

Input Output Control System

Sistemi Operativi

Antonino Staiano

Email: antonino.staiano@uniparthenope.it

Livelli IOCS

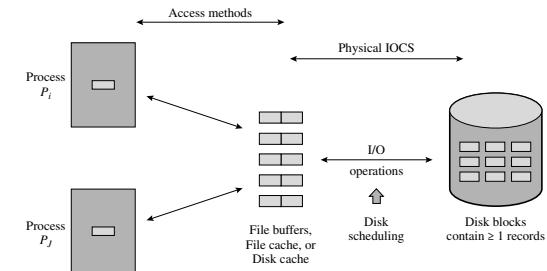


Figure 14.1 Implementation of file operations by the IOCS.

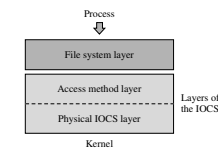
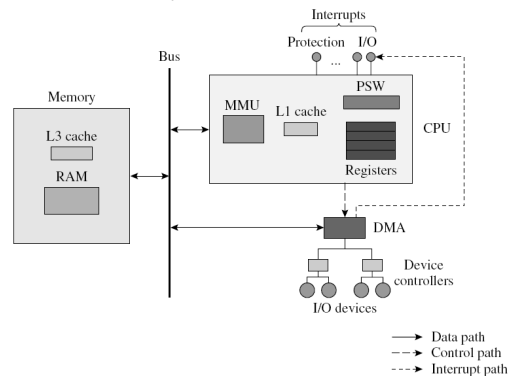


Figure 14.2 Layers of the file system and the IOCS.

Organizzazione dell'I/O



- Il sottosistema di I/O ha un percorso ai dati in memoria indipendente
- I dispositivi sono connessi ai controller dei dispositivi che sono connessi al DMA (Direct Memory Access)
 - un dispositivo è identificato dalla coppia (*controller id*, *device id*)
- Il DMA, un controller di dispositivo e il dispositivo implementano un'operazione di I/O

Operazioni di I/O

- Un'operazione di I/O coinvolge:
 - Operazione da eseguire: read, write ecc
 - Indirizzo del dispositivo di I/O
 - Numero di byte di dati da trasferire
 - Indirizzi delle aree in memoria e sul dispositivo di I/O coinvolte nel trasferimento
- La CPU avvia l'operazione di I/O mediante l'esecuzione di un'istruzione di I/O, ma non è coinvolta nel trasferimento
 - L'istruzione di I/O punta ad un insieme di comandi di I/O
 - Singole azioni coinvolte nel trasferimento dati
 - L'esecuzione di tali azioni è compito del DMA, del controller del dispositivo e del dispositivo di I/O
 - La CPU è libera di fare altro mentre l'operazione di I/O è in atto

Operazioni di I/O (cont.)

- Operazione di I/O *read* da un blocco del disco (*track_id*, *block_id*) eseguita mediante l'istruzione

I/O-init(controller_id, device_id), I/O_command_addr

Dove *I/O_command_addr* è l'indirizzo di partenza dell'area di memoria che contiene i seguenti comandi:

1. Posiziona la testina del disco sulla traccia *track_id*
2. Leggi il record *record_id* nell'area di memoria con indirizzo di partenza *memory_addr*

Third Party DMA

- Quando è eseguita un'istruzione di I/O
 - Il controller del DMA passa i dettagli dei comandi di I/O al controller del dispositivo di I/O
 - Il dispositivo consegna i dati al controller di dispositivo
 - Il trasferimento dei dati da controller del dispositivo a memoria avviene come segue
 - Il controller del dispositivo invia un segnale DMA request quando è pronto al trasferimento
 - Il DMA, ricevuto il segnale, ottiene il controllo del bus e vi pone l'indirizzo di memoria che partecipa la trasferimento. Infine, invia un DMA ack al controller del dispositivo
 - Il controller del dispositivo trasferisce i dati verso o dalla memoria
 - Alla fine del trasferimento, il controller del DMA genera un interrupt di completamento I/O con codice uguale all'indirizzo del dispositivo
 - La routine di servizio degli interrupt analizza il codice per trovare quale dispositivo ha completato la sua operazione di I/O e intraprende le azioni appropriate

