

# 1 Input/Output

## 1.1 MTTF

Dati 4 dischi, calcolare MTTF orario di un'organizzazione RAID livello 0 e quello di un'organizzazione RAID livello 1, sapendo che ogni disco  $i$  ha un tasso annuale di fallimento  $\lambda_i$ , dove:

- $\lambda_1 = 0.05$
- $\lambda_2 = 0.075$
- $\lambda_3 = 0.25$
- $\lambda_4 = 0.025$

Non è previsto l'utilizzo di dischi hot-spare. Inoltre, una volta che un disco si guasta non viene sostituito da uno funzionante.

L'aggregato dei tassi di fallimento è  $\lambda_{tot} = 0.05 + 0.075 + 0.25 + 0.025 = 0.4$ ; il  $MTTF_{tot}$  orario è quindi  $24 \cdot 365 / 0.4 = 21900$  ore: dato un set di  $k$  dischi, la possibilità che un disco su quei  $k$  fallisca è almeno uguale alla possibilità che il disco con minor  $MTTF_H$  fallisca. Quindi, il MTTF orario di un'organizzazione RAID livello 0 è 21900 ore.

In un'organizzazione RAID livello 1 i dischi sono mirrored (quindi sono 8): in questo caso il  $MTTF_{tot}$  orario è  $24 \cdot 365 / 0.8 = 10950$  ore.

## 1.2 RAID

Un cliente vi commissiona un sistema organizzato a RAID (non specificando il livello) specificando i seguenti requisiti:

- Massimizzare capacità e prestazioni
- Affidabilità non necessaria
- Budget: €2000

Considerando solo i costi relativi all'acquisto dei dischi (escludendo quindi quelli relativi all'acquisto di un controller RAID) e supponendo di adottare dischi da 1 TiB, ciascuno venduto al prezzo di €400, che organizzazione RAID suggerireste al cliente tra quelle standard (livello 0 fino a livello 6)? Motivare le risposte.

Dato il budget di €2000, è possibile acquistare un massimo di 5 dischi. Essendo l'affidabilità un requisito non necessario, si possono escludere tutte le organizzazioni che prevedano qualche forma di ridondanza o error checking, quindi RAID0 risulta essere la scelta migliore. Questa organizzazione RAID garantisce di massimizzare la capacità e le prestazioni (sia in scrittura, sia in lettura), in quanto non presenta alcuna forma di ridondanza, arrivando ad ottenere 5 TiB di spazio totale.

### 1.3 RAID

Un cliente vi commissiona un sistema organizzato a RAID (non specificando il livello) specificando i seguenti requisiti:

- Massimizzare affidabilità
- Nessun interesse su capacità e prestazioni
- Budget: €2000

Considerando solo i costi relativi all'acquisto dei dischi (escludendo quindi quelli relativi all'acquisto di un controller RAID) e supponendo di adottare dischi da 1 TiB, ciascuno venduto al prezzo di €400, che organizzazione RAID suggerireste al cliente tra quelle standard (livello 0 fino a livello 6)? Motivare le risposte.

Dati i requisiti, si possono escludere RAID0 (non offre alcuna affidabilità), RAID1 (non è possibile fare mirroring su 5 dischi), RAID 3,4,5 (sono limitati ad un solo disk failure possibile). L'organizzazione RAID ideale è quella di livello 6, che ammette almeno due disk fault (sui 5 acquistati). Con questa soluzione si ottiene capacità utile di 3 TiB.

### 1.4 RAID

Un cliente vi commissiona un sistema organizzato a RAID (non specificando il livello) specificando i seguenti requisiti:

- Buon compromesso tra affidabilità e prestazioni (tolleranza ad almeno 1 disco rotto)
- Capacità utile di almeno 4 TiB
- Budget: €2000

Considerando solo i costi relativi all'acquisto dei dischi (escludendo quindi quelli relativi all'acquisto di un controller RAID) e supponendo di adottare dischi da 1 TiB, ciascuno venduto al prezzo di €400, che organizzazione RAID suggerireste al cliente tra quelle standard (livello 0 fino a livello 6)? Motivare le risposte.

