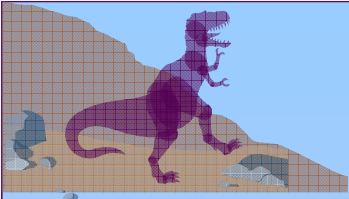


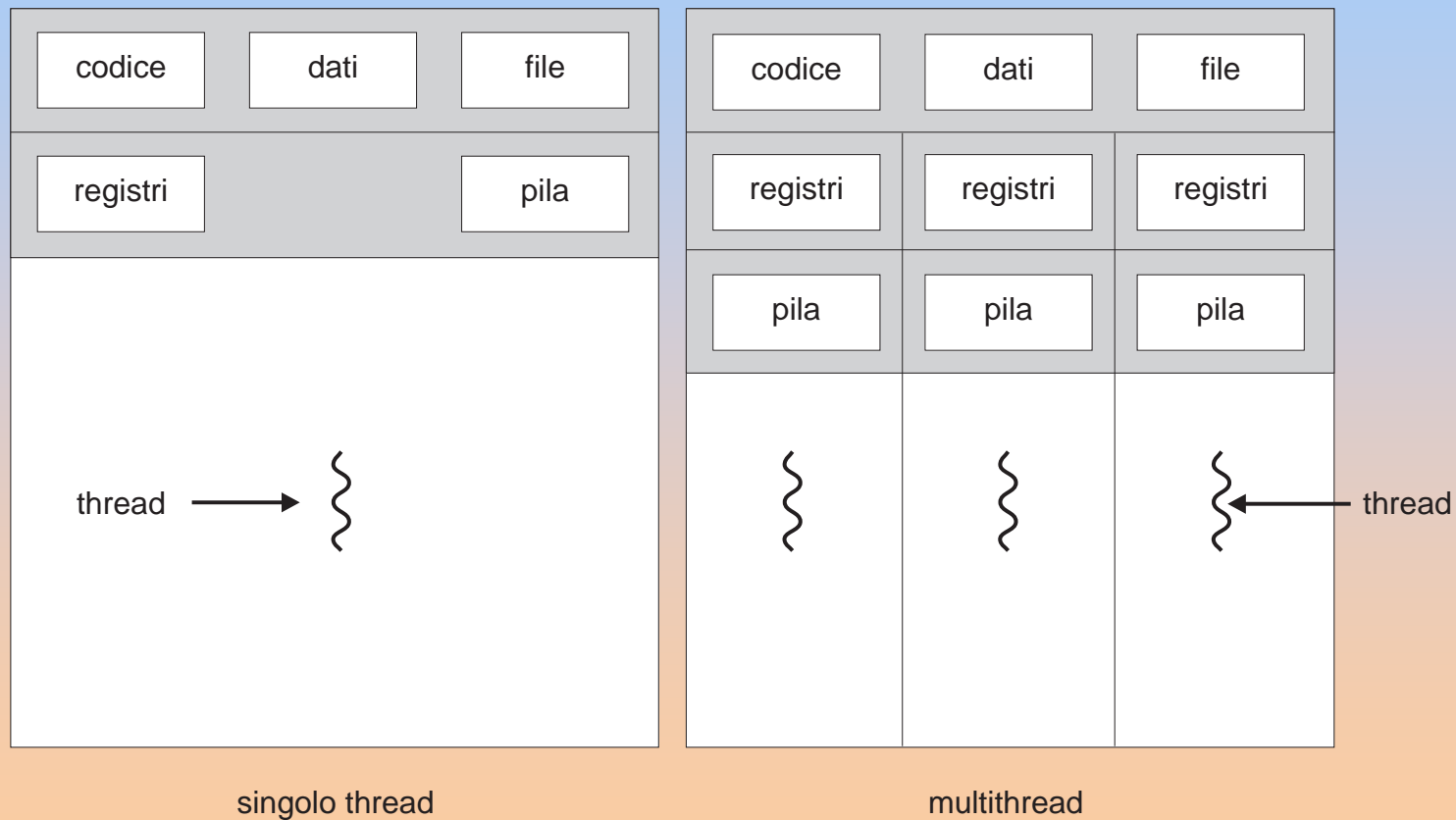
Capitolo 5: Thread

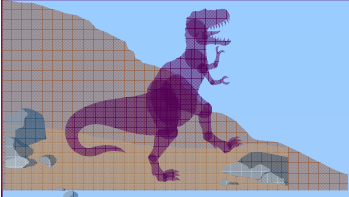
- Introduzione
- Modelli di programmazione multithread
- Questioni di programmazione multithread
- Pthreads
- Thread nel sistema Solaris 2
- Thread nel sistema Windows 2000
- Thread nel sistema Linux
- Thread nel linguaggio Java





