

PROGRAMMAZIONE DI SISTEMA

Esercizi proposti e svolti di teoria
Tratti da compiti di esame

Gestione Memorie di Massa e File System (esercizi relativi a cap 11, 13 e 14)

1. Siano dati due file system F1 e F2, su due volumi, basati rispettivamente su FAT e su Inode. I puntatori hanno dimensione di 32 bit, i blocchi hanno dimensione 4KB, entrambi i volumi hanno una parte riservata ai metadati (direttori, FAT e FCB (File Control Block) oppure Inode, blocchi indice, ecc.) e una parte ai blocchi di dato. Si supponga che lo spazio riservato ai blocchi di dato sia lo stesso per F1 e F2. Si sa che la FAT occupa 150 MB.

Quanti blocchi di dato possono contenere complessivamente, al massimo, i file presenti nei due file system?

- N blocchi F1: Ad ogni entry nella FAT corrisponde un blocco dato $N_{F1} = 150 \text{ MB} / 4 \text{ B} = 37,5 \text{ M blocchi}$
- N blocchi F2: $N_{F2} = N_{F1} = 37,5 \text{ M blocchi}$ (per definizione)

Quale è il numero massimo di blocchi dato liberi ?

- F1: (stesso valore di quelli occupati) 37,5 M blocchi
- F2: 37,5 M blocchi (come sopra)

Si supponga che i blocchi dato liberi di F1 siano gestiti mediante free list.

- La free list di F1 può essere rappresentata direttamente nella FAT?

(SI/NO, motivare) SI. La FAT contiene sia le liste dei riferimenti a blocchi occupati dai file che la free-list. Non ha senso fare altrimenti, in quanto gli indici (nella FAT) dei blocchi liberi sono disponibili (non usati per file), quindi utilizzabili per la free list.

Si vorrebbero organizzare blocchi liberi di F2 mediante lista concatenata di blocchi indice, ognuno dei quali contiene un puntatore al prossimo blocco indice in lista e N_{Free} puntatori a blocchi di dato liberi.

- E' possibile adottare tale rappresentazione (mentre per i file si usa una rappresentazione basata su Inode)?

(SI/NO, motivare) SI. La rappresentazione ad Inode non rende necessario che anche i blocchi liberi siano di fatto come un file aggiuntivo accessibile mediante Inode. Volendo, si potrebbe usare una bitmap o una lista semplice di blocchi. La scelta "lista di blocchi indice, ognuno contenente puntatori a blocchi liberi", è quindi lecita.

- (Se SI alla risposta precedente)
 - Quale è il valore di N_{Free} ? $N_{Free} = 4 \text{ KB} / 4 \text{ B} - 1 = 1023$ (numero totale di puntatori in un blocco – il puntatore per il prossimo in lista concatenata)
 - Quanti blocchi indice servono, al massimo, per la free list di F2? Ogni blocco indice punta a $N_{Free} = 1023$ blocchi liberi. Il massimo numero di blocchi liberi è $N_{F2} = 37,5 \text{ M blocchi}$. Il numero massimo di blocchi indice per la free list è $N_{IndFree} = 37,5 \text{ M} / 1023 = 37,5 \text{ K} * 1024 / 1023 \approx 37,5 \text{ K}$

- ~~(Se NO alla risposta precedente)~~
Dire come andrebbero organizzati i blocchi liberi

.....
.....

Siano dati un file "a.mp4" (in F1) di dimensione 317 MB e un file "b.mp4" (in F2) di dimensione 751 MB.

- Quanti blocchi di dato occupa a.mp4 ? $317 \text{ MB} / 4 \text{ KB} = 317 / 4 \text{ K} = 79,25 \text{ K blocchi}$
- Quanti indici nella FAT sono utilizzati per a.mp4 ? 79,25 K (uguale al numero di blocchi dato)
- Quanti blocchi di dato e di indice (escluso l'Inode) occupa b.mp4 ?
DATO: $751 \text{ MB} / 4 \text{ KB} = 751 / 4 \text{ K} = 187,75 \text{ K}$
INDICE: 10 blocchi dato sono puntati direttamente all'Inode. Un blocco indice punta a $1024 = 1 \text{ K}$ blocchi dato.
Indice indiretto singolo: 1 blocco indice
Indici indiretti doppi: a primo livello 1, a secondo livello $\lceil (187,75 \text{ K} - 10) / 1 \text{ K} \rceil - 1 = 187$ (basta indiretto doppio)
Totale: 1 (singolo) + 1 (doppio primo livello) + 187 (doppio secondo livello) = 189

2. Sia dato il modulo di un kernel che deve gestire la schedulazione delle richieste di accesso a un disco.

3a) Perché le politiche SCAN e C-SCAN non servono con dischi SSD?

