

Abilità Informatiche



Indice

- Il sistema operativo
- Architettura di un sistema operativo
- Bootstrap
- Panoramica dei principali sistemi operativi
 - MS DOS, MS Windows, UNIX, Linux, BSD, Mac OS

Software

- SW di sistema: controlla il comportamento del sistema stesso
 - controlla le risorse del sistema
 - semplifica l'interazione dell'utente col sistema
 - fornisce la base (servizi) su cui costruire tutti gli altri programmi
- SW applicativo: cerca di risolvere i problemi dell'utente, appoggiandosi sullo strato fornito dal sistema operativo

Premessa

- Macchina di Von Neumann
 - Uso della stessa memoria per istruzioni e dati
 - Processore singolo
 - Ma i concetti sono gli stessi anche per l'SMP
 - Elaborazione sequenziale
- Usare efficientemente le risorse → dividerle
- Problema: allocare e controllare le risorse condivise

Disaccoppiamento

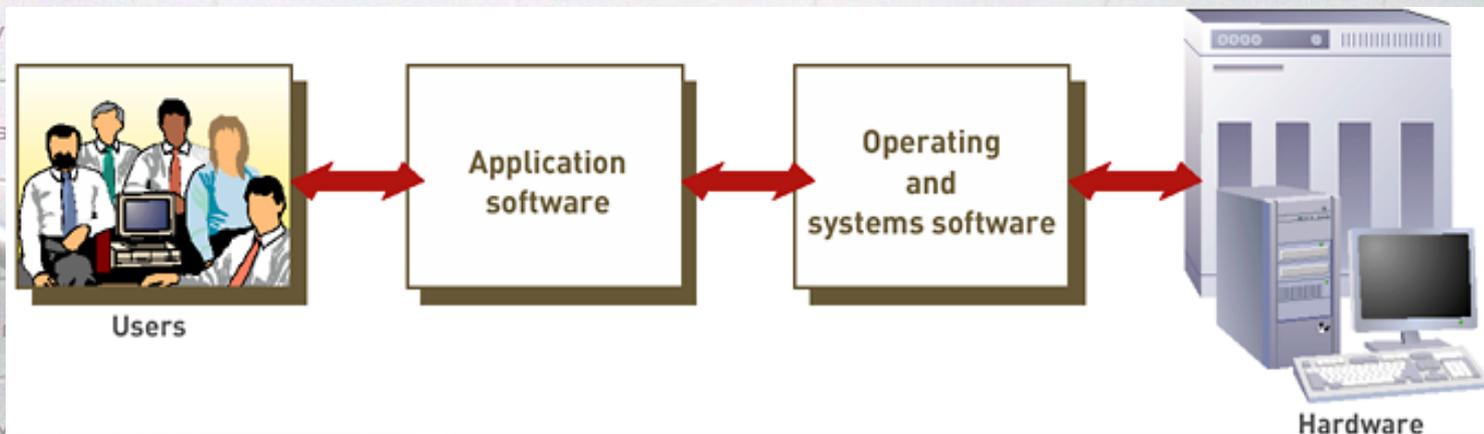
- L'evoluzione tecnologica produce innovazioni nei componenti HW e SW
 - A volte queste innovazioni hanno impatto sulle funzionalità dei programmi che usano tali componenti
- Q: Ha senso obbligare i programmatori di un prodotto (p. es. Firefox) ad aggiornarlo per ogni innovazione dell'hardware?
- Q: Come garantire l'intercambiabilità tra dispositivi di fornitori diversi?
- Introdurre punti di flessibilità permette di svincolare le linee di sviluppo

Coordinamento di dispositivi con velocità diverse

- I dispositivi elettronici sono più veloci di quelli elettromeccanici
- Simultaneous Peripheral Operation On-Line (spool)
 - Uso dei dischi come buffer per le periferiche lente
 - Rifletteteci quando stampate

Il sistema operativo

- Con il termine *sistema operativo* si intende l'insieme di programmi e librerie che opera direttamente sulla macchina fisica...
 - ... mascherandone le caratteristiche specifiche...
 - ... e fornendo agli utenti un insieme di funzionalità di alto livello (macchina astratta)



Il S.O. come infrastruttura

- Infrastruttura di supporto all'esecuzione concorrente di più processi da parte di più utenti
- Processo = programma in esecuzione
- I processi utente applicativi non possono stabilire autonomamente quando e come accedere alle risorse del sistema, ma sono costretti a chiedere al sistema operativo l'esecuzione di servizi

