11 - Input Output

Gestione dell'Input/Output

Classificazione dei dispositivi I/O

- Classificazione in base alla **sorgente/destinazione**:
  - **input**, e.g., tastiera, dischi, ...
  - **output**, e.g., video, dischi, ...
  - **rete**, e.g., IEE 802.11, BLE, Ethernet, ...
- Classificazione in base alle **modalita' di trasferimento dati**: blocchi, caratteri, speciali
  - dispositivi **a blocchi**: dati trasferiti a blocchi di dimensione fissa, e.g., dischi
  - dispositivi **a caratteri**: dati trasferiti un carattere alla volta senza alcuna struttura interna, e.g., tastiera, mouse, stampante, ...
  - dispositivi **speciali**: e.g., timer, genera interruzioni ad istanti programmati

Velocita' dei dispositivi

- Tastiera: 10 B/s
- Mouse: 100 B/s
- Modem PSTN: 7 KB/s
- Linea ISDN: 16 KB/s
- Stampante laser: 100 KB/s
- Scanner 400 KB/s
- Porta USB 1.5-1200 MB/s
- Disco IDE 5 MB/s
- CD-ROM 6 MB/s
- Fast Ethernet 12.5 MB/s
- Monitor XGA 60 MB/s
- Ethernet gigabit: 125 MB/s
- Fibra: 125 MB/s

Architettura hardware del sottosistema I/O

- La **CPU** legge e scrive i registri del controller mediante apposite istruzioni
- Il **deposito** invia e riceve le informazioni e i dati tramite i registri e il buffer del controller

Funzioni del livello dipendente dai dispositivi

Per ogni dispositivo esiste un programma **device driver** che implementa il protocollo operativo associato al dispositivo

- il device driver da parte del livello del SO devide-dependent
- le funzioni di gestione delle interruzioni generati dai dispositivi di periferici fanno parte del sottosistema di I/O devce-dependent

Organizzazione logica per la gestione dei dispositivi

- **Driver**: i driver solo la parte del sistema operativo che gestiscono i comandi
- Compito del driver e' di **inviare i comandi** appropriati ai dispositivi (al controller) e **gestire le interruzioni**
- E' la sola parte del sistema operativo che conosce i comandi del controller, il numero dei registri, etc.

Funzioni del livello indipendente dai simboli

Naming: ogni dispositivo e' identificato univocamente. In UNIX ogni dispositivo ha un nome simbolico all'internod ello spazio dei nomi del file system (si veda la directory/dev)

Buffering: aree buffer che ospitano i dati nel trasferimento tra i dispositivi e le aree di memoria dei processi applicativi.
Servono per:

1. mediare tra diverse velocita' di produzione/consumo tra processi e dispositivi
2. trasferire efficacemente dei blocchi dati
3. parallelizzare le operazioni di accesso I/O

Gestione eccezioni: nelle operazioni di I/O si possono verificare molti eventi anomali, che possono essere:

- mascherati e nascostio agli utenti (il sistema prova a completare le operazioni fallite)
- comunicati e propagati a processi e utenti

Spooling: tecnica di gestione per le risorse condivise (un processo gestore per ogni risorsa)

Input/Output a controllo di programma vs. guidato dalla interruzioni

- **A controllo di** programma: approccio sincrono, ogni processo che inizia un'operazione di I/O viene bloccato in attesa che il sistema operativo porti a termina l'operazione di I/O richiesta

- **Guidato dalle interruzioni**: approccio asincrono, il processo non si blocca ma al termine dell'operazione di I/O (per esempio lettura di un blocco di file da un disco) il controller del dispositivo lancia una **interruzione hardware** al sistema operativo che puo' quindi informatre il processo richiedente
- La gestione a interruzione evita l'inefficienza delle attese attive presente del SO nella dell'I/O eseguital al controllo di programma (polling)

Gestione degli Hard Disk

Gli Hard Disk sono dispositivi particolarmente importanti perche' offrono uno spazio di memoria di massa, utilizzato per il file system ma **anche per la memoria virtuale**

Organizzazione fisica dei dischi

Il **settore di una traccia** e' l'unita' minima di allocazione e di trasferimento (ordine di grandezza KB), identificato da:

- N. della faccia del disco
- N. della traccia (o cilindro)
- N. del settore dentro la traccia

Prestazioni Hard Disk

Le **prestazioni** di un Hard Disk sono valutate in termini di **tempo medio di trasferimento**:

TF = TA + TT
TF: Tempo medio di trasferimento
TA: Tempo medio di accesso (per posizionare testina)
TT: Tempo medio di trasferimento dati (per trasferire dati)

TA = ST + RL
ST: Seek Time, tempo per spostare longitudinalmente la testina del disco sulla traccia richiesta
RL: Rotational Time, tempo necessario per ruotare il disco in modo da leggere il settore richiesto.
Prestazioni dischi espresse in giri al minuto, tra 5.400 e 15.000

TT ordine microsecondi, ST e RL ordine millisecondi

Per ridurre tempi di accesso ai dati, progettare strategie, **politiche**, per:

- allocazione dei file (in settori se possibile contigui)
- schedulare le richieste di accesso ai dischi (per minimizzare tempi di spostamento testina)

Politiche scheduling di accesso Hard Disk

In un sistema concorrente, molti processi al file system, che si trova quindi a gestire molte richieste, che devono essere schedulate (adottano specifiche **politiche**) opportunamente per ridurre i tempi di attesa dei processi.

