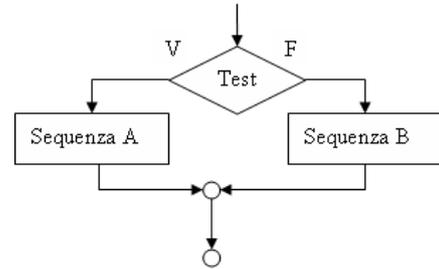
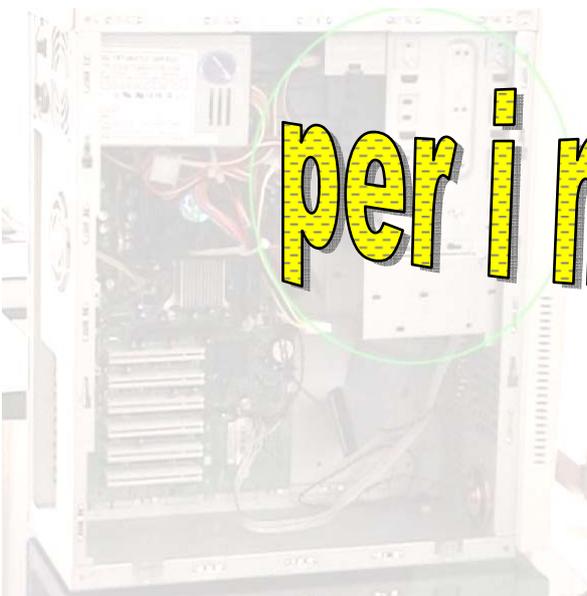


Prof. Bianco Mauro



L'informatica

per i miei studenti



Volume 3 - Classe quinta (Bozza)

A mia madre Gerardina, a mio padre Raffaele

Perché, per prima, si pensa a chi non c'è più

A mia moglie Antonietta

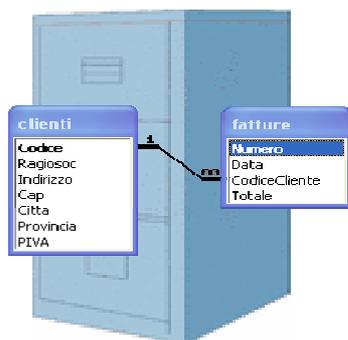
Il mio presente

A mio figlio Raffaele, a mia figlia Ilary

Il futuro

Ai miei studenti

Perché è per loro che ho scritto



Capitolo 9.

Data Base

9.1. Introduzione

La prima cosa importante che ritengo di dirvi è che questo capitolo è sicuramente uno dei più importanti di tutto il triennio, per cui vi chiedo una maggiore attenzione ed impegno al fine di raggiungere gli obiettivi prefissati; si tratta di un argomento molto interessante in cui teoria e pratica si fondono con grande armonia ed è per questo che la trattazione sarà ricca di esempi; in particolare farò spesso riferimento al data base di un'azienda commerciale ed a quello di una scuola, come la nostra per esempio; in seguito mi riferirò a questi esempi con i termini **progetto azienda** e **progetto scuola**. Il data base di un'azienda conterrà sicuramente le informazioni sui clienti, sui fornitori, gli articoli di magazzino, le fatture, ecc.; per la scuola troveremo l'archivio degli studenti e le relative assenze, i dati sui docenti, le classi, l'orario, ecc. .

La maggior parte dei linguaggi di programmazione attuali dispone di istruzioni per il trattamento dei soli file ad accesso sequenziale e diretto. La realizzazione di una complessa procedura per il trattamento dei dati di un'azienda o di una qualsiasi organizzazione, ovvero del loro data base, risulta poco agevole con un linguaggio di programmazione che offre solo la gestione di tali tipi di file. Con il trattamento dei file ad accesso diretto in VB.NET, abbiamo visto come la semplice operazione di ricerca richieda la realizzazione di un sottoprogramma e ancora più complicate sono risultate le operazioni di cancellazione ed inserimento di un nuovo record; anche una semplice variazione del tracciato record, quale l'inserimento o l'eliminazione di un campo o semplicemente la modifica del suo tipo e/o della sua lunghezza, comporta, in genere, modifiche più o meno complesse ai programmi applicativi; inoltre, ogni qual volta è necessario estrarre dati da uno o più file, occorre realizzare un algoritmo più o meno complesso.

La realtà oggi è completamente diversa; il programmatore gestisce il data base mediante sottoprogrammi, che rendono estremamente semplice il trattamento dei dati, messi a disposizione dal sistema.

Ho già definito il termine data base lo scorso anno ma ritengo opportuno riportarne adesso una definizione più articolata: **un data base è l'insieme di tutte le informazioni di una qualsiasi organizzazione per servire più utenti e più applicazioni.**

La base dei dati è accessibile, come detto, mediante una serie di programmi messi a disposizione dal sistema e che costituiscono il **DBMS (Data Base Management System)**; i linguaggi di programmazione rendono disponibili al programmatore metodi e proprietà di classi che utilizzano il software messo a disposizione dal DBMS; il DBMS, pertanto, è l'insieme dei programmi che consente l'accesso alla base dei dati cosicché una qualsiasi operazione sui dati registrati nel data base può avvenire soltanto attraverso le routine messe a disposizione dal DBMS. Questo approccio alla gestione della base dei dati porta a numerosi vantaggi:

- **Eliminazione della ridondanza:** la ridondanza è la duplicazione dell'informazione negli archivi; in altre parole la stessa informazione può comparire più volte in file diversi. Per esempio i dati anagrafici di un cliente possono essere registrati nell'archivio della contabilità e della posta elettronica; in un data base la stessa informazione compare una sola volta;
- **Inconsistenza:** la ridondanza può portare all'inconsistenza ovvero la stessa informazione rappresentata con dati diversi; in un tal caso non si sa quale sia l'informazione corretta. Con riferimento all'esempio di cui al punto precedente, l'indirizzo dello stesso cliente potrebbe essere riportato in maniera diversa nell'archivio della contabilità ed in quello della posta elettronica;
- **Indipendenza fisica:** i programmi applicativi, in genere, non devono essere corretti quando cambia il modo con il quale i dati sono registrati sulle memorie di massa; l'allocatione su tali memorie è una prerogativa del DBMS;
- **Indipendenza logica:** la modifica della struttura logica del data base, quale, ad esempio, la modifica di un campo, la sua cancellazione o l'aggiunta di un nuovo campo, rende minime le modifiche da apportare al software; insieme all'indipendenza fisica, riduce molto il costo della manutenzione dei programmi;
- **Integrità:** l'integrità fa riferimento alle relazioni esistenti tra i dati presenti nel data base; per esempio un cliente riceve più fatture, così come una classe è composta da più studenti; i data base, per esempio, garantiscono l'integrità della chiave primaria, che consiste nell'evitare che essa sia nulla. In seguito vedremo l'integrità referenziale. I software che gestiscono i data base consentono, al programmatore ed a qualsiasi utente che ha accesso ai dati, una semplice gestione delle transazioni, così come descritte nel capitolo relativo al software, garantendo così l'integrità dei dati;
- **Sicurezza:** tutti gli accessi alla base dei dati avvengono attraverso il DBMS che può facilmente gestire diritti di accesso e/o modifica ai dati mediante l'autenticazione dell'utilizzatore che avviene, di norma, mediante l'immissione di uno user name e di una password;
- **Efficienza:** gli algoritmi utilizzati dal DBMS sono ottimizzati per velocizzare innanzitutto l'accesso e la ricerca dei dati nonché tutte le altre operazioni necessarie quali inserimenti, cancellazioni, estrazioni, ecc.;

- **Condivisione:** i dati presenti nel data base sono condivisibili tra più utenti e più applicazioni; è possibile sviluppare programmi, che accedono allo stesso data base, con linguaggi di programmazione diversi; in passato ogni linguaggio gestiva i propri archivi con notevoli limitazioni di trasportabilità e riutilizzabilità dei dati in applicativi e piattaforme diverse.

9.2. Modelli

Nel capitolo relativo alle strutture dati abbiamo visto come questi possano aggregarsi tra loro creando strutture; analogamente i dati presenti nei data base possono relazionarsi tra loro in tanti modi diversi e possono essere registrati sulle memorie di massa secondo tecniche più o meno complicate; in altri termini abbiamo ben capito la differenza tra come i dati appaiono all'esterno e come essi siano effettivamente memorizzati: **aspetto logico e fisico dei dati**. Per la memorizzazione dei dati di un data base e delle numerose relazioni esistenti tra essi, di cui parleremo in maniera approfondita in questo capitolo, si utilizzano tutte le tecniche studiate nello scorso anno scolastico, ovvero strutture sequenziali ad indici, alberi di ricerca binaria, randomizzazione, ecc. .

Ai diversi modi con i quali possono essere legati tra loro e memorizzati sulle memorie di massa ci si riferisce con il termine **modello**. Un modello è organizzato in **livelli**:

- **Livello concettuale** o **esterno**: rappresenta i dati e le loro relazioni così come appaiono nella realtà; viene rappresentato mediante uno schema detto **schema concettuale**. Con riferimento al progetto scuola, si tratta di individuare i dati della realtà scolastica riassumibili in docenti, studenti e relative assenze, classi, orario etc.; è possibile che utenti o programmatori siano interessati ad una parte del database e allora si parla di **sottoschemi**;
- **Livello logico**: rappresenta l'organizzazione logica dei dati riferita alla loro memorizzazione sulle memorie di massa; in altri termini i dati descritti al punto precedente diventano file o, come vedremo in seguito, tabelle, tracciati record, relazioni tra tabelle, etc.; con riferimento al progetto scuola, si tratta, tra l'altro, di definire i tracciati record degli archivi:

STUDENTI (Matricola, cognome, nome, indirizzo,)

DOCENTI (Codice, cognome, nome,)

- **Livello fisico**: fa riferimento a come i dati vengono fisicamente allocati sulle memorie di massa. Il programmatore è, in genere, sollevato dai problemi derivanti dal livello fisico; il DBMS si occupa in maniera autonoma dell'allocazione dei dati sulle memorie di massa ottimizzando spazio e tempo necessario per operare su di essi.

La realizzazione di un data base, quindi, comporta l'analisi approfondita della realtà in cui si opera allo scopo di definirne lo schema concettuale dal quale passare, in genere in maniera piuttosto semplice, allo schema logico dei dati e, infine, alla loro memorizzazione sulle memorie di massa, compito dal quale il programmatore è sollevato.

9.2.1. Modello E/R

Tra i più diffusi modelli concettuali troviamo il **modello E/R**, **Entity/Relationship (entità/associazioni)**. Lo scopo di questo modello è quello di rappresentare la realtà dei dati in maniera assolutamente indipendente dai programmi applicativi che andranno ad utilizzarli e da come essi saranno memorizzati.

Si tratta di un modello che fa uso di simboli grafici per rappresentare i dati dello schema concettuale e delle relazioni esistenti tra essi. Si basa essenzialmente su:

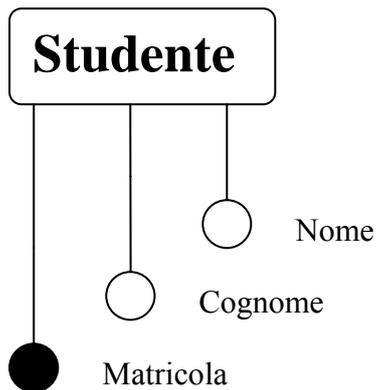
- **Entità**: l'oggetto sul quale raccogliamo l'informazione; una entità viene descritta mediante un rettangolo al cui interno ritroviamo il suo nome; una entità si compone di istanze; per esempio nel progetto scuola ritroviamo l'entità studente, l'entità docente, l'entità classi, etc.;



Ogni studente è una istanza dell'entità studente.

- **Attributi**: descrivono le proprietà dell'entità; un attributo è descritto mediante un nome, il tipo e la lunghezza; alcuni attributi possono essere opzionali nel senso che possono essere non valorizzati ovvero contenere il valore nullo (null); il valore di ciascun attributo appartiene ad un insieme detto **dominio**. Bisogna evitare di inserire attributi che possono essere derivati in maniera semplice ovvero mediante un calcolo perché porterebbero ad una ridondanza; per esempio, nell'entità studenti sarebbe ridondante l'attributo ANNI perché l'età dello studente può essere ottenuta dalla data di nascita e, inoltre, la data di nascita non varia mentre ANNI dovrebbe essere aggiornata ogni anno; così sarebbe ridondante per l'entità CLIENTI l'attributo SALDO in quanto facilmente calcolabile come AVERE-DARE.

Alcuni attributi costituiscono una **chiave** o **chiave primaria**, di cui ho parlato lo scorso anno nel capitolo 6, siete obbligati a rivedere tale argomento. Gli attributi chiave vengono evidenziati mediante una sottolineatura o il riempimento del corrispondente cerchio come appare nell'esempio seguente:



Oppure:

STUDENTE (Matricola, cognome, nome, indirizzo,)

- **Associazioni:** le entità di un data base non sono isolate ma relazionate tra loro; una associazione mette in relazione 2 entità. Per esempio un cliente riceve più fatture così come una classe è composta da più alunni. Il **grado** di un'associazione è il numero di istanze dell'entità di arrivo corrispondenti ad una istanza dell'entità di partenza. Una associazione può essere **obbligatoria**, quando a ciascuna istanza della prima entità **deve corrispondere** una o più istanze della seconda (grado>0) oppure **opzionale** (grado=0 per qualche istanza). **Ad ogni associazione è assegnato un nome;**

In funzione del grado, le associazioni possono essere di tre tipi:

- 1. Associazione 1:1 (uno a uno) o biunivoca:** ad una istanza della prima entità corrisponde una sola istanza dell'altra entità e viceversa;

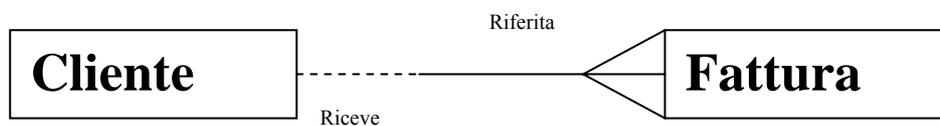
per esempio, riferendoci al progetto azienda, potrei avere la seguente entità ANAGRAFICA: ANAGRAFICA(Codice, RagioSoc, Indirizzo, Citta,)

L'entità CLIENTE ha bisogno di una anagrafica e dati contabili; in questo caso ogni istanza dell'entità CLIENTE non può contenere l'anagrafica perché sarebbe ridondante ma, semplicemente e per tali informazioni, viene associata ad una istanza dell'entità ANAGRAFICA; tra CLIENTI e ANAGRAFICA vi è una associazione 1:1.

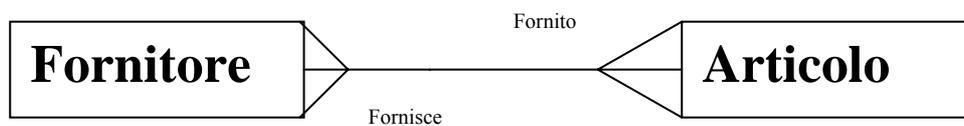


La **linea tratteggiata** indica l'**opzionalità** mentre la **linea continua** l'**obbligatorietà**; così ogni anagrafica **può** riferirsi ad un cliente e ogni cliente **possiede** una anagrafica;

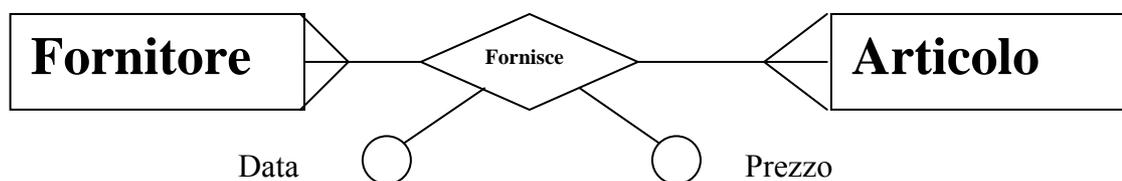
2. **Associazione 1:n (uno a molti):** ad una istanza della prima entità corrispondono più istanze della seconda mentre ad ogni istanza della seconda corrisponde al più una sola istanza della prima. Sono associazioni uno a molti CLIENTE-FATTURA, CLASSE-STUDENTE.



3. **Associazione n:m (molti a molti):** ad una istanza della prima entità corrispondono più istanze della seconda e viceversa. Sono associazioni molti a molti FORNITORE-ARTICOLO, DOCENTE-STUDENTE.



E' possibile utilizzare un rombo per indicare l'associazione in maniera tale da poter descrivere anche gli attributi dell'associazione:



Come si vede, gli attributi data e prezzo non fanno parte né dell'entità Fornitore né di Articolo ma dell'associazione, perché fanno riferimento al prezzo ed alla data di acquisto di un determinato articolo da un certo fornitore.

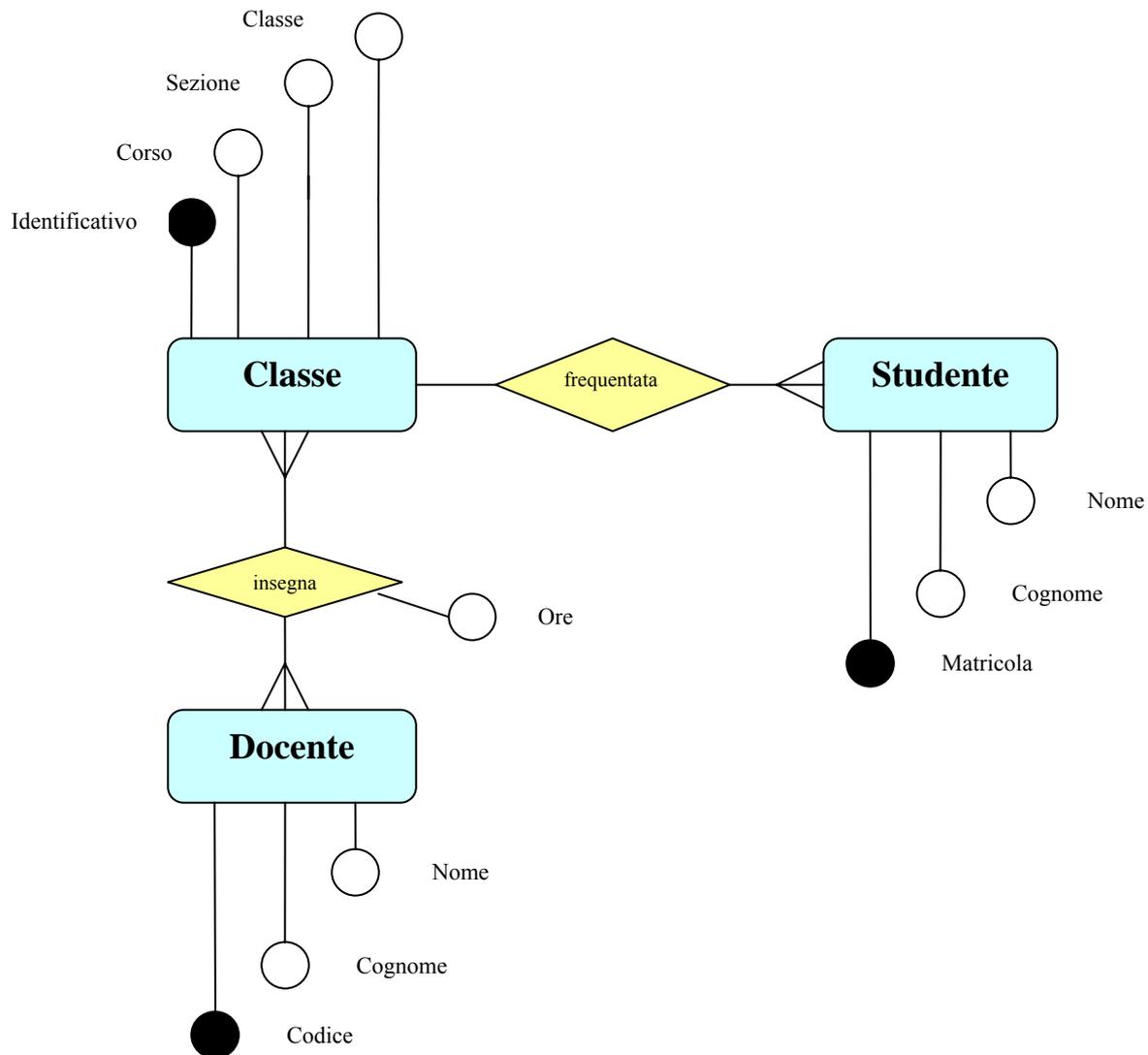
E' simpatica l'analogia con il matrimonio, cosicché l'associazione 1:1 fa riferimento al matrimonio tradizionale, 1:n alla poligamia e n:m al matrimonio di gruppo (e con i tempi che corrono non ci si stupisce più!!!!).

Dal livello concettuale si passa al livello logico in maniera molto semplice anche perché si rientra in una terminologia a noi già nota; in pratica una entità diventa un archivio e, come vedremo in seguito, una o più tabelle; un attributo non è altro se non un campo e l'insieme di tutti gli attributi è quello che noi abbiamo chiamato record. Così un record di un file corrisponde, in pratica, ad una istanza dell'entità. Per ogni archivio si definisce, in genere, una chiave primaria.

Realizzare le associazioni è molto semplice:

- L'associazione 1:1 può essere realizzata **unendo** le due entità in un unico archivio il cui tracciato record conterrà gli attributi di entrambe (ma non sempre dico io);
- L'associazione 1:n viene realizzata utilizzando la chiave primaria della prima entità come **attributo o chiave secondaria**, detta **chiave esterna** o **foreign key**, della seconda entità.
- L'associazione n:m viene realizzata mediante un **terzo archivio**, in aggiunta ai due delle entità associate; tale terzo archivio conterrà due chiavi esterne, che corrispondono alle due chiavi primarie delle entità di partenza, con eventuali attributi aggiunti, come abbiamo visto in precedenza per l'associazione FORNITORE-ARTICOLI. Per esempio, nell'associazione molti a molti tra CLASSE e DOCENTE, l'archivio che realizza l'associazione conterrà gli identificatori univoci di DOCENTE e CLASSE, che diventano chiavi secondarie esterne, ed altri attributi quale, ad esempio, ORE che rappresenta il numero di ore assegnate ad un docente su una classe; in questo modo una istanza di tale terzo archivio individua, in maniera univoca il docente, la classe ed il numero di ore che quel docente svolge in quella classe. E' chiaro che una tale informazione nell'archivio DOCENTE (CLASSE) indicherebbe il numero totale di ore settimanali del docente (della classe) cosa ben diversa da ORE nell'archivio che realizza l'associazione.

Nell'esempio che segue viene riportato lo schema concettuale relativo al progetto scuola di cui vi ho parlato ad inizio capitolo; lo schema è limitato alle entità studente, classe e docente.



9.2.2. Modelli logici

Una volta realizzato il modello concettuale, occorre passare allo schema logico del data base ovvero organizzare i dati in modo tale da ottimizzare le operazioni di accesso e manipolazione dei dati in esso contenuti.

L'organizzazione più semplice potrebbe prevedere un unico file; una tale organizzazione, detta **flat file**, è utilizzabile solo per data base piuttosto banali e non è utilizzabile nella maggior parte delle applicazioni reali.

Nel corso del tempo si sono sviluppati i seguenti modelli:

- **gerarchico**: sviluppato negli anni sessanta, è utilizzato quando è possibile individuare una gerarchia tra i dati da rappresentare; il primo DBMS, sviluppato da **IBM** e detto **IMS**, era di tipo gerarchico; il modello gerarchico è strettamente legato ad una struttura dati non lineare studiata lo scorso anno ovvero agli alberi;

- **Reticolare:** sviluppato negli anni settanta, costituisce una estensione del modello gerarchico e si basa su record e puntatori; il modello gerarchico è strettamente legato ad una struttura dati non lineare studiata lo scorso anno ovvero ai grafi;
- **Relazionale:** sviluppato negli anni settanta/ottanta, si basa sul concetto matematico di relazione ed i dati sono organizzati in tabelle; parlerò del modello relazionale nel prossimo paragrafo;
- **Orientato agli oggetti (OODB, Object Oriented Data Base):** sviluppato negli anni ottanta, fa uso di classi ed oggetti ed acquisisce i concetti tipici della programmazione orientata agli oggetti.

Attualmente i modelli gerarchico e reticolare sono obsoleti; la maggior parte dei data base commerciali odierni si basa sul modello relazionale, sia che vengano implementati su grandi elaboratori che su medi ed addirittura piccoli sistemi di elaborazione ovvero i personal computer.

9.3. Modello relazionale

Il modello relazionale si basa sul concetto matematico di relazione; dati n insiemi A_1, A_2, \dots, A_n , si dice relazione un sottoinsieme del prodotto cartesiano $A_1 \times A_2 \times \dots \times A_n$; una relazione, quindi, è un insieme nel quale ciascun elemento è una n -pla (a_1, a_2, \dots, a_n) con a_i appartenente ad A_i . La terminologia utilizzata per il modello relazionale definisce:

- **Dominio:** gli insiemi A_i sono denominati domini;
- **Tuple:** gli elementi della relazione sono, come detto, n -ple dette anche tuple;
- **Cardinalità:** il numero di elementi presenti nella relazione è detta cardinalità;
- **Grado:** il numero degli insiemi, n nel nostro caso, è detto grado.

Con riferimento al modello concettuale possiamo dire che una entità viene rappresentata mediante una relazione (direi una o più tabelle), un'istanza dell'entità non è altro se non una tupla della relazione, e gli attributi sono pari al grado della relazione ed ognuno di essi appartiene al corrispondente dominio.

Una relazione viene rappresentata mediante una tabella costituita da righe e colonne; ogni riga rappresenta una tupla o istanza della relazione, ogni colonna è un attributo il cui valore appartiene al corrispondente dominio; il numero di righe della tabella è la cardinalità ed il numero di colonne il grado:

Tabella studenti:

Matricola	Cognome	Nome	DataNascita
1234	Bianco	Mauro	12/10/1978

1235	Verdi	Carlo	11/11/1979
.....
2122	Rossi	Antonio	01/05/1977

La struttura di una tabella può anche essere semplificata con la notazione

NomeTabella(AttributoChiave, attributo, ..., attributo)

Come, per esempio,

STUDENTI(Matricola, Cognome, Nome, ...)

Per gli studenti, pertanto, i domini sono l'insieme di tutte le possibili matricole, di tutti i possibili cognomi e nomi, ecc.; con riferimento alla città di residenza o la provincia oppure il CAP, i domini sarebbero, rispettivamente, l'elenco dei comuni d'Italia, delle province e dei CAP.

Ancora una volta si sta “semplicemente” cambiando la terminologia perché, in pratica, una istanza di una entità diventa una istanza di una tabella ma altro non è se non quello che in passato chiamavamo record così come un attributo era un campo, la cardinalità era il numero dei record ed il grado era il numero dei campi; inutile dire che una chiave è costituita da uno o più attributi. D'ora in poi considereremo questi termini come sinonimi anche se sarebbe opportuno utilizzare la nuova terminologia introdotta.

La definizione di una relazione richiede la descrizione dei suoi domini e, pertanto, la descrizione di una tabella comporta la necessità di elencare gli attributi o campi che la costituiscono; per ciascun campo deve essere riportato il tipo e l'eventuale lunghezza oltre ad una serie di altre caratteristiche così come consentito dal DBMS utilizzato; in altre parole bisogna definire il **tracciato record**, così come definito per i file ad accesso diretto lo scorso anno scolastico; a tal proposito è possibile utilizzare il modulo DF riportato nel capitolo relativo alla programmazione.

Come appena accennato, oltre al tipo ed alla lunghezza del campo, i DBMS consentono di specificare, per ciascun attributo, numerosi altri parametri; per esempio è possibile utilizzare la clausola “**not null**” per indicare che il campo non può essere vuoto, oppure clausole come “**valid**”, ad esempio “valid > 0”, che indica la validità del campo, ovvero una condizione logica che deve essere soddisfatta dal valore assegnato al campo in fase di immissione dei dati; naturalmente occorre definire la chiave primaria e le eventuali chiavi secondarie. Approfondirò questo discorso in laboratorio durante le lezioni su MS-ACCESS.

Anche le tabelle, come le entità, non sono isolate ma legate tra loro da associazioni che, come per il modello concettuale, possono essere 1 a uno, uno a molti e molti a molti. Passare dallo schema concettuale a quello logico è estremamente semplice si dice, infatti, che **lo schema logico può essere derivato da quello concettuale applicando delle semplici regole**:

1. **Ogni entità** diventa una tabella raramente più di una tabella;
2. **Ogni attributo** diventa un dominio ovvero un campo della tabella le cui caratteristiche, compresa quella di essere una chiave, sono le stesse dell'attributo corrispondente;
3. **L'associazione 1:1** viene realizzata unendo le due entità in un'unica tabella oppure utilizzando la stessa chiave primaria in entrambe; per esempio il campo codice chiave primaria per clienti e anagrafica;
4. Per **l'associazione 1:m** si utilizza una terminologia che definisce **tabella madre** quella corrispondente all'entità 1 e **figlia** quella corrispondente all'entità molti; per esempio nell'associazione Clienti/Fatture, la tabella clienti è la tabella madre mentre la tabella fatture è la figlia. L'associazione 1:m viene realizzata in maniera molto semplice duplicando la chiave primaria della tabella madre all'interno della tabella figlia e definendola come secondaria; tale chiave, nella tabella figlia, è detta **chiave esterna o foreign key**. Per esempio la chiave primaria codice della tabella madre CLIENTI è riportata come chiave secondaria all'interno della tabella FATTURE.

L'obbligatorietà dell'associazione dalla tabella figlia alla tabella madre, da fatture a clienti per intenderci, è riferita come **integrità referenziale**. L'integrità referenziale impedisce la presenza di istanze nella tabella figlia che non abbiano un corrispondente nella tabella madre, una fattura senza cliente nel nostro esempio ovvero uno studente senza riferimento ad una classe. In maniera più precisa posso dire che, in una associazione 1:m, **l'integrità referenziale garantisce la presenza, nella tabella madre, di un record corrispondente a ciascun record, avente chiave esterna non nulla, della tabella figlia.**

Applicando l'integrità referenziale, non sarà possibile cancellare un record della tabella madre che ha corrispondenti nella tabella figlia a meno che non si decida di cancellare, automaticamente, anche tutti i record corrispondenti; allo stesso modo non sarà possibile modificare il valore della chiave primaria di una istanza a meno di non consentirne l'automatico aggiornamento in tutti i record corrispondenti della tabella figlia.

Si considerino le tabelle fatture e bolle; l'archivio fatture contiene sia le fatture differite che immediate; l'associazione è di tipo 1:m ma non c'è obbligatorietà in nessuno dei due versi quindi niente integrità referenziale; una bolla, infatti, potrebbe non essere riferita ad alcuna fattura perché relativa a merce consegnata per visione o comodato d'uso o reso; d'altra parte una fattura immediata non si riferisce ad alcuna bolla.

5. **L'associazione n:m** viene realizzata utilizzando una terza tabella oltre alle due relative alle entità associate; in questa terza tabella vengono riportate due chiavi secondarie esterne, corrispondenti alle chiavi primarie delle tabelle principali, e gli attributi dell'associazione; in altre parole l'associazione n:m viene spezzata in due associazioni uno a molti.

Nel caso dell'associazione molti a molti tra articoli e fornitori, la terza tabella conterrà le chiavi esterne codice articolo e codice fornitore oltre agli altri attributi dell'associazione quali prezzo e data; tali attributi si riferiscono al prezzo applicato da quel fornitore a quell'articolo in quella data.

Un esempio che a me piace molto si riferisce all'associazione CLASSE/DOCENTE, che chiamo INSEGNA; tutte e tre le tabelle possono contenere l'attributo ore ma con un significato completamente diverso:

- CLASSE.ORE: Ore settimanali della classe;
- DOCENTE.ORE: Ore settimanali del docente;
- INSEGNA.ORE: Ore settimanali di quel docente in quella classe; vi ricordo che un record o istanza della tabella INSEGNA ha un codice docente, un codice classe ed il campo ore, per esempio

(BIANCOM, 5AP, 5)

Che sta a indicare il numero di ore di insegnamento (5) che il prof. BIANCOM svolge nella classe 5AP.

9.3.1. Operazioni su tabelle

Gli operatori relazionali consentono di eseguire operazioni di interrogazione, ovvero di estrazione, e manipolazione dei dati presenti in un database. E' possibile con una sola operazione estrarre dati da una o più tabelle; il risultato, ovviamente, sarà una tabella come, per esempio, i clienti di una certa provincia o, a volte, uno scalare come nel caso della somma degli importi delle fatture relative ad un intervallo di tempo. Il risultato di ogni operazione, pertanto ed in genere, è un insieme di record. Esempi di interrogazioni sono:

- I clienti di una certa provincia;
- I clienti con un dato fatturato;
- I docenti di una materia;
- Gli studenti di una classe;
- Le fatture di un cliente.

Le operazioni più importanti sono:

1. **La selezione (Select):** applicata ad una tabella genera un insieme costituito solo dalle tuple della tabella di partenza che soddisfano una determinata condizione; vengono, cioè, estratte dalla tabella solo le righe con i valori degli attributi corrispondenti alla condizione prefissata.

Il risultato è una nuova tabella con lo stesso grado di partenza, stesso numero di colonne, e con cardinalità, numero di righe o record o istanze, minore o uguale alla tabella di partenza:

MATR	COGNOME	NOME	CITTA
.....
.....	ARIANO
.....	ARIANO
.....
.....
.....	ARIANO
.....

→ Selezione →

MATR	COGN	NOME	CITTA
.....	ARIANO
.....	ARIANO
.....	ARIANO

Per esempio i clienti di una città, le fatture con totale maggiore di, ecc. .

2. **La proiezione (Project):** applicata ad una tabella genera una nuova tabella che contiene tutte le righe della tabella originaria, stessa cardinalità, ma con le sole colonne scelte, grado minore al grado di partenza:

MATR	COGNOME	NOME	CITTA
.....
.....	ARIANO
.....	ARIANO
.....
.....
.....	ARIANO
.....

→ Proiezione →

MATR	COGNOME	NOME
.....
.....
.....
.....
.....
.....
.....

3. **La congiunzione (join):** si applica a 2 o più tabelle tra le quali è stata definita un'associazione e restituisce una nuova tabella le cui righe sono ottenute concatenando ogni riga della prima tabella con tutte quelle corrispondenti della seconda, ovvero quelle che hanno lo stesso valore nel campo chiave che realizza l'associazione. Il grado della nuova tabella sarà dato dalla somma dei gradi delle due tabelle originarie meno 1, il campo comune che viene riportato una sola volta; se la chiave è composta il grado è dato dalla somma dei gradi meno il numero di attributi comuni ovvero di quelli che costituiscono la chiave:

Tabella Classi:

Classe	Descrizione	Ubicazione
.....
3AP	Terza A Programmatori	Aula 23
4AP	Quarta A Programmatori	Aula 24
5AP	Quinta A Programmatori	Aula 25
.....

Tabella studenti:

Matricola	Cognome	Nome	Citta	Classe
.....
1025					3AP
1245					4AP
1354					5AP
1600					3AP
1652					3AP
1820					3AP
1888					4AP
2047					5AP
.....

Congiunzione tra classi e studenti:

Classe	Descrizione	Ubicazione	Matricola	Cognome	Nome
.....
3AP	Terza A Programmatori	Aula 23	1025			
3AP	Terza A Programmatori	Aula 23	1600			
3AP	Terza A Programmatori	Aula 23	1652			
3AP	Terza A Programmatori	Aula 23	1820			
4AP	Quarta A Programmatori	Aula 24	1245			
4AP	Quarta A Programmatori	Aula 24	1888			
5AP	Quinta A Programmatori	Aula 25	1354			
5AP	Quinta A Programmatori	Aula 25	2047			
.....

Si noti come il campo classe, presente in entrambe le tabelle, non è stato duplicato.

Per l'associazione Classi/Studenti, la tabella congiunta riporterà, per ciascuna classe, l'elenco degli studenti che la frequentano.

La cardinalità della tabella risultante non è definibile in maniera precisa e dipende dal tipo di join e dall'obbligatorietà o meno dell'associazione in entrambi i versi:

- **Equi join:** la tabella risultante contiene tutti i record della tabella madre che ha corrispondenti nella tabella figlia e tutti i record della tabella figlia che hanno il corrispondente nella tabella madre; tutti i clienti che hanno fatture e le fatture che corrispondono ad un cliente (tutte in quanto c'è integrità referenziale); tutte le fatture differite e solo le bolle fatturate;
- **Left join:** riporta tutti i record della tabella madre e solo quelli della tabella figlia che hanno il corrispondente nella tabella madre; riporta, per esempio, anche i clienti che non hanno fatture; riporta sia le fatture differite che immediate;
- **Right join:** riporta i record della tabella madre che hanno corrispondenti nella tabella figlia e tutti i record della tabella figlia; riporta, per esempio, anche le fatture che non hanno un corrispondente nella tabella madre; pessimo esempio! Perché? Rispondete voi. Altro esempio, riporta le fatture differite e tutte le bolle.

Questi ultimi due join sono riferiti anche come **join esterno (outer join)**.

- Un particolare tipo di join è il **self join** quando una tabella viene congiunta con se stessa; per esempio quando nella tabella articoli ogni record contiene il codice di un articolo alternativo, presente nello stesso archivio e che può essere proposto in caso di mancata disponibilità; in questo caso si potrebbe utilizzare un self join che riporti per ciascun articolo il suo alternativo.

Da notare che, in caso di obbligatorietà dell'associazione in entrambi i versi come nel nostro esempio, i tre tipi di join restituiscono tutti lo stesso risultato.

4. Altre operazioni, di minore importanza, sono **l'unione**, che unisce due tabelle omogenee accodando i record dell'una a quelli dell'altra, **l'intersezione**, che prende solo le righe, definite comuni in base ad una proprietà, di due tabelle omogenee, e la **differenza** che prende le righe della prima non presenti nella seconda.

Non vi svelo tutto ed in classe faremo numerosi esercizi sulla cardinalità e sul grado di tabelle risultanti da operazioni di selezione, proiezione e, soprattutto, congiunzione.

Ovviamente le operazioni possono essere combinate insieme per selezionare, congiungere e proiettare contemporaneamente ottenendo così con una sola operazione combinata, per esempio, l'elenco delle fatture dei clienti con saldo diverso da zero e che riporti soltanto il codice e qualche dato anagrafico del cliente, il numero, la data e l'importo delle fatture.

9.3.2. La normalizzazione

All'inizio del capitolo vi ho parlato dei vantaggi dei data base, quali l'eliminazione della ridondanza e dell'inconsistenza e garantire l'integrità dei dati; per raggiungere gli obiettivi di cui ad inizio capitolo e, soprattutto, per garantire i vantaggi appena riportati, l'organizzazione dei dati deve subire un **processo di normalizzazione** che passa attraverso varie **forme normali**; prima di elencare le varie forme normali, occorre dare e/o ripetere le seguenti definizioni:

- **Chiave o chiave primaria:** Uno o più attributi che consentono di individuare in maniera univoca una istanza o record; ad esempio il campo codice nell'archivio articoli costituisce una chiave primaria semplice mentre i campi corso, sezione e classe, nella tabella classi, costituiscono una chiave primaria composta;
- **Chiave candidata:** uno o più campi che possono essere utilizzati come chiave primaria; possiamo avere più chiavi candidate per una tabella ma una sola chiave primaria; per esempio, nella tabella CLIENTI, il codice, il codice fiscale e la partita IVA possono essere chiavi semplici candidate ma, tra queste, si sceglie il campo codice, o uno degli altri due, come chiave primaria;
- **Attributo non chiave:** qualsiasi attributo che non fa parte della chiave;
- **Dipendenza funzionale:** un attributo a_2 dipende funzionalmente dall'attributo a_1 quando un valore di a_1 determina un unico valore di a_2 e si indica con $a_1 \rightarrow a_2$; per esempio, dalla matricola dello studente dipende funzionalmente il suo nome;
- **Dipendenza funzionale transitiva:** $a_1 \rightarrow a_2$ e $a_2 \rightarrow a_3$; allora $a_1 \rightarrow a_3$ ovvero a_3 dipende funzionalmente da a_1 in modo transitivo.

Detto questo passo ad elencare le varie forme normali che costituiscono il processo di normalizzazione:

1. Prima forma normale: 1NF

Una relazione è in prima forma normale, denotata con 1NF, se:

- **Omogeneità delle righe:** tutte le righe della tabella sono costituite dallo stesso numero di colonne;
- **Omogeneità delle colonne:** tutte le colonne ovvero tutti gli attributi sono informazioni elementari, ovvero non ulteriormente scomponibili, e dello stesso tipo ovvero appartenenti allo stesso dominio (STRING, INT16, ci siamo capiti?); per esempio la tabella Fatture(Numero, data, totale, ArticoliVenduti) non è in prima forma normale perché l'attributo ArticoliVenduti non è elementare; la soluzione è quella di utilizzare una seconda tabella contenente gli articoli venduti e legata a Fatture da una associazione 1:m;
- **Righe non duplicate:** non esistono due righe che hanno gli stessi valori per tutti gli attributi;

- **Irrelevanza della posizione delle righe:** l'ordine con il quale le righe sono presenti all'interno della tabella è irrilevante.

