzione: **allocazione concatenata**

	Contigua	Concatenata	Indicizzata
a	201		
b	101		
c	1		
d	0		
e	98		
f	0		

- L'inserimento di un blocco richiede la scrittura del blocco da inserire e, se non è il primo, l'aggiornamento del puntatore del blocco che lo precede immediatamente per cui è necessario l'accesso sequenziale a tutti i blocchi che lo precedono

Soluzione: **allocazione concatenata**

	Contigua	Concatenata	Indicizzata
a	201	1	
b	101	52	
c	1	102	
d	0		
e	98		
f	0		

- L'inserimento di un blocco richiede la scrittura del blocco da inserire e, se non è il primo, l'aggiornamento del puntatore del blocco che lo precede immediatamente per cui è necessario l'accesso sequenziale a tutti i blocchi che lo precedono

Soluzione: **allocazione concatenata**

	Contigua	Concatenata	Indicizzata
a	201	1	
b	101	52	
c	1	102	
d	0		
e	98		
f	0		

- L'inserimento di un blocco richiede la scrittura del blocco da inserire e, se non è il primo, l'aggiornamento del puntatore del blocco che lo precede immediatamente per cui è necessario l'accesso sequenziale a tutti i blocchi che lo precedono
- L'eliminazione di un blocco, comporta l'accesso a tutti quelli che lo precedono (a meno che non sia il primo) ed al blocco stesso (a meno che non sia l'ultimo), per leggere il puntatore al blocco successivo, e la modifica del puntatore contenuto nel blocco che lo precede (a meno che non sia il primo)

Soluzione: allocazione concatenata

	Contigua	Concatenata	Indicizzata
a	201	1	
b	101	52	
c	1	102	
d	0	1	
e	98	51	
f	0	100	

- L'inserimento di un blocco richiede la scrittura del blocco da inserire e, se non è il primo, l'aggiornamento del puntatore del blocco che lo precede immediatamente per cui è necessario l'accesso sequenziale a tutti i blocchi che lo precedono
- L'eliminazione di un blocco, comporta l'accesso a tutti quelli che lo precedono (a meno che non sia il primo) ed al blocco stesso (a meno che non sia l'ultimo), per leggere il puntatore al blocco successivo, e la modifica del puntatore contenuto nel blocco che lo precede (a meno che non sia il primo)

Soluzione: **allocazione indicizzata**

	Contigua	Concatenata	Indicizzata
a	201	1	
b	101	52	
c	1	102	
d	0	1	
e	98	51	
f	0	100	

- Il costo dell'inserimento è dato dalla scrittura del blocco su disco (lettura del blocco ed aggiornamento, cioè lettura e scrittura, dell'indice costano 0, dato che entrambi i blocchi sono in memoria principale)
- L'eliminazione comporta il solo aggiornamento dell'indice, per cui ha costo 0

Soluzione: **allocazione indicizzata**

	Contigua	Concatenata	Indicizzata
a	201	1	1
b	101	52	1
c	1	102	1
d	0	1	0
e	98	51	0
f	0	100	0

- Il costo dell'inserimento è dato dalla scrittura del blocco su disco (lettura del blocco ed aggiornamento, cioè lettura e scrittura, dell'indice costano 0, dato che entrambi i blocchi sono in memoria principale)
- L'eliminazione comporta il solo aggiornamento dell'indice, per cui ha costo 0
- N.B. Inserimento e cancellazione hanno costo costante indipendentemente dalla posizione del blocco su cui si opera

Soluzione

	Contigua	Concatenata	Indicizzata
a	201	1	1
b	101	52	1
c	1	102	1
d	0	1	0
e	98	51	0
f	0	100	0

Esercizio 2 (vale 3 punti)

Dato il seguente frammento
di **tabella di directory**

Nome	Estensione	Primo blocco
A	EXE	53
B	TXT	51
...

e il seguente frammento
di **FAT**

indice	valore
...	...
50	55
51	56
52	58
53	50
54	46
55	57
56	59
57	48
...	...

e supponendo di avere **blocchi da 2KB**,
in **quali blocchi fisici** sono
memorizzati i seguenti byte?

