

# Sistemi Operativi: Concetti Fondamentali

## Kernel, Shell e Distribuzioni Linux

Prof. Massimo Fedeli

ITS Academy - Fabbrica Digitale

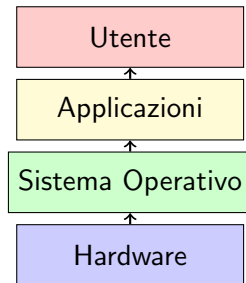
23 dicembre 2025

- 1 Introduzione ai Sistemi Operativi
- 2 Il Kernel
- 3 La Shell
- 4 GNU/Linux
- 5 Distribuzioni Linux

# Cos'è un Sistema Operativo?

**Definizione:** Un sistema operativo (SO) è un software di base che:

- Gestisce le risorse hardware del computer
- Fornisce servizi ai programmi applicativi
- Fa da intermediario tra utente e hardware
- Coordina l'esecuzione dei programmi



## ① Gestione dei processi

- Creazione, scheduling, terminazione
- Sincronizzazione e comunicazione tra processi

## ② Gestione della memoria

- Allocazione e deallocazione
- Memoria virtuale

## ③ Gestione del file system

- Organizzazione dei file
- Controllo degli accessi

## ④ Gestione I/O

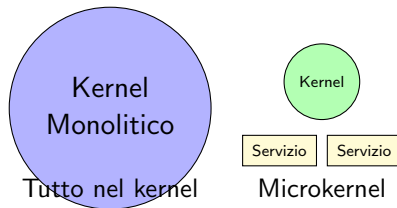
- Driver dei dispositivi
- Buffering e caching

## Per struttura:

- Monolitici
- A microkernel
- Ibridi
- A strati

## Per utilizzo:

- Desktop (Windows, macOS, Linux)
- Server
- Mobile (Android, iOS)
- Embedded
- Real-time



# Il Kernel: Cuore del Sistema Operativo

## Definizione

Il **kernel** è il nucleo del sistema operativo, il componente che ha controllo completo su tutto il sistema.

## Responsabilità principali:

- Gestione della CPU e scheduling dei processi
- Gestione della memoria RAM
- Gestione dei dispositivi hardware (driver)
- Gestione delle chiamate di sistema (system calls)
- Gestione della sicurezza e dei permessi

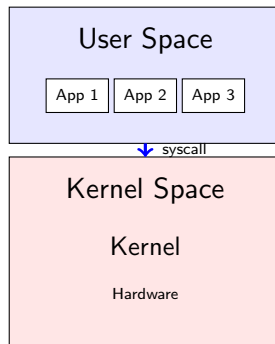
# Modalità di Esecuzione: User Mode vs Kernel Mode

## User Mode (Modalità utente):

- Accesso limitato alle risorse
- Esecuzione applicazioni utente
- Protezione del sistema
- Necessita system calls per operazioni privilegiate

## Kernel Mode (Modalità kernel):

- Accesso completo all'hardware
- Esecuzione codice kernel
- Nessuna restrizione
- Operazioni critiche del sistema



## 1. Kernel Monolitico

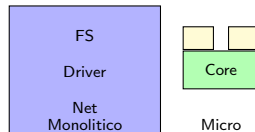
- Tutto il codice in un unico blocco
- Alte prestazioni
- Più difficile da mantenere
- Esempio: Linux, Unix tradizionale

## 2. Microkernel

- Kernel minimale
- Servizi in user space
- Maggiore stabilità
- Esempio: Minix, QNX

## 3. Kernel Ibrido

- Compromesso tra i due
- Alcuni servizi nel kernel
- Altri in user space
- Esempio: Windows NT, macOS





# Il Kernel Linux

## Caratteristiche:

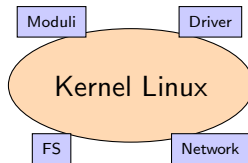
- Architettura monolitica modulare
- Open Source (GPL v2)
- Supporto multiplatforma
- Altamente configurabile
- Sviluppo collaborativo

