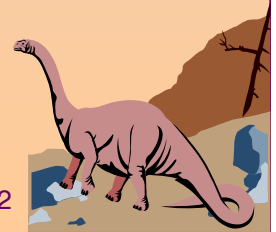
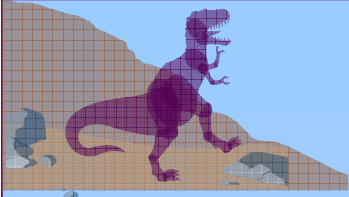


# Capitolo 3: Strutture dei sistemi operativi

- Componenti del sistema
- Servizi di un sistema operativo
- Chiamate del sistema
- Programmi di sistema
- Struttura del sistema
- Macchine virtuali
- Progettazione e realizzazione di un sistema
- Generazione di sistemi

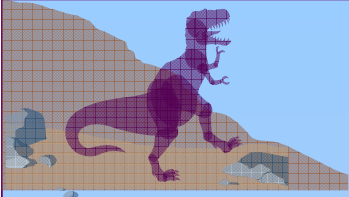




# Componenti del sistema

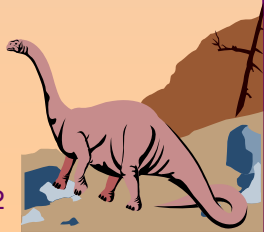
- Gestione dei processi
- Gestione della memoria centrale
- Gestione dei file
- Gestione del sistema di I/O
- Gestione della memoria secondaria
- Reti
- Sistema di protezione
- Interprete dei comandi

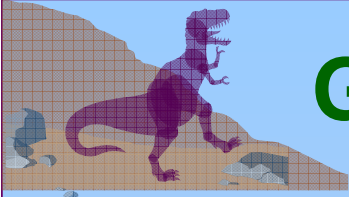




# Gestione dei processi

- Un **processo** è un programma in esecuzione. Per svolgere i propri compiti, un processo necessita di alcune risorse, tra cui tempo di CPU, memoria, file e dispositivi di I/O.
- Il sistema operativo è responsabile delle seguenti attività connesse alla gestione dei processi:
  - ☞ creazione e cancellazione dei processi utenti e di sistema
  - ☞ sospensione e ripristino dei processi
  - ☞ fornitura di meccanismi per:
    - 📄 sincronizzazione dei processi
    - 📄 comunicazione tra processi

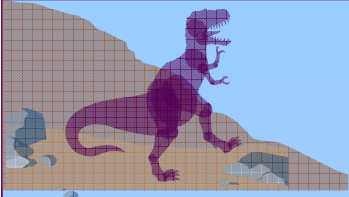




# Gestione della memoria centrale

- La memoria centrale è un vasto vettore di dimensioni che variano tra le centinaia di migliaia e i miliardi di parole, ciascuna delle quali è dotata del proprio indirizzo. È un magazzino di dati velocemente accessibile ed è condivisa dalla CPU e da alcuni dispositivi di I/O.
- La memoria centrale è volatile, e perde le informazioni in caso di guasto del sistema.
- Il sistema operativo è responsabile delle seguenti attività connesse alla gestione della memoria centrale:
  - ☞ tenere traccia di quali parti della memoria sono attualmente usate e da che cosa;
  - ☞ decidere quali processi si debbano caricare nella memoria quando vi sia spazio disponibile;
  - ☞ assegnare e revocare lo spazio di memoria secondo le necessità.

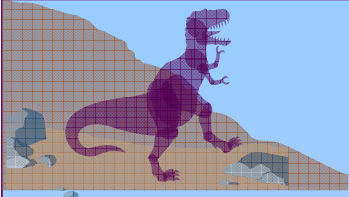




# Gestione dei file

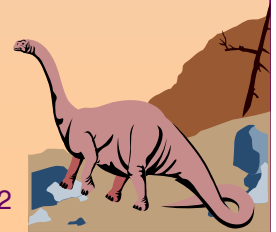
- Un file è una raccolta di informazioni correlate definite dal loro creatore. Comunemente, i file rappresentano programmi (sia sorgente sia oggetto) e dati.
- Il sistema operativo è responsabile delle seguenti attività connesse alla gestione dei file:
  - ☞ creazione e cancellazione di file;
  - ☞ creazione e cancellazione di directory;
  - ☞ fornitura delle funzioni fondamentali per la gestione di file e directory;
  - ☞ associazione dei file ai dispositivi di memoria secondaria;
  - ☞ creazione di copie di riserva (*backup*) dei file su dispositivi di memorizzazione non volatili.

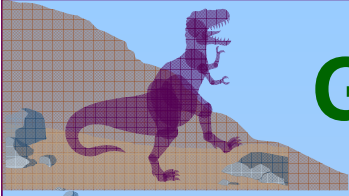




# Gestione del sistema di I/O

- Il sistema di I/O è composto dalle seguenti parti:
  - ☞ un sistema buffer-caching
  - ☞ un'interfaccia generale per i driver dei dispositivi
  - ☞ i driver per gli specifici dispositivi

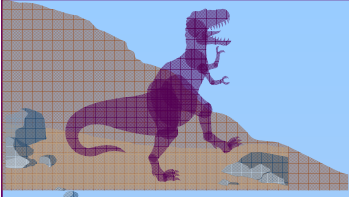




# Gestione della memoria secondaria

- Poiché la memoria centrale è troppo piccola per contenere tutti i dati e tutti i programmi, e il suo contenuto va perduto se il sistema si spegne, il calcolatore deve disporre di una **memoria secondaria** a sostegno della memoria centrale.
- La maggior parte dei moderni sistemi di calcolo impiega i dischi come principale mezzo di memorizzazione secondaria, sia per i programmi sia per i dati.
- Il sistema operativo è responsabile delle seguenti attività connesse alla gestione dei dischi:
  - ☞ gestione dello spazio libero
  - ☞ assegnazione dello spazio
  - ☞ scheduling del disco



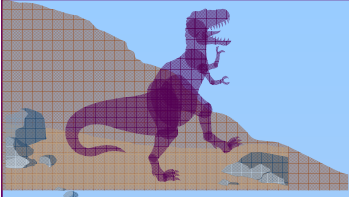


# Reti (sistemi distribuiti)

- Un **sistema distribuito** è un insieme di unità di elaborazione che non condividono la memoria, i dispositivi periferici o un clock; ciascun processore dispone di una propria memoria locale e di un suo clock.
- Le unità d'elaborazione sono collegate da una **rete di comunicazione**.
- La comunicazione avviene utilizzando un **protocollo**.
- Un sistema distribuito offre all'utente l'accesso alle varie risorse di sistema.
- L'accesso a una risorsa condivisa permette di:
  - ☞ accelerare il calcolo
  - ☞ aumentare la disponibilità dei dati
  - ☞ incrementare l'affidabilità



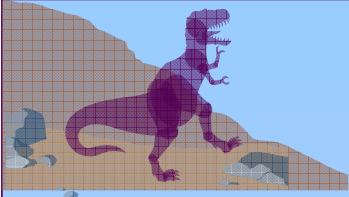




# Sistema di protezione

- La **protezione** è definita da ogni meccanismo che controlla l'accesso da parte di programmi, processi o utenti alle risorse di un sistema di calcolo.
- Il meccanismo di protezione deve:
  - ☞ distinguere tra uso autorizzato e non autorizzato.
  - ☞ specificare i controlli che devono essere attivati.
  - ☞ fornire strumenti di miglioramento dell'affidabilità.