Virtualizzazione delle risorse

- Il numero di processi può essere superiore al numero di processori
- È «come se» ci fossero più unità di elaborazione (virtuali)
- Ogni processo «non ha ragione di dubitare» di avere il processore tutto per sé

Virtualizzazione delle risorse

- Allo stesso modo ogni processo «crede» di avere a disposizione tutta la memoria
 - È «come se» gli altri processi girassero su altre macchine
- Il S.O. gestisce la condivisione mascherando la collocazione fisica dei dati

Virtualizzazione delle risorse

- Ricordate le gerarchie di memoria?
 - Perché funzionano?
 - Località degli accessi
- Lo spazio di memoria utilizzabile può essere superiore alla capacità della memoria fisica
 - Si può *simulare* una memoria più estesa trasferendo opportunamente dati da e verso il disco
 - Tenendo in RAM solo le parti di codice e dei dati che servono in quel momento
- È «come se» ci fosse più memoria (memoria virtuale)
 - Ma le prestazioni ne risentono

Virtualizzazione delle risorse

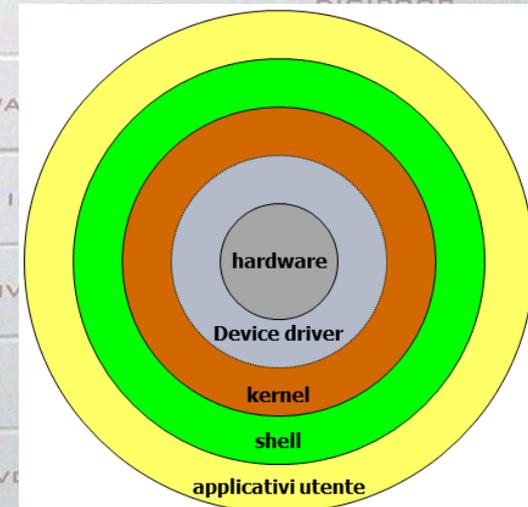
- Portando il concetto all'estremo ...

Cloud

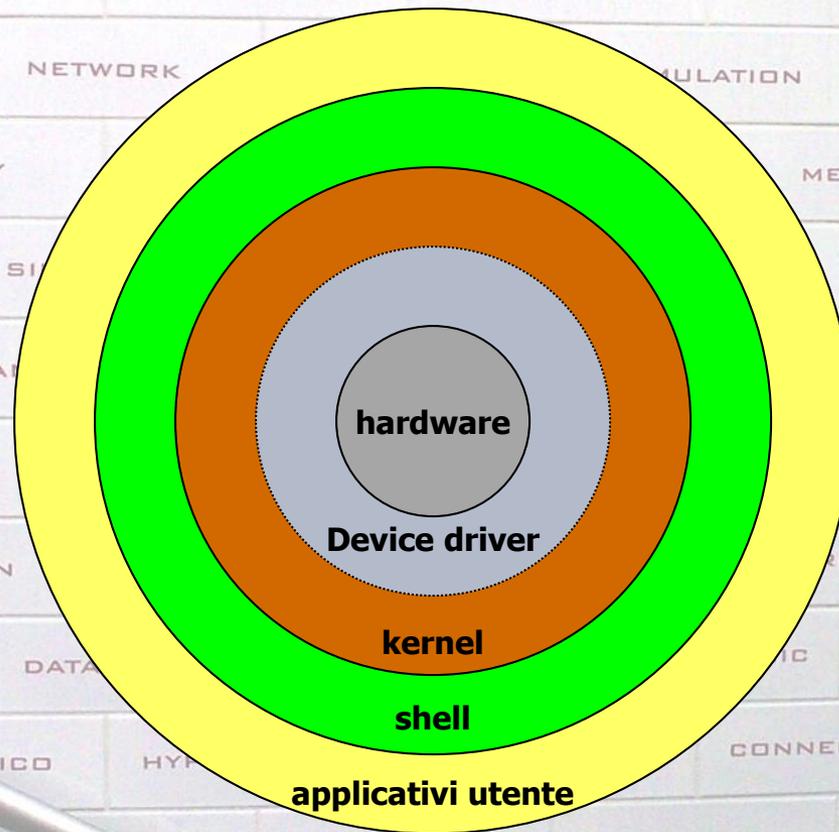


Organizzazione a strati

- Architettura «a cipolla»
- Ogni strato
 - Tramite le funzionalità offerte dallo strato precedente
 - Implementa una macchina astratta più potente
- Ogni macchina astratta è sempre meno legata ai dettagli dell'hardware



