

# Sistemi Operativi Modulo I

Secondo canale (M-Z)

A.A. 2021/2022

Corso di Laurea in Informatica

## 3. Strutture dei Sistemi Operativi

Paolo Ottolino

Sapienza Università di Roma

Dipartimento di Informatica

# Operating Systems: Course Program by Layers

- A. Intro: Operating Systems «Who-What-Why-When-Where-How»
- B. A Short Look Back: «Progettazione dei Sistemi Digitali», «Architettura degli Elaboratori»
- C. EndPoint: Programmable Computer (CPU, Memory, I/O)
- D. Operating Systems - basics: cenni storici, multiprogramming, time sharing, spooling; classificazione.
- E. Tipologie di Sistemi di Elaborazione
- F. **Processes: PCB, Context Switch, Signals, Scheduler**
- G. **Memory Management: Protection, Partitioning (Segmentation, Pagination, page Table), Layers (Cache, TLB, Swapping, LRU)**
- H. **I/O: Buffering, Caching, Scheduling**
- I. **Security & Protection: Threats, Attacks, Risks. CIA: Confidentiality-Integrity-Availability -> Authentication, Authorization, Accounting**
- J. **File System**
- K. **Process Interactions: Race & Communications**
- L. **Starvation, Livelock, Deadlock**

```
graph TD; Hardware[Hardware] --> Sistema[Sistema]; Sistema --> Processi[Processi]; Processi --> Interfacce[Interfacce]; Interfacce --> Processi; Processi --> Sistema; Sistema --> Hardware;
```

Il diagramma illustra la gerarchia dei livelli di un sistema informatico, con quattro componenti principali disposte verticalmente:

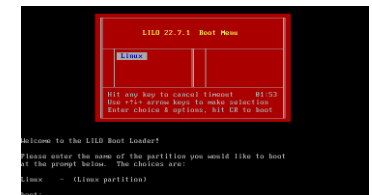
- Interfacce** (in alto, colore arancione)
- Processi** (secondo dall'alto, colore verde)
- Sistema** (terzo dall'alto, colore blu)
- Hardware** (in basso, colore rosso)

Le frecce indicano la direzione del flusso di comunicazione:

- Frecce verdi: da **Processi** verso **Interfacce** e da **Processi** verso **Sistema**.
- Frecce blu: da **Sistema** verso **Processi** e da **Sistema** verso **Hardware**.
- Frecce rosse: da **Hardware** verso **Sistema**.

The diagram illustrates a computer system architecture. A central horizontal line represents the system bus. Connected to this bus are several components: a CPU, two disk controllers (each with two disks above it), a printer controller (with a printer above it), a tape-drive controller (with three tape drives above it), a memory controller, and memory. The components are arranged in a hierarchical manner, with the bus acting as the central communication channel.

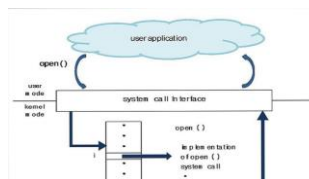
B	C
Formulas	Formula Results
=1/0	#DIV/0!
=A2/A3	#DIV/0!
=QUOTIENT(A2,A3)	#DIV/0!



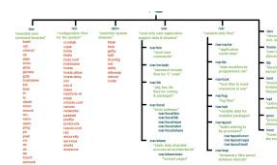
# Tipologie di Sistemi Operativi

- ## Obiettivi

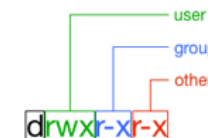
## Astrazione



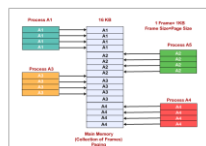
## Funzioni



## Servizi



# Virtualizzazione



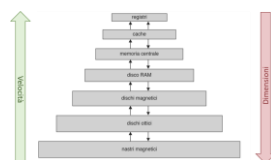
## Process Mgmt



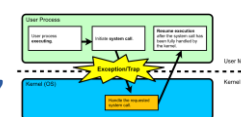
## IPC System Calls



## Input/Output



## Memory Mgmt Error Detection, Dual-Mode



## Boot

## Interrupt Handling

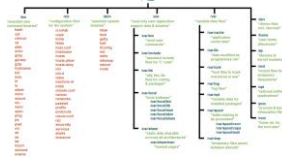
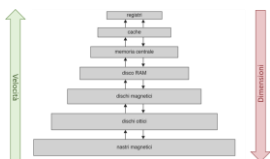
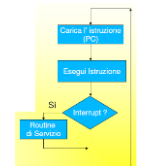


# Operating Systems: Organizzazione

## Monolitico

Generalmente:

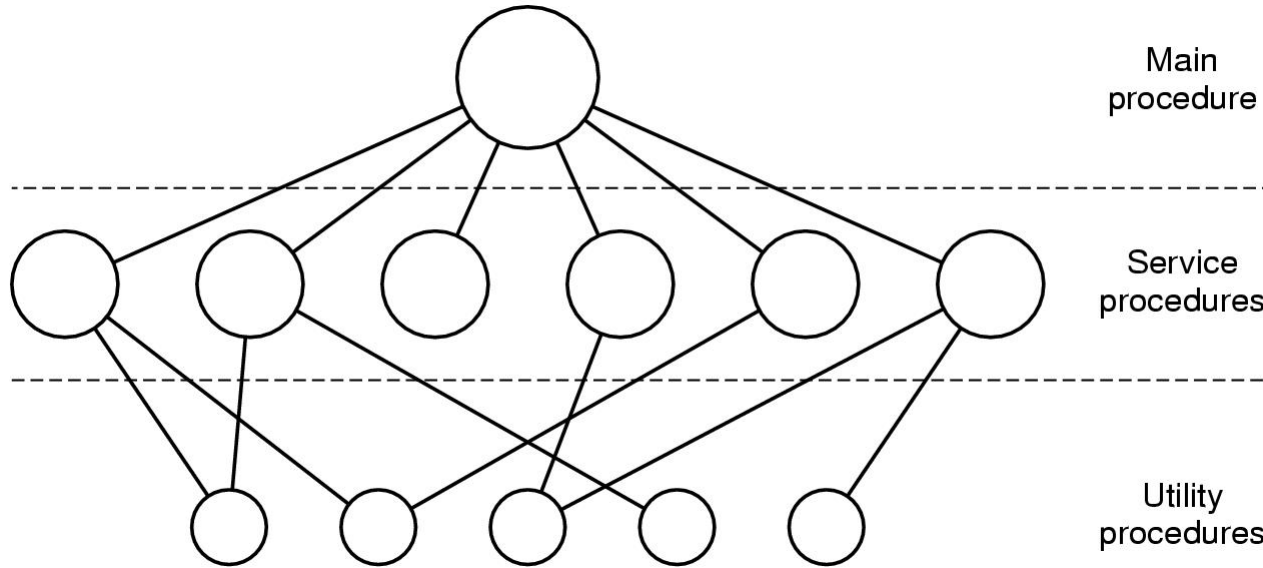
- **Interfaccia:** no Astrazione, no AA
- **Processi:** monoprocesso, no Virt, no IPS
- **Sistema:** no Dual-Mode, I/O etc incompleti