2. Seconda forma normale: 2NF

Una tabella è in 2NF se è in 1NF ed ogni attributo non chiave dipende dall'intera chiave; naturalmente questa forma normale si applica ai casi in cui la chiave primaria è composta e garantisce l'assenza di attributi non chiave dipendenti da una parte della chiave. Si consideri il seguente esempio: Un'azienda ha le proprie merci in più depositi dislocati in diverse città o in più punti nell'ambito della stessa città; la relazione ARTICOLI(Codice, Deposito, Quantita, Indirizzo) deve necessariamente avere una chiave primaria composta in quanto né il codice dell'articolo, né deposito possono esserlo; la chiave può essere costituita da Codice + deposito; in questo caso, però, ci sono attributi che non dipendono dall'intera chiave in quanto l'indirizzo del deposito dipende solo da deposito; la relazione non è in 2NF; faccio esplicitamente notare che, se un deposito viene trasferito in altra località, sarà necessario modificare il suo indirizzo in tutti i record di quel deposito, ovvero tante volte quanti sono gli articoli presenti in esso. Il problema si risolve facilmente spezzando la relazione di cui sopra nelle due seguenti:

ARTICOLI(Codice, Deposito, Quantita) e Depositi(Deposito, Indirizzo)

3. Terza forma normale: 3NF

Una tabella è in 3NF se è in 2NF e tutti gli attributi non chiave dipendono dalla chiave ovvero non sono presenti attributi non chiave che dipendono da altri attributi non chiave; questa forma normale, come evidente, elimina la dipendenza funzionale transitiva. Un esempio può essere fatto in maniera estremamente semplice: è sufficiente aggiungere alla tabella fatture tutti gli attributi dei clienti e non, come detto in precedenza, solo la chiave della tabella madre CLIENTI ovvero solo il codice del cliente; in questo caso nella tabella fatture così fatta:

Fatture(Numero, Data, Totale, CodiceCliente, RagioneSociale, Indirizzo, Città, ...)

sarebbero presenti i campi (RagioneSociale, Indirizzo, Città, ...) che dipendono funzionalmente da CodiceCliente e non dalla chiave Numero. Come si risolve? Ma è ovvio! Anche in questo caso una variazione dei dati di un cliente, quale, per esempio, l'indirizzo, comporterebbe la necessità di aggiornare tutti i record della tabella fatture relativi a quel cliente; se, invece, questi dati sono presenti solo nella tabella CLIENTI, vengono aggiornati solo una volta evitando così la ridondanza ed una possibile inconsistenza nei casi in cui non tutti i record venissero correttamente aggiornati.

4. Altre forme normali: BCNF, 4NF, 5NF

Esistono altre forme normali superiori alla terza, in particolare vi menziono la forma normale di Boyce-Codd, BCNF, e la quarta e quinta forma normale, che ritengo di non approfondire in quanto, in genere, è sufficiente portare le relazioni in terza forma normale.

E' possibile riassumere l'appartenenza di una relazione o tabella alla terza forma normale dicendo: **una relazione è in 3NF se ogni suo attributo non chiave dipende dalla chiave, l'intera chiave, nient'altro che la chiave.**

9.4. DBMS

Il DBMS (Data Base Management System), come già detto, è l'insieme dei programmi messi a disposizione dei programmatori e degli utenti per semplificare la gestione della base dei dati. Il DBMS è un componente del software di base e garantisce tutti i vantaggi di cui ad inizio capitolo, sia in termini di sicurezza che di efficienza.

Il DBMS deve permettere di definire la struttura concettuale e logica del data base e consentire l'accesso e la manipolazione dei dati. Questi obiettivi vengono raggiunti mettendo a disposizione degli utenti i seguenti linguaggi:

- **DDL – Data Definition Language:** è il linguaggio che consente di definire lo schema concettuale e logico del database. Un utilizzatore o un programmatore potrebbe essere interessato solo ad una parte dei dati presenti nel data base, per cui il DDL consente anche la definizione di sottoschemi derivati dallo schema concettuale generale nel quale sono riportati solo i dati cui si è interessati; si parla di **sottoschema** o **vista**;
- **DML – Data Manipulation Language:** Il DML è costituito da istruzioni che consentono di gestire i dati presenti nel data base; il linguaggio DML può essere un vero e proprio linguaggio di programmazione che, con semplici frasi della lingua inglese, consente, anche ad utenti non programmatori, di accedere alla base dei dati; in questo caso si parla di **QL - Query Language**. Il DML può essere costituito da istruzioni che possono essere utilizzate all'interno di altri linguaggi di programmazione, detti linguaggi ospiti, e ne costituisce una estensione; in questo modo è possibile e comodo accedere alla stessa base dei dati utilizzando programmi scritti in linguaggi di programmazione diversi.
- **DCL – Data Control Language:** consente di definire i diritti di accesso al data base da parte dei vari utenti; in pratica consente l'implementazione dei criteri di sicurezza;
- **DMCL – Device Media Control Language:** è il linguaggio con il quale è possibile definire la struttura fisica del data base ovvero come i dati saranno

memorizzati sulle memorie di massa; l'efficienza del data base dipende in maniera decisiva da come i suoi dati vengono allocati sulle memorie di massa.

La gestione di un database appare compito assai delicato e complesso in quanto comporta una serie assai diversificata di competenze, dalla definizione dello schema concettuale e logico allo schema fisico passando attraverso i diritti degli utenti; a questo va aggiunta la necessità di mantenere il data base, ovvero eseguire tutte quelle operazioni che consentono una maggiore efficienza, soprattutto nelle operazioni di accesso e, più in generale, di trattamento dei dati in esso contenuti. Per questo motivo è necessaria una nuova figura professionale detta **DBA - Data Base Administrator**, che si occupa dell'amministrazione del data base assegnando i diritti agli utenti, mantenendo il data base, effettuando regolarmente operazioni di copie di back-up, ecc. .

In genere interagiscono con il data base anche altre figure professionali e non, come **l'utente finale** che può, se autorizzato, accedere ai dati per visione, modifica e quant'altro necessario senza avere una specifica ed approfondita conoscenza informatica, ma utilizzando semplici comandi messi a disposizione dal DBMS, specifiche interfacce grafiche semplici da utilizzare o istruzioni del query language di cui il DBMS dispone. Molto semplici sono le **QBE-Query By Example**, interrogazioni da esempio, che, attraverso una interfaccia molto semplice, consente all'utilizzatore non solo l'estrazione dei dati da una o più tabelle ma anche la loro modifica mediante inserimenti di nuovi dati, cancellazioni e aggiornamenti.

Ad un livello superiore troviamo il **programmatore** che interagisce con la base dei dati tramite il DML, utilizzando o un linguaggio specifico di programmazione messo a disposizione dal DBMS oppure un qualsiasi linguaggio di programmazione disponibile sul sistema, di cui il DML ne costituisce una estensione, e che gli consente di realizzare applicativi completi che utilizzano i dati presenti nel data base.

9.5. Esercizi

1. Riportare analisi, modello concettuale e logico per un database contenente le informazioni relative a:
 - a) Medico di base;
 - b) Studio associato di medici con gestione appuntamenti;
 - c) Biblioteca;
 - d) Videoteca;
 - e) Campionato di calcio di serie A;
 - f) Campionati di calcio;
 - g) Orario scolastico;
 - h) Anagrafe cittadina;
 - i) Gestione contabile di un'azienda;
 - j) Data base completo relativo a classi, materie, studenti con assenze e voti, docenti, orario scolastico.

Capitolo 10. Il linguaggio SQL

10.1. Il linguaggio SQL

Il linguaggio **SQL-Structured Query Language**, linguaggio strutturato per le interrogazioni, è ormai uno standard tra i linguaggi non procedurali **per la gestione di basi di dati relazionali**. Nato alla fine degli anni settanta, l'**ANSI (American National Standard Institute)** ne pubblica uno standard nel 1986 e nel 1987 è l'**ISO (International Standard Organization)** a pubblicarne uno standard. Da allora il linguaggio è stato molto ampliato ma la sua grande peculiarità è che ne esiste uno **standard** ben definito. Molte aziende che hanno implementato il linguaggio hanno aggiunto funzionalità al di là dello standard ma in ogni caso contengono tutto quanto definito in esso. Il linguaggio SQL consente la **creazione** di data base di tipo relazionale e la **manipolazione dei dati** in essi contenuti nonché la **gestione di criteri di protezione**; si tratta, pertanto, di un **DBMS completo** con le funzioni dei linguaggi **DDL, DML, DCL e Query Language**. Gli utenti possono inviare comandi al DBMS attraverso semplici frasi scritte rispettando la sintassi del linguaggio oppure utilizzare interfacce del tipo **QBE-Query By Example** o, per i programmatori, utilizzare i comandi SQL all'interno di altri linguaggi di programmazione, in questo caso si parla di **Embedded SQL**.

Molto di quanto studiato nel linguaggio VB.NET si applica al linguaggio SQL: gli operatori matematici e di relazione, gli operatori logici, gli identificatori di tabelle e attributi o campi, che seguono le regole per i nomi delle variabili, nonché il riferimento ad un attributo mediante **NomeTabella.NomeAttributo**.

I tipi dei dati consentiti sono:

Tipo	Descrizione	Note
Character(n)	Stringa di lunghezza n	1<=n<=15.000 abbreviato: Char(n) Char come char(1)
Date	Data	
Time	Ora	
Integer(p)	Numero intero con p cifre	1<=p<=45
TINYINT	Byte	Da -128 a + 127
SmallInt	Fixed point 16 bit	Da -32.768 a +32.767
Integer	Fixed point 32 bit	Da -2 ³¹ a +2 ³¹ -1
Decimal(p,s)	p cifre intere e s cifre	1<=p<=45 e 0<=s<=p

	decimali	
Real	Numero reale con precisione 7	
Float	Numero reale con precisione 15	Oppure DoublePrecision
Float(p)	Numero reale con precisione p	$1 \leq p \leq 45$

Le costanti stringhe sono racchiuse tra un solo apice ‘ o un doppio apice “, ad esempio ‘Totale fattura’ o “Totale fattura”.

Un attributo non inizializzato ha valore **null** che è diverso da qualsiasi altro valore; in altre parole non è uguale a 0 in caso di campo numerico o a stringa nulla (“) in caso di stringa; un valore null non è maggiore, né minore e né uguale a nessun altro valore; si tratta di un valore sconosciuto ed il sistema può solo stabilire se un campo contiene o meno un valore null. In caso di ordinamento i valori null compaiono all’inizio se l’ordinamento è crescente ed in fondo in caso contrario.

La maggior parte delle implementazioni del linguaggio SQL mettono a disposizione dell’utente interfacce grafiche che consentono la creazione ed il trattamento dei dati presenti nel data base; operazioni complesse facenti parte dei linguaggi DDL, DML, DCL possono essere comandate al sistema DBMS attraverso interfacce grafiche molto semplici, come la già menzionata QBE, e utilizzabili anche da utenti meno esperti e non professionisti dell’information technology.

Seguono i formati delle istruzioni più comuni del linguaggio SQL, riportando il comando ed una breve descrizione, il formato, uno o più esempi ed eventuali note. Per gli esempi utilizzerò i semplici data base descritti nel capitolo sui data base ovvero quello degli studenti e quello dei clienti.

Create Table: Creazione di una tabella;

Occorre specificare il nome della tabella da creare seguita dall’elenco degli attributi e del relativo tipo.

CREATE TABLE nome (attributo tipo parametri [,attributo tipo parametri]....);

Per creare la tabella studenti:

Create Table studenti(Matricola integer **primary key**, Cognome char(20) **not null**, Nome char(20) not null, ..., classe tinyint **default 0**);

Alter Table: Modifica della struttura di una tabella;

Consente di inserire e/o cancellare campi della tabella.

ALTER TABLE ADD attributo;	per aggiungere un campo
ALTER TABLE DROP attributo;	per eliminare un campo
Alter Table Studenti Add DataNascita date;	
Alter Table Studenti Drop PartIva;	

Create Index: Consente la creazione di un indice.
CREATE [UNIQUE] INDEX nomeindice ON table (attributo [,attributo]...);
Create index generalita on studenti (cognome, nome);
Crea l'indice associato alla chiave composta da cognome e nome.
L'opzione unique, se utilizzata, inserisce nell'indice, per ogni gruppo di record aventi lo stesso valore nei campi chiave, solo il primo.

Drop: Cancella una tabella o un indice.
DROP TABLE nome tabella;
DROP INDEX nome indice ON nome tabella;
Drop Table Studenti;
Drop Index generalita on studenti;

Insert: Aggiunta di un nuovo record ad una tabella; formato 1.
INSERT INTO nome tabella VALUES (valore1, valore2,, valoren);
Insert Into Studenti Values (1234,'Bianco','Mauro', ..., 1);
Quando con il comando insert si utilizza values, bisogna specificare tanti valori per quanti sono gli attributi della tabella alla quale si aggiunge il record, altrimenti si avrà una segnalazione di errore. E' possibile inserire un nuovo record specificando solo i valori di alcuni campi; in questo caso i campi non specificati vengono inizializzati con valori di default. Vedi istruzione successiva.
Ovviamente il valore specificato in una certa posizione deve essere accettabile in funzione del tipo del campo corrispondente.

Insert: Aggiunta di un nuovo record ad una tabella; formato 2.
INSERT INTO nome tabella (elenco campi) VALUES (elenco valori);
Insert Into Studenti (matricola, cognome, nome)

```
values (1234,'Bianco','Mauro');
```

Ovviamente il numero di valori deve essere uguale al numero di attributi specificati.

Update: Aggiornamento dei dati in una tabella.

UPDATE nometabella

```
SET attributo=expr [, attributo=expr]...
```

```
[WHERE condizione]
```

```
[LIMIT numerorighe];
```

E' possibile aggiornare uno o più campi di tutti i record della tabella che soddisfano una certa condizione; è possibile limitare la ricerca ad un certo numero di record.

Update studenti

```
set classe=2
```

```
where classe=1;
```

Update studenti

```
set numerotelefono='1234568'
```

```
where matricola=1234;
```

Update studenti

```
set classe=classe+1 and promosso;
```

Update articoli

```
set prezzo=prezzo*1.1
```

```
where categoria='PENNE';
```

Omettere la clausola where equivale a where=true.

Delete: Cancellazione dei dati in una tabella.

DELETE FROM nometabella

```
[WHERE condizione]
```

```
[LIMIT numerorighe];
```

Delete from studenti

```
where classe=5;
```

Delete studenti

```
Limit 3;
```

Delete articoli

```
where codice='PENNABIC';
```

'Elimina i primi 3 record

Select

L'istruzione select ha una grande flessibilità e consente all'utente di estrarre dati da una o più tabelle e utilizzare, anche in maniera combinata, le operazioni di selezione, proiezione e congiunzione oltre a raggruppamenti e calcoli.

<pre>SELECT [ALL DISTINCT] elencoattributi * FROM elencotabelle WHERE condizione;</pre>	
<p>Elencoattributi: elenco dei campi da visualizzare (Proiezione); se si vuole visualizzare tutti i campi è sufficiente utilizzare il simbolo “*”;</p> <p>Elencotabelle: elenco della/delle tabelle dalle quali estrarre i dati (Congiunzione);</p> <p>Condizione: valore booleano cui devono soddisfare i record affinché siano riportati nella tabella risultante (Selezione).</p> <p>ALL: Visualizza tutti i record ed è il valore di default;</p> <p>DISTINCT: Visualizza solo i record che risultano diversi (1NF); vedi esempio più avanti.</p>	
<pre>Select Matricola, Cognome, Nome From Studenti;</pre>	Proiezione
<pre>Select Matricola, Cognome, Nome From Studenti Where Citta='Gesualdo' and classe=1;</pre>	Proiezione e selezione
<pre>Select * From Studenti Where Corso='P' and sezione='A' and Classe=5;</pre>	
<pre>Select [ALL] Corso, Sezione, Classe From Studenti;</pre>	<pre>Select DISTINCT Corso, Sezione, Classe From Studenti;</pre>
<p>La prima interrogazione fornisce l'elenco delle classi, con corso e sezione, e ciascuna terna (Corso, Sezione, Classe) compare tante volte per quanti sono gli studenti che la frequentano per cui la tabella risultante non è in 1NF; nel secondo caso, invece, ciascuna terna (Corso, Sezione, Classe) compare una sola volta ovvero dei record uguali, stesso valore per tutti e tre i campi, ne viene riportato uno solo per cui la tabella risultante è in 1NF. In altre parole, dall'archivio degli studenti, si estrae l'elenco delle classi; così come, per esempio, (continue voi).</p>	
<p>Campi calcolati, utilizzo della clausola AS e utilizzo di parametri variabili</p> <p>Con il comando Select è possibile aggiungere campi calcolati a partire da quelli contenuti nella/e tabella/e di partenza e con la clausola AS è possibile cambiare l'intestazione della colonna nella tabella risultante come negli esempi che seguono:</p> <pre>Select Matricola, Cognome, Nome, classe as classe attuale, classe+1 as successiva From Studenti;</pre>	

```
Select Codice, Ragione_sociale, Citta, Dare, Avere, Dare – Avere as Saldo
```

```
From Clienti;
```

In tutti gli esempi finora riportati sono state utilizzate delle costanti per indicare i valori da inserire nei record, quelli da selezionare o cancellare, ecc.; in genere, come, per esempio, è possibile fare in MS-ACCESS, si possono utilizzare le parentesi quadre per indicare un parametro come nel seguente esempio:

```
Select Matricola, Cognome, Nome
```

```
From Studenti
```

```
Where Citta=[Immettere città] and classe=[Immettere classe];
```

In questo caso l'utente potrà inserire i valori da utilizzare nella select utilizzando due input box successivi, che il sistema farà comparire e nei quali compariranno i suggerimenti "Immettere città" e "Immettere classe". In questo modo la stessa query può essere utilizzata per tutte le città e per tutte le classi.

Congiunzione

Il comando select va al di là di quanto esposto e consente anche la realizzazione di operazioni più complesse come la congiunzione tra due o più tabelle, per le quali è stata definita una associazione. Vi dico subito che, se le tabelle hanno campi con lo stesso nome, è necessario specificarli con la notazione **nometabella.nomecampo** altrimenti si può utilizzare anche il solo nome del campo:

```
Select *
```

```
From clienti, fatture
```

```
Where codice=codicecliente;
```

equivale a:

```
Select *
```

```
From clienti, fatture
```

```
Where clienti.codice=fatture.codicecliente;
```

e fornisce l'elenco delle fatture per ciascun cliente.

```
Select clienti.codice, clienti.ragiosoc, fatture.numero, fatture.data, fatture.totale
```

```
From clienti, fatture
```

```
Where clienti.codice=fatture.codicecliente;
```

effettua anche una proiezione sui soli campi elencati.

```
Select *
```

```
From fatture,clienti
```

```
Where codicecliente=codice;
```

```
Select *
```

```
From clienti, fatture;
```

```
Select *
```

```
From clienti, fatture
```

```
Where codice=[codice cliente];
```

E' evidente come con una sola select sia possibile effettuare operazioni di congiunzione, proiezione e selezione contemporaneamente:

```
Select clienti.codice, clienti.ragiosoc, fatture.numero, fatture.data, fatture.totale
```

```
From clienti, fatture
```

```
Where clienti.codice=fatture.codicecliente and
```

```
clienti.codice=[Immettere codice cliente];
```

Self Join: il self join consente di congiungere una tabella con se stessa; riprendiamo l'esempio della tabella degli articoli di magazzino nella quale abbiamo il campo "CodiceAlt", riportante il codice di un articolo alternativo; tale codice fa riferimento ad un record presente nella stessa tabella articoli in quanto anch'esso un articolo di magazzino. Se volessi avere per ciascun articolo il corrispondente articolo alternativo, devo congiungere la tabella articoli su se stessa con il seguente comando SQL:

```
Select *
```

```
From articoli as art1, articoli as art2
```

```
Where Art1.CodiceAlt=Art2.Codice;
```

Art1 e Art2 sono cosiddetti alias ovvero nomi alternativi che fanno riferimento, in questo caso, alla stessa tabella.

Equi, Left, Right join

Nel capitolo relativo ai database vi ho parlato dell'operazione di congiunzione che può avvenire secondo tre modalità: equi, left e right. E' arrivato il momento di vedere come il linguaggio SQL ci consente di realizzare queste operazioni e lo vediamo, in maniera estremamente semplice, con degli esempi:

```
SELECT clienti.codice, fatture.numero, fatture.data, fatture.totale
      FROM clienti INNER JOIN fatture ON clienti.codice = fatture.codicecli;
```

Equi join: Clienti che hanno fatture, fatture che corrispondono a clienti.

```
SELECT clienti.codice, fatture.numero, fatture.data, fatture.totale
      FROM clienti LEFT JOIN fatture ON clienti.codice = fatture.codicecli;
```

Left join: restituisce le istanze della query precedente più i record relativi ai clienti che non hanno fatture.

```
SELECT clienti.codice, fatture.numero, fatture.data, fatture.totale
      FROM clienti RIGHT JOIN fatture ON clienti.codice = fatture.codicecli;
```

Right join: restituisce le istanze dell'equi join più i record relativi alle fatture non corrispondenti ad un cliente. Vi faccio notare che abbiamo applicato l'integrità referenziale per cui non esistono istanze di fatture non corrispondenti ad una istanza di clienti; in altre parole equi e right join, in questo caso, sono identici.

E' importantissimo notare come con un semplice comando SQL sia possibile estrarre dati da una o più tabelle in maniera estremamente semplice. Se volessimo ottenere lo stesso risultato utilizzando un linguaggio di programmazione tradizionale, dovremmo scrivere un programma con numerose righe di codice **con uno, due o addirittura più iterazioni nidificate**. Avremo modo di verificare quanto appena detto proponendo algoritmi in VB.NET, realizzati utilizzando le istruzioni per il trattamento dei file ad accesso diretto, e la corrispondente semplice query in SQL.

INTO e INSERT INTO

E' possibile creare una nuova tabella, costituita dai record estratti mediante una select, utilizzando la clausola INTO:

```
Select * INTO StudGesualdo
      From studenti
      Where citta="Gesualdo";
```

Crea la tabella o relazione StudGesualdo contenente gli studenti che abitano a Gesualdo.

E' anche possibile aggiungere i dati di una select ad una tabella esistente:

```
INSERT INTO Studenti
```

```
    Select *
```

```
    From NuoviIscritti;
```

inserisce nella tabella studenti tutti i record della tabella NuoviIscritti.

Funzioni di aggregazione

Pensavate che il discorso Select fosse finito? Neanche per idea!

Nell'ambito di una select è, infatti, possibile utilizzare le seguenti funzioni di aggregazione:

- Count: Conteggio dei campi contenenti valori non null ad eccezione di Count(*) che conta anche tali valori;
- SUM: Somma, i campi null vengono considerati 0;
- AVG: Media aritmetica, i campi null vengono ignorati;
- MIN, MAX: Minimo e massimo, valori null ignorati;

E' possibile utilizzare anche la clausola DISTINCT che conta una sola volta i valori uguali contenuti nella colonna oggetto del conteggio oppure espressioni ottenute sui campi della tabella.

```
Select Count(Telefono)
```

```
    From Studenti;
```

restituisce il numero dei record della relazione studenti che nel campo telefono hanno un valore diverso da null ovvero gli studenti che hanno un numero di telefono.

Select Count(*)

From Studenti;

Conta tutti i record della tabella studenti senza gestire il valore null, per cui il valore restituito è la cardinalità della relazione.

```
Select Count(*)
```

```
    From Clienti
```

```
    Where citta='AVELLINO';
```

Conta i client residenti in AVELLINO.

```
Select Count(Distinct citta)
```

From Clienti

Where Provincia="AV";

Restituisce il numero dei comuni in cui risiedono i clienti della provincia di AV.

Select SUM(Totale)

From fatture

Where Codicecliente="Rossi";

Somma dei totali delle fatture del cliente Rossi.

Select SUM(dare-avere)

From Clienti;

Somma saldi di tutti i clienti.

Select AVG(Totale)

From fatture;

Media dei totali delle fatture.

Select MAX(Totale), MIN(Totale)

From fatture;

Fattura di importo massimo e minimo.

Select Min(Cognome), Max(Cognome)

From Studenti;

Cognome minimo e massimo secondo l'ordine alfabetico; in altre parole Max e Min possono accettare qualsiasi tipo di dato.

Ordinamenti e raggruppamenti

ORDER BY attributo [ASC|DESC]: Con la clausola order by è possibile ordinare le righe della tabella risultante in maniera crescente (ASCending) o decrescente (DESCending) sui valori di uno o più campi; se non specificato, l'ordinamento è crescente.

Select cognome, nome, datanascita, matricola

From studenti

Order By cognome, nome, datanascita;

Select cognome, nome, datanascita, matricola

From studenti

Order By datanascita DESC, cognome, nome;

GROUP BY elenco attributi: Consente il raggruppamento di record che hanno lo stesso valore nel/i campo/i specificati; per ogni raggruppamento, nella tabella risultante, sarà presente una sola istanza; per esempio, è possibile raggruppare le fatture di un cliente e sommarne i totali per ottenere il fatturato di ciascun cliente e/o mediante count(*) il conteggio:

```
SELECT fatture.codicecli, Sum(fatture.totale) AS Fatturato, Count(*) AS Fatture
FROM fatture
GROUP BY fatture.codicecli;
```

Se si usa la clausola group by all'interno di una select, è necessario che tutti i campi elencati dopo select siano riportati nella clausola group by oppure in una funzione di aggregazione.

La seguente select:

```
SELECT year(data), month(data), Sum(fatture.totale) AS Fatturato, Count(*) AS Fatture
FROM fatture
WHERE year(data)=[Immettere anno]
GROUP BY Year(data), Month(data);
```

Fornisce i fatturati ed il numero delle fatture riferiti ai mesi di un anno scelto dall'utilizzatore. Provate a scrivere il codice in VB.NET o qualsiasi altro linguaggio di programmazione!

La clausola **Having** consente di effettuare una ulteriore selezione sulle istanze ottenute dalla select:

```
SELECT Year(data) as Anno, Month(data) as Mese, Sum(fatture.totale) AS
Fatturato, Count(*) AS Fatture
FROM fatture
WHERE Year(data)=[Immettere anno]
GROUP BY Year(data), Month(data)
```

HAVING Sum(fatture.totale)>7000;

restituisce solo i mesi con fatturato maggiore di 7000; lo stesso risultato lo si può ottenere con una select su una select:

SELECT anno, mese, Fatturato, Fatture

FROM (SELECT Year(data) as Anno, Month(data) as Mese, Sum(fatture.totale) AS Fatturato, Count(*) AS Fatture

FROM fatture

WHERE Year(data)=[Immettere anno]

GROUP BY Year(data), Month(data))

WHERE Fatturato>7000;

In altre parole, Where agisce sui record delle tabelle di origine effettuandone una selezione e Having fa la stessa cosa sui record della tabella risultante.

Operatori

Il linguaggio SQL consente l'utilizzo degli operatori matematici, di relazione e logici che già conosciamo; è inoltre possibile utilizzare anche i seguenti:

- **BETWEEN**: per verificare se un valore è compreso tra due estremi, anch'essi inclusi; è possibile utilizzare anche la forma **NOT BETWEEN**; "Between 5 and 10" indica un valore compreso tra 5 e 10:

Select Matricola, Cognome, Nome

From studenti

Where DataNascita Between 01/01/1990 and 12/31/1990;

Si noti che per le date si us il formato mm/gg/aa.

- **IN**: verifica l'appartenenza di un valore ad un elenco; si può utilizzare NOT IN:

Select Codice, RagioneSociale, Telefono

From Clienti

Where Provincia IN ('AV', 'BN', 'CE', 'NA', 'SA');

- **LIKE**: Consente di confrontare un valore con una maschera che contiene **caratteri jolly**; vi ricordate i caratteri jolly * e ? di cui vi ho parlato nel capitolo relativo al sistema operativo Windows? Ebbene la logica è la stessa in SQL ma i caratteri jolly sono "_", jolly su una sola posizione, e "%", jolly che si ripete nelle posizioni successive:

- LIKE 'ROS%' combina con tutte le parole che cominciano con 'ROS' come

ROSSI, ROSSINI, ROSSetti, ecc.;

- LIKE ‘%CO’ combina con tutte le parole che terminano con ‘CO’ come BIANCO, TELEMACO, ecc.;

- LIKE ‘%NE%’ combina con tutte le parole che contengono ‘NE’ come VENEZIA, PONENTE, ecc.;

- LIKE ‘MA_E’ combina con le parole che cominciano per MA, poi un sol carattere qualsiasi e quindi una E, come MARE o MAUE; mentre la parola MAREA non combina;

Select Matricola, Cognome, Nome

From studenti

Where Cognome like ‘RO%’;

- **IS NULL**: restituisce vero se un campo ha valore null e falso altrimenti.

Select Matricola, Cognome, Nome

From studenti

Where Nome is not null; ‘oppure not (nome is null)

La sicurezza

I comandi **GRANT** e **REVOKE** consentono, rispettivamente, di assegnare o revocare diritti agli utenti:

GRANT lista permessi ON NomeTabella TO lista utenti

E’ possibile dare diritti a partire da una semplice colonna di una tabella fino a tutti i database presenti su un server. Alcuni permessi che è possibile concedere sono:

- **ALTER**: consente l'uso del comando ALTER TABLE, ma non è possibile creare o eliminare indici;
- **CREATE**: consente l'uso di CREATE TABLE e CREATE DATABASE;
- **DELETE**: consente l'uso del comando DELETE;
- **DROP**: consente l'uso di DROP TABLE e DROP DATABASE;
- **INSERT**: consente l'uso del comando INSERT;
- **SELECT**: consente l'uso del comando SELECT;
- **UPDATE**: consente l'uso del comando UPDATE;
- **INDEX**: consente la creazione ed eliminazione di indici;
- **ALL PRIVILEGES**: concede tutti i privilegi possibili.

Grant update

On studenti

To utente1, utente2;

Revoke Delete
 On studenti
 To utente1, utente2;

Grant update (Cognome, Nome)
 On studenti
 To utente1, utente2;

Viste – View

Una **vista** o **view** è un sottoinsieme dei dati presenti nel database. In genere gli utenti non sono interessati a tutto il contenuto del data base ma solo a una sua parte; analogamente un programmatore, che deve sviluppare un applicativo su una base di dati, ha necessità di utilizzare solo alcune o parti di tabelle che costituiscono il data base. Per esempio, un programmatore, che deve produrre una stampa delle assenze degli studenti, non sarà interessato alla tabella dei docenti o altre tabelle che non siano quella degli studenti e delle assenze. **Una vista, pertanto, consente di vedere un sottoinsieme del contenuto del data base ovvero consente di definire un sottoschema dello schema generale.** In questo modo l'utilizzatore o il programmatore non viene distratto, nella gestione dei dati, da informazioni non necessarie ai fini dello scopo che si prefigge.

Una vista viene creata con il comando **create view** che consente anche l'assegnazione di un nome alla vista. Su una vista è possibile operare con gli stessi comandi visti fino ad ora; la vista si riempie con i dati provenienti dal data base ed una modifica ai dati della vista vengono riportate nelle corrispondenti tabelle del data base, a meno che non vengano utilizzate le funzioni di aggregazione nel qual caso, per ovvi motivi, non sarà possibile trasferire le modifiche dalla vista al data base.

```
Create view nome
  As select ElencoAttributi
  From Elenco Tabelle
  Where condizione;
```

Vi dico subito che il formato del comando select può essere utilizzato con tutte le varianti e le clausole viste in precedenza.

```
Create view AssenzeStudenti
  AS Select studenti.*, Assenze.*
  From Studenti, Assenze;
```

```
Create view ClientiDare
  AS Select Clienti.Codice, Clienti.RagioneSociale, Clienti.Telefono,
```

```
Fatture.Data, Fatture.Totale
From Clienti, Fatture
Where Clienti.Dare-Clienti.Avere>0
```

Una vista può essere cancellata con il comando DROP VIEW:

```
Drop View ClientiDare;
```

E' possibile utilizzare il comando GRANT per assegnare diritti agli utenti sulle viste in modo analogo a quanto abbiamo visto per le tabelle.

Interrogazioni nidificate o subquery

E' ovviamente possibile utilizzare una select all'interno di un'altra select come nell'esempio che segue e che ho già riportato in precedenza:

```
SELECT anno, mese, Fatturato, Fatture
FROM (SELECT Year(data) as Anno, Month(data) as Mese, Sum(fatture.totale) AS
      Fatturato, Count(*) AS Fatture
      FROM fatture
      WHERE Year(data)=[Immettere anno]
      GROUP BY Year(data), Month(data))
WHERE Fatturato>7000;
```

con riferimento ad un anno immesso dall'utente, restituisce i mesi con fatturato maggiore di 7000 ed il relativo numero di fatture.

Se volessimo avere l'elenco dei clienti con fatturato maggiore di un certo valore potremmo seguire tre strade:

1. Realizzare una query su una query: il risultato di una query è una tabella sulla quale posso lavorare come tutte le altre tabelle del database;
2. Utilizzare la clausola Having:

```
SELECT Clienti.*, Sum(fatture.totale) AS Fatturato
      FROM Clienti INNER JOIN Fatture on Clienti.Codice=Fatture.Codicecli
      GROUP BY Cliente.Codice
      HAVING Sum(fatture.totale)>[Totale fatturato];
```

3. Utilizzare una query nidificata:

```
SELECT Clienti.*, fatturato
```

```
From (Select clienti.*, Sum(fatture.totale) AS Fatturato
      FROM Clienti INNER JOIN Fatture on Clienti.Codice=Fatture.Codicecli
      GROUP BY Cliente.Codice)
```

Where fatturato > [Totale fatturato];

Se volessi l'elenco delle fatture il cui totale è superiore alla media dei totali delle fatture potrei utilizzare la seguente query:

```
Select Fatture.*
```

```
From Fatture
```

```
Where Fatture.Totale > (select AVG(Fatture.Totale)
                        From Fatture);
```

La seconda select restituisce, ovviamente, uno scalare.

Molto più difficile, ma sicuramente più significativa, è la query che restituisce l'elenco degli studenti con un numero di assenze superiore alla media:

```
Select studenti.matricola, assenze
```

```
From (Select studenti.matricola, count(*) as assenze
```

```
      From Studenti Inner Join Assenze on Studenti.Matricola=Assenze.Matricola
      Group By Studenti.Matricola)
```

```
Where assenze > (Select Avg(Assenze)
```

```
      From (Select studenti.matricola, count(*) as assenze
```

```
            From Studenti Inner Join Assenze on Studenti.Matricola=Assenze.Matricola
            Group By Studenti.Matricola));
```

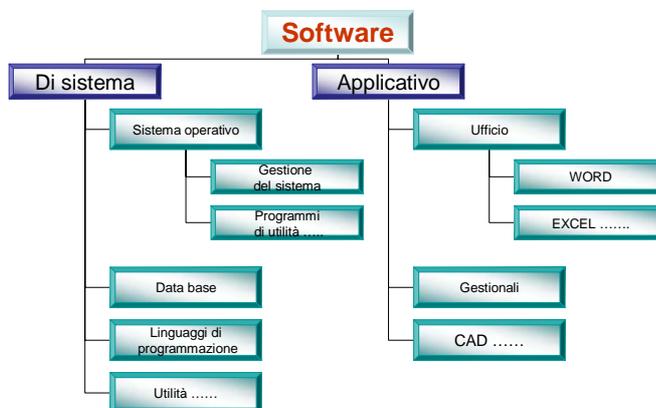
Capitolo 11. Software

11.1. Introduzione e non solo

Sono ormai acquisiti i concetti di Hardware e Software; con riferimento all'hardware molto è stato detto nei capitoli relativi agli anni precedenti mentre per il software è arrivato il momento di approfondire le vostre conoscenze. Il software è classificato in **software di base** e **software applicativo**. Il software di base è necessario per il corretto utilizzo del sistema di elaborazione e, in genere, si compone del sistema operativo e di una serie di programmi di utilità, anche se questi oggi sono integrati nel sistema operativo; c'è la tendenza, oggi, a identificare il software di base con il sistema operativo che, sicuramente ne è la componente più importante; in effetti una serie di programmi di utilità che venivano ritenuti fuori del sistema operativo, oggi ne fanno parte integrante ma, comunque, altri software ugualmente importanti, sono nel software di base ma non nel sistema operativo. Per esempio il DBMS è di base ma non parte integrante del sistema operativo con il quale, ovviamente, dialoga perché è quest'ultimo che si occupa dell'allocazione dei file e delle cartelle sulle memorie di massa; così come gli ambienti di sviluppo, quale il Visual Studio e tutti gli altri linguaggi di programmazione, utilizzati per sviluppare il software applicativo (e non solo).

Il software applicativo, invece, è quello che risolve i problemi dell'utente finale e, quindi, se vogliamo è quello a cui l'utente è più interessato; ma il software applicativo non può essere utilizzato in maniera semplice, correttamente ed efficacemente senza un buon software di base. Fanno parte di questa categoria i centinaia di migliaia di software sviluppati nel corso degli anni e ne fanno parte quelli destinati alle aziende, ai professionisti, ma anche all'utenza domestica come i giochi, le enciclopedie, ecc; in pratica non esiste una sola area di interesse umano per la quale non sia stato prodotto del software applicativo.

Classificazione del SW



Possiamo, pertanto, dire che il software applicativo risolve o aiuta a risolvere i problemi dell'utenza finale con l'ausilio di un software di base che ne consente l'esecuzione e la produzione nonché la cosiddetta elaborazione ausiliaria; per elaborazione ausiliaria dei dati si intende tutta quella serie di operazioni che non servono direttamente per la loro elaborazione ma che la completano e mi riferisco, per esempio, alle copie di back-up, all'ottimizzazione ed al controllo delle risorse del sistema, ecc.

Pertanto in questo capitolo dedicato interamente al software, vi descriverò:

- Il software di base e, soprattutto, l'organizzazione di un sistema operativo;
- la produzione del software applicativo, di cui abbiamo parlato spesso nel corso di questi anni, ma ora bisogna mettere ordine e capire che la produzione di un buon software applicativo è piuttosto complessa e le fasi di cui si compone sono numerose e articolate, lineari e trasversali;
- la valutazione del software con criteri che ne valutano la sua bontà; il software, pertanto, come qualsiasi altro bene o servizio va valutato secondo una serie di parametri che ne indicano l'efficienza.

Dopo aver parlato diffusamente e superficialmente dei sistemi operativi nei capitoli precedenti, è arrivato il momento di approfondire le conoscenze sul software fondamentale per un sistema di elaborazione; fondamentale perché è sicuramente il maggior responsabile dell'efficienza del sistema di elaborazione e della semplicità con la quale l'utente riesce, correttamente e proficuamente, ad utilizzarlo.

Il sistema operativo può essere definito come il gestore delle attività e delle risorse di un sistema di elaborazione. Una risorsa è qualsiasi dispositivo hardware o software necessario per l'esecuzione dei programmi; posso farvi numerosi esempi di risorse hardware tra le quali, per prima, occorre riportare la CPU ma, ovviamente,

anche RAM, memorie di massa fino a stampanti, tastiere, mouse etc.; una risorsa software è sicuramente il sistema operativo; ho già detto che nessun software applicativo può essere correttamente utilizzato o semplicemente eseguito senza il software di base e, quindi, della sua componente principale: il sistema operativo. Una attività può essere definita come l'esecuzione di un programma, detta processo, che coinvolge o meno l'utilizzatore del sistema. Un sistema operativo, quindi, gestisce risorse e processi e per fare questo deve anche, necessariamente, disporre di un'interfaccia, detta shell, verso l'utilizzatore che ne semplifichi l'accesso e l'utilizzo.

Le funzioni principali di un sistema operativo, pertanto, possono così essere riepilogate:

- **Gestione delle risorse;**
- **Gestione delle attività e dei processi;**
- **Interfacciare l'utilizzatore.**

Gestire le risorse significa ottimizzarne il loro utilizzo; le risorse hardware sono caratterizzate da tempi di lavoro e caratteristiche intrinseche che le rendono completamente diverse; per i tempi si va dai miliardesimi di secondo della CPU ai minuti necessari per produrre una stampa; è evidente che se la CPU è costretta, durante un'operazione di stampa, a lavorare con i tempi della periferica, il suo utilizzo rasenta lo zero; in altre parole la risorsa CPU è sottoutilizzata. Riuscire a ottimizzare risorse così diversificate ottimizzando, nel complesso, i tempi complessivi di elaborazione, è un compito assai difficile e delicato che il sistema operativo deve svolgere.

Le risorse possono essere classificate secondo diversi parametri in:

- **Attive:** le risorse attive elaborano l'informazione, come per esempio la CPU;
- **Passive:** le risorse passive non effettuano operazioni sui dati; le memorie sono risorse passive perché si limitano a memorizzare i dati ed a restituirli a richiesta;
- **Interrompibili:** una risorsa, come la CPU, è interrompibile nel senso che è possibile interrompere un lavoro per eseguirne un altro e poi ritornare su quello interrotto come se l'interruzione non fosse mai avvenuta; si tratta di un concetto importante da non confondere con la terminazione dell'esecuzione di un programma; per esempio, la stampante non è interrompibile perché, nel caso, la stessa stampa verrebbe prodotta su fogli non contigui o, addirittura, su fogli condivisi tra diverse stampe;
- **Condivisibili:** Una risorsa condivisibile può essere utilizzata da un processo in maniera concorrente con altri processi; ad esempio un disco rigido può essere condiviso tra più processi in esecuzione nel sistema perché essi possono utilizzare ovvero leggere e scrivere su aree diverse dello stesso volume;

- **Dedicate:** una risorsa dedicata, come ad esempio una stampante, può essere utilizzata per un solo lavoro alla volta; in altri termini il lavoro successivo può iniziare solo quando termina il precedente;
- **Virtuali:** si tratta di periferiche non reali che sembrano tali grazie ai meccanismi messi in atto dal sistema operativo; per esempio, il sistema operativo sostituisce la stampante, periferica dedicata, con il disco rigido, condivisibile, in modo tale che un processo può procedere all'operazione di stampa, anche se la stampante è impegnata in un altro lavoro, semplicemente dirottando la stampa in un'area del disco, per poi procedere alla sua stampa effettiva appena la stampante avrà terminato il lavoro corrente.

