

Cap 0 ("da studiare come un capitolo di valenza 0" cit.) 1-2-3-5 (da pag 183?) 8 (fino a pag 271)

## MODULO I

### CAP.0

INFORMAZIONE = dati + istruzioni

Attenzione verso gli strumenti: i contenuti definiscono la qualità dell'informazione.

Il calcolatore risponde al principio di astrazione (o di information hiding), sistema che nasconde agli utenti gran parte dell'informazione e il suo funzionamento può essere imparato tramite un processo top-down (dall'alto verso il basso). Il calcolatore digitale è un sistema programmabile, che esegue molteplici successioni di istruzioni (programmi o livello software) senza dover modificare la struttura fisica (livello hardware).

Per elaborare l'informazione esistono degli strumenti:

- 1) Formali = consentono di trattare entità di informazione dotate di una certa struttura, ricavandone conclusioni per via elaborativa
  - 2) Materiali = "materializzando" le entità di informazione su cui si opera, ne facilitano l'elaborazione
- Inizialmente i calcolatori erano usati come strumenti di supporto per la memorizzazione, poi furono inventate le calcolatrici meccaniche. In seguito furono create schede con dei fori, che corrispondevano alle operazioni da compiere (programma).

L'esigenza di un dispositivo per elaborare l'informazione, di general purpose (applicabilità generale) creò la "macchina di Von Neumann": i dati e le istruzioni possono essere codificati in uno stesso formato, così da poterli memorizzare entrambi in un unico dispositivo.

Essa è costituita da:

- 1) Interfaccia = interazione del calcolatore con l'ambiente esterno
- 2) Capacità (o quantità) di memoria = memorizzazione di dati e istruzioni
- 3) Elaborazione = esecuzione di istruzioni
- 4) Connessione reciproca dei 3 sottoinsiemi

Il sottosistema di interfaccia (1) rende possibile agli utenti l'input (inserimento in ingresso) con mouse, tastiere e microfoni, e al calcolatore l'output (produzione in uscita).

Il sottosistema di elaborazione (3) è l'unità centrale di elaborazione (CPU = Central Processing Unit) opera sequenzialmente. Questi sistemi sono detti a programma memorizzato, contrariamente a quelli a programma cablato, in cui ogni programma era eseguito su un dispositivo differente.

Esistono differenti interazioni uomo-macchina:

- 1) Interazione uomo-macchina, calcolatore stand alone (non connesso ad una rete)
- 2) Interazione tra un utente e un calcolatore remoto, mediata da un secondo calcolatore (es: navigare nel web)
- 3) Interazione tra utenti mediata da calcolatori (es: messaggio di posta elettronica)

L'utente invia un comando nella forma di testo scritto o di tipo point-and-click (GUI = Graphical User Interface), grazie ai sistemi di interfaccia grafica. Il calcolatore opera prima come interprete del comando (controlla di essere in grado di interpretare il comando) e poi come esecutore (esegue l'azione associata al comando).

Es: se scrivo "1"+2 viene interpretato come testo ciò che è tra le virgolette e l'operando + come concatenazione e mi dà come risultato un testo composto dal carattere 1 seguito da 2 ("12").

Per evitare ambiguità si adoperava un linguaggio semplice, con una semantica limitata.

La rigidità dei linguaggi di programmazione (Basic, linguaggio C e Java) è garanzia di precisione e correttezza della comunicazione.

I calcolatori operano anche su dati non numerici (testi, immagini, musica), che possono essere ricondotti a calcoli numerici grazie alla codifica numerica, cioè una regola per convertire ogni entità in ingresso al problema in una corrispondente entità numerica, grazie a molte regole diverse, es: codice ASCII (American Standard Code for Information Interchange).

Attraverso il calcolo, il dato iniziale (input) viene trasformato in un risultato (output), esso riduce le funzioni complesse a successioni complesse di funzioni semplici (elementari). Quello del calcolatore si chiama linguaggio macchina (di basso livello), mentre quello del programmatore linguaggio di programmazione (di alto livello).