Perché le politiche cercano di ridurre la distanza totale percorsa (misurata in differenze tra indici di blocco/cilindro): questo ha senso per i dischi magnetici, nei quali si riduce il tempo di posizionamento delle testine. Per i dischi SSD non ci sono penalità (costi) legati alla differenza tra indici di blocco.

E possibile che il sistema abbia un disco magnetico (A) di 500GB e uno SSD (B) di 200GB, organizzati in modo tale da formare un unico volume?

Sì. La tecnologia del disco viene vista al livello basso dei driver dei dispositivi. Partizionamento e formattazione dei dischi permettono sia di ricavare più volumi da un solo disco che di unire più dischi in un unico volume. Le prestazioni potranno ovviamente variare a seconda del dispositivo.

(Se Sì alla domanda precedente) quale dimensione ha il volume? $500GB + 200GB = 700GB$

3b) Nel sistema (ATTENZIONE: si tratta di un'altra sotto-domanda, indipendente dalla precedente) si vogliono supportare paginazione e swapping. E' più efficiente una partizione di swap raw oppure uno swapfile in un filesystem esistente? (motivare la risposta)

E' più efficiente, in termini di tempi di accesso, la partizione raw, in quanto si gestiscono/utilizzano direttamente blocchi fisici su disco, senza passare dai vari livelli del file system.

3c) Si vuole realizzare un nuovo tipo di filesystem, nel quale è possibile avere sia file che direttori condivisi (l'albero di direttori è un DAG!).

Perché eventuali cicli nel grafo dei direttori rappresentano un problema?

Appunto perché con i cicli il grafo non è un DAG, cioè un direttorio avrebbe come discendente/successore nell'albero/DAG un suo antenato/predecessore! Questo va evitato.

Quali strategie è possibile adottare per ovviare al problema (dei cicli)?

- a) *Condividere solo file (le foglie) e non i direttori (ma questo limiterebbe la richiesta: si vogliono "sia file che direttori condivisi")*
- b) *Garbage collection: si accettano i cicli (per non fare controlli ad ogni operazione di link/condivisione di file/direttorio), ma periodicamente si fa eseguire sul file system una funzione di controllo che individua e rimuove eventuali cicli*
- c) *Ad ogni generazione di un nuovo link (a file/direttorio esistente) si fa un controllo di presenza di eventuali cicli, che determina la validità o meno dell'operazione.*

3. (3 punti) Sia dato un file system Unix, basato su inode aventi 13 puntatori (10 diretti, 1 indiretto singolo 1 doppio e 1 triplo). I puntatori hanno dimensione di 32 bit e i blocchi hanno dimensione 2KB. Si sa che il file system contiene 1000 file di dimensione media 15MB e che la frammentazione interna totale è di 1MB, si calcoli il massimo numero possibile di file con indice indiretto triplo e con indice indiretto doppio che possono essere presenti nel file system.

R1	<p>Osservazioni</p> <p><i>Un blocco indice contiene $2KB/4B = 512$ puntatori.</i></p> <p><i>La dimensione (netta, indipendente dal tipo di file system) complessiva dei file è $15MB * 1000 = 15000MB$ (= 15GB circa)</i></p> <p><i>L'occupazione (questa dipende dal file system e tiene conto dei blocchi effettivamente usati), tenendo conto della frammentazione interna, è $15001MB = 15001 * 512$ blocchi.</i></p> <p>Calcolo massimo per N2/N3 (numero file con indice indiretto doppio/triplo)</p> <p>Per calcolare il numero massimo occorre considerare l'occupazione minima.</p> <p><i>Si indicano con MIN_2 e MIN_3 le occupazioni minime dei due tipi di file:</i></p> <p>$MIN_2 = (10 + 512 + 1)$ blocchi</p> <p>$MIN_3 = (10 + 512 + 512^2 + 1)$ blocchi</p> <p>$N2 = \lfloor 15001 * 512 / 523 \rfloor = 14685$</p> <p>$N3 = \lfloor 15001 * 512 / (523 + 512^2) \rfloor = 29$</p> <p><i>N2 supera il numero di file presenti (1000) quindi il numero massimo di file con indice indiretto doppio è limitato a 1000 (questo commento è stato aggiunto rispetto alla lezione videoregistrata).</i></p>
-----------	---

Qual è il massimo numero possibile di file privi di frammentazione interna (*motivare la risposta*)?