Per ottimizzare la gestione delle risorse, il sistema operativo adotta soluzioni assai diversificate a seconda della loro tipologia ma, il miglior risultato lo si ottiene grazie alla multiprogrammazione. Un sistema monoprogrammato consente l'esecuzione di un solo programma alla volta; pensate, non è possibile eseguire una nuova attività se prima non termina l'esecuzione di quella in corso; si tratta di sistemi obsoleti che, per esempio, negli anni ottanta utilizzavano il sistema operativo MS-DOS; vi dico solo che a quei tempi, una operazione di stampa, in pratica, congelava il sistema perché bisognava attendere la sua terminazione, che durava anche alcuni minuti o ore, prima di poter utilizzare il sistema per altre attività.

Nei sistemi multiprogrammati, invece, più programmi sono in esecuzione concorrentemente; concorrentemente non significa contemporaneamente ma solo che i tempi complessivi di esecuzione dei programmi si sovrappongono e che la CPU passa dall'esecuzione di un programma all'altro con interruzioni, che, ripeto, è la momentanea sospensione dell'esecuzione effettuata in maniera tale che essa possa riprendere successivamente come se l'interruzione non fosse mai avvenuta.

Grafico

In pratica i programmi sono come dei concorrenti che si contendono la CPU e le altre risorse del sistema allo scopo di procedere nell'esecuzione fino alla loro terminazione. Il numero massimo di programmi che il sistema riesce ad elaborare concorrentemente è detto grado.

Oggi è possibile anche una elaborazione contemporanea di più processi perché molte CPU, come quelle Core 2 e Core Quad della Intel, sono costituite da più CU, Control Unit.

Multitask

Le interruzioni possono essere interne ed esterne;

I sistemi sono ulteriormente classificati in monoutente e multiutente; nei sistemi monoutenti, come ad esempio i personal computer, una sola persona alla volta può utilizzare il sistema anche se questa unica persona può eseguire più programmi; nei sistemi multiutente, più persone sono collegati allo stesso sistema di elaborazione, in genere mediante terminali costituiti da una tastiera ed un monitor, ed ognuno di essi

può eseguire più programmi che vengono elaborati dalla stessa unità centrale; i terminali non hanno potenza di calcolo.

E' evidente che un sistema multiutente è necessariamente multiprogrammato ma non sempre è vero il viceversa; per esempio Windows è un sistema multiprogrammato ma non multiutente.

La gestione di un sistema multiprogrammato e/o multiutente è sicuramente molto complessa e lo scopo dei paragrafi successivi è proprio quello di chiarire alcuni aspetti di questi sofisticati sistemi di elaborazione.

11.2. Sistema operativo

I primi sistemi di elaborazione erano essenzialmente hardware mentre la componente software era assai limitata o addirittura assente; nel corso del tempo questa situazione è profondamente cambiata; il software ha assunto posizioni sempre più importanti e, oggi, possiamo dire prevalente; molto spesso le funzioni del software sono talmente sofisticate che si è portati a pensare che sia l'hardware ad eseguirla; è chiaro che è l'hardware ad eseguire le istruzioni dei programmi ma a sua volta è il software o, meglio, il sistema operativo che gestisce ed ottimizza la gestione delle risorse.

I moderni sistemi operativi operano oggi secondo le seguenti importanti modalità:

- **Batch o elaborazione a lotti:** i dati sono accumulati per un certo periodo di tempo e quindi elaborati; per esempio la fatturazione differita a fine mese;
- **Real time o tempo reale:** i dati sono elaborati con immediatezza producendo subito informazione; il termine immediatezza non può essere quantizzato in termini di tempo perché varia e deve essere adeguato al tipo di elaborazione che si sta eseguendo; per esempio nell'emissione di una fattura immediata qualche minuto è un tempo accettabile, mentre non lo è un'ora; l'immediatezza dell'esecuzione di una operazione ad uno sportello bancario è quantizzabile in qualche minuto e non di più altrimenti.....; se lo shuttle sta rientrando da una missione spaziale e va fuori rotta, probabilmente non riuscire a ritornare sulla stessa traiettoria nel giro di qualche secondo espone lo shuttle e, soprattutto, il suo contenuto umano a seri rischi; in questo caso l'individuazione del problema e la sua correzione deve avvenire in un tempo reale molto breve;
- **Transazionale:** una transazione è costituita da una serie di operazioni che vengono eseguite tutte o nessuna; si tratta di un concetto molto importante: si pensi ad un bonifico bancario che viene eseguito per metà ovvero avviene il prelievo su un conto e non l'accredito su un altro; e, ancora, nel caso dell'emissione di una fattura occorre modificare il contenuto di numerose tabelle quali fatture, magazzino, clienti, scadenziario, ecc; tali aggiornamenti non possono

avvenire contemporaneamente ma secondo una serie successiva di operazioni ben individuate e programmate dal programmatore; se gli aggiornamenti vengono eseguiti solo in parte si crea una inconsistenza nel data base che, in genere, non può essere risolta neanche eliminando la fattura emessa. Così il programmatore racchiude le operazioni da eseguire tra un <begin transaction> (<inizio transazione>) ed un <end transaction> (<fine transazione>) ed il sistema non procede alla scrittura degli aggiornamenti nel data base finchè non arriva l'indicazione della fine della transazione detta COMMIT. I sistemi sono così sofisticati e, direi, perfetti, che anche nel caso di interruzione accidentale dovuta, per esempio, alla mancata alimentazione, è in grado di ripristinare i dati nel data base allo stato precedente l'inizio della transazione tramite apposite procedure di roll-back che partono in automatico al riavvio del sistema. Così le operazioni richieste, per esempio per l'aggiornamento di un data base, comprese in una transazione ovvero tra Begin Transaction e End Transaction, se correttamente eseguite vengono chiuse con un comando COMMIT oppure tutto viene riportato allo stato precedente l'inizio della transazione mediante una procedura di Roll-Back.

I moderni sistemi operativi possono operare anche secondo diverse o tutte le modalità descritte a seconda del tipo di elaborazione richiesta.

Oggi la maggior parte delle elaborazioni avvengono in tempo reale; si pensi ad una banca che procede alla elaborazione dei saldi dei propri clienti ogni fine mese, probabilmente si accorgerebbe che è il fallimento!!!! Le elaborazioni di tipo real time richiedono apparecchiature hardware molto più sofisticate e, in genere, telematiche; pensate all'incasso di un assegno su una banca diversa da quella sulla quale l'assegno è emesso oppure ai prelievi bancomat o al pagamento del pedaggio autostradale mediante carta di credito, che richiede la verifica immediata sul conto corrente dell'intestatario della carta della disponibilità dell'importo da pagare, e

Secondo lo schema di Onion-Skin, i moderni sistemi di elaborazione sono organizzati in livelli, ognuno dei quali può dialogare solo con il precedente ed il successivo:



Per la macchina hardware ho già detto molto nel corso del primo anno e non ritengo di approfondire ulteriormente;

Il BIOS

Per poter gestire numerose attività in maniera concorrente e le risorse in maniera ottimale, il sistema operativo fa uso di macchine virtuali; una macchina virtuale è costituita da un sottoinsieme delle risorse reali del sistema di elaborazione e si occupa della gestione di una problematica del sistema operativo come, ad esempio, la gestione dei processi, oppure di un utente o di un processo. Il concetto di virtuale è molto utilizzato in informatica e, in genere, per virtuale, si intende una qualsiasi risorsa che appare reale solo grazie ai meccanismi messi in atto dal sistema operativo che reindirizza la risorsa o la rende reale sfruttando concetti quali la condivisione e l'interruzione. Così, per esempio, ad un utente di un sistema multiutente sembra che il sistema stia lavorando esclusivamente per lui e ad un utente di un sistema multiprogrammato sembra che il PC faccia più cose contemporaneamente quando riesce ad ascoltare la musica mentre sta stampando un documento, elaborando un foglio di calcolo e ricevendo un fax; in realtà il sistema sta gestendo macchine virtuali ripartendo il suo tempo e condividendo le risorse tra i vari processi.

11.2.1. Nucleo

Il nucleo del sistema operativo è quella componente sempre presente in memoria centrale; il sistema operativo non può, in genere, essere tutto caricato in memoria centrale perché questa ultima potrebbe essere insufficiente o comunque avere poco spazio restante per l'esecuzione dei programmi applicativi che è l'obiettivo da raggiungere; in altri termini un buon sistema operativo deve consentire l'esecuzione del maggior numero di programmi applicativi nell'arco di un certo tempo, perché sono i programmi applicativi che elaborano i dati dell'utilizzatore. Il Throughput, infatti, è un indice delle prestazioni del sistema operativo e indica il numero di lavori

eseguiti nell'unità di tempo; un altro importante parametro è il **System Overhead**, che indica il tempo totale che la CPU dedica ai processi del sistema operativo; tale valore, in genere, non deve superare il 10% del tempo complessivo di utilizzo della CPU. In pratica, essendo il sistema operativo un software, anch'esso utilizza la risorsa CPU; se la CPU esegue spesso processi del sistema operativo, i programmi applicativi avanzano lentamente e, quindi, le prestazioni del sistema sono scarse.

Il nucleo del sistema operativo si occupa della gestione dei processi ovvero dell'esecuzione dei programmi applicativi. Un processo può essere eseguito in foreground, ovvero in modo visibile per l'utente, o background, senza che l'utente se ne accorga.

Per evitare che un programma monopolizzi la CPU, non consentendo agli altri di avanzare nell'esecuzione, il sistema operativo utilizza la tecnica del time sharing o ripartizione di tempo, secondo la quale ad ogni processo viene assegnato un certo tempo della CPU, detto Quanto di tempo o time slice e dell'ordine di qualche decina di millisecondi, scaduto il quale la CPU passa all'esecuzione di un altro programma interrompendo quello in esecuzione; in questo modo la CPU passa dall'esecuzione di un programma all'altro e, per l'utente, è come se la CPU eseguisse contemporaneamente tutti i programmi o, nei sistemi multiutente, come se il sistema lavorasse solo per noi ma ciò è possibile solo grazie alla enorme velocità della CPU che, ripartendo il suo tempo tra i vari utenti ed i vari processi, dà l'impressione di svolgerli contemporaneamente.

Un processo, però, può essere interrotto anche per altre cause e, in generale, si classificano in:

- **interruzioni interne:** quando dovute a cause interne al programma come, ad esempio, una richiesta di operazione di Input/Output; le operazioni di I/O richiedono l'accesso ai dispositivi periferici che, per quanto veloci, richiedono sempre tempi molto lunghi per cui la CPU demanda il compito di eseguire tale operazione ad un canale o al DMA e passa all'esecuzione di un altro programma; per informazioni sul DMA si veda il capitolo relativo all'hardware. Un canale è, in genere, un sottosistema di elaborazione costituito, quindi, da una propria CPU ed una propria memoria; si tratta, pertanto, di un sistema che ha capacità elaborative proprie e che viene utilizzato esclusivamente per la gestione dei dispositivi periferici; in questo modo la CPU è libera dal dialogo con i dispositivi periferici, che sono molto più lenti;
- **interruzioni esterne:** dovute a cause esterne al programma quali la scadenza del time slice o alla richiesta di interruzione proveniente da un dispositivo periferico.

In genere i dispositivi periferici possono interrompere la CPU per comunicare, per esempio, l'avvenuto completamento di una operazione di I/O o per una richiesta dell'utente (si pensi alla pressione dei tasti CTRL+ALT+CANC). In pratica, per poter gestire o, come si suole dire, servire le richieste provenienti dai dispositivi periferici, la CPU può continuamente interrogarli con evidente perdita di tempo in relazione alle interrogazioni su dispositivi che non hanno bisogno di alcun servizio, oppure essere

interrotta dai dispositivi periferici quando hanno bisogno di gestire una operazione di I/O; la soluzione adottata è proprio quest'ultima. Ogni dispositivo può interrompere la CPU, che, quindi, esegue un programma adatto per servire la richiesta; è evidente che la richiesta di servizio porta informazioni sul dispositivo interrompente nel senso che la CPU è in grado di capire quale dispositivo richiede il servizio. Da questo punto di vista le interruzioni possono essere ulteriormente classificate in:

- **interruzioni mascherabili:** si tratta di interruzioni che la CPU può decidere di non ascoltare o servire;
- **interruzioni non mascherabili:** si tratta di interruzioni che la CPU deve assolutamente servire interrompendo immediatamente il lavoro che sta eseguendo; per esempio una interruzione proveniente dal gruppo di continuità che informa dell'assenza di alimentazione elettrica e autonomia in esaurimento della batteria non può essere ignorata ed il sistema esegue il programma di spegnimento ovvero quello che si chiama shutdown del sistema.

Il compito principale del nucleo del sistema operativo è quello di gestire l'esecuzione dei processi che, a volte, sono organizzati in JOB o lavori. Nel corso di una giornata un medio o grande sistema di elaborazione esegue migliaia di programmi; se, cessata l'esecuzione di un programma, occorresse l'intervento di un operatore per avviare il programma successivo, il numero di programmi gestiti dal sistema sarebbe molto ridotto a causa dei tempi alti dovuti agli interventi più o meno continui dell'operatore; si è pensato, pertanto, di organizzare, quando possibile, l'esecuzione di più programmi in JOB e sottoporre al sistema non più l'esecuzione di un programma ma di un lavoro; il sistema, in maniera assolutamente automatica ed autonoma, è in grado di procedere all'esecuzione del programma successivo senza alcun intervento dell'operatore fino al completamento del JOB; il numero di lavori eseguiti nell'unità di tempo ovvero quello che ho chiamato Throughput aumenta notevolmente. Un esempio è dato dai file batch del vecchio sistema operativo MSDOS che possono essere ancora utilizzati in ambiente Windows.

Le politiche di gestione dei processi possono essere di tipo event driven, ovvero pilotati dagli eventi quali interruzioni o terminazione dell'esecuzione di un processo, e time driven, ovvero pilotate dal tempo, nel qual caso si tratta della già descritta tecnica del time sharing secondo la quale l'interruzione di un processo avviene comunque dopo la terminazione del time slice.

E' evidente che il criterio event driven deve prevedere dei criteri correttivi per quei programmi che dovessero utilizzare la CPU per un tempo eccessivo monopolizzandola. Con la politica event driven è possibile adottare alcuni criteri per servire i processi:

- **FIFO (First In First Out):** i processi vengono serviti secondo l'ordine temporale di arrivo; un processo va in stato di attesa per effetto di una interruzione;
- **SJF (Shortest Job First):** vengono eseguiti per prima i programmi con il minor numero di istruzioni, ritenendo che possa essere terminato per prima; in effetti

poche istruzioni non significa complessità di tempo bassa nel senso che la presenza di numerosi cicli possono portare a tempi di elaborazione piuttosto lunghi;

- **SPTF (Shortest Process Time First):** vengono serviti per prima i processi che contengono molte istruzioni di I/O che, quindi, impiegano poco la CPU; vi ricordo che quando un programma richiede una operazione di I/O viene interrotto.

Questi criteri sono piuttosto semplici da applicare ma si corre il rischio che un processo aspetti un tempo infinito per cui si introducono dei criteri correttivi come, per esempio, aumentare la priorità di processi “sfortunati”.

La politica time driven, invece, garantisce che nessun programma aspetti un tempo infinito e, in genere, vengono serviti con una politica FIFO detta round robin; anche per la gestione round robin è possibile prevedere criteri che assegnino priorità diverse ai vari processi in maniera da velocizzarne l'esecuzione di quelli ritenuti più importanti; per esempio sarebbe possibile gestire una coda di code ad ognuna delle quali è assegnato un diverso quanto di tempo oppure servire con maggiore frequenza i processi di una coda rispetto all'altra.

Il time slice assegnato a ciascun programma può essere fisso oppure variare, per esempio, in funzione del numero di programmi in esecuzione.

Da quanto detto, si evince che un processo si può trovare in diversi stati, come per esempio, quello di interrotto, oppure in uno dei seguenti:

- **Submit:** è lo stato in cui il programma viene sottoposto al sistema per la sua esecuzione, per esempio mediante un comando immesso dall'utente o la chiamata di un altro programma; in questo stato il programma non è in esecuzione ma in attesa delle risorse necessarie;
- **Attesa:** il programma è in attesa delle risorse necessarie;
- **Pronto:** nello stato di pronto il programma è stato caricato in memoria, ha ricevuto le risorse necessarie ed attende la CPU per essere eseguito;
- **Esecuzione:** il programma è in esecuzione quindi un processo;
- **Interrotto:** il programma è stato interrotto per una delle cause già citate;
- **Terminazione:** il programma termina la sua esecuzione e le risorse impegnate, come ad esempio la memoria occupata, vengono rilasciate.

Il sistema operativo deve tenere memoria dello stato di ciascun processo per poter correttamente portarne a termine l'esecuzione ed a tal scopo fa uso di un PCB (Process control block) per ciascun processo.

Un rischio molto alto nella gestione dei processi è rappresentato dal cosiddetto stallo o deadlock che impedisce l'avanzamento di due o più processi che si bloccano a vicenda; per esempio il processo A è in attesa di una risorsa occupata da B che a sua volta attende una risorsa impegnata da C che è bloccato da A. I progettisti di sistemi operativi devono fare molta attenzione per evitare situazioni di stallo che

compromettono persino l'utilizzabilità del sistema stesso; in effetti a volte è piuttosto difficile individuare una situazione di stallo piuttosto che risolverla.

A proposito dello stallo, si racconta l'aneddoto dei filosofi che a me piace riportare. Otto filosofi siedono ad un tavolo circolare con 8 piatti e 8 forchette in modo tale che ciascuno di essi condivide la forchetta di sinistra. I filosofi pensano e quando decidono di mangiare devono prendere la forchetta di sinistra, quindi quella di destra e poi mangiare. Se i filosofi smettono di pensare contemporaneamente, tutti prendono la forchetta di sinistra e nessuno riesce a prendere quella di destra e muoiono di fame. Per evitare una così brutta fine sarebbe sufficiente che i filosofi, non potendo prendere la forchetta di destra, rilascino, per esempio dopo un tempo casuale, anche quella di sinistra consentendo al collega di mangiare.

In effetti a me piace fare anche un altro esempio molto semplice: quattro donne che arrivano contemporaneamente ad un quadrivio nel quale occorre, per tutte, dare la precedenza a destra. Non me ne vogliano le donne, si tratta di una battuta!

Una soluzione al problema dello stallo, pertanto, è data dal prerilascio delle risorse che costringe un processo in attesa a rilasciare le risorse occupate consentendo ad altri di proseguire nell'elaborazione. Si potrebbe anche assegnare ad un processo tutte le risorse di cui ha bisogno all'inizio dell'elaborazione, assegnazione statica delle risorse, ma queste risulterebbero inutilizzabili da parte di altri processi durante tutta l'esecuzione del processo che le ha ottenute anche se in un certo istante non sono utilizzate.

11.2.2. Gestione della memoria

Prima di passare a descrivere le tecniche di gestione della memoria centrale o RAM, spesso riferita semplicemente con il termine memoria, vi ricordo la classificazione dei dispositivi, passivi e condivisibili, in grado di memorizzare i dati, secondo un ordine decrescente delle prestazioni ovvero in ordine crescente sui tempi di lettura/scrittura:

- **CPU**: In genere non ha memoria se non pochi registri in grado di contenere l'istruzione in esecuzione e i dati coinvolti; tempi di elaborazione dell'ordine di milionesimi di secondo;
- **Cache memory**: oggi è quasi sempre integrata nel chip del processore; ha una capacità assai limitata, dell'ordine di qualche MB, ed una velocità paragonabile a quella della CPU;
- **RAM**: aumenta la capacità, ordine di qualche GB, ma i tempi di lettura/scrittura sono appena superiori a quelli della CPU;
- **HDD**: si tratta del dispositivo di memoria più capiente, dell'ordine di centinaia di GB, ma i tempi di accesso salgono a millesimi di secondo;

- **Memorie esterne:** appartengono a questa categoria i CD, DVD, Pen disk, ecc.; si tratta di memorie cosiddette off line, nel senso che può essere necessario un intervento dell'utilizzatore per poter essere utilizzabili dal sistema; si tratta delle memorie più lente presenti nel sistema ma offrono il vantaggio dell'estraibilità.

La memoria contiene i programmi in esecuzione ed i dati da elaborare; la sua gestione, soprattutto in ambienti multiprogrammati, è tutt'altro che semplice. Vi ricordo che un programma, per essere eseguito, deve essere caricato in memoria centrale e ciò avviene secondo tre tecniche di rilocalizzazione: assoluta, statica e dinamica; è fatto obbligo rivedere queste tecniche già descritte nel capitolo 7- Linguaggi di programmazione.

La gestione della memoria in ambienti monoprogrammati è piuttosto semplice; una parte della memoria, infatti, è occupata dal nucleo del sistema operativo e la restante parte può tranquillamente essere tutta assegnata all'unico programma in esecuzione che, per esempio, può essere rilocato in maniera statica ovvero al momento del caricamento.

In ambienti multiprogrammati è necessario che il sistema operativo protegga le aree occupate da ciascun processo e quelle occupate dal sistema operativo da una eventuale intrusione di altri processi, per esempio dovute ad errori di programmazione, che ne comprometterebbe la corretta elaborazione o, addirittura, il blocco del sistema. Il sistema operativo deve, pertanto, proteggere se stesso e gli utenti. Questo obiettivo è raggiunto assegnando agli utenti diritti sulle aree di memoria allocate; questi diritti possono essere:

- **R – Read:** l'utente può solo leggere nell'area di memoria;
- **W (Write):** l'utente ha diritti di lettura e scrittura;
- **E (Execute):** l'utente ha diritto di eseguire il codice contenuto in memoria.

Una delle prime tecniche per la gestione della memoria centrale in ambienti multiprogrammati è quella delle partizioni; la memoria viene divisa in tante partizioni ognuna delle quali contiene un programma in esecuzione. Le partizioni possono essere fisse o variabili; con la prima tecnica la memoria è suddivisa in tante parti di uguale grandezza mentre nel secondo caso ogni partizione ha una grandezza sufficiente a contenere il programma.

La tecnica delle partizioni fisse è piuttosto semplice da implementare perché il sistema deve solo tener memoria dello stato di una partizione: libera o occupata; un nuovo programma può essere caricato in una qualsiasi partizione libera. Questa tecnica spreca memoria per tutti i programmi la cui lunghezza è inferiore a quella delle partizioni.

Tale problema è risolto con la tecnica delle partizioni variabili secondo la quale la partizione si adatta alla grandezza del programma. Questa tecnica, però, non è esente da complicazioni. Dopo l'esecuzione di numerosi programmi, infatti, la memoria può essere frammentata ovvero lo spazio libero non è tutto fisicamente contiguo; per esempio il programma A occupa 3 MB e terminata la sua esecuzione

tale partizione è occupata dal programma B di 2,5 MB; si è creato un “buco” di 0,5 MB. Una eccessiva frammentazione può portare alla impossibilità di eseguire un programma per mancanza di spazio fisico contiguo in memoria; in altre parole lo spazio libero è sufficiente ma non contiguo. In questo caso il sistema procede ad una complicata ed onerosa, in termini di tempo, operazione di deframmentazione della memoria.

L'allocazione in memoria di un nuovo programma può avvenire secondo diverse tecniche:

- **First Fit:** Il programma viene allocato nella prima partizione libera capace di contenerlo;
- **Best Fit:** il programma viene allocato nella partizione che lascia meno spazio libero;
- **Worst Fit:** il programma è allocato nella partizione che lascia più spazio libero confidando di potervi caricare un altro programma.

Da quanto detto, si evince che un programma che superi la grandezza di una partizione, in quelle fisse, o la grandezza della memoria, in quelle variabili o in ambienti monoprogrammati, non può essere eseguito; il programmatore, in passato, doveva farsi carico di questo problema ed utilizzare la tecnica dell'overlay; secondo questa tecnica, la stessa area di memoria veniva utilizzata per caricare parti, mutuamente esclusive, dello stesso programma; tale gestione era a carico del programmatore.

Le tecniche finora viste si basano sul principio che un programma, per essere eseguito, deve essere caricato tutto in memoria ed in locazioni fisicamente contigue. I moderni sistemi operativi si sono svincolati da questa necessità che porta a numerosi vantaggi nella gestione della memoria e si fonda sul principio che alcune parti di un programma vengono eseguite una sola volta, come per esempio quella iniziale, ed altre potrebbero non essere eseguite affatto ma pur tuttavia occupano spazio in memoria. In altre parole sto dicendo che:

- un programma può essere caricato in memoria centrale solo in parte, quella in esecuzione;
- porzioni dello stesso programma, dette pagine, risiedono in memoria in maniera fisicamente non contigua;
- è possibile eliminare dalla memoria centrale pagine di programma non più necessarie o presunte tali per far posto a pagine di altri programmi.

Questa ottimizzazione è possibile grazie alla tecnica della paginazione che divide i programmi in pagine, aventi tutte la stessa grandezza, e la memoria in tante parti, dette cornici di pagina, ognuna delle quali è in grado di contenere una pagina di un programma.

All'avvio di un programma viene caricata in memoria la sua prima pagina e così avviene per tutti i programmi in esecuzione; nel momento in cui un programma

richiede l'esecuzione di codice presente in un'altra pagina non presente in RAM, evento detto page fault, sarà necessario caricarla in memoria centrale dalla memoria di massa; questa operazione, essendo di Input/Output, interrompe il programma e la CPU passa all'esecuzione di un altro programma; quando la pagina sarà stata caricata, il processo va di nuovo in stato di pronto ed attende la CPU per ritornare in stato di esecuzione. Quando una nuova pagina di un programma viene caricata, può succedere che l'area di memoria fisicamente contigua a quella dove è stata allocata l'ultima sua pagina sia già occupata, cosicché le pagine di uno stesso programma vengono ad essere allocate in maniera non contigua. Come è possibile intuire, si tratta di una gestione piuttosto complessa ed il sistema operativo, oltre ad avere informazioni sullo stato di ciascun processo, deve anche tenere una mappa dell'allocazione dei processi in memoria centrale; in generale tutte le informazioni che riguardano un processo sono memorizzate nel PCB.

La gestione della paginazione si complica ulteriormente quando, necessaria una nuova pagina, la memoria centrale risulta completamente occupata. In questo caso è necessario liberare una cornice di pagina in memoria, ma non sempre è possibile eliminare dalla memoria centrale una pagina senza incorrere in seri problemi. I processi, infatti possono essere classificati in puri e non puri; un processo puro distingue le aree che sono sottoposte a modifiche, come le aree dati, da quelle che non lo sono, come le aree contenenti codice di programma. In un processo non puro questa distinzione è piuttosto difficile; così, se la pagina che si vuole eliminare dalla memoria centrale non è pura, ovvero ha subito modifiche durante l'elaborazione, sarà necessario salvarla perché, se dovesse essere di nuovo necessaria, non sarà possibile ricaricare quella originale. Ma allora dove viene salvata la pagina da eliminare? Ovviamente sulla memoria di massa più veloce in un'area appositamente allocata dal sistema operativo e riferita con il termine memoria virtuale; quest'ultima, pertanto, viene ad essere un virtuale allungamento della memoria centrale; è evidente che il rapporto memoria virtuale/memoria centrale non deve essere molto alto, altrimenti lo scambio continuo pagine, page swapping, tra le due provoca un notevole rallentamento del sistema; in ambiente windows, in genere, si consiglia di impostare la memoria virtuale pari, al massimo, al doppio di quella reale.

L'algoritmo di scambio delle pagine è detto page in/page out; questo algoritmo deve scegliere, tra tutte le pagine presenti in memoria, quella da eliminare o da riportare nella memoria virtuale; qualsiasi algoritmo di scelta deve cercare di eliminare la pagina che presumibilmente non servirà più per l'esecuzione del programma cui appartiene in modo da evitare un altro page in; in genere, le tecniche utilizzate sono le seguenti:

- **FIFO:** viene eliminata la prima pagina caricata e così via;
- **LRU (Last Recently Used):** quella che da più tempo è inutilizzata.

Il primo algoritmo è molto semplice da attuare e mantiene basso il system overhead mentre il secondo, sicuramente più complesso, potrebbe, con maggiore certezza, individuare la pagina non più necessaria.

Una variante della paginazione è rappresentata dalla segmentazione; l'unica differenza consiste nel fatto che i segmenti non hanno la stessa grandezza. Si tratta di un metodo più complesso che porta ad un problema già riscontrato in passato: la frammentazione; l'eliminazione di un segmento, per esempio di 50k, necessaria per caricarne uno di 45k, lascia 5k liberi che sono difficili da utilizzare; la segmentazione però, può avere un enorme vantaggio riducendo il numero di riferimenti tra i segmenti; la divisione in pagine di un programma avviene come una scure in punti precisi senza alcuna possibilità di variazione; con la segmentazione, invece, la divisione avviene in maniera logica, per esempio alla fine di un ciclo o di una routine, in modo tale che l'esecuzione iterativa dello stesso gruppo di istruzioni non faccia riferimento ad un'altra pagina, con il conseguente rischio di page in/page out ogni volta che si esegue un passo dell'iterazione. Questa divisione logica non può essere fatta dal sistema operativo ma dovrebbero essere i compilatori, vedi capitolo 7- Linguaggi di programmazione, che generando il codice eseguibile, ad individuare i punti in cui dividere il programma. Per contro questa tecnica, oltre al già detto problema della frammentazione, comporta algoritmi di gestione più complicati e, quindi, un system overhead maggiore.

11.2.3. Gestione delle periferiche

Ho già dato una classificazione dettagliata delle risorse e, quindi, dei dispositivi periferici; in questo paragrafo mi occuperò semplicemente della gestione delle stampanti, che vi ricordo è una risorsa dedicata, e degli accessi al disco rigido, risorsa condivisibile.

I moderni sistemi operativi sostituiscono a dispositivi lenti e dedicati, quali stampanti e plotter, dispositivi veloci e condivisibili come i dischi rigidi, allo scopo di evitare lunghe interruzioni ai processi nel momento in cui hanno bisogno di stampare e la risorsa è occupata; vi ricordo che la stampante è un dispositivo lento che può impiegare anche minuti o, addirittura ore, per completare un lavoro di stampa. Le stampe o l'output destinato a periferiche dedicate lente vengono, quindi, memorizzate su disco rigido (virtualizzazione della stampa), in aree separate per i vari utenti e utilizzando un meccanismo chiamato SPOOL (Simultaneous Peripheral Operation On Line); un processo in background trasferisce le stampe, accodate su hard disk, una alla volta alla stampante in modo che, una volta avviata una operazione di stampa, questa avvenga senza soluzione di continuità ovvero senza interruzioni; il trasferimento alla stampante utilizza, in genere, una logica di tipo FIFO, ma non è esclusa la possibilità di gestire priorità, per esempio in base agli utenti, e privilegiare alcune stampe rispetto ad altre che, così, vengono effettuate per prima anche se arrivano in un secondo momento.

Durante una stampa, quindi, è possibile che altri programmi richiedano di stampare e, in tempi brevissimi, essi si liberano perché, tra l'altro, l'invio dei dati al

disco rigido piuttosto che ad un dispositivo dedicato e lento, impiega pochi secondi. Si crea, pertanto, una coda di stampa che viene liberata con i tempi del dispositivo periferico ma senza intralciare più di tanto l'esecuzione dei processi.

La gestione del disco rigido comporta ancora una volta le stesse problematiche: durante una operazione di I/O da hard disk, arrivano altre richieste sulla periferica che, pertanto, dovrà servirle secondo una logica stabilita e che porti ad ottimizzare l'utilizzo della risorsa; ciò significa, ovviamente, migliorare le prestazioni dell'intero sistema perché soprattutto la gestione degli accessi al disco rigido incide molto sulle prestazioni dell'intero sistema. Abbiamo visto, infatti, come il disco rigido sia il supporto ideale per la memoria virtuale e non solo, esso è il contenitore dei programmi applicativi e, soprattutto, dei dati che essi vanno ad elaborare; è evidente, quindi, la grande importanza che un disco rigido riveste per il sistema e la velocità degli accessi non dipende solo dalle caratteristiche hardware del disco ma, soprattutto, dalla logica con la quale il sistema gestisce la coda delle operazioni di lettura/scrittura.

Il sistema può procedere alla gestione delle richieste di lettura/scrittura su un disco rigido secondo diverse modalità, la cui bontà può essere compresa solo ricordando l'organizzazione hardware del disco rigido fatta di tracce, settori, cilindri e testine, che vi prego di ripetere rivedendo parte del capitolo 3-Hardware. Questi gli algoritmi utilizzabili dal sistema:

- **FIFO**: le operazioni richieste vengono servite secondo l'ordine di arrivo;
- **SSTF** – Shortest Seek Time First: le operazioni di I/O vengono servite in modo da minimizzare gli spostamenti delle testine, per cui l'operazione successiva avviene sul settore più vicino alla posizione delle testine di lettura/scrittura; si tratta di una grande ottimizzazione perché, come ricorderete, il tempo più alto in una operazione di I/O su disco rigido è quello di posizionamento; ovviamente il sistema operativo prevede dei correttivi per evitare che un processo aspetti un tempo infinito; infatti, se arrivassero continuamente richieste in prossimità della posizione delle testine, una operazione lontana da questa posizione rischierebbe di non essere mai eseguita; il sistema operativo, per esempio, potrebbe man mano aumentare la priorità di alcune richieste in modo che, ad un certo punto, esse vengano obbligatoriamente eseguite;
- **SCAN**: le operazioni vengono eseguite evitando il continuo cambio di direzione dell'avanzamento delle testine; per cui le testine si spostano dall'interno verso l'esterno finché ci sono operazioni da eseguire, per poi cambiare direzione quando nessuna richiesta è oltre la posizione attuale delle testine e così via; questa tecnica, limitando molto il cambio di direzione, preserva il disco dai guasti.

La scrittura su disco rigido, ovviamente, procede per cilindro e non per traccia, ovvero, dopo il completamento della scrittura di una traccia su una superficie, si passa alla traccia corrispondente sulla superficie inferiore dello stesso disco quindi a tutte le tracce corrispondenti sullo stesso cilindro, invece di passare alla traccia

successiva sulla stessa superficie; questo modo di procedere ottimizza gli spostamenti delle testine, quindi il tempo di posizionamento.

11.2.4. File system

Un hard disk è organizzato da un punto di vista fisico in tracce e settori; anche altri dispositivi di memoria di massa hanno organizzazioni fisiche analoghe e, comunque, ben lontane da come ci appaiono logicamente ovvero organizzate gerarchicamente in file e cartelle.

Se l'utilizzatore di un sistema volesse gestire l'allocazione dei dati su una memoria di massa riferendosi direttamente alla loro organizzazione fisica, il compito sarebbe troppo arduo e non garantirebbe alcuna sicurezza; al dire il vero sarebbe impossibile! Il sistema operativo, quale gestore di tutte le risorse, non poteva esimersi dalla gestione delle memorie di massa ed il componente che se ne occupa è il file system.

Il file system è, quindi, l'insieme dei programmi del sistema operativo che si occupa dell'allocazione dei dati sulle memorie di massa, organizzandoli gerarchicamente in volumi, file e cartelle. Così l'utente gestisce, mediante un'interfaccia amichevole, file e cartelle ed il sistema si preoccupa di allocarli sulle memorie di massa, riconducendo l'organizzazione logica a quella fisica, costituita da tracce e settori.

Il file system, però, non si limita alla allocazione di file e cartelle su disco rigido, ma le sue funzioni sono ben più ampie:

- **Allocazione:** l'allocazione comporta, tra l'altro, la gestione della creazione, della cancellazione, degli accessi e, in genere di tutte quelle operazioni su file e cartelle che abbiamo visto durante il primo anno e che ho riportato nel capitolo 4-Windows;
- **Protezione:** il file system deve garantire livelli di protezione tra i vari utenti che hanno accesso al sistema, siano essi locali o remoti, ovvero che utilizzano le risorse mediante servizi di rete; a ciascun utente possono essere assegnati o negati i seguenti diritti:
 - Controllo completo: l'utente può eseguire qualsiasi operazione;
 - Lettura/esecuzione: si applica ai file e ne consente la lettura o, nel caso di file di programma, la loro esecuzione;
 - Visualizza contenuto cartella: solo per le cartelle, l'utente ne può visualizzare il contenuto;
 - Scrittura: possibilità di modificare il contenuto del file;
 - Cancellazione: è possibile cancellare il file.

Vedremo la gestione delle protezioni applicate a file, cartelle e risorse in generale, quale ad esempio una stampante, in ambiente Windows XP.

- **Condivisione:** la condivisione di un file consente a più utenti, aventi gli opportuni diritti, di elaborarlo o meno contemporaneamente; è il sistema operativo che gestisce i blocchi dei file, evitando o consentendo che esso possa essere elaborato contemporaneamente da più utenti; si tratta di un aspetto fondamentale, si pensi ad una banca ed alla necessità che più operatori condividano i dati presenti nel data base; potrei farvi numerosissimi esempi ma li vedremo a lezione;
- **Ottimizzazione:** il file system deve ottimizzare l'allocazione dei file su disco, sia dal punto di vista dello spazio che, soprattutto, del tempo, necessario per le operazioni di accesso, ricerca e di tutte le altre operazioni necessarie sui file, come descritto nel capitolo 6-Strutture dati.

L'allocazione delle cartelle e dei file sulle memorie di massa non è assolutamente semplice, considerato che la struttura fisica di queste ultime è completamente diversa e molto più semplice rispetto all'aspetto logico di cui l'utente può disporre.

Un file, ormai è chiaro, viene allocato sui settori del disco; un settore, però, ha una grandezza assai limitata, solo 512 byte; il file system, pertanto, utilizza il cluster come la più piccola unità di allocazione; un cluster, in genere, occupa 8 settori, anche se la sua dimensione ha subito modifiche nel corso dell'evoluzione dei sistemi operativi ed oggi essa dipende dalla capacità del volume. In altre parole quando il file viene creato, il file system gli assegna uno o più cluster ed ogni volta che il file ha necessità di crescere, il file system gli assegna altri cluster; succede spesso che l'allocazione successiva di cluster ad un file avvenga in maniera discontinua, ovvero che i cluster successivi non siano allocati in maniera fisicamente contigua, semplicemente perché, nel frattempo, tale spazio è stato assegnato ad un altro file; i file, pertanto, risultano frammentati; una grande frammentazione fa aumentare i tempi di accesso al disco e riduce le prestazioni del sistema; il sistema operativo, in genere, mette a disposizione programmi di utilità che eliminano la frammentazione, come vedremo per il sistema operativo Windows.

E' evidente che un cluster grande riduce la frammentazione mentre un cluster piccolo aumenta lo spreco di spazio; questo perché un cluster assegnato ad un file e non interamente occupato da quest'ultimo, non può essere utilizzato da nessun altro file. In genere, il numero di settori per cluster dipende dalla grandezza del volume; a volte, l'utente può sceglierne la sua grandezza quando procede alla formattazione del volume, come è possibile in Windows XP e Vista.

La cancellazione di file e cartelle, ovviamente, frammenta anche lo spazio libero nel volume, inizialmente tutto allocato alla fine, e ciò aumenta la frammentazione globale.

Il file system, pertanto, deve memorizzare la struttura ad albero delle cartelle, con l'elenco dei file contenuti in ciascuna di esse, ed i cluster occupati da ciascun

file; per fare questo fa uso di una directory e di una FAT (File Allocation Table) che, per ciascun file e, in genere, mediante una lista concatenata, contiene l'elenco delle aree occupate. Il file system, ovviamente, deve anche tenere memoria dei cluster liberi ovvero non occupati da alcun file.

Tutti i file system consentono di individuare un file mediante il pathname o nome globale; si tratta del percorso di cui vi ho parlato nel capitolo 4-Windows. Il solo nome del file è spesso riferito come locale, perché relativo alla cartella che lo contiene.

Nel corso del tempo e con riferimento ai sistemi operativi sviluppati da Microsoft, si sono succeduti i seguenti tipi di file system:

- **File system a cartella unica:** negli anni '80 del secolo scorso, le memorie di massa erano costituite essenzialmente dai floppy disk con una capacità assai limitata, dell'ordine di qualche centinaio di KB, per cui non vi era necessità di una struttura ad albero visto che il supporto poteva contenere pochissimi file;
- **FAT16:** da questo file system in poi si adotta la struttura gerarchica; caratterizzata dall'utilizzo di 16 bit per l'indirizzamento di 65536 cluster;
- **FAT32:** il continuo aumentare della grandezza dei dischi rigidi ha reso la FAT16 inadeguata; con la FAT16 non si riusciva più ad indirizzare tutto lo spazio che un hard disk metteva a disposizione; la prima soluzione fu quella di aumentare la grandezza del cluster, che passò prima a 16 e poi, addirittura, a 32 settori, con uno spreco eccessivo dello spazio. Con la versione B di Windows 95, Microsoft introduceva la FAT32 che, utilizzando 32 bit per indirizzare i cluster, ne poteva ridurre la grandezza che, così, ritornava a 8 settori;
- **NTFS:** è il file system utilizzato da sistemi operativi come Windows NT prima, Windows 2000 poi, e XP e Vista oggi, nonché da tutti i sistemi operativi della linea Windows Server. I vantaggi che questo file system offre sono relativi soprattutto alla protezione, come vedremo in laboratorio e come brevemente descritto nel capitolo 4-Windows.