Tesi di Church-Turing = ogni funzione "naturalmente considerata calcolabile" è calcolabile da una macchina di Turing, che rappresenta i numeri mediante una codifica "a barre e vuoti", come avere un nastro organizzato in celle.

## **CAP.4 - APPROFONDIMENTI**

4.1 L'*equivocazione* è una grandezza che misura la quantità d'informazione che è stata persa mediamente da ogni simbolo nel suo passaggio attraverso il canale. Es: 0 bit/simbolo = completa assenza di rumore.

Se l'*equivocazione* è uguale alla quantità media d'informazione inviata dalla sorgente, i simboli in uscita dal canale sono statisticamente indipendenti dai simboli in ingresso, quindi il canale si comporterebbe come se fosse una nuova sorgente, risultando completamente inutile.

La *capacità di canale* si misura in bps (bit/simbolo) ed esprime la quantità massima d'informazione che può essere trasmessa lungo il canale ogni volta che si trasmette un simbolo. Il bit/simbolo si ottiene moltiplicando il valore della grandezza per la velocità con cui il canale è in grado di trasmettere ogni simbolo, ottenendo così la massima quantità d'informazione utile.

4.2 Un esempio di utilizzo della ridondanza è il bit di parità. Supponiamo che il trasmettitore abbia già convertito in serie di 0 e 1. La successione viene inviata al canale in blocchi di N simboli binari, in coda a ognuno dei quali viene aggiunto un ulteriore simbolo (il bit di parità): 0 se la somma dei precedenti N simboli è un numero pari e 1 se è dispari.

Si dice *parità pari* se il numero complessivo di 1, compreso il bit di parità, è pari e *parità dispari* in caso contrario.

Es: messaggio binario = 01011101 con N=4 verrà diviso in 2 blocchi di 4 cifre (0101 e 1101)

Se adottiamo la parità pari il messaggio codificato dopo sarà: 01010 11011 -> 1+1=2 pari; 1+1+1=3 dispari

Così il ricevitore sarà in grado di rilevare un eventuale errore, es: 11010 -> doveva esserci l'1 alla fine, perché 1+1+1=3, è dispari!

Il limite della codifica con bit di parità consente però solo di rilevare l'errore, ma non di correggerlo. Occorre quindi richiedere nuovamente al trasmettitore di inviare nuovamente il segnale originario.

4.4 Se il canale fosse ideale, il segnale emesso dalla sorgente che arriva al ricevitore, trasmetterebbe senza attenuazione tutte le frequenze che lo compongono.

Il valori rilevati sono interpretati come 0 e 1, se il punto cade nella corrispondente fascia, altrimenti viene considerato non accettabile.

Inoltre, aumentando la velocità di trasmissione (e quindi riducendo il periodo  $T$ ), per mantenere la stessa qualità del segnale ricevuto occorre estendere la larghezza di banda del canale.

In assenza di rumore, un canale con una larghezza di banda pari a  $B$  Hz può trasportare al più  $2B$  al secondo a una capacità di canale pari a  $C = 2B \log_2(n)$  con  $n =$  numero dei simboli possibili

Questo stabilisce un limite alla velocità con cui un canale è in grado di far transitare informazione utile, a partire dai suoi parametri fisici, larghezza di banda e rapporto segnale/rumore ( $S/N$ ).

## CAP.8

I tre sottosistemi da cui un'applicazione è costituita (interfaccia utente, logica applicativa e gestione dei dati) possono risiedere ed essere eseguiti anche su calcolatori diversi.

Nell'*architettura client-server* l'utente opera con un calcolatore connesso ad una rete, l'unico sottosistema che deve essere eseguito localmente è quello incaricato della gestione dell'interfaccia utente, chiamato componente client dell'applicazione.

I sottosistemi di logica applicativa e di gestione dei dati, che costituiscono il componente server dell'applicazione, possono risiedere ed essere eseguiti su un calcolatore remoto (fisicamente distante) purché però sia connesso. L'utente deve solo riferirsi all'applicazione in esecuzione sul suo calcolatore.

