

Esercizio Dispositivi-1

In un disco con 4 facce, 30 settori per traccia e 120 cilindri, il periodo di rotazione è di 3 msec: conseguentemente il tempo impiegato per percorrere un settore è di 0,1 msec. Ogni settore contiene un blocco di 1024 byte.

Al tempo t termina l'esecuzione di un'operazione sul settore 20 del cilindro 45 e, mentre questa operazione era in corso, sono pervenuti in rapida successione i seguenti comandi di lettura, relativi a blocchi logicamente e fisicamente consecutivi:

1. settore 5 della faccia 0 del cilindro 45
2. settore 6 della faccia 0 del cilindro 45
3. settore 7 della faccia 0 del cilindro 45
4. settore 8 della faccia 0 del cilindro 45
5. settore 9 della faccia 0 del cilindro 45
6. settore 10 della faccia 0 del cilindro 45.

Questi comandi vengono eseguiti a partire dal tempo t , nell'ordine che minimizza il tempo complessivo di esecuzione. Si suppone che al tempo t non siano pendenti altri comandi e che nei successivi 20 msec non ne vengano ricevuti altri.

Si chiede:

- a. l'ordine di esecuzione dei comandi da 1 a 6;
- b. il tempo di inizio della lettura di ciascun blocco (inteso come il tempo in cui inizia il trasferimento dei dati nel buffer);
- c. il tempo complessivo impiegato per eseguire tutte le letture, nelle seguenti ipotesi:

Ipotesi 1: Il controllore di DMA dispone di un unico buffer della capacità di un blocco. In ogni operazione di lettura, il buffer viene caricato con il contenuto di un settore e il suo contenuto viene quindi trasferito in memoria. Questo trasferimento inizia quando si inizia a percorrere il settore successivo a quello appena letto e richiede 0,06 msec;

Ipotesi 2: Il controllore di DMA dispone di due buffer, *Buffer0* e *Buffer1*, della capacità di un blocco, che vengono utilizzati a rotazione nella *fase pari* e nella *fase dispari* (nella *fase pari*, *Buffer0* riceve i dati dal settore corrente mentre *Buffer1* trasferisce il suo contenuto in memoria; nella *fase dispari*, *Buffer1* riceve i dati dal settore corrente mentre *Buffer0* trasferisce il suo contenuto in memoria). Il trasferimento di un blocco da un buffer alla memoria richiede 0,06msec.

Soluzione

Ipotesi 1

Settore	Inizio lettura: $t+$	Fine lettura	Inizio trasf. in memoria	Fine trasf. in memoria
5	$(35 - 21) \cdot 0,1 = 1,4$	1,5	1,5	1,56
7	$1,5 + 0,1 = 1,6$	1,7	1,7	1,76
9	$1,7 + 0,1 = 1,8$	1,9	1,9	1,96
6	$1,5 + 3 = 4,5$	4,6	4,6	4,66
8	$4,6 + 0,1 = 4,7$	4,8	4,8	4,86
10	$4,8 + 0,1 = 4,9$	5	5	5,06

- a. Ordine di esecuzione: 5, 7, 9, 6, 8, 10;
- b. Tempo di inizio della lettura: settore 5: $t+1,4$; settore 7: $t+1,6$; settore 9: $t+1,8$; settore 6: $t+4,5$; settore 8: $t+4,7$; settore 10: $t+4,9$;
- c. Tempo complessivo: 5,06 msec.

Ipotesi 2

Settore	Inizio lettura: $t+$	Fine lettura	Inizio trasf. in memoria	Fine trasf. in memoria
5	$(35 - 21) \cdot 0,1 = 1,4$	1,5	1,5	1,56
6	1,5	1,6	1,6	1,66
7	1,6	1,7	1,7	1,76
8	1,7	1,8	1,8	1,86
9	1,8	1,9	1,9	1,96
10	1,9	2	2	2,06

- a. Ordine di esecuzione: 5, 6, 7, 8, 9, 10;
- b. Tempo di inizio della lettura: settore 5: $t+1,4$; settore 6: $t+1,5$; settore 7: $t+1,6$; settore 8: $t+1,7$; settore 9: $t+1,8$; settore 10: $t+1,9$;
- c. Tempo complessivo: 2,06 msec.