La tabella seguente riporta la grandezza del cluster in funzione della capacità del volume e del tipo di file system:

Volume	FAT16	FAT32	NTFS
7 MB–16 MB	2 KB	Not supported	512 bytes
17 MB–32 MB	512 bytes	Not supported	512 bytes
33 MB–64 MB	1 KB	512 bytes	512 bytes
65 MB–128 MB	2 KB	1 KB	512 bytes

129 MB–256 MB	4 KB	2 KB	512 bytes
257 MB–512 MB	8 KB	4 KB	512 bytes
513 MB–1,024 MB	16 KB	4 KB	1 KB
1,025 MB–2 GB	32 KB	4 KB	2 KB
2 GB–4 GB	64 KB	4 KB	4 KB
4 GB–8 GB	Not supported	4 KB	4 KB
8 GB–16 GB	Not supported	8 KB	4 KB
16 GB–32 GB	Not supported	16 KB	4 KB
32 GB–2 TB	Not supported	Not supported	4 KB

11.2.5. Shell

L'utente deve dialogare con il sistema operativo per poter avviare i programmi applicativi e svolgere tutte quelle attività che gli consentono di elaborare i dati con semplicità, velocità e sicurezza. A tale scopo il sistema operativo dispone di un software di interfaccia con l'utilizzatore detto Shell. I primi sistemi operativi offrivano agli utenti shell testuali e prive del supporto del mouse; con una tale interfaccia, detta prompt dei comandi o linea di comando, l'utente poteva solo richiedere l'avvio di un software mediante un comando testuale immesso da tastiera e, terminata l'esecuzione, immettere un altro comando per eseguire un'altra attività.

Il comando impartito, in genere, conteneva il nome del programma da eseguire seguito da zero, uno o più parametri, che dettagliavano meglio l'operazione da eseguire; per esempio il comando <del nome file> cancella un file:

- **del relazio.doc**: cancella il file relazio.doc;
- **del *.***: cancella tutti i file contenuti nella cartella corrente;
- **del *.doc /p**: cancella tutti i file .doc contenuti nella cartella corrente chiedendo, per ognuno di essi, una conferma.

La linea di comando esiste ancora oggi nei sistemi operativi Windows ed è accessibile da <START/Tutti i programmi/Accessori/Prompt dei comandi>; in laboratorio vedremo qualche comando, solo per meglio capire come si interfacciavano i sistemi operativi di 20 anni fa.

I moderni sistemi operativi, invece, adottano interfacce di tipo grafico e supportano il mouse come periferica di input; una shell grafica risulta molto amichevole o user friendly; la rappresentazione mediante icone degli oggetti disponibili sul sistema e la gestione di due o più pulsanti del mouse, unito a quello per lo scroll, rendono l'utilizzo del sistema facile da apprendere e molto produttivo.

11.2.6. Programmi di utilità e software applicativo

Un sistema operativo mette a disposizione dell'utilizzatore una serie di *programmi di utilità* che svolgono le cosiddette *operazioni ausiliarie all'elaborazione dei dati* quali copie di sicurezza, ottimizzazione delle risorse, individuazione di errori e soluzioni, ecc.

In passato anche l'editor, necessario per scrivere i programmi sorgenti, il linker ed il debugger, di cui vi ho parlato nel capitolo 7-Linguaggi di programmazione, erano considerati programmi di utilità; sappiamo che oggi un linguaggio di programmazione offre un tool di sviluppo completo che integra tutto quanto è necessario per produrre un software.

Spesso i sistemi operativi offrono all'utilizzatore anche semplici programmi applicativi. Windows, per esempio, mette a disposizione dell'utente un software di navigazione in Internet, anche se Microsoft ritiene che sia un componente altamente integrato nel sistema ed ineliminabile, un software di gestione della posta elettronica, un semplice programma di scrittura (Wordpad), la calcolatrice, ecc.

Ritengo di non aggiungere altro ma lascio il passo al laboratorio, nel quale vedremo le utilità del sistema operativo Windows e qualche programma applicativo, tra l'altro brevemente trattati nel capitolo 4-Windows.

11.2.7. Accounting e controllo degli accessi

I sistemi operativi destinati a macchine server, ovvero gestori di una rete, di cui parlerò nel prossimo capitolo, gestiscono la contabilizzazione o accounting degli accessi al sistema riportando, per ciascun utente, data ed ora di accesso (Login) al sistema e di uscita (Logout); l'accounting è necessario quando l'utente deve un compenso al gestore del server per il tempo di utilizzo come, per esempio, avviene nelle connessioni ad internet attraverso un server ISP (Internet Service Provider).

Per procedere ad un corretto accounting, il server deve riconoscere l'utente che vi accede; per questo i sistemi fanno uso dell'autenticazione, che avviene durante l'operazione di login, obbligando l'utente a specificare un nome e, eventualmente, una password, che devono corrispondere ad un account registrato sul server; in caso di accounting a pagamento la password è sempre richiesta, a meno che la

contabilizzazione non avvenga secondo parametri diversi come, per esempio, il numero di telefono chiamante.

I server, in genere, prevedono delle restrizioni sugli account, che limitano l'accesso alle risorse del sistema, per esempio assegnando solo alcuni diritti e solo su alcune cartelle; è possibile anche definire, nell'ambito della settimana, i giorni e le ore in cui quell'account può essere utilizzato, impedendone l'accesso all'utente in orari diversi. Un account di posta elettronica, per esempio, assegna diritti solo su una cartella in un server di posta elettronica ed impedisce l'accesso a qualsiasi altra cartella presente nel sistema.

11.2.8. Avviamento del sistema

In questo paragrafo descriverò le fasi che portano dall'accensione del sistema, in genere un PC, fino al caricamento in memoria del sistema operativo:

- **accensione**: la CPU, che sa solo eseguire programmi, comincia l'esecuzione di quello contenuto nella ROM, che costituisce il BIOS (Basic Input Output System);
- **POST – Power On Self Test**: il programma caricato dalla ROM consente alla CPU di effettuare una breve diagnostica dei componenti principali del sistema, quale, per esempio, la RAM; eventualmente segnala un errore, che quasi sempre è un errore hardware, altrimenti passa alla fase successiva. Durante il POST, l'utente può accedere, in genere mediante la pressione del tasto <Canc>, ai parametri di configurazione del BIOS; si tratta di valori che riguardano la configurazione hardware del sistema e vanno modificati con molta cura, altrimenti si rischia di bloccare il corretto avvio del sistema; in laboratorio vi farò vedere brevemente alcuni parametri di configurazione del BIOS;
- **Bootstrap**: superato il POST, la CPU esegue il programma bootstrap, che accede ad uno dei dispositivi di memoria di massa, in genere il disco rigido, e carica il loader;
- **Loader**: il loader carica il sistema operativo dalla memoria di massa e porta il suo nucleo in memoria centrale;
- **Sistema operativo**: il sistema operativo prende il controllo dell'hardware e, dopo aver caricato alcuni programmi applicativi (detti residenti), quali, ad esempio, l'antivirus, mostra la shell all'utente, che può così disporre del sistema. Molti programmi avviati dal sistema operativo, subito dopo il suo caricamento in memoria, sono presenti o, in genere, hanno collegamenti in <Start/Tutti i programmi/esecuzione automatica>.

Da questo momento il sistema operativo ha il controllo delle risorse e delle attività; una qualsiasi causa che porta a malfunzionamenti del sistema operativo, comporta, quasi sempre, un blocco totale del sistema e la necessità di un suo riavvio.

11.3. Produzione del software applicativo

Da quanto studiato nel corso dei primi due anni scolastici di questo corso programmatori, ci siamo resi conto che produrre un buon software non è una cosa semplice. In questo paragrafo voglio mettere ordine nelle varie fasi necessarie per arrivare a produrre e ben documentare un buon software applicativo. Il modo migliore di cominciare ritengo sia quello di riepilogare quanto fatto fino ad ora in modo da renderci conto di come tutto questo si amalgami perfettamente ed armonicamente nell'attività di produzione del software.

Ebbene, nel corso del terzo anno, il primo per l'informatica, ho cominciato con il presentarvi i rudimenti della programmazione e semplici algoritmi sviluppati attraversando una serie di fasi, dalla **descrizione dei dati** in ingresso alla **documentazione delle variabili utilizzate** passando attraverso **l'analisi** e la **diagrammazione a blocchi**, rigorosamente strutturata; appena pronti per il laboratorio abbiamo **codificato** e **verificato**, mediante **debugging**, il buon funzionamento del programma realizzato.

Nel corso del quarto anno abbiamo imparato per prima la **programmazione procedurale**, in cui abbiamo diviso il **problema in sottoproblemi**, e poi quella ad **oggetti**, con **metodi e proprietà**, ed abbiamo utilizzato alcuni **moduli standard** per la documentazione del progetto e vi ricordo, tra l'altro, i **diagrammi di flusso**.

In quest'anno, infine, abbiamo trattato uno degli argomenti più importanti del corso di studio ovvero i data base; così ora sapete realizzare uno **schema concettuale e logico**, realizzare **algoritmi** e scrivere frasi del linguaggio **SQL**.

Tutto questo non è diviso in compartimenti stagni ma si amalgama perfettamente ed armoniosamente ogniqualvolta il progetto informatico è di una certa complessità e, pertanto, richiede:

- La **descrizione dei dati**, che non è più così semplice come due anni fa, ma richiede la progettazione di uno schema concettuale e logico;
- **L'analisi** delle operazioni da eseguire sui dati;
- **La realizzazione e la verifica** degli algoritmi che, in genere, non sono mai soltanto semplici frasi del linguaggio SQL, ma che richiedono lo sviluppo di algoritmi che utilizzano paradigmi procedurali o, sempre più spesso, paradigmi orientati agli oggetti;
- **La realizzazione di una ottima documentazione**, necessaria per poter mantenere con semplicità e rapidità il software prodotto;
- **La realizzazione del manuale utente** destinato all'utilizzatore, necessario per un proficuo utilizzo del software da non confondere con la documentazione di cui ho parlato fino ad ora e che descrive **come è fatto** il software e non **come si utilizza**.

Se il vostro futuro sarà quello di programmare, allora quasi sicuramente non sarete da soli, ma inseriti in un gruppo di esperti costituito da diverse figure professionali; poche o numerose persone che hanno bisogno di dialogare, di scambiarsi documentazione, ancora più necessaria quando si sostituisce una persona all'interno del gruppo; le persone all'interno dello stesso gruppo di lavoro devono utilizzare modulistica standard per la documentazione e per lo scambio di informazioni, fino ad arrivare a vere e proprie **metodologie di programmazione** che standardizzano non solo la **documentazione** ma **regolano tutto il processo di sviluppo del software**; in passato era molto famosa la metodologia **Warnier**.

Ho ancora una volta sottolineato l'importanza della documentazione, necessaria perché un software è **fortemente dinamico**, tanto da parlare di **ciclo di vita del software**, che non sfugge alla regola secondo la quale esso nasce, si evolve e muore. Si pensi a tutti gli applicativi prodotti con il sistema operativo MS-DOS, oggi quasi non ne esistono più. E' ovvio che le conoscenze acquisite dal gruppo di progetto, che ha realizzato un software, trova vantaggio dalle conoscenze acquisite durante la produzione del defunto software e, quindi, possiamo dire che, con riferimento a tutte le sue fasi di realizzazione, un nuovo software eredita sempre qualcosa dal vecchio.

Per completare la trattazione dell'argomento, ritengo opportuno parlarvi dei **funzionigramma** o **FH, Function Hierarchy**, ovvero **gerarchia delle funzioni**; abbiamo già utilizzato un diagramma gerarchico durante lo scorso anno, quando abbiamo diviso il programma da realizzare in una gerarchia di problemi e sottoproblemi, per ognuno dei quali abbiamo realizzato, in maniera top-down o botto-up, un sottoprogramma (SUB) o una funzione (FUNCTION).

Dopo aver descritto i dati che il progetto informatico deve trattare ovvero, in genere, riportato lo schema concettuale e logico del data base, è necessario descrivere le operazioni, che diciamo funzioni, da eseguire su questi dati per trasformarli ed ottenere gli output previsti; ancora una volta la relazione esistente tra le funzioni è di tipo gerarchico per cui si parte dal livello 0, il progetto, e si individuano le funzioni di primo livello e dei livelli successivi; non si tratta di un qualcosa di molto diverso da quanto già fatto, come detto, lo scorso anno per la suddivisione del problema in sottoproblemi e per realizzare un buon funzionigramma è sufficiente seguire le seguenti regole:

1. **La radice** contiene il nome del progetto;
2. **Ogni nodo riporta il nome della funzione** o dell'operazione che si vuole eseguire sui dati;
3. **Operazioni complesse, funzioni madre**, vengono ulteriormente suddivise in operazioni più semplici, **funzioni figlie**, fino ad arrivare a quelle elementari che non vengono ulteriormente suddivise;
4. Ogni **funzione madre** deve avere **almeno due** funzioni figlie;

5. Ogni **funzione** madre non deve avere **più di sei** funzioni figlie; in un tal caso, per esempio, la funzione madre X può essere ripartita in due funzioni figlie che, a loro volta, sono madri delle funzioni originarie di X;

Lo scorso anno utilizzavamo questa tecnica per suddividere algoritmi in parti di più semplice realizzazione, anche per applicare i concetti della programmazione modulare; adesso stiamo individuando le funzioni che ci consentono l'elaborazione dei dati, dove una **funzione è qualcosa di più articolato di un sottoprogramma** e, per esempio, potrebbe essere molto complessa come “Emetti fattura” o molto più semplice come “Stampa articoli sotto scorta”; si tratta di funzioni e non di algoritmi che, in altre parole, possono richiedere, per la loro implementazione, anche la realizzazione di numerosi sottoprogrammi.

Ho cominciato questo paragrafo riepilogando il passato, continuato con qualche nuovo argomento e vado a chiuderlo, anche a costo di sembrare ripetitivo, riassumendo velocemente **le fasi per la realizzazione di un progetto informatico**:

1. **Descrizione del problema:** è la fase in cui si individuano dati e le procedure da realizzare; l'informatico viene reso edotto su quello che l'utilizzatore richiede; si utilizzano molto, in questa fase, le **interviste** alle persone che dovranno utilizzare l'applicativo e che, in genere, già svolgono il lavoro in maniera non automatizzata o parzialmente automatizzata; si raccoglie modulistica e informazioni sui dati e le procedure per il loro trattamento; si interagisce molto con il personale, in genere costituito da non informatici, che lavora nell'ambiente in cui si va a realizzare l'applicativo;
2. **Analisi:** Fase di studio dei dati e degli algoritmi da realizzare.
 - 2.1. **Descrizione dati:** schema concettuale e logico;
 - 2.2. **Descrizione degli algoritmi:** Funzionigramma, diagrammi di flusso, diagrammi gerarchici degli algoritmi e relativi diagrammi a blocchi, descrizione degli oggetti da realizzare;
3. **Codifica e debugging:** scrittura del codice, utilizzando il linguaggio di programmazione scelto, e verifica della sua funzionalità; sottolineo ancora una volta l'importanza di questa fase perché è quella in cui si decide della bontà del software;
4. **Documentazione:**
 - 4.1. **Del programmatore:** non si tratta di una fase finale ma trasversale; ognuna delle fasi precedenti produce documentazione, alla fine è solo necessario organizzarla in maniera organica;
 - 4.2. **Dell'utente:** documentazione destinata all'utente finale; si realizza quello che si chiama manuale utente.
5. **Installazione o deployment:** Il software, realizzato e testato, viene immesso sul mercato o installato presso il committente; in questa fase, spesso, è compresa quella di addestramento del personale che dovrà utilizzare il nuovo software;

6. Manutenzione: Si tratta della fase successiva alla realizzazione del software; come detto un software non è statico ma fortemente dinamico e va modificato per numerosi motivi:

- **Eliminazione degli errori:** il software può fornire dati errati o concludersi in modo anomalo, per cui vanno individuati gli errori sfuggiti nella fase di debugging ed eliminati; in genere è l'utilizzatore che segnala la presenza di errori nella fase successiva all'installazione;
- **Miglioramento delle prestazioni:** un software potrebbe dimostrarsi inadeguato in alcune elaborazioni a causa di tempi piuttosto lunghi;
- **Miglioramento dell'interfaccia utente;**
- **Ampliamento delle funzionalità:** aggiunta di nuove funzioni al software, a volte su indicazione degli utilizzatori;
- **Personalizzazione:** personalizzazioni del software richieste da uno specifico utilizzatore ed alle quali non sono interessati tutti gli utenti;
- **Adeguamento a nuove normative:** Si pensi ad un software per la dichiarazione annuale dei redditi, che deve rispettare la legislazione vigente sempre in continua evoluzione.

11.4. La qualità del software

La **qualità** di un prodotto ha assunto una importanza fondamentale in tutti i settori della produzione; molti sono i marchi di qualità, alcuni obbligatori come, ad esempio, il marchio **CE** per i prodotti commercializzati nell'Unione Europea, altri no quali i vari marchi DOC, DOP, ecc. utilizzati per i prodotti alimentari. Quando si compra un prodotto, si deve sempre considerare **il rapporto qualità/prezzo**, per cui il prodotto più economico non sempre o, meglio, spesso non è il più conveniente.

A questa regola importante non può sfuggire il software; bisogna pertanto individuare dei parametri rispetto ai quali definire la qualità del software; così come, ad esempio, per un'automobile si considerano parametri importanti il consumo, l'affidabilità, il comfort ecc., per il software i parametri sono classificati in **interni**, che si riferiscono a come il software è realizzato ossia come viene visto dagli sviluppatori, ed **esterni**, che si riferiscono all'interfaccia utente ovvero da come il software viene percepito dagli utilizzatori:

1. Parametri interni:

- **Manutenibilità:** facilità con la quale è possibile mantenere il software, così come descritto alla fine del paragrafo precedente;
- **Portabilità:** Un software è portabile quando può essere utilizzato su sistemi diversi con poche o addirittura nessuna modifica; si pensi, per esempio, a

software che possono girare correttamente su macchine aventi sistemi operativi diversi.

2. Parametri esterni:

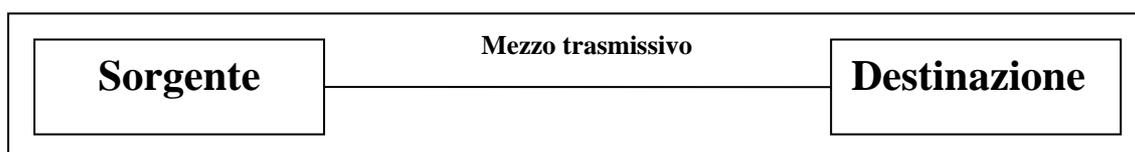
- **Correttezza**: ricordate l'efficacia? Un software corretto fornisce risultati esatti secondo le specifiche iniziali del progetto;
- **Affidabilità**: si riferisce ai malfunzionamenti del software; un software è tanto più affidabile quanto meno frequenti sono i suoi malfunzionamenti; un indice di affidabilità, utilizzato anche per le apparecchiature hardware, è dato dall'**MTBF, Mean Time Between Failure**, che è il tempo medio tra due malfunzionamenti successivi;
- **Robustezza**: corretta funzionalità garantita sempre, anche in situazioni o utilizzo particolari; in pratica il programma non va in errore neanche in situazioni non previste nelle specifiche iniziali del progetto ed a volte dovute a cause esterne al programma quali malfunzionamenti dell'hardware;
- **Efficienza**: il programma elabora i dati in tempi rapidi ovvero ha una buona complessità di tempo in tutte le elaborazioni svolte;
- **Facilità di utilizzo o usabilità**: l'utente utilizza maschere semplici ed intuitive per l'immissione dei dati e le richieste di elaborazione; in molti casi, come per i software ed i siti accessibili via Internet, ha assunto oggi una particolare importanza il termine **accessibilità**, che si riferisce alla semplicità con la quale il sito è usufruibile anche da persone disabili;
- **Scalabilità**: il software può essere utilizzato in ambiti con complessità assai diversificate; per esempio lo stesso software mantiene tutte le sue qualità nel caso in cui un utente utilizzi database piccoli, ovvero con pochi record, o database molto estesi, anche con milioni di record; in questi casi l'efficienza può essere migliorata semplicemente utilizzando sistemi hardware con migliori prestazioni come, ad esempio, hard disk con maggiore capacità e migliori tempi di accesso, CPU più veloci, ecc. .



Capitolo 12. Reti Informatiche

12.1. Trasmissione dell'informazione

In uno schema generale e molto semplificato di **trasmissione dell'informazione** o **telecomunicazione** ritroviamo una **sorgente**, che emette l'informazione, una **destinazione**, che la riceve, ed un **mezzo trasmissivo o canale di trasmissione**, che consente ai dati di trasferirsi dall'origine alla destinazione:



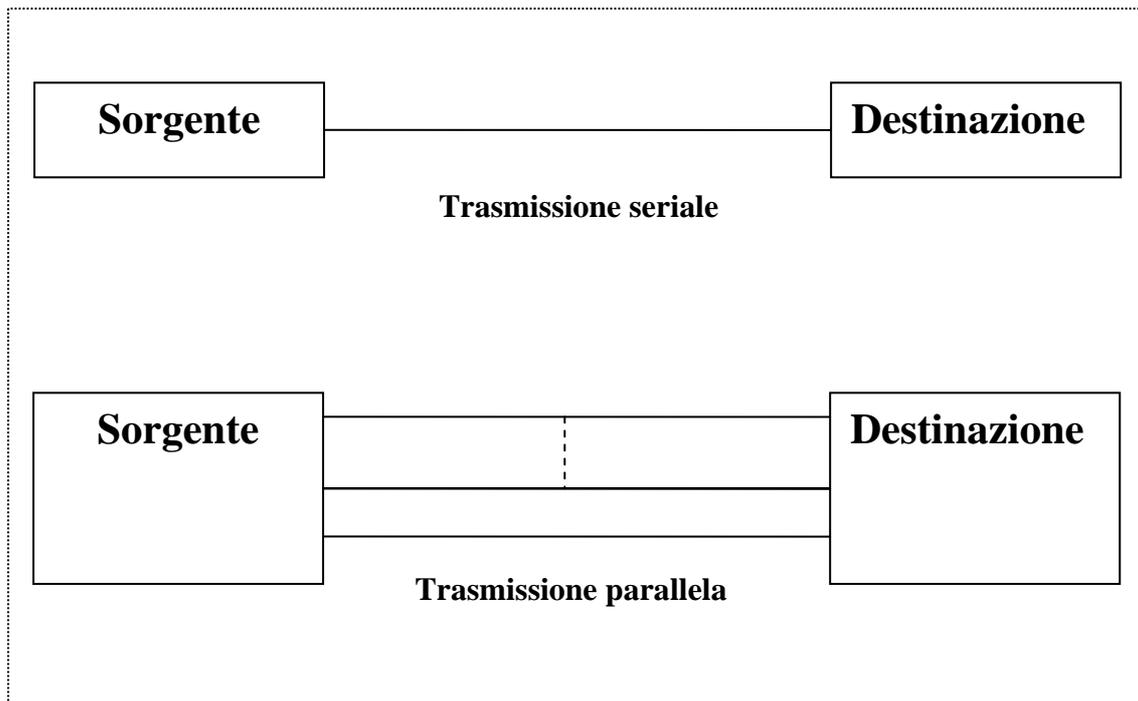
Il ruolo tra la sorgente e la destinazione può essere interscambiabile e dipende dal tipo di linea:

- **Simplex:** I dati possono viaggiare in una sola direzione; appartengono a questa categoria le comunicazioni radio televisive;
- **Half duplex:** I dati viaggiano in entrambe le direzioni ma in un sol senso alla volta; appartengono a questa categoria, per esempio, le comunicazioni che avvengono mediante walkie/talkie ed alcune reti obsolete basate su cavi coassiali;
- **Full duplex:** I dati viaggiano in entrambe le direzioni, anche contemporaneamente; appartengono a questa categoria le linee telefoniche.

La sorgente e la destinazione possono essere due sistemi di elaborazione o un sistema di elaborazione ed una risorsa, quale, per esempio, una stampante; le risorse, pertanto, possono essere classificate in:

- **Remote:** Il collegamento avviene mediante tecniche di trasmissione dati, come appena esposto, ed i due dispositivi possono trovarsi anche molto distanti tra loro;
- **Locali:** Il collegamento avviene in maniera locale, in genere mediante un cavo o comunicazioni locali wireless quali bluetooth, infrarossi e wifi.

Una linea può consentire una trasmissione **seriale**, ovvero un bit alla volta, o **parallela**, più bit contemporaneamente. La maggior parte delle comunicazioni avviene in maniera seriale. In passato le comunicazioni parallele consentivano una maggiore velocità di trasmissione; le comunicazioni seriali oggi hanno raggiunto velocità altissime, dell'ordine di Gigabit/secondo, per cui persino il collegamento locale con dispositivi periferici piuttosto veloci, come i dischi rigidi, avvengono in maniera seriale (vedi interfacce e SATA nel capitolo 3-Hardware).



Numerose sono le cause che possono deteriorare o impedire il corretto trasferimento dell'informazione dalla sorgente alla destinazione, quali, per esempio, un problema hardware ma, spesso, una errata o mancata comunicazione dell'informazione dipende da casuali interferenze, riferite con il termine **rumore**; in generale i sistemi di comunicazione sono in grado di rilevare l'errore e, a volte, di correggerlo oppure di richiedere la ritrasmissione dei dati ricevuti non correttamente; la verifica della corretta trasmissione avviene aggiungendo, in fase di trasmissione, caratteri di controllo e verificandoli in ricezione. Si tratta di soluzioni che abbiamo già visto numerose volte quale, ad esempio, il controllo mediante il bit di parità, e, in genere, la logica non è diversa da quella del controllo del codice fiscale; molti sistemi di trasmissione dati sono **fault tolerant**, tolleranza del guasto, ovvero consentono la trasmissione corretta dei dati anche in presenza di guasti; vi ricordo che il sistema RAID, di cui vi ho parlato lo scorso anno, è un sistema fault tolerant, perché consente di continuare a lavorare anche in presenza di un guasto ad un disco rigido.

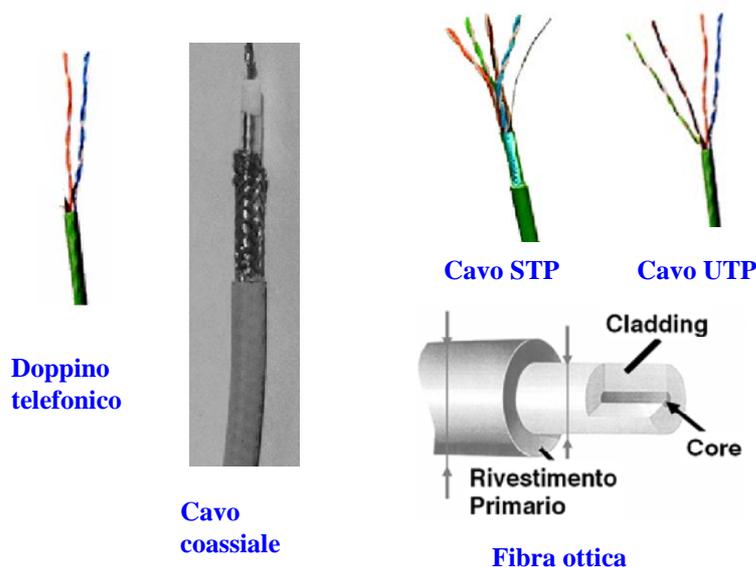
La tecnologia offre numerosi mezzi trasmissivi con caratteristiche e velocità di trasmissione dei dati assai diversificate:

- **Doppino telefonico:** si tratta del più diffuso mezzo trasmissivo, costituito da due cavi in rame avvolti a spirale; la maggior parte delle comunicazioni telefoniche ed internet viaggiano su questo tipo di cavo;
- **Cavo coassiale:** è costituito da un cavo in rame protetto dalle interferenze esterne mediante una schermatura; il cavo coassiale è molto utilizzato nelle comunicazioni radio-televisive;

- **Cavi UTP, STP:** sono costituiti da 4 coppie di cavi avvolti a spirale senza (UTP) e con (STP) schermatura; questa tipologia di cavi è molto utilizzata per la realizzazione di reti locali di cui vi parlerò tra poco;
- **Fibra ottica:** i cavi di cui in precedenza trasmettono informazione utilizzando segnali elettrici; la grande rivoluzione in atto nei sistemi di comunicazione prevede l'utilizzo di cavi in fibra di vetro in grado di trasmettere segnali ottici e non più elettrici; questa tipologia di cavo presenta numerosi vantaggi tra i quali una velocità centinaia di volte maggiore ed una eccezionale affidabilità dovuta alla eliminazione del rumore;
- **Wireless:** Le comunicazioni senza filo o wireless utilizzano onde elettromagnetiche per trasmettere l'informazione; il mezzo trasmissivo, in questo caso, è l'aria. Si tratta di una innovazione tecnologica forte che consente la libertà dai cavi, con tutti gli enormi vantaggi che ciò comporta; questo standard è basato sull'architettura cellulare, ovvero la rete viene vista e definita come un insieme di più celle, in cui ogni cella è controllata da una stazione base chiamata punto di accesso o **Access Point**.

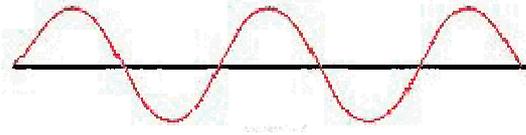
Naturalmente, nelle comunicazioni tra due stazioni, possono essere utilizzate anche altre tecnologie, come, per esempio, i **satelliti** o più tecnologie combinate fra loro.

Mezzi trasmissivi



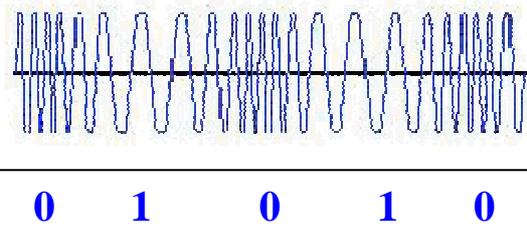
I mezzi trasmissivi utilizzano grandezze **analogiche**, segnali elettrici o ottici, mentre i sistemi di elaborazione rappresentano l'informazione in maniera **digitale**; affinché la trasmissione avvenga in maniera corretta, pertanto, è necessario, in trasmissione, convertire i segnali digitali in analogici per adattarli alla linea e, quindi, in ricezione, riconvertirli in digitale; il dispositivo elettronico che effettua questa

conversione è detto **modem**, **modulatore/demodulatore**; questa apparecchiatura è un semplice **convertitore analogico/digitale** e viceversa in grado di convertire, modulando, da digitale ad analogico e, demodulando, da analogico a digitale. Il segnale elettrico analogico, rappresentato su un asse cartesiano ortogonale, disegna una sinusoide:



E' possibile modificare le caratteristiche di questa sinusoide, per codificare lo zero e l'uno, mediante una tecnica detta modulazione; esistono tre tipi di modulazione:

- **Modulazione di ampiezza:** Viene modificata l'ampiezza dell'onda sinusoidale per codificare 0 e 1;
- **Modulazione di fase:** Viene modificato il punto di origine della sinusoide;
- **Modulazione di frequenza:** La frequenza viene aumentata per rappresentare un bit 0:



La velocità della trasmissione dipende dalla **larghezza di banda** data dalla differenza tra la frequenza massima e minima che il mezzo trasmissivo riesce a gestire.

La trasmissione seriale può avvenire in maniera **sincrona** ed **asincrona**; nella trasmissione asincrona la trasmissione di un carattere è preceduta da un **bit di start** e seguita da un **bit di stop** mentre in quella sincrona le due apparecchiature, dopo l'avvenuta sincronizzazione, scambiano dati senza soluzione di continuità ad intervalli di tempo regolari.

La velocità con la quale l'informazione viaggia dalla sorgente alla destinazione varia moltissimo in funzione dei mezzi trasmissivi, delle tecnologie disponibili e dell'hardware utilizzato; l'unità di misura della velocità di trasmissione è il **baud** che corrisponde a **1 bit/sec**:

- **Linee analogiche:** doppino telefonico 56K baud;
- **Linee digitali ISDN:** doppino telefonico 64K baud;
- **Linea ADSL:** doppino telefonico da 640K baud e, oggi, fino a 20M baud;

- **Fibra ottica:** 1G baud;
- **Wireless:** 11M baud, 22M baud, 54M baud, 108M baud;
- **Cavo coassiale:** 10M baud;
- **Cavo UTP:** 10/100 M baud;
- **Cavo FTP, STP:** 10/100/1000 M baud.

Nella comunicazione dei dati possono, ovviamente, essere utilizzate tecnologie e mezzi trasmissivi diversi, che riescono a comunicare tra loro grazie ai **media converter**, i quali, per esempio, possono convertire il segnale elettrico in ottico e viceversa oppure un segnale wireless in elettrico e viceversa; per esempio, quando si naviga in internet, la comunicazione tra il nostro PC e la centrale può essere di tipo analogico o ADSL mentre i server comunicano tra loro, in genere, mediante fibre ottiche.

12.2. Reti informatiche

Una rete informatica è un insieme di sistemi di elaborazione messi in comunicazione tra loro. Una rete informatica è rappresentata, come tutte le reti, mediante un **grafo**. Conosciamo tanti tipi di rete quali, ad esempio, le reti elettriche, idriche, ferroviarie, stradali, aeree, ecc. Ognuna di queste reti è caratterizzata da nodi e rami che li collegano: cavi elettrici, tubi, ecc. I nodi di una rete informatica sono costituiti dai **sistemi di elaborazione** o da una semplice periferica mentre i rami sono costituiti dai **mezzi trasmissivi** opportunamente gestiti da apparecchiature elettroniche. **Una rete informatica**, pertanto, viene ad essere costituita da un **sottosistema di elaboratori elettronici** e altre apparecchiature in grado di elaborare dati ed un **sottosistema di telecomunicazione**.

Una rete informatica, pertanto, appare come una organizzazione **molto complessa**; tale complessità è dovuta alla varietà dei sistemi di elaborazione e dei loro sistemi operativi, alla diversità dei mezzi trasmissivi e delle tecnologie disponibili nelle varie nazioni del mondo e nell'ambito della stessa nazione; affinché la comunicazione tra sistemi diversi con tecnologie di comunicazione diverse possa avvenire in maniera corretta è assolutamente necessario definire degli **standard**; l'organizzazione mondiale che si occupa di definire gli standard di comunicazione e non solo, ovvero l'insieme di tutte le regole e le funzionalità hardware delle diverse apparecchiature coinvolte da una rete mondiale, è l'**ISO (International Standards organization)**; il documento, prodotto dall'ISO, che riporta tutte le regole da rispettare nella realizzazione di hardware e software destinato alla comunicazione dei dati, è detto **OSI (Open System Interconnection)**.

I vantaggi di una connessione in rete possono essere così riassunti:

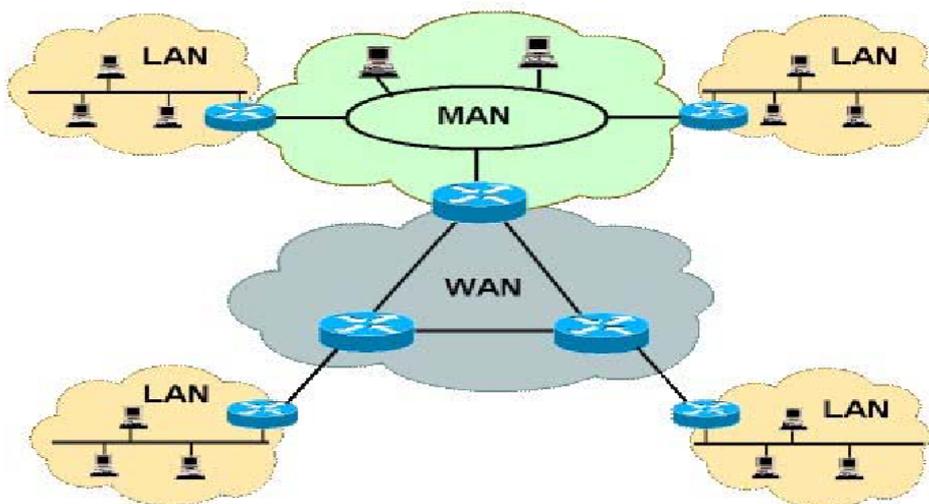
- **Condivisione delle risorse:** è possibile utilizzare risorse collegate ad altri sistemi di elaborazione attraverso i servizi di rete; in questo modo le risorse possono essere ulteriormente classificate in **locali**, come già detto, e di **rete** ovvero utilizzabili solo attraverso il collegamento in rete; l'utilizzazione delle risorse di rete è analoga a quelle locali, nel senso che l'utilizzatore può addirittura non accorgersi se una risorsa è locale o di rete. Si pensi al nostro laboratorio ed alle risorse che condividiamo quali, ad esempio, le stampanti;
- **Condivisione dei dati:** la possibilità di condividere i dati è assolutamente indispensabile nella maggior parte delle applicazioni informatiche; ad esempio, le casse di un supermercato devono condividere l'archivio magazzino allo scopo di scaricare correttamente le quantità degli articoli venduti da ciascuna di esse e procedere in maniera corretta e tempestiva al riapprovvigionamento degli articoli in esaurimento; si pensi anche alla cartella condivisa sul PC del professore nel nostro laboratorio, dalla quale prelevate gli appunti e tanti altri files; e se proprio volete, pensate al sistema bancario mondiale!
- **Scambio rapido di informazioni:** una sola parola è sufficiente a chiarire il concetto: Internet.

Da quanto detto emerge che una rete può essere costituita da un numero assai variabile di PC e la sua estensione può variare in maniera enorme; ho parlato, infatti, della rete locale del nostro laboratorio di informatica e della rete Internet; in base alla loro estensione, le reti sono classificate in:

- **PAN – Personal Area Network:** si tratta di reti domestiche costituite da 2-3 PC o un PC ed un notebook; rientrano in questa categoria anche i collegamenti tra PC e dispositivi diversi come cellulari, palmari, ecc.;
- **LAN – Local Area Network:** una rete LAN si sviluppa nell'ambito di un edificio o edifici contigui; sono caratterizzate da un'alta velocità di comunicazione; il nostro laboratorio, per esempio, è una rete locale ma, più precisamente, tutti i laboratori ed i PC disponibili nelle aule costituiscono la rete locale del nostro Istituto;
- **MAN – Metropolitan Area Network:** rete a sviluppo cittadino;
- **WAN – Wide Area Network:** si tratta di reti mondiali ovvero un nodo della rete può trovarsi, in genere, in qualsiasi punto della superficie terrestre; un esempio di tali reti è Internet;
- **GAN – Global Area Network:** e siamo fuori del nostro pianeta; in queste reti i sistemi possono trovarsi anche fuori della superficie terrestre come, per esempio, su satelliti artificiali.