- a) byte 800 del file A.EXE
- b) byte 5000 del file A.EXE
- c) byte 3990 del file B.TXT

Soluzione: byte 800 del file A.EXE

tabella di directory

Nome	Estensione	Primo blocco
A	EXE	53
B	TXT	51
...

FAT

indice	valore
...	...
50	55
51	56
52	58
53	50
54	46
55	57
56	59
57	48
...	...

blocco 53: essendo la dimensione dei blocchi pari a 2048 byte, il quoziente intero della divisione di 800 per 2048 è 0, quindi

- il byte 800 si trova nel blocco logico n. 0 del file A.EXE che è il primo blocco fisico del file, cioè il blocco 53, come indicato nella tabella di directory

Soluzione: byte 5000 del file A.EXE

tabella di directory

Nome	Estensione	Primo blocco
A	EXE	53
B	TXT	51
...

FAT

indice	valore
...	...
50	55
51	56
52	58
53	50
54	46
55	57
56	59
57	48
...	...

blocco 55: il quoziente intero della divisione di 5000 per 2048 è 2, quindi

- il byte 5000 si trova nel blocco logico n. 2 del file A.EXE
- il blocco fisico 53 ospita il blocco logico n. 0 e, in base alle informazioni contenute nella FAT, il blocco fisico successivo è 50 e ospita il blocco logico n. 1, e quindi il blocco fisico successivo, 55, ospita il blocco logico il n. 2

Soluzione: byte 3990 del file B.TXT

tabella di directory

Nome	Estensione	Primo blocco
A	EXE	53
B	TXT	51
...

FAT

indice	valore
...	...
50	55
51	56
52	58
53	50
54	46
55	57
56	59
57	48
...	...

blocco 56: il quoziente intero della divisione di 3990 per 2048 è 1, quindi

- il byte 3990 si trova nel blocco logico n. 1 del file B.TXT
- il blocco fisico 51 ospita il blocco logico n. 0 e dalle informazioni contenute nella FAT segue che il blocco fisico successivo, 56, ospita il blocco logico n. 1

Esercizio 3 (vale 3 punti)

Si consideri un file system che utilizza 16 blocchi fisici ed usa la tecnica del vettore di bit per tenere traccia della memoria libera. Si consideri la seguente configurazione iniziale del vettore di bit: **0111 0011 1111 0101** con la convenzione che i blocchi di indice minore sono rappresentati dai bit più a sinistra e il bit di un blocco è a 1 quando il blocco è libero. Supponendo che il file system allochi per primi i blocchi disponibili di indice minore (non richiedendo necessariamente il vincolo della contiguità), mostrare la **configurazione del vettore di bit** dopo l'esecuzione di ciascuna delle operazioni della sequenza seguente:

- a) creazione del file A: 4 blocchi
- b) creazione del file B: 3 blocchi
- c) aggiunta di 2 blocchi ad A
- d) cancellazione di B

Soluzione

0111 0011 1111 0101

a) creazione del file A: 4 blocchi

Soluzione

0111 0011 1111 0101

a) creazione del file A: 4 blocchi

0000 0001 1111 0101

b) creazione del file B: 3 blocchi

Soluzione

0111 0011 1111 0101

a) creazione del file A: 4 blocchi

0000 0001 1111 0101

b) creazione del file B: 3 blocchi

0000 0000 0011 0101

c) aggiunta di 2 blocchi ad A

Soluzione

0111 0011 1111 0101

a) creazione del file A: 4 blocchi

0000 0001 1111 0101

b) creazione del file B: 3 blocchi

0000 0000 0011 0101

c) aggiunta di 2 blocchi ad A

0000 0000 0000 0101

d) cancellazione di B

Soluzione

0111 0011 1111 0101

a) creazione del file A: 4 blocchi

0000 0001 1111 0101

b) creazione del file B: 3 blocchi

0000 0000 0011 0101

c) aggiunta di 2 blocchi ad A

0000 0000 0000 0101

d) cancellazione di B

0000 0001 1100 0101

Esercizio 4 (vale 5 punti)

Si consideri un file system FAT16 su una **partizione da 2GB**, in cui sia stata scelta la dimensione minima dei blocchi di allocazione. Si supponga che la partizione sia **piena al 50%**, che la dimensione di almeno la **metà dei file sia inferiore a 2 KB** e che questi occupino il **25% della partizione**.