Esercizio Dispositivi-2

Un disco è organizzato con $NCilindri = 50$ (numerati da 0 a 49), $NFacce = 4$ (numerate da 0 a 3) e $NSettori = 20$ (numerati da 0 a 19). Ogni settore contiene 1024 byte.

A livello logico i blocchi del disco corrispondono a settori e sono individuati con *indici di blocco*, interi compresi nell'intervallo $[0, capacità_in_blocchi)$.

Gli indici di blocco sono definiti secondo la sequenza *cilindro, faccia, settore* (ad esempio, il blocco 22 corrisponde al settore 2 della faccia 1 del cilindro 0).

Si chiede:

- 1) la capacità del disco, in numero di blocchi e di byte
- 2) l'indice di blocco corrispondente all'indirizzo fisico (c, f, s), con c= 39, f= 2, s= 10)
- 3) la terna corrispondente al blocco 1525

Soluzione

- 1) Ogni cilindro contiene $4 * 20 = 80$ blocchi;

la capacità del disco è : $50 * 80 \text{ blocchi} = 4000 \text{ KByte} \sim 3,9 \text{ MByte}$

- 2) Dalla formula: $b = c * (NFacce * NSettori) + f * NSettori + s$,

l'indice di blocco corrispondente all'indirizzo (39, 2, 10) è:

$$b = 39 * 4 * 20 + 2 * 20 + 10 = 3170$$

- 3) Dalle formule:

$$c = b \text{ div } (NFacce * NSettori);$$

$$f = (b \text{ mod } (NFacce * NSettori)) \text{ div } NSettori;$$

$$s = (b \text{ mod } (NFacce * NSettori)) \text{ mod } NSettori ,$$

la terna (c, f, s) corrispondente all'indice 1525 è:

$$c = 1525 \text{ div } 80 = 19;$$

$$f = (1525 \text{ mod } 80) \text{ div } 20 = 5 \text{ div } 20 = 0;$$

$$s = (1525 \text{ mod } 80) \text{ mod } 20 = 5$$

Esercizio Dispositivi-3

Un disco con 4 facce, 30 settori per traccia e 120 cilindri ha un tempo di seek proporzionale al numero di cilindri attraversati e pari a 0,5ms per ogni cilindro. Il periodo di rotazione è di 3 msec: conseguentemente il tempo impiegato per percorrere un settore è di 0,1 msec.

A un certo tempo (convenzionalmente indicato come $t=0$) termina l'esecuzione dei comandi sul cilindro 58 e sono pervenute, nell'ordine, le seguenti richieste di lettura o scrittura:

- settore 5 e settore 15 della faccia 0 del cilindro 45
- settore 10 della faccia 2 del cilindro 65 e settore 25 della faccia 3 del cilindro 65
- settore 6 della faccia 1 del cilindro 101
- settore 18 della faccia 3 del cilindro 7
- settore 8 della faccia 1 del cilindro 110

Al tempo 60 arriva un ulteriore comando di lettura per il settore 15 della faccia 0 del cilindro 20.

Calcolare il tempo necessario per eseguire tutte queste operazioni supponendo i comandi pendenti per uno stesso cilindro devono essere eseguiti nell'ordine in cui sono stati ricevuti e che per i comandi relativi a cilindri diversi si adottino le seguenti politiche di scheduling:

- a) politica FIFO
- b) politica Shortest Seek Time First (SSTF)
- c) politica SCAN, con fase di discesa attiva al tempo 0.