## Componenti principali:

- Process scheduler
- Memory manager
- Virtual File System (VFS)
- Network stack
- Device drivers

## Storia

- Creato da Linus Torvalds nel 1991
- Versione 0.01: 10.000 righe
- Oggi: oltre 27 milioni di righe
- Migliaia di contribuitori



# Versioning del Kernel Linux

## Schema di numerazione:

X.Y.Z

- **X** = Versione principale (major)
- **Y** = Versione minore (minor)
- **Z** = Revisione (patch level)

## Esempi:

- 5.15.0 → Versione 5, release 15, nessuna patch
- 6.1.38 → Versione 6, release 1, patch 38

## Tipi di rilascio:

- **Mainline:** versione in sviluppo attivo
- **Stable:** versioni stabili per produzione
- **Longterm (LTS):** supporto esteso (2-6 anni)

## Comando per visualizzare la versione:

- `uname -r`

# La Shell: Interfaccia a Riga di Comando

## Definizione

La **shell** è un interprete di comandi che fornisce un'interfaccia tra l'utente e il kernel del sistema operativo.

## Funzioni principali:

- Interpreta ed esegue comandi
- Gestisce input/output
- Supporta scripting e automazione
- Gestisce variabili d'ambiente
- Controllo dei processi
- Espansione di caratteri speciali (wildcards)

# Tipi di Shell in Linux

## Shell principali:

### ① sh (Bourne Shell)

- Shell originale Unix
- Standard POSIX

### ② bash (Bourne Again Shell)

- Shell predefinita in molte distro
- Compatibile con sh
- Funzionalità avanzate

### ③ zsh (Z Shell)

## Altre shell:

- csh (C Shell)
- tcsh (TENEX C Shell)
- ksh (Korn Shell)
- dash (Debian Almquist Shell)

## Controllare la shell corrente:



## ① Prompt dei comandi

- Indica che la shell è pronta
- Personalizzabile (PS1)
- Mostra info utili (utente, path, ecc.)

## ② Interprete di comandi

- Parsing della linea di comando
- Esecuzione programmi

## ③ Espansione e sostituzione

- Wildcards: \*, ?, [ ]
- Variabili: \$VAR
- Command substitution: \$(comando)

## ④ Redirezione I/O

- Input: j
- Output: >, >>
- Pipe: —

# Comandi Shell Fondamentali

## Navigazione file system:

```
pwd # directory  
corrente  
ls # elenco file  
cd /path # cambia  
directory  
mkdir dir # crea  
directory  
rmdir dir # rimuove  
directory
```

## Gestione file:

```
cp src dst # copia  
mv a b # sposta/  
rinomina  
rm file # rimuove
```

## Informazioni di sistema:

```
uname -a # info  
sistema  
whoami # utente  
corrente  
date # data e ora  
df -h # spazio  
disco  
free -h # memoria
```

## Gestione processi:

```
ps # processi  
attivi  
top # monitor  
processi  
kill PID # termina
```

Uno script shell è un file di testo con una sequenza di comandi.

```
#!/bin/bash
# Primo script di esempio

# Variabili
NOME="Mario"
echo "Ciao_ $NOME!"

# Controllo condizionale
if [ -f /etc/passwd ]; then
echo "Il_file_esiste"
fi

# Ciclo
for i in {1..5}; do
echo "Iterazione_$i"
done
```



# Variabili d'Ambiente

**Variabili che influenzano il comportamento del sistema e dei programmi.**

## Variabili comuni:

- PATH – percorsi eseguibili
- HOME – directory home
- USER – nome utente
- SHELL – shell corrente
- LANG – lingua sistema
- PWD – directory corrente

## Gestione variabili:

```
# Visualizzare
echo $PATH
env | grep PATH

# Impostare (temporanea)
export MYVAR="valore"

# Impostare (permanente)
# aggiungere in ~/.bashrc
export EDITOR=vim

# Rimuovere
unset MYVAR
```

## Progetto GNU (1983)

- Fondato da Richard Stallman
- Sistema operativo libero
- Filosofia Free Software
- GNU's Not Unix
- Strumenti: gcc, glibc, bash, ecc.