- a) Calcolare la **dimensione dei blocchi** di allocazione
- b) Stimare il **numero dei file** di dimensione inferiore a 2 KB
- c) Stimare (per difetto) lo **spazio su disco sprecato** a causa della frammentazione interna ai blocchi
- d) Stimare (per eccesso) lo **spazio che occupano gli stessi file** nel caso in cui la partizione venga formattata con FAT32 e blocchi da 4 KB
- e) Stimare (per eccesso) la nuova **percentuale di occupazione** della partizione

Soluzione (a)

a) Calcolare la dimensione dei blocchi di allocazione

- La dimensione minima dei blocchi dipende dalla dimensione della partizione e dal file system e si ottiene quando si usa il massimo numero possibile di blocchi
 - Nel caso di FAT16, è 2^{16}
- Pertanto, dato che la partizione è da $2\text{GB} = 2^{31}$ byte, si ha che la dimensione minima dei blocchi è $2^{31} \text{ byte} / 2^{16} = 2^{15} \text{ byte} = 32 \text{ KB}$

Dati:

- partizione da 2GB, piena al 50%, blocchi di dimensione minima
- la dimensione di almeno la metà dei file è inferiore a 2 KB e questi occupano il 25% della partizione

Soluzione (b)

b) Stimare il numero dei file di dimensione inferiore a 2 KB.

- Si sa che i file piccoli:
 - occupano il 25% della partizione cioè complessivamente 512 MB
 - costituiscono, in numero, almeno la metà di tutti i file
- Un file, anche se più piccolo di 2 KB, occupa comunque un blocco intero. Quindi una stima del numero dei file 'piccoli' è: $N_{small} = 512 \text{ MB} / 32 \text{ KB} = 2^{29} \text{ byte} / 2^{15} \text{ byte} = 2^{14}$
- Dell'altra metà dei file si può solo dire che sono $N_{large} \leq N_{small}$ e che occupano complessivamente 512 MB

Dati:

- partizione da 2GB, piena al 50%, blocchi di dimensione minima
- la dimensione di almeno la metà dei file è inferiore a 2 KB e questi occupano il 25% della partizione

Soluzione (c)

c) Stimare (per difetto) lo spazio su disco sprecato a causa della frammentazione interna ai blocchi.

- In totale i file piccoli occupano nell'organizzazione attuale 512 MB
- Essendo di dimensioni inferiori a 2 KB essi dovrebbero occupare (al più): $S_{min} = N_{small} \cdot 2KB = 2^{14} \cdot 2^{11} \text{ byte} = 2^{25} \text{ byte} = 32MB$
- Quindi, lo spazio sprecato attribuibile ai file piccoli è (perlomeno) $512 \text{ MB} - S_{min} = 512MB - 32MB = 480MB!$
- Per gli altri file non è possibile alcuna stima quindi, assumendo di non avere per essi alcuno spreco, nella migliore delle ipotesi, lo spazio sprecato complessivamente è **480 MB**

Dati:

- partizione da 2GB, piena al 50%, blocchi di dimensione minima
- la dimensione di almeno la metà dei file è inferiore a 2 KB e questi occupano il 25% della partizione

Soluzione (d)

d) Stimare (per eccesso) lo spazio che occupano gli stessi file nel caso in cui la partizione venga formattata con FAT32 e blocchi da 4 KB.