Dispositivi di I/O

- Esistono differenti tipologie di dispositivi di I/O che funzionano sulla base di vari principi fisici
 - Generazione di segnali elettromeccanici
 - Memorizzazione dati ottica o elettromagnetica
- I dispositivi di I/O possono essere classificati sulla base dei seguenti criteri:
 - Scopo: dispositivi di input, di stampa e di memorizzazione
 - Natura dell'accesso
 - Sequenziale: tastiera, mouse, rete, nastro
 - Casuale: dischi
 - Modalità trasferimento dati: caratteri o blocchi
- L'informazione letta o scritta in un comando di I/O costituisce un record

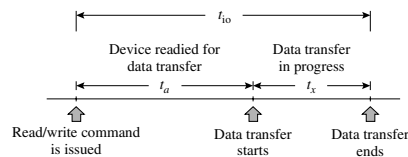
Dispositivi di I/O: modalità trasferimento dati

- Dipende dalla velocità di trasferimento
 - Dispositivo di I/O lento (tastiera, mouse e stampante sono dispositivi a carattere)
 - Lavora nella modalità carattere: è trasferito un carattere per volta tra memoria e periferica
 - Contiene un buffer che memorizza il carattere
 - Il controller genera un interrupt in conseguenza di una lettura dal buffer (dispositivo di input) o di una scrittura nel buffer (dispositivo di output)
 - I controller possono essere connessi direttamente al bus
 - Dispositivi di I/O veloci (nastri, dischi)
 - Lavora in modalità a blocco
 - Connesso ad un controller di DMA
 - Devono trasferire i dati a specifiche velocità
 - I dati sono trasferiti tra la periferica di I/O e un *buffer del DMA*

Dispositivi di I/O: tempo di accesso e tempo di trasferimento

- t_{io} (tempo di I/O) -> intervallo tra esecuzione istruzione inizio I/O e completamento operazione
- t_a (tempo di accesso) -> intervallo tra un comando read o write e l'inizio del trasferimento
- t_x (tempo di trasferimento) -> tempo necessario per trasferire dati da/verso una periferica durante un'operazione read o write (inizio trasferimento primo byte, fine trasferimento ultimo byte)

$$t_{io} = t_a + t_x$$



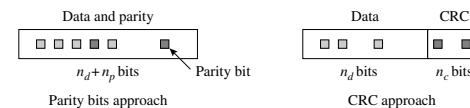
Dispositivi di I/O: Individuazione e correzione errori

- Gli errori possono verificarsi durante la scrittura o la lettura dei dati o durante il trasferimento tra un dispositivo di I/O e la memoria
- I dati trasmessi sono visti come flusso di bit
 - Usati codici speciali per rappresentarli
- Individuazione errori
 - Si memorizza informazioni ridondanti con i dati
 - Informazione di individuazione errori
 - Determinata dai dati con tecniche standard
 - Quando i dati sono letti da una periferica di I/O sono lette anche le informazioni di individuazione errori
 - Inoltre, tali informazioni sono calcolate nuovamente dai dati letti, usando la stessa tecnica
 - Si confrontano le info lette dal mezzo di I/O e quelle determinate dai dati letti
 - Un mismatch indica l'occorrenza di un errore in fase di memorizzazione
- La correzione dell'errore è fatta in modo analogo
 - Si usano algoritmi per determinare l'informazione di correzione
 - Tali informazioni possono sia individuare un errore che suggerire come correggerlo

Dispositivi di I/O: Individuazione e correzione errori (cont.)

- La memorizzazione e la lettura di informazioni ridondanti causa overhead
 - La correzione degli errori comporta maggiore overhead rispetto alla loro all'individuazione
- Approcci all'individuazione e correzione
 - Bit di parità
 - Sono calcolati n_p bit di parità da n_d bit di dati
 - Non distinguibili dai bit di dati se non all'algoritmo di individuazione/correzione
 - Controllo di ridondanza ciclico (CRC)
 - È memorizzato in un campo CRC di ogni record un numero (CRC) di n_c bit
 - n_c non dipende da n_d
- In ambo gli approcci si usa l'aritmetica modulo-2
 - L'addizione è rappresentata come un OR-esclusivo

Individuazione e correzione errori (cont.)



