

Appunti di sistemi operativi

Appunti per il corso universitario di sistemi operativi, riferito a sistemi Unix/Windows.
Si discute su problemi di sincronizzazione, memoria e scheduling dei processi.



ARGOMENTI

[INTRODUZIONE](#)

[INPUT/OUTPUT](#)

[GESTIONE DEI PROCESSI](#)

[ALGORITMI DI SCHEDULING](#)

[SCHEDULING MULTI-CPU](#)

[SISTEMI REAL TIME](#)

[SCHEDULING SU LINUX](#)

[SCHEDULING SU WINDOWS](#)

[OPERAZIONI SUI PROCESSI](#)

[COMUNICAZIONE TRA PROCESSI](#)

[THREAD](#)

[SINCRONIZZAZIONE TRA PROCESSI](#)

[GESTIONE MEMORIA](#)

COS'È UN SISTEMA OPERATIVO?

un software che controlla e coordina l'uso dell'hardware da parte di programmi/utenti, e che funge da:

- **macchina virtuale** (astrazione software della macchina fisica)
- **interfaccia** tra utente (e applicazioni) e hardware
- **gestore di risorse** (gestire l'hardware in modo efficiente ed equo)

Le applicazioni possono essere **utente** o **sistema**.

il kernel sta sotto entrambe ed è "software di sistema che è necessariamente sempre attivo".

SISTEMI BATCH

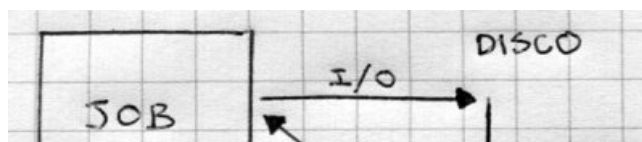
batch = "lotto" nel senso di "gruppo di cose" che mando in elaborazione; ossia:

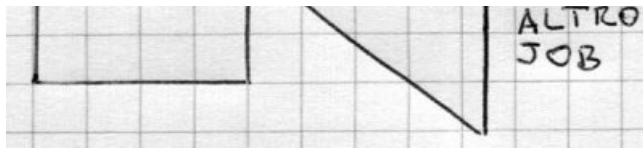
- raggruppamento di programmi "simili" che vengono mandati in esecuzione perché magari utilizzano le stesse risorse uguali.
- S.O. **monitor residente** che legge il codice del programma, lo carica in memoria, lo esegue e quindi passa a quello successivo; fasi:
 1. inizializzazione hardware
 2. caricamento programma (job)
 3. passa il controllo al job
 4. riprende il controllo
 5. torna al punto 2

BATCH MULTIPROGRAMMATI

multiprogrammazione = tecnica che mira a disporre dell'esecuzione contemporanea di più processi per massimizzare l'utilizzo della CPU (durante la richieste di I/O):

- S.O. carica in memoria un insieme di job
- quando un job attende un'operazione di I/O manda in esecuzione un altro job: il S.O. funge quindi da macchina virtuale (perché le richieste di I/O dei job passano attraverso esso) e da gestore delle risorse





in questo tipo di sistemi non ho interazione durante l'esecuzione dei job

SISTEMI TIME-SHARING

time-sharing = "condivisione di tempo" = partizione del tempo che mira a commutare l'utilizzo della CPU tra i vari processi così frequentemente da permettere all'utente di poter interagire con ciascun programma in esecuzione

singola CPU (n.b. in sistemi a singola CPU possiamo eseguire un solo processo alla volta)

la CPU commuta tra i job (processi) a una frequenza tale da ottenere:

- **esecuzione "simultanea"** di job (processi)
- **interazione** utente durante l'esecuzione dei job (processi)

EVOLUZIONE DEI CALCOLATORI

Personal Computer (PC) - anni '70:

- spesso un solo utente
- molto I/O e poco uso CPU

...evoluzione con le reti ed Internet:

- più protezione
- migliore gestione risorse

-> ai giorni nostri tutti usano time-sharing, sia pc che mainframe.

SISTEMI PARALLELI

TIGHTLY COUPLED SYSTEMS

= sistemi fortemente accoppiati: sistemi con **più processori** nella stessa macchina

lo scopo è quello di migliorare le prestazioni (tempo di esecuzione $\sim 1/n$; n = numero di CPU):

i sistemi **SMP (symmetric multiprocessing)** possono eseguire "una copia" del sistema per ogni CPU e quindi più processi, che sono veramente simultanei

sono molto usati in caso di esigenze **mission critical**, perché grazie alla **ridondanza** godono di una buona **fault-tolerance** (tolleranza ai guasti), infatti nel caso in cui una CPU fallisca ne interviene un'altra; c'è bisogno però di un controllore, il che può provocare ulteriori problemi

LOOSELY COUPLED SYSTEMS

debolmente accoppiati: **sistemi distribuiti**

vengono chiamati **AMP (asymmetric multiprocessing)** e si usano ad esempio nel clustering, in situazioni in cui i vari elaboratori siano connessi tra loro in una rete affidabile

SISTEMI REAL-TIME

HARD

sono sistemi dedicati che devono rispettare dei vincoli temporali: sistemi critici (Pilotaggio, ABS, ecc)

SOFT

in questi il vincolo temporale è “desiderato”: streaming audio/video, sistemi industriali, (situazioni “meglio tardi che mai”...anche se vedo il video a scatti non muore qualcuno!!) in ogni caso anche questi sono abbastanza precisi

[continua..](#)

[Ritorna sopra](#) | [Home page](#) | [Xelon](#)