



# Introduzione a Linux

Lezione 6  
Amministrazione di base

Ruggero Donida Labati

**Laboratorio di Sistemi Operativi**

Università degli Studi di Milano  
Dipartimento di Informatica  
A.A. 2024/2025

*Materiale prodotto originariamente da Ruggero Donida Labati e Angelo Genovese*

RUGGERO DONIDA LABATI – INTRODUZIONE A LINUX – LEZIONE 6 – AMMINISTRAZIONE DI BASE

1

## Panoramica della lezione

- Verranno presentati alcuni aspetti più “avanzati”
  - Kernel
  - Servizi
  - Log
- Saranno introdotte alcune tecniche per la loro configurazione



RUGGERO DONIDA LABATI – INTRODUZIONE A LINUX – LEZIONE 6 – AMMINISTRAZIONE DI BASE

2

## Sommario (1/2)

### 1. Introduzione al kernel di Linux

- Definizione
- Caratteristiche
- Struttura
- Inizializzazione
- Messaggi

### 2. Servizi in Linux

- Definizione di servizio
- I servizi principali
- Comandi da shell per la gestione dei servizi
- Configurazione dei servizi



RUGGERO DONIDA LABATI – INTRODUZIONE A LINUX – LEZIONE 6 – AMMINISTRAZIONE DI BASE

3

## Sommario (2/2)

### 3. Log di sistema

- Posizione dei file di log
- Log principali
- Visualizzazione dei file di log

### 4. Gestione del backup

- Backup con *Tar*
- Backup con *Rsync*
- Backup con *dd*

### 5. Esercizi

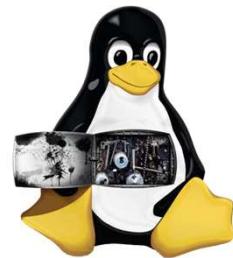


RUGGERO DONIDA LABATI – INTRODUZIONE A LINUX – LEZIONE 6 – AMMINISTRAZIONE DI BASE

4

## 1. Introduzione al kernel di Linux

1. Definizione
2. Caratteristiche
3. Struttura
4. Inizializzazione
5. Messaggi



RUGGERO DONIDA LABATI – INTRODUZIONE A LINUX – LEZIONE 6 – AMMINISTRAZIONE DI BASE

5

1. INTRODUZIONE AL KERNEL DI LINUX – DEFINIZIONE DI KERNEL

### Definizione di kernel (1/2)

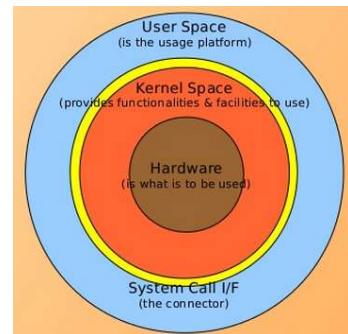
- Cuore (letteralmente) del sistema operativo
- Ha il controllo di tutto ciò che succede nel sistema
- Prima parte del sistema operativo ad essere caricata in fase di avvio

RUGGERO DONIDA LABATI – INTRODUZIONE A LINUX – LEZIONE 6 – AMMINISTRAZIONE DI BASE

6

## Definizione di kernel (2/2)

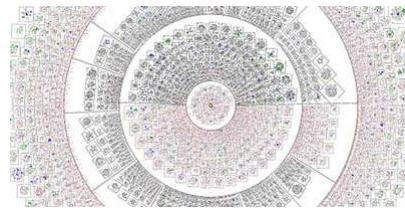
- Contiene le primitive per gestire le funzioni base del sistema operativo
  - Gestione della memoria
  - Scheduling dei processi
  - Comunicazione tra i processi
  - Gestione dell'hardware



- ## Caratteristiche del kernel di Linux (1/2)
- Scritto sulla base di diversi kernel UNIX
  - Monolitico
    - Contiene tutte le funzioni di base del sistema operativo e i driver per gestire l'hardware
    - Organizzato in moduli logici
      - Moduli aggiuntivi possono essere caricati a runtime
  - Multi-threaded