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP 	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output 	Dual-Mode	Boot Interrupt Handling 

# Operating Systems: Organizzazione

## Monolitico

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Dual-Mode	Boot Interrupt Handling



SO che non hanno una struttura ben definita:

- procedure di servizio compilate in un unico oggetto
- ogni procedura può chiamare tutte le altre
- ogni processo esegue parzialmente in modo kernel
- system call bloccanti

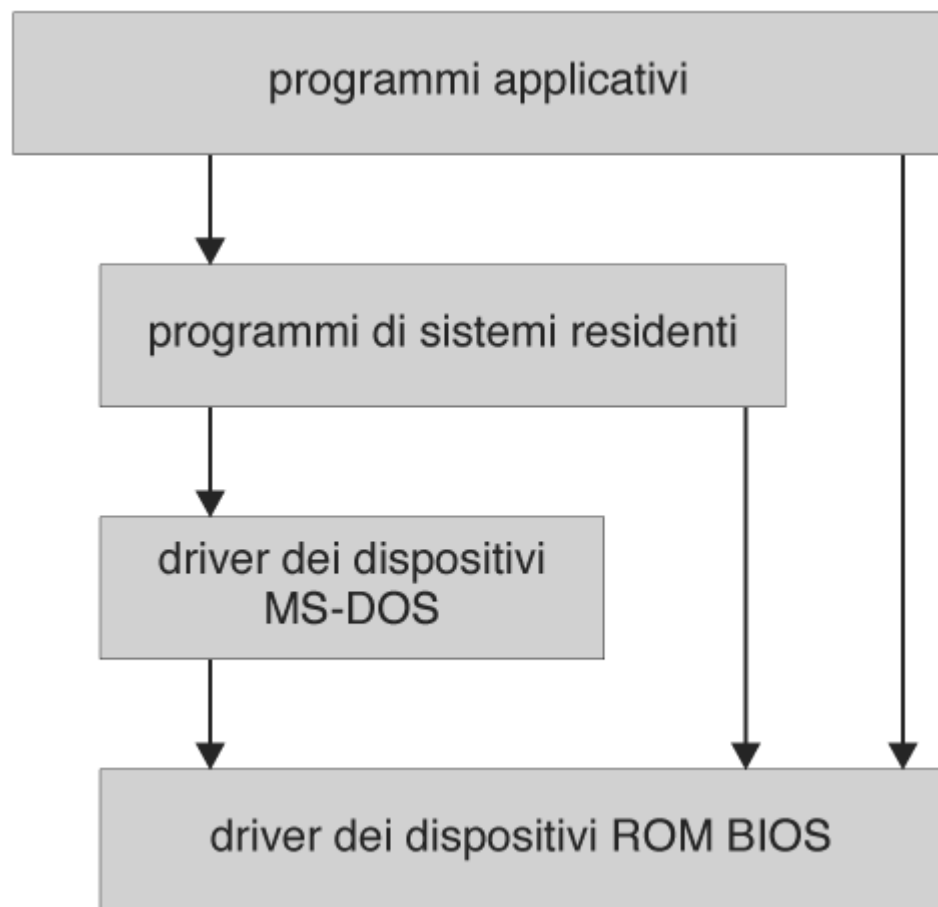
- Manutenzione difficile
- Vulnerabilità agli errori e agli attacchi
- Blocco del sistema
- Insufficiente protezione fornita dal SO

# Operating Systems: Organizzazione

## Monolitico

nati come sistemi piccoli, semplici e limitati, per poi crescere al di là dello scopo originario (es. MS-DOS, UNIX originario)

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Dual-Mode	Boot Interrupt Handling



Es. MS-DOS:

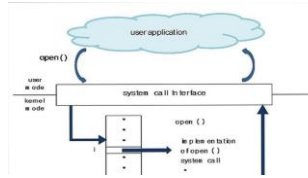
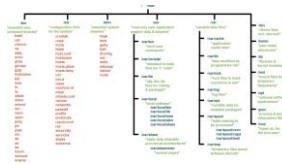
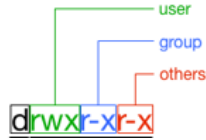
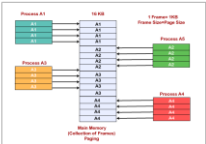


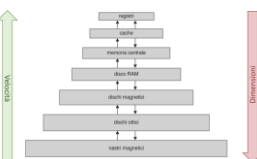
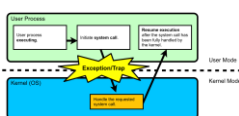

- Dipendente dall'HW per cui sono stati pensati (Intel 8086, 8088: senza bit di stato)
- libertà ai programmi applicativi di accedere direttamente all'hardware
- No ASLR (Address Space Layout Randomization): spostamento casuale della posizione di moduli e strutture in memoria ad ogni esecuzione
- No DEP (Data Execution Prevention) impedisce determinati settori di memoria, ad es. lo stack, dall'essere eseguito

# Operating Systems: Organizzazione

## Stratificato

Generalmente:

- **Interfaccia:** programmi di sistema
- **Processi:** gestione memoria, processi, comunicazione
- **Sistema:** Dual-Mode, I/O etc completi

Obiettivi	Funzioni	Servizi
<b>Astrazione</b> 	<b>File System</b> <b>Sh/GUI/OLTP</b> 	<b>Authentication/Authorization/Accounting</b> 
<b>Virtualizzazione</b> 	<b>Process Mgmt</b> 	<b>IPC</b> <b>System Calls</b> 
<b>Input/Output</b> 	<b>Memory Mgmt</b> <b>Error Detection,</b> <b>Dual-Mode</b> 	<b>Boot</b> <b>Interrupt Handling</b> 