## Linux Kernel (1991)

- Creato da Linus Torvalds
- Inizialmente hobby personale
- Rilasciato con licenza GPL
- Completa il sistema GNU

## Timeline:

- 1983 - Progetto GNU inizia
- 1991 - Linux 0.01 rilasciato
- 1992 - Licenza GPL per Linux
- 1993 - Slackware (prima distro)
- 1993 - Debian fondata
- 1994 - Red Hat fondata
- 2004 - Ubuntu rilasciata
- 2011 - Linux 3.0
- 2022 - Linux 6.0

## GNU/Linux

Sistema operativo completo = Kernel Linux +  
Strumenti GNU + Altro software

## Le Quattro Libertà Fondamentali:

0. **Libertà di eseguire** il programma per qualsiasi scopo
1. **Libertà di studiare** come funziona il programma e modificarlo
2. **Libertà di redistribuire** copie del programma
3. **Libertà di migliorare** il programma e distribuire i miglioramenti

## Open Source vs Free Software

- **Free Software:** enfasi sulle libertà etiche
- **Open Source:** enfasi sui vantaggi pratici
- Spesso sovrapposti ma filosofie diverse

# Principali Licenze Open Source

## Licenze Copyleft:

- **GPL (GNU General Public License)**
  - Versioni: GPLv2, GPLv3
  - Derivati devono essere GPL
  - Usata da Linux
- **LGPL (Lesser GPL)**
  - Permette linking con software proprietario
  - Usata per librerie
- **AGPL**
  - Come GPL ma per software web

## Licenze Permissive:

- **MIT License**
  - Molto permissiva
  - Breve e semplice
- **BSD License**
  - 2-clause, 3-clause
  - Simile a MIT
- **Apache License 2.0**
  - Include protezione brevetti
  - Usata da Apache, Android

**Differenza chiave:** Copyleft richiede che i derivati mantengano la stessa licenza; permissive permettono uso anche in software proprietario.

## Definizione

Una **distribuzione Linux** (distro) è un sistema operativo completo basato sul kernel Linux, che include:

- Kernel Linux
- Strumenti GNU e utilità di sistema
- Package manager
- Desktop environment (opzionale)
- Applicazioni preinstallate
- Strumenti di configurazione
- Sistema di init (systemd, OpenRC, ecc.)

## Perché esistono tante distribuzioni?

- Scopi diversi (desktop, server, embedded)
- Filosofie diverse
- Target di utenza diverso

## 1. Famiglia Debian

- Debian
- Ubuntu (e derivate)
- Linux Mint
- Pop!\_OS
- Package: .deb (apt)

## 2. Famiglia Red Hat

- Red Hat Enterprise Linux (RHEL)
- Fedora
- CentOS / Rocky Linux / AlmaLinux
- Package: .rpm (dnf/yum)

## 3. Famiglia Arch

- Arch Linux
- Manjaro
- EndeavourOS
- Package: pacman

## 4. Altre famiglie

- SUSE (openSUSE, SLES)
- Gentoo (source-based)
- Slackware
- Alpine Linux
- Distribuzioni indipendenti

# Ubuntu: La Distribuzione più Popolare

## Caratteristiche:

- Basata su Debian
- Sviluppata da Canonical
- Focus su usabilità
- Rilasci ogni 6 mesi
- LTS ogni 2 anni (5 anni supporto)
- Desktop environment: GNOME (default)

## Varianti ufficiali:

- Kubuntu (KDE)
- Xubuntu (Xfce)
- Lubuntu (LXQt)
- Ubuntu MATE
- Ubuntu Server

## Versioni LTS:

- 18.04 (Bionic Beaver)
- 20.04 (Focal Fossa)
- 22.04 (Jammy Jellyfish)
- 24.04 (Noble Numbat)