- Nella nuova organizzazione ciascun file 'piccolo', cioè di dimensioni inferiore a 2 KB, occupa 1 blocco da 4 KB.
- Quindi, lo spazio complessivo occupato dai file piccoli è:
$$S_{\text{FAT32}} = N_{\text{small}} \cdot 4\text{KB} = 2^{14} \cdot 2^{12} \text{ byte} = 2^{26} \text{ byte} = 64\text{MB}$$
- Per gli altri file non è possibile alcuna stima, quindi possiamo assumere che essi mantengano tuttalpiù l'occupazione originaria pari a 512 MB
- L'occupazione totale è quindi $64\text{MB} + 512\text{MB} = 576\text{MB}$
- Lo spazio recuperato (con la migliore allocazione dei file piccoli) è $512 \text{ MB} - 64 \text{ MB} = 448 \text{ MB}$

Dati:

- partizione da 2GB, piena al 50%, blocchi di dimensione minima
- la dimensione di almeno la metà dei file è inferiore a 2 KB e questi occupano il 25% della partizione

Soluzione (e)

a) Stimare (per eccesso) la nuova percentuale di occupazione della partizione.

- La percentuale di occupazione totale della partizione è al più pari a: $576 \text{ MB} / 2 \text{ GB} \approx 28\%$

Dati:

- partizione da 2GB, piena al 50%, blocchi di dimensione minima
- la dimensione di almeno la metà dei file è inferiore a 2 KB e questi occupano il 25% della partizione

Esercizio 5 (vale 3 punti)

Calcolare la **dimensione (in byte) della FAT** per una **partizione da 1GB** con **blocchi da 16 KB** e **indirizzi dei blocchi a 16 bit**, e dire **quanti blocchi** della partizione occupa la FAT.

Dire inoltre **quanti accessi al disco** sarebbero necessari, nel caso peggiore, per leggere il byte 163.830 di un file, nell'ipotesi che solo la tabella della directory in cui si trova il file risieda in memoria principale.

Soluzione

Calcolare la **dimensione (in byte) della FAT** per una partizione da 1GB con blocchi da 16 KB e indirizzi dei blocchi a 16 bit, e dire **quanti blocchi** della partizione occupa la FAT.

- La dimensione della partizione è di $1\text{GB} = 2^{30}$ byte e quella dei blocchi è di $16\text{ KB} = 2^{14}$ byte
- Quindi, il numero dei blocchi è $2^{30}/2^{14} = 2^{16}$
- Perciò, il campo di indirizzamento della FAT (16 bit) consente di indirizzare tutti i blocchi
- Dato che ogni indirizzo è di 2 byte, la dimensione della FAT è di $2^{16} \times 2\text{ byte} = 2^{17}\text{ byte} = 128\text{ KB}$
- Pertanto, occupa $2^{17}/2^{14} = 8\text{ blocchi}$ della partizione

Soluzione

Dire inoltre **quanti accessi al disco** sarebbero necessari, nel caso peggiore, per leggere il byte 163.830 di un file, nell'ipotesi che solo la tabella della directory in cui si trova il file risieda in memoria principale.

- Per leggere il byte 163.830, se si usa la FAT e i blocchi della partizione vengono letti uno alla volta (caso peggiore), sono necessari al massimo 8 accessi al disco per leggere la FAT (che si trova nella partizione su disco) e un ulteriore accesso per leggere il blocco che contiene il byte richiesto
- Quindi, nel caso peggiore, occorrono **9 accessi**

Soluzione

Dire inoltre **quanti accessi al disco** sarebbero necessari, nel caso peggiore, per leggere il byte 163.830 di un file, nell'ipotesi che solo la tabella della directory in cui si trova il file risieda in memoria principale.

E se invece non si usasse la FAT, quanti accessi al disco sarebbero

In tal caso, sarebbero necessari 11 accessi al disco, perché andrebbero letti, uno alla volta, tutti i primi 11 blocchi del file

Soluzione

Dire inoltre **quanti accessi al disco** sarebbero necessari, nel caso peggiore, per leggere il byte 163.830 di un file, nell'ipotesi che solo la tabella della directory in cui si trova il file risieda in memoria principale.