Dati i requisiti, si possono escludere RAID0 (non offre alcuna affidabilità), RAID 1,2,6 (non permettono di raggiungere capacità utile di 4 TiB). Tra le organizzazioni RAID 3,4,5, quella ottimale risulta essere RAID 5 in quanto tra queste è quella che offre le prestazioni in lettura e scrittura migliori, ammettendo un disk fault.

### 1.5 DMA

Si supponga che un sistema utilizzi il DMA per il trasferimento di dati dal controller del disco alla memoria principale del computer. Si assuma inoltre che l'acquisizione dell'accesso al bus richieda  $t_1$  nsec in media e il trasferimento di una parola sul bus richieda  $t_2$  nsec (con  $t_1 \gg t_2$ ). Dopo che la CPU ha programmato il controller DMA, quanto tempo sarà necessario per trasferire 1000 parole dal controller del disco alla memoria principale, se (a) è utilizzata la modalità cycle stealing (una parola alla volta) (b) è usata la modalità burst? Si assuma che l'invio di comandi al controller

del disco richieda l'acquisizione del bus per spedire una parola e che il riscontro richieda anch'esso l'acquisizione del bus per spedire una parola.

Nel caso venga usata la modalità *cycle stealing*, sarà necessario un tempo pari a  $(t_1+t_2)*1000$  nsec, in quanto si può trasferire una sola parola per volta, dovendo richiedere ogni volta l'accesso al bus (richiede tempo  $t_1$ ). Nel caso venga usata la modalità *burst*, sarà necessario un tempo pari a  $t_1+(t_2*1000)$ , in quanto è necessario acquisire il bus una sola volta, ed effettuare il trasferimento delle parole in batch.

## 1.6 Dischi

Un disco ruota a 7200 RPM. Nel suo cilindro più esterno ci sono 500 settori di 512 bytes.

- Supponendo che la testina sia già posizionata sul cilindro più esterno e all'inizio di un settore, quanto ci s'impiega a leggere un singolo settore da quel cilindro?
- Qual è il massimo tasso di trasferimento dati (in byte/sec)?

Il tempo di trasferimento medio per un settore è pari a *massima latenza rotazionale/settori per traccia*; la massima latenza rotazionale è pari a  $1/vel. \text{ rotazione}$ . Quindi, per leggere un settore ci s'impiega  $T_T = 1/vel. \text{ rotazione} * 1/settori \text{ per traccia} = \frac{1}{7200/60} * 1000 * \frac{1}{500} = 8 * 0.002 = 0.016 \text{ msec}$ .

Il massimo tasso di trasferimento dati è pari a  $byte \text{ settore}/T_T = \frac{512}{0.016/1000} = 32000000 \text{ byte} = 32 \text{ megabyte/sec}$ .

## 1.7 Dischi

Si consideri un disco magnetico con 16 testine e 400 cilindri. Il disco ha 4 zone da 100 cilindri ciascuna, dove i cilindri (e quindi le tracce) di zone differenti contengono 160, 200, 240, e 280 settori, rispettivamente. Si assuma che ogni settore contenga 512 byte, che il tempo medio di seek tra cilindri adiacenti sia 1 msec, e che il disco ruoti a 7200 RPM. Calcolare:

- la capacità del disco
- il cylinder skew ottimale di ogni zona
- il massimo tasso di trasferimento dati (in byte/sec) raggiungibile

La capacità del disco si ottiene moltiplicando il numero di settori di ogni cilindro per il numero di cilindri, sommando i risultati, moltiplicando per 512 e per il numero di superfici (numero testine):  $(16000 + 20000 + 24000 + 28000) * 512 * 16 = 720896000 \text{ byte} = \text{circa } 720 \text{ megabyte}$ .

Il cylinder skew si calcola tramite la seguente formula  $\lceil \frac{\text{tempo di seek}}{\text{intervallo settori}} \rceil$ . Per la prima zona:  $\lceil \frac{1}{8/160} \rceil = 20$  settori. Per la seconda zona:  $\lceil \frac{1}{8/200} \rceil = 25$  settori. Per la terza zona:  $\lceil \frac{1}{8/240} \rceil = 30$  settori. Per la quarta zona:  $\lceil \frac{1}{8/280} \rceil = 35$  settori. Il massimo tasso di trasferimento dati si calcola nella zona con più settori ed è pari a  $byte \text{ settore}/T_T = \frac{512}{\frac{1}{7200/60} * 1000 * \frac{1}{280}} = \text{circa } 17 \text{ kilobyte/sec}$ .