L'architettura client-server serve per condividere dati e servizi applicativi, accessibili a tutti gli utenti che dispongono del programma e sono autorizzati ad accedere.

Attraverso un programma client, l'utente invia una richiesta a un programma server che la interpreta e, se corretta, la soddisfa producendo una risposta che invia al client, il tutto attraverso la rete.

Perché si instauri una comunicazione tra client e server si devono soddisfare 2 condizioni:

- 1) client e server si devono trasmettere successioni di bit;
- 2) client e server devono concordare ciò che attiene agli aspetti applicativi del dialogo che intercorre tra essi. Quest'insieme di regole di livello applicativo è chiamato *protocollo applicativo*.

Client diversi possono operare con uno stesso server purché tutti implementino lo stesso protocollo applicativo.

La posta elettronica, per esempio, consente di impiegare i server come intermediari tra i client ossia come strumenti di memorizzazione dei messaggi, nel caso di temporanea irraggiungibilità del client del destinatario. L'architettura del sistema è complessa e richiede l'implementazione di 2 protocolli applicativi:

1) Il mittente opera con un client e invia un messaggio al suo server incaricato della spedizione (il protocollo usato in questa fase si chiama SMTP (Simple Mail Transfer Protocol).

2) Il server del mittente invia il messaggio al server che mantiene la mailbox del destinatario, sempre mediante SMTP. Dato che il server del mittente e quello del destinatario sono connessi con Internet, il protocollo SMTP accetta la possibilità di una loro temporanea irraggiungibilità e implementa dei meccanismi di "connessione robusta"

3) Una volta che il messaggio è giunto al server del destinatario, questo accede alla sua mailbox e ottiene il messaggio (il protocollo applicativo impiegato in questa fase è POP3 (Post Office Protocol) oppure IMAP4 (Internet Message Access Protocol)). Quindi, perché un utente possa ricevere un messaggio è necessario che il suo client sia correttamente configurato per le opzioni del server POP o IMAP.

L'architettura client-server è alla base di tutti i servizi applicativi di internet compreso World Wide Web (www o in breve web); qui il client è il navigatore web (browser) e il server si chiama server web.

Il web è inteso come un'unica grande applicazione distribuita, costituita dall'insieme di tutti i server Web accessibili su internet e dai browser che gli utenti connessi impiegano per navigare, per rendere loro accessibili i file memorizzati in particolari directory del file system del calcolatore.

Esempio:

- Un utente scrive nella barra dell'indirizzo nel browser

"http://www.curioso.it/CoseInteressanti/BelloQuesto.txt (cerca il file BelloQuesto, nella cartella CoseInteressanti quindi) e preme invio

- Il browser genera il testo "GET/Cose Interessanti/BelloQuesto.txtHTTP/1.1

Host: [www.curioso.it](http://www.curioso.it)

User-Agent: Mozilla/5.0 (Windows NT 5.1) Firefox/1.5

Accept: \*/\* " e attraverso internet lo invia al server

- Il server web, ricerca il file richiesto e una volta trovato manda un messaggio "HTTP/1.1 200OK

Server: Apache-Coyote/1.1

Content-Type: text/plain

Content-Length:14

Saluti a tutti" che invia come risposta al browser da cui aveva ricevuto richiesta (la prima linea specifica un codice di successo, 200, o errore, 404 in caso di file non trovato)

- Il browser riceve il messaggio di risposta, legge l'intestazione (tipo e lunghezza del file contenuto), legge il corpo del messaggio e lo visualizza come tale per l'utente

HTTP (HyperText Transfer Protocol) è il protocollo applicativo adottato nella comunicazione tra browser e server. Una sua caratteristica è la *semplicità* e le comunicazioni http possono essere gestite senza stato (stateless), ossia senza tenere traccia di altre connessioni effettuate precedentemente e quindi indipendentemente l'una dall'altra. I server possono operare solo in modo istantaneo, producendo risposte alle richieste senza dover memorizzare la successione: può essere un limite, in quanto tutte le volte è necessario mantenere traccia dello stato della connessione (es: servizi con autenticazione dell'utente).