Processi a singolo thread e multithread

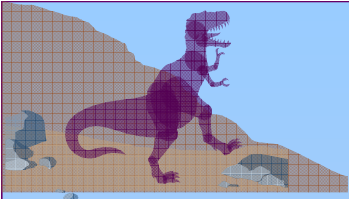




Vantaggi

- Tempo di risposta
- Condivisione delle risorse
- Economia
- Uso di più unità d'elaborazione

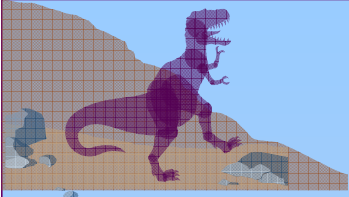




Thread al livello d'utente

- Gestiti come uno strato separato sopra il nucleo del sistema operativo, e realizzati tramite una libreria di funzioni per la creazione, lo scheduling e la gestione, senza alcun intervento diretto del nucleo.
- Esempi:
 - Libreria POSIX *Pthreads*
 - Libreria *C-threads* del sistema Mach
 - *UI-threads* del sistema Solaris 2

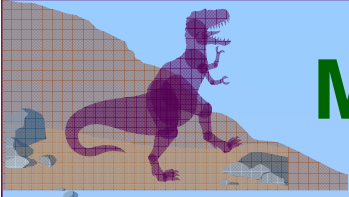




Thread al livello del nucleo

- Gestiti direttamente dal sistema operativo.
- Esempi:
 - Windows 95/98/NT/2000
 - Solaris
 - Tru64 UNIX
 - BeOS
 - Linux

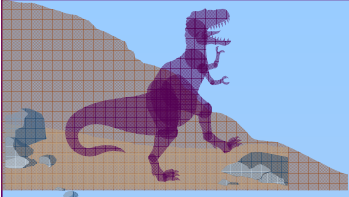




Modelli di programmazione multithread

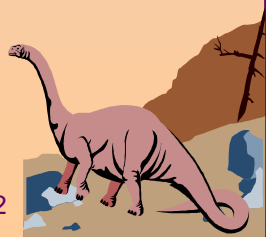
- Modello da molti a uno
- Modello da uno a uno
- Modello da molti a molti