Organizzazione a strati



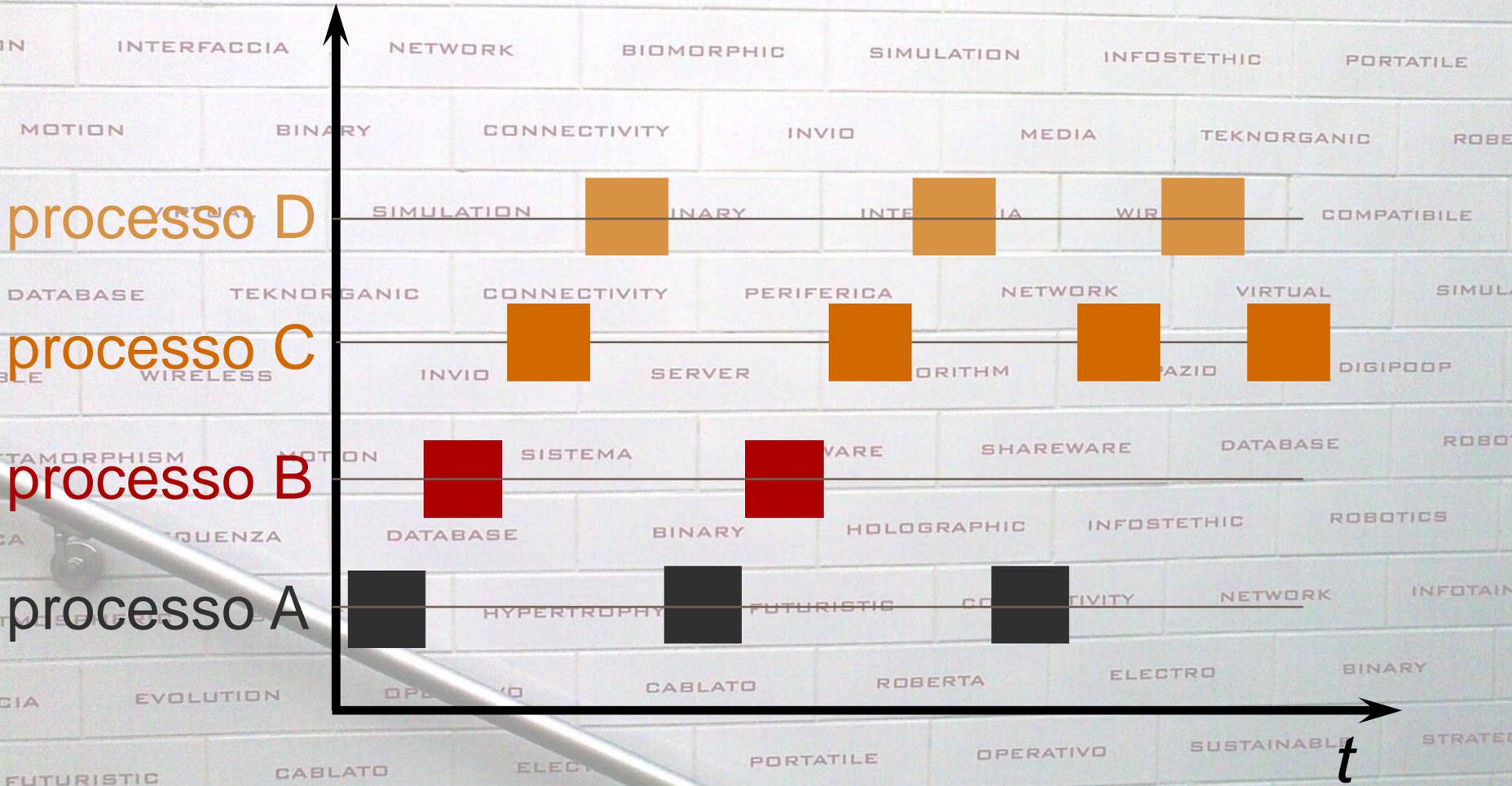
Organizzazione per componenti

- Gestione dei processi
- Gestione della memoria centrale
- Gestione dei file
- Gestione dell'I/O
- Reti
- Protezione
- Interprete dei comandi

Gestione dei processi

- Un processo è un programma in esecuzione
- **Virtualizzazione della CPU**
 - «delle» CPU, se il sistema è multiprocessore
- Il S.O. esegue più processi in modo *apparentemente simultaneo*, dedicando *ciclicamente* la CPU a ognuno di essi per piccoli *quanti di tempo*
- Vantaggio: un processore veloce è tenuto sempre occupato

