

Accademia BBAA Albertina di Torino, Torino

Corsi LM e PM, a.a. 2025-25 - Docente Mario Chiari

Basic whereabouts per gli insegnamenti ABTEC40/AFAM095 - T/B

A – CONOSCERE LA VOSTRA MACCHINA

La vostra macchina è il vostro strumento di lavoro e di produzione, dovete averne cura e conoscenza professionale.

Consulta <https://en.wikipedia.org/> per tutte le nozioni di sotto indicate.

Per tutto ciò che riguarda l'informatica - soprattutto per le delucidazioni di base di cosa si intenda con questa o quella nozione - le voci di wikipedia sono in genere chiare, esaustive e attendibili. Molto, molto, molto meglio la versione inglese!

1. Delle principali componenti **hardware** della tua macchina, devi sapere individuare nome, azienda produttrice, caratteristiche principali:

- **cpu**,
- **gpu**,
- **ram**,
- **scheda madre**,
- **monitor**,
- **hard disk**,
- ecc.

Per chi ha un vecchio desktop a casa, apritelo e curiosate, e individuate le diverse componenti (*non aprite il vostro laptop!*).

2. Sai individuare il **sistema operativo** - e la versione - installato sulla tua macchina (e anche quello del tuo smartphone)

As of September 2024, [Android](#) is the most popular operating system with a 46% market share, followed by [Microsoft Windows](#) at 26%, [iOS](#) and [iPadOS](#) at 18%, [macOS](#) at 5%, and [Linux](#) at 1%. Android, iOS, and iPadOS are mobile operating systems, while Windows, macOS, and Linux are desktop [e laptop] operating systems.

I sistemi operativi che ci possono interessare sono o della famiglia Windows (Microsoft) o derivati Unix. Le macchine della Apple usano oggi macOS, una versione di Unix.

Se siete interessati (*lo dovrete essere!*), scoprite cosa sia [Linux](#) e cosa si intende per [distribuzione Linux](#) (le più note sono [Fedora](#), [Ubuntu](#), [Debian](#), [Mint](#), ma ce ne sono molte altre). Capite come si usa gergalmente il termine **piattaforma**.

Per saperne di più:

- su [Microsoft Windows](#). Ricordatevi anche di [MS-DOS](#) (MicroSoft Disk Operating System) e della sua storia.
- su [Unix](#). Fate conoscenza con [Dennis Ritchie](#), uno dei creatori dell'era digitale. Unix e derivati sono una galassia di sistemi, vedi per esempio una [Unix Timeline](#);
- su [Linux](#), e il suo creatore [Linus Torvalds](#);
Cercate di capire il ruolo delle diverse componenti di seguito indicate: *A typical Linux distribution comprises a [Linux kernel](#), [GNU tools and libraries](#), additional software, documentation, a [window system](#) (the most common being the [X Window System](#)), a [window manager](#), and a [desktop environment](#).*
- per le macchine della Apple, vedi [MacOS](#), e prendi nota che: *macOS originally named Mac OS X until 2012, then rebranded to OS X until 2016*. Ricorda anche che fino a circa il 2001, le macchine Apple erano dotate di un sistema operativo, di proprio design, denominato [Classic Mac OS](#), il cui successo fu dovuto soprattutto alla correlata GUI.

Avvertenza: Se hai un computer della Apple, devi sapere come Apple distingue le [diverse versioni](#) e quale sia la tua.

3. Distingui tra sistema operativo e l'**interfaccia grafica utente** [GUI](#) (*Graphical user interface*)

- per le recenti versioni del sistema operativo Windows, la distinzione OS/GUI è attualmente difficile da formulare perchè *Microsoft* ha via via integrato sempre più OS e GUI. Storicamente Microsoft è nota per *MS-DOS* (OS, dal 1981 ai primi anni 2000) e per *Windows* (dal 1984 ad oggi), quest'ultima inizialmente solo una GUI, ma poi via ha assunto il ruolo di un Sistema Operativo Grafico. La storia in verità è un poco più complicata, se interessa leggi [Microsoft Windows](#) e [MS-DOS](#).
- per le macchine della azienda Apple, la GUI è chiamata [Aqua](#) (vedi anche [qui](#)).
- infine per le distribuzioni Linux, la GUI è un prodotto ben distinto dal sottostante SO, ve ne sono diverse, sviluppate da comunità diverse di sviluppatori. Tra le più note [Gnome](#) e [KDE](#), ma ve ne sono diverse altre.

