

*Corso di Laurea in Informatica A.A. 2023-2024*

---

# Laboratorio di Sistemi Operativi

---

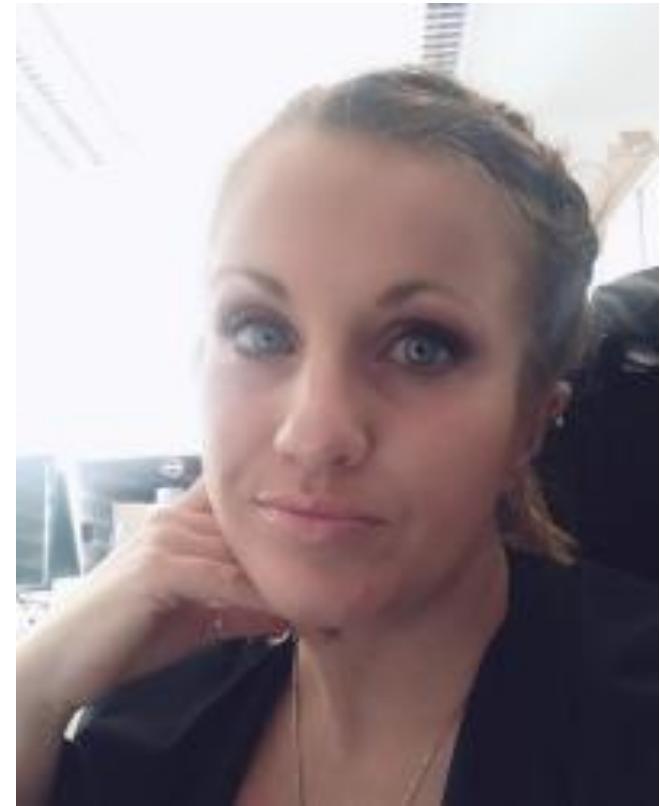
Alessandra Rossi

---

# Riferimenti

**Docente:** Alessandra Rossi  
**Email:** [alessandra.rossi@unina.it](mailto:alessandra.rossi@unina.it)

**Ricevimento:** su appuntamento  
**Luogo:** in presenza



---

# Informazioni Generali

---

- ❖ Crediti: 8 CFU
- ❖ Orario:
  - ❖ Lunedì: 14:30-16:30 aula CL-II-3
  - ❖ Martedì: 12:30-14:30 aula PT-II-A
  - ❖ Venerdì: 12:30-14:30 aula CL-II-3

---

# Informazioni Generali

---

- ❖ Canale unico
- ❖ Le lezioni saranno svolte in presenza a meno di casi eccezionali
- ❖ Esercitazioni

---

# Informazioni Generali

---

- ❖ Propedeuticità 2023/2024
  - ❖ Sistemi Operativi I, Algebra
  - ❖ In ogni caso **fate riferimento alla guida dello studente**
- ❖ Consigliabile aver seguito:
  - ❖ Il corso di Sistemi Operativi I
  - ❖ Programmazione II
  - ❖ Linguaggio C

---

# Ricevimento

---

❖ **Studio:**

- ❖ PRISCA lab, Piazzale Tecchio, 3 piano corpo arretrato a sinistra

❖ **Ricevimento:**

- ❖ Su richiesta
- ❖ Inviare una mail per prenotare il ricevimento due giorni prima
- ❖ Il ricevimento si effettuerà, previo appuntamento a mezzo email

---

# Obiettivi del Corso

---

- ❖ Strumenti e metodologie per la gestione di sistema e per lo sviluppo di applicazioni in ambiente Unix e Virtualizzazione

---

# Modalità di Esame

---

- ❖ Prova scritta su scripting e System Programming
- ❖ Progetto Linux (Architettura Client-Server)

---

# Modalità di Esame

---

Il Progetto consiste in:

- Realizzazione di un software con allegata relazione
- I progetti verranno assegnati verso la metà del corso
- Discussione sul software
  - ❖ Problematiche affrontate
  - ❖ Scelte implementative
  - ❖ Soluzioni particolari
  - ❖ ...

---

# Modalità di Esame

---

Il progetto viene assegnato a gruppi composti da **al più 2/3** studenti

- ❖ Sono ammessi gruppi composti da un solo studente solo in casi particolari (e.g., studenti lavoratori, etc..)
- ❖ Tutti i membri del gruppo devono discutere il Progetto.
- ❖ Non possono discutere il progetto separatamente.
- ❖ Non siete obbligati a sostenere l'esame nello stesso appello.

---

# Modalità di Esame

---

- ❖ Il progetto e la prova scritta sono indipendenti
- ❖ Il voto finale include l'esito dello scritto e dei progetti
- ❖ Durante la discussione del progetto sono possibili altre domande

---

# Programma di Massima Unix

---

## Introduzione a Unix/Linux

- concetti di base, file system, processi

## Shell

- uso dei più importanti comandi shell
- espressioni regolari
- script

## Programmazione di sistema

- breve introduzione al C
- compilatore e make, programmi eseguibili, librerie, debugger
- system calls e gestione di: memoria, file, processi, segnali, pipe, fifo, socket

## Moderni sistemi per il deployment

- La virtualizzazione dei sistemi operativi
- le macchine virtuali
- Docker, Docker Composed, ADE Development Environment, accenni a Vagrant

# Libri di Testo - UNIX

- ❖ **Testo di riferimento:**
  - ❖ Advanced Programming in the UNIX Environment - di W.R. Stevens e S.A. Rago - Addison Wesley  
(Almeno seconda edizione)
- ❖ **Altri testi consigliati:**
  - ❖ Advanced Unix Programming di Rochkind, Marc J. **2nd edition**
  - ❖ Expert C Programming: Deep C Secrets : Van Der Linden, Peter
- ❖ **Documenti utili:**
  - ❖ [Introduction to Linux](#)
  - ❖ [Bash Guide for Beginners](#)
  - ❖ Advanced Bash-Scripting Guide: [Inglese Italiano](#)
  - ❖ [Manuale ufficiale Bash](#)
  - ❖ [Guida alla Programmazione in Linux \(GAPIL\)](#)
  - ❖ Pagine del manuale di sistema
- ❖ Documenti segnalati a lezione e sul canale teams del corso

---

# Installare Linux

---

- ❖ Fondamentale disporre di Linux
  - ❖ Dual boot Windows Linux
  - ❖ Macchina dedicata: [www.ubuntu-it.org](http://www.ubuntu-it.org) (vanno bene anche altre distribuzioni Linux)
  - ❖ Macchina virtuale su windows:
    - <https://www.virtualbox.org/>
    - virtual machines for VirtualBox <https://www.osboxes.org/>
    - guida <https://www.osboxes.org/guide/>

# Introduzione a Unix

**1965** Bell Labs (con MIT e General Electrics) lavora ad un nuovo sistema operativo: Multics. Principali caratteristiche: multi-utente, multi-processo (time-sharing), con file system multi-livello (gerarchico).