## 1.8 RAID

Si supponga di effettuare le seguenti operazioni in sequenza su un sistema RAID 5 costituito da 5 dischi identici (inizialmente vuoti) e con blocchi (strip) da 1 byte:

1. Scrittura della sequenza di byte: 01000101, 00000110, 10110100, 11101101, 11000111, 10000101, 01110111, 01010101
2. Lettura secondo e settimo byte di dati
3. Modifica del terzo byte di dati da 10110100 a 01001011

Per ogni punto, s'illustrino le operazioni compiute dal sistema, evidenziando quante READ e quante WRITE vengono effettuate, e quante di queste sono fatte in parallelo.

NOTA:

- Per ognuno dei suddetti punti, il controller RAID riceve i comandi di scrittura/lettura di byte come un'unica richiesta.
- L'ordine di scrittura sui dischi è da sinistra verso destra, dall'alto verso il basso.
- Racchiudere ogni blocco di parità tra parentesi tonde.
- Il sistema RAID non è dotato di dischi hot-spare.

1. Essendo i dischi vuoti, vengono effettuate solo operazioni di scrittura (1 per ogni byte + 2 per blocchi di parità = 10). I blocchi di parità sono settati a (00011010) e (01100000).

Stripe	Disco 0	Disco 1	Disco 2	Disco 3	Disco 4
0	01000101	00000110	10110100	11101101	(00011010)
1	11000111	10000101	01110111	(01100000)	01010101

2. Il secondo e settimo byte sono letti in parallelo in quanto si trovano su dischi diversi. Sono effettuate 2 letture e 0 scritture.

3. Per modificare il terzo bit, vengono effettuate 2 letture (blocco da modificare e blocco di parità) e due scritture (blocco modificato e nuovo bit di parità) in parallelo: in questo caso infatti può essere usata la parità sottrattiva.

Stripe	Disco 0	Disco 1	Disco 2	Disco 3	Disco 4
0	01000101	00000110	01001011	11101101	(11100101)
1	11000111	10000101	01110111	(01100000)	01010101

## 1.9 RAID

Si supponga di effettuare le seguenti operazioni in sequenza su un sistema RAID costituito da 4 dischi identici (inizialmente vuoti), assumendo che il sistema operativo utilizzi blocchi di dimensione 2 byte:

- Scrittura delle sequenza di blocchi: 0100010100000110, 1011010011101101, 0111000111000011, 1000111110101100

- Lettura del secondo e del quarto blocco
- Modifica del primo e del terzo blocco in: 1011101001001011, 1000111000111100

Spiegare come vengono effettuate le suddette operazioni utilizzando le seguenti organizzazioni RAID:

- RAID di livello 0 con strip da 1 byte
- RAID di livello 1 con blocchi da 1 byte

Per ogni punto, s'illustrino le operazioni compiute dal sistema, evidenziando quante READ e quante WRITE vengono effettuate, e quante di queste sono fatte in parallelo. NOTA:

- Per ognuno dei suddetti punti, il controller RAID riceve i comandi di scrittura/lettura di byte come un'unica richiesta.
- L'ordine di scrittura sui dischi è da sinistra verso destra, dall'alto verso il basso.
- Racchiudere ogni blocco di parità tra parentesi tonde.
- Il sistema RAID non è dotato di dischi hot-spare.

Con organizzazione RAID 0:

1. Vengono effettuate 8 scritture, delle quali 4 in parallelo (per due volte)

Stripe	Disco 0	Disco 1	Disco 2	Disco 3
0	01000101	00000110	10110100	11101101
1	01110001	11000011	10001111	10101100

2. Le letture del secondo e quarto blocco avvengono sequenzialmente (4 letture, 0 scritture)
3. Il primo e il terzo blocco sono scritti sequenzialmente (0 letture, 4 scritture)

Con organizzazione RAID 1:

1. Vengono effettuate 16 scritture, delle quali 4 in parallelo (per 4 volte)

Stripe	Disco 0	Disco 1	Disco 2	Disco 3
0	01000101	00000110	01000101	00000110
1	10110100	11101101	10110100	11101101
2	01110001	11000011	01110001	11000011
3	10001111	10101100	10001111	10101100

2. Le letture del secondo e quarto blocco avvengono in parallelo, dato che possono essere letti anche i dischi clone (4 letture, 0 scritture)
3. Le scritture del primo e terzo blocco avvengono sequenzialmente (0 letture, 8 scritture)

## 1.10 RAID

Si consideri un sistema RAID di livello 4 composto da  $N$  dischi (dove  $N$  comprende il disco di parità e quindi  $N > 1$ ) a cui arriva una richiesta di scrittura per  $B$  blocchi che interessa una sola stripe (quindi  $B < N$ ). Determinare il valore di  $B$  per cui risulti più efficiente (in termini di numero totale di operazioni di I/O su disco) usare il metodo della parità additiva piuttosto che quello della parità sottrattiva.

La soglia per cui risulta più efficiente utilizzare il metodo della parità sottrattiva è  $B = \lfloor \frac{n}{2} - 1 \rfloor$ . Prendendo per esempio un sistema RAID4 composto da 6 dischi (5 + 1 di parità): Parità sottrattiva (i numeri dell'elenco sono il numero di blocchi coinvolti):

1. 2 letture, 2 scritture
2. 3 letture, 3 scritture
3. 4 letture, 4 scritture
4. 5 letture, 5 scritture
5. 6 letture, 6 scritture

Parità additiva:

1. 4 letture, 2 scritture
2. 3 letture, 3 scritture
3. 2 letture, 4 scritture
4. 1 lettura, 5 scritture
5. 0 letture, 6 scritture

## 1.11 Seek

Al driver del disco arrivano, nell'ordine specificato, le richieste di cilindri 11, 23, 21, 3, 41, 7, e 39. Ogni operazione di seek tra due cilindri consecutivi (track-to-track seek time) impiega 6 msec. Specificare l'ordine di visita dei vari cilindri e il tempo totale di seek che si ottengono utilizzando i seguenti algoritmi:

1. First-Come, First-Served (FCFS)
2. Shortest-Seek First (SSF)
3. LOOK
4. C-LOOK

In tutti i casi, il braccio del disco è inizialmente posizionato sul cilindro 31. Per gli algoritmi LOOK e varianti, la direzione iniziale è DOWN.

1. Ordine di visita: 11, 23, 21, 3, 41, 7, 39.  
Tempo totale di seek:  $(20 + 12 + 2 + 18 + 38 + 34 + 32) * 6 = 936$  msec
2. Ordine di visita: 23, 21, 11, 7, 3, 39, 41 oppure 39, 41, 23, 21, 11, 7, 3.  
Tempo totale di seek:  $(8 + 2 + 10 + 4 + 4 + 36 + 2) * 6 = 396$  msec oppure  $(8 + 2 + 18 + 2 + 10 + 4 + 4) * 6 = 288$  msec
3. Ordine di visita: 23, 21, 11, 7, 3, 39, 41.  
Tempo totale di seek:  $(8 + 2 + 10 + 4 + 4 + 36 + 2) * 6 = 396$  msec
4. Ordine di visita: 23, 21, 11, 7, 3, 41, 39.  
Tempo totale di seek:  $(8 + 2 + 10 + 4 + 4 + 38 + 2) * 6 = 408$  msec

## 1.12 Risparmio energetico

Un processo richiede l'utilizzo della CPU ogni 0.5 sec. Si supponga che la CPU risponda immediatamente alla richiesta e impieghi 100 msec per processarla. Inoltre, sia  $P$  W il consumo elettrico della CPU e si supponga che il consumo elettrico della CPU in stato idle sia nullo (cioè, il consumo elettrico statico è zero). Rispondere alle seguenti domande:

1. Se il voltaggio  $V$  della CPU è ridotto a  $V/n$ , qual è il valore ottimale di  $n$ ?
  2. Qual è il risparmio energetico che si ottiene in 1 sec, riducendo il voltaggio della CPU come al punto (1) rispetto a non ridurlo?
1. Con  $n = 5$ , la CPU è ancora in grado di processare la richiesta (ora richiede 0.5 secondi).
  2. Quando il voltaggio non è ridotto, la CPU consuma  $E_{nocut} = P * 0.1sec + 0 = 0.1P$   
Quando è ridotto ( $P/n^2$ ),  $E_{cut} = \frac{P}{25} * 0.5sec + 0 = 0.02P$ .  
Il risparmio energetico è pari a  $\Delta E = E_{nocut} - E_{cut} = 0.1 - 0.02/E_{nocut} = 0.08/0.1 = 80\%$

## 1.13 Risparmio energetico

Un processo richiede l'utilizzo della CPU ogni 0.5 sec. Si supponga che la CPU risponda immediatamente alla richiesta e impieghi 100 msec per processarla. Inoltre, si supponga che il consumo elettrico statico della CPU sia  $PS = 30$  W e quello dinamico sia  $PD = 10$  W. Rispondere alle seguenti domande:

1. Se il voltaggio  $V$  della CPU è ridotto a  $V/n$ , qual è il valore ottimale di  $n$ ?
  2. Qual è il risparmio energetico che si ottiene in 1 sec, riducendo il voltaggio della CPU come al punto (1) rispetto a non ridurlo?
1. Con  $n = 5$ , la CPU è ancora in grado di processare la richiesta (ora richiede 0.5 secondi).
  2. Quando il voltaggio non è ridotto, la CPU consuma  $E_{nocut} = 10 * 0.2sec + 30 * 1sec = 32W$   
Quando è ridotto ( $P/n^2$ ),  $E_{cut} = \frac{10}{25} * 0.5sec + 30 = 30.2W$ .  
Il risparmio energetico è pari a  $\Delta E = E_{nocut} - E_{cut} = 32 - 30.2/32 = 1.8/32 = 5.6\%$

## 1.14 Risparmio energetico

Si supponga di voler attuare una politica di risparmio del consumo energetico basata sull'utilizzo del disco tale per cui in base a una stima  $T_{est}$  dell'arrivo della prossima richiesta di utilizzo del disco si decide se lasciare i suoi piatti in movimento (stato "active") o arrestarne la rotazione (stato "sleep"). In particolare, si consideri un disco con le seguenti caratteristiche:

- consumo elettrico nello stato "active"  $P_w=6$  Watt
- consumo elettrico nello stato "sleep"  $P_s=2$  Watt
- tempo per passare da "active" a "sleep"  $T_{sd}=10$  sec
- energia elettrica consumata nel passaggio da "active" a "sleep"  $E_{sd}=26$  Joule
- tempo per passare da "sleep" a "active"  $T_{wu}=5$  sec
- energia elettrica consumata nel passaggio da "sleep" a "active"  $E_{wu}=60$  Joule

Calcolare il valore  $T_d$  tale per cui per  $T_{est} > T_d$  diventi vantaggioso porre il disco da "active" a "sleep".

$T_d$  è il break-even point, ed è pari a  $T_d = (E_{sd} + E_{wu} - P_s * (T_{sd} + T_{wu})) / P_w - P_s = (26 + 60 - 2 * (10 + 5)) / 6 - 2 = (26 + 60 - 8 * 15) / 4 = 14sec$ .

Quindi, conviene porre disco da "active" a "sleep" se l'intervallo tra le richieste è più di 14 secondi.

## 1.15 Interrupt

Si considerino le seguenti situazioni e indicare quali rappresentano interrupt precisi e quali interrupt imprecisi. Il simbolo PC denota il Program Counter. L'ordine delle istruzioni va dal basso verso l'alto. Motivare le risposte.

(Riferirsi al pdf degli esercizi per le immagini)

Le quattro proprietà dell'interrupt preciso:

1. Il PC è salvato in un luogo ben definito
2. Tutte le istruzioni prima di quella puntata dal PC sono state completate
3. Nessuna istruzione dopo quella puntata dal PC è stata completata
4. Si conosce lo stato d'esecuzione dell'istruzione puntata dal PC

Quindi:

- a. L'immagine rappresenta interrupt preciso, poichè l'istruzione prima di quella puntata dal PC è stata completata, mentre quelle successive devono ancora essere completate.
- b. L'immagine rappresenta interrupt preciso, l'istruzione puntata dal PC può essere sia già eseguita, sia in fase di esecuzione.
- c. L'immagine rappresenta interrupt impreciso, poichè l'istruzione prima di quella puntata dal PC non è stata ancora completata.



- d. L'immagine rappresenta interrupt impreciso, poichè le istruzioni successive al PC sono già state completate.

## 1.16 CHS

Si consideri uno schema di indirizzamento CHS in cui sono utilizzati 14 bit per il numero di cilindri, 5 bit per il numero di testine, e 11 bit per il numero di settori. Si converta l'indirizzo LBA 248209144 in notazione CHS (C,H,S).

Formule per tradurre da LBA a CHS (*n.d.r: div è la divisione intera, mod il resto della divisione intera*):

- $C = LBA \text{ div } (N_h * N_s)$
- $H = (LBA \text{ div } N_s) \text{ mod } N_h$
- $S = (LBA \text{ mod } N_s) + 1$

I cilindri sono  $2^{14} = 16384$ , le testine sono  $2^5 = 32$ , i settori sono  $2^{11} = 2048$ . Quindi:

- $C = 248209144 \text{ div } (32 * 2048) = 3787$
- $H = (248209144 \text{ div } 2048) \text{ mod } 32 = 11$
- $S = (248209144 \text{ mod } 2048) + 1 = 1785$

Indirizzo in notazione CHS: (3787, 11, 1785).

## 1.17 Dischi

Si consideri uno disco caratterizzato da una velocità rotazionale pari a 10000 RPM, un tempo medio di seek pari a 5 msec, un numero medio di settori per traccia pari a 500. Determinare:

1. Latenza rotazionale media
2. Tempo medio di trasferimento
3. Tempo medio di posizionamento
4. Tempo medio di accesso a un settore

I risultati devono essere espressi in msec.

1. Latenza Rotazionale media = latenza rotazionale massima / 2 =  $(1 / \text{velocità di rotazione}) / 2 = \frac{1}{10000} * 60 * 1000 * \frac{1}{2} = 3 \text{ msec}$
2. Tempo medio di trasferimento = latenza rotazionale massima / media settori per traccia =  $\frac{1}{10000} * 60 * 1000 * \frac{1}{500} = 0.012 \text{ msec}$
3. Tempo medio di posizionamento = tempo medio di seek + latenza rotazionale media =  $5 + 3 = 8 \text{ msec}$
4. Tempo medio di accesso a un settore = tempo medio di seek + latenza rotazionale media + tempo medio trasferimento =  $5 + 3 + 0.012 = 8.012 \text{ msec}$

## 1.18 Sector interleaving

Si calcoli il valore del sector interleaving per un disco caratterizzato da una velocità rotazionale pari a 10000 RPM, un numero di settori per traccia pari a 1500, e un tempo di trasferimento di un settore in memoria (incluso il tempo di controllo dell'ECC) pari a 40  $\mu$ sec.

Formula per il calcolo del sector interleaving:  $\lceil \frac{\text{tempo trasferimento settore}}{\text{intervallo settori}} \rceil$ .

Tempo intervallo settori = tempo medio trasferimento = latenza rotazionale massima / media settori per traccia =  $\frac{1}{10000} * 60 * 1000 * \frac{1}{1500} = 0.004 \text{ msec} = 4 \mu\text{sec}$

Quindi:  $\lceil \frac{40}{4} \rceil = 10$  settori

## 2 File Systems

### 2.1 Path

(Riferirsi al pdf degli esercizi per l'immagine)

Si supponga che si voglia accedere al file foo.txt e che la directory corrente sia “/a/d” Dati i seguenti path:

- a. /a/b/c/foo.txt
- b. ./foo.txt
- c. /b/./a/c/foo.txt
- d. /a/c/foo.txt
- e. /a/c/./foo.txt
- f. ../c/foo.txt
- g. /b/./a/./b/./a/d/./c/foo.txt
- h. .././a/c/foo.txt

Rispondere alle seguenti domande:

- 1. Quali dei suddetti path sono validi?
- 2. Quali tra i path validi sono assoluti, relativi, o canonici?

