

Thread

È la suddivisione di un processo in sottoprocessi

DIFFERENZE
COI PROCESSI

PROCESSO:	PID
	CODICE
	STACK
	RISORSE

THREAD:		PID
		CODICE
		DATI
T ₁	T ₂	
TID	TID	
STACK ₁	STACK ₂	

I thread condividono i dati e le risorse

THREAD → UTENTE
 → KERNEL (gestiti dal OS)

Mei thread è possibile fare FORK come per i processi ma anche un JOIN che permette di convergere due thread

THREAD UTENTE: gestiti dai processi attraverso determinate librerie

- + migliore compatibilità con OS diversi
- + migliore personalizzazione della gestione dei thread
- + context switch più efficiente
- L'OS vede i thread come singoli programmi e manda in waiting tutto il processo per una richiesta (es. read) fatta da un singolo thread
- parallelismo virtuale, non reale

THREAD KERNEL: trasparenti al sistema operativo, implica scheduling della CPU a livello dei singoli thread

- I vantaggi e svantaggi sono gli stessi ma invertiti, in certi OS è possibile effettuare però UPCALL per avvisare il processo che passerà in WAITING in maniera tale da togliere l'CPU dal thread bloccato

Metodi di assegnazione delle unità di computazione

(1-1) per ogni thread stante assegniamo un'unità di computazione ^{thread kernel}

(1-molti) tanti thread stante per una sola unità di computazione

(molti-molti) ^{più utilizzato} ad ogni processo un numero di thread limitato, chiamati ^{light weight process} LWP, visti dal processo come risorse