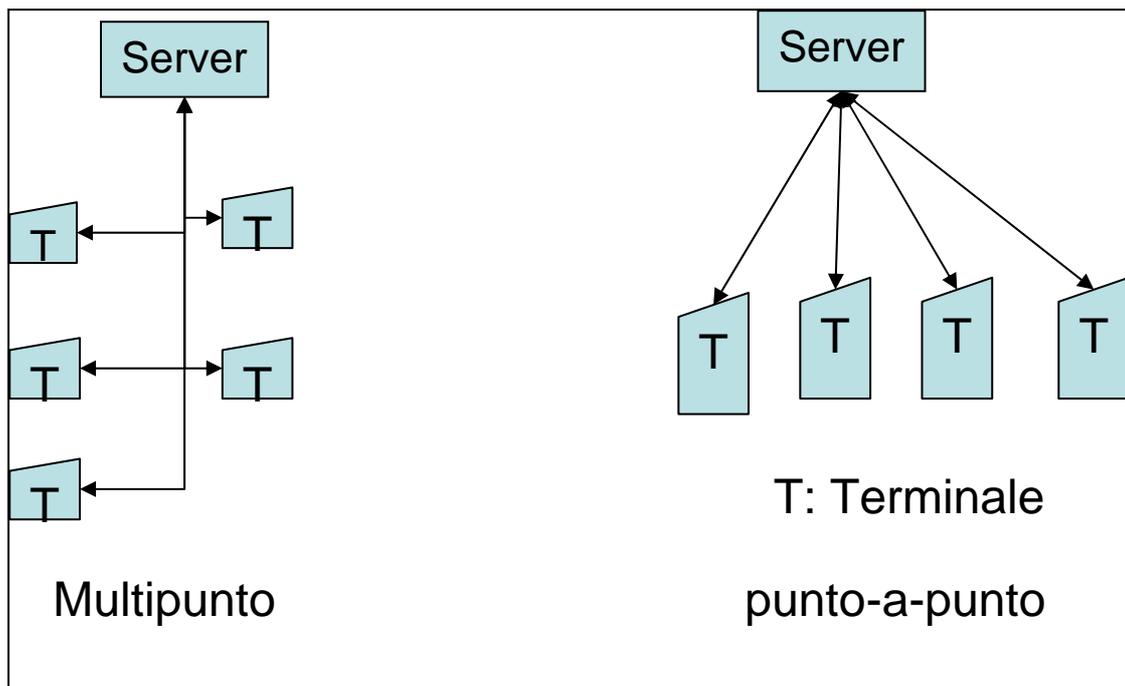
Area coperta	Distanza	Tipo di rete
Stanza	10 metri	LAN
Edificio	100 metri	LAN
Campus	1 kilometro	LAN
Città	10 chilometri	MAN
Area metropolitana	100 chilometri	MAN
Stato o Nazione	1.000 chilometri	WAN
Continente	5.000 chilometri	WAN
Pianeta	10.000 chilometri	WAN

Un esempio di come reti eterogenee possono essere interconnesse è mostrato nella figura seguente:



Le reti possono essere classificate anche rispetto alla **topologia** ovvero al modo con il quale i vari nodi possono comunicare tra loro; la rappresentazione topologica di una rete può essere irregolare, ovvero non riconducibile ad alcuno schema geometrico, oppure regolare. Distinguiamo le seguenti **topologie di rete**:

- **Collegamento punto a punto:** Il collegamento tra diversi dispositivi avviene mediante connessioni dirette; una interruzione su un ramo non impedisce ad altri nodi di comunicare;
- **Collegamento multipunto:** in un collegamento multipunto più dispositivi condividono lo stesso canale di trasmissione; una interruzione sul canale di trasmissione impedisce la comunicazione tra tutti i nodi della rete;



- Reti a **BUS**: in una rete con topologia a bus, tutti i nodi costituenti la rete si affacciano su un bus condiviso che consente di trasmettere le informazioni tra una qualsiasi loro coppia; una interruzione in qualsiasi punto del bus non consente alcuna comunicazione sulla rete;
- Reti a **stella**: in una rete a stella i nodi sono tutti collegati ad un dispositivo centrale detto **centro stella** che agisce da server di rete e gestisce il traffico tra i vari nodi; una interruzione su un ramo isola un solo nodo e non impedisce la comunicazione tra tutti gli altri; il centro stella è il componente delicato, un suo guasto non consente alcuno scambio di dati tra i nodi; si tratta della tipologia più utilizzata ed offre un vantaggio enorme: per poter collegare due sottoreti a stella, in modo che diventino un'unica rete, è sufficiente mettere in comunicazione un nodo qualsiasi di una delle due con uno qualsiasi dell'altra;
- Reti ad **anello**: è una estensione della topologia a BUS; si tratta, infatti, di un BUS chiuso; anche in questo caso una interruzione in qualsiasi punto della rete isola tutti i nodi;
- **Irregolari**: tutte le reti non riconducibili alle topologie precedenti si dicono irregolari; sono irregolari le reti di grande estensione quali GAN, WAN e MAN mentre sono regolari le reti LAN.

quando un PC entra, dopo essersi autenticato, in un dominio, può utilizzare le risorse di rete secondo diritti gestiti dal server di dominio;

- **DHCP Server:** assegna un indirizzo univoco, detto indirizzo IP, di cui parlerò in seguito, ai sistemi di elaborazione collegati in rete;
- **DNS Server:** server che traducono indirizzi simbolici di macchine, come, per esempio, popmail.libero.it, o indirizzi URL, come www.barilla.it, in indirizzo IP numerico univoco, di cui parlerò in seguito.

Un unico sistema di elaborazione può svolgere diverse funzioni server; avere sulla stessa rete più di un server dipende dal carico di lavoro che bisogna gestire. Per esempio, nel nostro laboratorio di informatica, la postazione del docente costituisce sia un file server che un print server. In alcuni casi i server sono piccoli dispositivi dedicati a svolgere un determinato lavoro; ad esempio oggi un print server, che consente l'accesso alla stessa stampante a tutti i client di una rete, è un piccolo dispositivo elettronico non più grande di un pacchetto di sigarette ed a volte è integrato nella stampante; è evidente che, se c'è la necessità di gestire decine di stampanti con priorità e diritti di accesso, ho bisogno di un dispositivo più complesso.

➤ **PEER to PEER:** ogni sistema presente sulla rete può essere sia client che server.

Indipendentemente dalla topologia di rete, i sistemi client e server scambiano informazioni attraverso quello che ho definito sottosistema di comunicazione rappresentato da un grafo; attraverso una rete mondiale, come per esempio Internet, comunicano centinaia di milioni di sistemi di elaborazione scambiandosi pacchetti informativi detti genericamente **messaggi**. Un messaggio deve essere instradato dalla sorgente alla destinazione attraverso un cammino ovvero una successione di rami contigui sul grafo che rappresenta la rete; ebbene, per poter trasmettere un messaggio dalla sorgente alla destinazione, è necessaria una scelta, detta **commutazione**, dei rami che consentano un collegamento tra la sorgente e la destinazione; la commutazione può avvenire secondo le seguenti due modalità:

- **Commutazione di circuito:** tra la sorgente e la destinazione viene individuato un **percorso fisico** attraverso il quale l'informazione raggiunge la destinazione, come avviene per le conversazioni telefoniche;
- **Commutazione di messaggio:** il messaggio viene inviato al nodo intermedio più vicino attraverso un canale trasmissivo libero quindi memorizzato da tale nodo intermedio e rispedito appena possibile ad un altro fino ad arrivare a destinazione; questa tecnica è detta **store and forward**;
- **Commutazione di pacchetto:** il messaggio viene suddiviso in **pacchetti numerati** ognuno dei quali raggiunge la destinazione anche attraverso percorsi diversi; all'arrivo viene ricostruito, a partire dai pacchetti, il messaggio originario; non necessariamente la ricezione dei pacchetti avviene secondo l'ordine di invio; è possibile, in altri termini, che un pacchetto con numerazione più alta arrivi prima

di uno o più di quelli che lo precedono nell'ambito del messaggio. Questa tecnica di commutazione è utilizzata nelle reti WAN, come, ad esempio, Internet;

- **Collegamento dedicato:** raramente si utilizza un collegamento dedicato tra una sorgente ed una destinazione ovvero l'uso esclusivo di un canale trasmissivo, perché potrebbe comportare, ovviamente, una spesa eccessiva.

Un messaggio deve contenere, oltre all'informazione che si vuole veicolare dalla sorgente alla destinazione, anche l'indicazione del **mittente** e del **destinatario**; un pacchetto conterrà anche un **contatore**, che ne indica la sua posizione nell'ambito del messaggio.

Il continuo e velocissimo progresso dell'Informatica e delle Telecomunicazioni consente la nascita di una nuova scienza chiamata **Telematica**, parola derivante dalla contrazione di **Telecomunicazioni** e **Informatica**.

12.2.1. Telematica: Hardware

Abbiamo ormai ben capito che tutto quello che ha a che fare con l'informatica ha una componente hardware ed una software; non sfugge a questa regola una rete informatica. Numerose sono le componenti hardware che ritroviamo in una rete informatica.

Vi dico subito che questi componenti possono essere classificati in **apparati attivi** e **passivi**. Un apparato passivo lascia passare l'informazione senza eseguire alcuna elaborazione; sono apparati passivi, per esempio, i mezzi trasmissivi di cui ho parlato ad inizio capitolo. Gli apparati attivi, invece, gestiscono, elaborano o instradano correttamente il segnale elettrico o ottico affinché il messaggio arrivi correttamente dalla sorgente alla destinazione. Come vedremo, alcune funzioni, necessarie per la gestione e la sicurezza di una rete, possono essere svolte da un componente hardware oppure un software che gira su un server o un client.

L'hardware che ritroviamo su una rete è assai diversificato; con riferimento all'hardware, ho già parlato dei server e dei client che sono i nodi della rete, dei mezzi trasmissivi che ne costituiscono la sottorete di comunicazione e del modem che adatta la linea analogica al digitale dei sistemi di elaborazione; tutto questo, però, non è sufficiente a garantire il corretto e sicuro trasferimento dell'informazione; in una rete, in genere, ritroviamo anche il seguente altro hardware:

- **Scheda di rete:** ogni PC o server deve disporre di una scheda elettronica che ne consente l'interfacciamento con la rete; sono oggi disponibili schede di rete **wired**, ovvero che utilizzano un cavo, in genere UTP o STP, per il collegamento alla rete, o **wireless** che utilizzano la più moderna tecnologia senza fili;
- **HUB** o **concentratore:** è il gestore del centro stella; ad esso sono collegati i vari sistemi di elaborazione di una rete ed ha il compito di smistare il traffico da un

sistema all'altro; in pratica un HUB ha una scarsa capacità di elaborazione dei messaggi e, in genere, un messaggio viene inviato a tutti i PC collegati alla rete, comunicazione detta **broadcast**, anche se solo uno di questi lo gestirà, quello che riconoscerà il proprio indirizzo contenuto all'interno del messaggio;

- **SWITCH:** come l'HUB ma invia il messaggio solo al destinatario; considerata la crescente riduzione dei prezzi, oggi sul mercato sono disponibili solo switch; una delle caratteristiche fondamentali di uno switch è il **numero di porte** ovvero il numero massimo di PC che è possibile collegare; nel caso in cui la rete cresce come numero di postazioni, non è necessario sostituire il concentratore ma collegarlo in cascata con un altro, utilizzando due delle stesse porte destinate ai PC;
- **Bridge:** si tratta di un dispositivo che ha una funzione molto simile allo switch, perché consente di collegare due reti tra loro; in maniera molto semplificata, può essere considerato uno switch con due sole porte, perché non avrebbe senso parlare di un ponte con più di due connessioni; è forte l'analogia con un ponte che unisce le due rive di un fiume;
- **Router o instradatore:** i router sono i gestori del traffico sulla rete; instradano il messaggio dalla sorgente alla destinazione; i router sono utilizzati, nell'ambito di una rete locale, per collegarla alla rete internet; i router attraverso la funzione **NAT (Network Address Translation)** sono in grado di ridistribuire il traffico proveniente dalla rete geografica Internet verso la rete locale e viceversa ovvero, da un lato, raccogliere le richieste di tutti i PC locali ed inviarle sulla rete Internet e, dall'altra, indirizzare ciascun pacchetto proveniente da Internet verso il PC, nell'ambito della rete locale, che lo ha richiesto; in questo modo i numerosi PC della rete locale possono condividere ed utilizzare la stessa unica connessione a Internet;
- **Gateway:** ha funzioni molto simili ad un router, con la differenza di consentire il dialogo tra dispositivi che utilizzano protocolli diversi; è possibile spiegare la differenza tra un router ed un gateway con qualche esempio: due persone, che parlano a telefono utilizzando la stessa lingua, hanno bisogno solo della linea che trasmette la voce dalla sorgente alla destinazione; se i due interlocutori utilizzano lingue diverse, però, hanno bisogno di un sistema di traduzione automatica, che trasformi quanto detto da uno dei due nella lingua dell'altro e viceversa.

Un altro esempio molto interessante è costituito dallo scambio di mail tra utenti che utilizzano protocolli diversi; in questo caso c'è bisogno di un software che effettui la traduzione da un protocollo all'altro.

Un router, pertanto, trasmette i dati dalla sorgente alla destinazione senza operare alcuna trasformazione su di essi, al più, si fa per dire, li divide in pacchetti, gestisce il traffico, lo indirizza, ecc., mentre il gateway elabora le informazioni da trasmettere adattandole alla destinazione prima di ritrasmetterli;

- **ACCESS POINT:** Un access point gestisce il traffico proveniente da PC o portatili con schede di rete wireless; ovviamente un access point si collega alla rete wired in modo tale che, indipendentemente dal mezzo trasmissivo, la rete è unica; il vantaggio degli access point? La libertà dai cavi!
- **Firewall: il firewall migliora la sicurezza sulla rete intercettando ed eliminando il traffico pericoloso derivante da tentativi di accesso non autorizzati alla rete;** è possibile proteggere la rete da accessi non autorizzati mediante firewall hardware o software; Windows XP, dal service pack 2, integra le funzioni firewall; è possibile utilizzare software firewall di altri produttori tra i quali vi ricordo Norton Internet Security, una suite di programmi che comprende, oltre alla protezione firewall, anche un antivirus, un antispamming ed un controllo dei genitori, che consente di monitorare e rendere sicura la navigazione dei minorenni, impedendo loro l'accesso a siti pericolosi. Vi sono anche soluzioni freeware, quali ad esempio, Zone Alarm.

12.2.2. Telematica: Software

Le apparecchiature hardware, descritte nel paragrafo precedente e che consentono la comunicazione di messaggi tra sistemi di elaborazione, sono, ovviamente, gestite da software. Ogni apparecchiatura è gestita, inoltre, da un **firmware** interno che ne consente la configurazione; a tal proposito vedremo come si configura un access point ed un router. **Il dialogo tra i vari componenti di una rete è regolato da un insieme di regole detto protocollo e implementato mediante software.**

Un protocollo è, quindi, un insieme di regole che consente la corretta trasmissione dell'informazione. I protocolli sono classificati in **BOP, Bit Oriented Protocol**, e **BCP, Byte Oriented Protocol**, e fanno riferimento, rispettivamente, ad una trasmissione il cui elemento atomico è il bit o il byte.

In genere la gestione della trasmissione può avvenire in maniera centralizzata o distribuita secondo i seguenti protocolli:

- **Protocollo polling/selecting:** si tratta di un protocollo utilizzato nei collegamenti multipunto; la gestione centralizzata, mediante un server, della comunicazione avviene mediante le fasi di **polling** e di **selecting**.

Il polling è utilizzato per interrogare i terminali e, eventualmente, ricevere pacchetti da essi; il server invia un messaggio di interrogazione ad un terminale, utilizzandone l'indirizzo univoco; se il terminale ha pacchetti da trasmettere risponde all'interrogazione; i terminali vengono interrogati ciclicamente ed uno alla volta in maniera tale che la linea comune risulta, in ogni istante, utilizzata da una sola comunicazione.

Nella fase di **selecting**, invece, il server invia un messaggio destinato ad un terminale; il messaggio inviato sulla rete raggiunge tutti i terminali ma solo uno riconosce il proprio indirizzo, riportato nell'ambito del pacchetto, e lo gestisce mentre gli altri lo ignorano.

Questo protocollo garantisce che la linea sia utilizzata da una sola conversazione alla volta ed evita il problema della collisione di cui vi dirò tra poco;

- **Protocollo a contesa:** i nodi della rete inviano un pacchetto senza preoccuparsi che la linea sia libera o occupata; può succedere, pertanto, che due nodi inviino dati sulla rete quasi contemporaneamente; in un tal caso si parla di **collisione** che impedisce ai pacchetti di raggiungere le destinazioni. Una tecnica di gestione delle collisioni è detta **CSMA/CD** (Carrier Sense Multiple Access/ Collision Detection): un nodo, che invia un pacchetto, resta in ascolto ovvero legge quanto viaggia sulla rete e, se diverso da quanto inviato, si accorge della collisione, aspetta un tempo casuale e invia di nuovo il pacchetto sulla rete;
- **Protocollo Token Passing:** si tratta di un protocollo utilizzato su reti ad anello; sulla rete viaggia un pacchetto detto **token**, **gettone**, che indica linea libera; il nodo che trattiene il gettone, ovvero lo elimina dalla rete, invia un messaggio e, al termine della comunicazione, immette sulla rete il gettone in modo che un altro nodo possa comunicare. Con questa tecnica si evitano le collisioni perché l'invio di un messaggio avviene quando la linea è sicuramente libera;
- **Protocollo TCP/IP (Transport Control Protocol / Internet Protocol):** è il protocollo a commutazione di pacchetto più importante, perché utilizzato sia per la gestione di piccole reti locali che di reti geografiche; si tratta del protocollo utilizzato dalla rete mondiale internet.

Considerata l'enorme importanza che questo protocollo riveste nella gestione di una rete e la sua grande diffusione, ritengo opportuno descriverlo in un paragrafo a parte e, soprattutto, di svolgere numerose esercitazioni, che vi consentano di correttamente configurare e gestire una rete locale e l'accesso a Internet, non esclusa la configurazione di apparecchiature ormai molto diffuse quali router e access point; la teoria è fondamentale ma è l'applicazione che la fissa in maniera indelebile nelle nostre conoscenze.

12.2.2.1. Il protocollo TCP/IP

Il protocollo **TCP/IP**, **Transport Control Protocol/Internet Protocol**, è sicuramente il più diffuso per le reti di qualsiasi dimensione. Esso si basa sull'assegnazione di un indirizzo univoco, detto **indirizzo IP**, attribuito ad ogni sistema della rete. Si tratta di un indirizzo assegnato dal software da non confondere con **l'indirizzo hardware MAC (Media Access Control)**, anch'esso univoco per qualsiasi apparecchiatura di rete; in altre parole ogni scheda di rete, router, switch o

altra apparecchiatura attiva presente su una rete, ha un indirizzo fisico univoco assegnato dal produttore; così un dispositivo di rete può essere indirizzato mediante l'indirizzo fisico MAC o mediante l'indirizzo TCP/IP che, come vedremo, può essere assegnato in maniera statica o dinamica.

Un indirizzo IP è costituito da **32 bit** ovvero 4 byte ognuno dei quali, ovviamente, può avere un valore, in base 10, che va da 0 a 255; tali valori sono separati da un punto, così un possibile indirizzo IP assume la forma 192.168.0.123.

Un indirizzo IP è distinto in due parti:

Indirizzo della sottorete: Net-ID	Indirizzo del sistema: Host-ID
--	---------------------------------------

L'indirizzo **Net-ID** identifica la rete mentre l'**Host-ID** il sistema, PC o altra apparecchiatura, nell'ambito della rete; due sistemi possono comunicare solo se appartengono alla stessa rete. Due indirizzi, nell'ambito di ciascuna rete, vengono utilizzati per scopi particolari:

- **Indirizzo di rete:** tutti i bit della parte host sono impostati a zero, esempio 192.168.0.0;
- **Indirizzo broadcast:** tutti i bit della parte host sono impostati a 1, per esempio 192.168.255.255; l'indirizzo broadcast è utilizzato per inviare un pacchetto a tutti i nodi della rete.

Il numero di host possibili in una rete, pertanto, dipende da quanti bit sono utilizzati per la parte Host-ID; per esempio se in una rete 16 bit identificano la rete e 16 gli host, sarà possibile indirizzare $2^{16}-2=65534$ host; al numero degli host indirizzabili occorre sottrarre gli indirizzi di rete e broadcast. Il numero di bit, utilizzati per indicare la parte Net-ID, è specificato alla fine dell'indirizzo IP e separato da quest'ultimo da “/”; così, per esempio, con riferimento all'indirizzo 192.168.0.123/16 di un host, possiamo dire che:

- **192.168** è la parte Net-ID con indirizzo di rete 192.168.0.0 e indirizzo broadcast 192.168.255.255;
- **0.123** è l'indirizzo host;
- **i possibili indirizzi host** vanno da 192.168.0.1 a 192.168.255.254

Il numero di bit di un indirizzo IP vanno ripartiti tra il Net-ID e l'Host-ID e, proprio in funzione di questa ripartizione, le reti basate su IP sono classificate in cinque classi, individuate attraverso opportuna configurazione dei primi bit:

- **Classe A:** Net-ID di 8 bit e Host-ID di 24 bit; pertanto una rete di classe A può contenere fino a $2^{24}-2=16.777.214$ host; si tratta di solo $2^7=128$ reti con moltissimi host; sono individuate dal primo bit che è sempre 0;

- **Classe B:** Net-ID di 14 bit e Host-ID di 16 bit; si tratta di reti di medie dimensioni ovvero di $2^{14}=16384$ reti con $2^{16}-2=65534$ host; sono individuate mediante la prima coppia di bit che è sempre 10;
- **Classe C:** Net-ID di 21 bit e host-ID di 8 bit pari a $2^{21}=2.097.152$ reti con $2^8-2=254$ host; sono reti di piccole dimensioni individuate da 110;
- **Classe D:** Riservati al multicasting ovvero all'indirizzamento di gruppi di host; individuate da 1110;
- **Classe E:** Riservata per usi futuri e individuate da 11110.

Alcune classi di indirizzi IP sono **riservate per le reti LAN**; tali indirizzi, in altre parole, non sono utilizzati da sistemi nell'ambito di una rete pubblica; chiunque realizzi una rete locale, deve utilizzare, per gli indirizzi locali degli host, una delle seguenti sottoreti:

Classe	Indirizzo di rete	Indirizzi host
A	10.0.0.0/8	Da 10.0.0.1 a 10.255.255.254
B	172.16.0.0/12	Da 172.16.0.1 a 172.31.255.254
C	192.168.0.0/16	Da 192.168.0.0 a 192.168.255.254

Un indirizzo IP è costituito anche da una **SUBNET MASK**, costituita da 32 bit comincianti con una serie di 1 seguita da una serie di 0, ad esempio 255.255.255.0; la subnet mask consente di realizzare delle sottoreti nell'ambito della stessa rete; l'indirizzo della sottorete viene individuato mediante un AND tra l'indirizzo IP e la subnet mask; solo i PC facenti parte della stessa sottorete possono scambiarsi messaggi.

L'indirizzo IP viene assegnato ad una interfaccia di rete e non ad un sistema di elaborazione; in altre parole, se un sistema dispone di più di una scheda di rete, esso avrà tanti indirizzi MAC e IP per quante schede sono state installate; per esempio, i notebook hanno quasi sempre due interfacce di rete: una wired, ovvero utilizzabile mediante un cavo STP o UTP, ed una Wireless, ossia che non necessita di fili; ognuna di queste interfacce di rete ha un indirizzo MAC univoco e per ognuna di esse il sistema riceve un indirizzo IP univoco diverso.

L'indirizzo IP può essere assegnato in maniera **statica** o **dinamica**.

Con l'assegnazione statica, a cura dell'amministratore della rete, un PC utilizza sempre lo stesso indirizzo IP che non varia, pertanto, tra una accensione e l'altra; l'amministratore della rete deve stare molto attento ad assegnare indirizzi diversi, per la parte Host-ID, ai vari sistemi altrimenti ad un PC, avviato per ultimo ed avente un indirizzo IP già assegnato, viene disattivata l'interfaccia di rete con l'impossibilità di usufruire dei servizi.

L'assegnazione dinamica, invece, avviene mediante un **server DHCP**, al quale il sistema lo richiede subito dopo l'avvio; l'assegnazione dinamica, quindi, richiede la presenza di un server DHCP, anche se Windows riesce ad assegnare un indirizzo in maniera dinamica anche senza la presenza di un tale server; vi sconsiglio di utilizzare l'assegnazione dinamica in assenza di un server DHCP, perché, nel caso in cui i PC ricevono un indirizzo IP relativo a reti diverse (NET-ID), non riescono a scambiarsi pacchetti. Con l'assegnazione dinamica, pertanto, i nodi della rete utilizzano probabili indirizzi diversi ad ogni loro avvio o accensione. L'assegnazione dinamica libera l'amministratore della rete dal compito dell'assegnazione degli indirizzi IP e si rivela molto comoda soprattutto in reti di medie/grandi dimensioni. In laboratorio vedremo come si assegna un indirizzo IP ad un PC che utilizza Windows XP.

Il protocollo TCP/IP non si occupa solo dell'instradamento e della consegna dei pacchetti dall'origine alla destinazione, ma provvede anche ad indicare la **natura del pacchetto**; in altri termini, un pacchetto deve essere gestito dalla destinazione, che ha bisogno di sapere se il pacchetto fa riferimento, per esempio, ad un messaggio di posta elettronica, ad una pagina WEB, allo scaricamento di un file da Internet, ecc.; l'indirizzo IP, pertanto, porta anche l'indicazione del servizio cui il pacchetto fa riferimento, in modo tale che esso possa essere correttamente "passato" al programma che lo deve gestire; per questo, ogni sistema ha **punti di arrivo dei pacchetti**, chiamati **porte**; ogni porta è identificata mediante un intero positivo e, quindi, fa riferimento al tipo di servizio cui il pacchetto è destinato, consentendo l'individuazione del programma che, nell'ambito del sistema, lo deve gestire. L'indirizzo IP viene così ad essere seguito dall'indicazione del servizio: 132.125.16.214:80, dove 132.125.16.214 è l'indirizzo IP e 80 la porta o il servizio cui il pacchetto ricevuto fa riferimento. **L'insieme dell'indirizzo IP e della porta è chiamato socket**. La maggior parte degli applicativi utilizzati su reti WAN sono di tipo CLIENT/SERVER, in cui i client utilizzano i servizi messi a disposizione dai server; si è ritenuto opportuno, pertanto, assegnare numeri di porta, detti **well-known ports**, ai servizi più comunemente messi a disposizione dai server; tali porte non possono essere utilizzate per scopi diversi da quelli assegnati. Una porta è individuata da un numero a 16 bit, da 0 a 65535; tale numerazione è distinta in due parti:

➤ **da 0 a 1023:** well known ports, alcune delle quali sono riportate nella seguente tabella:

Porta	Protocollo	Servizio
21	FTP	Trasferimento file
22	SSH	Terminale virtuale criptato
23	TELNET	Terminale virtuale non criptato
25	SMTP	Posta in uscita
53	<i>DNS</i>	Dominio

80	HTTP	WWW
110	POP	Posta in entrata

- **da 1024 a 65535:** porte libere che ogni applicativo può utilizzare per fornire o richiedere servizi.

Il protocollo IP appena descritto è relativo alla sua **versione 4, IPv4**; la rete Internet è cresciuta al di là di ogni previsione, per cui è possibile che, tra alcuni anni, non vi siano più indirizzi IP liberi; ciò comporta l'impossibilità di aggiungere nuovi sistemi di elaborazione alla rete, quindi nessun altro client e, soprattutto, nessun altro server. Sembra impossibile che ciò possa avvenire, perché con 32 bit si indirizzano ben $2^{32} = 4.294.967.296$ nodi, però, a causa della scomposizione degli indirizzi in net-id e host-id, in effetti non tutti questi indirizzi sono utilizzabili.

Per risolvere questo problema è stato recentemente introdotto il **nuovo protocollo IP versione 6, IPv6**, che presenta i seguenti notevoli vantaggi rispetto al suo predecessore:

- **Spazio di indirizzamento maggiore:** IPv4 si basa su un indirizzamento a 32 bit mentre IPv6 utilizza **indirizzi di ben 128 bit**; questo consentirà di evitare l'utilizzo del protocollo NAT, assegnando ad ogni PC un indirizzo univoco; vi ricordo che il NAT consente a tutti i PC della rete di condividere un unico indirizzo IP pubblico (vedi router);
- **Supporto al "mobile computing":** connettività illimitata e completa, attraverso la rete per la telefonia mobile GPRS ed UMTS, per PDA (palmari), telefoni cellulari, ecc.;
- **Gestione integrata della sicurezza:** IPV6 include tecniche di crittografia, autenticazione e compressione.

Concludo il paragrafo con un brevissimo cenno al **protocollo UDP, User Datagram Protocol**; a differenza del protocollo TCP, il protocollo UDP non fornisce comunicazioni affidabili, perché non utilizza tecniche di controllo degli errori e non prevede che il destinatario comunichi la corretta ricezione dei pacchetti; per questo un pacchetto può essere perso, ovvero non raggiunge la destinazione, o può arrivare in ordine diverso da quello di partenza; pur tuttavia, essendo meno ricco di regole, si rivela un protocollo molto agile, che semplifica e velocizza la comunicazione tra sorgente e destinazione; in genere è utilizzato in ambito di reti locali e per il trasferimento di piccoli volumi di dati, per esempio, un sol pacchetto.

12.2.2.2. Altri protocolli

Il protocollo TCP/IP, come detto, si occupa dell'instradamento e della consegna dei pacchetti dall'origine alla destinazione; una volta arrivato a destinazione, il pacchetto deve essere gestito da un programma applicativo, individuato, come detto, mediante la porta o servizio; questo programma, a sua volta, deve rispettare le regole imposte da un altro protocollo, che consente di gestire correttamente l'informazione ricevuta e che è lo stesso utilizzato da chi ha inviato il messaggio. Sto parlando dei seguenti protocolli:

- **HTTP (Hyper Text Transfer Protocol):** è il protocollo utilizzato per lo scambio di pagine WEB in Internet; si tratta di codice di programma scritto con il **linguaggio di programmazione HTML** ed eseguito in locale dal Browser, quale Internet Explorer;
- **HTTPs - Hyper Text Transfer Protocol Secure:** è una estensione del protocollo HTTP al quale è aggiunta la crittografia per migliorare la sicurezza e la segretezza delle comunicazioni tra la sorgente e la destinazione; si tratta di un protocollo utilizzato da banche e tutte le volte in cui la sicurezza assume una importanza fondamentale;
- **POP - Post Office Protocol e SMTP - Simple Mail Transfer Protocol:** protocolli non criptati, utilizzati per lo scambio di E-Mail, rispettivamente, in entrata ed in uscita;
- **FTP - File Transfer Protocol:** protocollo utilizzato per lo scambio di file e cartelle; è un protocollo che rende questa operazione molto semplice, perché offre all'utente la possibilità di gestire file e cartelle remoti con una interfaccia del tutto simile a gestione o esplora risorse di WINDOWS; così l'utente può trasferire file e cartelle da e su un server remoto, rinominare, cancellare, ecc., con la stessa interfaccia utilizzata per gestire i volumi locali;
- **Telnet:** Terminale virtuale; consente il collegamento a sistemi remoti, simulando un terminale costituito da monitor e tastiera; si tratta, in genere, di una interfaccia testuale;
- **DNS - Domain Name Service:** per semplificare l'utilizzabilità dei siti internet e l'accessibilità ai server, i sistemi mettono a disposizione dell'utente nomi simbolici, come PCProf, PC01, popmail.libero.it, ecc., per indicare server sulla rete e, come www.libero.it, www.barilla.it, www.nutella.it, ecc., per indicare siti WEB; noi sappiamo, però, che l'individuazione di un sistema di elaborazione in rete o di un WEB Server, avviene mediante il suo indirizzo IP; ebbene il protocollo DNS consente di tradurre l'indirizzo o il nome simbolico in indirizzo IP;
- **NNTP - News Network Transfer Protocol:** si tratta di un protocollo utilizzato per lo scambio di news; è possibile iscriversi ad un gruppo costituito da persone interessate allo scambio di informazioni o a discussioni su un certo argomento; l'iscrizione ad un tale gruppo consente di ricevere, nella propria casella di posta elettronica, tutte le news prodotte o inviate dagli iscritti;

- **SSL (Secure Sockets Layer):** Protocollo di crittografia, per migliorare la sicurezza nella trasmissione dell'informazione.

12.2.3. Il modello OSI

Nei paragrafi precedenti ho descritto le numerose componenti hardware e software di una rete; affinché tante e così diversificate apparecchiature possano essere correttamente gestite dal software, è necessario definire uno standard complesso, fortemente articolato ed efficiente; vi ricordo che l'obiettivo è quello di consentire centinaia di milioni di scambi contemporanei di pacchetti, per esempio sulla rete mondiale Internet, e non solo, se si considera che la stessa rete fisica è utilizzata per le conversazioni telefoniche e, ultimamente, anche per TV e radio.

Di questo, come detto, si occupa l'ISO. **Il modello OSI**, descritto dall'**ISO**, è organizzato in **7 livelli o strati**; ogni livello può dialogare con lo strato precedente, dal quale riceve servizi, e quello successivo, al quale fornisce servizi:

- 1. Livello 1 - Fisico:** è il livello che si occupa della trasmissione dei segnali elettrici o ottici che codificano i bit; è il livello che interfaccia l'hardware;
- 2. Livello 2- Datalink:** permette il trasferimento dei pacchetti attraverso il livello fisico, gestendo il controllo ed il recupero degli errori;
- 3. Livello 3 – Rete:** Si occupa della gestione della connessione secondo una delle tecniche di commutazione viste in precedenza;
- 4. Livello 4 – Trasporto:** Mentre i livelli precedenti si occupano di connessioni tra nodi contigui, il trasporto si occupa solo della sorgente e della destinazione; per questo è anche detto **livello end to end**;
- 5. Livello 5 – Sessione:** consente l'utilizzo di indirizzi simbolici e, eventualmente, consente la tariffazione per l'addebito della spesa derivante dall'utilizzo della rete;
- 6. Livello 6 – Presentazione:** trasforma i dati da trasmettere in un formato standard e offre servizi quali la crittografia e la compressione;
- 7. Livello 7 – Applicazione:** è il livello che interfaccia l'utente; siamo, per intenderci, al protocollo HTTP, SMTP, ecc.

12.3. Internet

Nel **1969** veniva progettata ed entrava in funzione negli Stati Uniti D'America **ARPANET**, una rete voluta dal Dipartimento della Difesa per consentire, ai ricercatori americani prima e della NATO dopo, di scambiarsi idee, opinioni, progetti; la cosa più importante era la possibilità di comunicare anche in presenza di

uno o più guasti sulla rete ed anche in caso di guerra; si era, infatti, all'epoca della guerra fredda e la cosa più importante era comunicare non certo il tempo necessario; nasceva così una rete a commutazione di pacchetto che nel giro di qualche decennio, ormai inutili gli scopi militari, veniva utilizzata per scopi civili, fino a diventare **INTERNET**, la rete delle reti, che ormai mette in comunicazione centinaia di milioni di macchine in tutto il mondo e che offre decine di interessanti servizi; possiamo anche dire, semplicemente, che **Internet è una rete geografica**.

Il protocollo utilizzato da Internet è, ovviamente, il TCP/IP; la connessione a internet, che permette di usufruire dei numerosi servizi messi a disposizione degli utenti, richiede un PC, oggi addirittura un semplice telefonino o un palmare, un modem, una linea telefonica ed un account su un **ISP (internet Service Provider)**, che fornisce, attraverso i server messi a disposizione, l'accesso alla rete delle reti. La rete telefonica tradizionale è oggi affiancata da nuove tecnologie, che consentono l'accesso a Internet in assoluta libertà, come il **GPRS** e l'**UMTS**, che consentono l'accesso disponendo di un semplice telefonino o appositi adattatori **USB** o **PCMCIA**, ed il **WIFI**, che vede di giorno in giorno aumentare il numero degli **Hot Spot**, i quali consentono l'accesso alla rete senza l'ausilio di fili.

I servizi disponibili sono numerosi e l'elenco che vi riporto è sicuramente riduttivo, ma dà una precisa idea, ammesso che ancora vi sia sconosciuta, delle grandi potenzialità della rete Internet:

- **WWW – World Wide WEB:** è il mondo delle informazioni, dei siti e delle pagine WEB; ormai si parla di miliardi di pagine accessibili in maniera facile, anche grazie ai **motori di ricerca**; i siti sono organizzati gerarchicamente con indirizzi di primo livello e successivi; ogni livello è gestito da una **Authority** che assegna gli indirizzi agli utenti che ne fanno richiesta; per esempio, l'Authority italiana assegna gli indirizzi con suffisso “**it**” ovvero il cui primo livello è “**it**”; un elenco dei suffissi più utilizzati sono riportati nel capitolo 4-Windows;
- **Motori di ricerca:** l'informazione è accessibile nel WWW attraverso l'indirizzo o **URL (Unique Resource Locator)** del sito che la contiene; non sempre si è a conoscenza di tale indirizzo ed ecco un servizio eccezionale: i motori di ricerca che, semplicemente specificando una o più parole da cercare, in pochi secondi riportano l'elenco di numerosi siti inerenti l'argomento di nostro interesse; tra i più famosi motori di ricerca vi ricordo, ammesso che ce ne sia bisogno, **GOOGLE** accessibile da www.google.it, **Yahoo** da www.yahoo.com e numerosi altri li troverete in rete;
- **E-Mail (Electronic Mail o posta elettronica):** inizialmente la maggior parte dei pacchetti scambiati in Internet era costituito da E-Mail; il servizio E-Mail è molto simile a quello tradizionale postale con qualche “piccolo” vantaggio:
 - Il messaggio arriva in tempo reale;
 - E' possibile allegare file o intere cartelle, che si trasferiscono nel PC del destinatario;

- E' possibile gestire elenchi di destinatari ed inviare una E-Mail a centinaia o migliaia e anche più destinatari con pochi clic del mouse;
- E' possibile avere immediatamente una conferma dell'avvenuta consegna del messaggio;
- E' possibile dare risposte automatiche alle mail in arrivo senza effettuare alcuna azione; per esempio, è possibile rispondere alle mail che arrivano su una casella obsoleta di posta elettronica con un messaggio che avvisi il mittente di utilizzare il nuovo indirizzo oppure che si è assenti dall'ufficio e che si risponderà alla mail appena possibile;
- Numerosissimi altri vantaggi li scoprirete utilizzando un programma di posta elettronica, come Outlook o Outlook Express;

Non si pensi che non ci siano degli svantaggi, quali, ad esempio, la continua violazione della **privacy**, dovuta alla ricezione di messaggi di posta elettronica indesiderati;

- **FTP – File Transfer Protocol:** ho già descritto questo servizio poco fa a proposito del protocollo FTP;
- **News:** vedi protocollo NNTP;
- **CHAT:** c'è forse bisogno di dare spiegazioni? Conoscete benissimo Microsoft Live Messenger e Skype;
- **E-Commerce:** commercio elettronico; famosissimo www.ebay.it. Ma sempre più aziende utilizzano tale strumento per la vendita diretta attraverso Internet; l'ordine viene inviato e ricevuto attraverso la rete e la merce arriva comodamente a casa dopo qualche giorno, con pagamento contrassegno o anticipato a mezzo bonifico bancario, ovviamente on line, o carta di credito;
- **E-Learning:** apprendimento elettronico. Si tratta di corsi di formazione fruibili attraverso la rete, quindi 24 ore su 24; in genere viene messo a disposizione del discente un tutor, contattabile ma è ovvio, a mezzo E-Mail;
- **Home Banking:** offre la possibilità di usufruire dei servizi bancari, quali la visione dell'estratto conto, il trasferimento di denaro, ecc.

Capitolo 13. Sistemi informativi

Sistemi informativi

(Fonte Internet e cervello del prof)

Si dice Sistema informativo l'insieme di tutti i dati di un'azienda, un ente pubblico o una qualsiasi organizzazione e le procedure per il loro trattamento. In altre parole si tratta del sistema **automatizzato o meno** che consente il trattamento dell'informazione ovvero, come detto qualche anno fa, che consente di acquisire, archiviare, elaborare e trasmettere l'informazione. Ho detto automatizzato o meno, infatti è possibile che piccole organizzazioni abbiano un trattamento manuale dei dati ovvero senza l'ausilio di un sistema di elaborazione; altri sistemi informativi potrebbero essere parzialmente automatizzati nel senso che coesistono procedure informatiche e procedure manuali.

In genere i sistemi informativi sono automatizzati; si pensi alla grande diffusione dei personal computer ed il loro basso costo, che consente l'automazione anche a piccoli commercianti, professionisti e persino noi con il nostro personal computer gestiamo un micro sistema informativo.

In genere in un sistema informativo ritroviamo:

1. **I dati**;
2. **Operazioni, procedure o attività** che si devono svolgere per la gestione delle informazioni,
3. **Modalità organizzative** con cui devono essere condotte tali attività,
4. **Strumenti tecnologici** con cui svolgerle;
5. **Le persone.**

In un ambiente sempre più dinamico come quello odierno, le imprese si trovano in una situazione di grande complessità gestionale e nell'esigenza di dover gestire quantità sempre maggiori di informazioni in modo sempre più efficace, efficiente e tempestivo per poter così rispondere ai continui cambiamenti del mercato e delle sue esigenze: prendere decisioni velocemente richiede la possibilità di disporre di tutte le informazioni necessarie in tempi rapidi, il che è possibile solo se l'impresa è dotata di un sistema informativo in grado di rendere disponibili le informazioni in tempo reale.

Le fasi di realizzazione di un sistema informativo sono generalmente le seguenti:

- **nascita dell'esigenza:** tende a migliorare la situazione esistente tramite il perseguimento di determinati obiettivi.

- **studio di fattibilità:** serve a definire in maniera per quanto possibile precisa i costi delle varie alternative possibili, ad effettuare la conseguente analisi costi/benefici ed a stabilire le priorità della realizzazione delle varie componenti del sistema.
- **raccolta e analisi dei requisiti:** consiste nell'individuazione e nello studio delle proprietà e delle funzionalità che il sistema informativo dovrà avere. Questa fase richiede un'interazione con gli utenti del sistema e produce una descrizione completa ma generalmente informale dei dati coinvolti e delle operazioni su di essi. Vengono inoltre stabiliti i requisiti software e hardware del sistema informativo.
- **progettazione:** si divide generalmente in progettazione dei **dati**, progettazione delle **applicazioni** e progettazione **dell'architettura tecnica di sistema**. Nella prima si individua la struttura e l'organizzazione che i dati dovranno avere, nella seconda si definiscono le caratteristiche dei progetti applicativi. Queste due attività sono complementari e possono procedere in parallelo o in cascata. Le descrizioni dei dati e delle applicazioni prodotte in questa fase sono formali e fanno riferimento a specifici modelli. La progettazione dell'architettura tecnica di sistema, infine, rappresenterà l'infrastruttura individuandone le caratteristiche in termini di sistemi (server), connettività, sicurezza fisica e logica.
- **sviluppo:** consiste nella realizzazione del sistema informativo secondo la struttura e le caratteristiche definite nella fase di progettazione. Viene costruita e popolata la base di dati e viene prodotto il codice dei programmi.
- **validazione e collaudo:** serve a verificare il corretto funzionamento e la qualità del sistema informativo. La sperimentazione deve prevedere, per quanto possibile, tutte le condizioni operative.
- **avviamento:** è la fase di messa in funzione del sistema. Si erogano i corsi di formazione, si travasano i dati da eventuali realizzazioni preesistenti che vengono sostituite, si attivano i collegamenti con le altre applicazioni e si parte con l'operatività reale.
- **funzionamento:** in questa fase il sistema informativo diventa operativo a regime ed esegue i compiti per i quali era stato originariamente progettato. Se non si verificano malfunzionamenti o revisioni delle funzionalità del sistema, questa attività richiede solo operazioni di gestione e manutenzione.
- **manutenzione:** con la manutenzione **correttiva** si consolida il sistema, mentre con la manutenzione **evolutiva** lo si completa ed arricchisce di funzionalità inizialmente non individuate.