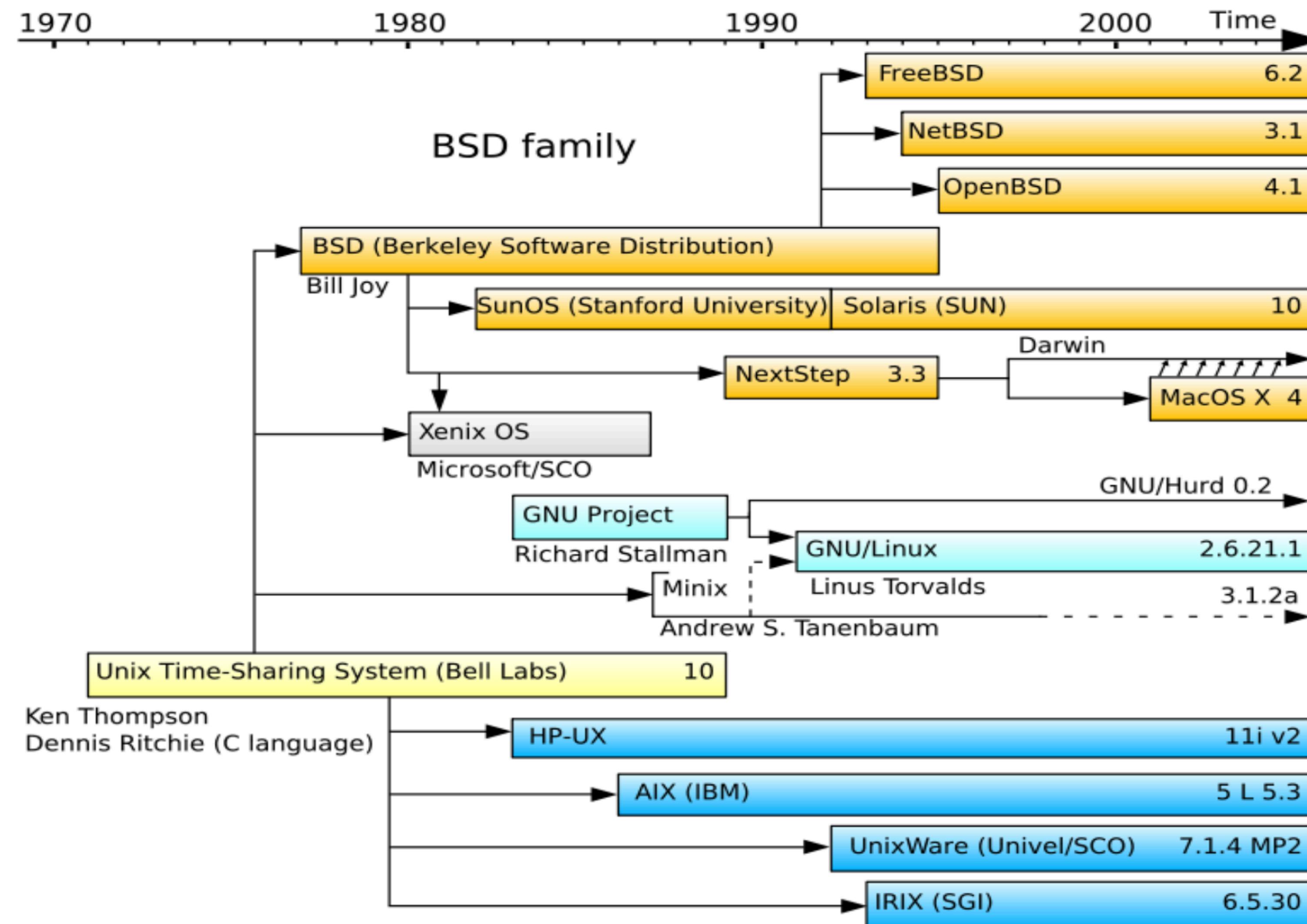
**1969** Bell Labs abbandona Multics. Ken Thompson, Dennis Ritchie, Rudd Canaday e Doug McIlroy progettano e implementano la prima versione di Unix insieme ad alcune utility. Il nome Unix è di Brian Kernighan: gioco di parole su Multics.

**1970** Prima versione di Unix, per PDP-7

**1973** Unix viene riscritto in C (Dennis Ritchie)



# UNIX SO



---

# UNIX

---

## Caratteristiche:

- Multi-utente, con sofisticato sistema di protezioni e permessi
- Multi-processo, con time-sharing
- Filesystem gerarchico con radice unica
- Basato su **kernel**:
  - il nucleo del SO è l'unica porzione che deve essere adattata all'Hw
  - e l'unica porzione che gira in modalità privilegiata

# Kernel

Il **nucleo** del sistema (**kernel**) gestisce le risorse essenziali: la CPU, la memoria, le periferiche

Tutto il resto, anche l'interazione con l'utente, è ottenuto tramite programmi eseguiti dal kernel, che accedono alle risorse hardware tramite delle richieste a quest'ultimo

# Kernel

Il kernel si occupa di:

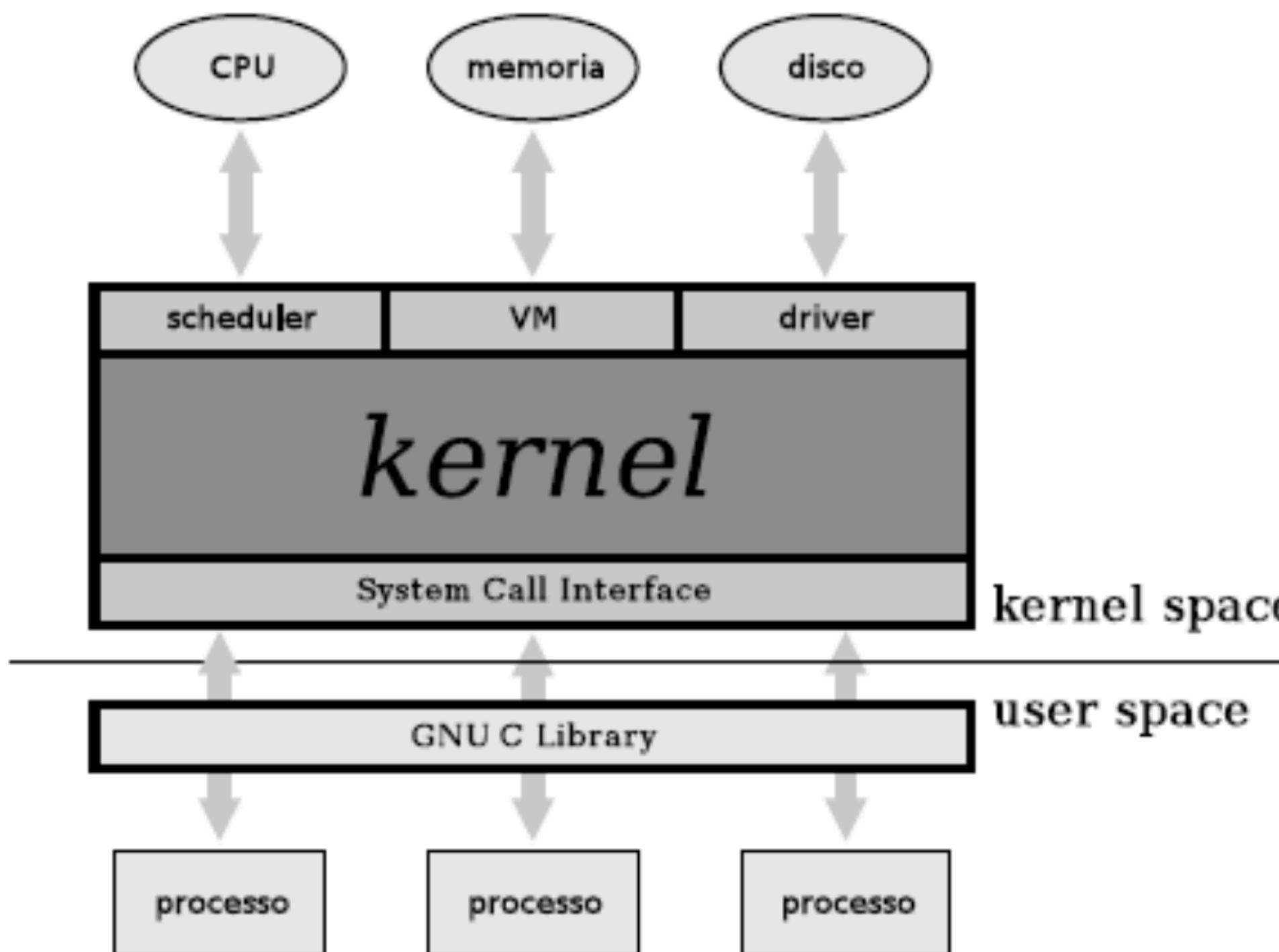
**CPU:** Lo **scheduler** stabilisce, ad intervalli fissi e sulla base priorità, quale “processo” deve essere mandato in esecuzione