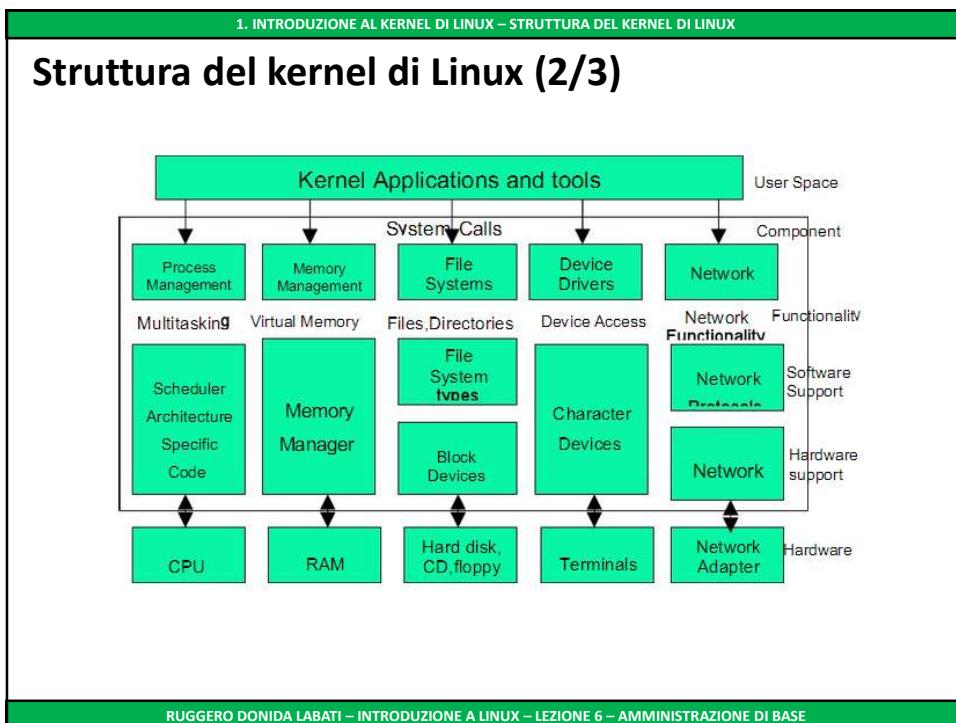
## Caratteristiche del kernel di Linux (2/2)

- Non-preemptive
  - Diverse funzioni del kernel devono essere eseguite senza poter essere interrotte
- Supporta molti dei file system attuali

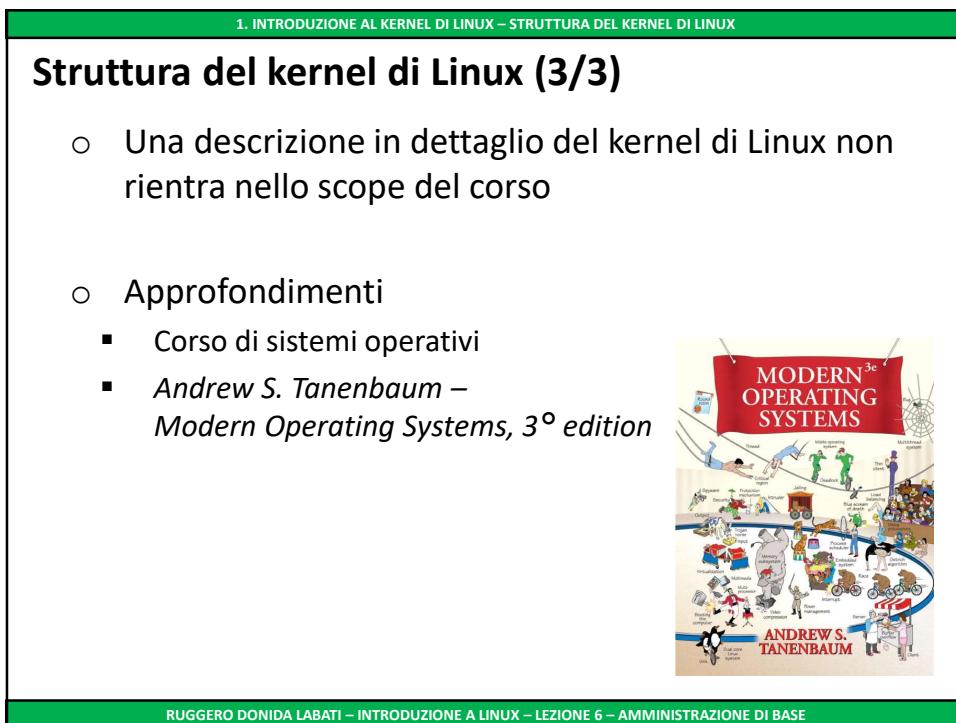


## Struttura del kernel di Linux (1/3)

- Gestione della memoria
- Gestione dei processi
- Gestione del file system
- Driver dell'hardware
- Gestione della rete