Calculating a parity bit

A parity bit is computed from a collection of data bits by modulo-2 arithmetic, i.e., by using the exclusive OR operator \oplus . For example, the parity bit for 4 data bits b_i, b_j, b_k and b_l is computed as follows: $p = b_i \oplus b_j \oplus b_k \oplus b_l \oplus c_1$, where c_1 is a constant which is 1 for *odd parity* and 0 for *even parity*.

Cyclic redundancy check (CRC)

Step 1: A bit stream is looked upon as a binary polynomial, i.e., a polynomial each of whose coefficients is either a 0 or a 1. For example, a bit stream 1101 is looked upon as a binary polynomial $1 \times x^3 + 1 \times x^2 + 0 \times x^1 + 1 \times x^0$, i.e., $x^3 + x^2 + 1$. Here a + is interpreted as modulo-2 addition, i.e., an exclusive-OR operation \oplus .

Step 2: The data in a received record is augmented by adding n_c zeroes at its end. The polynomial obtained from the augmented data is divided by a predefined polynomial of degree $n_c + 1$. The remainder of this division is a polynomial of degree n_c . Coefficients in this polynomial form the CRC. For example, the CRC for data 11100101 using a predefined 5-bit polynomial 11011 is 0100.

Step 3: When a record is received, the receiver computes the CRC from the data part of the record and compares it with the CRC part of the record. A mismatch indicates error(s). Alternatively, the receiver computes the CRC from the entire record. An error exists if the computed CRC is not 0.

Disco magnetico

- L'elemento di memorizzazione è un oggetto circolare chiamato piatto che ruota intorno al proprio asse
 - La superficie circolare è ricoperta di materiale magnetico
- Una singola testina di lettura-scrittura registra e legge dalla superficie
 - Un byte è memorizzato in modo seriale lungo una traccia circolare sulla superficie del disco
 - La testina può muoversi radialmente lungo il piatto
 - Per ogni posizione della testina, l'informazione registrata forma una traccia circolare separata
 - In un disco non è usata l'informazione di parità ma è scritto un CRC per il rilevamento errori
 - E' marcata la posizione di avvio su ogni traccia ed ai record di una traccia sono attribuiti numeri seriali rispetto a tale posizione
 - Il disco può accedere ad ogni record con indirizzo (*numero traccia, numero record*)

13

Disco magnetico (cont.)

- Il tempo di accesso è:
 - $t_a = t_s + t_r$
 - t_s tempo di ricerca, tempo per posizionare la testina sulla traccia richiesta
 - t_r latenza rotazionale, tempo per accedere il record desiderato sulla traccia
- La latenza rotazionale media è il tempo richiesto per una rivoluzione di metà del disco
 - 3-4 ms
- Maggiori capacità sono ottenute montando molti piatti
 - Una testina di lettura e scrittura per ogni superficie circolare del piatto
 - Una testina sopra ed una sotto
 - Tutte le testine sono montate su un singolo braccio (attuatore)
 - Tutte le testine sono posizionate sulle stesse tracce di superfici diverse

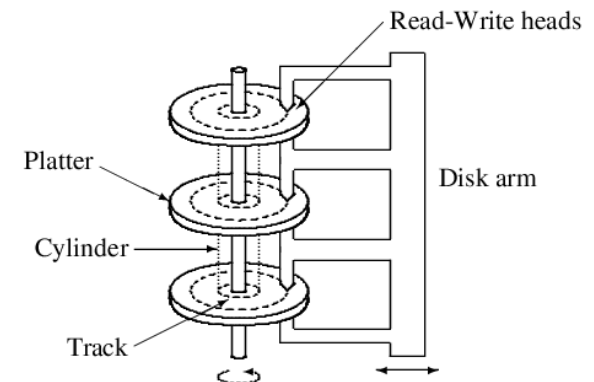
14

Disco magnetico (cont.)