Va precisato che il processo non è quasi mai strettamente sequenziale, in quanto spesso durante l'esecuzione di una attività citata bisogna rivedere decisioni prese nell'attività precedente. Quello che si ottiene è un ciclo di operazioni. Inoltre alle attività citate si aggiunge quella di **prototipizzazione**, che consiste nell'uso di specifici strumenti software per la realizzazione rapida di una versione semplificata del sistema informativo, con la quale sperimentare le sue funzionalità. La verifica del

prototipo può portare a una modifica dei requisiti e una eventuale revisione del progetto.

La parte del sistema informativo composta dai **calcolatori**, dalle **reti informatiche**, dalle procedure per la **memorizzazione** e la **trasmissione** elettronica delle informazioni prende il nome di **sistema informatico**. Infatti, i concetti fondamentali alla base di un sistema informativo sono dati, informazioni e processi, e non presuppongono l'utilizzo di tecnologie informatiche. Ciononostante, anche se l'esistenza del sistema informativo è indipendente dalla sua automazione, il relativo sistema informatico ricorre quasi sempre all'utilizzo di uno o più database per l'archiviazione e il reperimento delle informazioni, e ad appositi moduli software per l'inserimento e la gestione.

I primi sistemi informativi si basavano su un grosso centro di elaborazione dati detto **CED (Centro Elaborazione Dati)** e l'elaborazione delle informazioni era riferita con il termine **EDP (Electronic Data Processing)**. Oggi la situazione è cambiata molto e si tende al **decentramento** ovvero avere una distribuzione del sistema informativo in luoghi diversi ed a volte molto lontani, si pensi per esempio, alle banche. Tutto questo è stato reso possibile grazie alla **Telematica**.

Per quanto riguarda **l'hardware** abbiamo già detto abbastanza nel corso degli anni precedenti e per il **software** molto è stato detto nel corso di quest'anno. Riferendomi a quest'ultima componente del sistema informatico, ritengo opportuno sottolineare l'importanza, innanzitutto, del **DBMS**, quindi del **sistema operativo**.

E' necessario aprire una parentesi per alcuni pacchetti software che hanno assunto un importante ruolo nell'ambito dei sistemi informativi:

- **Pacchetti per ufficio** che migliorano la produttività personale; si tratta di una suite di software che coprono una fetta importante del lavoro d'ufficio costituito da scrittura di documenti, elaborazione di fogli di calcolo, presentazioni multimediali, ecc; molto noto il pacchetto **Office di Microsoft** e **Open Office di SUN**, gratuito ed addirittura sviluppato in open source; l'obiettivo che ci si propone di raggiungere è quello di avere **uffici senza carta** ovvero **paperless**;
- **Software CAD/CAM: Computer Aided Design**, cioè Progettazione Assistita da Elaboratore; CAD indica il settore dell'informatica volto all'utilizzo di tecnologie software e in particolare della computer grafica per supportare l'attività di progettazione (*design*) di manufatti sia virtuale che reali. I sistemi di *Computer Aided Design* hanno come obiettivo la creazione di modelli, soprattutto 3D, del manufatto. L'espressione **CAD/CAM** si riferisce all'impiego congiunto e integrato di sistemi software per la progettazione assistita da computer (*Computer-Aided Design*, CAD) e fabbricazione assistita dal computer (*Computer-Aided Manufacturing*, CAM). L'uso di sistemi integrati di CAD/CAM rende più semplice il trasferimento di informazioni dalla prima alla seconda fase del processo.
- **CAI: Computer Aided Instruction** è una forma di apprendimento mediante computer che presenta due importanti caratteristiche: apprendimento bidirezionale e individuale. Non è definibile come metodo in quanto il materiale CAI è uno

strumento che promuove l'auto-apprendimento. Risulta quindi essere un aiuto per gli insegnanti nel facilitare tale processo. L'utente utilizza corsi autodidattici, relativi alle più svariate materie, utilizzando un computer munito di uno specifico software detto **CBT (Computer Based Training)**.

L'e-learning sfrutta le potenzialità rese disponibili da Internet per fornire formazione **sincrona** e/o **asincrona** agli utenti, che possono accedere ai contenuti dei corsi in qualsiasi momento e in ogni luogo in cui esista una connessione online. La componente Internet e/o web e la presenza di una tecnologia specifica come **Web Based Training (WBT)** lo distingue da altre versioni di formazione a distanza.

Una componente importante di sistema informativo è costituita dalle **persone**; è evidente che, in un sistema informativo complesso, ritroviamo numerose **figure professionali** che concorrono al suo buon funzionamento:

- **Utente:** è l'utilizzatore delle procedure informatiche;
- **Operatore:** cura il buon funzionamento delle unità periferiche verificando anche la disponibilità dei materiali di consumo, quale carta o toner per le stampanti e così via;
- **Programmatore:** Realizza programmi sulla base delle specifiche dell'analista e con il rispetto di standard condivisi, cura la manutenzione e l'aggiornamento dei programmi per nuove esigenze, realizza la documentazione tecnica dei programmi, predispone istruzioni operative per gli utenti e, ove applicabile, ne cura l'addestramento;
- **Analista-Programmatore:** Figura intermedia;
- **Analista:** Collabora con il Capoprogetto alla definizione degli obiettivi tecnici, sviluppa l'analisi delle procedure studiando con l'utente le caratteristiche richieste, progetta la struttura della base dati necessaria, sviluppa l'analisi di dettaglio e le specifiche per il programmatore, contribuisce al collaudo e all'avviamento delle nuove procedure in stretta collaborazione con l'utente finale;
- **Capoprogetto o project manager:** stende il piano di sviluppo del progetto, assegnando risorse e definendo, insieme all'utente finale, i tempi di realizzazione, mantiene una visione costantemente aggiornata dello stato avanzamento del progetto, studia e risolve ritardi, incidenti di percorso, assenze, malfunzionamenti;
- **DBA-Data Base Administrator:** Cura, in collaborazione con il team di sviluppo, la progettazione fisica del Data Base e delle funzioni dirette di gestione, segue l'evoluzione del Data Base in termini di volumi, accessi, tempi di risposta, malfunzionamenti, organizza ed esegue le operazioni di manutenzione ordinaria e straordinaria e le procedure di sicurezza, come i backup ordinari e straordinari e gli eventuali ripristini a seguito di malfunzionamenti;
- **Sistemista:** Assicura il monitoraggio dell'evoluzione tecnologica dei sistemi, la loro ottimizzazione ed il costante aggiornamento, fornisce assistenza ai team di sviluppo sull'utilizzo ottimale del software di base;
- **System integrator:** Cura l'armonizzazione di funzionamento e di servizio di sottosistemi di diversa natura e/o di diversi fornitori; la natura delle integrazioni è

la più varia: va dal software applicativo, ai data base, alla gestione delle reti; riassume in sé capacità ed esperienze tipiche di analisti, programmatori, sistemisti ed amministratori di base dati;

- **Network Administrator:** Amministratore della rete;
- **Responsabile della sicurezza;**
- **Web designer:** si occupa della progettazione tecnica, strutturale e grafica di un sito web;
- **Webmaster o webengineer:** progetta, costruisce ed è responsabile di un sito web.

Sistemi informativi molto importanti sono stati realizzati negli ultimi anni per il commercio elettronico che si distingue in:

- **B2B – Business-to-Business:** scambio di transazioni tra imprese;
- **B2C – Business-to-Consumer:** tra imprese e consumatori, negozi virtuali;
- **B2A – Business-to-administration e B2G--Business to Government:** transazioni tra imprese e pubblica amministrazione;
- **C2A – Consumer-to-Administration e C2G – Consumer-to-Government:** tra cittadini e pubblica amministrazione;
- **B2E – Business-to-Employees:** Impresa e propri impiegati.

Concludo con sistemi informativi ai quali si applica **l'Intelligenza Artificiale**; per intelligenza artificiale si intende lo studio che mira a realizzare macchine con comportamenti simili all'uomo; fino ad ora si può dire che si è ancora lontani dal realizzare una macchina pensante; noi sappiamo bene che un sistema di elaborazione non fa altro se non eseguire codice di un programma scritto dall'uomo, anche se ad una velocità enorme; le aree di applicazione dell'intelligenza artificiale sono essenzialmente due:

- **La robotica:** applicazione dell'IT alla produzione; stiamo parlando delle catene di montaggio;
- **I Sistemi Esperti:** Si tratta di software in grado di modificare il proprio comportamento nel corso della sua elaborazione; un esempio su tutti: un software in grado di giocare a scacchi e migliorare il proprio livello durante il gioco.

Capitolo 14.

Informatica e diritto, sicurezza

Fonte: Internet.

14.1. Diritto d'autore

La nascita del diritto d'autore può farsi risalire all'invenzione della stampa ed in particolare alla possibilità di realizzare con essa un notevole numero di copie delle opere in poco tempo.

Il **diritto d'autore** è un istituto che intende attribuire a colui che abbia realizzato un'opera dell'ingegno a carattere creativo un fascio di facoltà, dirette soprattutto a riservare all'autore qualsiasi attività di utilizzazione economica dell'opera. In [Italia](#) è disciplinato dalla [legge 22 aprile 1941, n. 633](#). Al momento della sua emanazione, la legge era sostanzialmente conforme alla tutela minima prevista dalla [Convenzione di Berna](#). Nel corso del tempo le sue disposizioni sono state modificate in numerose occasioni, nel recepimento, tra l'altro, di diverse [disposizioni comunitarie](#), oltre che nell'adeguamento al dettato della successiva [Costituzione repubblicana](#).

Gli [artt. 1-5](#) forniscono gli elementi per individuare le opere protette dal diritto d'autore. Nella tutela rientrano tutte le opere dell'ingegno avente carattere creativo, qualunque ne sia il modo o la forma di espressione. A titolo esemplificativo, la legge fornisce un elenco di categorie in cui siano ricomprese le opere riconducibili:

- alla [letteratura](#): opere letterarie, drammatiche, scientifiche, didattiche e religiose (compresi i programmi per elaboratore e le banche dati), sia in forma scritta che orale
- alla [musica](#): opere e composizioni musicali, con o senza parole, opere drammatico-musicali e variazioni musicali purché costituiscano un'opera originale in sé
- alle arti figurative: opere [di scultura, pittura, disegni](#), incisioni o appartenenti ad arti figurative similari, compresa la [scenografia](#)
- [all'architettura](#): i disegni e le opere dell'architettura, le opere del disegno industriale che presentino carattere creativo e valore artistico
- al [teatro: opere coreografiche e pantomimiche](#) (con o senza traccia scritta)
- [alla cinematografia: opere cinematografiche](#), mute o con sonoro, [fotografiche](#)

Inoltre sono protette anche le cosiddette "elaborazioni di carattere creativo", come ad esempio le traduzioni in un'altra lingua, le trasformazioni da una forma letteraria o artistica in un'altra, gli adattamenti, le riduzioni, ecc.

A seguito del recepimento delle direttive [96/9/CE](#) e [91/250/EEC](#) inoltre, sono ora ricompresi nell'elenco:

- i [programmi per elaboratore](#)
- le [banche di dati](#)

Con riferimento ai software si parla di:

- Il diritto d'autore ([copyright](#)).

Il copyright (termine di lingua inglese che letteralmente significa diritto di copia) è l'insieme delle normative sul diritto d'autore in vigore nel mondo anglosassone e statunitense.

Col tempo, ha assunto in Italia un significato sempre più prossimo ad indicare le "norme sul diritto d'autore vigenti in Italia", da cui in realtà il copyright differisce sotto vari aspetti.

È solitamente abbreviato con il simbolo ©. Quando tale simbolo non è utilizzabile si riproduce con la lettera "c" posta tra parentesi: (c) o (C).

- Il permesso d'autore ([copyleft](#))

L'espressione inglese **copyleft**, lett. "(diritto di) copiatura (esclusiva) abbandonato", è gioco di parole su [copyright](#) (lett. "diritto (esclusivo) di copiatura"), in cui la seconda parola del composto, "right" è scambiata con "left" individua un modello alternativo di gestione dei diritti d'autore basato su un sistema di [licenze](#) attraverso le quali l'autore (in quanto detentore originario dei diritti sull'opera) indica ai fruitori dell'opera che essa può essere utilizzata, diffusa e spesso anche modificata liberamente, pur nel rispetto di alcune condizioni essenziali. Nella versione pura e originaria del copyleft (cioè quella riferita all'ambito informatico) la condizione principale obbliga i fruitori dell'opera, nel caso vogliano distribuire l'opera modificata, a farlo sotto lo stesso regime giuridico (e generalmente sotto la stessa licenza). In questo modo, il regime di copyleft e tutto l'insieme di libertà da esso derivanti sono sempre garantiti.

I software, in genere, non possono essere copiati se non a scopo precauzionale; in altre parole l'acquirente ha il diritto di effettuare una sola copia a solo scopo di backup.

L'utilizzo di un software è quasi sempre regolato dal contratto con l'utente finale detto **EULA (End User License Agreement)**; in genere qualsiasi software è accompagnato da un EULA che ne stabilisce i diritti di utilizzo.

Rispetto ai diritti dell'autore il software può essere classificato in:

- **Software a pagamento** o software proprietario per i quali è necessario pagare un corrispettivo per ottenerne l'EULA; si tratta di software coperto da copyright. Software che ha restrizioni sul suo utilizzo, sulla sua modifica, riproduzione o redistribuzione, solitamente imposti da un proprietario. Queste restrizioni vengono ottenute tramite mezzi tecnici o legali.

Si rende pubblico solo il [codice binario](#) del software, trattenendone il [codice sorgente](#). In questi casi la modifica del software risulta molto difficile, ottenibile solo grazie a [disassemblatori](#) e ad elevate capacità informatiche.

- Il **software libero** (*free software*)

Il software libero è software rilasciato con una licenza che permette a chiunque di utilizzarlo e che ne incoraggia lo studio, le modifiche e la redistribuzione; per le sue caratteristiche, si contrappone al software proprietario ed è differente dalla concezione open source, incentrandosi sulla libertà dell'utente e non solo sull'apertura del codice sorgente.

- **Il software di pubblico dominio**

Si tratta di software non sottoposto ad alcun tipo di licenza; fanno parte di questa categoria, per esempio, le implementazioni del protocollo TCP/IP.

- **Il software Open Source**

In informatica, open source (termine inglese che significa sorgente aperto) indica un software i cui autori (più precisamente i detentori dei diritti) ne permettono, anzi ne favoriscono il libero studio e l'apporto di modifiche da parte di altri programmatori indipendenti. Questo è realizzato mediante l'applicazione di apposite licenze d'uso.

La collaborazione di più parti (in genere libera e spontanea) permette al prodotto finale di raggiungere una complessità maggiore di quanto potrebbe ottenere un singolo gruppo di lavoro. L'open source ha tratto grande beneficio

da Internet, perché esso permette a programmatori geograficamente distanti di coordinarsi e lavorare allo stesso progetto.

I software open source attualmente più diffusi sono Firefox, OpenOffice, VLC, Gimp, 7-Zip, oltre ad un gran numero di progetti rivolti non all'utente finale ma ad altri programmatori.[1] Sono inoltre degne di nota le famiglie di sistemi operativi BSD, GNU e il kernel Linux, i cui autori e fautori hanno contribuito in modo fondamentale alla nascita del movimento. La comunità open source è molto attiva, comprende decine di migliaia di progetti, numero che cresce quotidianamente.

Alla filosofia del movimento open source si ispira il movimento **open content (contenuti aperti)**: in questo caso ad essere liberamente disponibile non è il codice sorgente di un software ma contenuti editoriali quali testi, immagini, video e musica. Wikipedia è un chiaro esempio dei frutti di questo movimento. Attualmente l'open source tende ad assumere rilievo filosofico, consistendo di una nuova concezione della vita, aperta e refrattaria ad ogni oscurantismo, che l'open source si propone di superare mediante la condivisione della conoscenza.

Diritti di utilizzazione economica

Come si legge all'art.25: i diritti di utilizzazione economica dell'opera durano tutta la vita dell'autore e **sino al termine del settantesimo anno solare dopo la sua morte**. Nel caso in cui l'opera sia frutto del lavoro di più coautori, si considera come termine sulla vita il coautore che muore per ultimo. Nelle opere collettive la durata dei diritti di utilizzazione dell'opera come un tutt'uno è di settant'anni dalla prima pubblicazione.

Per le opere anonime o pseudonime devono trascorrere settant'anni dalla prima pubblicazione (qualunque sia la forma in cui viene effettuata), se l'autore si rivela o viene rivelato da persone autorizzate, l'opera torna a sottostare alle normali leggi. In caso di parti di opera, di volumi e/o di opere periodiche, la durata dei diritti decorre dall'anno della pubblicazione.

Per le opere pubblicate da Amministrazioni dello Stato, fra le quali sono comprese accademie, ed enti pubblici culturali, ed alle quali sono assimilati gli enti privati senza fini di lucro, va notato che il diritto decade dopo venti anni.

Cosa è possibile utilizzare liberamente?

Esistono alcune opere che possono essere, sotto determinate condizioni, liberamente utilizzate. Ecco alcuni esempi (per un elenco completo si vedano gli [artt. 65-71 quinquies](#) della legge n. 633/41 che regola il diritto d'autore):

- articoli di attualità, economici o politico religiosi, pubblicati in riviste o giornali possono essere riprodotti su altre riviste o giornali purché la riproduzione non sia stata espressamente riservata e vengano indicati
 - nome della rivista/giornale
 - data e numero della rivista/giornale
 - nome dell'autore (se l'articolo è firmato)
- discorsi tenuti in pubblico, purché si indichi
 - la fonte
 - il nome dell'oratore
 - la data e il luogo in cui è stato tenuto il discorso
- il riassunto, la citazione, la riproduzione di brani o parti di opera per scopi di critica, discussione o insegnamento purché non costituiscano concorrenza all'utilizzazione economica dell'opera e vengano menzionati
 - titolo dell'opera
 - autore
 - editore
 - eventuale traduttore

Estinzione del diritto economico

Il diritto di utilizzo economico si estingue, nella maggior parte degli ordinamenti occidentali, decorso un certo periodo dalla morte dell'autore; pertanto agli eredi è in genere garantito un periodo di tutela di questo diritto che solitamente copre un tempo equivalente ad una o due generazioni. Attualmente tale tutela nella maggior parte dei paesi occidentali (tra cui l'Italia) è di settant'anni dalla morte dell'ultimo dei coautori dell'opera. Il diritto morale non si estingue mai, sempre restando da riferirsi all'autore, in qualunque tempo, la titolarità creativa dell'opera.

Estinto il diritto d'autore, l'opera diviene di pubblico dominio ed è liberamente utilizzabile da chiunque, anche a fini economici, purché sia rispettato il diritto morale alla titolarità artistica.

Diritti connessi all'esercizio del diritto d'autore: diritti relativi al ritratto

Il ritratto di una persona non può essere esposto, riprodotto o messo in commercio senza il consenso del soggetto tranne quando la riproduzione dell'immagine è giustificata dalla notorietà o dall'ufficio pubblico coperto, da necessità di giustizia o polizia, da scopi scientifici, didattici o culturali, o quando la riproduzione è collegata a fatti avvenimenti, cerimonie di interesse pubblico o svoltisi in pubblico. Il ritratto non può essere messo in commercio o esposto se pregiudica l'onore, la reputazione o comunque il decoro della persona ritratta.

Estinto il diritto d'autore, l'opera diviene di pubblico dominio ed è liberamente utilizzabile da chiunque, anche a fini economici, purché sia rispettato il diritto morale alla titolarità artistica.

La normativa italiana consente la riproduzione analogica ad uso privato di file musicali registrati (non scaricati) da radio e TV che trasmettono in streaming.

Vari controlli contro le violazioni del diritto d'autore vengono svolti con programmi di sniffing che accedono nelle reti peer-to-peer e registrano gli indirizzi IP e i provider con il quale sono connessi quanti stanno scambiando illegalmente file. Tali informazioni sono contenute nell'intestazione di ogni pacchetto TCP/IP inviato su Internet da un generico nodo della rete.

Per la normativa italiana, ogni Internet Service provider deve tenere un registro che abbina il numero telefonico del chiamante all'IP assegnato a chi chiede la connessione. Tramite questo log è possibile risalire al numero e identificare il chiamante per procedere a una denuncia.

Diritto d'autore - Sentenza

Clamorosa decisione della Terza sezione penale della Corte di Cassazione: annullata la condanna a tre mesi e 10 giorni di reclusione a due ragazzi che avevano scaricato musica, film e software. Una decisione che farà parlare molto. Intanto però esultano i sostenitori del peer-to-peer: [Scaricare](#) film, musica o programmi da Internet non è reato se questo non implica anche un guadagno economico. Anche se il materiale scaricato è protetto dal diritto d'autore. Così ha deciso la Terza sezione penale della Corte di Cassazione che ha annullato la condanna (3 mesi e 10 giorni) inflitta dalla Corte d'Appello di Torino a due ragazzi che avevano scaricato e condiviso in rete [File](#) protetti da copyright.

Tramite il pc di un'associazione studentesca del Politecnico di Torino i due avevano messo in piedi una rete p2p con il sistema più semplice e più vecchio del mondo: un collegamento via [Ftp](#) e chiavi di accesso al [Server](#) distribuite soltanto a chi decideva di condividere il proprio archivio di musica, film, software o videogiochi. Come accadeva tanti anni fa con le [Bbs](#).

Molto del materiale condiviso era protetto dal diritto d'autore. E secondo i giudici piemontesi i due ragazzi, con questo sistema di scambio file alla pari, avevano violato gli articoli 171 bis e 171 ter della legge sul diritto d'autore (n. 633/41) che punisce chi "a scopo di lucro diffonde o duplica file e contenuti multimediali protetti da copyright". Eppure, come spiega la sentenza n. 149 della Suprema Corte (depositata il 9 gennaio), l'attività degli imputati non aveva alcun fine di lucro. Dunque nessuna violazione della legge 633/41. "I giudici di merito - si legge nelle motivazioni della sentenza - hanno erroneamente attribuito all'imputato un'attività di duplicazione dei programmi e di opere dell'ingegno protette dal diritto d'autore, poiché la duplicazione in effetti avveniva ad opera dei soggetti che si collegavano al sito Ftp e da esso, in piena [Autonomia](#), prelevavano i file e nello stesso ne scaricavano altri".

A questo la Corte di Cassazione aggiunge che in relazione al sequestro in casa di uno degli imputati

di un software per generare numeri di codice [Seriale](#) per registrare illegalmente alcuni software "doveva escludersi ogni fine commerciale". Per questo motivo i giudici della Suprema Corte, rilevando che "le operazioni di [Download](#) sul server Ftp di materiale informatico non coincide con le ipotesi criminose fatte dai giudici torinesi", e che per scopo di lucro deve intendersi "un fine di guadagno economicamente apprezzabile o di incremento patrimoniale da parte dell'autore del fatto, e che può identificarsi con un vantaggio di altro genere", ha annullato la condanna senza rinvio. Cioè i due ragazzi sono stati prosciolti in via definitiva.

14.2. Criminalità informatica

Per "criminalità informatica" si intendono tutti quei reati che vengono perpetrati a mezzo del web. Essi si distinguono sostanzialmente in due categorie: crimini che abbisognano necessariamente della rete per essere commessi (es. abuso informatico, frode telematica, falso informatico, etc.) e quelli che invece vengono realizzati anche senza il tramite della grande rete (es. sfruttamento sessuale in danno dei minori, pedofilia, diffamazione, eversione, etc.).

L'ordinamento comunitario è stato arricchito di alcune decisioni riguardanti questo genere di crimini:

- Decisione del Consiglio 29 maggio 2000, relativa alla lotta contro la pornografia infantile su Internet;
- Decisione 276/1999/CE del Parlamento Europeo del Consiglio del 25 gennaio 1999, che adotta un piano pluriennale d'azione comunitario per promuovere l'uso sicuro di internet attraverso la lotta alle informazioni di contenuto illegale e nocivo diffuso attraverso le reti globali;
- Comunicazione CE 16 ottobre 1996, n. 487, relativo alle informazioni di contenuto illegale e nocivo su Internet.

Per quanto concerne invece il nostro ordinamento interno, vi sono state novità non soltanto legislative.

In primis, le forze dell'ordine sono state dotate di strumenti tecnologici e professionali in grado di arginare fenomeni criminali informatici. Inizialmente nel 1992, presso la Polizia di Stato è stato istituito un gruppo di esperti in materia di "High Tech Crime" e nel 1998 è nato il **Servizio Polizia Postale e delle Comunicazioni**, dotata di un'organizzazione capillare e flessibile presente in tutto il territorio nazionale attraverso strutture centrali e periferiche dislocate nei capoluoghi di regione e provincia.

Esso è specializzato nell'attività di repressione e prevenzione dei reati telematici e partecipa con propri rappresentanti ai lavori di consessi internazionali in materia di high tech crime (G8, UE, OCSE, EUROPOL, INTERPOL, Consiglio d'Europa), collaborando inoltre con omologhi uffici di polizia di altri Paesi (1).

A livello interno, tale servizio gode del supporto offerto ad altre istituzioni quali ad esempio il Ministero delle Comunicazioni e l'Autorità per le Garanzie nelle Comunicazioni.

Altra innovazione importante riguarda l'aggiornamento della normativa penale e l'introduzione di alcune leggi formulate ad hoc.

E' necessario tenere presenti alcuni problemi precipui della realtà virtuale in rete: alcuni ambienti

(ad esempio, le chat rooms) legittimano gli utenti a creare individui fittizi che, operando anonimamente, possono persino garantire alle organizzazioni criminali e terroristiche di perseguire i propri scopi indisturbatamente; inoltre, appare difficile l'individuazione del c.d. "locus commissi delicti" (data la transnazionalità ed aterritorialità della rete), con correlative problematiche di diritto penale internazionale.

I reati più comuni riguardano gli accessi abusivi, la diffusione di virus informatici, il defacement, il denial of service, il netstrike e le frodi perpetrate da operatori TLC.

Da alcuni rilevamenti eseguiti dal Computer Security Institute risulta che i crimini informatici di natura economica sono riconducibili nella maggior parte dei casi ai white collar crime (4) (crimine dal colletto bianco) e non alle attività degli hackers.

Inoltre, dai risultati di recenti indagini, pare che molti degli attacchi subiti dalle aziende siano causati dall'effetto di virus informatici contratti in seguito a comportamenti negligenti da parte dei dipendenti nell'uso delle postazioni internet, nel corso di connessioni a siti di natura pornografica, con punte massime nelle ore d'ufficio.

Altro problema, manifestatosi in sede investigativa ed emerso in seguito a complesse indagini, riguarda la cd. cifra nera dei reati non denunciati dalle aziende colpite. Ne consegue che, in assenza di denuncia, la Polizia non può intervenire ed i responsabili degli illeciti rimangono impuniti.

Note

1) Il 19 marzo 1998 il Consiglio europeo ha proposto di istituire una rete atta allo scambio di informazioni per combattere la criminalità ad alta tecnologia nell'ambito del G8. La rete è stata gradualmente realizzata ed al momento ne fanno parte Australia, Brasile, Canada, Danimarca, Finlandia, Francia, Germania, Italia, Giappone, Paesi Bassi, Russia, Spagna, Svezia, Regno Unito e Stati Uniti d'America. Inoltre, il 23 novembre 2001 ventinove stati hanno sottoscritto a Budapest la Convenzione sul Cybercrime.

2) Art. 392 c.p.: "si ha altresì violenza sulle cose allorché un programma informatico viene alterato, modificato o cancellato in tutto o in parte ovvero viene impedito o turbato il funzionamento di un sistema informatico o telematico".

3) La frode informatica è un'evoluzione della frode intesa in senso tradizionale: gli "artifici e raggiri", caratterizzanti la truffa, prevista all'art. 640 del c.p. si estendono alla versione tecnologica, comprendente, ad esempio, la manipolazione di dati.

4) Per "colletti bianchi" si intendono soggetti dotati di capacità informatiche comuni (al contrario degli hackers) che utilizzano le proprie conoscenze o la propria posizione in azienda per porre in essere condotte delittuose dalle quali trarre un ingiusto profitto con l'altrui danno.

14.3. Sicurezza informatica

La **Sicurezza informatica** è quella branca dell'informatica che si occupa della salvaguardia dei sistemi informatici da potenziali rischi di perdita e/o violazioni dei dati. I principali aspetti di protezione del dato sono la confidenzialità, l'integrità e la disponibilità. La protezione dagli attacchi informatici viene ottenuta agendo su più livelli: innanzitutto a livello fisico e materiale, ponendo i server in luoghi il più possibile sicuri, dotati di sorveglianza e/o di controllo degli accessi; anche se questo accorgimento fa parte della sicurezza normale e non della "sicurezza informatica" è sempre il caso di far notare come spesso il fatto di adottare le tecniche più sofisticate generi un falso senso di sicurezza che può portare a trascurare quelle semplici. Il secondo livello è normalmente quello logico che prevede l'autenticazione e l'autorizzazione di un'entità che rappresenta l'utente nel sistema; l'autenticazione avviene, in genere, attraverso l'operazione di login che consente di fornire al sistema le credenziali dell'utente, costituite da user name e password. Successivamente al processo di autenticazione, le operazioni effettuate dall'utente sono tracciate in file di log. Questo processo di monitoraggio delle attività è detto audit o accountability.

Tipi di sicurezza e di attacchi

Proprio sulla base di queste osservazioni, quando si parla di "sicurezza informatica" spesso si distinguono i concetti di *sicurezza passiva* e di *sicurezza attiva*.

Sicurezza passiva

Per *sicurezza passiva* normalmente si intendono le tecniche e gli strumenti di tipo *difensivo*, ossia quel complesso di soluzioni il cui obiettivo è quello di impedire che utenti non autorizzati possano accedere a risorse, sistemi, impianti, informazioni e dati di natura riservata.

Sicurezza attiva

Per *sicurezza attiva* si intendono, invece, le tecniche e gli strumenti mediante i quali le informazioni ed i dati di natura riservata sono resi intrinsecamente sicuri, proteggendo gli stessi sia dalla possibilità che un utente non autorizzato possa accedervi (confidenzialità), sia dalla possibilità che un utente non autorizzato possa modificarli (integrità). E' evidente che la sicurezza passiva e quella attiva sono tra loro complementari ed entrambe indispensabili per raggiungere il desiderato livello di sicurezza di un sistema. Le possibili tecniche di attacco sono molteplici, perciò è necessario usare contemporaneamente diverse tecniche difensive. Spesso l'obiettivo dell'attaccante non è rappresentato dai sistemi informatici in sé, quanto piuttosto dai dati in essi contenuti, quindi la sicurezza informatica deve preoccuparsi di impedire l'accesso ad utenti non autorizzati, ma anche a soggetti con autorizzazione limitata a certe operazioni, per evitare che i dati appartenenti al sistema informatico vengano copiati, modificati o cancellati. Le violazioni possono essere molteplici: vi possono essere tentativi non autorizzati di accesso a zone riservate, furto di identità digitale o di file riservati. La sicurezza informatica si occupa anche di prevenire i DoS. Questi sono attacchi sferrati al sistema con l'obiettivo di rendere non utilizzabili alcune risorse in modo da danneggiare gli utilizzatori del sistema.

Sicurezza nelle aziende

Dal momento che l'informazione è un bene che aggiunge valore all'impresa, e che ormai la maggior parte delle informazioni sono custodite su supporti informatici, ogni organizzazione deve essere in

grado di garantire la sicurezza dei propri dati. Per questo esistono, a carico delle imprese, precisi obblighi in materia di privacy, tra cui quello di redigere annualmente uno specifico **documento programmatico sulla sicurezza**. È stato anche approvato a livello internazionale il nuovo Standard ISO 27001:2005 finalizzato alla standardizzazione delle modalità adatte a proteggere i dati. Una fase indispensabile di ogni pianificazione della sicurezza è la valutazione del rischio e la gestione del rischio.

Sicurezza dei programmi

Il problema della sicurezza dei programmi e soprattutto dell'invio e ricezione di dati confidenziali protetti, si è posto come conseguenza della sensibile crescita dell'uso degli strumenti informatici e di internet. Per quanto riguarda la produzione di software "protetti" possiamo partire col definire il concetto di **sicurezza** come **l'assenza da condizioni conflittuali capaci di produrre danni mortali o irreparabili ad un sistema**.

Principali tecniche di attacco:

- **Hacking**: Studio di sistemi informatici al fine di potenziarne capacità e funzioni.
- **Exploit**: Metodo che, sfruttando una vulnerabilità, porta all'acquisizione di privilegi.
- **Shellcode**: E' un programma in linguaggio assembly che esegue una shell. Quest'ultima è un programma che permette di comunicare con il sistema e avviare i programmi.
- **Cracking**: E' la modifica di un software per rimuovere la protezione dalla copia, oppure per ottenere accesso ad un'area altrimenti riservata.
- **Backdoor**: Sono paragonabili a *porte di servizio* che consentono di superare in parte o in tutto le procedure di sicurezza attivate in un sistema informatico.
- **Sniffing**: Attività di intercettazione passiva dei dati che transitano in una rete telematica.
- **Spoofing**: L'arte della contraffazione dei pacchetti, intendendo per pacchetto una qualsiasi sequenza di dati distinta trasmessa su una rete.
- **Il phishing** è una frode informatica, realizzata con l'invio di e-mail contraffatte, finalizzata all'acquisizione, per scopi illegali, di dati riservati; sostituzione di identità.
- **Trojan**: E' un tipo di malware cioè, un qualsiasi software creato con il solo scopo di causare danni più o meno gravi al computer su cui viene eseguito.
- **Virus informatici**
- **DoS**: In questo tipo di attacco si cerca di portare il funzionamento di un sistema informatico che fornisce un servizio, ad esempio un sito web, al limite delle prestazioni, lavorando su uno dei parametri d'ingresso, fino a renderlo non più in grado di erogare il servizio.

Principali tecniche di difesa:

- **Antivirus**: consente di proteggere il proprio personal computer da software dannosi conosciuti come virus. Un buon antivirus deve essere costantemente aggiornato ad avere in continua esecuzione le funzioni di scansione in tempo reale. Per un miglior utilizzo l'utente deve avviare con regolarità la scansione dei dispositivi del PC (dischi fissi, CD, DVD e dischetti floppy), per verificare la presenza di virus, worm. Per evitare la diffusione di virus è inoltre utile controllare tutti i file che si ricevono o che vengono spediti tramite posta elettronica facendoli verificare dall'antivirus correttamente configurato a tale scopo.
- **Anti-Spyware**: software facilmente reperibile sul web in versione freeware, shareware o a pagamento. E' diventato un utilissimo tool per la rimozione di "file spia", gli spyware

- appunto, in grado di carpire informazioni riguardanti le attività on line dell'utente ed inviarle ad un'organizzazione che le utilizzerà per trarne profitto.
- **Firewall:** installato e ben configurato un buon personal firewall garantisce un sistema di controllo degli accessi verificando tutto il traffico che lo attraversa. Protegge contro aggressioni provenienti dall'esterno e blocca eventuali programmi presenti sul computer che tentano di accedere ad internet senza il controllo dell'utente.
 - **Firma digitale, crittografia:** è possibile proteggere documenti e dati sensibili da accessi non autorizzati utilizzando meccanismi di sicurezza specifici quali: la crittografia, la firma digitale, e l'utilizzo di certificati digitali e algoritmi crittografici per identificare l'autorità di certificazione, un sito, un soggetto o un software.
 - **Backup:** più che un sistema di difesa si tratta di un utile sistema per recuperare dati eventualmente persi o danneggiati. Il backup consiste nell'esecuzione di una copia di sicurezza dei dati di un personal computer o comunque di dati considerati importanti onde evitare che vadano perduti o illeggibili.
 - **Intrusion Detection System (IDS):** è un dispositivo software utilizzato per identificare accessi non autorizzati ai computer o alle reti locali.
 - **Network Intrusion Detection System (NIDS):** sono degli strumenti informatici, software o hardware, dediti ad analizzare il traffico di uno o più segmenti di una LAN al fine di individuare anomalie nei flussi o probabili intrusioni informatiche.
 - **Crittografia:**
 - **Sistema di autenticazione:** potrebbe rivelarsi utile, in particolare nelle aziende, l'utilizzo di software per l'autenticazione sicura con un secondo elemento di autenticazione basato su un insieme di caratteri disposti in uno schema suddiviso in file e colonne conosciute dall'utente che dovrà poi inserirle in una combinazione di valori per dimostrare di essere in possesso dei dati corretti. Altro sistema, più sofisticato, è quello del riconoscimento dell'utente tramite l'utilizzo dell'impronta digitale come forma di autenticazione.

14.4. Virus informatici

Cosa è un virus, dove si trova e come funziona

Un virus è composto da un insieme di istruzioni, come qualsiasi altro programma per computer. solitamente composto da un numero molto ridotto di istruzioni, (da pochi byte ad alcuni kilobyte), ed è specializzato per eseguire soltanto poche e semplici operazioni e ottimizzato per impiegare minor numero di risorse, in modo da rendersi il più possibile invisibile. Caratteristica principale un virus è quella di riprodursi e quindi diffondersi nel computer ogni volta che viene aperto il file infetto.

Un virus, per essere attivato, deve infettare un programma ospite oppure è costituito da un programma a sé stante che si inserisce, spesso, nei meccanismi di avvio del sistema o di altri software. La tecnica solitamente usata dai virus è quella di infettare i file eseguibili: il virus inserisce una copia di sé stesso nel file eseguibile che deve infettare, pone tra le prime istruzioni di tale eseguibile un'istruzione di salto alla prima linea della sua copia ed alla fine di essa mette un altro salto all'inizio dell'esecuzione del programma. In questo modo quando un utente lancia un programma infettato viene dapprima impercettibilmente eseguito il virus, e poi il programma. L'utente vede l'esecuzione del programma e non si accorge che il virus è ora in esecuzione in memoria e sta compiendo le varie operazioni contenute nel suo codice.

Si possono distinguere due fasi di un virus:

- quando è solo presente su un supporto di massa (disco fisso, floppy, CD, ...) il virus è inerte, anche se copiato sul proprio PC non è in grado di fare nulla fino a quando non viene eseguito il programma che lo ospita;
- quando è stato caricato in memoria RAM il virus diventa attivo ed inizia ad agire.

Principalmente un virus esegue copie di sé stesso spargendo l'epidemia, ma può avere anche altri compiti molto più dannosi (cancellare o rovinare dei file, formattare l'hard disk, far apparire messaggi, disegni o modificare l'aspetto del video, ...)

Componenti di un virus

virus informatici più semplici sono composti da due parti essenziali, sufficienti ad assicurarne la replicazione:

- **una routine di ricerca**, che si occupa di ricercare dei file adatti ad essere infettati dal virus controlla che gli stessi non ne contengano già una copia, in modo da evitare l'infezione ripetuta di uno stesso file;
- **una routine di infezione**, con il compito di copiare il codice del virus all'interno di ogni file selezionato dalla routine di ricerca in modo che venga eseguito ogni volta che il file infetto viene aperto, in maniera trasparente rispetto all'utente.
- **la routine di attivazione**, che contiene i criteri in base ai quali il virus decide se effettuare o meno l'attacco (es. una data, o il raggiungimento di un certo numero di file infetti);
- **una routine di mutazione**, che si occupa di modificare le routine di cifratura e decifratura per ogni nuova copia del virus.

Tipologie di virus

Alcuni virus vengono denominati in maniera particolare a seconda che possiedano a meno determinate caratteristiche:

Virus polimorfico

un virus, di solito, viene criptato lasciando in chiaro solo la routine di decriptazione. Un virus polimorfico modifica il codice della routine di decriptazione ad ogni nuova infezione (lasciando ovviamente invariato l'algoritmo) mediante tecniche di inserimento di codice spazzatura, permutazione del codice, etc...

Virus metamorfico

simile al *virus polimorfico*, è però in grado di mutare completamente il proprio codice, è più potente del virus polimorfico in quanto alcuni [software antivirus](#) possono riconoscere un virus dal codice anche durante l'esecuzione. Inoltre a volte impiega tecniche di mascheramento avanzate basate sulla divisione del proprio codice e successivo inserimento delle parti all'interno di diversi punti del file infetto (i virus convenzionali inseriscono il codice integralmente in fondo al file cambiando l'*entry point* per far eseguire per primo il codice maligno), lasciando inoltre invariato l'*entry point* per rendere ancora più difficile la vita agli antivirus. C'è da dire anche che la criptazione dei virus metamorfici non è necessaria.

Exe virus

virus che infettano i file eseguibili .EXE.

Com virus

virus che infettano i file di comando .COM (ormai rari).

Virus di boot

un tipo di virus ormai poco diffuso, che infetta il [boot sector](#) dei dischi (floppy disk o hard disk) invece che i singoli file.