Per rendersi conto di come le cose sono evolute velocemente, vedi questa [storia](#) visuale. E [questa](#) è un'altra interessante risorsa storica.

4. Impara ad aprire il terminale (o *prompt* dei comandi), sia nei sistemi Microsoft (un tempo nella cartella Accessori, oppure con il comando *cmd*) sia nei Mac (in genere

cercando il *terminal*), sia nella varie distribuzioni Linux.

Imparate i comandi principali disponibili in Dos/Win e in Linux (+macOS):

Table C-1. Similar Commands

Command's Purpose	MS-DOS	Linux	Basic Linux Example
Copies files	copy	cp	cp <i>thisfile.txt</i> /home/ <i>thisdirectory</i>
Moves files	move	mv	mv <i>thisfile.txt</i> /home/ <i>thisdirectory</i>
Lists files	dir	ls	ls
Clears screen	cls	clear	clear
Closes prompt window	exit	exit	exit
Displays or sets date	date	date	date
Deletes files	del	rm	rm <i>thisfile.txt</i>
"Echoes" output on the screen	echo	echo	echo <i>this message</i>
Edits files with simple text editor	edit	pico[a]	pico <i>thisfile.txt</i>
Compares the contents of files	fc	diff	diff <i>file1 file2</i>
Finds a string of text in a file	find	grep	grep <i>this word or phrase</i> <i>thisfile.txt</i>
Formats a floppy	format a: (if floppy's in A:)	mke2fs (or mformat[b])	/sbin/mke2fs /dev/fd0 (/dev/fd0 is the Linux equivalent of A:)
Displays command help	command /?	man[c]	man <i>command</i>
Creates a directory	mkdir	mkdir	mkdir <i>directory</i>
Screens through a file	more	less[d]	less <i>thisfile.txt</i>
Renames a file	ren	mv	mv <i>thisfile.txt</i> <i>thatfile.txt</i> [e]
Shows your location in the file system	chdir	pwd	pwd
Changes directories with a specified path (<i>absolute path</i>)	cd <i>pathname</i>	cd <i>pathname</i>	cd / <i>directory/directory</i>
Changes directories with a <i>relative path</i>	cd ..	cd ..	cd ..
Displays the time	time	date	date
Shows amount of RAM and use	mem	free	procinfo

Notes:
a. Pico is a simple text editor; other editors you can use in place of pico include emacs and vi.
b. This formats a disk for the DOS filesystem.
c. Or you can use **info** for some commands.
d. You can also use another *pager*, called **more**, to scroll through a file a screen at a time.
e. The **mv** command serves double-duty, because it can both move a file and, if you want to rename a file in the same directory, you "move" that file to the same directory with a new name, as in this example.

tabella da [qui](#)

Imparate come minimo a navigare l'albero delle cartelle e file con *cd* (si usa sia in Dos sia in Unix), e ricordatevi che da quale Directory (cartella) lanciate un comando può essere importante.

B – INTERNET

5. Cerca di capire l'idea di base della comunicazione telematica Internet, per iniziare la distinzione:

Commutazione di circuito vs. Commutazione di pacchetto

Nelle voci wikipedia, leggi i paragrafi iniziali, per capire di che cosa si stia parlando, il resto sono nozioni per ingegneri della tele-comunicazione. Sappi spiegare – in termini elementari - la differenza tra Circuito e Pacchetto.

Domanda: di quale precedente infrastruttura tecnologica Internet è stata l'evoluzione? Cercate di confrontare la struttura della comunicazione Internet con quelle fornite dal sistema postale, da radio/televisione, dalla telefonia e dalla stampa quotidiana: differenze e similitudini.