- Nozione di cilindro
 - Consiste di tracce posizionate in modo uguale su tutti i piatti di un disco
 - Tutte le sue tracce possono essere accedute dalla stessa posizione della testina
 - L'uso riduce il movimento della testina
 - Pone dati adiacenti di un file su tracce dello stesso cilindro
- Indirizzo di un record: (*numero cilindro, numero superficie, numero record*)
- Per ottimizzare l'uso della superficie del disco le tracce sono organizzate in settori
 - Slot di dimensione standard in una traccia per un record
 - Dimensione scelta per minimizzare lo spreco di capacità di memorizzazione
- La suddivisione in settori può essere parte dello hw (hard sectoring) o implementata da software (soft sectoring)

15

Struttura di un Hard Disk



16

Organizzazione dei dati su disco

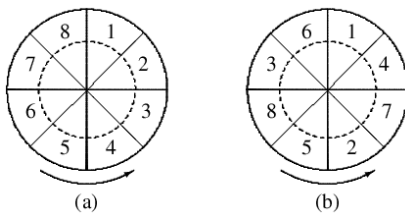
- I dati devono essere organizzati in modo da garantire un accesso efficiente
 - Un disco ruota leggermente mentre le testine del disco si muovono per accedere ad una nuova traccia
 - Assicura che i dati da accedere passano sotto le testine di lettura/scrittura dopo che il loro movimento è completato

Tecniche di distribuzione dati

- Alternanza dei settori
 - I vecchi dischi usati per contenere un buffer per memorizzare i dati letti dal disco o da scrivere
 - In un'operazione read, i dati erano prima letti da un settore del disco nel buffer
 - I dati dal buffer erano poi trasferiti in memoria
 - Il disco era pronto per una nuova operazione
 - Ma il prossimo settore era passato già sotto la testina!
 - Tecnica: un piccolo numero di settori sono saltati mentre si memorizzano record adiacenti di un file
 - In numero di settori saltati è chiamato fattore di alternanza (inf)

Alternanza dei settori

- (a) nessuna alternanza; i record adiacenti in un file occupano settori adiacenti
- (b) fattore di alternanza = 2; ci sono due settori tra record adiacenti



Tecniche di distribuzione dei dati

- Testina asimmetrica
 - Il disco richiede del tempo per commutare dalla lettura dei dati di una traccia ai dati di un'altra traccia in un cilindro (tempo commutazione testina)
 - Alcuni settori (record/blocchi) passano sotto la testina durante questo tempo
 - Asimmetria testina: distribuisce i dati sulle tracce
 - Il primo settore di una nuova traccia deve passare sotto la testina solo dopo che la testina del disco è pronta per leggere
- Asimmetria cilindro
 - Il disco ruota mentre le testine si spostano sulle tracce di un cilindro adiacente
 - Asimmetria cilindro: i dati sono resi asimmetrici come per l'asimmetria della testina

Redundant Array of Inexpensive Disks (RAID)

- E' usato un array di dischi poco costosi anziché un unico disco
- Sono usate diverse disposizioni per fornire tre benefici
 - Affidabilità
 - Memorizza i dati in modo ridondante
 - Legge/scrive i record di dati ridondanti in parallelo
 - Tassi veloci di trasferimento dati
 - Memorizza i dati dei file su più dischi nel RAID
 - Legge/scrive i dati in parallelo
 - Accesso veloce
 - Memorizza due o più copie dei dati
 - Per leggere i dati, accede alla copia che è accessibile in modo più efficiente




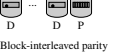
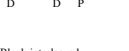

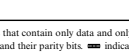
21




Disk stripe

- Un disk strip contiene i dati (è come un settore o blocco)
- Un disk stripe è una collezione di strip posizionate allo stesso modo su dischi diversi nel RAID
 - I dati scritti sugli strip in uno stripe possono essere letti in parallelo
 - Questa disposizione fornisce elevati tassi di trasferimento

22

Table 14.3 RAID Levels

Level	Technique	Description
Level 0	Disk striping 	Data is interleaved on several disks. During an I/O operation, the disks are accessed in parallel. Potentially, this organization can provide an n -fold increase in data transfer rates when n disks are used.
Level 1	Disk mirroring  Disk 1 Disk 2	Identical data is recorded on two disks. During reading of data, the copy that is accessible faster is used. One of the copies is accessible even after a failure occurs. Read operations can be performed in parallel if errors do not arise.
Level 2	Error correction codes 	Redundancy information is recorded to detect and correct errors. Each bit of data or redundancy information is stored on a different disk and is read or written in parallel. Provides high data transfer rates.
Level 3	Bit-interleaved parity 	Analogous to level 2, except that it uses a single parity disk for error correction. An error that occurs while reading data from a disk is detected by its device controller. The parity bit is used to recover lost data.
Level 4	Block-interleaved parity 	Writes a <i>block</i> of data, i.e., consecutive bytes of data, into a stripe and computes a single parity stripe for stripes of a stripe. Provides high data transfer rates for large read operations. Small read operations have low data transfer rates; however, many such operations can be performed in parallel.
Level 5	Block-interleaved distributed parity 	Analogous to level 4, except that the parity information is distributed across all disk drives. Prevents the parity disk from becoming an I/O bottleneck as in level 4. Also provides better read performance than level 4.
Level 6	P + Q redundancy 	Analogous to RAID level 5, except that it uses two independent distributed parity schemes. Supports recovery from failure of two disks.

