

Laboratorio di programmazione per sistemi ciberfisici

1. Introduzione

Enrico Martini
October 5, 2025

Organizzazione del corso

Informazioni generali:

- ▶ Il corso è diviso in due moduli:
 - ▶ Teoria: Bogdan Maris
 - ▶ Laboratorio: Enrico Martini
- ▶ Orario ricevimento (Laboratorio):
 - ▶ Alla fine di ogni lezione di laboratorio
 - ▶ Su appuntamento, vie e-mail: enrico.martini@univr.it

Materiale del corso

Manuale base:

- ▶ *Programmare in C* - Stephen G Kochan, Pearson 2020 (IV edizione)

Manuale intermedio/avanzato:

- ▶ *Programmare in C* - Marco Liverani, Società Editrice Esculapio

Piattaforma Moodle:

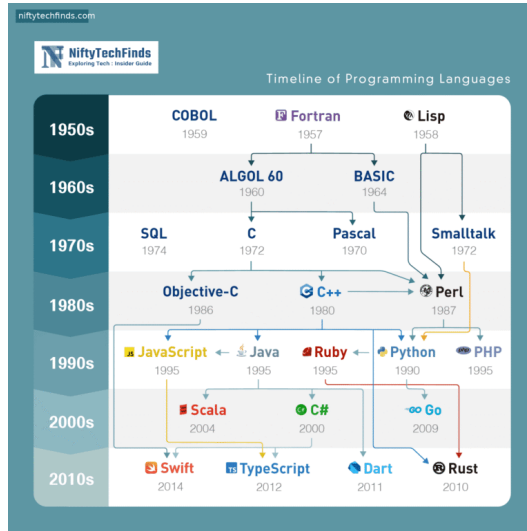
- ▶ Slides
- ▶ Esercizi Svolti
- ▶ Quiz Coderunner

Modalità d'esame

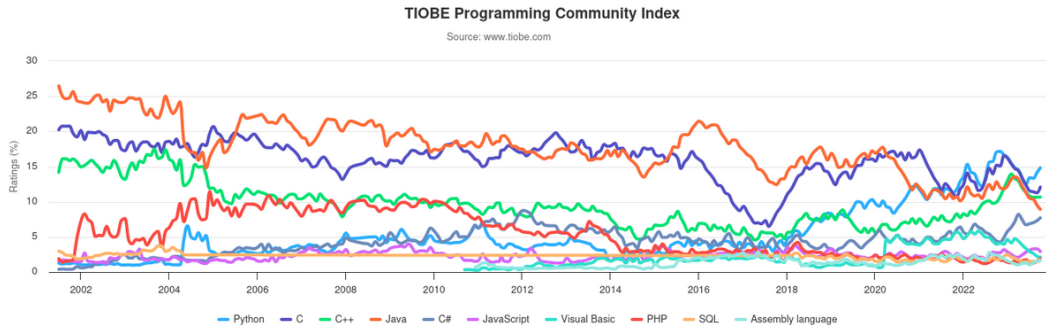
Prova congiunta con il modulo di teoria: 3 esercizi su *Quiz Coderunner*

- ▶ Criteri di valutazione:
 - ▶ Codice compilante
 - ▶ Codice risolutore del problema proposto
 - ▶ Codice ben indentato
 - ▶ Codice commentato
 - ▶ Codice ottimizzato

Perchè imparare il linguaggio C



Perchè imparare il linguaggio C



Sistemi di elaborazione

- ▶ **Software applicativo (SW)**: i programmi che eseguiamo
- ▶ **Sistema Operativo (OS)**: Ubuntu, Windows, etc
- ▶ **Firmware (FW)**: programma che gestisce le componenti fisiche
- ▶ **Componenti fisici della macchina (HW)**: CPU, GPU, RAM, etc

Linux

- ▶ Linux è un **sistema operativo** open source basato su Unix.
- ▶ Il suo “cuore” è il **kernel**, che gestisce risorse hardware e software.
- ▶ È usato in server, computer desktop, smartphone (Android), dispositivi IoT e supercomputer.
- ▶ Principali caratteristiche: gratuito, sicuro, stabile e altamente personalizzabile.



Ubuntu

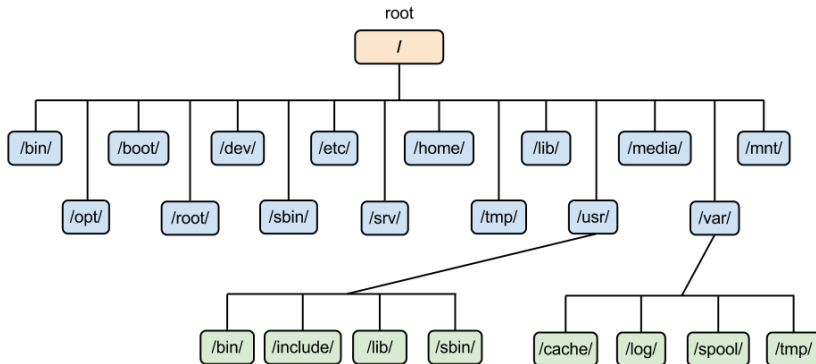
- ▶ Ubuntu è una distribuzione (distro) di Linux, cioè una **versione completa del sistema operativo** basata sul kernel Linux.
- ▶ Include interfaccia grafica, programmi preinstallati e aggiornamenti frequenti.



File System

- ▶ **File**: contenitore logico di dati o istruzioni
- ▶ **Directory**: contenitore logico di più file
- ▶ **Collegamenti**: riferimento ad oggetti nel file system

Struttura del File System



La Shell Bash (Terminale)

Interfaccia del Sistema Operativo con l'utente

- ▶ Parte del sistema operativo
- ▶ E' un interprete di comandi

```
zaira@zaira:~/shell-tutorial$  
zaira@zaira:~/shell-tutorial$  
zaira@zaira:~/shell-tutorial$ ./greetings.sh Zaira|
```

Comandi Di Navigazione

```
1  pwd                # mostra la cartella in cui si trova l'utente
2  mkdir <nome>       # crea una nuova cartella
3  rmdir <nome>       # elimina la cartella (solo se vuota)
4
5  cd <percorso>      # sposta l'utente al percorso specificato
6  cd ..             # sposta l'utente alla cartella superiore
7  cd /              # sposta l'utente alla cartella base (root)
8  cd ~              # sposta l'utente alla cartella base dell'utente
9
10 ls <percorso>      # elenca i file contenuti nella cartella <percorso>
11 ls                # elenca i file contenuti nella cartella corrente
```

Comandi Di Gestione File

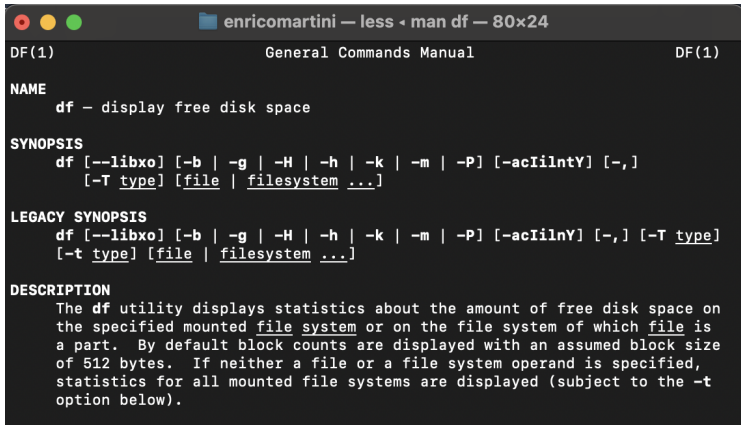
```
1 cp <sorgente> <destinazione>      # copia file/directory
2 mv <sorgente> <destinazione>      # sposta file/directory (oppure rinominare)
3 rm <percorso>                      # cancella file
4 touch <filename>                   # crea un file vuoto
5 cat <filename>                     # stampa il contenuto del file
6 locate <filename>                  # trova il file nel sistema
```

Comandi Generici

```
1 echo <testo>      # stampa sul terminale il testo
2 man <comando>     # documentazione sul comando e le opzioni
3 du                # mostra le statistiche sull'utilizzo (-H consigliato)
4 df                # mostra lo spazio libero su disco (-H consigliato)
5 history           # mostra la cronologia dei comandi usati
6 clear             # pulisce lo schermo del terminale
```

Esempio

```
1 man df
```



```
enricomartini — less ◀ man df — 80x24
DF(1)                                General Commands Manual                                DF(1)

NAME
    df — display free disk space

SYNOPSIS
    df [--libxo] [-b | -g | -H | -h | -k | -m | -P] [-acIilntY] [-,]
      [-T type] [file | filesystem ...]

LEGACY SYNOPSIS
    df [--libxo] [-b | -g | -H | -h | -k | -m | -P] [-acIilnY] [-,] [-T type]
      [-t type] [file | filesystem ...]

DESCRIPTION
    The df utility displays statistics about the amount of free disk space on
    the specified mounted file system or on the file system of which file is
    a part. By default block counts are displayed with an assumed block size
    of 512 bytes. If neither a file or a file system operand is specified,
    statistics for all mounted file systems are displayed (subject to the -t
    option below).
```


Editor di testo

Un editor di testo è un programma che permette di creare e modificare file di testo semplice (senza formattazione).

Da terminale:

- ▶ vim
- ▶ nano

Grafici (GUI):

- ▶ gedit
- ▶ vs code

Vim

Come lanciarlo da terminale:

```
1 vim <filename>
```

Ha 5 comandi base:

- ▶ i entra in modalità inserimento
- ▶ ESC esci dalla modalità d'inserimento
- ▶ :w salva il file
- ▶ :q chiudi il file
- ▶ :wq salva e chiudi il file

Nano

Come lanciarlo da terminale:

```
1 nano <filename>
```

Ha 2 comandi base:

- ▶ Ctrl+o salva il file
- ▶ Ctrl+x chiudi il file

Script Bash

File di testo che contiene una **sequenza di comandi** da eseguire nel terminale, utile per automatizzare operazioni ripetitive.

Esempio:

```
1 #!/bin/bash
2 echo "hello world!"
3 echo "hello world!"
4 echo "hello world!"
```

Una volta salvato e chiuso il file, da terminale eseguire lo script:

```
1 bash <nome-script>
```

Esercizio 1

1. Crea una cartella chiamata `es1` nella **Scrivania**
2. Entra nella cartella e crea un file vuoto `file1.txt`
3. Rinomina il file in `file2.txt`
4. Cancella il file `file2.txt`
5. Torna sulla Scrivania
6. Cancella la cartella `es1`

Esercizio 2

1. Crea una cartella chiamata `es2` nella **Scrivania**
2. Entra nella cartella e crea 3 file vuoti `a.txt`, `b.txt`, `c.txt`
3. Rinomina il file `c.txt` in `d.txt`
4. Inserisci il testo "*hello*" all'interno del file `a.txt` con vim
5. Inserisci il testo "*ciao*" all'interno del file `b.txt` con nano
6. Visualizza il testo contenuto all'interno del file `a.txt`
7. Stampa la data, il giorno e l'ora attuali utilizzando il comando `date`

Esercizio 3

Crea, utilizzando un editor a tua scelta, uno script bash nella cartella **Scrivania** chiamato `script.sh`, che stampa la seguenti frasi:

```
1 Ciao Mondo!  
2 Hello World!  
3 Oggi è Mar  7 Ott 2025 10:17:15 CEST
```

NB: per inserire l'ora precisa, utilizza il comando `$(date)`

Esercizio 4

Crea, utilizzando un editor di testo a tua scelta, uno script bash che risolva l'esercizio 2.