Qui trovi la storia di Internet [scritta dai protagonisti-inventori](#) (pdf).

Quando la tua macchina è collegato in rete, controlla l'indirizzo IP che gli viene assegnato con il comando *ipconfig* (win) o *ifconfig* (linux): apparirà - tra altre informazioni - l'indirizzo IP, una stringa composta da quattro numeri, ciascuno tra 1 e 254, con cui la vostra macchina è identificata nella LAN, o in Internet; dovrebbe essere qualcosa del tipo 192.168.50.123.

6. Cerca la definizione (per cosa sta l'acronimo) e le caratteristiche dei principali **protocolli** Internet:

- [TCP-IP](#)
- [SMTP](#)
- [FTP](#)
- [HTTP \(HTML\)](#) ← *questo è il protocollo per trasmettere pagine Web*

P è sempre per *Protocol*; **T** è per *Transfer*. Controlla di ciascuno la data di quando sono stati formulati e divenuti di uso comune.

Consultate www.ietf.org, ente dedicato alla sviluppo protocolli Internet. Questo è per informatici, ma è sempre utile vedere cosa si intende per protocollo, per esempio vedi l'ultima versione della formulazione del [protocollo per la posta elettronica](#). Lo IETF pubblica dei documenti indicati come RFC (*Request For Comments*), che contengono la descrizione dei protocolli (quelli proposti da qualcuno, e quelli adottati secondo una complessa procedura)

Per esempio nella RFC2616, si trovano le definizioni di cosa si deve intendere per *client* e per *server*:

client

A program that establishes connections for the purpose of sending requests.

server

An application program that accepts connections in order to service requests by sending back responses. Any given program may be capable of being both a client and a server; our use of these terms refers only to the role being performed by the program for a particular connection, rather than to the program's capabilities in general.[...]

7. DNS

C – TROVARE INFORMAZIONI, RISORSE e TOOLS IN RETE **(prima della AI – da valutare come le cose stanno cambiando)**

8. Risorse, motori di ricerca, directory, ecc.

- uso di motori di ricerca, ovviamente soprattutto **Google** (ma ce ne sono altri)
Vi consiglio di cercare in Inglese, ovvero impostate Google indicando *Inglese* come lingua, e *Stati Uniti* come area geografica. Inoltre, utilizzate *search word* chiare e tecnicamente appropriate.
- **Wikipedia** Rimane un progetto importante, anche se con aspetti discutibili.
Se vi ritenere competenti in un qualche settore, potreste tentare di registrarvi per scrivere un paio di voci di wikipedia, e fare pratica delle modalità di scrittura, revisione, discussione di wikipedia.
- **www.w3.org** Sviluppo protocolli Web e tecnologie collegate. Estremamente ricco, ci vuole un po' per imparare ad orientarsi, e trovare le url dei tutorial che possono servire, alcuni anche elementari e molto ben scritti;
- **www.php.net** Sito ufficiale del linguaggio PHP. Documentazione molto utile anche per iniziare con piccoli esempi <http://www.php.net/manual/en/>;
- **www.mysql.com** Sito ufficiale di MySQL, documentazione a <http://dev.mysql.com/doc/>;
- **www.mozilla.org/foundation/** *The Mozilla Foundation*, sviluppatori di Firefox, e di molto altro. Per noi utili le sezioni su **JavaScript**, e sul **DOM**;
- **stackexchange** Forse meno noto, ma estremamente utile: un sito estremamente performante che organizza comunità on line di Q&A (*questions and answers*). Dovete iscrivervi a una delle comunità tematiche, e poi porre domande o suggerire risposte. Molto interessante il meccanismo di calcolo della vostra reputazione. (Vi dovete iscrivere)
- **w3schools** Uno degli innumerevoli siti con tutorial, guide, lezioni interattive, video, podcast, ecc. su tutto ciò che stiamo cercando di capire, di qualsiasi livello, dall'introduttivo all'*advanced*. Cercate pure in rete, e divertitevi.
- **awesome-creative-coding**
Incredibilmente ricca e utilissima directory di tutto ciò che compone il variegato mondo del Creative Codings. Un ottimo punto di partenza per scoprire tools, intro, materiali, ecc.. **Da consultare.**

D – CAVI E PRESE

banale ma utile e necessario

9. Competenze pratiche banali e date per scontate, ma necessarie e utili: dovete saper

collegare la vostra macchina alla rete elettrica di alimentazione, alla presa **Ethernet** per accedere a Internet, all'accesso wi-fi (e al hot spot), e ai proiettori disponibili nell'aula, e quindi saper distinguere uscite/prese e cavi corrispondenti. Semplifica la vita conoscere gli standard più comuni per cavi e prese: **consultate [qui](#) e [qui](#)** (o simili risorse in rete)

Ricordatevi: per le presentazioni durante il corso, **e in sede di Esame**, siete voi a dovervi preoccuparvi di presentarvi con cavi e spinotti per collegare la Vostra macchina al proiettore messo a disposizione in aula

NOTA: chi ha macchina Apple, con uscite non standard, deve munirsi (comprare!) i necessari connettori.

E – UN POCO DI GERGO

10. Per ognuno dei seguenti **acronimi**, consultare la corrispondente [voce di wikipedia](#):

- **CPU, GPU, RAM**
- **OS, GUI, CLI, BIOS**
- **PSTN**
- **ITU, ISO/OSI, IETF, ICANN, RIR, ISOC, RFC**
- **TCP-IP, FTP, SMTP, HTTP-HTML, CSS, DNS**
- **URL, URI**
- **LAN, WAN**
- **PHP**
- **SQL**
- **(L)(M)(W)AMP**
- **GNU GPL, CC**
- **IDE, WYSIWYG**
- **DVI, HDMI, VGA, USB, Ethernet, RCA, DisplayPort,**
- **.... (altri??)**

Via via impar ad usare le seguenti nozioni in modo appropriato, comprendendo i rispettivi ambiti semantici:

- **Protocollo**
- **Standard**
- **Algoritmo**
- **Programma**
- **Linguaggio (di programmazione)**
- **(pacchetto) Applicativo**
- **Macchina (computazionale)**

(controllate come si traducono questi termini nella vostra lingua madre e in Inglese)

Fate attenzione all'uso corretto di alcuni **termini italiani** (alcuni forse desueti, o poco usati), consultate un buon dizionario, e le seguenti voci Treccani: [Informatica](#) - [Telematica](#) - [Cibernetica](#) (io non lo uso)

F – COME RENDERE LA VOSTRA MACCHINA OPERATIVA

11. Installa un *software bundles* (L)(M)(W)AMP, dove

- **(L)(M)(W)** = (Linux)(Mac)(Windows), il vostro sistema operativo;
- **A** = Apache, il web (http) server;
- **M** = mySQL, il programma di gestione database;
- **P** = l'interprete di PHP. PHP è il linguaggio di programmazione orientato al web.

Per ***software bundle*** (o *suite*) si intende in genere un pacchetto di applicativi composto da più pacchetti indipendenti (per produzione, funzionalità e possibilità d'uso) ma che sono di fatto interdipendenti nei loro utilizzi più comuni e che quindi merita installare in modo coordinato. Un buon *software bundle* comprende inoltre qualche applicativo ausiliario, in genere per installare e/o configurare i pacchetti principali, per il loro utilizzo e monitoraggio, nonché una *pulsantiera* (*pulsantiera* un termine non tecnico, che uso tanto per intenderci) utile per lanciare o chiudere e i diversi tool (talvolta detti servizi) principali (e quindi non confondete il lanciare la pulsantiera stessa con il lanciare/accendere i servizi principali). Ogni *bundle* è in genere sviluppato per uno specifico SO, ovvero devi installare la versione adatta al tuo SO, e anche al tuo hw, se 32bit o 64bit.

(Se siete interessati: provate a installare Apache, mySQL e PHP separatamente, e vedrete il numero di difficoltà da superare).

Di seguito elenco i *bundle* AMP che ho visto usare negli ultimi anni (non tutti per ogni SO):

- MAMP, www.mamp.info
Forse il più popolare, ora anche versione per Win: MAMP per Win
- XAMPP, <http://www.apachefriends.org/en/xampp.html>
- Deserver -- easyPHP, <http://www.easyphp.org/>
- WAMPServer, <http://www.wampserver.com/en/>
- AMPPS <http://www.ampps.com/>
- *altri??*

Se usate una distribuzione Linux, i programmi AMP dovrebbero essere già presenti. Anche su Windows e Mac potrebbero essere già installati, ma installare una opportuna *bundle* è più pratico (per voi e per me).

In teoria, potete avere più installazioni di AMP sulla vostra macchina, ma nel caso dovete fare un poco di attenzione a che non entrino in conflitto (in genere basta una qualche attenzione alle porte su cui sono in ascolto).

12. Installa il browser (http-client) Firefox, [versione developer](#) (Fortemente Consigliato!)

+ esplorate gli **add-on** (componenti aggiuntivi), dal menù Strumenti/*Tools* di Firefox, tra cui:

- **WebDeveloper**, estremamente utile (integrato nelle versioni recenti)
- **ColorZilla**, per gestire i colori, utile e divertente
- **Nimbus**, per prendere screenshot di una pagina web
-

13. installa un *Syntax Highlighter Editor* (editore di codice)

Inoltre avete bisogno di un tool che **highlight** il codice mentre lo scrivete, ovvero che colori le diverse categorie sintattiche di cui sono composti i linguaggi che imparerete a conoscere (html+css, php, mysql, JS) con colori diversi. Per quanto apparentemente banale, questo permette una scrittura estremamente più facile e veloce, ed aiuta anche a capire come sono strutturati i diversi linguaggi. I *Syntax Highlighter Editor* sono in genere diversi per OS diversi (alcuni hanno versioni multiplatforma), in particolare potete usare:

- **BEdit** per OSX (Apple)
- **Notepad++** per Win
- **Gedit**, per distribuzioni Linux.
- **Atom** (multiplatforma). Forse il migliore, uso meno intuitivo.
- **Visual Code** (multiplatforma). Forse il più performante.

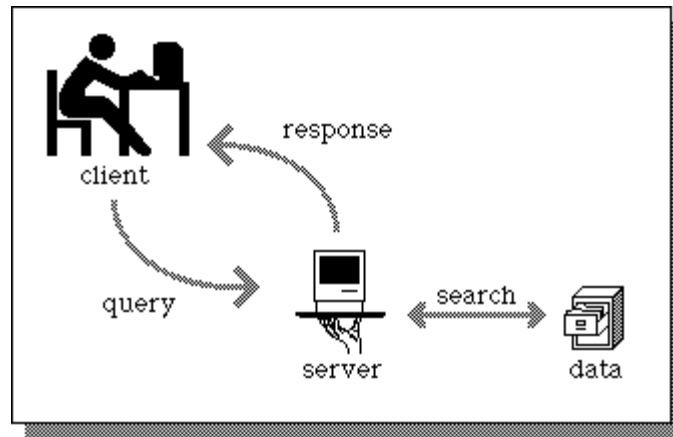
In alcuni di questi, sono integrati strumenti di organizzazione di famiglie di script in progetti, strumenti di compilazione e debugging, e oggi di consultazione motori AI.

G – CLIENT – SERVER

IMPORTANTE E SEMPLICE

14. Per capire alcuni aspetti della programmazione orientata al web, è essenziale capire i

ruoli di *client* e di *server* tra entità che comunicano, in genere rispetto a un certo protocollo di comunicazione. Quindi, per ogni protocollo, dovete saper individuare e operare con gli applicativi che svolgono i ruoli di *client* e rispettivamente di *server*, in particolare per almeno i seguenti protocolli: **http-html**, **ftp**, **smtp**. Per il corso, il caso importante è ovviamente il primo, ma potremmo utilizzare anche un client ftp.