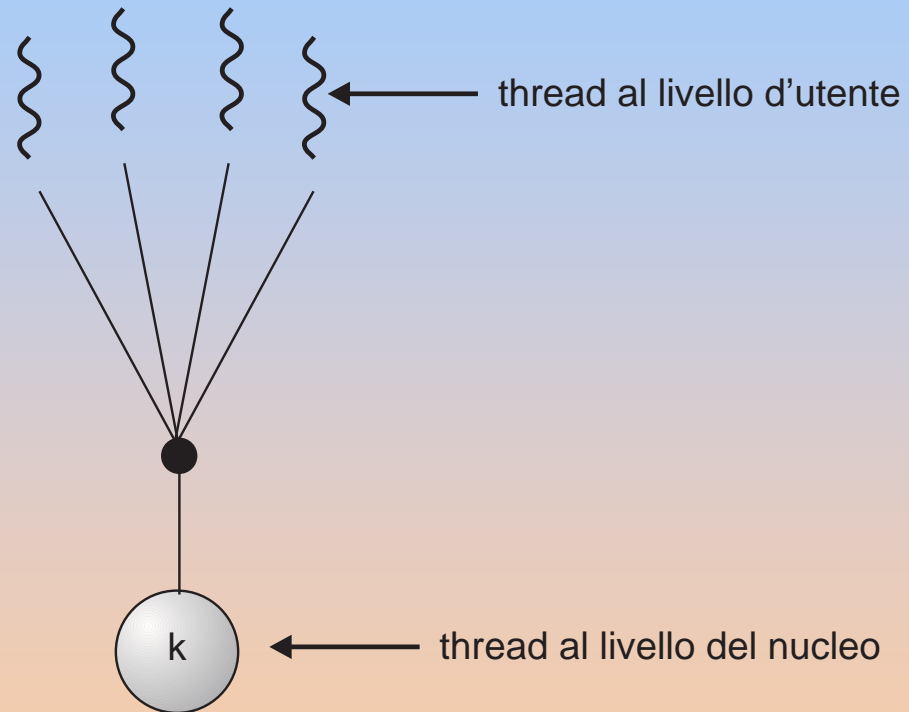


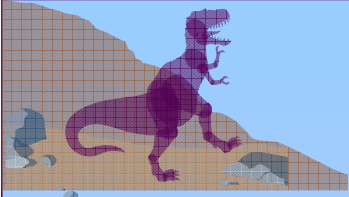
Modello da molti a uno

- Fa corrispondere molti thread al livello d'utente a un singolo thread al livello del nucleo.
- Usato su sistemi che non gestiscono i thread al livello del nucleo.



Modello da molti a uno



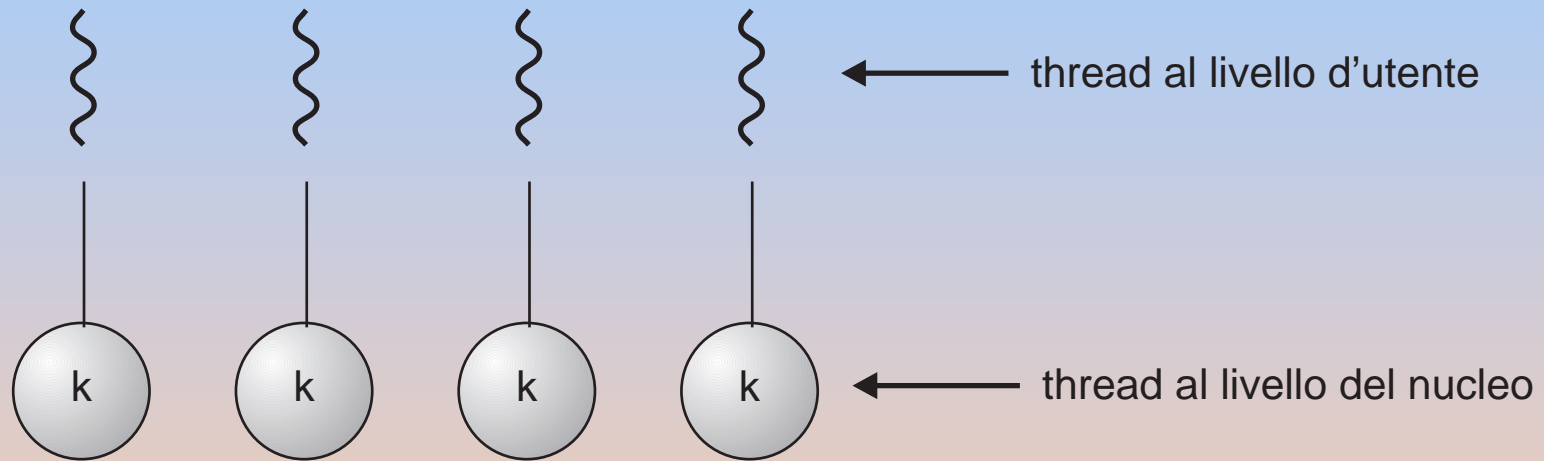


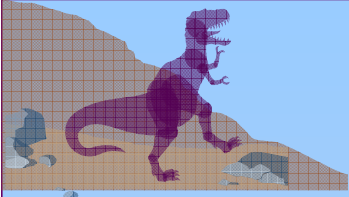
Modello da uno a uno

- Mette in corrispondenza ciascun thread del livello d'utente con un thread del livello del nucleo.
- Esempi:
 - Windows 95/98/NT/2000
 - OS/2