Il tempo di esecuzione di ogni operazione è uguale alla somma dell'eventuale tempo di *seek*, del ritardo rotazionale (tempo necessario per raggiungere il settore indirizzato) e del tempo di percorrenza del settore indirizzato. Il controllore è dotato di sufficiente capacità di buffering ed è sempre in grado di eseguire senza ritardo comandi relativi a settori consecutivi dello stesso cilindro.

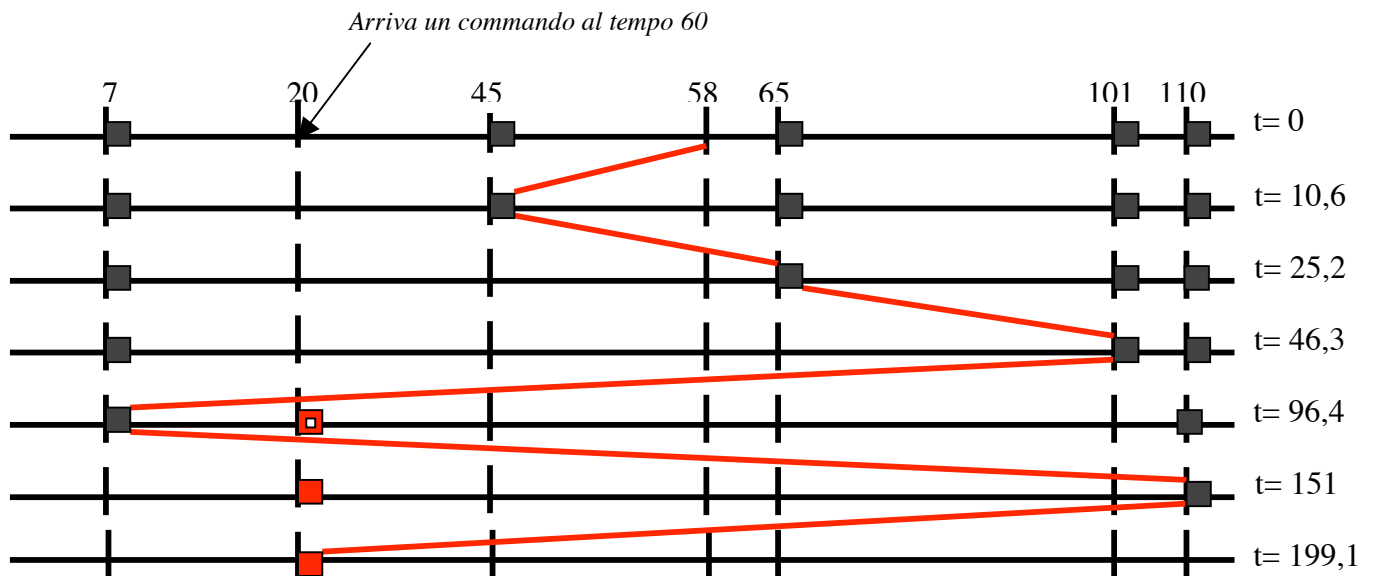
Per il ritardo rotazionale dopo un'operazione di *seek* si assume sempre il valore di caso peggiore, pari a un periodo di rotazione (3 msec).

a) Soluzione con Politica FIFO

Al tempo 0 termina l'esecuzione dei comandi sul cilindro 58 e sono pervenute, nell'ordine, le seguenti richieste di lettura o scrittura:

- settore 5 e settore 15 della faccia 0 del cilindro 45
- settore 10 della faccia 2 del cilindro 65 e settore 25 della faccia 3 del cilindro 65
- settore 6 della faccia 1 del cilindro 101
- settore 18 della faccia 3 del cilindro 7
- settore 8 della faccia 1 del cilindro 110

Al tempo 60 arriva un ulteriore comando di lettura per il settore 15 della faccia 0 del cilindro 20.



