Laboratorio di Sistemi Operativi  
Appunti del corso a cura dello studente Gianni Adamo

A.A. 2024 - 2025

Sommario

[0. Premessa 3](#_Toc178236307)

[0.1 Riferimenti 3](#_Toc178236308)

[0.2 Info e Contatti 3](#_Toc178236309)

[1. Introduzione a Unix [BETA] 4](#_Toc178236310)

[2. [LEZIONE 2] [BETA] 5](#_Toc178236311)

# 0. Premessa

Questi appunti sono stati scritti con il solo scopo di approfondire e assistere lo studio degli argomenti trattati nell’omonimo corso *“Laboratorio di Sistemi Operativi”*, tenuto dalla prof. Alessandra Rossi presso l’*Università degli Studi di Napoli Federico II*, per il CDL in Informatica. Tale documento fa riferimento al corso dell’A.A 2024-2025.

Ulteriori informazioni sono contenute in fondo alla pagina corrente.

## 0.1 Riferimenti

Il materiale didattico utilizzato per la creazione di questo documento è così elencato:

* *Slides del corso “Laboratorio di Sistemi Operativi” – Prof. Alessandra Rossi*
* *Appunti degli anni precedenti (riferiti al medesimo corso)*
* *Annotazioni scritte a lezione*
* *Approfondimenti trovati in rete*
* *Esercitazioni e test vari effettuati da terminale*

Tutte le informazioni contenute in questo documento sono state accuratamente selezionate per coprire al meglio gli argomenti trattati durante il corso e fornire allo studente una guida per assimilare i concetti spiegati a lezione. L’autore non si assume alcuna responsabilità riguardo la completezza dell’elaborato e agli esiti che ne susseguono. Eventuali variazioni che il corso potrebbe subire negli anni a venire non saranno indicate all’interno del documento. Per ulteriori informazioni, consultare la sezione seguente *Info e Contatti.*

Tale documento è stato interamente scritto dallo studente Giovanni Adamo, iscritto al CDL in Informatica presso l’*Università degli Studi di Napoli Federico II.*

## 0.2 Info e Contatti

Di seguito, alcuni canali e piattaforme tramite cui è possibile raggiungermi, per contatti o visione di ulteriori produzioni (in ambito accademico e/o personale).

* **Email:** [gianni.adamo@hotmail.it](mailto:gianni.adamo@hotmail.it)
* **GitHub:** <https://github.com/Gazen27>
* **Social:** Mi trovi su tutti i social come @gazen27

# 1. Introduzione a Unix [BETA]

I sistemi operativi Unix si basano sullo standard Posix e ne esistono molti (sicuramente tra i più conosciuti ci sono Linux e OSX); le caratteristiche principali di Unix sono molteplici: la possibilità di multi-processi con time-sharing, ma anche l’uso di multi-utenti, dove è ogni utente può accedere alle proprie risorse o a quelle generali dell’OS grazie a un sistema di protezioni e permessi; tali permessi vengono stabiliti dall’utente amministratore, nei sistemi Unix chiamato **root**.

Il sistema operativo di base di tutti i sistemi Unix viene chiamato **File System** gerarchico: è rappresentabile come un albero, dove la radice, è proprio *root*.

Il sistema Unix è basato su **Kernel**, ovvero il nucleo dell’OS, che si deve adattare all’architettura dell’hardware (ad esempio sistemi intel o AMD), ed è l’unica porzione dell’OS alla quale un utente generico non può accedere se non ha i permessi di *root* (modalità privilegiata); il *kernel* gestisce tutte le risorse essenziali (CPU, memorie e periferiche, trattate come file), mentre tutto il resto, anche l’interazione con l’utente, è ottenuto tramite programmi eseguiti dal *kernel*, che accedono alle risorse hardware mediante richieste a quest’ultimo.

Il *kernel* comunica con le varie applicazioni tramite le *system call*, che consentono la comunicazione con il *kernel* e tutte le varie risorse.

I sistemi Unix sono *multi-utente*, dove ogni utente ha un username, una password e un identificativo univoco (user id o semplicemente id). Ogni utente ha un set di permessi associati che vanno a definire cosa quell’utente può fare o meno; l’unico *superuser* è *root*, il cui *user-id* è 0.

La **shell** è un interprete di comandi che si interpone tra utente e sistema operativo; nei sistemi Unix, qualsiasi operazione può essere eseguita da una serie di comandi da *shell*.

La *shell* può eseguire uno script (file bash) oppure interagire in modalità interattiva a linea di comando; l’utente può infatti fornire al prompt un comando tra:

* Nome di un comando built-in
* Nome di un file eseguibile
* Nome di uno script (dotato di permesso di esecuzione)

Tutti i comandi Unix hanno una struttura molto simile; il nome del comando è generalmente una sorta di abbreviazione dell’azione che si vuole effettuare e una serie di argomenti, che possono essere delle opzioni o flag (iniziano con “-”), dei parametri oppure dei file. L’ordine delle opzioni solitamente è irrilevante, mentre l’ordine dei parametri è molto importante.   
Inoltre, Unix è case sensitive.

Alcuni comandi Unix ricorrenti:

|  |  |
| --- | --- |
| ***ls*** | Visualizza la lista di file nella directory corrente |
| ***cd*** | Seguito dal nome di una directory, serve a spostarsi nella directory scelta |
| ***mkdir*** | Seguito dal nome di una directory inesistente, ne crea una con quel nome |
| ***rmdir*** | Seguito dal nome di una directory già esistente, la elimina |
| ***pwd*** | Visualizza qual è la directory corrente |
| ***exit*** *o* ***logout*** | Lascia la sessione |

Nei sistemi Unix, tutto è un file, altrimenti è un processo;

I tipi principali di file sono:

* **Ordinario:** …
* **Directory:** intesa come le cartelle e i loro percorsi, nei sistemi Unix.
* **Speciale:** tutto ciò che è identificato come una periferica I/O è visto come un file speciale, ovvero un link che viene aggiunto al file system.

Il sistema assegna biunivocamente a ciascun file un identificatore numerico detto *i-number* (“index-number”) per rintracciarli nel file system (all’indice 0 corrisponde *root*).

PERCORSO ASSOLUTO: arrivare al file da root  
PERCORSO RELATVO: arrivare al file da un dato punto (solitamente da dove ci troviamo)

# 2. [LEZIONE 2] [BETA]

Un *File System* Unix è sempre unico, ma può avere parti residenti su device rimuovibili:

* “montate” prima di potervi accedere (comando *mount*)
* “smontate” prima di rimuovere il supporto (comando *unmount*)

Chown cambia owner, chgrp cambia gruppo, chmod cambia i permessi.

Possono esistere gruppi vuoti ma mai utenti che non hanno un gruppo di appartenenza.