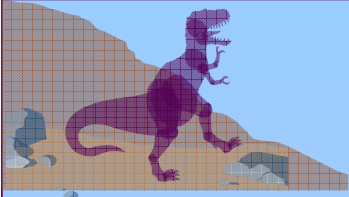




# Interprete dei comandi

- Molti comandi si impartiscono al sistema operativo attraverso istruzioni di controllo che riguardano:
  - creazione e gestione di processi
  - I/O
  - gestione della memoria secondaria
  - gestione della memoria centrale
  - accesso al file-system
  - protezione
  - reti

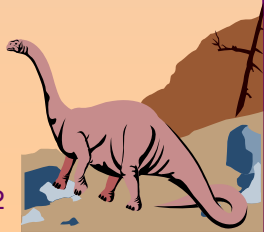


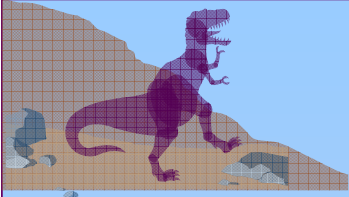


# Interprete dei comandi (*Cont.*)

- Il programma che legge e interpreta le istruzioni di controllo ha diversi nomi:
  - ☞ interprete di schede di controllo (*control-card interpreter*)
  - ☞ interprete di riga di comando (*command-line interpreter*)
  - ☞ *shell* (nello UNIX)

La sua funzione consiste nel prelevare ed eseguire la successiva istruzione di comando.

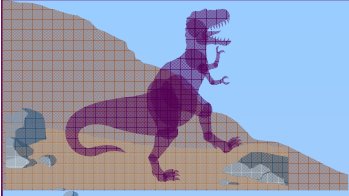




# Servizi di un sistema operativo

- **Esecuzione di un programma:** capacità del sistema di caricare un programma nella memoria ed eseguirlo.
- **Operazioni di I/O:** poiché i programmi utenti non possono eseguire direttamente operazioni di I/O, il sistema operativo deve offrire mezzi adeguati.
- **Gestione del file system:** capacità del programma di leggere, scrivere, creare e cancellare file.
- **Comunicazioni:** scambio di informazioni tra processi in esecuzione nello stesso calcolatore e tra processi in esecuzione in calcolatori diversi collegati per mezzo di una rete. La comunicazione si può realizzare tramite una *memoria condivisa* o attraverso lo *scambio di messaggi*.
- **Rilevamento d'errori:** capacità di rilevare eventuali errori che possono verificarsi nella CPU e nei dispositivi di memoria, nei dispositivi di I/O e nei programmi utenti.



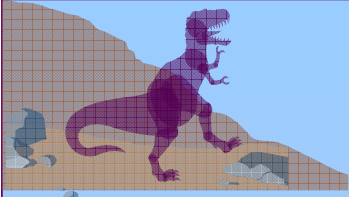


# Funzioni aggiuntive di un sistema operativo

Esiste anche un'altra serie di funzioni del sistema operativo che non riguarda direttamente gli utenti, ma assicura il funzionamento efficiente del sistema stesso.

- **Assegnazione delle risorse:** se sono in corso più sessioni di lavoro di utenti o sono contemporaneamente in esecuzione più processi, il sistema operativo provvede all'assegnazione delle risorse necessarie a ciascuno di essi.
- **Contabilizzazione dell'uso delle risorse:** registrazione degli utenti che usano il calcolatore, con segnalazione di quali e quante risorse vengono impiegate, a fini di addebito dei costi o di preparazione di statistiche.
- **Protezione:** assicura che l'accesso alle risorse del sistema sia controllato.



