

Problemi nel progetto del sistema operativo

Index

- [Obiettivi](#)
 - [Efficienza](#)
 - [Generalità](#)
 - [Progettazione gerarchica](#)
 - [Dispositivo locale](#)
 - [Dispositivo di comunicazione](#)
 - [File system](#)

Obiettivi

I sistemi operativi devono però gestire i dispositivi I/O ponendosi degli obiettivi

Efficienza

Uno dei problemi più importanti è il fatto che la maggior parte dei dispositivi di I/O sono molto lenti rispetto alla memoria principale. Bisogna dunque sfruttare il più possibile la multiprogrammazione per evitare che questo problema di velocità diventi basso utilizzo del processore.

Nonostante ciò l'I/O potrebbe comunque non tenere il passo del processore (il numero di processi ready si riduce fino a diventare zero); come soluzione si potrebbe pensare che sia sufficiente portare altri processi sospesi in memoria principale (medium-term scheduler), ma anche questa è un'operazione di I/O

Risulta dunque necessario cercare soluzioni software dedicate, a livello di SO, per l'I/O (in particolare per il disco)

Generalità

Nonostante siano tanti e diversi i dispositivi di I/O, bisogna comunque cercare di gestirli in maniera uniforme.

Bisogna quindi fare in modo che ci sia un'unica istruzione di `read` che in base

all'argomento che gli viene dato sa come gestire suddetto dispositivo. Per questo motivo è necessario nascondere la maggior parte dei dettagli dei dispositivi di I/O nelle procedure di basso livello

Le funzionalità da offrire sono: `read`, `write`, `lock`, `unlock`, `open`, `close`, ...

Progettazione gerarchica

Per fare ciò è necessario utilizzare una progettazione gerarchica basata su livelli, in cui ogni livello si basa sul fatto che il livello sottostante sa effettuare operazioni più primitive, fornendo servizi al livello superiore.

Inoltre ogni livello contiene funzionalità simili per complessità, tempi di esecuzione e livello di astrazione

Hint

Modificare l'implementazione di un livello non dovrebbe avere effetti sugli altri

Esistono sostanzialmente 3 macrotipi di progettazioni gerarchiche

Dispositivo locale

Riguarda dispositivi attaccati esternamente al computer (es. stampante, monitor, tastiera...)



Logical I/O → il dispositivo viene visto come risorsa logica (`open`, `close`, `read`, ...)

Device I/O → trasforma le richieste logiche in sequenze di comandi di I/O

Scheduling and Control → esegue e controlla le sequenze di comandi, eventualmente gestendo l'accodamento

Dispositivo di comunicazione

Riguarda quei dispositivi che permettono la comunicazione (scheda Ethernet, WiFi, ...)



Strutturato come prima, ma al posto del logical I/O c'è una architettura di comunicazione, tramite la quale il dispositivo viene visto come risorsa logica. A sua volta questa consiste in un certo numero di livelli (es. TCP/IP)

File system

Riguarda i dispositivi di archiviazione (es. HDD, SSD, CD, DVD, floppy disk, USB, ...)



Qui il logical I/O è tipicamente diviso in tre parti

Directory Management → fornisce tutte le operazioni atte a gestire i file (creare, spostare, cancellare, ...); da nomi di file a identificatori di file

File system → struttura logica ed operazioni (apri, chiudi, leggi, scrivi, ...)

Organizzazione fisica → si occupa di allocare e deallocare spazio su disco (quando ad esempio si chiede di creare un file); da identificatori di file a indirizzi fisici su disco

Scheduling & Control → ovviamente è qui che ci sono i vari **SCAN**