Modello da uno a uno

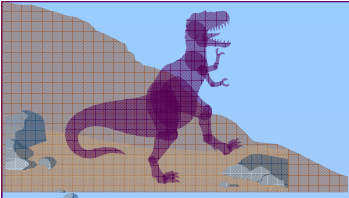




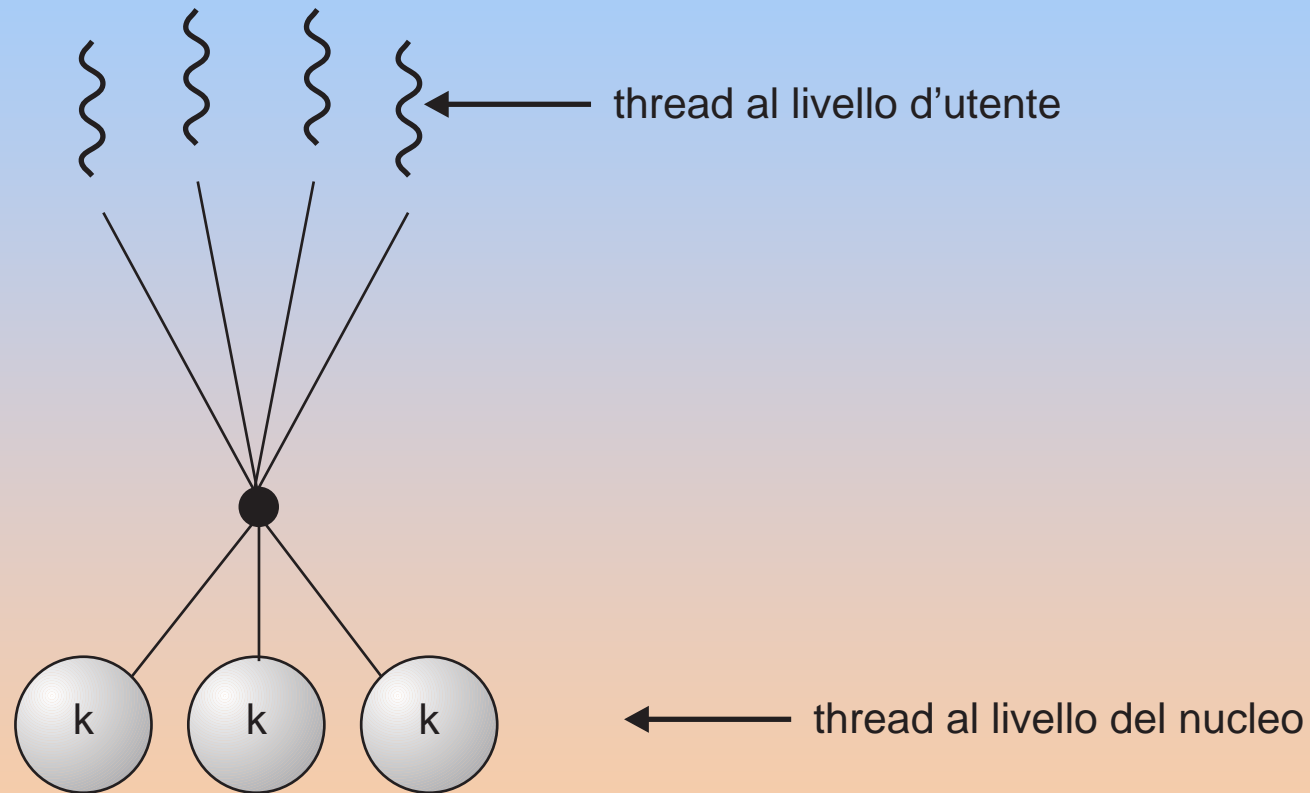
Modello da molti a molti

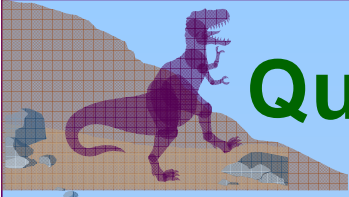
- Mette in corrispondenza più thread del livello d'utente con un numero minore o uguale di thread del livello del nucleo.
- consente ai programmatori di creare liberamente i thread che ritengono necessari.
- Solaris 2
- Windows NT/2000 con il pacchetto *ThreadFiber*





Modello da molti a molti

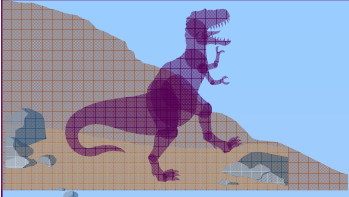




Questioni di programmazione multithread

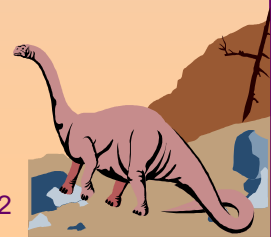
- chiamate del sistema `fork()` ed `exec()`
- Cancellazione
- Gestione dei segnali
- Gruppi di thread
- Dati specifici dei thread