Macrovirus

può essere contenuto generalmente in un documento di [Microsoft Word](#), [Microsoft Excel](#) o [Microsoft PowerPoint](#) e consiste in una [macro](#); può diffondersi a tutti i documenti che vengono aperti con quella particolare applicazione. Questo tipo di virus può infettare i sistemi operativi su cui gira ms-office (windows e mac), anche se non è detto che possano funzionare correttamente su altri sistemi operativi coinvolti.

Retrovirus

Virus che si annida nei programmi antivirus e li mette fuori uso. Il nome deriva dai [retrovirus](#) biologici, in grado di attaccare il [sistema immunitario](#) (come, ad esempio, l'[HIV](#)).

14.5. Dialer

Un **dialer** è un [programma](#) per [computer](#) di pochi [kilobyte](#) che crea una connessione ad [Internet](#), a un'altra [rete di calcolatori](#) o semplicemente a un altro [computer](#) tramite la comune [linea telefonica](#) o un collegamento [ISDN](#).

Dialer legittimi

Il mercato dei dialer italiano è il più fiorente e denso di insidie al mondo. L'[Italia](#), infatti, è il Paese nel quale a partire dai primi [anni novanta](#) fino a oggi le truffe per dialer hanno registrato il maggior numero di denunce, e dove le numerazioni "speciali" autorizzate (nonché gli utili dichiarati dalle società operanti in questo mercato) sono in assoluto le più numerose.

Il Ministero delle Telecomunicazioni italiano ha vietato la pubblicazione delle ultime tre cifre delle chiamate ricevute, in una distorta interpretazione della [direttiva 97/66 CE](#) del [15 dicembre 1997](#), il cui scopo era normalizzare la trasparenza verso gli utenti. [Sfruttando infatti la possibilità di garantire l'anonimato di chi chiama \(ma non di chi viene chiamato\)](#), viene resa difficoltosa l'individuazione del dialer senza il tabulato, che nei primi anni di applicazione era disponibile solo per chi lo chiedeva esplicitamente.

A fronte della mancanza di un servizio di "[Whois](#)" e della completa lista dei numeri chiamati, non era dunque possibile capire nell'immediatezza la causa di bollette particolarmente "salate", né effettuare appropriate denunce in tempi ragionevoli; ciò è anche dovuto al fatto che, secondo la legge italiana, l'Autorità Garante per la Concorrenza e il Mercato può agire solo in presenza di una denuncia circostanziata.

Un provvedimento dell'autorità garante della primavera [2002](#) prevede alcune norme per la pubblicazione dei costi, nella stessa pagina in cui i dialer vengono proposti. [Diversamente dalle normative europee per l'etichettatura, nonché di quelle per la pubblicità dei servizi telefonici, non](#)

sono state regolamentati gli obblighi di visualizzazione dei tariffari, riguardo a colore, formato e dimensione dei caratteri.

Dialer illegali

La maggioranza di questi [programmi](#) sono creati per connettersi a numeri a tariffazione speciale, [ad insaputa dell'utente](#). Solo una frazione limitata di questi dispositivi contiene l'indicazione corretta e visibile del costo, mentre la maggior parte dei dialer impostati per connettersi a numeri a tariffazione speciale utilizza metodi illegali, rientrando [così nel reato di truffa](#).

In mancanza di una legge, si moltiplicano le cause di conciliazione arbitrale fra i danneggiati dai dialer e gli operatori telefonici, una via più rapida di una causa civile per ottenere una riduzione o annullamento degli addebiti. Gli incontri hanno luogo con la mediazione delle associazioni dei consumatori, a sostegno di chi ha pagato la quota d'iscrizione.

L'unico provvedimento che impone delle limitazioni in materia di dialer è il regolamento [n.385 del 13 luglio 1995](#), che prevede la comunicazione dei costi del servizio; peraltro, si tratta di un regolamento disatteso nella stragrande maggioranza dei casi.

Detto regolamento rende possibile l'abilitazione di servizi dialer anche da parte di enti pubblici; un esempio di servizio dialer pubblico abbastanza diffuso è la consultazione delle [Gazzette Ufficiali](#) degli anni precedenti.

Essendo i dialer classificati come "servizi a valore aggiunto", vengono abilitati all'utente senza sua richiesta, e [la disabilitazione può essere a pagamento](#). Si tratta di un controsenso presente ai clienti di molti operatori telefonici, che si trovano a pagare per terminare un servizio mai richiesto.

Se [non avvertono della connessione a pagamento, i dialer sono illegali](#). Spesso l'avviso è presente, **ma poco visibile, a fondo pagina, scritto con caratteri piccoli o in lingue straniere che spingono il navigatore a credere si tratti di un innocuo [plug-in](#) da installare**. La legge, però, non prevede alcuna specifica sulla dimensione del carattere o sulla visibilità delle tariffe.

La maggior parte dei dialer maligni usa però metodi ancora più discutibili:

- [sfruttano](#) i [bug](#) dei [browser](#) o dei [programmi](#) di [posta elettronica](#) per installarsi automaticamente;
- [disabilitano](#) l'altoparlante del [modem](#) e i messaggi che normalmente appaiono durante la connessione, in modo che l'utente non si accorga della disconnessione o della composizione di un numero diverso da quello dell'ISP dai toni dei numeri durante la composizione;
- si [sostituiscono](#) alla connessione predefinita, in modo da essere utilizzato inconsapevolmente dall'utente ad ogni collegamento ad [Internet](#), con risultati devastanti per le finanze della vittima;
- [tentano](#) inoltre di impedire la propria disinstallazione, avviando automaticamente all'avvio un processo che provvede alla reinstallazione qualora l'utente tenti di cancellare il [malware](#).

Danni causati dai dialer

La conseguenza più rilevante della presenza di un dialer maligno sul [computer](#) è il [consistente addebito nella bolletta telefonica](#) causato dall'utilizzo inconsapevole di numeri a tariffazione speciale. Come già accennato, infatti, il fatto che la vittima sia convinta di utilizzare la propria

connessione usuale, magari a tariffe agevolate, e che il dialer faccia di tutto per tenere nascosta la sua presenza, conduce ad addebiti anche molto consistenti.

14.6. LEGGE SULLA PRIVACY

Per **privacy** s'intende il **diritto alla riservatezza delle informazioni personali e della propria vita privata**; spesso si traduce nella capacità di una persona (o di un gruppo di persone), di impedire che le informazioni che la riguardano diventino note ad altri, inclusi organizzazioni ed enti, qualora il soggetto non abbia volontariamente scelto di fornirle. I fondamenti costituzionali sono ravvisabili negli artt. 14, 15 e 21 Cost., rispettivamente riguardanti il domicilio, la libertà e segretezza della corrispondenza, e la libertà di manifestazione del pensiero.

La **legge sulla privacy** ha subito negli ultimi tempi una serie di modifiche. Dopo la "vecchia" **legge 675/96** (ARTICOLI) che regolava la materia, è entrato in vigore il nuovo Codice di cui al D.Lgs. n. **196/2003** il cui testo ha riaperto una nuova fase operativa in ordine alla gestione degli adempimenti e la loro trasparenza. Va segnalato comunque che il nuovo regolamento non sconvolge profondamente l'assetto preesistente e, in molti casi, introduce semplificazioni considerevoli; solo alcuni elementi costituiscono una vera novità in termini applicativi. Il più rilevante di questi è la costruzione del **Documento Programmatico per la sicurezza** ed una serie di nuove disposizioni.

Dal 1 Gennaio 2004 è in vigore, in Italia, il "Codice in materia di protezione dei dati personali" (Decreto legislativo n. 196 del 30/6/2003) che riforma interamente la disciplina sulla privacy. Il Codice abroga e sostituisce tutte le precedenti leggi, decreti e regolamenti in materia, riunendo in un unico organico contesto l'intera normativa sulla privacy.

Tutte le aziende sono tenute a rispettare la nuova normativa, la cui corretta applicazione consente, non solo di adempiere agli obblighi di legge, ma anche di migliorare l'organizzazione aziendale, i processi di lavoro e la qualità dei risultati. Il Codice richiede l'adozione di diverse misure di sicurezza per garantire che i dati trattati siano custoditi e controllati secondo alcune misure minime di sicurezza: a titolo di esempio possiamo citare il fatto che ogni persona che accede alla banca dati (dall'anagrafica dei clienti ad archivi contenenti dati sensibili) deve essere preventivamente **incaricata** e **riconosciuta** dal sistema attraverso un identificativo associato ad una **password** che deve essere lunga almeno otto caratteri, modificata all'atto del primo collegamento e non deve essere agevolmente riconducibile al proprietario. Altre misure minime prevedono **l'aggiornamento costante del software** con le ultime "patch" rilasciate dal produttore e il **backup settimanale** dei dati. Oltre alle misure minime il Codice prevede altre misure di sicurezza più ampie "idonee" a garantire ulteriormente la protezione dei dati e dei sistemi.

In generale il Codice prescrive di fatto la realizzazione di un vero e proprio sistema di sicurezza che protegga i dati custoditi all'interno dell'azienda.

Le misure di sicurezza adottate devono essere riportate in un Documento Programmatico annuale sulla Sicurezza (DPS) la cui redazione o aggiornamento deve essere riportata nella relazione accompagnatoria al bilancio aziendale. Il DPS deve essere redatto entro il 31 Marzo di ogni anno. La mancata adozione delle misure minime di sicurezza rende penalmente perseguibili gli inadempienti (ovvero coloro che hanno firmato la relazione accompagnatoria al bilancio), salvo l'adozione di un ravvedimento operoso. La mancata adozione delle misure minime o di quelle idonee può portare il soggetto cui si riferiscono i dati e che è stato danneggiato a chiedere un risarcimento dei danni.

DOCUMENTO PROGRAMMATICO

La redazione del documento programmatico per la sicurezza definito anche DPS è una “misura minima” prevista dalla legge e si traduce in un **rapporto di analisi** contenente la distribuzione dei compiti, l'analisi dei rischi sul trattamento dei dati e la documentazione di alcune soluzioni adottate ed altro ancora.

La precedente disciplina (la legge 675/96 detta legge sulla privacy) prevedeva già l'obbligo di predisporre e aggiornare il DPS, almeno annualmente, in caso di trattamento di **dati sensibili** o relativi a determinati provvedimenti giudiziari effettuato mediante elaboratori accessibili mediante una rete di telecomunicazioni disponibili al pubblico. I soggetti tenuti a predisporre il DPS hanno potuto redigerlo per la prima volta entro il 29 marzo 2000 o, al più tardi, entro il 31 dicembre 2000; dovendo rispettare l'obbligo di revisione almeno annuale, hanno dovuto aggiornare il rapporto negli anni successivi, anche nel 2003.

In base al nuovo **Codice della privacy**, la misura minima del DPS deve essere ora adottata dal titolare di un trattamento di dati sensibili o giudiziari effettuato con strumenti elettronici.

Come accennato, il DPS deve essere redatto da alcuni soggetti che non vi erano precedentemente tenuti (ad esempio, da chi trattava dati sensibili o giudiziari, ma con elaboratori non accessibili mediante una rete di telecomunicazioni disponibili al pubblico).

Infine, il contenuto stesso del DPS è arricchito da nuovi elementi che si aggiungono a quelli necessari in base alla precedente disciplina o ne specificano alcuni aspetti. Ad esempio, nel DPS occorre descrivere ora i criteri e le modalità per ripristinare la disponibilità dei dati in caso di distruzione o danneggiamento delle informazioni o degli strumenti elettronici.

Il DPS deve essere aggiornato entro il 31 marzo di ogni anno, non deve essere inviato a nessun ente e nemmeno al Garante per la Privacy. Andrà conservato presso l'impresa e verrà esibito in caso di richiesta da parte degli organi di verifica (Guardia di Finanza).

ADEMPIMENTI E OPPORTUNITA'

Chiunque tratta dati personali è tenuto a rispettare gli obblighi prescritti dal “Codice in materia di protezione dei dati personali”. Aziende, imprese, ditte, studi professionali, banche, assicurazioni, organizzazioni ed esercenti le professioni sanitarie, ed ogni altra categoria, privata e pubblica, indipendentemente dalle loro dimensioni, sono tenute ad operare nel rispetto di precise regole che riguardano la sicurezza dei dati e dei sistemi al fine di ridurre al minimo le fonti di rischio e garantire correttezza, integrità ed aggiornamento delle informazioni.

Tra gli interventi richiesti per la sicurezza dei dati e dei sistemi, a secondo dei casi è necessario organizzare e disciplinare l'uso di:

- sistemi di autenticazione informatica;
- credenziali di autenticazione (password, codici identificativi, carte a microprocessore, certificati digitali, rilevatori di caratteristiche biometriche);
- sistema di autorizzazione informatica;

- protezione dei dati e sistemi dalle intrusioni di virus, internet worm, programmi maligni;
- aggiornamenti delle vulnerabilità individuate con *patch*, *hot fix*, *service pack*;
- protezione da intrusioni nei sistemi informatici;
- back up dei dati e organizzazione del ripristino;
- redazione di un aggiornato documento programmatico sulla sicurezza;
- tecniche di cifratura.

E' necessario anche che i sistemi di rilevazione biometria, di videosorveglianza, di localizzatori di persone, di lavoro a distanza, siano organizzati in conformità al Codice.

LA SICUREZZA COME VANTAGGIO COMPETITIVO

La corretta applicazione delle misure di sicurezza consente non solo di adempiere agli obblighi di legge, ma anche di migliorare l'organizzazione aziendale ottimizzando i processi di lavoro ed operare nella consapevolezza che i dati trattati siano corretti, integri, aggiornati – *come richiesto dal Codice all'art. 11* - e costituiscano, perciò, vere informazioni d'impresa. La stessa impresa avrà quindi la garanzia di operare in sicurezza e in ottemperanza della legge e potrà godere della piena fiducia della propria clientela. In tal modo la sicurezza si trasforma da costo a investimento e vantaggio competitivo.

DIRITTO ALLA PROTEZIONE DEI DATI PERSONALI

Il Codice, all'art.1, introduce un nuovo e fondamentale diritto: *“chiunque ha diritto alla protezione dei dati personali che lo riguardano”*. I dati che devono essere protetti riguardano sia le persone fisiche (tutti noi), sia le persone giuridiche (i dati riferiti ad aziende, imprese, ditte, ecc.). La protezione dei dati personali è garantita da idonee e preventive misure di sicurezza obbligatorie per chi tratta i dati personali.

MISURE DI SICUREZZA,SANZIONI PENALI E AMMINISTRATIVE PER IL RESPONSABILE DELL'AZIENDA

Le misure di sicurezza richieste dal Codice sono articolate in due gruppi:

- quelle “minime”, la cui mancata adozione comporta sanzioni penali per il responsabile legale dell'azienda e/o se designato per il responsabile del trattamento (solitamente l'amministratore delegato o una figura di alto livello), ma anche per chiunque essendovi tenuto omette di adottarle;
- quelle più ampie, o “idonee”, decise in autonomia dal titolare in relazione alle proprie specificità che, se non adottate, in caso di danno dovuto a trattamenti di dati non protetti adeguatamente concorreranno all'individuazione delle responsabilità e del conseguente risarcimento economico.

Sono previste, inoltre, misure per titolari particolari quali i fornitori di un servizio di comunicazione elettronica accessibile al pubblico o gli organismi e gli esercenti le professioni sanitarie.

CREDENZIALI,AUTENTICAZIONI,AUTORIZZAZIONI

Alla base delle nuove misure minime sono poste le modalità per l'accesso ai dati che devono avvenire solo da parte delle persone autorizzate ed esplicitamente incaricate; ad esse dovranno essere assegnate o associate “credenziali di autenticazione”, cioè **parole chiave**, **codici identificativi**, **carte a microprocessore**, **token**, **certificati digitali**, o dispositivi che riconoscono le caratteristiche **biometriche**. Tali credenziali dovranno consentire “l'autenticazione informatica” delle persone incaricate del trattamento di dati. Inoltre,

quando più persone incaricate accedono ai dati, è necessario associare ad ogni soggetto uno specifico profilo per l'accesso. Il profilo identifica dei trattamenti di dati che possono essere svolti e costituisce "l'ambito del trattamento consentito"; l'intero processo è definito "sistema di autorizzazione" per l'accesso ai trattamenti consentiti e preventivamente individuati.

La legge definisce anche i criteri con cui le credenziali devono essere scelte; ad esempio la parola chiave usata in un sistema di autenticazione deve essere composta da almeno otto caratteri oppure, nel caso in cui lo strumento elettronico non lo permetta, da un numero di caratteri pari al massimo consentito; essa non deve contenere riferimenti agevolmente riconducibili all'incaricato e deve essere modificata da quest'ultimo al primo utilizzo e, successivamente, almeno ogni sei mesi. In caso di trattamento di dati sensibili e di dati giudiziari la parola chiave dovrà essere modificata almeno ogni tre mesi.

BACKUP,SUPPORTI RIMOVIBILI,RIPRISTINO

Se si trattano i cosiddetti dati sensibili o giudiziari questi dati dovranno essere protetti da ulteriori misure di sicurezza, quali:

- strumenti elettronici che evitano gli accessi abusivi (intrusioni);
- procedure per la generazione e la custodia di copie di sicurezza dei dati (back up);
- istruzioni organizzative e tecniche per la custodia e l'uso dei supporti rimovibili su cui sono memorizzati i dati al fine di evitare accessi non autorizzati e trattamenti non consentiti;
- disposizioni per riutilizzare o distruggere i supporti rimovibili sui quali sono registrati tali dati;
- strumenti per il ripristino della disponibilità dei dati e dei sistemi entro tempi certi e compatibili con i diritti degli interessati, non superiori a sette giorni.

TERMINOLOGIA

Dati

- "**dato personale**": qualunque informazione relativa a persona fisica, persona giuridica, ente od associazione, identificati o identificabili, anche indirettamente, mediante riferimento a qualsiasi altra informazione, ivi compreso un numero di identificazione personale;

- "**dati identificativi**": i dati personali che permettono l'identificazione diretta dell'interessato;

- "**dati sensibili**": i dati personali idonei a rivelare l'origine razziale ed etnica, le convinzioni religiose, filosofiche o di altro genere, le opinioni politiche, l'adesione a partiti, sindacati, associazioni od organizzazioni a carattere religioso, filosofico, politico o sindacale, nonché i dati personali idonei a rivelare lo stato di salute e la vita sessuale.

Soggetti

- "**titolare**": la persona fisica, la persona giuridica, la pubblica amministrazione e qualsiasi altro ente, associazione od organismo cui

competono, anche unitamente ad altro titolare, le decisioni in ordine alle finalità, alle modalità del trattamento di dati personali e agli strumenti utilizzati, ivi compreso il profilo della sicurezza;

- "**responsabile**": la persona fisica, la persona giuridica, la pubblica amministrazione e qualsiasi altro ente, associazione od organismo preposti dal titolare al trattamento di dati personali;
- "**incaricati**": le persone fisiche autorizzate a compiere operazioni di trattamento dal titolare o dal responsabile.

QUALI SONO QUINDI I DATI PERSONALI?

- a) Il nome, il cognome, l'indirizzo, il numero di telefono, il codice fiscale, la Partita Iva, dati bancari...
- b) Informazioni circa la composizione del nucleo familiare, la professione esercitata da un determinato soggetto, sia fisico che giuridico, la sua formazione...

E QUELLI SENSIBILI?

- c) **Fotografie, radiografie, video, suoni, impronte...**
- d) Informazioni relative al profilo creditizio, alla **retribuzione...**
- e) Informazioni relative alla salute di un soggetto, alla vita sessuale, alla partecipazione ad associazioni di categoria, a partiti, trattenute sindacali, cartelle cliniche, rilevazioni di presenze

Misure minime previste dalla legge sulla privacy per il trattamento dei dati con l'ausilio di mezzi informatici

Le misure minime riguardano:

- a) autenticazione informatica;
- b) adozione di procedure di gestione delle credenziali di autenticazione;
- c) utilizzazione di un sistema di autorizzazione;
- d) aggiornamento periodico dell'individuazione dell'ambito del trattamento consentito ai singoli incaricati e addetti alla gestione o alla manutenzione degli strumenti elettronici;
- e) protezione degli strumenti elettronici e dei dati rispetto a trattamenti illeciti di dati, ad accessi non consentiti e a determinati programmi informatici;
- f) adozione di procedure per la custodia di copie di sicurezza, il ripristino della disponibilità dei dati e dei sistemi;
- g) tenuta di un aggiornato documento programmatico sulla sicurezza;
- h) adozione di tecniche di cifratura o di codici identificativi per determinati trattamenti di dati idonei a rivelare lo stato di salute o la vita sessuale effettuati da organismi sanitari.

Capitolo 15. HTML

Fonte: Internet

15.1. Documenti HTML

Cos'è un documento HTML

Un documento HTML è un semplice testo (ASCII) che può essere creato con un qualsiasi programma di scrittura. Se si usano programmi sofisticati, quale Word, bisognerà ricordarsi di salvare il documento con l'opzione "solo testo con interruzione di riga" (text only with line breaks).

Tags

Per definire i vari elementi del testo in un documento HTML usate *tags*. I tags HTML sono composti, in ordine, da un simbolo (<), un comando e un simbolo (>). I comandi sono spesso in coppia (ad esempio, <H1> e </H1>) per iniziare e terminare l'esecuzione di una certa istruzione. Il comando (tag) che indica la fine dell'istruzione è come quello iniziale, salvo il simbolo (/) che precede il testo entro le parentesi.

Alcuni elementi possono ammettere un "*attributo*", che fornisce una informazione aggiuntiva inclusa nel comando iniziale. Per esempio, si può specificare l'allineamento delle immagini rispetto al testo (sopra, in mezzo, sotto) includendo l'attributo appropriato (top, middle, bottom) nel comando che chiama l'immagine.

NOTA: *Il linguaggio HTML non è "case sensitive", non distingue caratteri minuscoli e maiuscoli all'interno dei comandi. <title> è equivalente a <TITLE> oppure <TiTlE>.*

Un documento HTML minimo

Tutti i documenti HTML dovrebbero contenere alcuni comandi standard. Ogni documento consiste di una intestazione (head) e un corpo (body). L'intestazione contiene almeno il titolo della pagina Web, mentre il corpo contiene il testo vero e proprio, comprendente paragrafi, liste, e altri elementi. I browser si aspettano specifiche istruzioni in quanto sono stati programmati sulla base degli standard HTML.

Gli elementi base sono mostrati in questo semplice documento:

```
<html>
<head>
<TITLE>Un semplice esempio di HTML</TITLE>
</head>
<body>
<H1>Il linguaggio HTML è facile da imparare</H1>
<P>Benvenuti nel mondo HTML.
```

```
Questo e' il primo paragrafo. Anche se e' cosi' corto e'  
comunque un paragrafo!</P>  
<P>E questo e' un secondo paragrafo.</P>  
</body>  
</html>
```

Markup Tags

HTML

Questo comando comunica al vostro browser che il documento contiene informazioni in linguaggio HTML. Anche l'estensione `.html` nel nome del documento indica che il documento ha il formato HTML e per questo va utilizzata (se il vostro computer ammette estensioni di tre caratteri soltanto, potete scrivere `htm` al posto di `html`; ad esempio, `LeeHome.htm`, invece di `LeeHome.html`).

INTESTAZIONE (HEAD)

Il comando di intestazione identifica la prima parte del documento HTML, che contiene il titolo. Il titolo viene mostrato nella barra orizzontale in alto della finestra del browser.

TITOLO (TITLE)

Il titolo viene specificato con il comando `TITLE`. Oltre ad essere visualizzato nella finestra del browser, esso identifica il documento anche nell'elenco dei bookmarks e viene utilizzato dai motori di ricerca. Quindi e' opportuno utilizzare titoli brevi, unici e descrittivi.

BODY

La seconda e piu' ampia parte del documento HTML e' il corpo (body), che contiene tutta l'informazione che verra' visualizzata nella finestra del browser. I comandi elencati sotto servono per gestire questa parte del documento.

Headings

HTML ha sei livelli di titolazione (headings), numerati da 1 a 6, in ordine decrescente di importanza. L'intestazione principale di un documento usualmente si realizza con il livello 1. Ogni livello di heading corrisponde a caratteri di dimensioni maggiori rispetto al testo normale e normalmente in grassetto.

La sintassi degli headings e':

```
<Hy>Titolo di una sezione </Hy>
```

dove `y` e' un numero da 1 a 6 che specifica l'importanza del titolo della sezione.

Paragrafi

Diversamente da quanto accade con normali programmi per scrittura di testi (word processor), in HTML il controllo "carriage returns" (il tasto "Invio" per le tastiere italiane) non ha alcun significato. Così non vi dovete preoccupare della lunghezza delle righe del

vostro documento (anche se e' opportuno comunque non superare i 72 caratteri per riga). Il browser leggerà la sequenza di parole del testo senza badare alle interruzioni di linea e alle righe vuote. Una sequenza di righe vuote sarà semplicemente collassata in un singolo spazio bianco tra due parole.

Nell'esempio dato nella sezione "un documento HTML minimo", il primo paragrafo e' scritto come:

```
<P>Benvenuti nel mondo HTML.  
Questo e' il primo paragrafo. Anche se e' cosi' corto e'  
comunque un paragrafo!</P>
```

Nel documento sorgente appaiono dunque 3 righe distinte. Un qualunque browser ignora le interruzioni di linea, scrivendo tutto di seguito, e comincia un nuovo paragrafo solo se incontra un nuovo comando <P>.

Importante: Dovete indicare i paragrafi con il comando <P>. Il browser ignora qualsiasi indentazione del testo sorgente o eventuali righe bianche. Senza <P> il documento apparirà come un unico lungo paragrafo (ad eccezione del testo scritto con il comando di preformattazione, che vedremo dopo). Per mantenere la leggibilità del documento HTML, mettete i titoli in righe distinte e usate righe bianche per separare paragrafi e diverse sezioni del testo. Questo vi aiuterà nella scrittura e nella correzione dei documenti, anche se poi il browser ignorerà tutti gli spazi aggiunti.

NOTA: Il comando di chiusura </P> può essere omissso. Infatti, quando un browser incontra un nuovo comando <P>, ne deduce che il paragrafo precedente e' finito anche in assenza di una sua chiusura esplicita.

Usare i comandi <P> e </P> in coppia permette anche di includere l'eventuale attributo di allineamento, `ALIGN=alignment`.

```
<P ALIGN=CENTER>  
Questo e' un paragrafo centrato. [Si veda la versione formattata  
sotto.]  
</P>
```

Questo e' un paragrafo centrato.

Liste

Il linguaggio HTML permette di creare liste non numerate (unnumbered), numerate (numbered) e liste di definizioni. Potete anche includere una sottolista in una lista, anche se l'uso di troppo liste concatenate può rendere difficile la scrittura del documento.

Liste non numerate (UL)

Per creare una lista non numerata, con un pallino davanti ad ogni elemento della lista,

1. si inizi con il comando
2. si scriva il comando (list item) seguito dal primo elemento della lista; non e' richiesto il comando di chiusura
3. si introducano gli ulteriori elementi della lista e si termini il tutto con .

Ecco un esempio di lista con tre elementi:

```
<UL>
<LI> mele
<LI> banane
<LI> pompelmi
</UL>
```

L'output e':

- o mele
- o banane
- o pompelmi

Il comando puo' essere seguito da diversi paragrafi, ciascuno separato da <P>.

Liste numerate

Una lista numerata (*ordered list*, OL) e' identica come struttura a quella non numerata, salvo sostituire con . Gli elementi della lista vengono introdotti con lo stesso comando . Il seguente testo HTML

```
<OL>
<LI> mele
<LI> banane
<LI> pompelmi
</OL>
```

produce come output:

7. mele
8. banane
9. pompelmi

Liste di definizioni

Una lista di definizioni (<DL>) contiene normalmente un alternanza di "definition term" (<DT>) e "definition definition" (<DD>). Generalmente un browser scrivera' la definizione su righe distinte.

Esempio:

```
<DL>
<DT> NCSA
<DD> NCSA, the National Center for Supercomputing Applications,
      is located on the campus of the University of Illinois
      at Urbana-Champaign.
<DT> Cornell Theory Center
<DD> CTC is located on the campus of Cornell University in Ithaca,
      New York.
</DL>
```

L'output e':

NCSA

NCSA, the National Center for Supercomputing Applications, is located on the campus of the University of Illinois at Urbana-Champaign.

Cornell Theory Center

CTC is located on the campus of Cornell University in Ithaca, New York.

Le parti <DT> e <DD> possono contenere vari paragrafi (separati da <P>), liste o altre liste di definizioni.

Nel caso definizioni brevi, si puo' usare il comando COMPACT per rendere la lista piu' compatta. Ad esempio, se si elencano le opzioni di un sistema operativo, queste possono essere scritte ciascuna su un'unica riga.

```
<DL COMPACT>
<DT> -i
<DD>invokes NCSA Mosaic for Microsoft Windows using the
initialization file defined in the path
<DT> -k
<DD>invokes NCSA Mosaic for Microsoft Windows in kiosk mode
</DL>
```

L'output e':

```
-i
invokes NCSA Mosaic for Microsoft Windows using the initialization file defined in the
path.
-k
invokes NCSA Mosaic for Microsoft Windows in kiosk mode.
```

Liste concatenate

Le liste possono essere incluse in altre liste. Si possono avere anche piu' paragrafi, contenenti a loro volta liste, in un singolo elemento di lista.

Esempio:

```
<UL>
<LI> A few New England states:
  <UL>
    <LI> Vermont
    <LI> New Hampshire
    <LI> Maine
  </UL>
<LI> Two Midwestern states:
  <UL>
    <LI> Michigan
    <LI> Indiana
  </UL>
</UL>
```

L'output e':

- o A few New England states:

- Vermont
- New Hampshire
- Maine
- Two Midwestern states:
 - Michigan
 - Indiana

Testo preformattato

Si usi il comando `<PRE>` per generare testi con "fixed-width font" (caratteri di larghezza fissata). Lo stesso comando rende visibili gli spazi aggiunti nel documento HTML e le interruzioni di linea. Questo e' utile, tra l'altro, per riprodurre listati di programmi. Per esempio,

```
<PRE>
Questo è un testo
formattato
con il comando PRE
</PRE>
```

e' mostrato dal browser come:

```
Questo è un testo
formattato
con il comando PRE
```

Il comando `<PRE>` puo' essere usato con l'opzione `WIDTH` che specifica il numero massimo di caratteri per linea. Il comando `WIDTH` segnala anche al browser di scegliere un tipo di carattere appropriato.

Nelle sezioni preformattate con `<PRE>` possono essere inclusi anche riferimenti ipertestuali (hyperlinks). Dovreste evitare invece di includere altri comandi HTML.

Si noti che, siccome i simboli `<`, `>`, e `&` hanno un significato speciale in HTML, non possono essere usati separatamente nel testo. Se si vuole che appaiano nell'output e' necessario usare particolari definizioni (escape sequences), come `<`, `>`, and `&`.

Extended Quotations

Si puo' usare il comando `<BLOCKQUOTE>` per visualizzare una lunga citazione in un blocco separato dal resto del testo. La maggior parte dei browser modifica i margini del testo per separare ed evidenziare la parte inclusa in `<BLOCKQUOTE>`.

Ad esempio,

```
<BLOCKQUOTE>
<P>Omit needless words.</P>
<P>Vigorous writing is concise. A sentence should contain no
unnecessary words, a paragraph no unnecessary sentences, for the
same reason that a drawing should have no unnecessary lines and a
machine no unnecessary parts.</P>
```

```
--William Strunk, Jr., 1918  
</BLOCKQUOTE>
```

ha come risultato

Omit needless words.

Vigorous writing is concise. A sentence should contain no unnecessary words, a paragraph no unnecessary sentences, for the same reason that a drawing should have no unnecessary lines and a machine no unnecessary parts.

--William Strunk, Jr., 1918

Addresses

Il comando <ADDRESS> viene usato generalmente per specificare l'autore del documento, il suo indirizzo, e la data di revisione del documento. Di solito si tratta dell'ultima parte di un documento HTML.

Per esempio,

```
<ADDRESS>  
A Beginner's Guide to HTML / NCSA / pubs@ncsa.uiuc.edu / revised April  
96  
</ADDRESS>
```

ha come risultato:

A Beginner's Guide to HTML / NCSA / pubs@ncsa.uiuc.edu / revised April 96

NOTA: <ADDRESS> non e' da intendere come indirizzo postale. Si veda la sezione "Interruzione di riga forzata" per una scrittura semplice di indirizzo postale.

Interruzione di riga

Il comando
 (break) forza l'interruzione di riga senza dare spazi extra tra le linee di testo. Ad esempio, si puo' usare
 per un indirizzo postale:

```
National Center for Supercomputing Applications<BR>  
605 East Springfield Avenue<BR>  
Champaign, Illinois 61820-5518<BR>
```

Il risultato sara':

National Center for Supercomputing Applications
605 East Springfield Avenue
Champaign, Illinois 61820-5518

Linee orizzontali

Il comando <HR> (horizontal rule) produce una linea orizzontale utile per separare visivamente diverse sezioni del testo. Per esempio, e' usuale aggiungere una linea di separazione tra il testo del documento e la parte <address> posta in fondo.

Si puo' variare lo spessore della linea e la sua lunghezza variando i parametri WIDTH e SIZE. Ad esempio,

```
<HR SIZE=4 WIDTH="50%">
```

produce:



Formato dei caratteri

grassetto (bold)

<I>

italico

<TT>

Escape Sequences

Le escape sequence permettono di visualizzare:

- o i caratteri speciali di HTML
- o i caratteri non disponibili nel "plain ASCII character set"

Tre caratteri ASCII -- (<), (>), e (&) -- hanno significati speciali in HTML e non possono essere usati "come sono" nel testo. Se volete che siano visualizzati dal browser come parte del testo, dovete usare le escape sequence corrispondenti:

< ;

escape sequence per <

> ;

escape sequence per >

& ;

escape sequence per &

Altre sequenze di caratteri permettono di visualizzare ulteriori caratteri, come accenti, umlaut, tilde, ecc.:

ö ;

o con umlaut: ö

ñ ;

n con tilde: ñ

È ;

E con accento grave: È

NOTA: a differenza del resto di HTML, le escape sequences sono "case sensitive", cioè distinguono tra maiuscole e minuscole. Non potete, ad esempio, usare `<` invece di `<`.

15.2. Riferimenti ipertestuali (Linking)

La maggior potenza di HTML viene dal fatto di permettere la connessione tra un testo e/o una immagine con altri documenti, o sezioni particolari di documenti. Un qualsiasi browser segnala la possibile connessione rendendo evidenti alcune parole e/o immagini (con sottolineatura, bordatura, uso di un colore particolare, ecc.) che indicano il riferimento ipertestuale (hyperlink, o semplicemente link). La connessione si attiva portando il puntatore del mouse sulla parte di testo evidenziata e facendo clic.

Il comando HTML associato ad un hyperlink e' `<A>`, che sta per *anchor*. Per includere un "anchor" nel vostro documento:

14. si inizi con `<A` (si includa uno spazio dopo A)
15. si specifichi il documento a cui ci si collega, inserendo il parametro `HREF="filename"` seguito dalla parentesi di chiusura (`>`)
16. si scriva il testo che serve come riferimento ipertestuale nel documento in fase di scrittura
17. si chiuda il riferimento con il comando ``

Ecco un esempio di link in un documento chiamato `US.html`:

```
<A HREF="MaineStats.html">Maine</A>
```

Questa sequenza rende la parola *Maine* un link ipertestuale al documento `MaineStats.html`, che e' nella stessa directory del documento di partenza.

Riferimenti relativi e assoluti

Potete creare un link a documenti in altre directory specificando il percorso relativo, a partire dal documento in corso di scrittura. Per esempio, un link al documento `NYStats.html` situato nella subdirectory `AtlanticStates` sarebbe:

```
<A HREF="AtlanticStates/NYStats.html">New York</A>
```

Questi sono chiamati link relativi, dato che si specifica il percorso relativo ad un dato documento. Si possono anche utilizzare link assoluti (indirizzi URL completi), ma link relativi sono spesso piu' efficienti nell'accesso al server.

Il percorso (path) relativo ad un dato documento viene specificato usando la sintassi standard UNIX.

Ad esempio, se siete nel documento `NYStats.html` e fate riferimento al documento originale `US.html`, il link dovra' essere nella forma:

```
<A HREF=" ../US.html">United States</A>
```

In generale e' preferibile usare riferimenti relativi perche':

18. e' piu' facile muovere un gruppo di documenti da una macchina all'altra o da un utente all'altro, mantenendo validi i riferimenti relativi
19. e' piu' efficiente la connessione al server
20. i riferimenti sono piu' brevi da scrivere

Tuttavia bisogna usare riferimenti assoluti quando si ha un link a documenti che non sono direttamente collegati. Ad esempio, si consideri un gruppo di documenti che fanno riferimento ad un manuale. I link tra i documenti del gruppo conviene sia relativo. Il link al manuale conviene sia assoluto. In tal modo lo spostamento in blocco dei documenti, escluso il manuale, non necessita di alcun cambiamento di riferimenti.

Riferimenti a sezioni specifiche

Il riferimento ipertestuale puo' essere fatto anche a *particolari sezioni* di un documento (sia dello stesso documento che di altri). Questa operazione e' detta "named anchor" dato che, per creare il link, si inseriscono opportuni "nomi" nel documento HTML.

Riferimenti a sezioni di diversi documenti

Supponiamo che si voglia creare un link in un documento A (`documentA.html`) ad una sezione di un altro documento (`MaineStats.html`).

Allora, si inserisca il link in `documentA.html` nella forma:

```
Oltre a molti altri parchi, nello stato del Maine si trova anche  
<a href="MaineStats.html#ANP">l'Acadia National Park</a>.
```

Pensate ai caratteri dopo il simbolo # come a un indicatore inserito nel documento `MaineStats.html`. Questo indicatore ("ANP", in questo esempio) dice al vostro browser di visualizzare il documento `MaineStats.html`, anziche' dall'inizio, a partire dal punto indicato con il nome "ANP". E' necessario che questo indicatore esista nel documento `MaineStats.html`. Se non c'e', dovete inserirlo nel punto voluto, usando il "named anchor":

```
<A NAME="ANP">Acadia National Park</a>
```

Con entrambi questi elementi al loro posto potete indirizzare il lettore direttamente all'Acadia National Park con un semplice link posto in `documentA.html`.

Riferimenti a sezioni di uno stesso documento

La tecnica e' la stessa salvo omettere il nome del documento.

Per esempio, un link alla sezione nominata "ANP" entro lo stesso documento MaineStats, avra' la forma:

```
...informazioni su <A HREF="#ANP">Acadia National Park</a>  
si trovano piu' avanti in questo testo.
```

Siate sicuri di includere il comando nel punto in cui volete saltare.

I riferimenti di tipo "named anchor" sono particolarmente utili se pensate che il vostro documento debba essere stampato integralmente oppure quando avete una serie di brevi informazioni sparse nel documento, che volete sintetizzare in un indice.

Mailto

Potete fare in modo che un lettore della vostra pagina mandi facilmente un messaggio elettronico (e-mail) ad un indirizzo dato; basta usare l'attributo `mailto` nel link ipertestuale con il seguente formato:

```
<A HREF="mailto:emailinfo@host">Name</a>
```

Per esempio, scrivete:

```
<A HREF="mailto:pubs@ncsa.uiuc.edu">NCSA Publications  
Group</a>
```

15.3. Inserimento di immagini

Molti browser possono visualizzare immagini inserite nel testo; le immagini visualizzabili hanno il formato X Bitmap (XBM), oppure GIF o JPEG. Altri formati saranno probabilmente accettati dalle future versioni dei browser. Ogni immagine impiega un po' di tempo per essere processata e visualizzata, rallentando l'accesso al documento. Quindi la scelta delle immagini da includere deve essere fatta con attenzione.

Per inserire una immagine nel testo, usare il comando

```
<IMG SRC=" ImageName" >
```

dove *ImageName* e' l'indirizzo URL del file contenente l'immagine.

Se l'immagine e' un file in formato GIF, il nome del file deve terminare con `.gif`. Immagini in formato X Bitmap hanno nomi che terminano `.xbm`; infine, il formato JPEG deve avere un nome che termina con `.jpg` o `.jpeg`.

Dimensionamento

Potete includere due attributi nel comando per assegnare una dimensione appropriata all'immagine. Gli attributi HEIGHT e WIDTH permettono al browser di predisporre lo spazio (in pixel) riservato all'immagine.

Per esempio, per includere un'immagine autoritratto, una possibile scelta può essere:

```
<IMG SRC="SelfPortrait.gif" HEIGHT=100 WIDTH=65>
```

Allineamento

Avete una certa flessibilità nel posizionare le vostre immagini rispetto al testo. Potete separare l'immagine dal testo che precede e/o segue. Potete allineare l'immagine a destra o sinistra del testo. Potete provare le varie possibilità finché trovate quella che vi sembra migliore.

Allineamento di testo e immagine

Per default (cioè, ogni volta che non specificate diversamente) il lato inferiore dell'immagine è allineato con la base dei caratteri del testo che segue, come in questo paragrafo. Potete anche allineare il testo al bordo superiore dell'immagine usando l'attributo ALIGN=TOP.

Usando il comando , il testo è stato allineato al bordo superiore dell'immagine.