**Memoria:** Il kernel gestisce la **memoria virtuale**, che consente di assegnare a ciascun processo uno spazio di indirizzi “virtuale” che il kernel, con l’ausilio della unità di gestione della memoria, rimappa sulla memoria fisica (RAM o disco)

**Periferiche:** Le periferiche vengono viste attraverso un’interfaccia astratta che permette di trattarle come fossero file, secondo il concetto per cui “everything is a file” (le interfacce di rete fanno eccezione).

# Kernel e System Calls

Le interfacce con cui i programmi possono accedere all'hardware vanno sotto il nome di chiamate al sistema (**system call**): un insieme di funzioni che un programma può chiamare, per le quali viene generata un'interruzione del processo, passando il controllo dal programma al kernel.



Queste chiamate al sistema vengono rimappata in funzioni definite dentro opportune librerie (Libreria Standard del C).

# Sistema Multi-utente

Il kernel Unix nasce fin dall'inizio come sistema multiutente

Ogni utente ha un nome (*username*), una password e un identificativo numerico (*user id* o *uid*)

Sono previsti meccanismi di *permessi* e *protezioni* per impedire che utenti diversi possano danneggiarsi a vicenda o danneggiare il sistema

E' presente un utente speciale privilegiato, superuser, il cui username è di norma *root*, ed il cui uid è zero: è l'amministratore del sistema

# La Shell

- La shell (*guscio*) è un interprete di comandi
- Si interpone tra l'utente ed il sistema operativo
- In sistemi Unix qualsiasi operazione può essere eseguita da una sequenza di comandi shell

Tra le Shell più utilizzate ci sono:

- Bourne shell, /bin/sh (Bell Labs)
- **Bourne-again shell, /bin/bash (Linux)**
- Cshell, /bin/csh (Berkeley)
- Korn shell, /bin/ksh (Bell Labs)
- TENEX C shell /bin/tcsh (BBN tech)

Il file /etc/shells contiene l'elenco delle shell installate dall'amministratore e disponibili a tutti gli utenti.

---

# La Shell

---

La shell può eseguire uno script oppure interagire in modalità interattiva

L'utente fornisce al prompt uno dei seguenti comandi:

- Nome di un comando built-in
- Nome di un file eseguibile
- Nome di uno script, cioè file testuale dotato del permesso di esecuzione

---

# I Comandi UNIX

---

**comando [argomenti]**

Gli argomenti possono essere:

- opzioni o flag (cominciano con un trattino “-”)
- parametri

separati da almeno un separatore (di default il carattere spazio)

L'ordine delle **opzioni** è, in genere, irrilevante

L'ordine dei **parametri** è, in genere, rilevante

**Unix è CASE SENSITIVE**

Esempio: gcc

# I Comandi UNIX

Comando	Significato
<b>ls</b>	visualizza la lista di file nella directory, come il comando <b>dir</b> in DOS
<b>cd directory</b>	cambia directory
<b>passwd</b>	cambia password
<b>file filename</b>	visualizza il tipo di file o il tipo di file con nome filename
<b>cat textfile</b>	riversa il contenuto di textfile sullo screen
<b>pwd</b>	visualizza la directory di lavoro corrente
<b>exit or logout</b>	lascia la sessione
<b>man command</b>	leggi pagine manuale su <b>command</b>
<b>info command</b>	leggi pagine Info su <b>command</b>
...	

---

# File System

---

- Struttura gerarchica (albero di directory)
- File senza struttura (byte streams)
- Sofisticato sistema di permessi

*"On a UNIX system, everything is a file; if something is not a file, it is a process."*

---

# I File UNIX

---

I tipi principali di File sono:

- **File ordinari**
- **Directory**
- **File speciali**

Il sistema assegna biunivocamente a ciascun file un identificatore numerico, detto ***i-number*** ("index-number"), che gli permette di rintracciarlo nel file system

# I File Ordinari

- Sono sequenze di byte (byte streams)
- Possono contenere informazioni qualsiasi (dati, programmi sorgente, programmi oggetto, ...)
- Il sistema non impone alcuna struttura interna



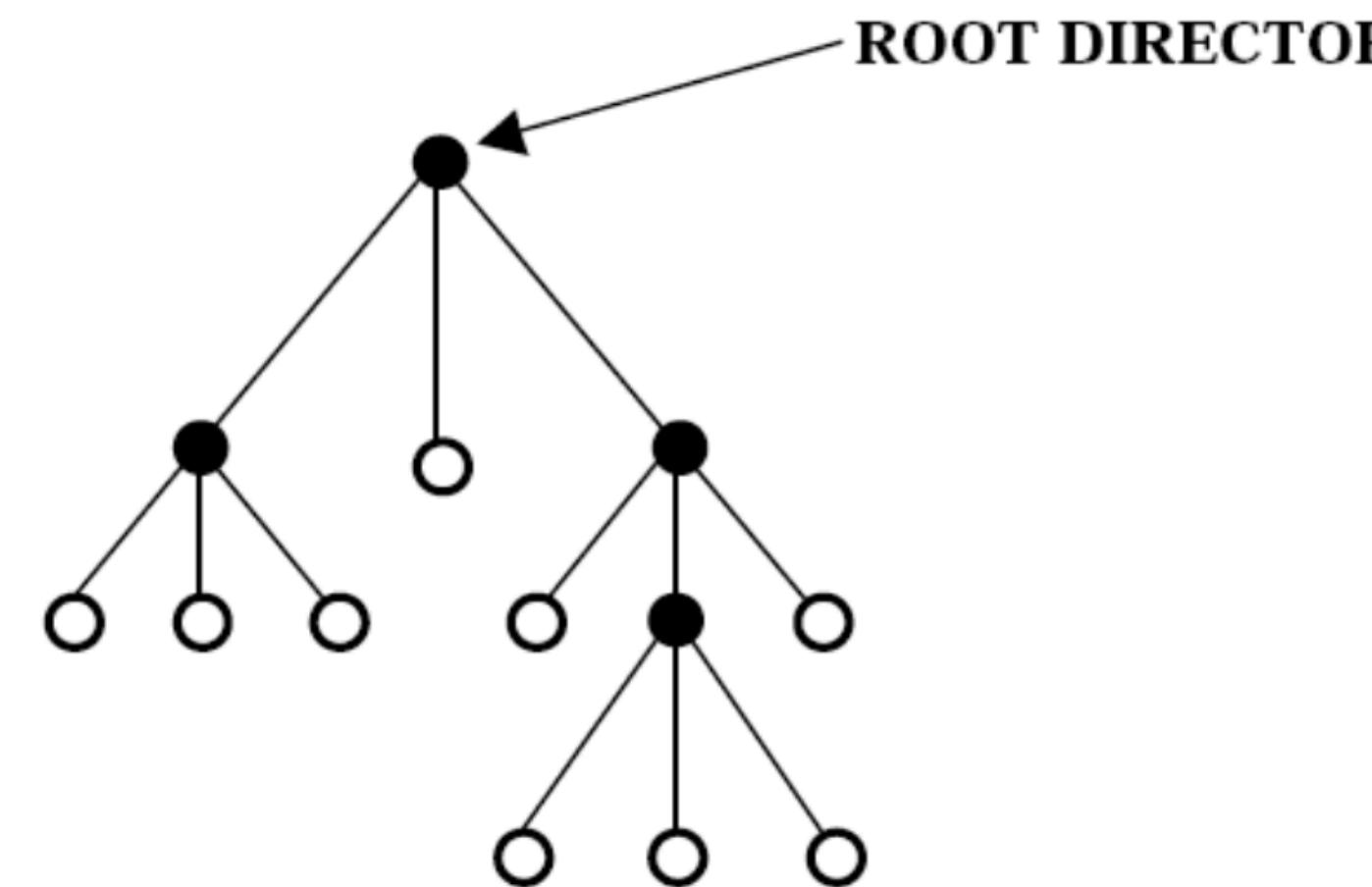
text file



binary file

# Organizzazione dei File

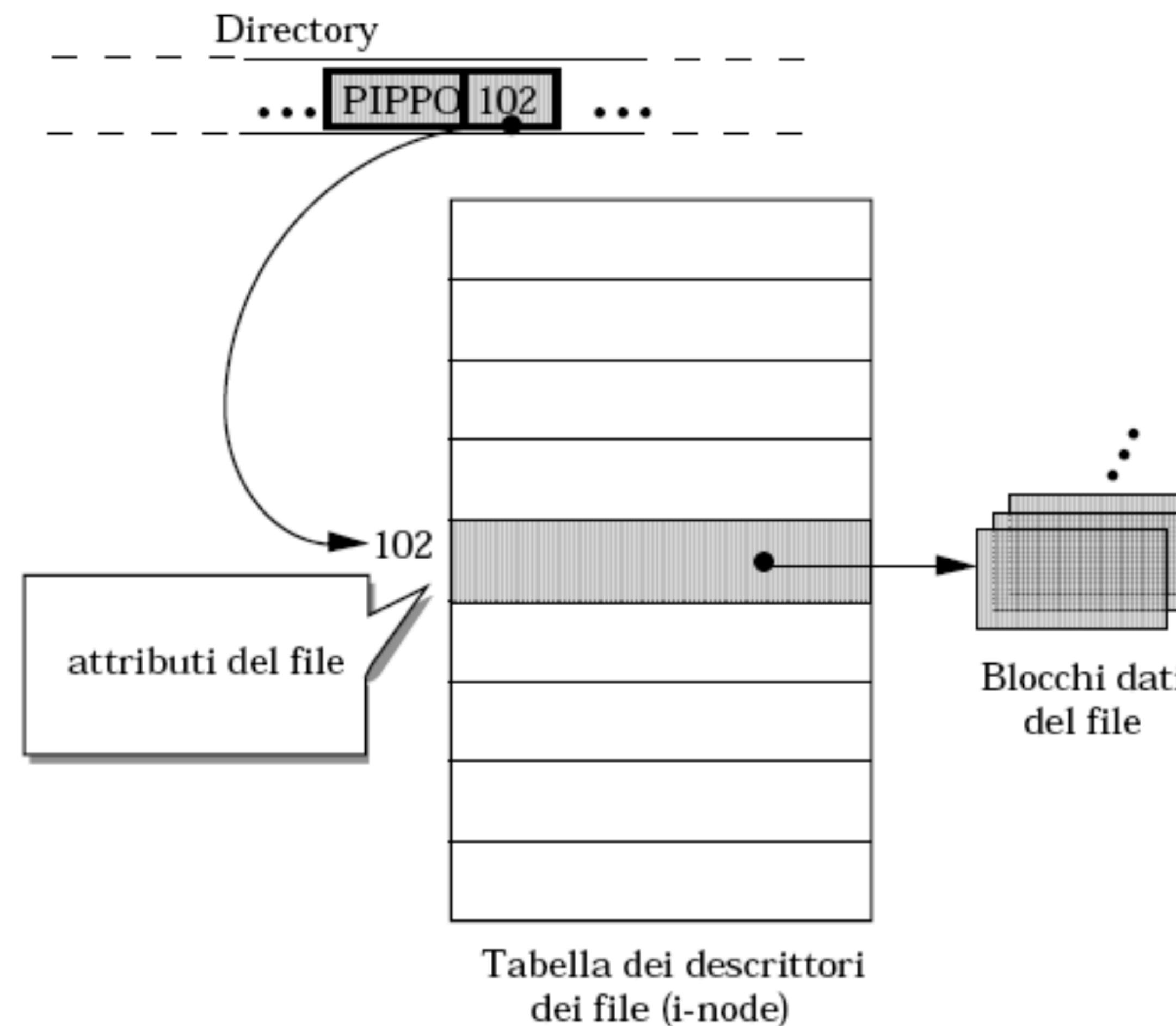
Per consentire all'utente di rintracciare facilmente i propri file, Unix permette di raggrupparli in cartelle, dette **directories**, organizzate in una (unica) struttura gerarchica:



● : directory

○ : file ordinario  
directory (vuota)  
file speciale

# Implementazione dei File



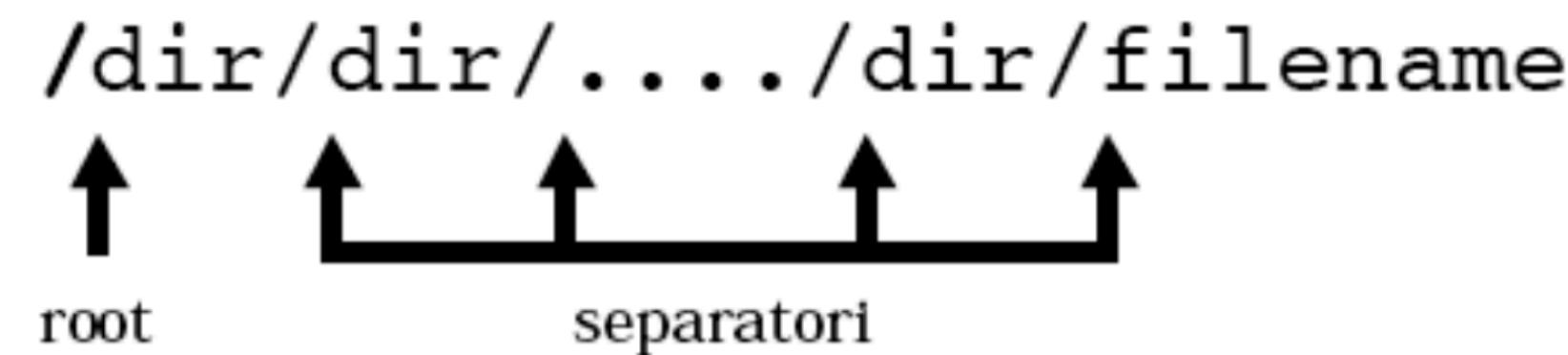
# Attributi di un File

Per ogni file (ordinario, directory, speciale) Unix mantiene le seguenti informazioni nel descrittore del file:

<b>Tipo</b>	ordinario, directory, speciale?
<b>Posizione</b>	dove si trova?
<b>Dimensione</b>	quanto è grande?
<b>Numero di links</b>	quanti nomi ha?
<b>Proprietario</b>	chi lo possiede?
<b>Permessi</b>	chi può usarlo e come?
<b>Creazione</b>	quando è stato creato?
<b>Modifica</b>	quando è stato modificato più di recente?
<b>Accesso</b>	quando è stato l'accesso più recente?

# Path Assoluti e Relativi

Si può identificare un file tramite il suo **path assoluto**, che descrive il cammino dalla root-directory al file:



Se un path **non** comincia con “/”, allora si intende **relativo alla working directory**

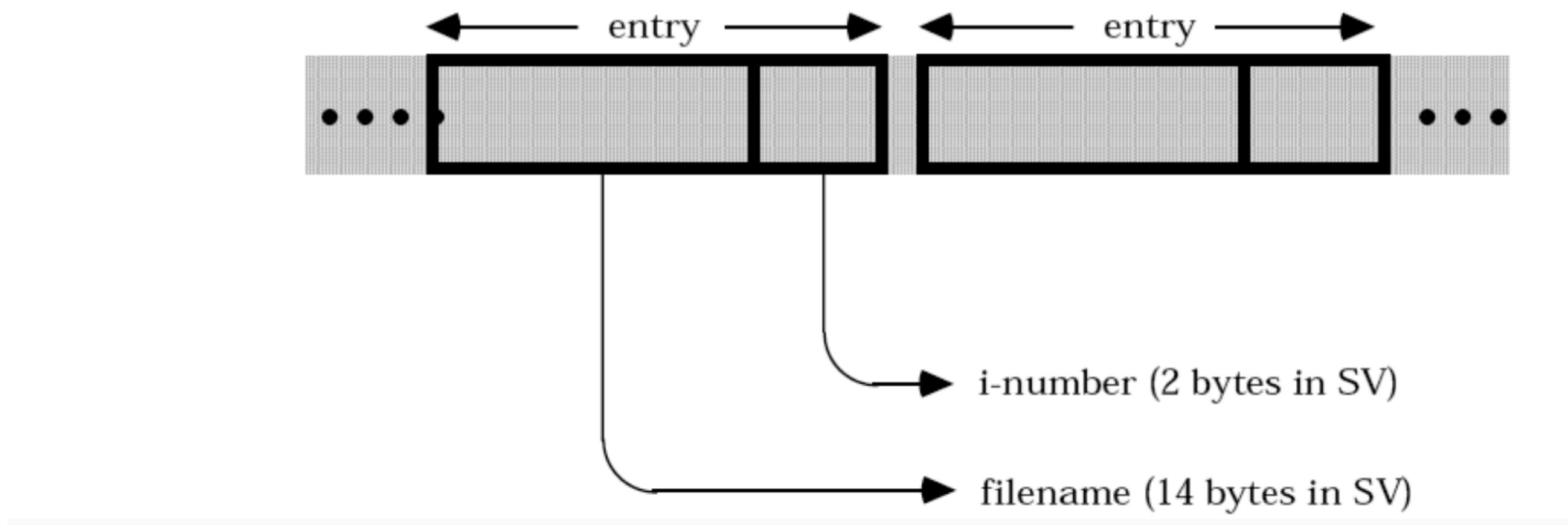
`dir/dir/dir/.../filename`

# Directories

Sono sequenze di bytes come i file ordinari

Differiscono dai file ordinari solo perché non possono essere scritte da programmi ordinari

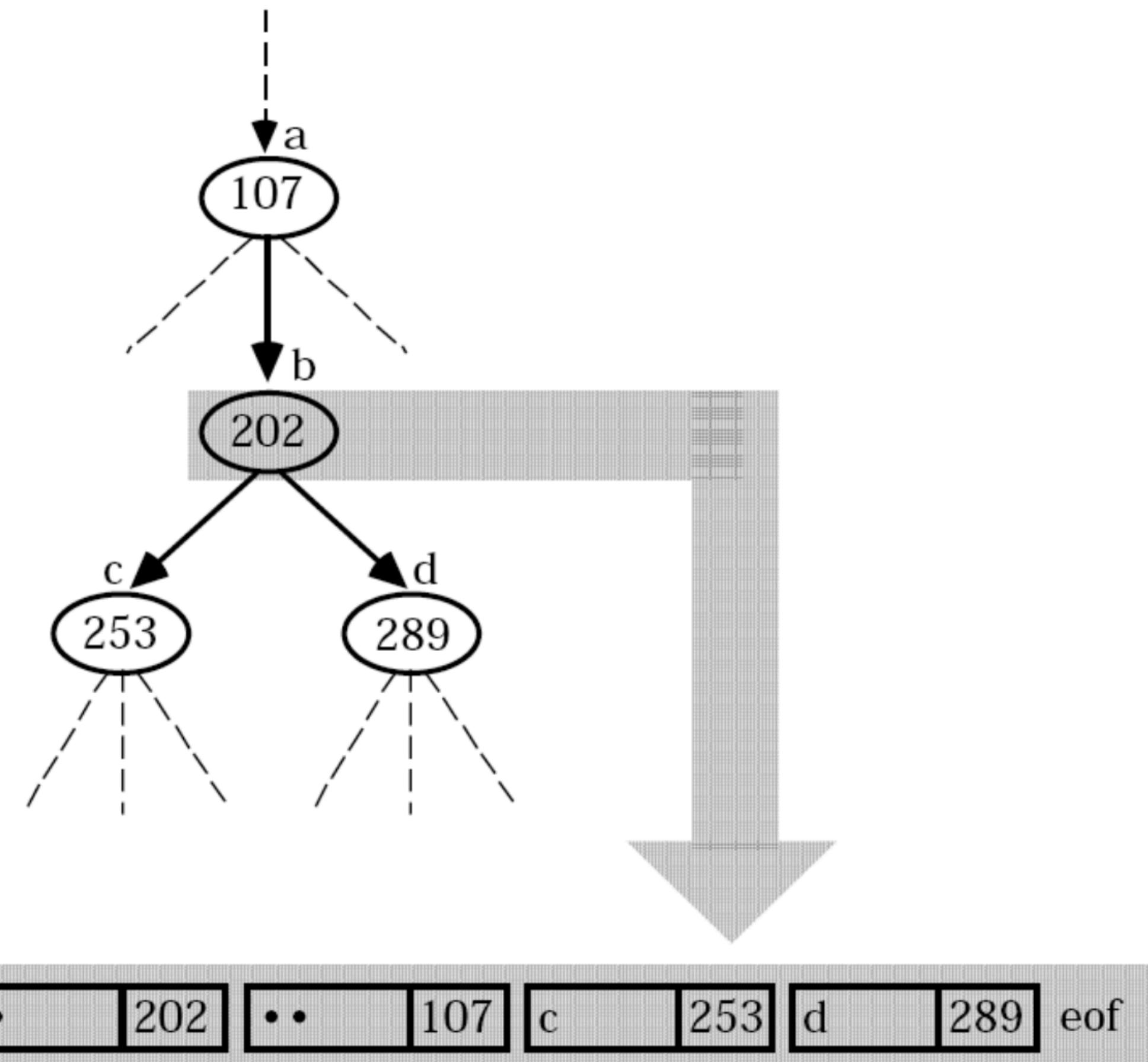
Il loro contenuto è una serie di **directory entries**: coppie formate da un nome di file e un i-number



# Directories

Almeno due entry in ogni directory:

1. la directory stessa “.”
2. la directory padre “..”



---

# Directories

---

- /bin (binary) comandi eseguibili
- /dev file speciali (I/O devices)
- /etc file per l'amministrazione del sistema, ad esempio:  
    /etc/passwd
- /lib librerie di programmi
- /tmp area temporanea usata dal sistema
- /home home directories degli utenti
- /usr Programmi, librerie, doc. etc. per i programmi user-related

---

# Working Directory

---

Ogni istanza della shell opera, ad ogni istante, su una directory corrente, o **working directory**

Subito dopo il login, la working directory è la **home** directory dell'utente

L'utente può cambiare la working directory con il comando **cd** (**c**hange **d**irectory)

---

# Home Directory

---

- Ad ogni utente viene assegnata dal system administrator una directory di base (**home directory**) che ha come nome lo username del proprietario
- Ad essa, l'utente potrà aggiungere file (o subdir)

Per denotare la propria home directory si può usare l'abbreviazione "~"

---

# Gestire le Directory

---

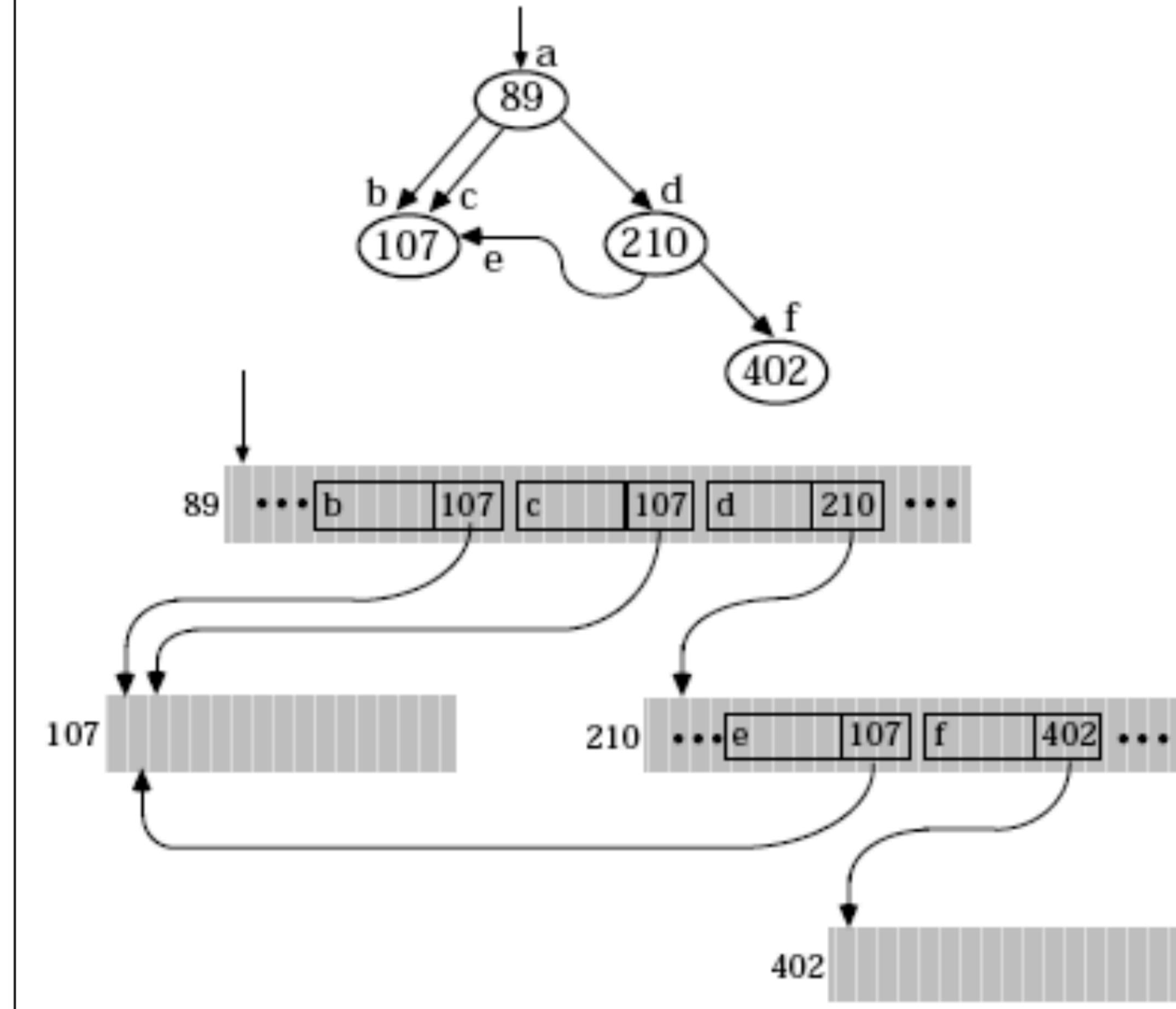
- **pwd** print working directory
- **cd** change directory
- **mkdir** make directory
- **rmdir** remove directory

# I Link

Un file può avere  
più filename  
(ma sempre un  
solo i-number)

Il file 107 ha 3 link

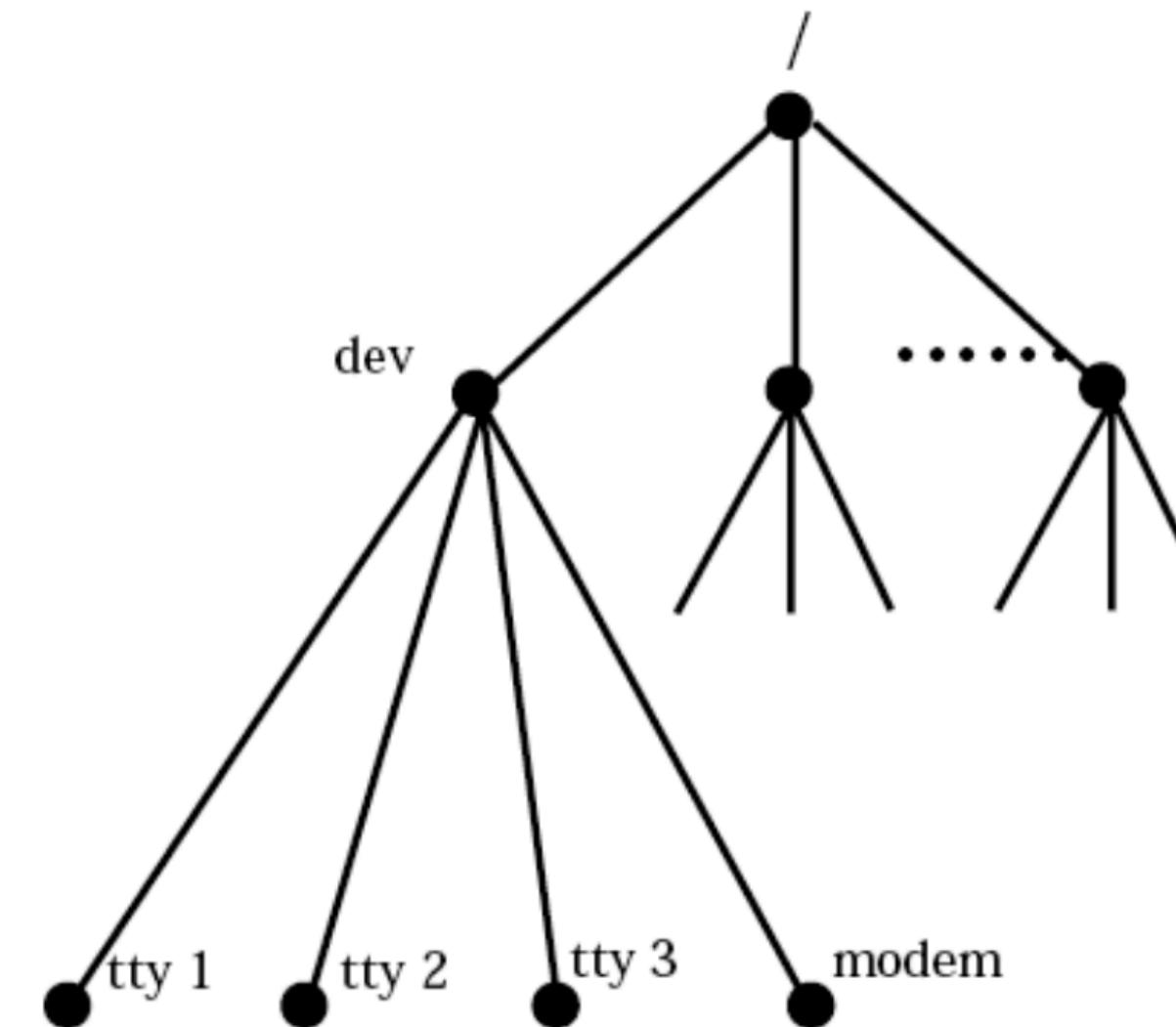
Esempio:



# File Speciali - I/O

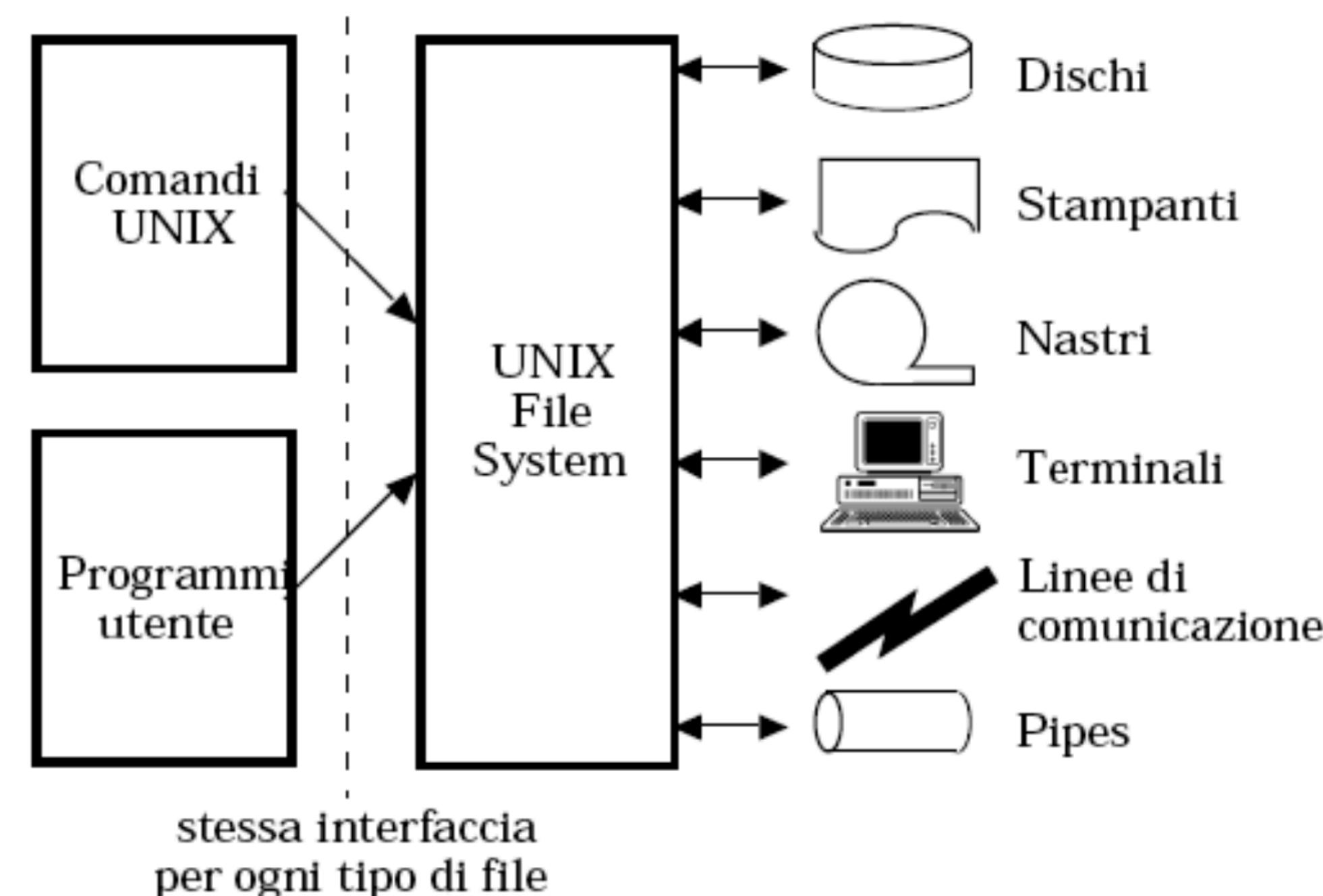
Ogni device di I/O viene visto, a tutti gli effetti, come un file (**file speciale**)

Richieste di lettura/scrittura da/a files speciali causano operazioni di input/output dai/ai devices associati



# File Speciali

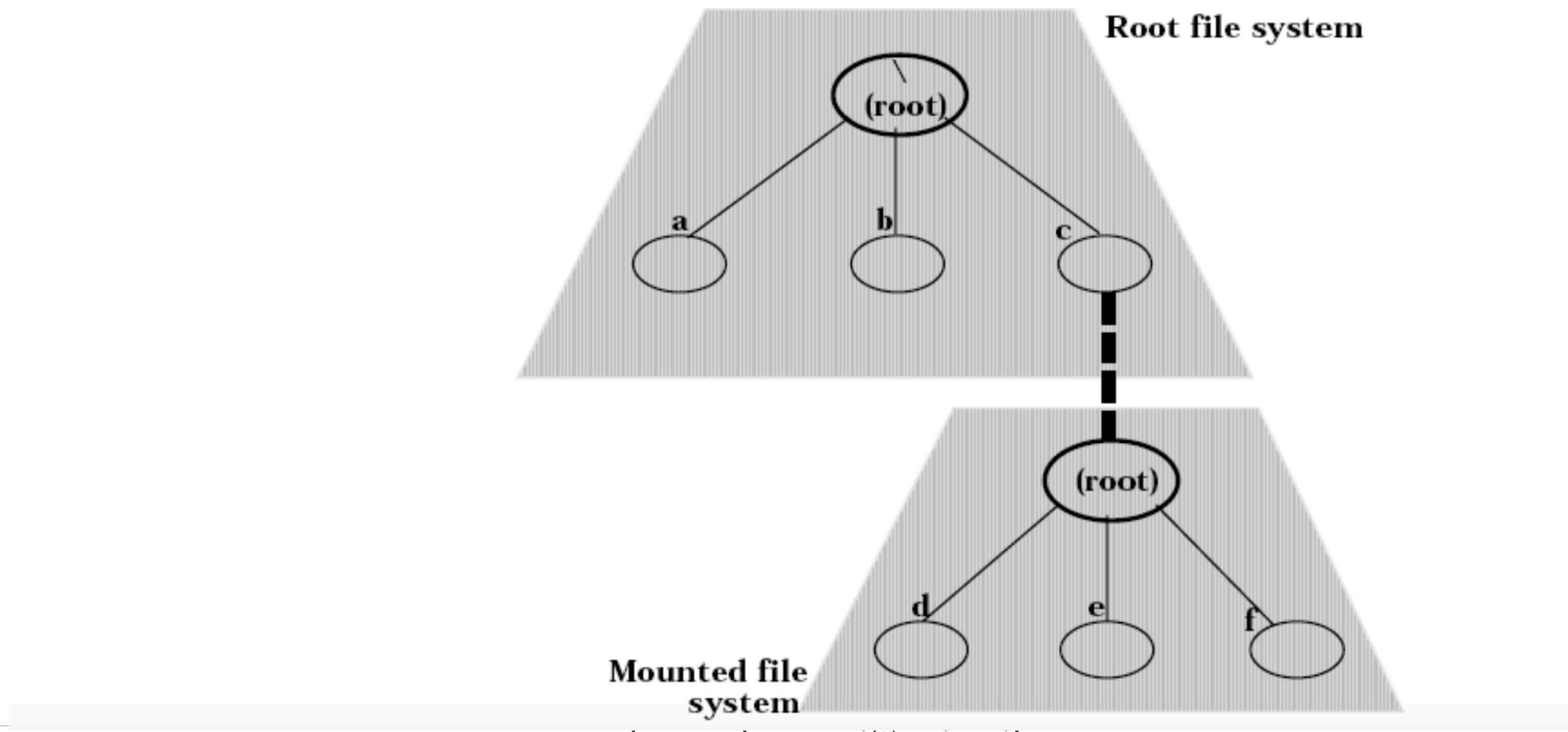
Trattamento uniforme di files e devices  
I programmi non hanno bisogno di sapere se operano su un file o su un device



# I File System Montabile

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 umount)



---

# Gestione delle Directory

---

**ls**    list directory  
**ln**    link

---

# I Comandi UNIX

---

**ls** [options] [directory1 file2 ...]

- Lista (in ordine alfabetico) il contenuto della o delle directories indicate
- Accetta anche nomi di file
- Senza parametri, elenca il contenuto della working directory
- Possiede numerose opzioni

---

# I Comandi UNIX

---

- **-l** (**long**) formato esteso con informazioni aggiuntive
  - **-a** (**all**) mostra anche i file “nascosti” (dotfiles)
- 
- **-R** (**Recursive**) visita ricorsivamente le sottodirectory
  - **-i** mostra l'i-number
  - **-t** (**time**) lista nell'ordine di modifica (prima il file modificato per ultimo)  
... e molte altre

# I Comandi UNIX

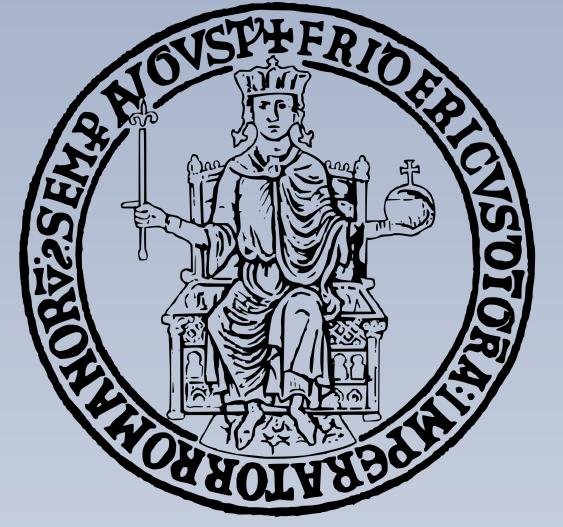
Totale dimensione occupata (in blocchi)							
		Riferimenti al file	Dimensione (byte)	Nome			
<code>lso:~&gt;ls -l</code>							
<code>total 12</code>							
<code>-rw-rw-r--</code>	1	lso	lso	10 Mar	4 13:29	a	
<code>-rw-rw-r--</code>	1	lso	lso	10 Mar	4 14:12	b	
<code>drwxrwxr-x</code>	2	lso	lso	4096 Mar	4 14:29	c	

Diagramma che illustra la struttura del comando `ls -l`. I campi sono etichettati come segue:

- Tipo**: Indica il tipo di file (d, l, c, b, -).
- Permessi**: Indica i permessi di lettura (r), scrittura (w) e esecuzione (x) per utente, gruppo e altri.
- Riferimenti al file**: Indica il numero di riferimenti al file.
- Proprietario**: Indica il proprietario del file.
- Gruppo**: Indica il gruppo del file.
- Dimensione (byte)**: Indica la dimensione del file in byte.
- Data ultima modifica**: Indica la data dell'ultima modifica del file.
- Nome**: Indica il nome del file.

Legenda per i tipi di file:

- (d)irectory
- (l)ink
- (c)haracter special file
- (b)lock special file
- (-) ordinary file



*Universitá degli Studi di Napoli Federico II*

# THANK YOU FOR YOUR ATTENTION

TRUST ME ...  
I AM A ROBOT!