# Operating Systems: Organizzazione

## Stratificato: UNIX

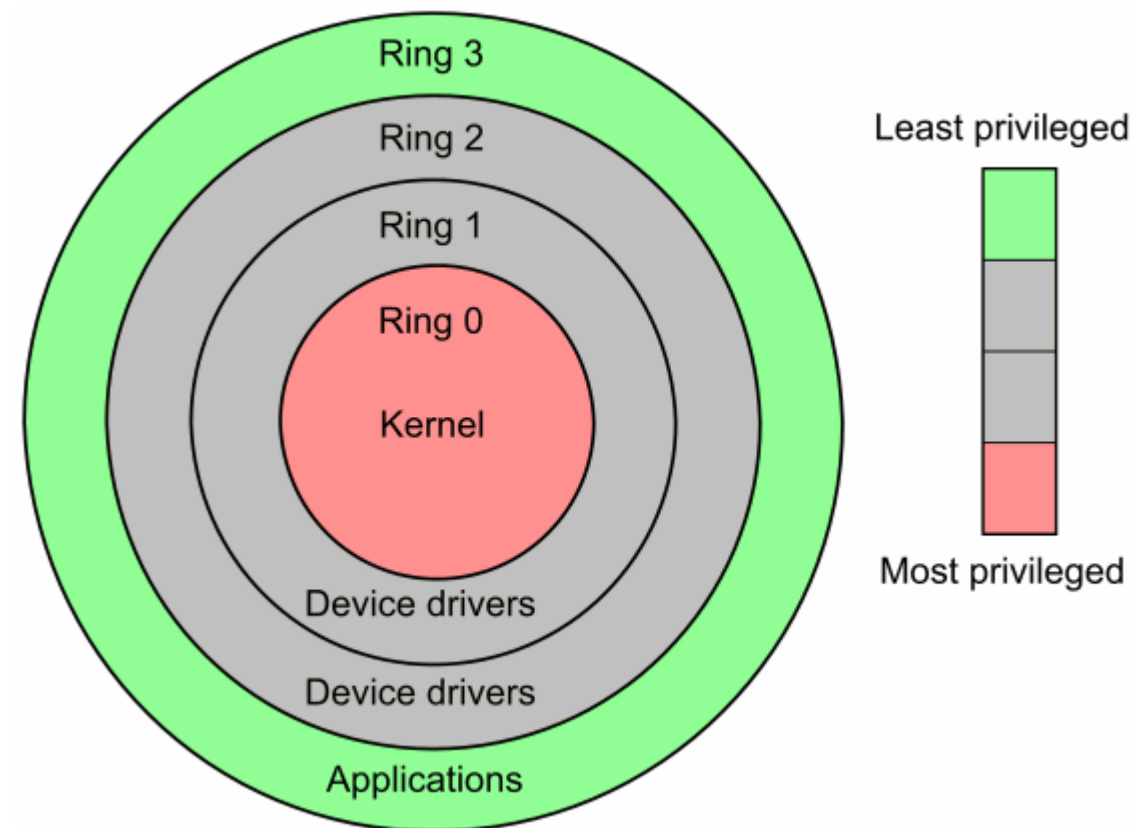
Obiettivi	Funzioni	Servizi
Autoreazione	File System SVGA/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection Dual-Mode	Boot Interrupt Handling

UNIX consiste di due parti separabili:

1. **Programmi di Sistema** (es. **libc**, **init**, **inet**, **daemon**, etc)
2. Nucleo (kernel): qualsiasi parte si trovi sotto l'interfaccia di chiamata di sistema e sopra l'hardware fisico
  - Un'enorme quantità di funzioni combinate in un solo livello:
  - gestione del file-system, schedulazione della CPU, gestione della memoria, ecc..

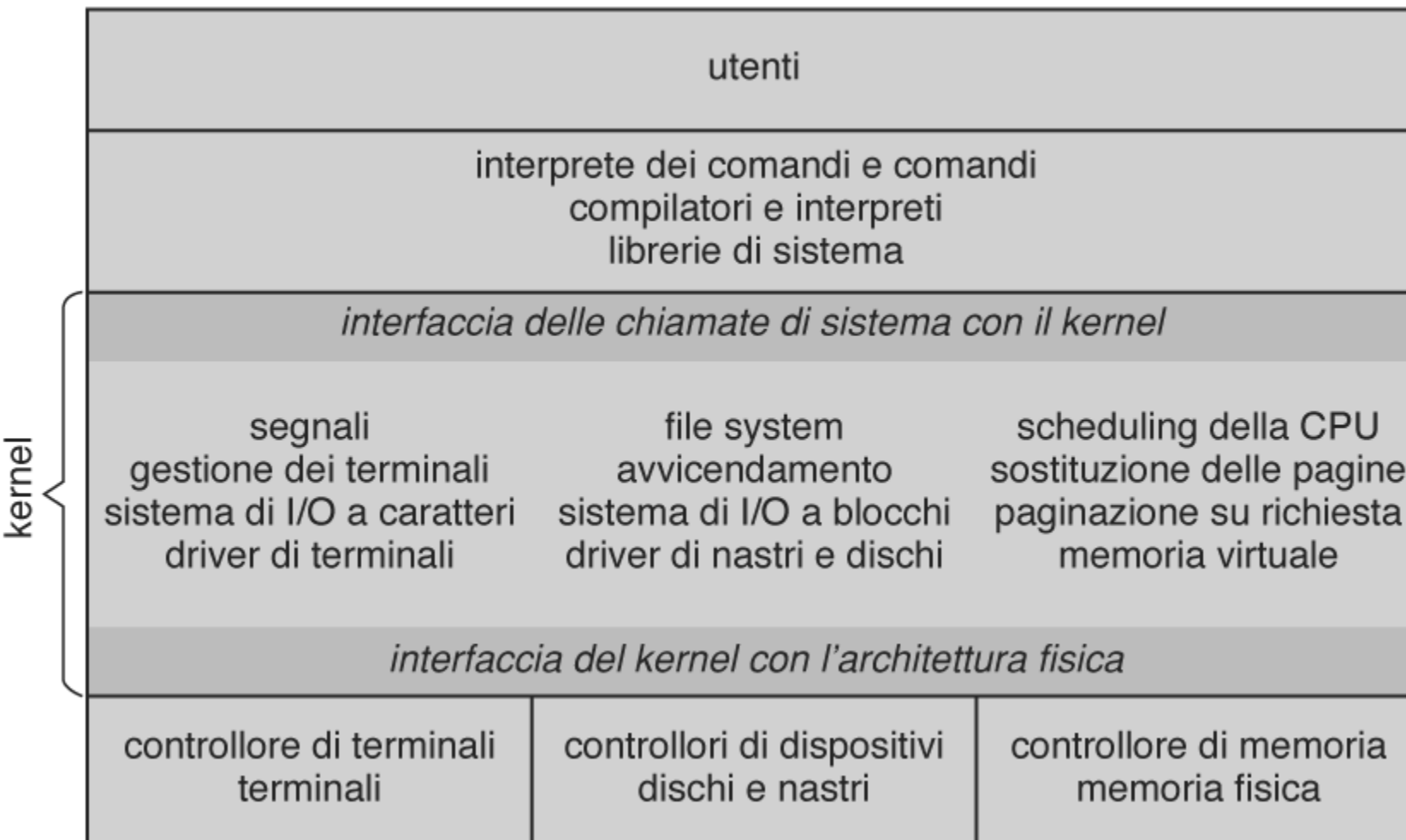
Separazione gerarchica o a livelli delle funzioni

- Il SO è diviso in un certo numero di strati (livelli, o layers), ognuno costruito sulla sommità dello strato inferiore
- Lo strato inferiore (il livello 0) è l'hardware; quello più elevato (il livello N) è l'interfaccia utente
- L'interfaccia degli strati è stabile
- L'implementazione può variare



# Operating Systems: Organizzazione

Stratificato: UNIX



Vantaggi:

- Ogni strato offre una virtualizzazione di un certo numero di funzioni (macchina virtuale)
- La dipendenza dall'hardware è limitata al livello più basso (portabilità)

Svantaggi:

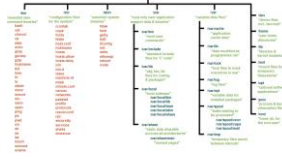
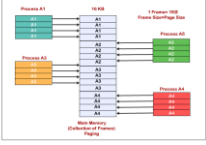

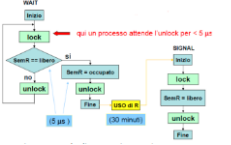
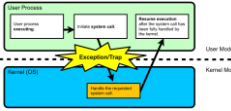
- La portabilità si paga con minor efficienza perché una chiamata di sistema deve attraversare più strati (eventuale adattamento dei dati passati)
- Più difficile da progettare

# Operating Systems: Organizzazione

## MicroKernel

Generalmente:

- **Interfaccia:**  
programmi di sistema (no kernel)
- **Processi:**  
gestione memoria, processi, comunicazione
- **Sistema:** Dual-Mode

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP 	Authentication/ Authorization/ Accounting
Virtualizzazione 	Process Mgmt 	IPC System Calls 
Input/Output	Memory Mgmt Error Detection, Dual-Mode 	Boot Interrupt Handling

# Operating Systems: Organizzazione

## MicroKernel

Sposta il maggior numero di funzionalità possibili dal kernel allo spazio utente (come programmi di sistema)

- Il microkernel offre solo servizi per gestire processi, memoria e comunicazione.
- La comunicazione tra moduli avviene tramite scambio di messaggi

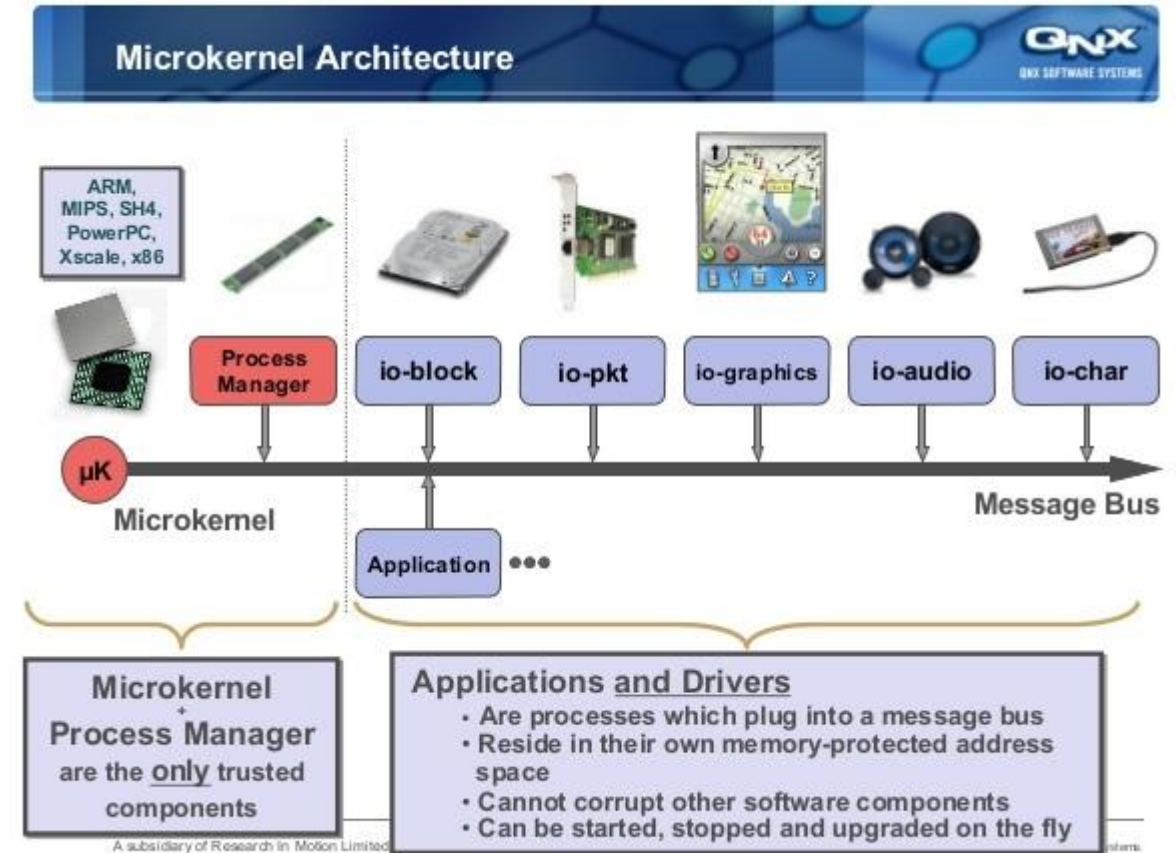
Vantaggi:

- Ottimale per i sistemi Real Time (RTOS)
- Facilità di estendere il SO (lasciando invariato il microkernel)
- Maggiore portabilità del SO da un'architettura HW a un'altra
- Maggiore affidabilità e sicurezza (meno codice viene eseguito in modalità kernel)

Svantaggi:

- I microkernel possono soffrire di calo di prestazioni per l'aumento di sovraccarico di funzioni di sistema

Obiettivi	Funzioni	Servizi
Abstrazione	File System SV/Gui/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection Dual-Mode	Host Interrupt Handling

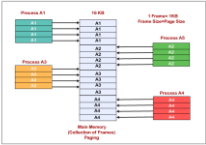

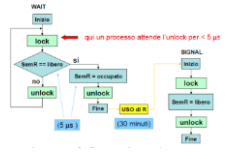
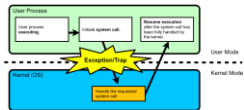
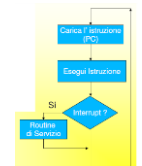


# Operating Systems: Organizzazione

Modulare (Kernel a Moduli Funzionali)

Generalmente:

- **Interfaccia:**  
Funzioni e Servizi operati da moduli
- **Processi:**  
gestione interna al kernel
- **Sistema:**  
operati da moduli appositi

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione 	Process Mgmt 	IPC System Calls 
Input/Output	Memory Mgmt Error Detection, Dual-Mode 	Boot Interrupt Handling 