## Punti di forza:

- Documentazione eccellente
- Comunità molto attiva
- Ampio supporto hardware
- Repository ricco
- Installazione semplice

# Fedora: Innovazione e Cutting Edge

## Caratteristiche:

- Sponsorizzata da Red Hat
- Tecnologie all'avanguardia
- Rilascio ogni 6 mesi
- Base per RHEL
- Desktop environment: GNOME (Workstation)
- Package manager: DNF

## Edizioni:

- Fedora Workstation (desktop)
- Fedora Server
- Fedora IoT
- Fedora CoreOS

## Spin ufficiali:

- KDE Plasma
- Xfce
- LXQt
- MATE-Compiz
- Cinnamon
- LXDE
- i3 (tiling WM)

## Filosofia:

- Software libero prioritario
- Innovazione
- "First" - prime implementazioni
- SELinux di default



## Caratteristiche:

- Progetto comunitario (no azienda)
- Rilasci "when ready"
- Contratto sociale
- Supporto multi-architettura
- Repository enormi (59.000+ pacchetti)

## Rami (branches):

- **Stable** - produzione
- **Testing** - test per prossima stable
- **Unstable (Sid)** - sviluppo
- **Experimental** - sperimentale

## Versioni recenti:

- Debian 10 (Buster)
- Debian 11 (Bullseye)
- Debian 12 (Bookworm)

## Punti di forza:

- Estrema stabilità
- Vasta scelta di software
- Base per molte distro
- Governata democraticamente
- Supporto lungo

# Arch Linux: Semplicità e Controllo

## Caratteristiche:

- Rolling release
- Configurazione manuale
- Filosofia KISS (Keep It Simple, Stupid)
- Sempre aggiornata
- Pacman come package manager
- AUR (Arch User Repository)

## Filosofia:

- L'utente controlla tutto
- Documentazione eccellente (ArchWiki)
- Software vanilla (non modificato)
- Installazione da zero

## Vantaggi:

- Software sempre recente
- Massima personalizzazione
- Leggera e veloce
- Imparare Linux a fondo

## Derivate user-friendly:

- Manjaro
- EndeavourOS
- Garuda Linux
- ArcoLinux

## Target: Utenti esperti

# Gestione dei Pacchetti

I package manager gestiscono l'installazione, aggiornamento e rimozione del software.

DNF (Fedora/RHEL):

APT (Debian/Ubuntu):

```
#
```

```
sudo
```

```
#
```

```
Aggiornare
```

```
repository
```

```
apt
```

```
update
```

```
Aggiornare
```

```
#
```

```
sudo
```

```
#
```

# Desktop Environment

**Ambiente grafico che fornisce l'interfaccia utente.**

## GNOME

- Moderno e minimalista
- Workflow basato su attività
- Richiede risorse medie-alte
- Default: Ubuntu, Fedora

## KDE Plasma

- Altamente personalizzabile
- Look simile a Windows
- Molte funzionalità
- Default: Kubuntu

## Xfce

- Leggero e veloce

## MATE

- Fork di GNOME 2
- Tradizionale
- Medio-leggero
- Default: Ubuntu MATE

## Cinnamon

- Da Linux Mint
- Simile a Windows
- Elegante
- Default: Linux Mint

## LXQt / LXDE

- Leggerissimi

# File System Hierarchy Standard (FHS)

## Struttura standard delle directory in Linux:

- **/** - root, radice del filesystem
- **/bin** - comandi essenziali
- **/boot** - file boot loader
- **/dev** - file dispositivi
- **/etc** - configurazioni sistema
- **/home** - directory utenti
- **/lib** - librerie condivise
- **/media** - mount point removibili
- **/mnt** - mount point temporanei
- **/opt** - software opzionale
- **/proc** - filesystem virtuale
- **/root** - home dell'utente root
- **/sbin** - comandi amministrazione
- **/tmp** - file temporanei
- **/usr** - programmi utente
- **/var** - dati variabili