Attenzione: le entità che comunicano, nel ruolo di *client* o di *server*, sono in genere collocate su macchine diverse, a indirizzi diversi della rete Internet, in località geografiche diverse, come viene raffigurato nel disegno sopra. Ma le entità che comunicano sono programmi (applicativi) installati su una macchina, non macchine (non confondere). **E quindi nota bene** – un programma *client* e un programma *server* possono comunicare tra di loro anche se sono installati sulla stessa macchina! È importante capire come far interagire tale coppie di programmi client/server su una singola macchina, perché questo è quello che dovremo fare.

(Nota: nel disegno è mostrato anche il terzo personaggio che incontreremo spesso, l'archivio dati, o *database* (db), qui raffigurato da uno schedario. Il db può trovarsi su una macchina dedicata, collegata alle altre, oppure su la stessa macchina degli altri.)

Si utilizza il termine *browser* per indicare un qualche *http client*. Gli applicativi che sono dei client http sono i noti *Firefox*, *IE*, *Safari*, *Opera*, *Chrome*, *SeaMonkey* e altri meno noti. Il *server http* più diffuso e più noto (ma non l'unico) è *Apache*.

Quindi: da un punto operativo dovete assai velocemente imparare a distinguere tra operare lato client e operare lato server: Devi aver ben chiaro la distinzione tra operare lato Server e operare lato Client dopo aver installato la suite AMP controllate che tutto funzioni e che avete capito come muoversi tra il lato client e il lato server. Imparate la seguente procedura:

- a) Individuate la cartella nel vostro hd in cui porre tutto ciò che costituirà il vostro

sito (nei Wins, viene in genere creata la cartella **www**; con MAMP in genere la cartella **htdocs** (nei Mac apribile da Applicazioni, sotto la cartella MAMP); nel mio Linux Fedora, la trovo al percorso `/var/www/html/`). Per intenderci chiameremo tale cartella **cartella server**. Quando modificate il contenuto di questa cartella state agendo lato Server.

- b) Lanciate il servizio Apache (preferibilmente dalla pulsantiera della vostra bundle; in alcuni bundle apparirà una spia verde=acceso, o un checkbox spuntato)
- c) Create un file `primo.html` con le seguenti righe di codice html e salvatelo nella cartella server:

```
<html>
<head>
<title> mio primo file </title>
</head>
<body>
<p> Hello <b>World</b>! </p>
</body>
</html>
```

(**Attenzione**: se usate Win: prestate attenzione a se il file è salvato con estensione esattamente .html: Win ha spesso delle configurazioni di salvataggio automatico che aggiungono estensioni non volute e inoltre non sempre visibili (da qualche parte potete settare Win che mostri sempre le estensioni dei file).

- d) Dalla barra di navigazione del vostro browser, interrogate/chiamate il vostro server con la URL (indirizzo Internet) <http://127.0.0.1/primo.html>, o (nome di dominio) <http://localhost/primo.html>.
Se vedete il testo: Hello **World!** (ma non le tag <p>,), allora il tutto funziona. OTTIMO!

In alcuni casi dovete indicare anche la porta di connessione, per es. <http://localhost:8888/primo.html>

Quando, con un browser, eseguite una tale chiamata http, state agendo lato Client.

- e) **Attenzione**: se invece digitate nella barra di navigazione qualcosa tipo <file:///percorso--a/www> allora state utilizzando il vostro browser semplicemente come visualizzatore di codice html, senza connessione al server.
Se confondete questi due casi, lo considero un errore grave!

H – MULTIMEDIALITÀ

15. In genere, tendo a non usare il termine *multimedialità*, che considero in parte obsoleto in parte poco significativo. Consiglio però di tenere presente e consultare il seguente testo, con una varietà di informazioni tecniche assai utili:

Fundamentals of Multimedia, 3rd ed.

by Ze-Nian Li, Mark S. Drew, Jiangchuan Liu

<https://www.cs.sfu.ca/mmbook/>

(se cercate, al sito per ogni sezione del libro vi sono link a ulteriori risorse e informazioni)