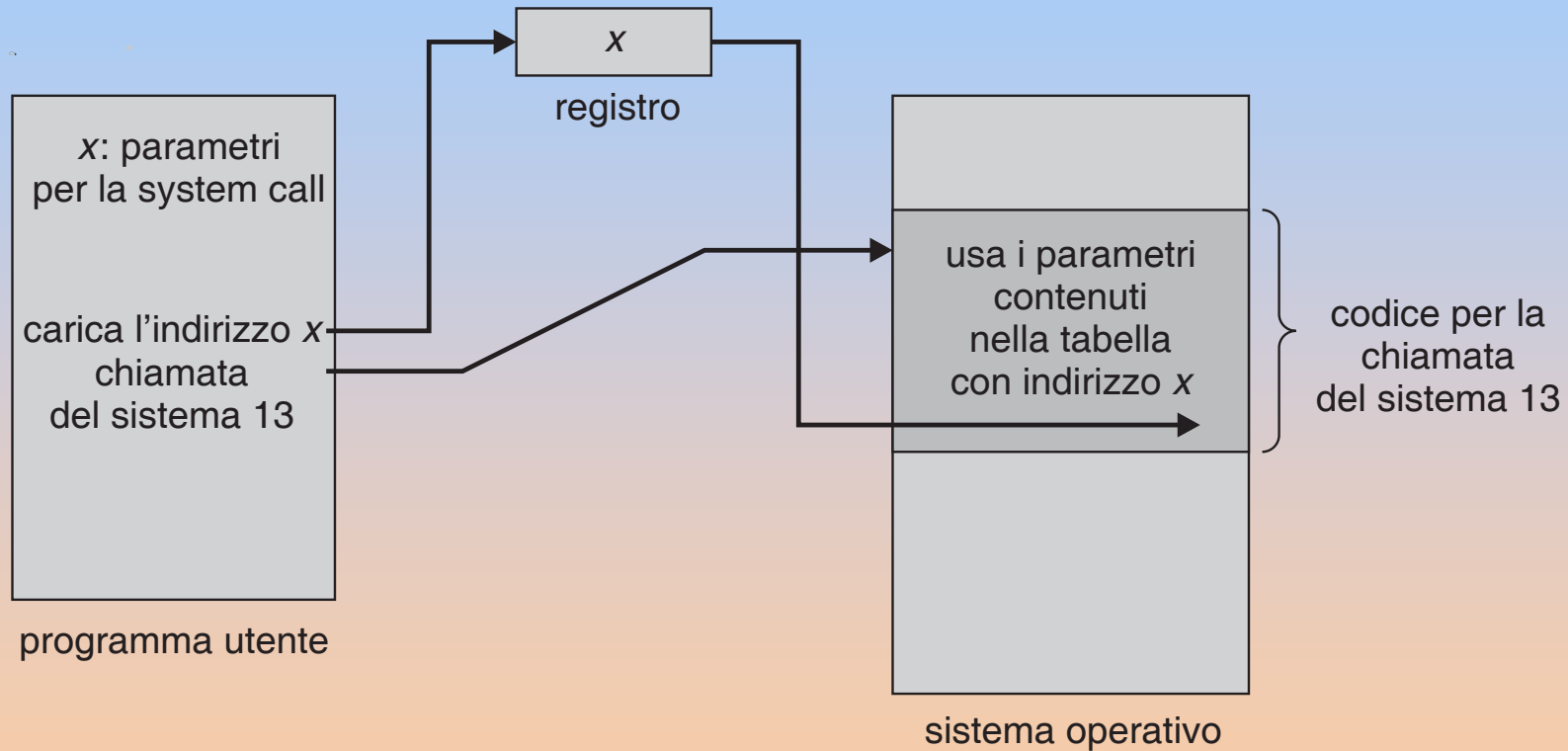


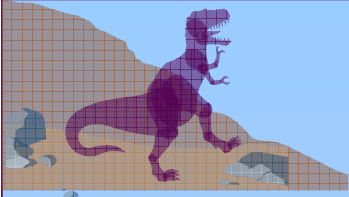
# Chiamate del sistema

- Le chiamate del sistema costituiscono l'interfaccia tra un processo e il sistema operativo.
  - ☞ Generalmente disponibili in forma di istruzioni in linguaggio assemblativo.
  - ☞ Certi sistemi consentono che le chiamate del sistema siano invocate direttamente da un programma scritto in un linguaggio di alto livello (es.: C, C++, Perl)
- Per passare parametri al sistema operativo si usano tre metodi generali.
  - ☞ Passare i parametri in *registri*.
  - ☞ Memorizzare i parametri in un *blocco* o tabella di memoria e passare l'indirizzo del blocco, in forma di parametro, in un registro.
  - ☞ *Collocare (push)* i parametri in una pila da cui sono prelevati (*pop*) dal sistema operativo.



# Passaggio di parametri in forma di tabella



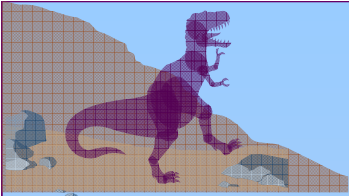


# Tipi di chiamate del sistema

- Controllo dei processi
- Gestione dei file
- Gestione dei dispositivi
- Gestione delle informazioni
- Comunicazione



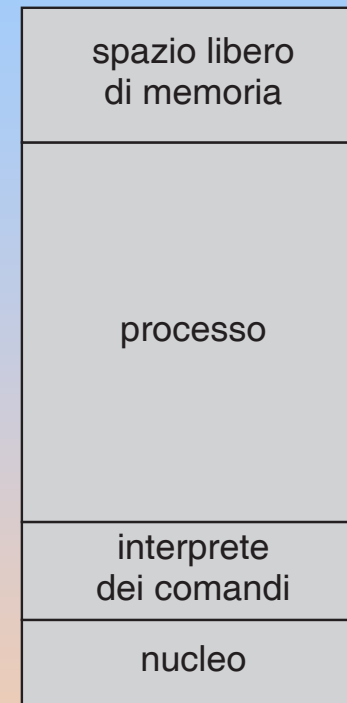




# Esecuzione nell'MS-DOS



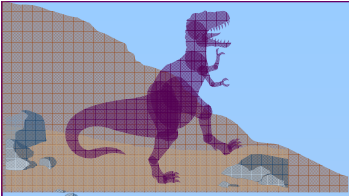
(a)



(b)

All'avviamento del sistema    Durante l'esecuzione di un programma



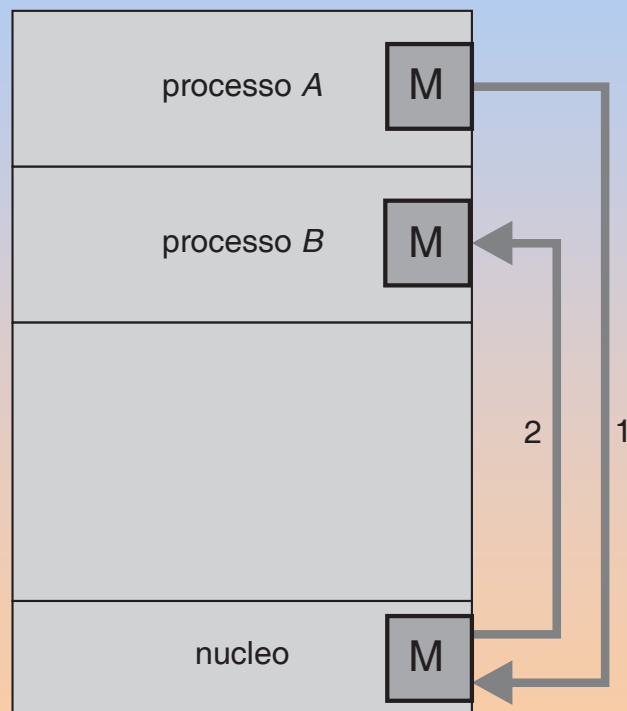


# Esecuzione di più programmi nel sistema operativo UNIX



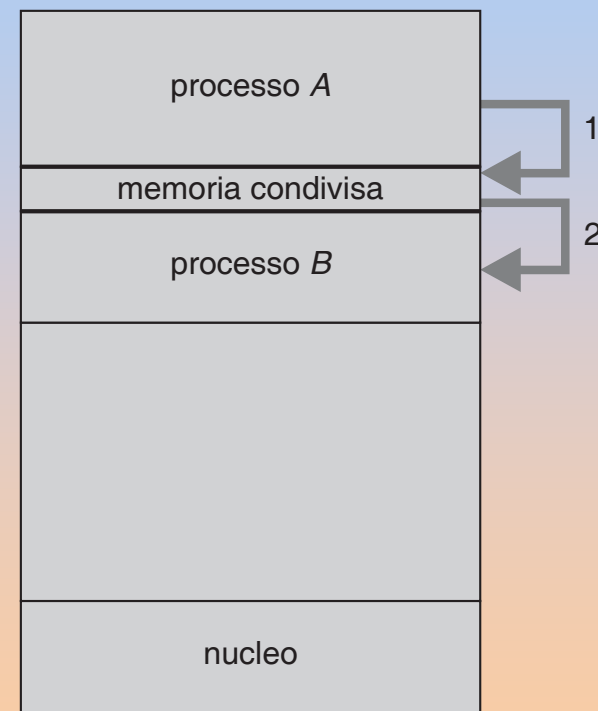
# Modelli di comunicazione

- La comunicazione può avvenire o attraverso il modello a **scambio di messaggi** o mediante il modello a **memoria condivisa**.