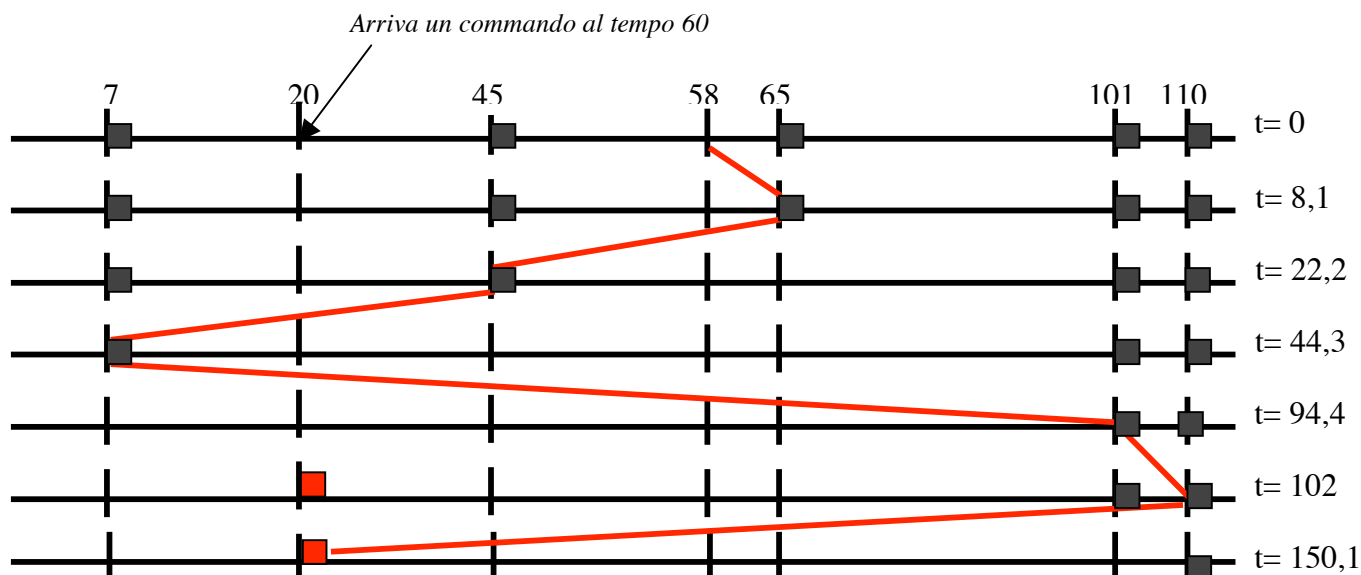
op. su cilindro:	45	settore:	5	faccia :	0				
inizio:	0	seek:	$(58-45)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	9,6
op. su cilindro:	45	settore:	15	faccia :	0				
inizio:	9,6	seek:	0	rotazione:	$(15-6)*0,1$	percorrenza:	0,1	fine:	10,6
op. su cilindro:	65	settore:	10	faccia :	2				
inizio:	10,6	seek:	$(65-45)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	23,7
op. su cilindro:	65	settore:	25	faccia :	3				
inizio:	23,7	seek:		rotazione:	$(25-11)*0,1$	percorrenza:	0,1	fine:	25,2
op. su cilindro:	101	settore:	6	faccia :	1				
inizio:	25,2	seek:	$(101-65)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	46,3
op. su cilindro:	7	settore:	18	faccia :	3				
inizio:	46,3	seek:	$(101-7)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	96,4
Al tempo $t=60$ arrivato comando per cilindro 20									
op. su cilindro:	110	settore:	8	faccia :	1				
inizio:	96,4	seek:	$(110-7)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	151
op. su cilindro:	20	settore:	15	faccia :	0				
inizio:	151	seek:	$(110-20)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	199,1

b) Soluzione con Politica SSTF

Al tempo 0 termina l'esecuzione dei comandi sul cilindro 58 e sono pervenute, nell'ordine, le seguenti richieste di lettura o scrittura:

- settore 5 e settore 15 della faccia 0 del cilindro 45
- settore 10 della faccia 2 del cilindro 65 e settore 25 della faccia 3 del cilindro 65
- settore 6 della faccia 1 del cilindro 101
- settore 18 della faccia 3 del cilindro 7
- settore 8 della faccia 1 del cilindro 110

Al tempo 60 arriva un ulteriore comando di lettura per il settore 15 della faccia 0 del cilindro 20.



op. su cilindro:	65	settore:	10	faccia :	2				
inizio:	0	seek:	$(65-58)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	6,6
op. su cilindro:	65	settore:	25	faccia :	3				
inizio:	6,6	seek:		rotazione:	$(25-11)*0,1$	percorrenza:	0,1	fine:	8,1
op. su cilindro:	45	settore:	5	faccia :	0				
inizio:	8,1	seek:	$(65-45)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	21,2
op. su cilindro:	45	settore:	15	faccia :	0				
inizio:	21,2	seek:	0	rotazione:	$(15-6)*0,1$	percorrenza:	0,1	fine:	22,2
op. su cilindro:	7	settore:	18	faccia :	3				
inizio:	22,2	seek:	$(45-7)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	44,3
op. su cilindro:	101	settore:	6	faccia :	1				
inizio:	44,3	seek:	$(101-7)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	94,4
Al tempo t= 60 arrivato comando per cilindro 20									
op. su cilindro:	110	settore:	8	faccia :	1				
inizio:	94,4	seek:	$(110-101)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	102
op. su cilindro:	20	settore:	15	faccia :	0				
inizio:	102	seek:	$(110-20)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	150,1

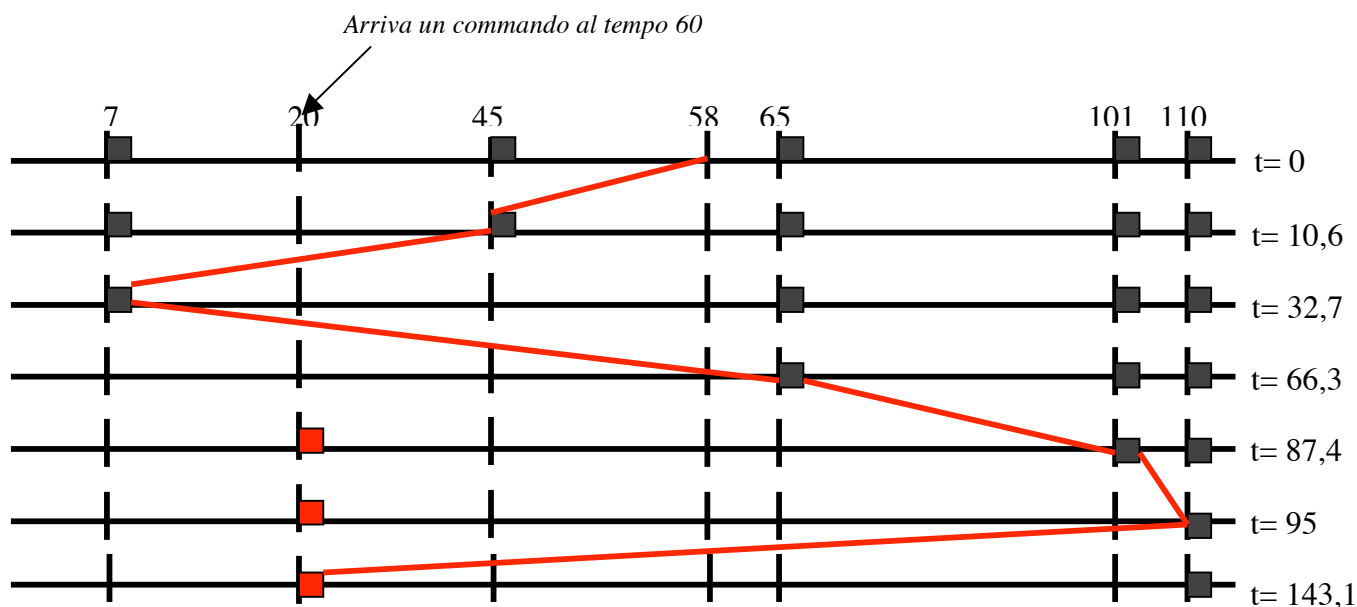
c) Soluzione con Politica SCAN

Al tempo 0 termina l'esecuzione dei comandi sul cilindro 58 e sono pervenute, nell'ordine, le seguenti richieste di lettura o scrittura:

- settore 5 e settore 15 della faccia 0 del cilindro 45
- settore 10 della faccia 2 del cilindro 65 e settore 25 della faccia 3 del cilindro 65
- settore 6 della faccia 1 del cilindro 101
- settore 18 della faccia 3 del cilindro 7
- settore 8 della faccia 1 del cilindro 110

Al tempo 60 arriva un ulteriore comando di lettura per il settore 15 della faccia 0 del cilindro 20.

Al tempo 0 è attiva la fase di discesa.



op. su cilindro:	45	settore:	5	faccia :	0				
inizio:	0	seek:	$(58-45)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	9,6
op. su cilindro:	45	settore:	15	faccia :	0				
inizio:	9,6	seek:	0	rotazione:	$(15-6)*0,1$	percorrenza:	0,1	fine:	10,6
op. su cilindro:	7	settore:	18	faccia :	3				
inizio:	10,6	seek:	$(45-7)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	32,7
op. su cilindro:	65	settore:	10	faccia :	2				
inizio:	32,7	seek:	$(65-7)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	64,8
Al tempo t= 60 arrivato comando per cilindro 20									
op. su cilindro:	65	settore:	25	faccia :	3				
inizio:	64,8	seek:	0	rotazione:	$(25-11)*0,1$	percorrenza:	0,1	fine:	66,3
op. su cilindro:	101	settore:	6	faccia :	1				
inizio:	66,3	seek:	$(101-65)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	87,4
op. su cilindro:	110	settore:	8	faccia :	1				
inizio:	87,4	seek:	$(110-101)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	95
op. su cilindro:	20	settore:	15	faccia :	0				
inizio:	95	seek:	$(110-20)*0,5$	rotazione:	3	percorrenza:	0,1	fine:	143,1

Esercizio Dispositivi-4

Un disco con 2 facce, 200 settori per traccia e 1000 cilindri ha un tempo di seek pari ad un valore costante di 40 msec più 0,1 msec per ogni cilindro attraversato. Alla conclusione di ogni operazione di seek le testine del disco sono posizionate all'inizio del settore 0 del cilindro.

Il periodo di rotazione è di 20 msec: conseguentemente il tempo impiegato per percorrere un settore è di 0,1 msec.

A un certo tempo (convenzionalmente indicato come $t=0$) termina l'esecuzione dei comandi sul cilindro 860 e sono pervenute, nell'ordine, le seguenti richieste di lettura o scrittura:

- cilindro 890, faccia 0, settore 55
- cilindro 630, faccia 0, settore 100
- cilindro 47, faccia 1, settore 120
- cilindro 850, faccia 1, settore 189
- cilindro 345, faccia 1, settore 5
- cilindro 930, faccia 0, settore 76

Al tempo 40 arrivano anche le seguenti richieste:

- cilindro 630, faccia 0, settore 88
- cilindro 47, faccia 0, settore 50

Calcolare il tempo necessario per eseguire tutte queste operazioni supponendo che si adotti la politica di scheduling SCAN e supponendo che al tempo 0 il disco sia in fase di discesa.

Il tempo di esecuzione di ogni operazione è uguale alla somma dell'eventuale tempo di *seek*, del ritardo rotazionale (tempo necessario per raggiungere il settore indirizzato) e del tempo di percorrenza del settore indirizzato.