Time-sharing

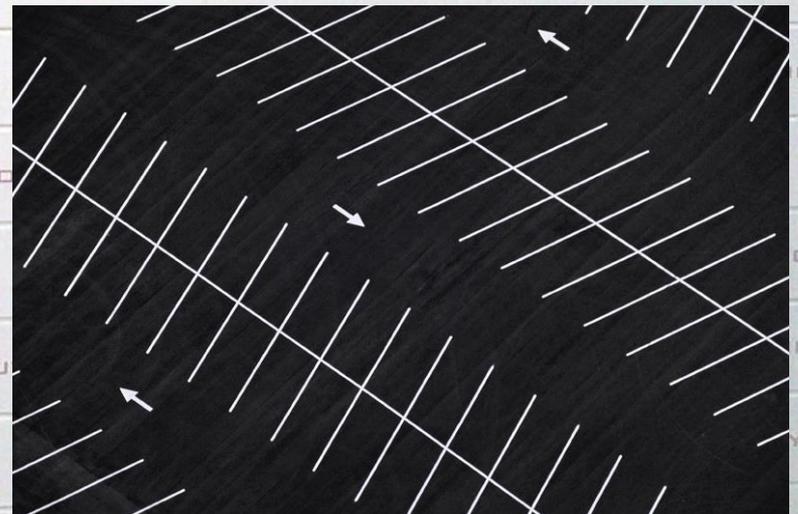


Il gestore della memoria

- Il gestore della memoria
 - Realizza le funzionalità di allocazione della memoria
 - Supera i limiti della memoria fisica e mostra ai processi uno spazio di memoria virtuale
 - Partiziona la memoria tra i vari processi che la richiedono
 - Più processi possono essere caricati in memoria contemporaneamente
 - Occorre garantire la protezione delle diverse zone di memoria

Paginazione

- Le aree di memoria dei processi vanno caricate, quando necessario, nella RAM
- Come organizzarle?
 - Paragone: parcheggio (non precisissimo; perché?)



Paginazione

- Paginazione: Divisione della memoria in blocchi (**pagine**) di dimensione fissa
- Si sacrifica un po' di spazio in cambio di facilità di gestione

Gestione dell'I/O

- Il controllo dei dispositivi di I/O avviene attraverso speciali programmi detti *Device Driver*
 - Spesso realizzati dai produttori dei dispositivi
- Mascherano le caratteristiche di dettaglio dei dispositivi
 - Ecco perché ci sono nella confezione di una periferica c'è spesso un CD
- Disciplinano gli accessi simultanei

Device driver

- Isolamento delle peculiarità dell'hardware
 - Altrimenti il S.O. dovrebbe supportare direttamente tutto l'hardware compatibile col computer
 - Il S.O. può, e deve, essere sviluppato ignorando l'hardware specifico
- *Modus operandi*
 - Il device driver riceve richieste generiche, *device-independent*,
 - e le onora manipolando l'hardware
- Interfaccia standard verso l'hardware, che standard non è

Plug and play

- Collegamento di un dispositivo
- Il dispositivo «si fa riconoscere»
- Il S.O. seleziona un driver
 - Scelta del «miglior» driver per il dispositivo
- Installazione del driver
 - Indicazione al S.O. di usare quello specifico driver per il dispositivo

Gestione dei file

- Astrazione della memoria di massa
- Un **file** è una raccolta di informazioni correlate
- Meccanismo per l'identificazione dei file
- Rendere trasparente la struttura fisica del supporto di memorizzazione
- Protezione dei dati degli utenti
 - Garantire l'accesso ai dati per gli utenti autorizzati e negarlo agli altri

