

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.

Sistemi di Gestione Ohsas 18001 - ISO 14001 Modello Organizzativo 231 - OdV www.idramanagement.com/

Green Applications Migliora l'efficienza del tuo gestionale limitando i costi www.micronixnetwork.com

XFlar – Qprel srl XFlar – Flexible Linux Arm Flexible Arm Environment www.xflar.net

Annunci Google

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](#)

SISTEMI IN TEMPO REALE

sono di 2 tipi:

- hard
- soft

SISTEMI REAL-TIME HARD

hard = con vincoli che devono essere rispettati (es. ABS automobile); utilizzano sistemi dedicati, non time-sharing e non multi-tasking (o almeno questo è preferibile)

se vogliamo il multi-tasking in questi sistemi dobbiamo utilizzare delle tecniche particolari:

- prenotazione delle risorse
- stima molto precisa di I/O e tempo di CPU

SISTEMI REAL-TIME SOFT

soft = i processi real-time possono coesistere con processi non real-time; utilizzano:

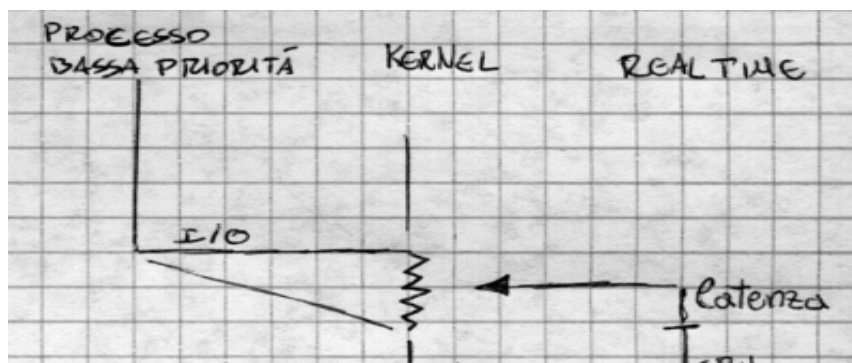
1. scheduling a priorità
2. processi real-time con priorità statica
3. poca latenza nel dispatch dei processi real-time

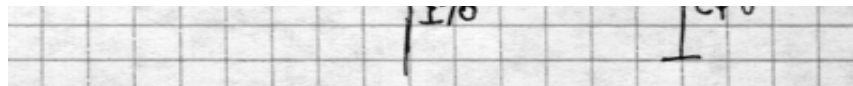
in particolare i sistemi real-time soft si distinguono fra dotati di kernel preemptible e non

KERNEL NON INTERRUPIBILE

nel momento in cui processo real-time richiede la CPU manda interrupt; il kernel se ha un processo in esecuzione lo interrompe (linea ondeggiata = intervento del kernel)

se la richiesta arriva nel pezzo ondeggiato con questo tipo di kernel siamo obbligati a introdurre una latenza sul processo real-time; l'unica cosa che possiamo fare è ridurre il più possibile tale latenza



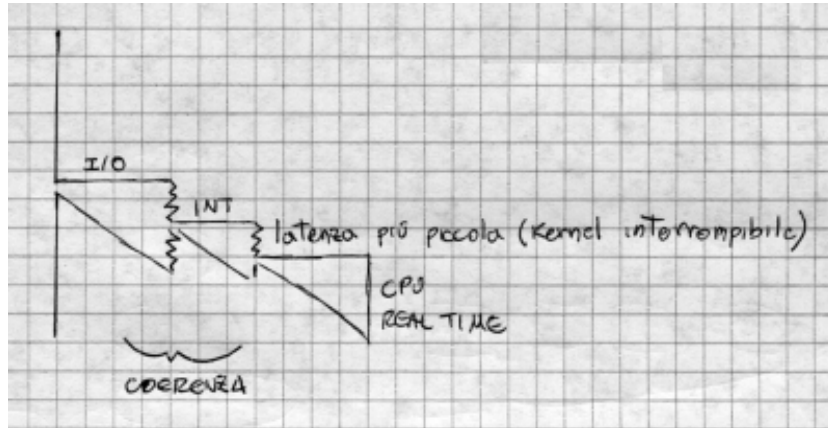


qui ho il problema di **ridurre le latenze** del sistema (dischi ecc.)

KERNEL INTERRUPIBILE

con questo tipo di kernel posso interrompere l'I/O, ma sorge un problema: se io stavo manipolando una struttura dati e il processo real-time che interviene ha bisogno di quella struttura, potrei avere delle conseguenze indesiderate provocate da una lettura di dati non corretti)

con opportuni accorgimenti e comunque possibile realizzare un kernel di questo tipo funzionante, ma è molto complesso



qui ho il problema di **mantenere la coerenza** delle strutture dati

[continua..](#)