Il controllore è dotato di sufficiente capacità di buffering ed è sempre in grado di accettare senza ritardo i dati letti dal disco o quelli da scrivere sul disco. Si assume inoltre che i comandi sullo stesso cilindro vengano eseguiti nell'ordine che minimizza il ritardo rotazionale.

SOLUZIONE

Tempi	op. su cilindro:	850	settore:	189				
	inizio:	0	seek:	41	rotazione:	18,9	percorrenza:	0,1
							fine:	60
Tempi	op. su cilindro:	630	settore:	88				
	inizio:	60	seek:	62	rotazione:	8,8	percorrenza:	0,1
							fine:	130,9
Tempi	op. su cilindro:	630	settore:	100				
	inizio:	130,9	seek:	0	rotazione:	1,1	percorrenza:	0,1
							fine:	132,1
Tempi	op. su cilindro:	345	settore:	5				
	inizio:	132,1	seek:	68,5	rotazione:	0,5	percorrenza:	0,1
							fine:	201,2
Tempi	op. su cilindro:	47	settore:	50				
	inizio:	201,2	seek:	69,8	rotazione:	5	percorrenza:	0,1
							fine:	276,1
Tempi	op. su cilindro:	47	settore:	120				
	inizio:	276,1	seek:	0	rotazione:	6,9	percorrenza:	0,1
							fine:	283,1
Tempi	op. su cilindro:	890	settore:	55				
	inizio:	283,1	seek:	124,3	rotazione:	5,5	percorrenza:	0,1
							fine:	413
Tempi	op. su cilindro:	930	settore:	76				
	inizio:	413	seek:	44	rotazione:	7,6	percorrenza:	0,1
							fine:	464,7

Esercizio Dispositivi-5

In un disco RAID livello 0 il disco virtuale V ha la capacità di 2^{10} blocchi di 1 kByte, numerati da 0 a $2^{10}-1$, ed è realizzato mediante 4 dischi fisici, D(0), D(1), D(2) e D(3), ciascuno della capacità di 2^8 blocchi di 1 kByte, numerati da 0 a 2^8-1 . Il blocco b del disco V è mappato nel blocco $b \text{ div } 4$ del disco fisico di indice $b \bmod 4$.

Supponendo che vengano eseguite le seguenti operazioni, ciascuna delle quali interessa più blocchi consecutivi del disco virtuale:

- Lettura dei blocchi virtuali 66, 67, 68;
- Scrittura dei blocchi virtuali 132, 133, 134, 135;
- Lettura dei blocchi virtuali 250, 251, 252, 253, 254, 255, 256,

Si chiede, per ciascuna operazione:

- Quali sono i dischi fisici e i relativi blocchi fisici interessati;
- Per eseguire l'operazione, qual è il massimo numero di accessi per disco fisico.

Soluzione

- a) Lettura dei blocchi virtuali 66, 67, 68

Dalle formule $DiscoFisico = b \bmod 4$, $BloccoFisico = b \text{ div } 4$, i dischi fisici e i relativi blocchi fisici interessati sono:

Blocco virtuale	Disco fisico	Blocco fisico
66	2	16
67	3	16
68	0	17

Massimo numero di accessi per disco fisico: 1

- b) Scrittura dei blocchi virtuali 132, 133, 134, 135

Dalle formule $DiscoFisico = b \bmod 4$, $BloccoFisico = b \text{ div } 4$, i dischi fisici e i relativi blocchi fisici interessati sono:

Blocco virtuale	Disco fisico	Blocco fisico
132	0	33
133	1	33
134	2	33
135	3	33

Massimo numero di accessi per disco fisico: 1

- c) Lettura dei blocchi virtuali 250, 251, 252, 253, 254, 255, 256

Dalle formule $DiscoFisico = b \bmod 4$, $BloccoFisico = b \text{ div } 4$, i dischi fisici e i relativi blocchi fisici interessati sono:

Blocco virtuale	Disco fisico	Blocco fisico
250	2	62
251	3	62
252	0	63
253	1	63
254	2	63
255	3	63
256	0	64

Massimo numero di accessi per disco fisico: 2

Esercizio Dispositivi-6

Un disco RAID di livello 4 è composto da 5 dischi fisici, numerati da 0 a 4. Le strip corrispondono a blocchi.

Il disco 4 è ridondante e il suo blocco di indice i contiene la parità dei blocchi di indice i dei dischi non ridondanti, cioè dei dischi 0, 1, 2 e 3.

I blocchi di indice 5 dei dischi non ridondanti contengono rispettivamente:

Disco 0: 0 1 1 0 1 0 0 1

Disco 1: 0 1 0 0 1 1 0 1

Disco 2: 1 1 1 0 1 1 0 0

Disco 3: 0 1 1 1 1 0 0 1

Si chiede;

- 1) il contenuto del blocco di indice 5 del disco ridondante
- 2) come cambia il contenuto del blocco di indice 5 del disco ridondante se in seguito a una scrittura il contenuto del blocco 5 del disco 2 diviene 0 0 1 1 0 0 1 0

Soluzione

1) Contenuto iniziale del blocco di indice 5 del disco ridondante

Disco 0	0	1	1	0	1	0	0	1
Disco 1	0	1	0	0	1	1	0	1
Disco 2	1	1	1	0	1	1	0	0
Disco 3	0	1	1	1	1	0	0	1
Disco 4	1	0	1	1	0	0	0	1

2) Contenuto del blocco di indice 5 del disco ridondante dopo la scrittura nel blocco 5 del disco 2

Prima della scrittura:

Disco 0	0	1	1	0	1	0	0	1
Disco 1	0	1	0	0	1	1	0	1
Disco 2	1	1	1	0	1	1	0	0
Disco 3	0	1	1	1	1	0	0	1
Disco 4	1	0	1	1	0	0	0	1

Dopo la scrittura:

Disco 0	0	1	1	0	1	0	0	1
Disco 1	0	1	0	0	1	1	0	1
Disco 2	0	0	1	1	0	0	1	0
Disco 3	0	1	1	1	1	0	0	1
Disco 4	0	1	1	0	1	1	1	1

Esercizio Dispositivi-7

Un disco RAID di livello 4 è composto da 4 dischi fisici, numerati da 0 a 3. Le strip corrispondono a blocchi.

Il disco 3 è ridondante e il suo blocco di indice i contiene la parità dei blocchi di indice i dei dischi non ridondanti, cioè dei dischi 0, 1 e 2.

Al tempo t i contenuti dei blocchi di indice 7 dei dischi fisici sono i seguenti:

Disco 0: 1 1 0 0

Disco 1: 1 0 0 1

Disco 2: 1 1 1 1

Disco 3: 1 0 1 0

Al tempo $t+1$ si esegue una scrittura, per effetto della quale il contenuto del blocco 7 del disco fisico 2 diviene 0 0 1 1. Si chiede:

- 1) Quali sono i blocchi i cui contenuti vengono modificati per effetto della scrittura
- 2) Dopo la scrittura, qual è il contenuto dei blocchi di indice 7 di tutti i blocchi fisici
- 3) Quali blocchi è necessario leggere per eseguire la scrittura

Soluzione

- 1) Oltre al blocco di indice 7 del disco 2 si modifica anche il blocco 7 del disco 3, per l'aggiornamento della parità.
- 2) Dopo la scrittura, i blocchi di indice 7 dei 4 dischi fisici contengono:

Disco 0: 1 1 0 0 (invariato)

Disco 1: 1 0 0 1 (invariato)

Disco 2: 0 0 1 1 (modificato)

Disco 3: 0 1 1 0 (modificato)

- 3) Le letture sono necessarie per ricalcolare la parità.

E' necessario leggere (prima della scrittura) i blocchi di indice 7 dei dischi 2 e 3 oppure, in alternativa, i blocchi di indice 7 dei dischi 0 e 1