**Tutto è un file** in Linux: dispositivi, processi, configurazioni...

# Systemd: Sistema di Init Moderno

## Cos'è systemd?

- Sistema di init e service manager
- Sostituisce SysV init
- Avvio parallelo dei servizi
- Gestione dipendenze
- Logging centralizzato (journald)

## Componenti principali:

- systemd - processo init (PID 1)
- systemctl - gestione servizi
- journalctl - visualizza log
- Unit files - configurazione servizi

## Comandi comuni:

- `systemctl start servizio`
- `systemctl stop servizio`
- `systemctl restart servizio`
- `systemctl status servizio`
- `systemctl enable servizio`
- `systemctl disable servizio`

## Visualizzare log:

- `journalctl -xe`
- `journalctl -u servizio`
- `journalctl -f`

# Sistema di Permessi in Linux

**Linux è un sistema multiutente con controllo degli accessi.**

**Modificare permessi:**

```
#  
$  
chmo  
u  
-  
2  
:  
:  
chmo  
8  
-
```

# Vantaggi di GNU/Linux

## Libertà e Apertura:

- Codice sorgente disponibile
- Nessun costo di licenza
- Personalizzazione totale
- Privacy e controllo

## Stabilità e Sicurezza:

- Meno malware
- Sistema di permessi robusto
- Aggiornamenti frequenti
- Server affidabili

## Prestazioni:

- Efficiente nell'uso risorse
- Ottimizzabile
- Lunga durata hardware
- Veloce anche su PC datati

## Flessibilità:

- Molteplici distribuzioni
- Desktop environments vari
- Adatto a ogni utilizzo
- Da embedded a supercomputer

**Linux domina:** Server (96% top500 supercomputer), Cloud, Android, IoT



## ① Server e Data Center

- Web server (Apache, Nginx)
- Database server (MySQL, PostgreSQL)
- Cloud computing (AWS, Azure, Google Cloud)

## ② Sviluppo Software

- Ambienti di sviluppo
- DevOps e CI/CD
- Container (Docker, Kubernetes)

## ③ Sistemi Embedded

- Router e dispositivi di rete
- Smart TV
- Dispositivi IoT

## ④ Desktop e Workstation

- Programmazione
- Grafica e multimedia
- Uso quotidiano

## ⑤ Educazione e Ricerca

- Laboratori didattici

## Documentazione ufficiale:

- Man pages (comando `man`)
- Info pages (`info`)
- `/usr/share/doc`
- Wiki delle distribuzioni

## Risorse online:

- ArchWiki - documentazione eccellente
- Linux Documentation Project
- Ubuntu Documentation
- Stack Overflow / Unix StackExchange

## Community:

- Forum delle distribuzioni
- Reddit (`r/linux`, `r/linuxquestions`)
- IRC e Discord

## Punti chiave:

- Il **sistema operativo** è il software fondamentale che gestisce l'hardware
- Il **kernel** è il cuore del SO, con accesso diretto all'hardware
- La **shell** fornisce l'interfaccia a riga di comando per interagire con il sistema
- **GNU/Linux** combina il kernel Linux con gli strumenti GNU
- Le **distribuzioni** offrono sistemi completi per diverse esigenze
- Linux è **libero, sicuro, stabile e flessibile**

Linux non è solo un sistema operativo,  
è una filosofia di libertà e condivisione!

## Libri:

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2014). *Modern Operating Systems* (4th ed.). Pearson.
- Love, R. (2010). *Linux Kernel Development* (3rd ed.). Addison-Wesley.
- Shotts, W. (2019). *The Linux Command Line* (2nd ed.). No Starch Press.
- Ward, B. (2021). *How Linux Works* (3rd ed.). No Starch Press.

## Risorse Online:

- Linux Kernel Archives: <https://www.kernel.org/>
- The Linux Documentation Project: <https://tldp.org/>
- ArchWiki: <https://wiki.archlinux.org/>
- GNU Project: <https://www.gnu.org/>
- Free Software Foundation: <https://www.fsf.org/>

## Standard: