

Breve storia dei Sistemi Operativi

1 Timesharing

Il **timesharing** è un'estensione della multiprogrammazione in cui a ogni job viene assegnato un **quanto di tempo (time slice)**: quando questo scade, la CPU viene sottratta al job (anche in assenza di I/O o preemption) e assegnata a un altro.

In questo modo, un singolo processore può essere condiviso da più terminali interattivi: siccome nessun job può usare troppo a lungo la CPU, l'interazione è garantita a tutti gli utenti. A tale scopo, lo scheduler non può essere basato sulle priorità, ma deve invece assegnare la CPU equamente a tutti i job, eseguendo ciascuno di essi a **turno (round robin)**.

Siccome in un sistema con timesharing sono presenti più utenti, il SO deve garantire, oltre alla *protezione della memoria* dei singoli job, anche la **protezione dei dati** di ciascun utente.

Rispetto alla multiprogrammazione, il timesharing penalizza il throughput, ma migliora i **tempi di risposta** per gli utenti interattivi, privilegiando quindi la user convenience rispetto all'uso efficiente delle risorse.

2 Riassunto: multiprogrammazione e timesharing

- **Multiprogrammazione:**
 - obiettivo: massimizzare il **throughput**;
 - tecniche: **priorità, preemption**.
- **Timesharing:**
 - obiettivo: migliorare i **tempi di risposta**;
 - tecniche: **time slice, round robin**.

3 Context switch e overhead

Sottrarre la CPU a un job e assegnarla a un altro (**context switch**) è un'operazione costosa in termini di tempo: essa è implementata da una parte del codice del SO, che deve essere eseguita sulla CPU stessa.

Il tempo necessario a effettuare il context switch, durante il quale la CPU non può eseguire alcun job, costituisce quindi un **overhead**.

In un sistema con timesharing, aumentare la frequenza dei context switch garantisce maggiore interattività, ma aumenta l'overhead, penalizzando il throughput. Bisogna allora trovare un equilibrio.

4 Generazione 4: circuiti integrati LSI/VLSI

La diminuzione dei costi dei computer permette una più ampia varietà di impieghi, oltre ai calcoli scientifici e all'elaborazione di dati commerciali. In particolare, con la diffusione dei **PC** (a partire dagli anni '80) diventa importante la facilità d'uso, e vengono quindi introdotte le interfacce grafiche (**GUI, Graphical User Interface**).

Nello stesso periodo si diffondono anche i **sistemi real time**, usati nelle situazioni in cui è necessario rispondere a un ambiente esterno entro scadenze temporali fissate dall'ambiente stesso (come ad esempio i controlli industriali).

Negli anni '90 si diffondono poi i **sistemi operativi distribuiti** (per la gestione dei *sistemi distribuiti*), che introducono i concetti di:

- *controllo distribuito*;
- *trasparenza delle risorse e dei servizi*;
- *Remote Procedure Call (RPC)*.

5 SO moderni

I sistemi moderni combinano le varie tecniche sviluppate durante l'evoluzione dei SO per soddisfare le diverse esigenze dei vari utenti.