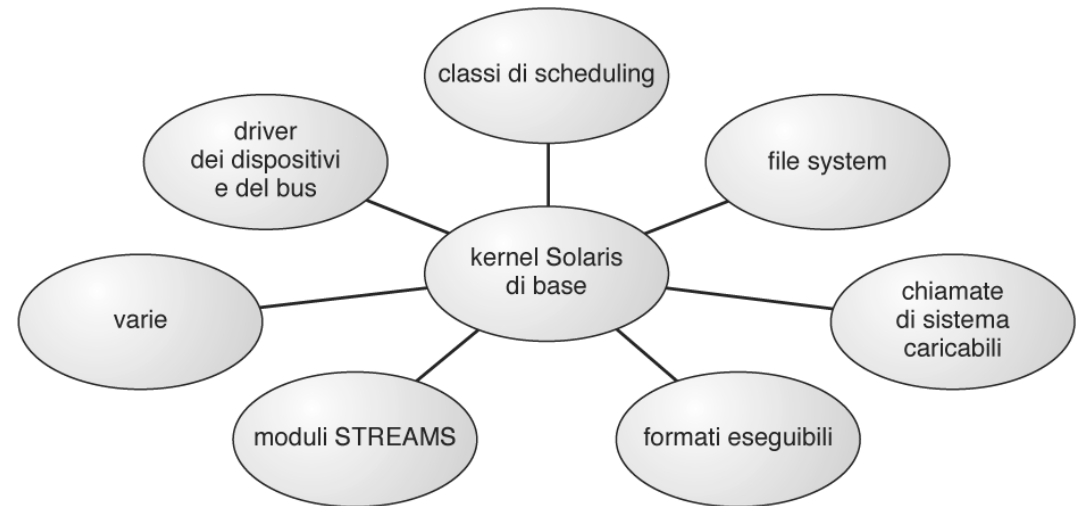
# Operating Systems: Organizzazione

## Modulare (Kernel a Moduli Funzionali)

Moderna metodologia di progetto che realizza kernel modulari

- attraverso tecniche object-oriented
- ogni componente core è separata e interagisce con le altre attraverso interfacce ben note
- A differenza della struttura a stati i moduli possono comunicare con tutti gli altri, non solo con quelli di livello inferiore
- ogni componente è caricabile dinamicamente nel kernel al momento del boot e/o durante l'esecuzione

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection Dual-Mode	Boot Interrupt Handling



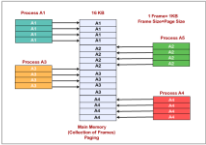


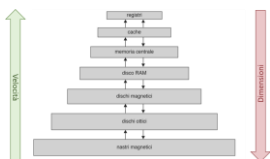
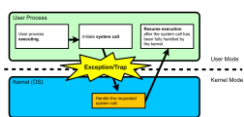
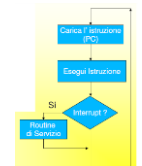
Stratificato: maggiormente flessibile  
MicroKernel: più efficiente (no messaggi)

# Operating Systems: Organizzazione

## Macchine Virtuali

Generalmente:

- **Interfaccia:**  
minimale
- **Processi:**  
gestione  
interna al  
kernel
- **Sistema:**  
operati da  
moduli  
appositi

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione 	Process Mgmt 	IPC System Calls 
Input/Output 	Memory Mgmt Error Detection, Dual-Mode 	Boot Interrupt Handling 

# Operating Systems: Organizzazione

## Macchine Virtuali

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection, Dual-Mode	Boot Interrupt Handling

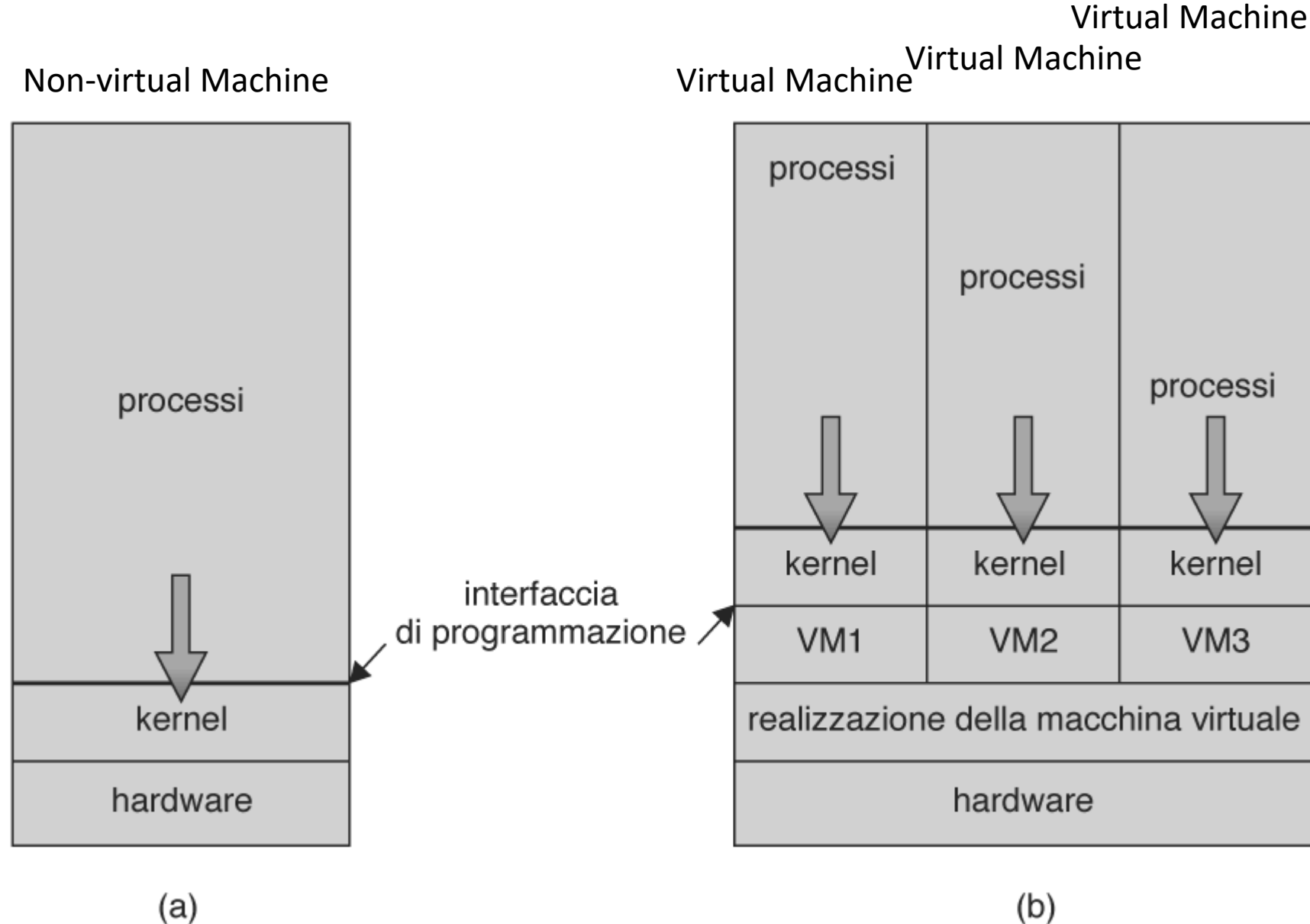
L'approccio a strati trova il suo logico sbocco nel concetto di macchina virtuale