Note: D and P indicate disks that contain only data and only parity information, respectively.  indicates a stripe.  indicates bits of a byte that are stored on different disks, and their parity bits.  indicates a stripe containing only parity information.

23

Livelli RAID

- Organizzazione dei RAID diverse (livelli RAID) forniscono diversi benefici
 - RAID 0: striping del disco
 - Tassi di trasferimento elevati
 - RAID 1: mirroring del disco
 - Gli stessi dati sono scritti su due dischi
 - Per leggere, la copia accessibile in modo più veloce è acceduta
 - RAID 0+1:
 - striping del disco come nel RAID 0, ogni stripe è mirrored
 - RAID 1+0:
 - I dischi sono prima mirrored, poi striped
 - Fornisce una migliore affidabilità del RAID 0+1

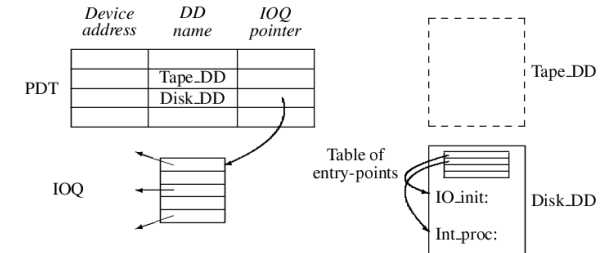
24

Livelli RAID

- Livello 2: Codici di correzione errori
 - I dati e i bit ridondanti sono registrati su dischi diversi
 - Ad esempio, codice di Hamming (12,8)
- Livello 3: bit di parità alternato
 - Simile al livello 2, ma usa un disco di parità singolo
 - Il controller del dispositivo individua l'errore, il bit di parità è usato per correggerlo
- Livello 4: parità di blocco alternato
 - Gli strip contengono byte consecutivi; strip di parità contengono i bit di parità
- Livello 5: parità di blocco distribuita alternata
 - Come il livello 4, ma le strip di parità sono sparse su diversi dischi
- Livello 6: ridondanza P+Q

Driver di dispositivo

- L'entrata della tabella dei dispositivi fisici di un dispositivo contiene il nome del driver
- Un driver gestisce le operazioni di I/O su una specifica classe di dispositivi, inizia le operazioni di I/O e gestisce gli interrupt dal dispositivo nella classe
- Il driver ha degli entry-point per funzionalità standard come avvio I/O, gestione interrupt di I/O, ecc.



Scheduling del disco

- Una politica di scheduling del disco esegue le operazioni di I/O in un ordine che ottimizza il throughput del disco
 - FIFO
 - Shortest Seek Time First (SSTF)
 - Tempo di ricerca: tempo speso nel movimento della testina
 - SCAN/Look
 - Le testine sono mosse da un'estremità del piatto all'altra, servendo le richieste di I/O (Look le muove solo fino all'ultima richiesta in una direzione)
 - La direzione del movimento della testina è invertita; è avviato un altro SCAN
 - CSCAN / C-Look (C sta per circolare)
 - La direzione del movimento non è invertita; è semplicemente avviato un altro scan

Richieste di I/O per lo scheduling del disco

t_c and t_{pt} = 0 msec and 1 msec, respectively
 Current head position = Track 65
 Direction of last movement = Towards higher numbered tracks
 Current clock time = 160 msec

I/O requests:

Serial number	1	2	3	4	5
Track number	12	85	40	100	75
Time of arrival	65	80	110	120	175

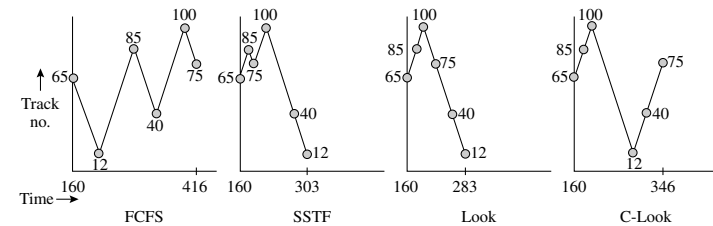
- Le testine del disco si muovono verso le tracce con numeri maggiori
- Le richieste sono fatte in tempi diversi

Dettagli scheduling del disco

Policy	Details	Scheduling decisions					Σ Seek time
		1	2	3	4	5	
FCFS	Time of decision	160	213	286	331	391	
	Pending requests	1,2,3,4	2,3,4,5	3,4,5	4,5	5	
	Head position	65	12	85	40	100	
	Selected request	1	2	3	4	5	
	Seek time	53	73	45	60	25	
SSTF	Time of decision	160	180	190	215	275	
	Pending requests	1,2,3,4	1,3,4,5	1,3,4	1,3	1	
	Head position	65	85	75	100	40	
	Selected request	2	5	4	3	1	
	Seek time	20	10	25	60	28	
SCAN	Time of decision	160	180	195	220	255	
	Pending requests	1,2,3,4	1,3,4,5	1,3,5	1,3	1	
	Head position	65	85	100	75	40	
	Selected request	2	4	5	3	1	
	Seek time	20	15	25	35	28	
CSCAN	Time of decision	160	180	195	283	311	
	Pending requests	1,2,3,4	1,3,4,5	1,3,5	3,5	5	
	Head position	65	85	100	12	40	
	Selected request	2	4	1	3	5	
	Seek time	20	15	88	28	35	

Req	Track	Time
1	12	65
2	85	80
3	40	110
4	100	120
5	75	175

Prestazioni degli algoritmi di scheduling



Tempo di trasferimento nello scheduling del disco

- Supponiamo che nella coda delle richieste di un'unità disco composta da 200 tracce si trovano le richieste di dati nei blocchi
 - 39700 – 304 – 115 – 2600 – 2120 – 270 – 321 – 0 – 760 – 20000
 - il blocco *i-esimo* è memorizzato nella traccia *i mod 200*
- La testina ha eseguito l'ultimo movimento portandosi dalla traccia 85 alla traccia 97
- Si ipotizzi che lo spostamento da una traccia ad un'altra richieda tempo medio pari a 40 μs per traccia, che l'inversione della direzione di movimento richieda in media 80 μs e la velocità di rotazione sia di 7200 giri
- Si vuole determinare il tempo richiesto, complessivamente, per accedere alle tracce indicate per le politiche SSTF, C-SCAN e LOOK.

Soluzione

- Latenza rotazionale: $60 / (2 \times 7200) = 4.17 \text{ ms}$
 - Dobbiamo determinare la traccia alla quale si trova il blocco (**i modo 200**)
 - Blocchi: 39700 – 304 – 115 – 2600 – 2120 – 270 – 321 – 0 – 760 – 20000
 - Tracce: 100 – 104 – 115 – 0 – 120 – 70 – 121 – 200 – 160 – 0
- SSTF. La sequenza di scheduling è:
 - 97 100 104 115 120 121 160 200 70 0
 - Le distanze tra tracce della sequenza sono: 3 4 11 5 1 39 40 130 70
 - Il tempo di accesso è $t_a = (303 \times 40 \mu s) + 80 \mu s + (9 \times 4.17 \text{ ms}) = 12.12 \text{ ms} + 0.08 \text{ ms} + 37.53 \text{ ms} = 49.73 \text{ ms}$

Soluzione

- 2. C-SCAN
 - 97 100 104 115 120 121 160 200 0 70
 - Le distanze tra le tracce
 - 3 4 11 5 1 39 40 200 70
 - $t_a = (373 \times 40 \mu s) + 80 \mu s + (9 \times 4.17 ms) = 14.92 ms + 0.08 ms + 37.53 = 52.53 ms$
- 3. LOOK
 - 97 100 104 115 120 121 160 200 70 0
 - 3 4 11 5 1 39 40 130 70
 - $t_a = (303 \times 40 \mu s) + 80 \mu s + (9 \times 4.17 ms) = 12.12 ms + 0.08 ms + 37.53 ms = 49.73 ms$