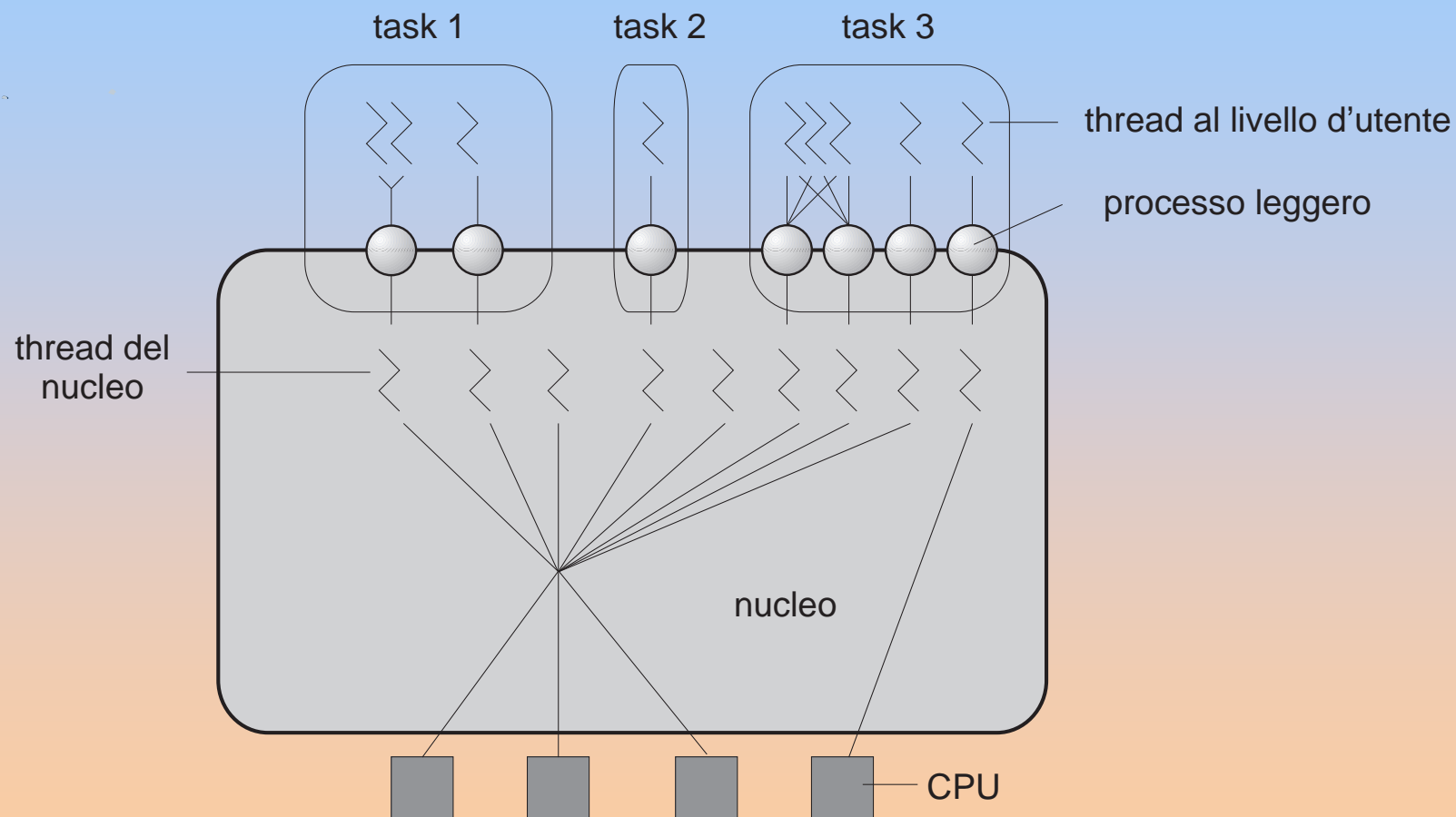


Pthreads

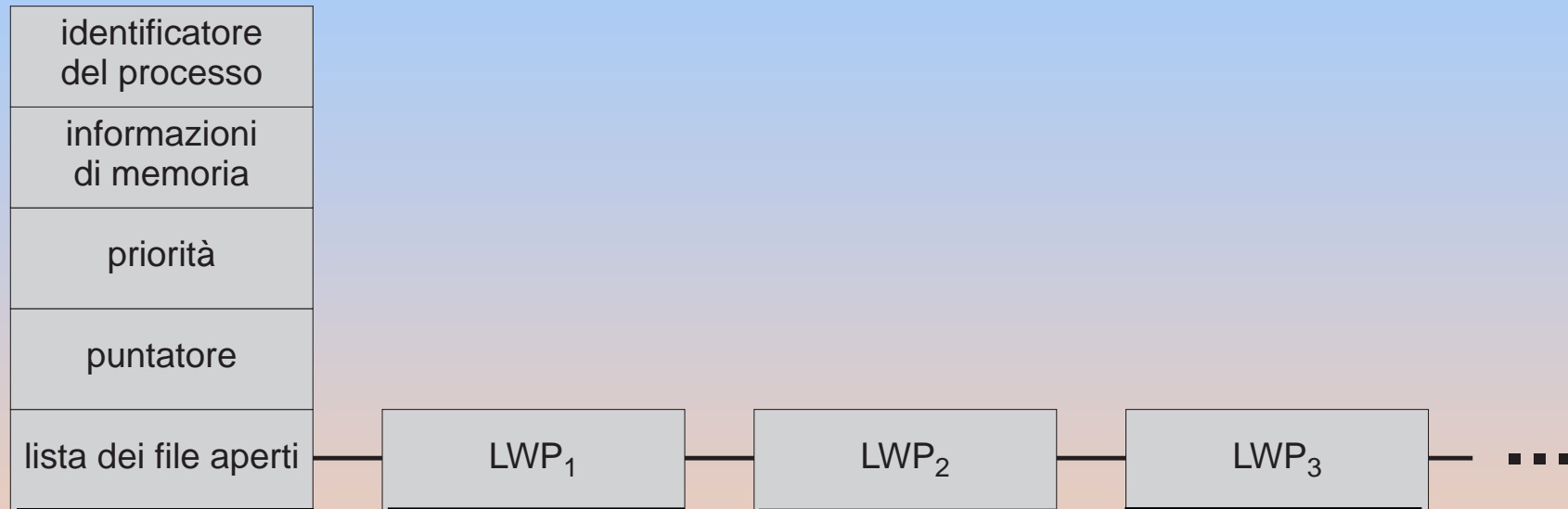
- Lo standard POSIX (IEEE 1003.1c) che definisce l'API per la creazione e la sincronizzazione dei thread.
- Non si tratta di una *realizzazione* ma di una *definizione* del comportamento dei thread; i progettisti di sistemi operativi possono realizzare le API così definite come meglio credono.
- Comuni nei sistemi UNIX.

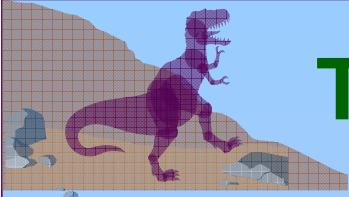


Thread del sistema Solaris 2



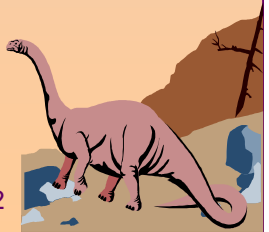
Processo in Solaris 2

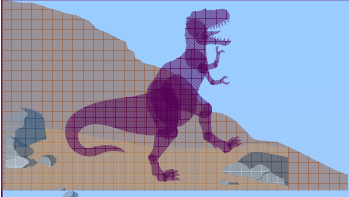




Thread nel sistema Windows 2000

- Impiega il modello da uno a uno.
- Ciascun thread contiene:
 - un identificatore di thread (ID)
 - un insieme di registri
 - una pila d'utente e una pila del nucleo
 - un'area di memoria privata

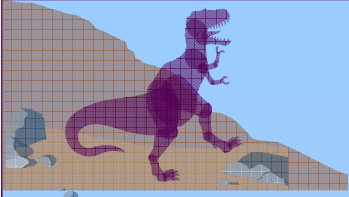




Thread nel sistema Linux

- Generalmente si usa il termine *task* anziché processo o thread.
- La creazione di un thread avviene attraverso la chiamata del sistema `clone()`.
- `Clone()` anziché creare una copia del processo chiamante, crea un processo distinto che condivide lo spazio d'indirizzi del processo chiamante.





Thread nel linguaggio Java

- I thread nel linguaggio Java possono essere creati
 - ☞ creando una nuova classe derivata dalla classe **Thread**
 - ☞ sovrascrivendo il metodo **run** di quella classe
- I thread nel linguaggio Java sono gestiti dalla macchina virtuale (JVM).