- Costruzione gerarchica di macchine astratte (o virtuali) in esecuzione su uno stesso hardware
- Una macchina virtuale fornisce un'interfaccia software che è identica al puro hardware sottostante
- Il SO crea l'illusione che un processo ospite abbia un proprio processore ed una propria memoria (virtuale)
- Il processo ospite è tipicamente un SO



# Operating Systems: Organizzazione

## Macchine Virtuali



Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection, Dual-Mode	Boot Interrupt Handling

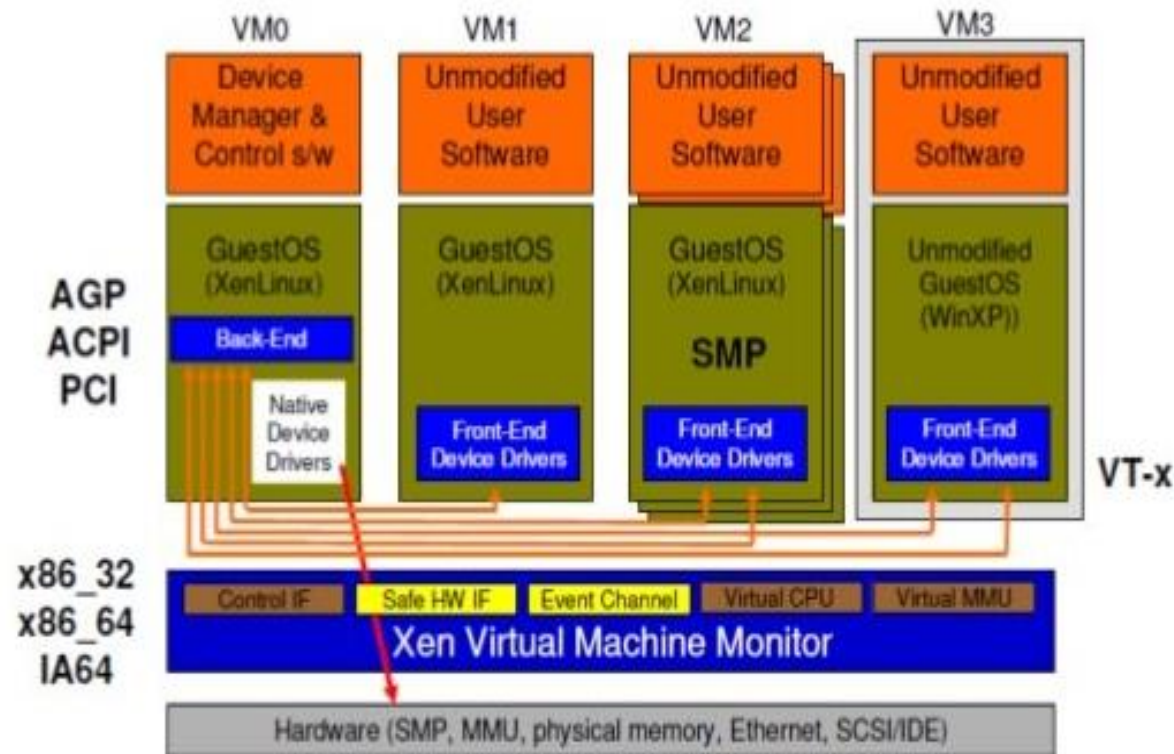
Il computer fisico mette in condivisione le proprie risorse per generare macchine virtuali

- La schedulazione della CPU e la memoria virtuale può dare l'impressione che ogni processo ospite (solitamente un SO) abbia un proprio hardware dedicato – virtual HW
- Lo spooling e il file system possono fornire lettori di schede e stampanti in linea virtuali, e altri dispositivi – virtual devices, virtual memory

# Operating Systems: Organizzazione

Macchine Virtuali: Vmware, Xen, Hyper-V, KVM, etc

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection, Dual-Mode	Boot Interrupt Handling



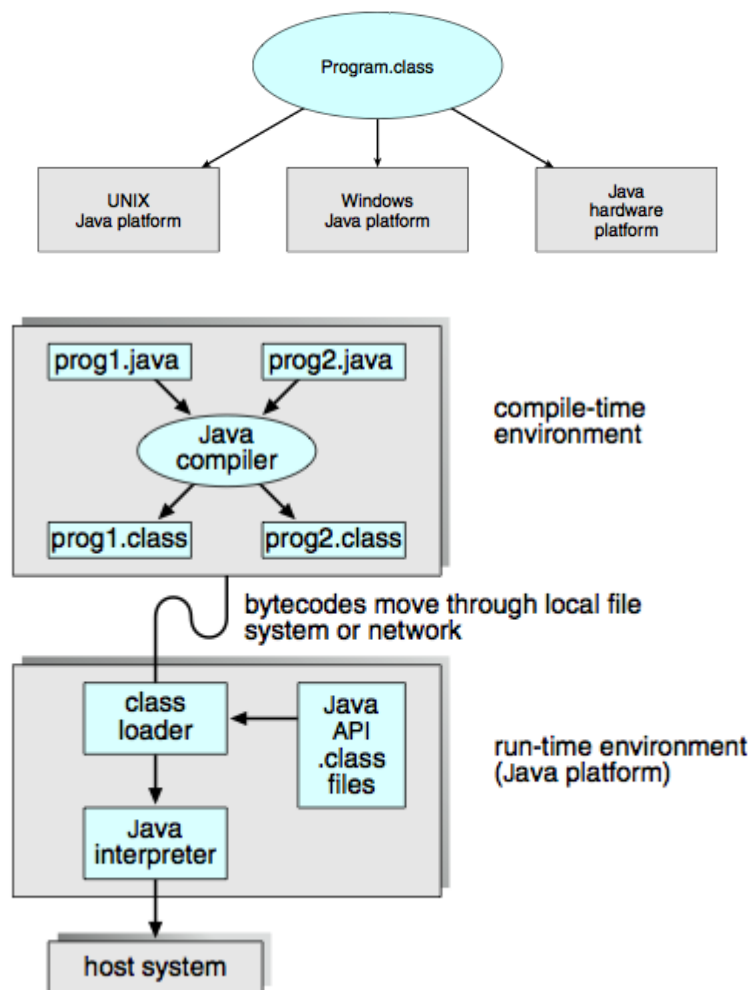
Lo strato di virtualizzazione è scritto per funzionare in modalità utente

- Trasforma HW in macchine virtuali che ospitano SO
- Il disco virtuale è in realtà un file (Si possono così creare snapshot)

# Operating Systems: Organizzazione

## Macchine Virtuali: Java Virtual Machine

Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection, Dual-Mode	Boot Interrupt Handling



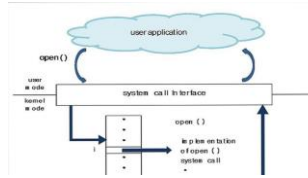
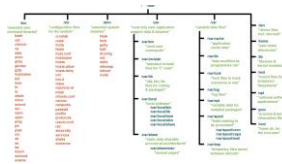
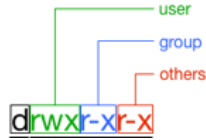
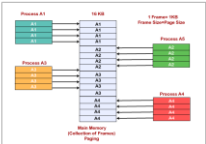


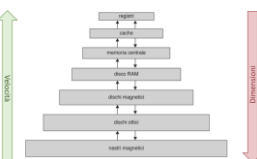
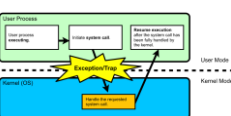

- I programmi Java compilati sono bytecode indipendenti dall'architettura ed eseguiti da una Java Virtual Machine (JVM)
- La JVM è un calcolatore astratto su un sistema ospitante vero
- La JVM utilizza un interprete puro, senza influenzare troppo negativamente le prestazioni, in quanto utilizza un Just In Time compiler (JIT)
- porzioni di codice eseguite numerose volte vengono compilate per aumentare le prestazioni
- Diversi HotSpot (client e server). La JVM ufficiale è quella mantenuta da Oracle

# Operating Systems: Organizzazione

## Client/Server (Distribuito)

Generalmente:

- **Interfaccia:** transattiva tra processi
- **Processi:** client e server
- **Sistema:** operati da moduli appositi

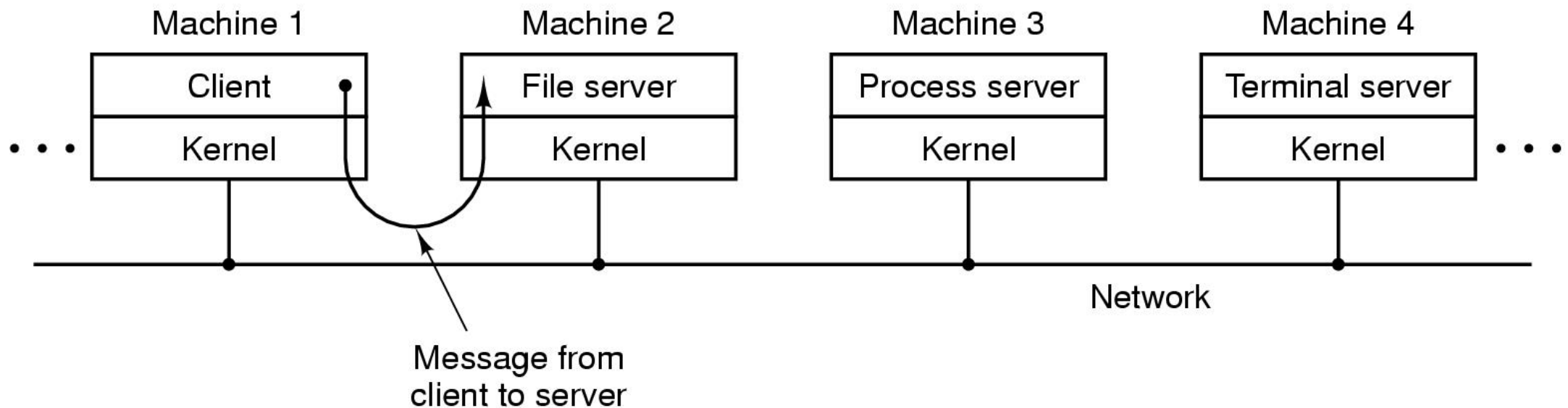
Obiettivi	Funzioni	Servizi
<b>Astrazione</b> 	<b>File System</b> <b>Sh/GUI/OLTP</b> 	<b>Authentication/ Authorization/ Accounting</b> 
<b>Virtualizzazione</b> 	<b>Process Mgmt</b> 	<b>IPC</b> <b>System Calls</b> 
<b>Input/Output</b> 	<b>Memory Mgmt</b> <b>Error Detection,</b> <b>Dual-Mode</b> 	<b>Boot</b> <b>Interrupt Handling</b> 

# Operating Systems: Organizzazione

## Client/Server (Distribuito)










Obiettivi	Funzioni	Servizi
Astrazione	File System Sh/GUI/OLTP	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection, Dual-Mode	Boot Interrupt Handling

- **Interfaccia:** Più calcolatori tra loro collegati, ciascuno con una propria copia del kernel ed un certo numero di processi client e/o server
- **Processi:** I processi client richiedono servizi inviando le richieste tramite il kernel

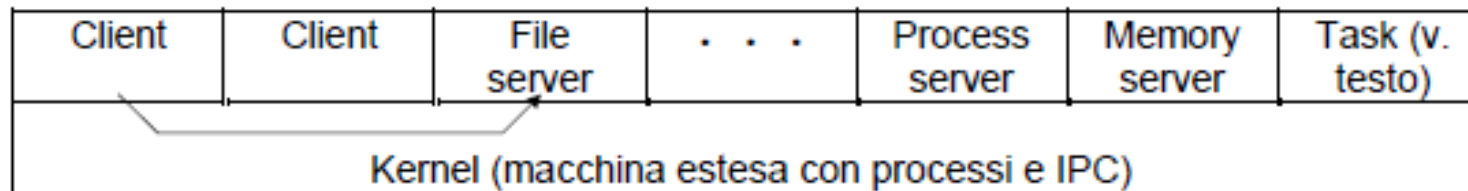


# Operating Systems: Organizzazione

## Client/Server (Distribuito)

Obiettivi	Funzioni	Servizi
Astrazione 	File System Sh/GUI/OLTP 	Authentication/ Authorization/ Accounting 
Virtualizzazione 	Process Mgmt 	IPC System Calls 
Input/Output 	Memory Mgmt Error Detection, Dual-Mode 	Boot Interrupt Handling 

- I processi usano i servizi messi a disposizione dal kernel per comunicare
- Un processo utente (processo client) richiede un servizio (ad es. lettura di un file) ad un processo di SO (processo server)
- Client e server operano in modalità utente
- I processi server realizzano le politiche di gestione delle risorse
- Il kernel include solo i dati che descrivono un processo e realizza i
- meccanismi di comunicazione tra processi



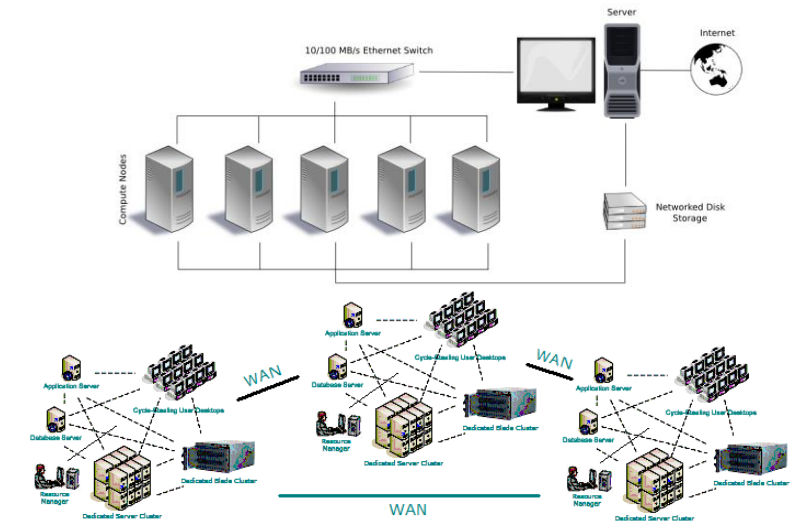
# Operating Systems: Organizzazione

## Client/Server (Distribuito): es. Clusters

In generale, un cluster si riferisce a un insieme di computer collegati tra loro.