Gestione dei file

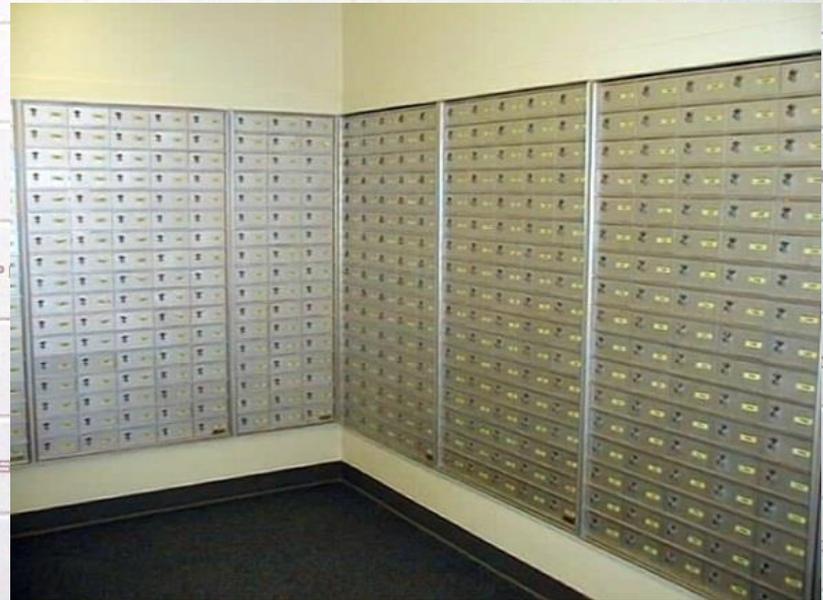
- Recupero di dati precedentemente memorizzati
- Eliminazione di dati obsoleti
- Modifica e aggiornamento di dati preesistenti
- Copia di dati
 - Tra supporti di memorizzazione diversi (ad es. da HD a CD)
 - In aree diverse nello stesso supporto

File

- Un file memorizza permanentemente una serie di informazioni
 - Una applicazione (elaboratore testi, visualizzatore di clip multimediali, ...)
 - Dati (un documento, un video musicale, ecc...)
- Un file è identificato da un nome (filename)
 - Come è fatto lo spazio dei nomi?

Namespace piatto

- Non possono esistere due oggetti con lo stesso nome
- Q: ma allora, se due utenti vogliono chiamare un file «foto1.jpg» che succede?
- Q: cercare tra migliaia di oggetti; e la scalabilità?



Organizzazione gerarchica

- Quasi tutti i sistemi operativi utilizzano un'organizzazione *gerarchica* del File System
- L'elemento utilizzato per raggruppare più file insieme è la *directory*
- Una directory (o cartella, o folder) è un contenitore di oggetti, che possono essere file o altre directory