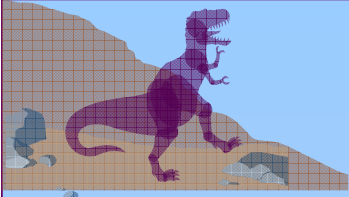
(a)

Scambio di messaggi



(b)

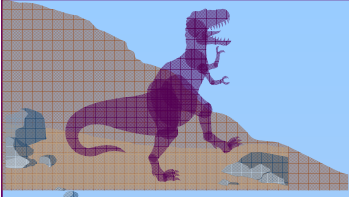
Memoria condivisa



# Programmi di sistema

- I programmi di sistema offrono un ambiente conveniente per lo sviluppo e l'esecuzione dei programmi; in generale si possono classificare nelle seguenti categorie:
  - ☞ Gestione dei file
  - ☞ Informazioni di stato
  - ☞ Modifica dei file
  - ☞ Ambienti d'ausilio alla programmazione
  - ☞ Caricamento ed esecuzione dei programmi
  - ☞ Comunicazioni
  - ☞ Programmi d'applicazione
- Per la maggior parte degli utenti, l'interfaccia col sistema operativo è definita dai programmi di sistema piuttosto che dalle effettive chiamate di sistema.





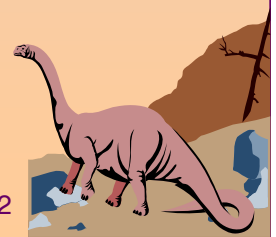
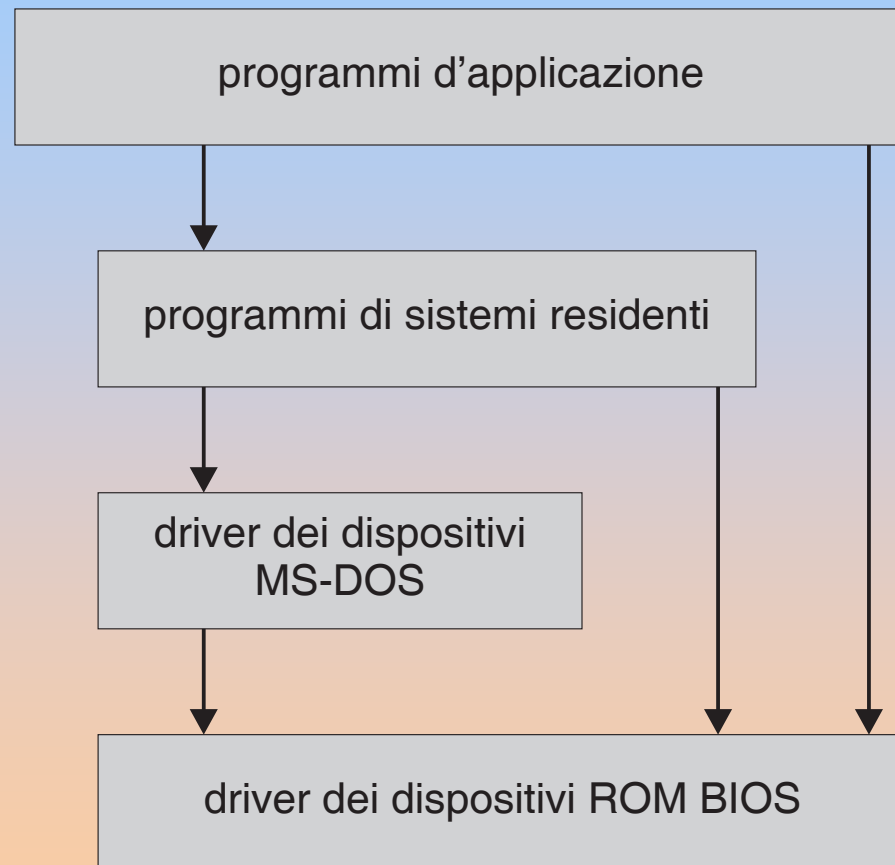
# Struttura del sistema MS-DOS

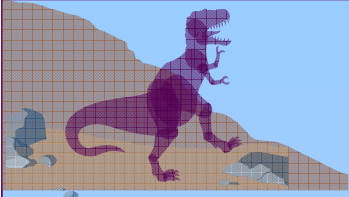
- MS-DOS: progettato per fornire la massima funzionalità nel minimo spazio
  - ☞ non suddiviso in moduli
  - ☞ anche se dotato di una semplice struttura, le sue interfacce e i livelli di funzionalità non sono ben separati.