R2	<p>Osservazioni <i>La frammentazione interna complessiva è nota (1MB). Il massimo numero di file a frammentazione interna nulla (N_{MAX}) si ottiene ripartendo la frammentazione sul minimo numero N_{min} di file, aventi quindi frammentazione massima di 1 blocco – 1 Byte ciascuno</i></p> <p>Calcolo $N_{min} = \lceil 1MB / (2KB-1B) \rceil = \lceil 1MB / 2047 \rceil = 513$ $N_{MAX} = 1000 - 513 = 487$</p>
-----------	--

Si sa che per un file “a.dat” si utilizzano 2000 blocchi di indice. Quanti blocchi di dato occupa e quale dimensione ha il file “a.dat” (*qualora non siano possibili valori univoci, si determinino valori minimi e massimi*)?

R3	<p>Osservazioni <i>Si osservi che il numero di blocchi di indice per un file che occupa totalmente il secondo livello è 1 (singolo) + 1 (doppio primo liv.) + 512 (doppio secondo liv.). Per 2000 blocchi indice occorre quindi indice indiretto triplo, tale da utilizzare $2000 - 514 = 1486$ blocchi indice</i></p> <p>Calcolo <i>Occorre calcolare quanti blocchi indice triplo a secondo e terzo livello (usati solo in parte) ci sono (a primo livello ce n'è uno solo):</i></p> <p>$\lceil 1486/512 \rceil = 3 \Rightarrow 1 \text{ (triplo, I liv.)} + 3 \text{ (triplo, II liv.)} + 1482 \text{ (triplo, III liv.)}$</p> <p>Occupazione: <i>numero blocchi dato NBL (i valori minimo e massimo dipendono da quanti indici ci sono nell'ultimo blocco indici (min 1 – max 512)):</i> $NBL_{min} = 10 \text{ (diretti)} + 512 \text{ (singolo)} + 512 \cdot 512 \text{ (doppio)} + 1481 \cdot 512 + 1 \text{ (triplo)} = 1020939$ $NBL_{MAX} = NBL_{min} + 511 = 1021450$</p> <p>Dimensione: <i>La minima si ottiene da assumendo frammentazione interna massima (2047 byte) – La massima si ottiene da assumendo frammentazione 0.</i> $DIM_{min} = NBL_{min} \cdot 2KB - 2047B$ $DIM_{MAX} = NBL_{MAX} \cdot 2KB$</p>
-----------	--

4. (3 punti) Sia dato un disco di tipo “solid state” (SSD).
 Quale è l'operazione che ne limita la vita e perché?

<p>La durata di un disco SSD è limitata dal numero massimo di cancellazioni. Le cancellazioni sono effettuate per blocchi/pagine e hanno impatto anche sul numero massimo di scritture, in quanto una scrittura (ri-scrittura) deve essere preceduta da cancellazione.</p>
--

Si supponga che un disco SSD da 1TB (TeraByte) sia garantito per 3 anni, supportando un massimo di 5TB scritti al giorno. Quale è il numero massimo (complessivo) di Byte che possono essere letti e scritti, durante l'intera vita del disco? (due risposte, una per le letture e una per le scritture, motivando la risposta e indicando i passaggi per calcolare il risultato)

<p>Letture: non c'è limite, le letture non sono un problema Scritture: il limite è dato dal limite giornaliero, moltiplicato per i giorni in 3 anni. $N_{write,max} = 5TB \cdot 3 \cdot 365 = 5 \cdot 1095 TB \approx 5.5 \cdot 10^3 TB$</p>

Sia dato un disco organizzato con struttura RAID. Che cosa si intende, in relazione a tale disco e alla probabilità di guasto e/o tempo medio tra guasti (MTTF: Mean Time To Failure), con i termini seguenti?

- Mean Time To Repair (MTTR):

Tempo Medio necessario per Riparare (e ripristinare) il disco una volta che si è guastato.

- Mean Time To Data Loss (MTTDL):

(Risposta integrata e completata rispetto alla prima versione) Tempo Medio (inverso di frequenza/probabilità) tra due perdite di dato (oppure fino alla prima perdita di dato): un dato viene perso perché interviene un secondo guasto dopo un primo, prima che la riparazione sia terminata.

Si consideri una struttura RAID con 2 dischi in configurazione “mirrored”. Se ognuno dei dischi può guastarsi in modo indipendente dall'altro, con MTTF = 50000 ore e MTTR = 20 ore, quanto sarà il MTTDL? (non basta il risultato, indicare i passaggi per calcolarlo)

(per una spiegazione dettagliata si vedano i lucidi Silberschatz modificati: [ch11 \(con aggiunte su dischi RAID\)](#))

$$\text{MTTDL} = \text{MTTF}^2 / (2 * \text{MTTR}) = 50000^2 / (2 * 20) \text{ ore} = 25/4 * 10^7 \text{ ore} = 6.25 * 10^7 \text{ ore}$$