I principali servizi applicativi di Internet sono:

- Telnet (impiega il proprio calcolatore come terminale di un calcolatore remoto, es: accedere a cataloghi di biblioteche e banche dati)

- Posta elettronica (e-mail) (consente agli utenti di scambiarsi messaggi e file tramite allegati, es: mailing list, ovvero liste di distribuzione in cui ogni messaggio viene automaticamente reindirizzato a tutti gli utenti della lista)

- Trasferimento file (basato sul protocollo FTP File Transfer Protocol, es: per distribuire dati e programmi in calcolatori che accettino che qualsiasi utente, anche anonymous, cioè non identificato, possa connettersi; oppure consultare l'elenco dei file presenti e copiare sul proprio calcolatore, tramite downloading, quanto è di proprio interesse, come E-Mule)

- Servizi vari come il ping (la verifica che un calcolatore connesso alla rete sia effettivamente accessibile) e il traceroute (la verifica del percorso che seguono i pacchetti dal proprio calcolatore per giungere a un calcolatore destinatario)

Un fattore che ha contribuito alla diffusione del web è stata la standardizzazione di uno schema uniforme di identificazione applicativa che consente di assegnare un metodo di accesso e un indirizzo chiamati URL (Uniform Resource Location) a ognuno dei tanti file presenti sulla rete (pagine web o risorse). Alla base del concetto URL c'è l'ipotesi che ogni "calcolatore presente sulla rete con funzioni di server" (detti anche host) disponga di un indirizzo univoco.

Ogni processo accessibile su un host deve essere attivo su una definita porta TCP dell'host stesso.

Un URL include l'identificatore dell'host, della porta TCP e il nome della risorsa mantenuta:

protocollo://host:portaTCP/risorsa

l'URL: <http://server1.cetic.liuc.it:80/dida/calendario.html> (negli URL il separatore di directory è "/" in stile Unix/Linux, non "\" come in MS DOS/Windows)

Per http la porta TCP di default è 80.

Ogni volta che l'URL viene inviato come richiesta si ottiene sempre la stessa risposta, perciò il web si comporta come un'entità statica.

Può esserci anche un extraURL, che lo separa col carattere "?" ponendo una lista di coppie: nomeparametro=valore, ognuna separata dal carattere "&".

Il programma messo in esecuzione dietro il server web, chiamato *back end*, implementa la logica applicativa e la gestione dei dati dell'applicazione, mentre il server web deve gestire la comunicazione con i browser e

controllare i processi in back end. Esso è l'unico sottosistema che dev'essere costruito ad hoc per quanto riguarda la standardizzazione, perché non opera in modo indipendente dalla specifica applicazione.

Il sistema è basato sulla combinazione di 2 sottosistemi client-server: il browser opera come client verso il web e il server web assume temporaneamente un ruolo di client.

Il principio delle deleghe successive avviene perché un client non è in grado di svolgere un compito applicativo e si rivolge perciò a un server. Può esserci una specializzazione, in cui ogni sottosistema del sistema complessivo può essere realizzato per svolgere un compito particolare.

Il Web è un'entità dinamica, un ambiente per interagire con molteplici applicazioni in rete tramite un solo client, il browser.

Il funzionamento dei motori di ricerca si rifà alla logica della ricerca full text, ovvero l'intero testo di ogni pagina viene scandito, per stabilire se la stringa cercata è presente.

Un motore di ricerca è un'applicazione web dinamica, con un'architettura a 4 componenti. La generalità della logica della ricerca full text ha il limite di poter operare solo sul testo delle pagine e non anche sulle immagini in esse.

Il principale punto debole dei motori di ricerca è che operano in una prospettiva di tipo sintattico, es: digitando "albero" il motore di ricerca mi fornirà link a pagine che trattano di vegetazione, meccanica (alberi motore), genealogie (alberi genealogici), ma non includerà pagine che contengono il termine "tree" o "arbre".