- **High Performance Computer (HPC):** fisicamente situati uno vicino all'altro, al fine di risolvere i problemi in modo più efficiente. Generalmente, eseguono la stessa immagine di Sistema Operativo.
- **Grid Computing:** uso di una griglia computazionale (stazioni di lavoro, server blade, ecc.) applicando le risorse della griglia, tramite rete, a un singolo problema allo stesso tempo, mentre si superano i confini politici e teorici.
- **High Availability (HA):** un sistema informatico funge da sistema di backup per uno o più sistemi primari, tutti situati uno vicino all'altro. Quando c'è un guasto in un sistema primario, le applicazioni critiche in esecuzione su quel sistema vengono trasferite al sistema di backup designato.
  - **Load Balancing (LB):** come HA ma tutti i sistemi funzionanti ed aderenti al cluster dividono il carico di lavoro (Attivo/Attivo).
  - **Cluster Geografico:** come HA ma i sistemi sono situati a distanza.
- **Three Tier:** architetture si sistema a 3 livelli: presentazione (web Server), elaborazione (App), dati (DB). Ognuno su server diversi, in HA/LB e SO distinti

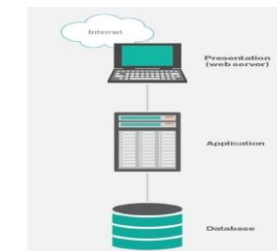
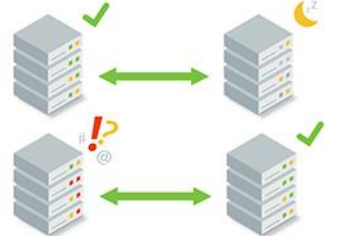
Obiettivi	Funzioni	Servizi
Astrazione	File System SMB/CIFS	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection Disk Abstraction	Host Interrupt Handling



Active / Active Design



Active / Passive

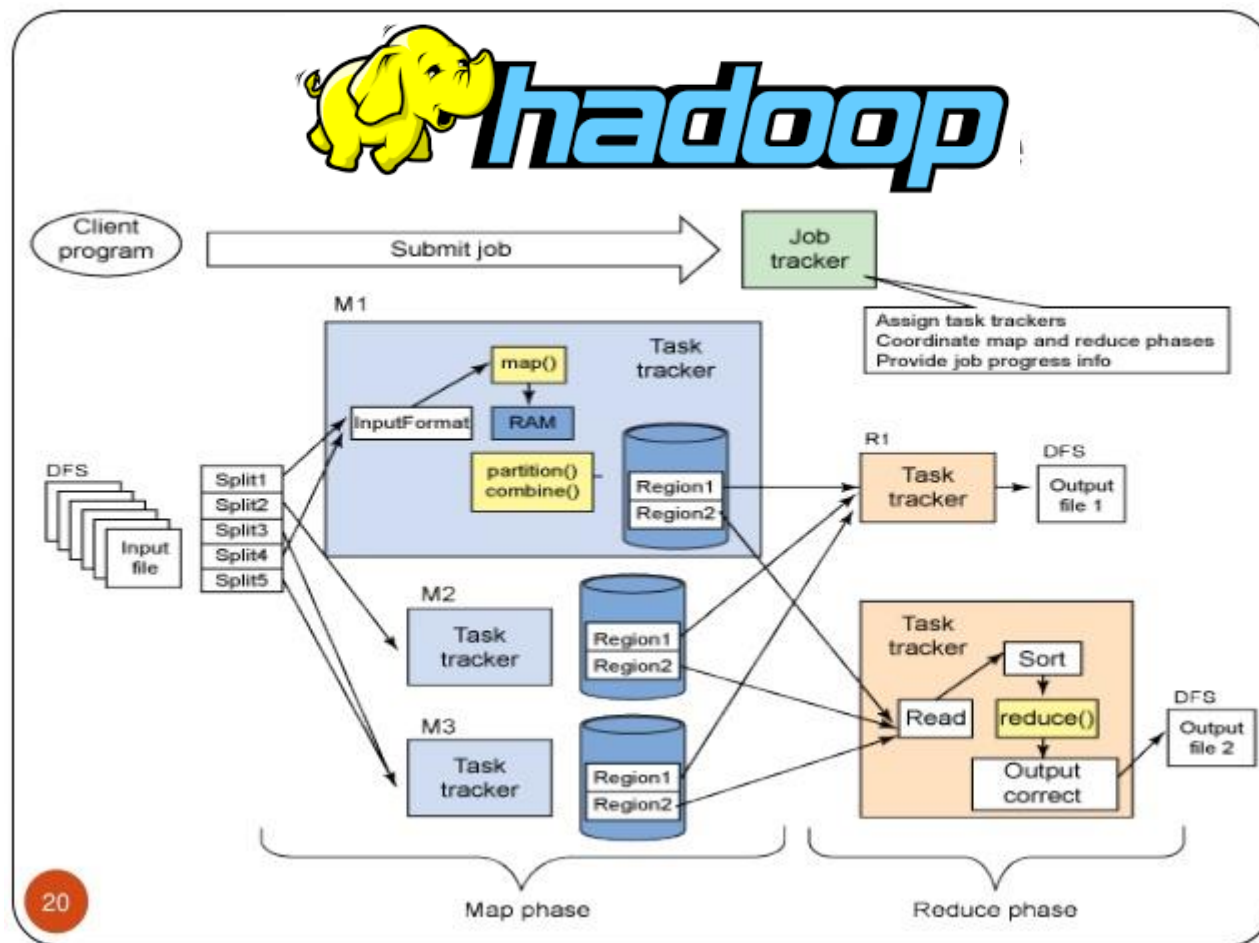




# Operating Systems: Organizzazione

Client/Server (Distribuito): es. Hadoop

Obiettivi	Funzioni	Servizi
Astrazione	File System Shell/CLI/OTF	Authentication/ Authorization/ Accounting
Virtualizzazione	Process Mgmt	IPC System Calls
Input/Output	Memory Mgmt Error Detection Dual Mode	Root Interrupt Handling



- **Interfaccia:** HDFS (Hadoop Distributed File System)
- **Processi:** Map/Reduce (capacità di dividere ed elaborare terabyte di dati in parallelo, ottenendo risultati più rapidi.)
- **Sistema:** si basa sui SO sottostanti