I path validi sono:

- c. assoluto
- d. canonico
- e. assoluto
- f. relativo
- g. assoluto
- h. relativo

## 2.2 Allocazione

Si consideri un file system installato su un disco da 1TiB. Si supponga che il file system utilizzi blocchi di 2 KiB e indirizzi a blocchi a 64 bit, e si consideri un file “foo.txt” cui sono stati allocati i seguenti blocchi di dati (elencati in ordine di allocazione): 1080, 1081, 1082, 7400, 7401, 7407, 12510, 22418, 22419, 22420, 23710, 1079.

- a. Si disegni la struttura dati utilizzata per tenere traccia di tali blocchi nei seguenti casi:
  1. Allocazione basata su lista concatenata di blocchi.
  2. Allocazione basata su lista concatenata con File Allocation Table (FAT; si usi -1 come codice di terminazione di lista).
  3. Allocazione indicizzata basata su i-node con 4 blocchi diretti, 1 puntatore indiretto singolo e 1 puntatore indiretto doppio (si usi -1 per indicare un campo della struttura non utilizzato). Il blocco contenente l'i-node ha indirizzo 512.
- b. Si dica qual è la dimensione massima di un file nei tre casi del punto (a).
- c. Si dica quali informazioni occorre memorizzare nella entry di directory del file nei tre casi del punto (a)

- b.
  1. Con lista concatenata è possibile sfruttare l'intero disco, quindi la dimensione è limitata dalla capacità del disco/partizione su cui è installato il file system. Tuttavia non è possibile sfruttare l'intero spazio a disposizione in quanto ogni blocco ha un overhead dovuto al puntatore al nodo successivo della lista.
$$S_{totale} - S_{puntatori} = 2^{40} - \lfloor 2^{40}/2^{11} \rfloor * 8$$
  2. Il limite della dimensione del file è dato dalla capacità del disco/partizione dove è installato il file system. Non è possibile sfruttare l'intero spazio a disposizione in quanto parte dello spazio è riservato per la FAT.
$$S_{totale} - S_{FAT} = 2^{40} - \lfloor 2^{40}/2^{11} \rfloor * 8$$
  3. Il limite è dato da:
$$N_{pDiretti} + N_{pIndiretti} + N_{pIndiretti}^2 * D_{blocco} = 4 + \lfloor 2^{11}/2^3 \rfloor + \lfloor 2^{11}/2^3 \rfloor^2 * 2 = 131592KiB$$
- c. Per gli schemi di allocazione basati su lista basta memorizzare, oltre al nome del file, il puntatore al primo blocco.  
Per lo schema indicizzato con i-node, è sufficiente memorizzare, oltre al nome del file, il puntatore al blocco contenente l'i-node.

## 2.3 fsck

Si consideri la procedura fsck per la verifica della consistenza nei file system Unix e si esamini il contenuto delle seguenti tabelle:

	0	1	2	3	4	5	6	7	8	9
Blocchi in uso	1	0	1	0	1	3	0	0	1	0
Blocchi liberi	0	2	0	1	0	0	1	1	0	0

Si indichi quali sono le situazioni d'inconsistenza (se ce ne sono) e, per ognuna di esse, si dica come *fsck* riporta il file system in uno stato consistente. Si assuma che il file system tenga traccia dei blocchi liberi tramite una lista concatenata.

I tipi di inconsistenza sono:

1. blocco con contatore a 0 in entrambe le tabelle, *fsck* lo aggiunge alla tabella dei blocchi liberi
2. blocco con contatore a 1 in entrambe le tabelle, *fsck* può rimuoverlo da uno delle tabelle
3. blocco con contatore  $> 1$  nella tabella dei blocchi liberi, *fsck* ricostruisce la tabella dei blocchi liberi per eliminare l'inconsistenza
4. blocco con contatore  $> 1$  nella tabella dei blocchi in uso, *fsck* alloca un blocco libero, copia il contenuto del blocco duplicato in quel blocco e inserisce la copia in uno dei file

In questo caso:

1. la voce ad indice 9
2. (nessuna inconsistenza di questo tipo)
3. la voce ad indice 1
4. la voce ad indice 5

## 2.4 Cache

Si consideri la cache di un file system che utilizza la tecnica di hashing per velocizzare il reperimento dei blocchi in essa presenti, e la politica LRU per il rimpiazzamento dei blocchi. Si ipotizzi inoltre che tale cache preveda l'uso di 8 "slot" per la tabella di hash, indirizzati da 0 a 7, e che la funzione di hashing utilizzata sia  $h(B) = B \bmod 8$ , dove  $B$  è l'indirizzo del blocco da memorizzare nella cache.