11



12

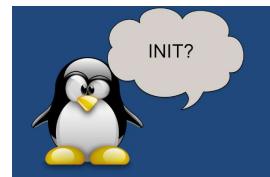
## Inizializzazione del kernel (1/2)

- Accensione della macchina
- Il firmware della macchina (BIOS) inizializza il disco di avvio
- Il firmware esegue il bootloader contenuto nel disco di avvio
  - Es. GRUB



## Inizializzazione del kernel (2/2)

- Il bootloader esegue il kernel
- Il kernel riconosce le periferiche
- Il kernel esegue il processo *init*
  - `/bin/init`



## Messaggi del kernel

- Il kernel è il primo programma eseguito dopo il bootloader
- Il kernel invia messaggi durante la fase di boot
  - Es. le varie periferiche riconosciute
- I messaggi del kernel possono essere letti
  - Con il comando *dmesg*
  - visualizzando il file */var/log/dmesg*
  - Utile in caso di periferiche non riconosciute, errori vari, ecc.

## 2. Servizi in Linux

1. Definizione di servizio
2. I servizi principali
3. Comandi da shell per la gestione dei servizi
4. Configurazione dei servizi



## Definizione di servizio (1/2)

- Sono programmi che spesso vengono eseguiti automaticamente al riavvio.
- Chiamati anche *daemon*
- All'installazione ogni servizio ha associato un file che ne gestisce l'avvio e lo spegnimento
- Questi file si trovano in */etc/init.d*

## Definizione di servizio (2/2)

- I file che gestiscono i servizi accettano diversi parametri
  - start
    - Fa partire il servizio
  - stop
    - Blocca il servizio
  - status
    - Mostra lo stato di esecuzione del servizio
  - restart
    - Blocca e fa ripartire il servizio
  - reload
    - Impone al servizio di rileggere i file di configurazione

## I servizi principali (1/4)

- *atd*
  - Gestisce l'esecuzione temporizzata dei comandi
- *crond*
  - Permette di schedulare dei comandi di sistema
- *gpm*
  - Gestisce il mouse in modalità testo
- *httpd*
  - Server web
- *identd*
  - Restituisce informazioni sui proprietari dei processi di rete

## I servizi principali (2/4)

- *init*
  - Gestisce l'avvio del sistema operativo
- *ipchains e iptables*
  - Gestiscono il firewall e il NAT
- *lpd*
  - Gestore di stampa
- *named*
  - Gestisce il DNS
- *networking*
  - Gestisce lo stack di rete

## I servizi principali (3/4)

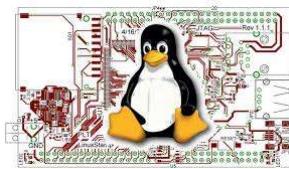
- *nfs*
  - Network File System, condivisione file tra UNIX
- *ntpd*
  - tiene l'orologio di sistema regolato
- *postfix / sendmail*
  - MTA, gestori di posta elettronica
- *postgresql / mysql*
  - Gestori di database SQL
- *random*
  - Garantisce entropia nel generatore di numeri casuali

## I servizi principali (4/4)

- *smb*
  - Permette la condivisione di file con Windows
- *squid*
  - Proxy server
- *sshd*
  - Accesso sicuro da remoto
- *syslog*
  - Gestisce i log del sistema

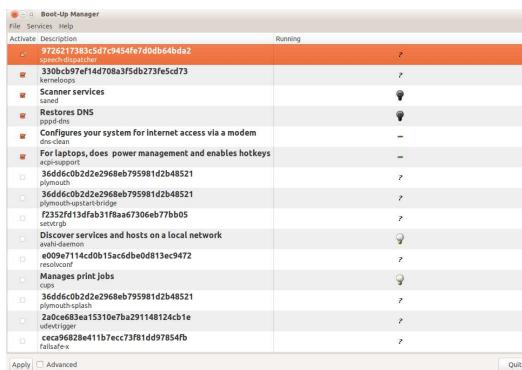
## Comandi da shell per la gestione dei servizi (1/2)

- Lista dei servizi in esecuzione
  - `sudo service --status-all`
  
- Avvio / arresto / riavvio di un servizio
  - `sudo service <nome_servizio> start`
  - `sudo service <nome_servizio> stop`
  - `sudo service <nome_servizio> restart`



## Comandi da shell per la gestione dei servizi (2/2)

- Tool con GUI per la gestione dei servizi
  - Boot-Up Manager (*BUM*)
  - `sudo apt-get install bum`
  - `sudo bum`



## Configurazione dei servizi

- La configurazione dei servizi non è centralizzata
- Ogni servizio ha il suo file di configurazione
- Sono sparsi nella directory `/etc`
- In rete c'è molta letteratura per i servizi più comuni
  - Es. server Apache per il servizio `httpd`

## 3. Log di sistema

1. Posizione dei file di log
2. Log principali
3. Visualizzazione dei file di log



## Posizione dei file di log

- Molti demoni usano *syslogd* per gestire i log
- Il suo file di configurazione è */etc/syslogd.conf*
- Tipicamente mette i log in */var/log*
- Il file di default è */var/log/messages*  
(o */var/log/syslog*)

## Log principali (1/2)

- **messages** (o *syslog*)
  - Il file principale di log, tutti gli eventi di sistema e i messaggi inoltrati a dmesg vengono scritti su questo file
- **boot.log**
  - status dei servizi durante le fasi di avvio del sistema
- **dmesg**
  - Eventi inerenti al processo di boot del kernel
- **cron**
  - Eventi relativi allo scheduling delle attività e del servizio crond

## Log principali (2/2)

- cups
    - Eventi relativi alle code di stampa
  - maillog
    - Eventi relativi ai mail server smtp
  - secure
    - Eventi inerenti alla sicurezza del sistema
  - wtmp
    - Eventi inerenti agli accessi effettuati sul sistema con una dettagliata reportistica

## Visualizzazione dei file di log (1/2)

- I file di log sono semplici file di testo
  - È però poco pratico leggere un file di log interamente
  - Linux mette a disposizione strumenti per la lettura facilitata
    - Ricerca testuale
    - Accesso alla parte finale del file
    - Visualizzazione di piccole parti

## Visualizzazione dei file di log (2/2)

- Ricerca di file o directory
  - *find <espressione>*
    - Es. *find hello\**
- Ricerca testuale
  - *grep <stringa\_da\_cercare> <nome\_file>*
- Visualizzazione ultime righe del file
  - *tail <nome\_file>*
- Visualizzazione di alcune parti alla volta
  - *less <nome\_file>*

```
root@debian:~# logrotate -d /etc/logrotate.d/apache2.conf
reading config file /etc/logrotate.d/apache2.conf
Handling 1 logs
rotate 3
rotate 3 in the config file
rotating pattern: /var/log/apache2/* 10485760 bytes (3 rotations)
empty log files are rotated, old logs are removed
considering log /var/log/apache2/access.log
log does not need rotating
considering log /var/log/apache2/access.log.1
log does not need rotating
considering log /var/log/apache2/access.log.2.gz
log does not need rotating
considering log /var/log/apache2/access.log.3.gz
log does not need rotating
considering log /var/log/apache2/access.log.4.gz
log does not need rotating
considering log /var/log/apache2/access.log.5.gz
log does not need rotating
```

*Annotations:*

- rotate 3 in the config file
- log rotates examines all log files inside /var/log/apache2 and rotates them against the directives in the config file.

## 4. Gestione del backup

1. Backup con *Tar*
2. Backup con *Rsync*
3. Backup con *dd*



## Backup con Tar (1/3)

- Con Ubuntu è possibile eseguire un backup con lo strumento utilizzato per archiviare o comprimere file
  - *tar*
- Spostarsi nella directory /
  - *cd /*
- Backup dell'intero sistema
  - *sudo tar -cvpzf /backup.tgz --exclude=/proc --exclude=/lost+found --exclude=/backup.tgz --exclude=/mnt --exclude=/sys /*

## Backup con Tar (2/3)

- Opzioni
  - *tar*
    - È il programma necessario a fare il backup.
  - *c*
    - Crea un nuovo archivio di backup.
  - *v*
    - Modalità verbose
  - *p*
    - Mantiene i permessi di tutti i file.
  - *z*
    - Comprime il file di backup con gzip per renderlo più piccolo.