Dischi RAID

Per migliorare ulteriormente le **prestazioni**, si possono utilizzare in parallelo piu' dischi fissi. Questo puo' permettere anche di migliorare l'**affidabilita'** e la **tolleranza ai guasti** (tramite ridondanza dei dati)

Sistemi **RAID** ((Redundant Array of Independent Disks)

RAID livello 0 (striping)

Si crea **un solo volume** logico su tutti i dischi.

I dati sono allocati su dischi diversi, per **parallelizzare** operazioni di I/O

RAID livello 1 (mirroring)

Tutti i dati sono **replicati su due dischi.** Il sistema scrive un dato sempre su due dischi.

- Lettura puo' essere parallelizzata sui due dischi
- Possibile mirroring anche arree del sistema
- Tolleranza al guasto di un disco
- Elevato **costo** (utilizzo dischi del 50%)

RAID livello 5 (striping con parita')

- Ogni sezione di parita' contiene **l'XOR (or-esclusivo)** delle 4 sezioni dati corrispondenti
- Nel caso di perdita di UNA delle sezioni dati, il sistema ricostruisce la perdita utilizzando la sezione di parita'
- Minore costo rispetto a mirroring (in questo esempio, costo del 20%)
- Ogni scrittura richiede modifica sezione di parita'

RAID livello 6 (striping con doppia parita')

- Molto simile al RAID livello 5 ma con **un blocco di parita' aggiuntivo**: stripng dei dati su tutti i dischi con due blocchi di parita'
- Le operazioni di scrittura sono piu' costose a causa dei calcoli della parita' ma le letture non hanno svantaggi prestazionali
- Maggiore affidabilita' rispetto al RAID livello 5

Serial Advanced Technology Attachment (SATA)

L'interfaccia SATA (composta da 6 unita') permette lo scambio di dati tra un host e un device SATA attraverso un **link seriale**

- vengono raccolti dati da un host e formato un frame di informazioni attraverso la Data Processing Unit
- i dati nel frame vengono poi codificati e serializzati prima di essere trasmessi al device
- i dati ricevuti dall'interfaccia vengono de-serializzati e de-codificati attraverso un processo inverso per poi essere processati e restituiti all'host

Tutte le operazioni dell'interfaccia vengono monitorate dal Controller SATA

Unita' a stato solido (Solid State Drive - SSD)

Dispositivi **molto veloci** con **prestazioni asimmetriche** di lettura e scrittura, non contengono parti mobili.

Ache se alcuni sono conformi allo standard SATA concepito per dischi meccanici, sempre piu' SSD si interfacciano al sistema attraverso **Non-Volatile Memory express (NVMe)**

Non-Volatile Memory express (NVMe)

- **Standard di accesso** ultraveloce alle memorie non volatili che sfrutta eglio la velocita' di connessione e il parallelismo disponibile negli SSD
- Supportando l'**uso di code multiple** rende possibile **elaborare richieste in parallelo** attraverso le sue molteplici pagine e chip (in aggiunta ai tanti core a disposizione nei moderni elaboratori)
- La macchina ha bisogno di meno dispositivi per supportare lo stesso numero di operazioni I/O e inoltre riducono molto i requisiti di energia e raffreddamento
- Permette un accesso diretto al bus PCIe e all'SSD --> in NVMe sono coinvolti meno strati software rispetto alle operazioni SATA
- Offre una **coda di comandi** (submission queue) e una **coda di risposte** (completion queue) per ciascun core
- Per eseguire richieste memorizzazione un core scrive i comandi di I/O nella sua coda richieste e NVMe scrivera' in un registro chiamato **doorbell** quando i comandi sono pronti

SSD e RAID

Rispetto ai dischi magnetici, gli SSD offrono prestazioni molto migliori e una maggiore affidabilita'. C'e' ancora bisogno del RAID?

Tipicamente si, un RAID di piu' SSD puo' offrire prestazioni e affidabilita' migliori di uno singolo:

- un RAID livello 0 fornisce prestazioni di lettura e scrittura sequenziali migliori rispetto al singolo SSD
- anche RAID livello 5 e 6 sono utilizzati con SSD: migliorano prestazioni e affidabilità ma al costo di operazioni di scrittura molto intense e costose, che nel lungo periodo aumentano l'usura degli SSD