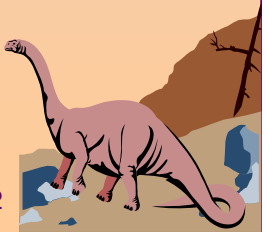
# Struttura degli strati dell'MS-DOS

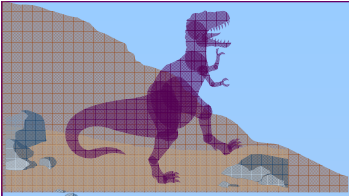




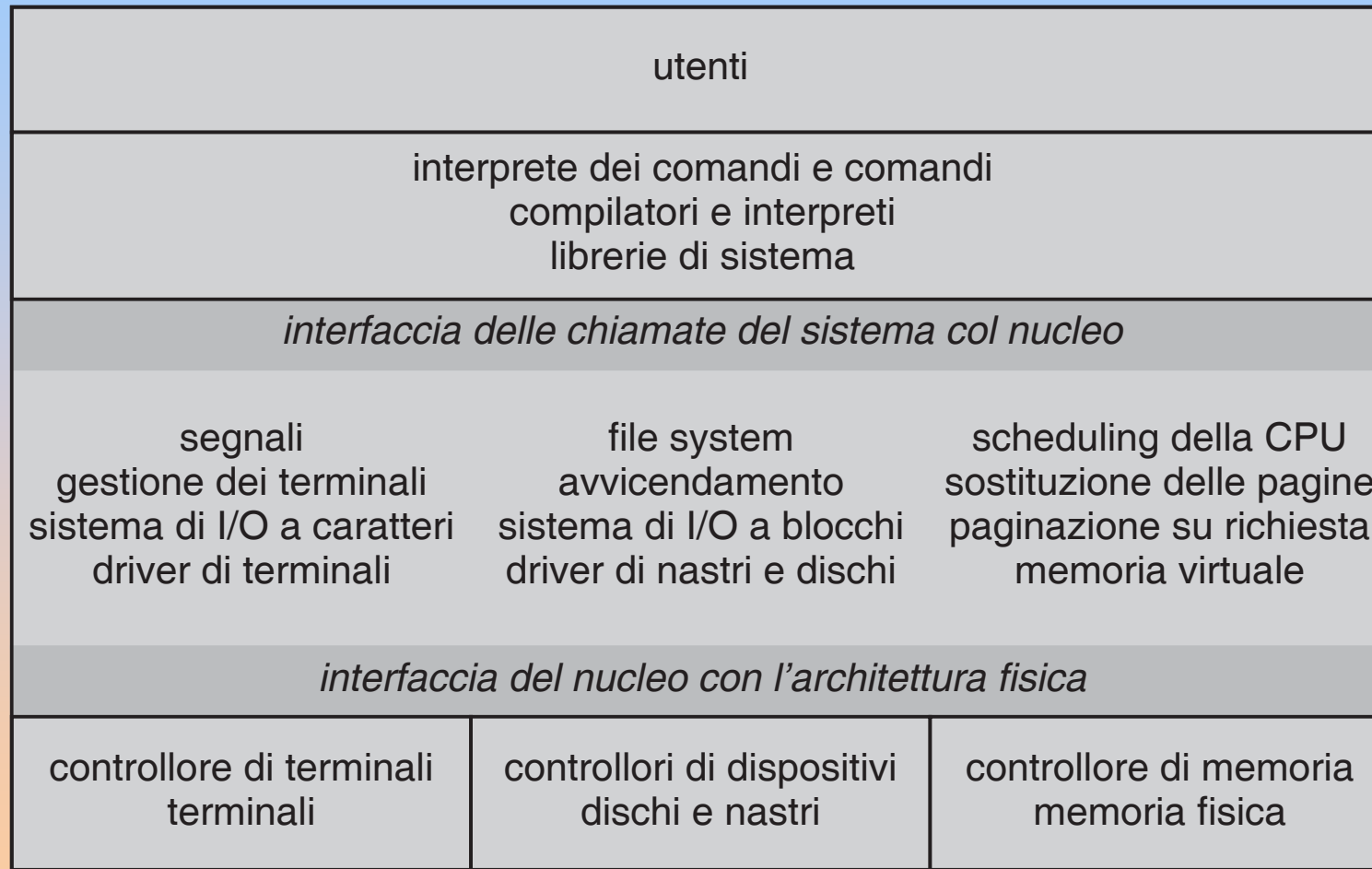
# Struttura del sistema UNIX

- Lo UNIX è un altro esempio di strutturazione che inizialmente era limitata dalle funzioni dell'architettura sottostante. E' formato da due parti:
  - Programmi di sistema
  - Nucleo:
    - ☞ tutto quello che si trova sotto l'interfaccia delle chiamate di sistema e sopra i dispositivi fisici è il nucleo
    - ☞ fornisce il file system, lo scheduling della CPU, la gestione della memoria e altre funzioni riguardanti il sistema operativo: in un solo livello sono combinate un'enorme quantità di funzioni.

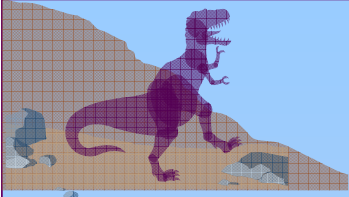




# Struttura del sistema UNIX

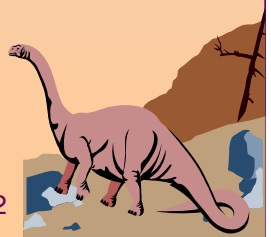




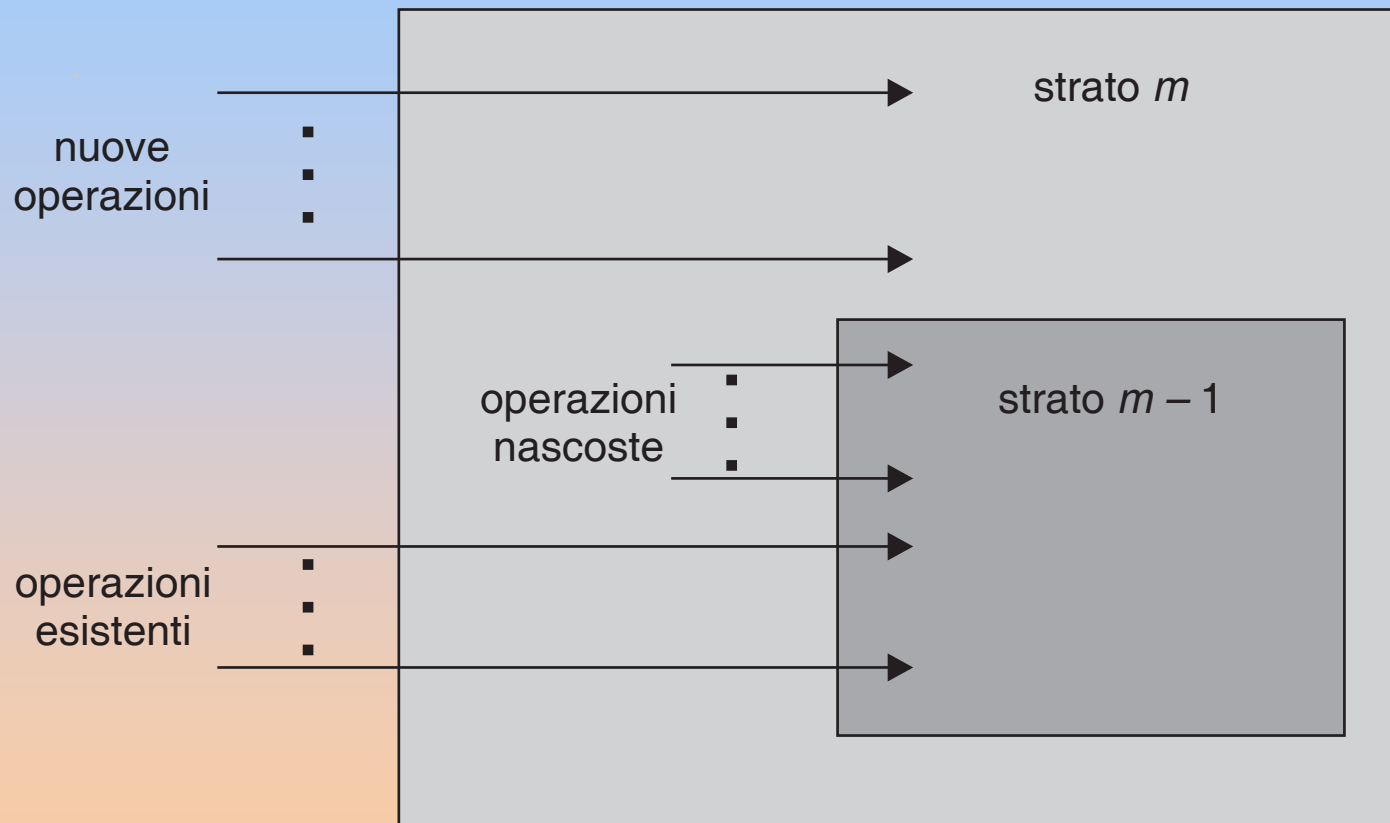


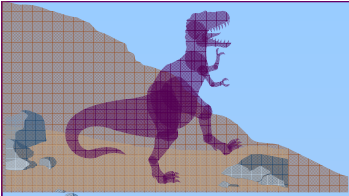
# Metodo stratificato

- Con il metodo stratificato si suddivide il sistema operativo in un certo numero di strati (o livelli), ciascuno costruito sopra gli strati inferiori. Lo strato più basso (0) è lo strato fisico; quello più alto (strato  $n$ ) è l'interfaccia d'utente.
- Ogni strato si realizza impiegando unicamente le operazioni messe a disposizione dagli strati inferiori.

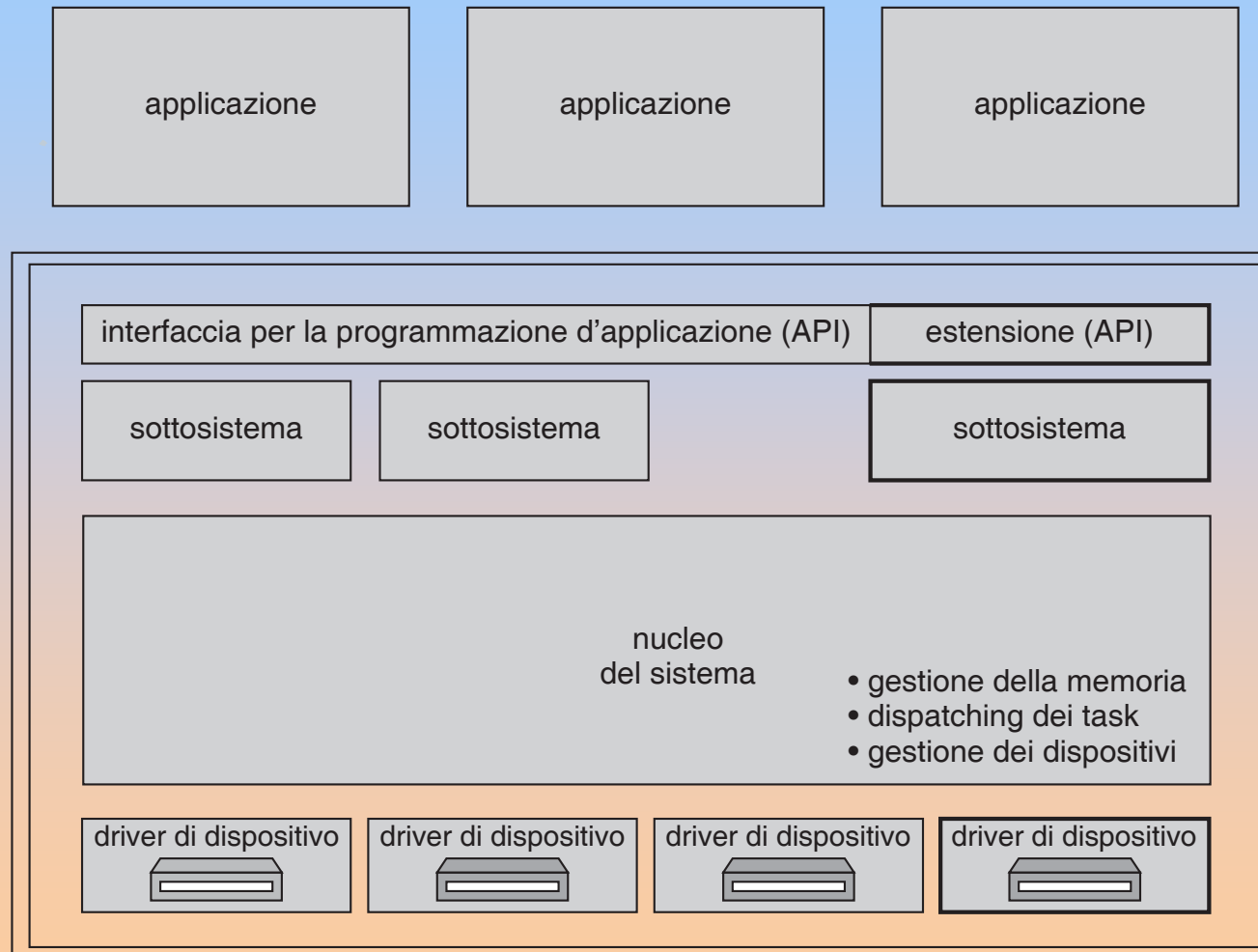


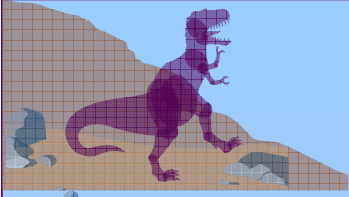
# Uno strato di sistema operativo





# Struttura stratificata dell'OS/2

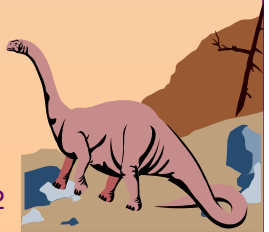




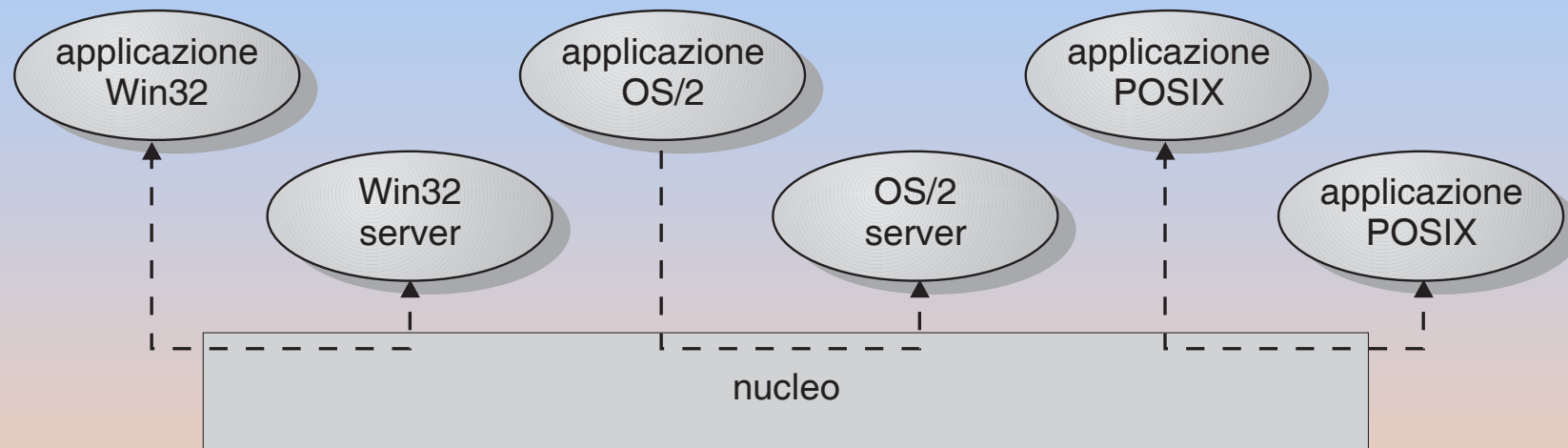
# Microkernel

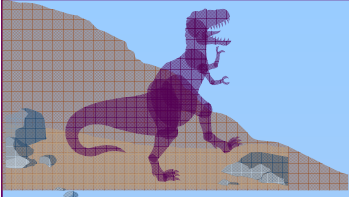
## Orientamento a micronucleo

- Secondo questo orientamento, si progetta il sistema operativo rimuovendo dal nucleo tutti i componenti non essenziali, realizzandoli come programmi del livello d'utente e di sistema.
- La comunicazione si realizza secondo il modello a scambio di messaggi.
- Vantaggi:
  - facilità di estensione del sistema operativo
  - più semplice da adattare alle diverse architetture
  - più affidabile (i servizi si eseguono in gran parte come processi utenti, non come processi del nucleo)
  - più sicuro



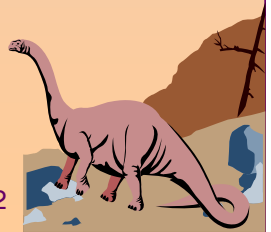
# Struttura client-server del sistema operativo Windows NT

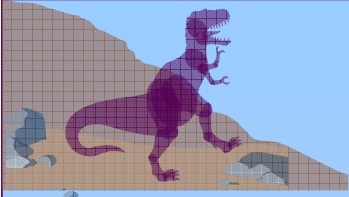




# Macchine virtuali

- Il concetto di **macchina virtuale** si sviluppa logicamente dal metodo stratificato. I programmi d'applicazione possono considerare quello che si trova a un livello gerarchico inferiore come se fosse parte della macchina stessa, anche se i programmi di sistema si trovano a un livello superiore.
- Una macchina virtuale è un'interfaccia *identica* all'architettura sottostante.
- Il sistema operativo crea l'illusione che un processo disponga della propria CPU con la propria memoria (virtuale).



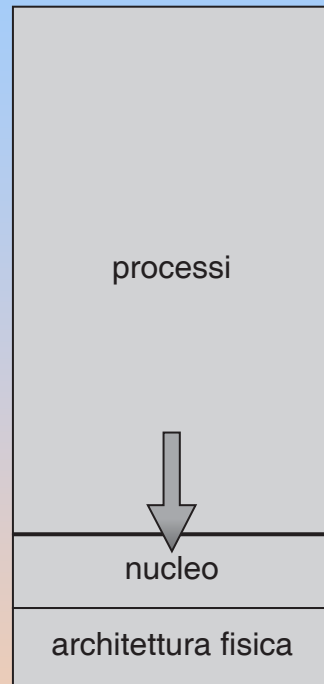


# Machine virtuali (Cont.)

- Il calcolatore fisico condivide le risorse in modo da creare macchine virtuali.
  - ☞ La partizione del tempo d'uso della CPU si può usare sia per condividere la CPU sia per dare l'illusione che gli utenti dispongano di una propria CPU.
  - ☞ La gestione asincrona delle operazioni di I/O e dell'esecuzione di più processi, unita a un file system, consente di creare lettori di schede e stampanti virtuali.
  - ☞ Un normale terminale di un sistema a partizione del tempo funziona da console d'operatore della macchina virtuale.

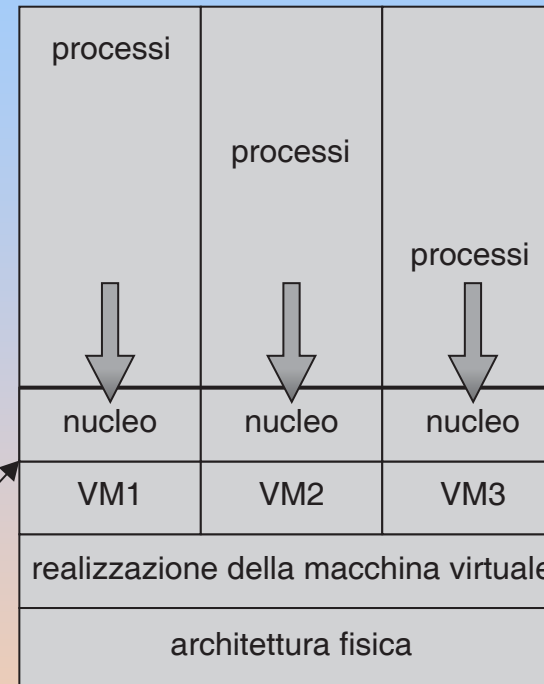


# Modelli di sistema



(a)

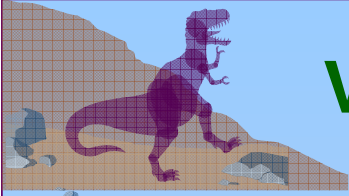
Semplice



(b)

Macchina virtuale

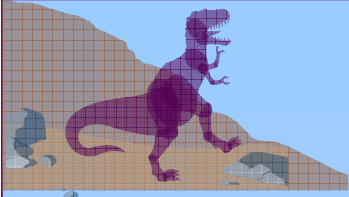




# Vantaggi/svantaggi delle macchine virtuali

- L'uso delle macchine virtuali protegge completamente le risorse di sistema poiché ciascuna macchina virtuale è isolata dalle altre. Uno svantaggio di questo tipo d'ambiente è che non c'è una condivisione *diretta* delle risorse.
- Un sistema di macchine virtuali è un perfetto mezzo di ricerca e sviluppo dei sistemi operativi. Lo sviluppo avviene sulla macchina virtuale non sulla macchina fisica, evitando così modifiche che potrebbero causare oscuri errori di programmazione in altri punti.
- A una maggiore complessità della macchina da emulare corrisponde una maggiore difficoltà di realizzazione di un'accurata macchina virtuale, e una maggiore lentezza nell'esecuzione.





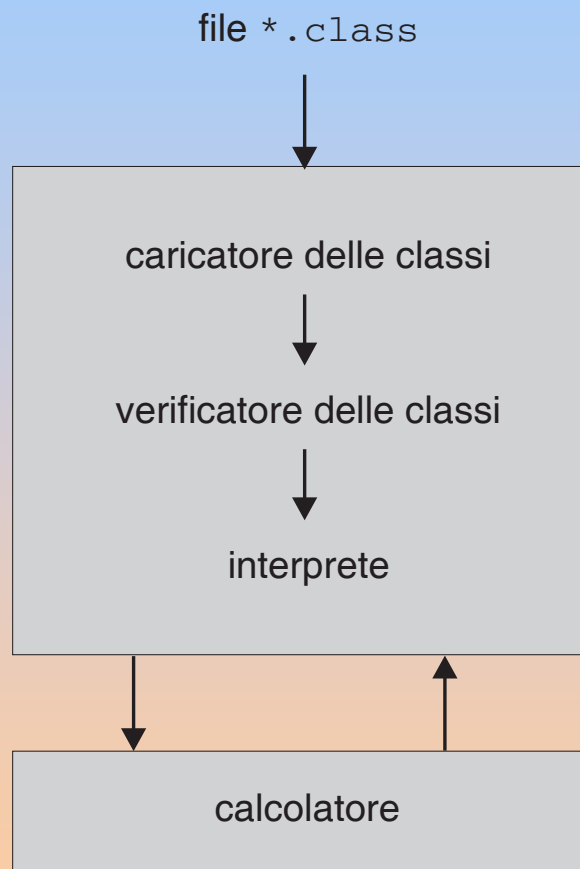
# Macchina virtuale Java

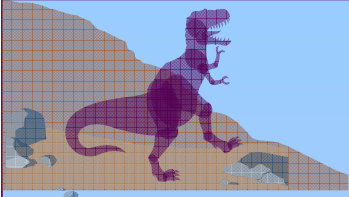
- I programmi Java sono *bytecode* indipendente dall'architettura sottostante eseguiti dalla macchina virtuale Java (JVM, *Java Virtual Machine*).
- La JVM consiste di:
  - un caricatore delle classi
  - un verificatore delle classi
  - un interprete del linguaggio che esegue il bytecode
- Il compilatore istantaneo Just-In-Time (JIT) ne migliora le prestazioni





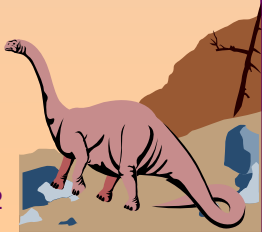
# Macchina virtuale dell'ambiente Java

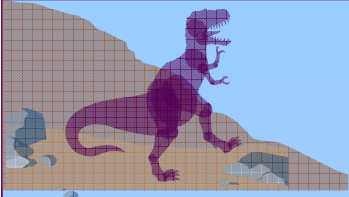




# Scopi della progettazione

- **Scopi degli utenti:** gli utenti desiderano che un sistema sia utile, facile da imparare e usare, affidabile, sicuro ed efficiente.
- **Scopi del sistema:** il sistema operativo deve essere di facile progettazione, realizzazione e manutenzione; deve essere flessibile, affidabile, senza errori ed efficiente.

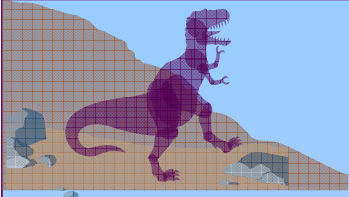




# Meccanismi e criteri

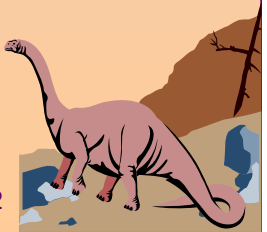
- I **meccanismi** determinano *come* eseguire qualcosa; i **criteri** invece stabiliscono *cosa* si debba fare.
- La distinzione tra meccanismi e criteri è molto importante, e consente la massima flessibilità, poiché i criteri sono soggetti a cambiamenti rispetto alle situazioni o ai momenti.

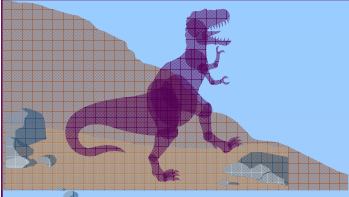




# Realizzazione

- Tradizionalmente, i sistemi operativi si scrivevano in un linguaggio assembler; attualmente si scrivono spesso in linguaggi di alto livello come il C o il C++.
- Il codice in un linguaggio di alto livello:
  - ☞ può essere scritto più rapidamente
  - ☞ è più compatto
  - ☞ è più facile da capire e mettere a punto.
- Un sistema operativo scritto in un linguaggio di alto livello è più facile da adattare a un'altra architettura (*porting*).





# Generazione di sistemi (SYSGEN)

- I sistemi operativi sono progettati per un impiego su macchine di una stessa classe con configurazioni diverse.
- Il processo di generazione di sistemi (SYSGEN) configura o genera il sistema per ciascuna situazione specifica.
- **Booting**: avviamento di un calcolatore attraverso il caricamento del nucleo.
- **Bootstrap program**: piccolo segmento di codice memorizzato in una ROM che individua il nucleo, lo carica nella memoria e ne avvia l'esecuzione.