- Si disegni lo stato della cache dopo l'inserimento dei seguenti blocchi: 4, 20, 44, 244, 9, 17, 41, 57, 159, 231.
- Partendo dal punto precedente, si disegni lo stato della cache dopo l'inserimento del blocco 347.

Si supponga che:

- la dimensione massima della cache sia di 10 blocchi,
- l'ordine di arrivo dei blocchi sia da sinistra verso destra (ad es., nel punto (a), il primo blocco ad arrivare è il blocco 4),
- la cache sia inizialmente vuota.

Nei diagrammi, indicare quale elemento rappresenta l'elemento LRU e quale quello MRU.

Valori hash: 4, 4, 4, 4, 1, 1, 1, 1, 7, 7

## 2.5 I-node

Si consideri un disco di 512 GiB e un file system installato su di esso che impieghi gli i-node come metodo di allocazione di blocchi ai file e una bitmap per la gestione dei blocchi liberi. Supponendo di utilizzare blocchi di 4 KiB ed indirizzi a 32 bit, si risponda alle seguenti domande:

- Qual è la dimensione massima di un file in tale file system se gli i-node includono 16 puntatori diretti, un puntatore indiretto singolo, 2 puntatori indiretti doppi e 2 puntatori indiretti tripli?
- Qual è la dimensione della bitmap per la gestione dei blocchi liberi? Per semplicità, si assuma che l'intero disco possa essere utilizzato per memorizzare blocchi di dati.

$$\text{Dimensione disco} = 2^{39}$$

$$\text{Dimensione blocco} = 2^{12}$$

$$\text{Numero blocchi} = \lfloor 2^{39}/2^{12} \rfloor = 2^{27}$$

$$\text{Numero indirizzi per blocco} = \lfloor 2^{12}/2^2 \rfloor = 2^{10} = 1024$$

$$\text{Numero massimo blocchi indirizzabili} = N_{pDiretti} + N_{pIndiretti} + N_{pIndiretti}^2 + N_{pIndiretti}^3 = 16 + 1024 + 1024^2 * 2 + 1024^3 * 2$$

$$\text{Dimensione bitmap} = \lfloor 2^{39}/2^{12} \rfloor = 2^{27} \text{ bit}$$

## 2.6 Formattazione

Si vuole formattare un disco da 512 GiB con un particolare file system usando blocchi da 4 KiB. Determinare quali delle seguenti scelte progettuali sono valide per rappresentare tutti i blocchi del disco:

- Indirizzi di blocchi a 64 bit
- Indirizzi di blocchi a 16 bit

Limitandosi alle scelte progettuali corrette e supponendo di dover memorizzare un file composto da 32 KiB di dati, determinare overhead e spazio sprecato nel caso si utilizzi:

- l'allocazione basata su FAT,
- l'allocazione indicizzata con blocco indice da 4 KiB,
- l'allocazione basata su lista concatenata con cluster da 16 KiB.

Il numero di blocchi totale è  $2^{27}$ , quindi non è possibile rappresentarli con indirizzi a 16 bit, servono indirizzi a 64 bit.

- Overhead =  $(D_{FAT}/D_{disco})\% = (2^{30}/2^{39})\% = 0.19\%$ , spazio sprecato =  $(1 - (D_{file}/(N_{blocchi} * D_{blocco}))) = 0\%$
- Overhead =  $(N_{blocchiIndexBlock}/N_{blocchiFile})\% = (1/9)\% = 11\%$ , spazio sprecato =  $(1 - (2^3 * 2^3 + 2^{15})/(9 * 2^{12}))\% = 11\%$
- Overhead =  $(2^3/(2^2 * 2^{12}))\% = 0.05\%$ , spazio sprecato =  $(1 - (2 * 2^3 + 2^{12})/(2^{15} * 2^{12}))\% = 0\%$