## Backup con Tar (3/3)

- Opzioni
  - *f <nameofile>*
    - Specifica dove salvare il backup, */backup.tgz* è il file utilizzato in questo esempio.
  - *--exclude*
    - Le directory da escludere dal backup
    - Assicurarsi di non includere anche il file di backup stesso, altrimenti si potrebbero avere strani risultati
    - Consigliato non includere le cartelle */mnt* e */media* per non fare il backup di altre partizioni o dispositivi montati
  - Directory di cui vogliamo fare il backup
    - Consigliata */*

## Backup con Rsync

1. Introduzione a Rsync
2. Backup locale
3. Backup in rete
4. Interfacce grafiche



## Introduzione a Rsync

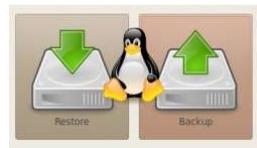
- Strumento a riga di comando per il backup locale o remoto
  - `rsync -help`
  - `man rsync`
- Supporta backup incrementali
- Richiede filesystem con permessi Unix-like
  - No NTFS o FAT

## Backup in locale (1/2)

- Comando a terminale
  - `sudo rsync --exclude=<file_da_escludere> --delete -azvv </directory_di_origine> </directory_di_destinazione>`
- Opzioni
  - `--exclude`
    - Esclude i file specificati dal backup
  - `--delete`
    - Cancella nella cartella di destinazione i file non più presenti

## Backup in locale (2/2)

- Opzioni
  - `-a`
    - Copia ricorsivamente e conserva i permessi e le date dei file
  - `-z`
    - Comprime i dati
  - `-vv`
    - Modalità verbose



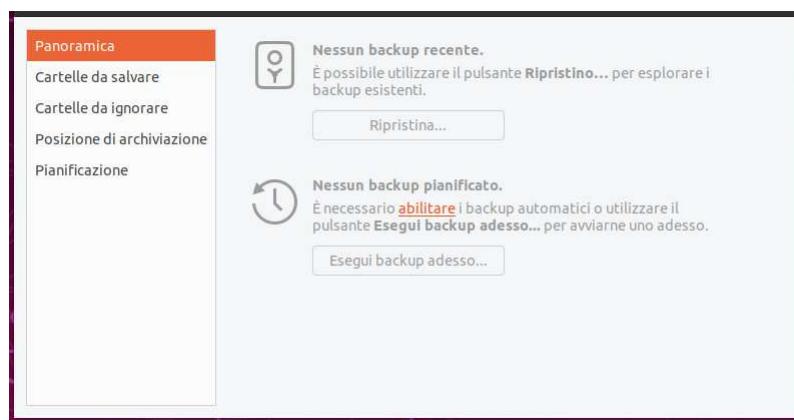
## Backup in rete

- Comando a terminale
  - `sudo rsync --delete -azvv -e ssh </origine> utenteremoto@hostremoto:</destinazione>`
- Opzioni
  - `-e`
    - Specifica la shell remota da utilizzare (Es. ssh)

## Interfacce grafiche per Rsync (1/2)

- È possibile utilizzare Rsync attraverso diverse interfacce grafiche
  - Backup (Ubuntu)
    - Permette di effettuare un backup del sistema in modo abbastanza facile e del tutto funzionale
  - Grsync
    - Semplice interfaccia grafica che permette l'utilizzo delle funzioni principali di Rsync
  - Back in time
    - Semplice strumento che utilizza Rsync per compiere backup incrementali

## Interfacce grafiche per Rsync (2/2)



## Backup con dd

- Comando Unix che copia i dati a blocchi
- Permette di ottenere una copia della partizione
  - `dd if=/dev/hda1 of=<nome_immagine.dd>`
- Permette di ripristinare la copia salvata
  - `dd if=<nome_immagine.dd> of=/dev/hda1`

```
[~]$ dd
```

## In sintesi

1. Introduzione al kernel di Linux
2. Servizi in Linux
3. Log di sistema
4. Gestione del backup



## 5. Esercizi (1/3)

- Visualizzate i messaggi di sistema e cercate di interpretarne l'output
  - Visualizzatene solo l'ultima parte
  - Cercate gli eventi relativi al processo «init»
- Elencate tutti i file di log
- Cercate il file di log di samba

## 5. Esercizi (2/3)

- Archiviate dei file con *Tar*
  - Prima senza comprimere
  - Poi comprimendo
    - Prima *gzip*, poi *bzip2*
  - Cambia la dimensione del file?

## 5. Esercizi (3/3)

- Installate il server http, quindi provate ad interrompere e riavviare il servizio
- Provate ad interrompere e riavviare il servizio che gestisce la rete
- Cercate il vostro nome utente nei principali file di log del sistema
- Create un piano di backup con *Sbackup*