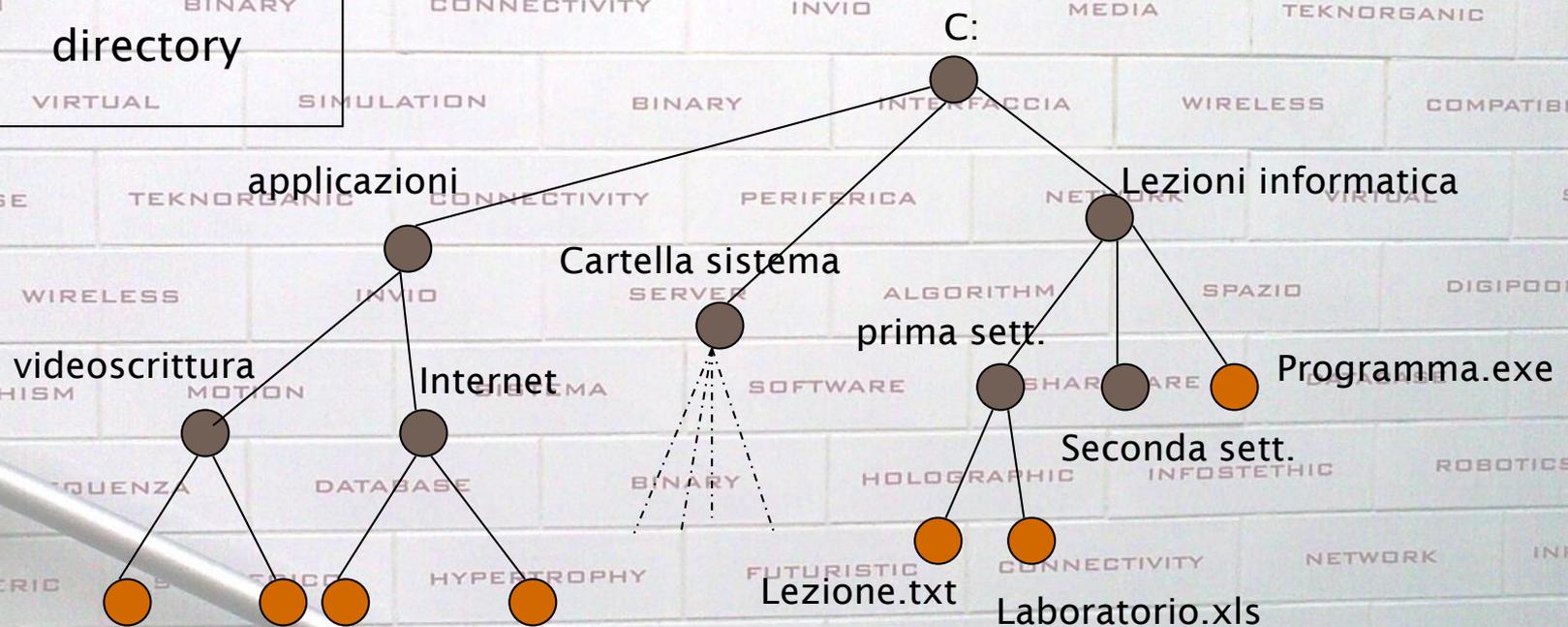
Directory

- File system strutturati
 - informazioni memorizzate in modo ordinato
 - localizzare efficientemente le informazioni
- Progettare la struttura delle directory

Organizzazione gerarchica

- Un file system ha quindi una struttura gerarchica (in particolare, ad albero) nella quale
 - I file sono le estremità (foglie)
 - Le directory sono i nodi intermedi
 - Q: ma ci deve essere una cartella che contiene tutte le altre
 - A: esatto, la si chiama **root** (radice)

Struttura ad albero



Organizzazione ad albero

- Ciascun file, quindi, viene univocamente identificato attraverso il percorso nell'albero (percorso assoluto perché parte dalla radice)
- Il nome completo si ottiene scrivendo ordinatamente i nomi dei vari nodi che si incontrano a partire dalla radice e intervallati dal carattere di separazione, che dipende dal S.O.
 - Barra (slash) per Unix '/'
 - Barra retroversa (backslash) per Windows '\'
- È possibile avere due file con lo stesso filename **purché in cartelle diverse.**

Organizzazione ad albero

- Ciascun processo ha in ogni istante definita una **directory corrente**, che identifica la posizione attuale nel file system
- È possibile quindi identificare un file anche con il suo percorso **relativo**, partendo cioè dalla directory corrente

Il gestore del file system

- Ad ogni file sono poi associati altri dati
 - Date di creazione / modifica
 - Lunghezza del file (in byte)
 - Utenti/Gruppi che possono accedere ai file
- Protezione dei dati di un utente dagli altri utenti

Protezione

- I processi vanno protetti gli uni dagli altri
- L'accesso all'hardware va disciplinato
- È necessaria per la modularità
 - Si evita la propagazione degli errori
- Rilevamento degli errori
- Permette di controllare l'uso del sistema
 - Livelli di autorizzazione graduati

Interprete dei comandi

- Interfaccia fra l'utente ed i sistema
- CUI (Character User Interface)
 - Terminale, Prompt dei comandi
- GUI (Graphical User Interface)
 - Windows, X-Windows, Mac-OS
- L'interprete dei comandi è un modulo a cui l'utente può direttamente accedere
- La sua funzione è quella di interpretare i comandi che gli giungono e di attivare i programmi corrispondenti

Bootstrap: avvio del S.O.

- All'inizio, la CPU «punta» a un indirizzo nella ROM dove inizia il programma di bootstrap che è sempre memorizzato lì
- Il programma di bootstrap
 - trasferisce il kernel del Sistema Operativo da una parte prestabilita della memoria di massa (hard-disk, floppy, CD,USB,...) in memoria principale
 - esegue il kernel
- Tipicamente, il kernel, appena avviato, fa una ricognizione dell'hardware e carica altri componenti software:
 - driver delle periferiche installate
 - programmi di sistema lanciati automaticamente all'avvio

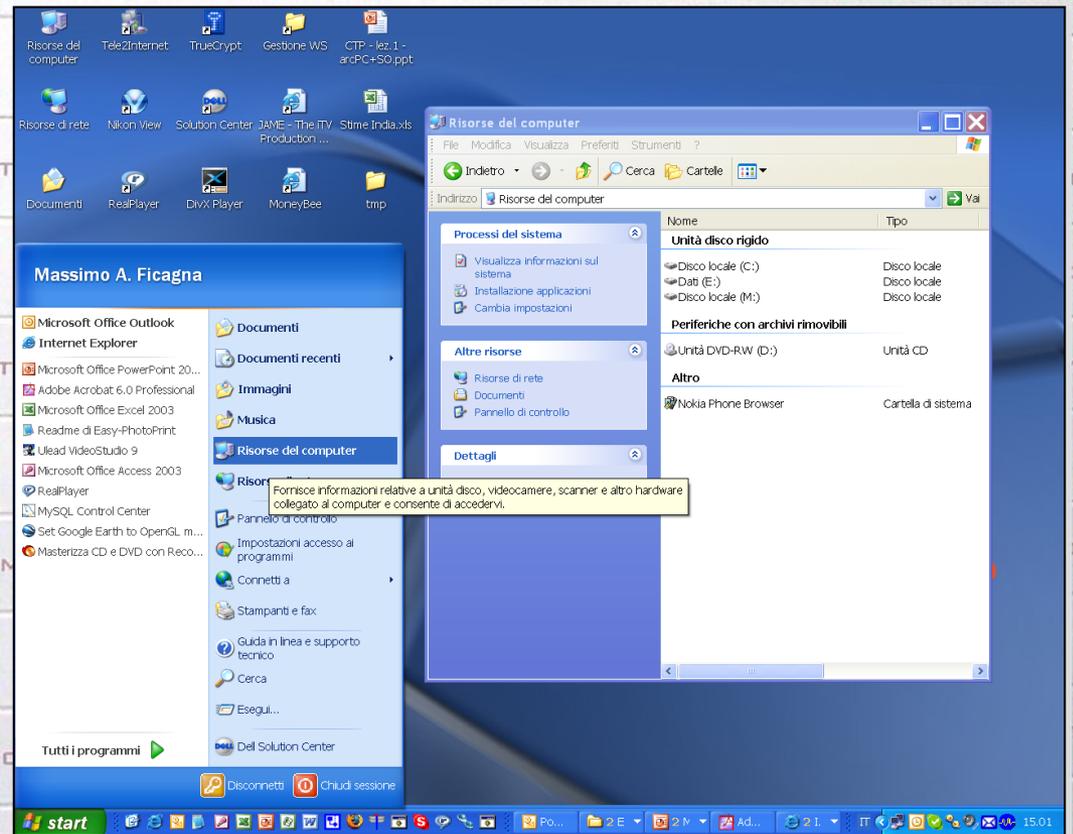
Il bootstrap non è banale

- Ogni fase può appoggiarsi ad un insieme di servizi limitato

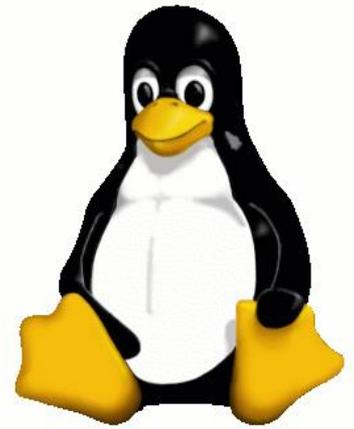


MS Windows – Elementi principali

- Interfaccia grafica
- Mouse che sposta un puntatore
- Drag & drop (trascina e lascia)
- Icone associate a file, directory, dischi, ...
- Directory come cartelle
- Pulsanti
- Finestre: cornici con strumenti
- Menu di comandi
- Cut & paste (copia e incolla)

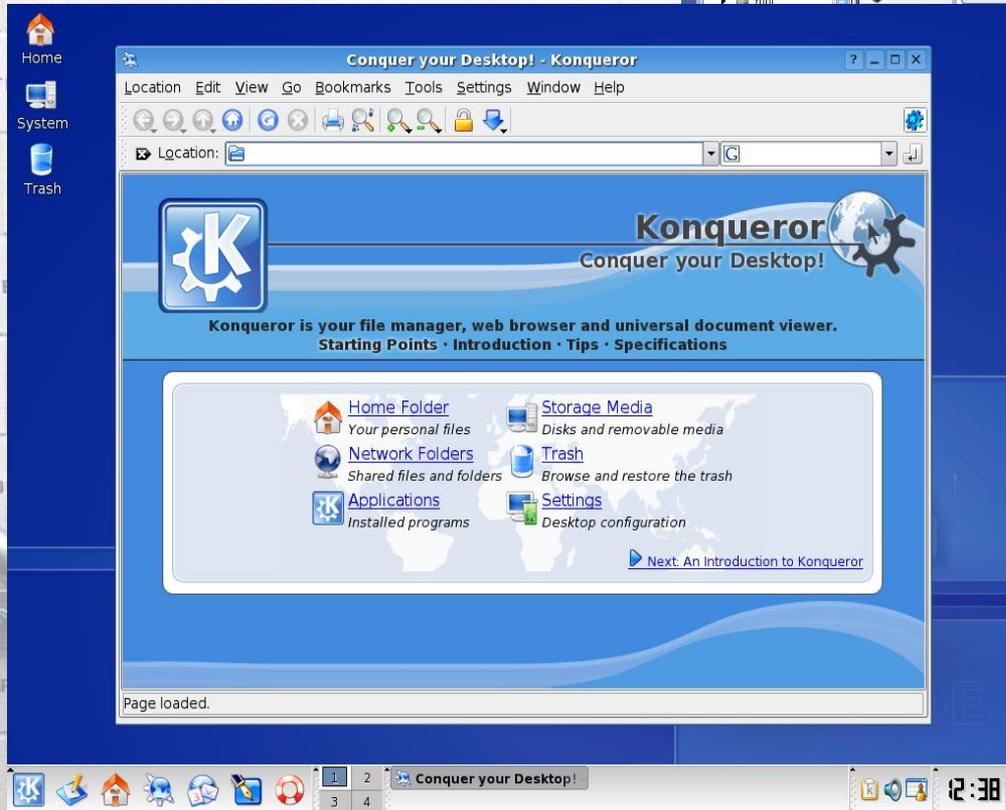
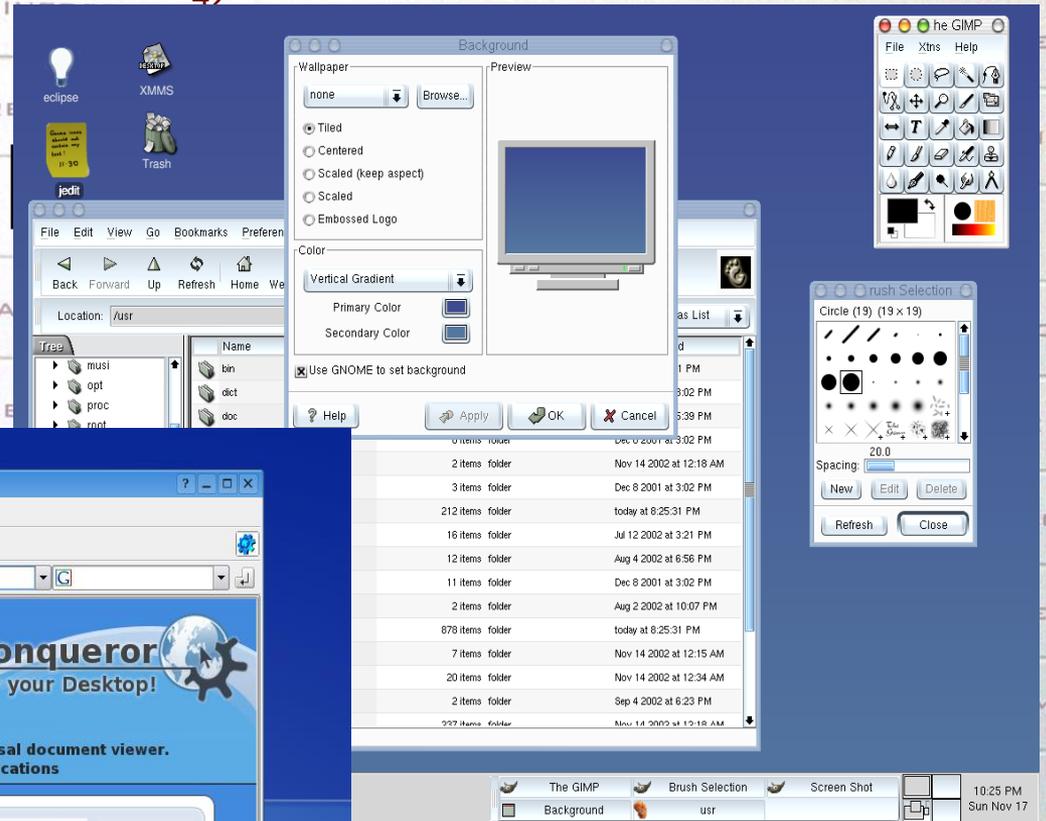


Linux



- Variante di Unix
- Open Source (= il cui codice sorgente è liberamente disponibile)
- Esistono molte **distribuzioni** di Linux, distinte per diverse combinazioni di kernel, interfacce grafiche (le più note sono Gnome e KDE) e programmi di utilità
- L'interfaccia grafica è **opzionale**
 - tant'è che molti sistemi embedded (p.es. anche molti decoder DVB-T) sono basati su Linux

Gnome →



← KDE



Mac OS X

- Enfasi sulla semplicità d'uso, grafica di grande qualità
- Hardware standardizzato
- €€€
- Basato su BSD
 - variante di Unix