Infatti, supponendo che i blocchi del file siano numerati a partire da 0, il byte 163.830 si trova nel blocco (logico) 10 (che è il quoziente intero della divisione di 163.830 per 16.382, valore ottenuto dalla dimensione dei blocchi $2^{14} = 16.384$ diminuita di 2 per considerare che ogni blocco contiene anche l'indirizzo, di 2 byte, del blocco successivo)

NB. Poiché la tabella della directory è in memoria, determinare l'indice del blocco iniziale del file non richiede accessi al disco

Esercizio 6 (vale 5 punti)

Un hard disk ha la **capienza di 2^{38} byte**, ed è formattato in **blocchi da 512 byte**.

- a) **Quanti accessi al disco** sono necessari per leggere l'ultimo blocco di un file A della dimensione di 4096 byte, assumendo che il numero del primo blocco del file stesso sia già in RAM e che venga adottata una allocazione concatenata dei blocchi del file?
- b) Qual è lo **spreco di memoria** dovuto alla frammentazione interna nella memorizzazione di A?
- c) Se si adottasse una **allocazione indicizzata** dei file, quanti accessi al disco sarebbero necessari per leggere l'ultimo byte di un file B grande 100KB? (specificare le assunzioni fatte nel rispondere a questa domanda e motivare la risposta)

Soluzione (a)

a) **Quanti accessi al disco** sono necessari per leggere l'ultimo blocco di un file A della dimensione di 4096 byte, assumendo che il numero del primo blocco del file stesso sia già in RAM e che venga adottata una allocazione concatenata dei blocchi del file?

- Il numero dei blocchi è $2^{29} = 2^{38}/2^9$; quindi per esprimere il numero di un blocco sono sufficienti 4 byte
- Perciò ogni blocco memorizza 4 byte per il puntatore al blocco successivo e 508 byte di dati, e sono necessari $9 = \lceil 4096/508 \rceil$ blocchi per memorizzare l'intero file
- Sono quindi necessari **9 accessi** in tutto perché, essendo l'allocazione dei blocchi del file concatenata, per accedere l'ultimo blocco sarà prima necessario accedere tutti quelli che lo precedono

Dati: capienza disco 2^{38} byte, dimensione blocco 512 byte

Soluzione (b)

b) Qual è lo spreco di memoria dovuto alla frammentazione interna nella memorizzazione di A?

- Poiché la divisione intera della lunghezza in byte del file A, 4096, per il numero di byte dati contenuti in un singolo blocco, 508, restituisce quoziente 8 e resto 32, abbiamo che il file A occupa 9 blocchi, di cui i primi 8 per intero e l'ultimo per soli 32 byte
- Lo spreco è allora dato da $508 - 32 = 476$ byte

Dati: capienza disco 2^{38} byte, dimensione blocco 512 byte

Soluzione (c)

c) Se si adottasse una **allocazione indicizzata** dei file, quanti accessi al disco sarebbero necessari per leggere l'ultimo byte di un file B grande 100KB? (specificare le assunzioni fatte nel rispondere a questa domanda e motivare la risposta)

- Poiché un blocco è di 512 byte e sono necessari 4 byte per scrivere il numero di un blocco, in un blocco indice possono essere memorizzati $128 = 2^7$ puntatori a blocco
- Quindi, con un blocco indice possiamo gestire uno spazio disco pari a $128 \times 512 \text{ byte} = 2^7 \times 2^9 \text{ byte} = 2^{16} \text{ byte} = 64\text{KB}$
- Un solo blocco indice non è quindi sufficiente a indicizzare B
- Assumendo una allocazione indicizzata concatenata, usando un secondo blocco indice è possibile indicizzare l'intero file
- Se il numero (non il blocco stesso!) del primo blocco indice è già in RAM, per leggere l'ultimo byte del file sono necessari **3 accessi al disco**: lettura del primo blocco indice, lettura del secondo blocco indice, lettura dell'ultimo blocco del file