Immagini senza testo

Per visualizzare una immagine da sola (ad esempio, il logo della vostra azienda all'inizio di ogni pagina), basta trattarla come un paragrafo separato, a cui si possono associare eventuali attributi di posizionamento per visualizzarla al centro, a destra o a sinistra della finestra del browser. Ad esempio,

```
<p ALIGN=CENTER>  
<IMG SRC = "images/casa.gif">  
</p>
```

Background

Versioni recenti dei browser possono caricare immagini per usarle come sfondo (background) di una pagina.:

```
<BODY BACKGROUND="filename.gif">
```

Background Color

Per default (cioè, ogni volta che non specificate diversamente) il browser mostra il testo con caratteri neri su sfondo grigio. Tuttavia potete cambiare il colore di entrambi a vostro piacimento.

Per cambiare il colore del testo, dei link da visitare e quelli visitati, basta usare attributi del comando <BODY>. Per esempio, il comando

```
<BODY BGCOLOR="#000000" TEXT="#FFFFFF" LINK="#9690CC">
```

crea una finestra con sfondo nero (BGCOLOR), testo bianco (TEXT), e link argentati (LINK).

15.4. Tabelle

Elementi della tabella	
Elemento	Descrizione
<TABLE> ... </TABLE>	Definisce una tabella in HTML. Se e' presente l'attributo BORDER, il vostro browser mostrera' una tabella con bordo.
<CAPTION> ... </CAPTION>	Definisce la didascalia della tabella (o titolo). La posizione del titolo e' centrata in alto. L'attributo ALIGN=BOTTOM puo' essere usato per mettere la didascalia sotto la tabella. NOTA: Qualsiasi tipo di comando HTML puo' essere usato nella didascalia
<TR> ... </TR>	Specifica una righe entro la tabella. Potete definire attributi validi per l'intera riga: ALIGN (LEFT, CENTER, RIGHT) e/o VALIGN (TOP, MIDDLE, BOTTOM). Si veda il significato degli attributi alla fine di questa tabella.
<TH> ... </TH>	Definisce una cella di intestazione (header). Per default il contenuto di tale cella risulta centrato e in grassetto. Possono essere utilizzati ulteriori attributi per specificare l'aspetto delle celle "header" (si veda sotto).
<TD> ... </TD>	Definisce una cella per i dati. Per default il testo in queste celle sara' allineato a sinistra e centrato verticalmente. Anche per queste celle possono essere usati attributi opzionali.

Attributi per tabelle	
NOTA: Attributi assegnati entro i comandi <TH> ... </TH> o <TD> ... </TD> hanno prioritá rispetto a quelli scritti in <TR> ... </TR>.	
Attributi	Descrizione
ALIGN (LEFT, CENTER, RIGHT)	Allineamento orizzontale della cella.
VALIGN (TOP, MIDDLE, BOTTOM)	allineamento verticale della cella.
COLSPAN= <i>n</i>	Il numero (<i>n</i>) di colonne per singola cella.
ROWSPAN= <i>n</i>	Il numero (<i>n</i>) di righe per singola cella.
NOWRAP	Disattiva il "word wrapping" entro una cella.

Formato generale di una tabella

Il formato di una tabella appare come:

<code><TABLE></code>	<code><== start of table definition</code>
<code><CAPTION> caption contents </CAPTION></code>	<code><== caption definition</code>
<code><TR></code>	<code><== start of first row def.</code>
<code><TH> cell contents </TH></code> head)	<code><== first cell in row 1 (a</code>
<code><TH> cell contents </TH></code> head)	<code><== last cell in row 1 (a</code>
<code></TR></code>	<code><== end of first row def.</code>
<code><TR></code>	<code><== start of second row def.</code>
<code><TD> cell contents </TD></code>	<code><== first cell in row 2</code>
<code><TD> cell contents </TD></code>	<code><== last cell in row 2</code>
<code></TR></code>	<code><== end of second row def.</code>
<code><TR></code>	<code><== start of last row def.</code>
<code><TD> cell contents </TD></code>	<code><== first cell in last row</code>
...	
<code><TD> cell contents </TD></code>	<code><== last cell in last row</code>
<code></TR></code>	<code><== end of last row def.</code>
<code></TABLE></code>	<code><== end of table def.</code>

I comandi `<TABLE>` e `</TABLE>` **devono** contenere tutte le altre definizioni della tabella. Il primo dato entro la tabella deve essere `CAPTION`, che e' opzionale. Poi si puo' inserire un numero qualsiasi di righe definite da `<TR>` e `</TR>`. Entro una riga si puo' avere un numero qualsiasi di colonne definite da `<TD>...</TD>` o `<TH>...</TH>`. Ogni riga della tabella ha un formato indipendente da quello delle righe che precedono e seguono. In questo modo si puo' realizzare una singola cella, come quella "Attributi per tabelle" sopra, che occupa tutte le colonne della tabella.

15.5. I FOGLI CSS

I fogli di stile rappresentano una svolta di rilievo per i progettisti di pagine web, poiché espandono le possibilità di migliorare l'aspetto delle loro pagine. Le tecniche per ottenere questi miglioramenti, includono:

- Uso di estensioni proprietarie dell'HTML

- Conversione di testo in immagini
- Uso di immagini per il controllo degli spazi bianchi
- Uso di tabelle per la formattazione della pagina
- Scrivere un programma invece di usare l'HTML

Con i fogli di stile è facile specificare la quantità di spazio bianco tra righe di testo, la misura dei rientri di riga, i colori usati per il testo e per lo sfondo, la grandezza e lo stile dei caratteri, ed una moltitudine di altri dettagli.

Per esempio, il seguente breve foglio di stile (archiviato nel file "speciale.css") imposta a verde il colore del testo di un paragrafo e lo circonda con un bordo rosso continuo:

```
P.speciale {
color : green;
border: solid red;
}
```

Gli autori possono collegare questo foglio di stile al loro documento sorgente HTML tramite l'elemento [LINK](#):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <LINK href="speciale.css" rel="stylesheet" type="text/css">
  </HEAD>
  <BODY>
    <P class="speciale">Questo paragrafo dovrebbe avere uno speciale
    testo verde.
  </BODY>
</HTML>
```

Impostare il linguaggio predefinito per i fogli di stile

Gli autori devono specificare il linguaggio per fogli di stile delle informazioni di stile associate con un documento HTML.

Gli autori dovrebbero usare l'elemento [META](#) per impostare il linguaggio per fogli di stile predefinito di un documento. Per esempio, per impostare CSS come linguaggio predefinito, gli autori dovrebbero inserire la seguente dichiarazione nell'[HEAD](#) dei loro documenti:

```
<META http-equiv="Content-Style-Type" content="text/css">
```

Il linguaggio per fogli di stile predefinito può anche essere impostato per mezzo di intestazioni HTTP. La precedente dichiarazione [META](#) è equivalente all'intestazione HTTP:

```
Content-Style-Type: text/css
```

Informazioni di stile in riga

Definizioni di attributo

style = [style \[CN\]](#)

Questo attributo specifica informazioni di stile per l'elemento corrente.

Questo esempio di CSS imposta le informazioni sul colore e la grandezza dei caratteri per il testo di uno specifico paragrafo.

```
<P style="font-size: 12pt; color: fuchsia">Non sono meravigliosi i
fogli di stile?
```

In CSS, le dichiarazioni di proprietà hanno la forma "nome : valore" e sono separate tramite un punto e virgola.

Per specificare le informazioni di stile per più di un elemento, gli autori dovrebbero servirsi dell'elemento `STYLE`. Per una flessibilità ottimale, gli autori dovrebbero definire gli stili in fogli di stile esterni.

Informazioni di stile nell'intestazione: l'elemento `STYLE`

```

type          %ContentType; #REQUIRED -- tipo di contenuto del linguaggio di
stile --
media         %MediaDesc;  #IMPLIED  -- progettato per l'uso con questi media
--
title         %Text;       #IMPLIED  -- titolo di avvertimento --
>
```

Marcatore iniziale **obbligatorio**, Marcatore finale: **obbligatorio**

Definizioni di attributo

type = [content-type \[CI\]](#)

Questo attributo specifica il linguaggio per fogli di stile del contenuto di un elemento e sovrascrive il linguaggio per fogli di stile predefinito. Il linguaggio per fogli di stile è specificato come un tipo di contenuto (es.: "text/css"). Gli autori dovrebbero fornire un valore per questo attributo; non esiste un valore predefinito per questo attributo.

media = [media-descriptors \[CI\]](#)

Questo attributo specifica il mezzo di destinazione designato per le informazioni di stile. Può essere un descrittore per un singolo mezzo oppure un elenco separato da virgole. Il valore predefinito di questo attributo è "screen".

Attributi definiti

[lang](#) ([informazioni sulla lingua](#)),

[dir](#) ([direzione del testo](#))

[title](#) ([titolo dell'elemento](#))

L'elemento [STYLE](#) consente agli autori di inserire regole per fogli di stile nell'intestazione del documento. L'HTML ammette un qualsiasi numero di elementi [STYLE](#) nella sezione [HEAD](#) di un documento.

La seguente dichiarazione di [STYLE](#) CSS aggiunge un bordo intorno ad ogni elemento [H1](#) nel documento e lo centra nella pagina.

```
<HEAD>
  <STYLE type="text/css">
    H1 {border-width: 1; border: solid; text-align: center}
  </STYLE>
</HEAD>
```

Per specificare che queste informazioni di stile dovrebbero essere applicate soltanto agli elementi [H1](#) di una specifica classe, le modifichiamo come segue:

```
<HEAD>
  <STYLE type="text/css">
    H1.miaclasse {border-width: 1; border: solid; text-align: center}
  </STYLE>
</HEAD>
<BODY>
  <H1 class="miaclasse"> Questo H1 è influenzato dal nostro stile
</H1>
  <H1> Quest'altro non è influenzato dal nostro stile </H1>
</BODY>
```

Ed infine, per limitare la sfera d'influenza delle informazioni di stile ad una singola occorrenza di [H1](#), impostiamo l'attributo [id](#):

```
<HEAD>
  <STYLE type="text/css">
    #mioid {border-width: 1; border: solid; text-align: center}
  </STYLE>
</HEAD>
<BODY>
  <H1 class="miaclasse"> Questo H1 non è influenzato </H1>
  <H1 id="mioid"> Questo H1 è influenzato dallo stile </H1>
  <H1> Questo H1 non è influenzato </H1>
</BODY>
```

Benché sia possibile impostare informazioni di stile per la quasi totalità degli elementi HTML, due elementi, [DIV](#) e [SPAN](#), risultano particolarmente utili per il fatto che non impongono alcuna semantica di presentazione

Quando combinati con i fogli di stile, questi elementi consentono agli utenti di estendere indefinitamente l'HTML, in particolare se adoperati con gli attributi [class](#) e [id](#).

Nell'esempio successivo, adoperiamo l'elemento [SPAN](#) per impostare come maiuscoletto lo stile dei caratteri di alcune delle parole iniziali di un paragrafo.

```
<HEAD>
  <STYLE type="text/css">
```

```

    SPAN.sc-ex { font-variant: small-caps }
  </STYLE>
</HEAD>
<BODY>
  <P><SPAN class="sc-ex">Alcune delle parole</SPAN> iniziali di
questo
  paragrafo appaiono in maiuscoletto.
</BODY>

```

Nell'esempio successivo, usiamo [DIV](#) e l'attributo [class](#) per impostare la giustificazione del testo in una serie di paragrafi, che compongono il brano di compendio di un articolo scientifico. Queste informazioni di stile potrebbero essere riutilizzate per altri brani di compendio, impostando l'attributo [class](#) da qualche altra parte nel documento.

```

<HEAD>
  <STYLE type="text/css">
    DIV.Compendio { text-align: justify }
  </STYLE>
</HEAD>
<BODY>
  <DIV class="Compendio">
    <P>La gamma di prodotti Capotribù è il nostro
    articolo di punta per l'anno venturo. Questo
    rendiconto spiega come posizionare Capotribù rispetto
    ai prodotti antagonisti.

    <P>Capotribù rimpiazza la gamma Comandante, che rimarrà
    sul listino prezzi fino ad ulteriore avviso.
  </DIV>
</BODY>

```

Fogli di stile esterni

Gli autori possono separare i fogli di stile dai documenti HTML. Ciò offre numerosi benefici:

- Autori e gestori di siti possono condividere i fogli di stile attraverso una quantità di documenti (e siti).
- Gli autori possono modificare il foglio di stile senza che sia necessario apportare cambiamenti al documento.
- I programmi utente possono caricare selettivamente i fogli di stile (in base alle descrizioni dei media).

Fogli di stile preferenziali ed alternativi

L'HTML consente agli autori di associare ad un documento un qualsiasi numero di fogli di stile esterni. Il linguaggio per fogli di stile definisce l'interazione tra molteplici fogli di stile esterni (ad esempio, le regole "a cascata" CSS).

Gli autori possono specificare un certo numero di fogli di stile che si escludono a vicenda, chiamati fogli di stile *alternativi*. Gli utenti possono selezionare i loro preferiti tra questi, in base alle proprie preferenze. Ad esempio, un autore può specificare un foglio di stile progettato per gli schermi piccoli ed un altro per gli utenti ipovedenti (ad es. con caratteri

grandi). I programmi utente dovrebbero consentire agli utenti di scegliere tra i fogli di stile alternativi.

L'autore può specificare che uno dei fogli di stile alternativi è *preferenziale*. I programmi utente dovrebbero applicare il foglio di stile preferenziale dell'autore, a meno che l'utente abbia selezionato un'alternativa differente.

Specificare fogli di stile esterni

Gli autori specificano fogli di stile esterni tramite i seguenti attributi dell'elemento [LINK](#):

- Si imposti il valore di [href](#) alla posizione del file di foglio di stile.
- Si imposti il valore dell'attributo [type](#) in modo da indicare il linguaggio della risorsa collegata (il foglio di stile). Ciò permette al programma utente di evitare lo scaricamento di un foglio di stile in un linguaggio per fogli di stile non supportato.
- Si specifichi se il foglio di stile è persistente, preferenziale o alternativo:
 - Per rendere persistente un foglio di stile, si imposti l'attributo [rel](#) al valore "stylesheet" e non si imposti l'attributo [title](#).
 - Per rendere preferenziale un foglio di stile, si imposti l'attributo [rel](#) al valore "stylesheet" e si attribuisca un nome al foglio di stile per mezzo dell'attributo [title](#).
 - Per specificare un foglio di stile alternativo, si imposti l'attributo [rel](#) al valore "alternate stylesheet" e si attribuisca un nome al foglio di stile per mezzo dell'attributo [title](#).

I programmi utente dovrebbero fornire agli utenti un sistema per esaminare un elenco di stili alternativi e scegliere da esso. Si raccomanda il valore dell'attributo [title](#) come nome di ciascuna scelta.

In questo esempio, specifichiamo dapprima un foglio di stile persistente, posto nel file `miostile.css`:

```
<LINK href="miostile.css" rel="stylesheet" type="text/css">
```

L'impostazione dell'attributo [title](#) ne fa il foglio di stile preferenziale dell'autore:

```
<LINK href="miostile.css" title="compatto" rel="stylesheet" type="text/css">
```

L'aggiunta della parola chiave "alternate" all'attributo [rel](#) lo rende un foglio di stile alternativo:

```
<LINK href="miostile.css" title="intermedio" rel="alternate stylesheet" type="text/css">
```

CSS esterni e interni

E' **esterno** un foglio di stile definito in un file separato dal documento. Si tratta di semplici documenti di testo editabili anche con il Blocco Note o TextEdit ai quali si assegna l'estensione **.css**.

Un foglio di stile si dice invece **interno** quando il suo codice è compreso in quello del documento. A seconda che si lavori con un CSS esterno o interno variano sintassi e modalità di inserimento. Rispetto a queste diverse modalità si parla di fogli di stile **collegati**, **incorporati** o **in linea**.

Fogli collegati

Per caricare un foglio esterno in un documento esistono due possibilità. La prima e più compatibile è quella che fa uso dell'elemento **<LINK>**. La dichiarazione va sempre collocata all'interno della sezione **<HEAD>** del documento (X)HTML:

```
<html>
<head>
<title>Inserire i fogli di stile in un documento</title>
<link rel="stylesheet" type="text/css" href="stile.css">
</head>
<body>...
```

L'elemento **<LINK>** presenta una serie di attributi di cui è importante spiegare significato e funzione:

1. rel: descrive il tipo di relazione tra il documento e il file collegato. E' **obbligatorio**. Per i CSS due sono i valori possibili: **stylesheet** e **alternate stylesheet**. Approfondimenti nella lezione 5, "Fogli di stile alternativi".

2. href: serve a definire l'URL assoluto o relativo del foglio di stile. E' **obbligatorio**.

3. type: identifica il tipo di dati da collegare. Per i CSS l'unico valore possibile è **text/css**. L'attributo è **obbligatorio**.

4. media: con questo attributo si identifica il supporto (schermo, stampa, etc) cui applicare un particolare foglio di stile. Attributo **opzionale**. L'argomento sarà approfondito nella prossima lezione.

Usare @import

Un altro modo per caricare CSS esterni è usare la direttiva **@import** all'interno dell'elemento **<STYLE>**:

```
<style>
@import url(stile.css);
</style>
```

Questo sistema è uno dei modi più sicuri per risolvere problemi di compatibilità tra vecchi e nuovi browser. Ci torneremo quindi più avanti. Per il momento basti notare che il CSS va collegato definendo un URL assoluto o relativo da racchiudere tra parentesi tonde (ma vedremo che altri modi sono accettati) e che la dichiarazione deve chiudersi con un punto e virgola.

Fogli incorporati

I fogli incorporati sono quelli inseriti direttamente nel documento (X)HTML tramite l'elemento **<STYLE>**. Anche in questo caso la dichiarazione va posta all'interno della sezione **<HEAD>**:

```
<html>
<head>
<title>Inserire i fogli di stile in un documento</title>
<style type="text/css">
body {
background: #FFFFCC;
}
</style>
</head>
<body>...
```

Come si vede il codice inizia con l'apertura del tag **<STYLE>**. Esso può avere due attributi:

1. type (obbligatorio)

2. media (opzionale)

per i quali valgono le osservazioni viste in precedenza. Seguono le regole del CSS e la chiusura di **</STYLE>**.

Fogli in linea

L'ultimo modo per formattare un elemento con un foglio di stile consiste nell'uso dell'attributo **style**. Esso fa parte della collezione di attributi (X)HTML definita **Common**: si tratta di quegli attributi applicabili a tutti gli elementi. La dichiarazione avviene a livello dei singoli tag contenuti nella pagina e per questo si parla di fogli di stile in linea. La sintassi generica è la seguente:

```
<elemento style="regole_di_stile">
```

Se, ad esempio, si vuole formattare un titolo **H1** in modo che abbia il testo di colore rosso e lo sfondo nero, scriveremo:

```
<h1 style="color: red; background: black;">...</h1>
```

Le cose da osservare nel codice sono due. Come valore di **style** si possono dichiarare più regole di stile. Esse vanno separate dal punto e virgola. I due punti si usano invece per introdurre il valore della proprietà da impostare.

```
/* Definisco le regole per il corpo del documento */
body {
    background: White; /* Prima dichiarazione: notate il punto e virgola
finale!*/
```

```
        font-family: Verdana, Geneva, Arial, Helvetica, sans-serif; /* Seconda
dichiarazione*/
        font-size: 12px; /* Terza dichiarazione */
    }

/* Titolo di primo livello */
h1 {
    color: black;
}

/* Paragrafo generico */
p {
    color: black;
    font: 12px/1.5 Georgia, "Times New Roman", serif;
}

#div1 {
    background: Silver; /* Sfondo colr argento */
    width: 300px;
}

#div2 {
    background: Silver;
    width: 300px;
    position: absolute;
    top: 400px;
    left: 400px;
}
```

Capitolo 16. Esempio svolgimento traccia esame di stato

M733 - ESAME DI STATO DI ISTITUTO TECNICO COMMERCIALE

CORSO DI ORDINAMENTO

Indirizzo: PROGRAMMATORI

Tema di: INFORMATICA GENERALE ED APPLICAZIONI GESTIONALI

(Testo valevole per i corsi di ordinamento e per i corsi sperimentali dei Progetti “SIRIO” – programmatori e “MERCURIO”)

Il Codice della Privacy prevede una serie di adempimenti per i soggetti pubblici e privati che trattano dati sensibili con supporti informatici. Il candidato illustri le possibili procedure per:

- garantire l'accesso ai documenti soltanto a soggetti autorizzati;
- gestire i *backup* periodici dei soli documenti contenenti dati sensibili;
- simulare periodicamente la perdita dei dati e il recupero degli stessi dalle copie di *backup*;
- modificare periodicamente i parametri di accesso ai documenti protetti.

Successivamente sviluppi il seguente punto.

Uno studio medico associato, in cui operano medici di base e medici specialisti, dà incarico a una società d'informatica di progettare e realizzare un database contenente i dati anagrafici e professionali dei medici, l'orario delle visite, i tempi medi previsti per ogni visita, il costo delle singole prestazioni, i dati anagrafici dei pazienti e le prestazioni richieste.

Elaborare, dopo aver descritto le caratteristiche dell'applicativo che si intende utilizzare, il progetto definendone i dati necessari e dettagliando la procedura per risolvere la prima e due richieste tra quelle indicate:

1. elenco giornaliero delle visite prenotate per ogni singolo medico;
2. elenco giornaliero delle visite prenotate e non effettuate;
3. elenco settimanale contenente gli appuntamenti di ciascun medico suddivisi per giorno e per ora;
4. elenco cronologico delle visite usufruite da ciascun paziente.

Realizzare, infine, la pagina *web* con la quale lo studio medico pubblicizza la propria attività fornendo l'indicazione dei servizi e il quadro orario.

Dati mancanti opportunamente scelti.

Durata massima della prova: 6 ore.

È consentito soltanto l'uso di calcolatrice non programmabile.

Non è consentito lasciare l'Istituto prima che siano trascorse 3 ore dalla dettatura del tema.

Svolgimento

Parte 1

La **legge sulla privacy**, numero 196/2003, obbliga tutti i soggetti che trattano dati personali di terzi, ad adottare una serie di provvedimenti per evitare l'accesso non autorizzato ai dati e la loro perdita. L'obbligo è esteso persino ai soggetti privati, per cui chiunque tratta dati personali di altre persone fisiche o giuridiche deve rispettare le norme della legge sulla privacy. L'obbligo principale è quello di redigere il documento programmatico sulla sicurezza, DPS, che deve riportare tutte le procedure messe in atto per conformarsi alla legge quali, ad esempio, la gestione delle password, la gestione degli aggiornamenti per i software tra i quali, soprattutto, il sistema operativo e l'antivirus, la gestione delle copie di sicurezza, ecc.

Le pene previste dalla legge sono di tipo civile e penale per chi ne viola le disposizioni; le pene sono particolarmente severe in caso di divulgazione di dati sensibili; i dati, infatti, sono così classificati dalla legge sulla privacy:

- Dato personale: qualunque informazione relativa a persona fisica, persona giuridica, ente od associazione, identificati o identificabili, anche indirettamente, mediante riferimento a qualsiasi altra informazione, ivi compreso un numero di identificazione personale;
- Dati identificativi: i dati personali che permettono l'identificazione diretta dell'interessato;
- Dati sensibili: i dati personali idonei a rivelare l'origine razziale ed etnica, le convinzioni religiose, filosofiche o di altro genere, le opinioni politiche, l'adesione a partiti, sindacati, associazioni od organizzazioni a carattere religioso, filosofico, politico o sindacale, nonché i dati personali idonei a rivelare lo stato di salute e la vita sessuale.

Per proteggere i dati dagli accessi non autorizzati, i moderni sistemi di elaborazione mettono a disposizione del gestore del sistema e del gestore del data base l'autenticazione dell'utente, necessaria affinché chiunque possa accedere ai dati. Un buon sistema di gestione delle password deve prevedere livelli diversi di protezione, ovvero con diritti diversificati in funzione dell'autenticazione, e la loro modifica periodica, per limitare il rischio che esse possano essere scoperte da persone non autorizzate; la scadenza delle password ed il conseguente obbligo di modifica vanno gestiti dal sistema; in altre parole, ad intervalli regolari, il sistema obbliga l'utente a cambiare la propria password di accesso.

La sicurezza degli accessi, però, non basta per essere in regola con la legge sulla privacy; è infatti necessario provvedere con regolarità alle copie di backup,

perché i dati non devono essere esposti a rischi di perdita totale o parziale dovuta sia a guasti hardware che furti, incendi, alluvioni ecc. La copia dei soli dati sensibili può essere pianificata nel sistema in modo da non essere legata all'iniziativa di un operatore, comunque necessario solo per correttamente riporre in un armadio di sicurezza o cassaforte i supporti utilizzati per le copie. Per effettuare la sola copia dei dati sensibili, è sufficiente specificare, nell'operazione pianificata di backup, solo le cartelle contenenti tali dati; per esempio è possibile effettuare un backup completo ad inizio mese, eventualmente su destinazioni diverse per i mesi pari e dispari, ed un backup differenziale o incrementale ogni giorno, la cui scelta dipende dal volume dei file che subiscono quotidianamente modifiche. La scelta mesi pari/mesi dispari consente di evitare che il backup completo di inizio mese sovrascriva il precedente che, comunque, sarà sovrascritto dal mese successivo pari o dispari.

Una verifica della correttezza delle copie può essere, innanzitutto, già fatta subito dopo l'operazione di backup, in quanto i software di backup consentono di effettuare la “verifica dopo il backup” e, comunque, in via differita ripristinando i dati in una cartella o su un computer diverso da quello in cui sono presenti i dati originali onde evitare che la verifica li sovrascriva con evidenti rischi; si può pertanto procedere alla loro verifica confrontandoli con quelli originali, se la verifica è condotta subito dopo la copia, oppure aprendo i documenti ripristinati su altra destinazione ed effettuando una verifica a campione; è evidente che le copie saranno protette da password, in modo da evitare che esse siano leggibili a persone non autorizzate che dovessero riuscire ad impadronirsi, anche in maniera illegale, dei supporti.

Parte 2

Da una attenta analisi della traccia proposta si evince, in maniera immediata, la necessità di definire una entità MEDICO, che raccolga i dati relativi ai medici dello studio associato ovvero dati anagrafici e professionali; in particolare utilizzerò un campo booleano per stabilire se il medico è di base e tre campi specializzazioni, ritenendo ampiamente sufficiente la possibilità di archiviare anche le specializzazioni relative a medici che dovessero averne più di una; in questo modo si evita una associazione molti a molti che, comunque, in una direzione avrebbe poca molteplicità: un medico mai avrà decine di specializzazioni. In ogni caso non sarebbe un errore utilizzare una entità SPECIALIZZAZIONE e la scelta adottata è solo dettata da criteri di semplicità della base dei dati; in questo modo, tra l'altro, non si esclude la possibilità di avere medici di base specializzati. Per l'orario delle visite non è chiaro se si richiedono i giorni della settimana ed i relativi orari in cui ciascun medico è disponibile; ritengo sia questo il caso così che la procedura di prenotazione delle visite possa consigliare orari idonei. Si rende, pertanto, necessaria una seconda entità, ORARIO, che riporti, per ciascun medico, i giorni della settimana ed i relativi orari in cui esso è disponibile per effettuare visite. Una menzione a parte è necessaria per i campi User e Password che consentiranno l'accesso all'area riservata del sito

WEB; il campo User sarà indicizzato con duplicati non ammessi per evitare che due medici abbiano le stesse credenziali per l'accesso al sito. Si potrebbe utilizzare il codice del medico che, come vedremo, è chiave primaria, ma, in questo caso, la scelta sarà fatta dall'operatore dello studio associato; per questo motivo si decide di avere un campo distinto. Si potrebbe anche utilizzare l'indirizzo e-mail come nome utente per l'accesso al sito.

In maniera altrettanto immediata si rileva l'entità PAZIENTI, contenente i dati sui pazienti: anagrafici e relativi allo stato di salute. Anche in questo caso inserirò i campi User e Password per l'accesso all'area riservata del sito WEB.

Per concludere è necessaria l'entità PRESTAZIONE, riferita a tutte le prestazioni che lo studio può offrire ai pazienti, con i relativi costi ed i tempi in genere necessari per la sua attuazione.

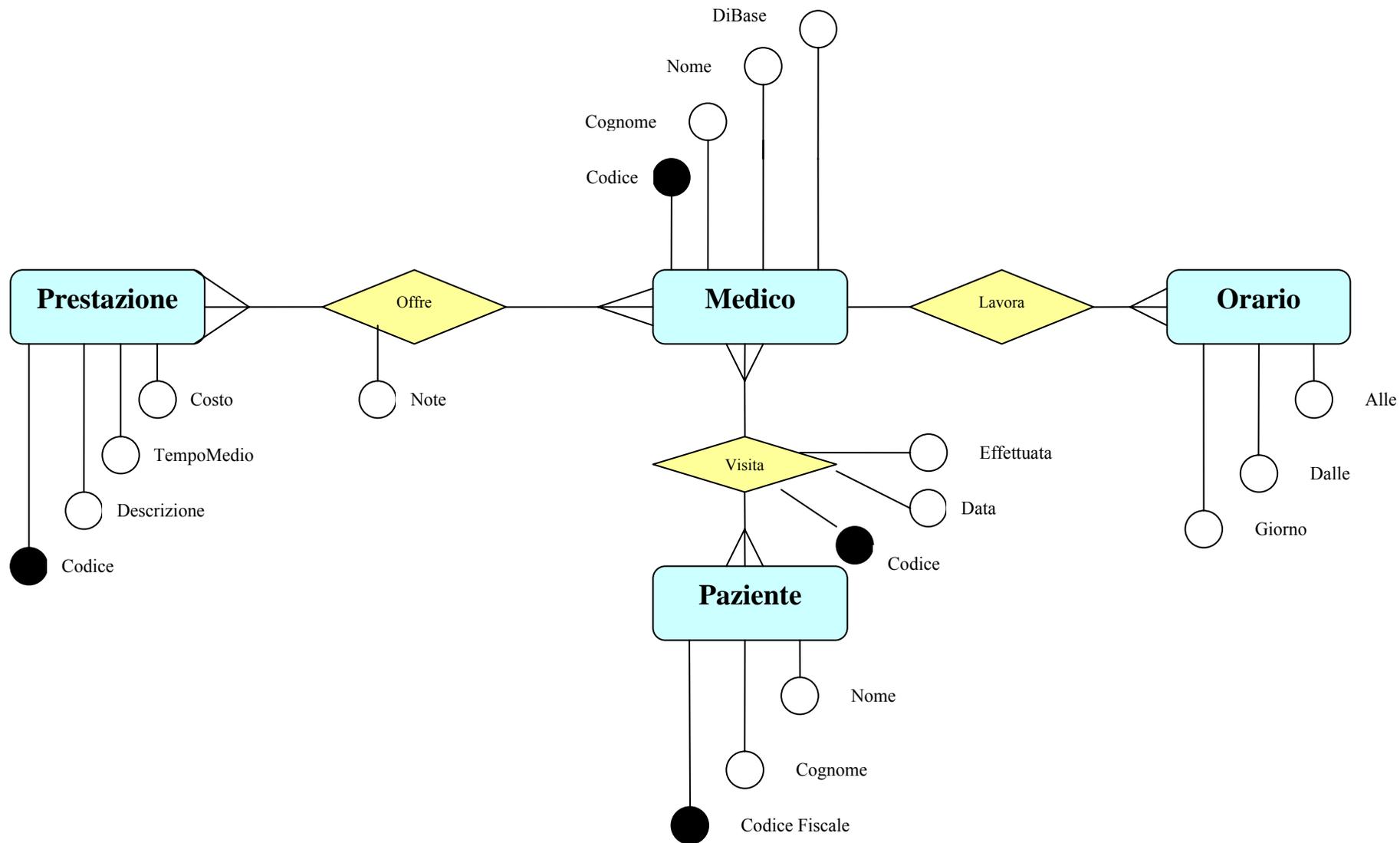
Riepilogando, ritengo di utilizzare le seguenti entità:

- MEDICO: Dati sui medici dello studio associato;
- ORARIO: Necessaria per le disponibilità orarie settimanali dei medici;
- PRESTAZIONI: Elenco delle prestazioni di cui un paziente può usufruire rivolgendosi allo studio;
- PAZIENTE: Dati sui pazienti che si rivolgono allo studio associato.

Le associazioni tra le entità di cui sopra saranno:

- MEDICO/PRESTAZIONI: Molti a molti;
- MEDICO/ORARIO: Uno a molti con integrità referenziale;
- MEDICO/PAZIENTE: Molti a molti.

Da quanto detto si ricava il seguente schema concettuale:



Dallo schema concettuale si passa allo schema logico costituito, tra l'altro, dalle tabelle PreMed, che realizza l'associazione molti a molti tra MEDICO e PRESTAZIONE, e VISITE, che realizza l'associazione molti a molti tra MEDICO e PAZIENTE.

1. Tabella MEDICI: Derivata dall'entità MEDICO, avrà il seguente tracciato record:

Nome	Tipo	Lunghezza	Note
Codice	String	10	Chiave Primaria
Cognome	String	20	
Nome	String	20	
DiBase	Boolean		Vero se medico di base
Specializzazione1	String	20	
Specializzazione2	String	20	
Specializzazione3	String	20	
Indirizzo	String	25	
CAP	String	5	
Citta	String	20	
Provincia	String	2	
Telefono	String	20	
Cellulare	String	20	
Fax	String	15	
E-Mail	String	30	
Codice Fiscale	String	16	
PartitaIVA	String	11	
Note	String	Memo	Note generiche
User	String	20	
Password	String	12	

Campi indicizzati oltre alle chiavi: Nessuno, archivio sicuramente costituito da pochi record per cui non si ritiene opportuno indicizzare su alcun campo.

Si noti la scelta di un codice come chiave primaria mnemonica scelta dall'operatore, che ne consentirà una più semplice gestione rispetto al codice fiscale.

Sulla tabella saranno eseguite le normali operazioni di inserimento, aggiornamento, ricerca, visualizzazioni e stampe varie.

2. Tabella ORARIO: Derivata dall'entità ORARIO, avrà il seguente tracciato record:

Nome	Tipo	Lunghezza	Note
Medico	String	10	Chiave esterna
Giorno	Byte		Vedi nota
Dalle	Date		Ora/Minuti iniziali
Alle	Date		Ora/Minuti finali

Campi indicizzati oltre alle chiavi: nessuno.

Sulla tabella saranno eseguite le normali operazioni di inserimento, aggiornamento, ricerca, visualizzazioni e stampe varie, anche in congiunzione con l'archivio MEDICI.

Il campo "Giorno" sarà sottoposto a controllo del software e potrà contenere solo i valori da 1 a 7 corrispondenti a Lunedì, Martedì, ..., Domenica; la Domenica non è stata esclusa perché è possibile che si decida di lavorare, anche se in maniera occasionale. In questo modo, per ciascun medico, sarà possibile definire più intervalli di tempo anche nell'ambito della stessa giornata.

3. Tabella PRESTAZIONI: Derivata dall'entità PRESTAZIONE, avrà il seguente tracciato record:

Nome	Tipo	Lunghezza	Note
Codice	String	10	Chiave Primaria
Descrizione	String	50	
Tempo	Byte		In minuti
Costo	Currency		

Campi indicizzati oltre alle chiavi: Nessuno.

Sulla tabella saranno eseguite le normali operazioni di inserimento, aggiornamento, ricerca, visualizzazioni e stampe varie.

4. Tabella PreMed: Tabella che realizza l'associazione molti a molti tra prestazioni e medici.

Nome	Tipo	Lunghezza	Note
Prestazione	String	10	Chiave esterna
Medico	String	10	Chiave esterna
Tempo	Byte		In minuti
Costo	Currency		

Campi indicizzati oltre alle chiavi: nessuno.

Sulla tabella saranno eseguite le normali operazioni di inserimento, aggiornamento, ricerca, visualizzazioni e stampe varie, anche e soprattutto in congiunzione con l'archivio MEDICI e PRESTAZIONI.

I campi Tempo e Costo sembrano ridondanti ma, in pratica, sono utilizzati nell'associazione perché è possibile che la stessa prestazione sia praticata da medici diversi con tempi e costi diversi; ma allora potrebbe essere eliminato dalla tabella prestazioni e invece si decide di tenerli per una semplificazione nell'immissione dei dati; la procedura ritiene che i valori da utilizzare come default sono quelli nella tabella prestazioni nel caso in cui quelli corrispondenti di PreMed siano zero altrimenti li/lo sostituisce con quest'ultimi. Vedremo che questi campi sono riportati

anche nella tabella VISITE, ma, in questo caso, saranno i valori effettivi relativi a quella visita praticata da quel medico su quel paziente; non è escluso, infatti, che, per esempio, si possa applicare, in alcune visite, uno sconto sui prezzi normalmente praticati.

5. Tabella Pazienti: Derivata dall'entità PAZIENTE, avrà il seguente tracciato record:

Nome	Tipo	Lunghezza	Note
CodiceFiscale	String	16	Chiave Primaria
Cognome	String	20	
Nome	String	20	
DataNascita	Date		Data di nascita
Indirizzo	String	25	
CAP	String	5	
Citta	String	20	
Provincia	String	2	
Telefono	String	20	
Cellulare	String	20	
Fax	String	15	
E-Mail	String	30	
GruppoS	String	5	Gruppo sanguigno
Allergie	String	100	Allergie
Patologie	String	100	Patologie
PressioneMin	Byte		Pressione minima
Pressione Max	Byte		Pressione massima
Peso	Byte		Peso in KG
Altezza	Byte		Altezza in cm
Corporatura	String	20	Grasso, robusto, ecc.
Note	String	Memo	Note generiche
User	String	20	
Password	String	12	

Campi indicizzati oltre alle chiavi: Cognome & Nome con duplicati ammessi, per semplificare la ricerca, e User con duplicati non ammessi.

Sulla tabella saranno eseguite le normali operazioni di inserimento, aggiornamento, ricerca, visualizzazioni e stampe varie.

6. Tabella Visite: Tabella che realizza l'associazione molti a molti tra medici e pazienti.

Nome	Tipo	Lunghezza	Note
ID	Contatore		Chiave primaria
Paziente	String	16	Chiave esterna
Medico	String	10	Chiave esterna

Data	Date		
CodicePrestazione	String	10	Chiave esterna
Effettuata	Boolean		True se effettuata
Costo	Currency		Costo effettivo
PressioneMin	Byte		Pressione minima
Pressione Max	Byte		Pressione massima
Peso	Byte		Peso in KG
Altezza	Byte		Altezza in cm
Referto	Memo		Referto
Prescrizione	Memo		Prescrizione medica
Tempo	Byte		In minuti
Note	Memo		Note generiche

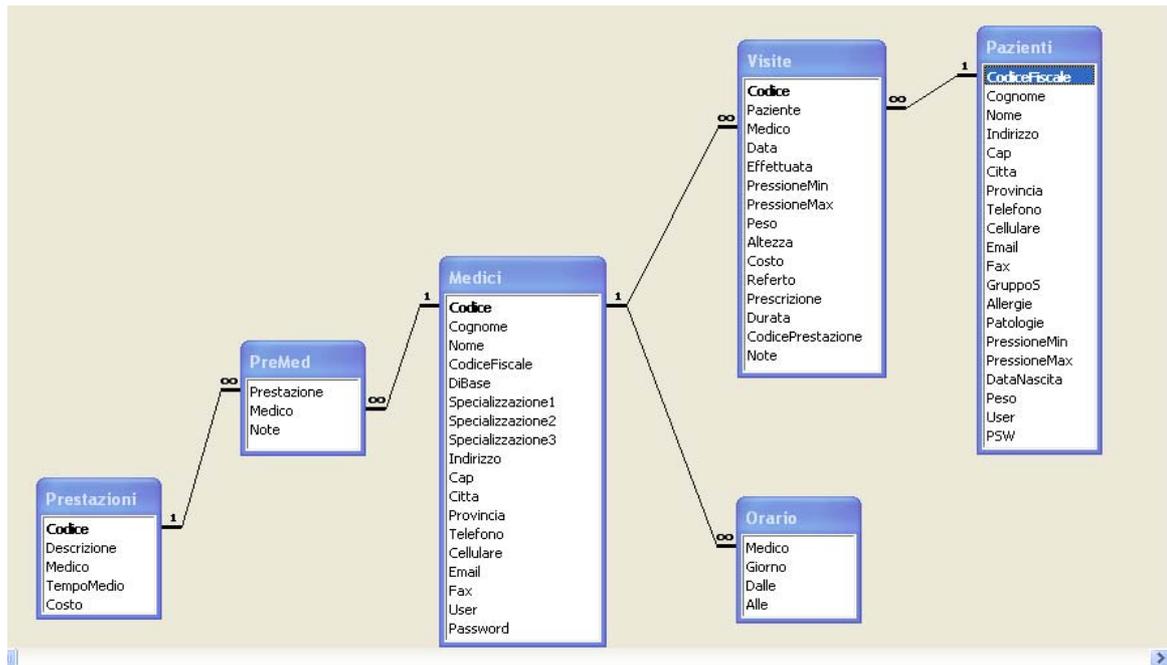
Campi indicizzati oltre alle chiavi: Nessuna.

Questa tabella viene utilizzata sia per le prenotazioni delle visite che per memorizzare i risultati di una visita effettuata. Lo stato di una visita può essere:

- Prenotata: $Data \geq Data\ sistema$;
- Non effettuata: $Effettuata = false$ and $Data \leq Data\ sistema$;
- Effettuata: $Effettuata = true$.

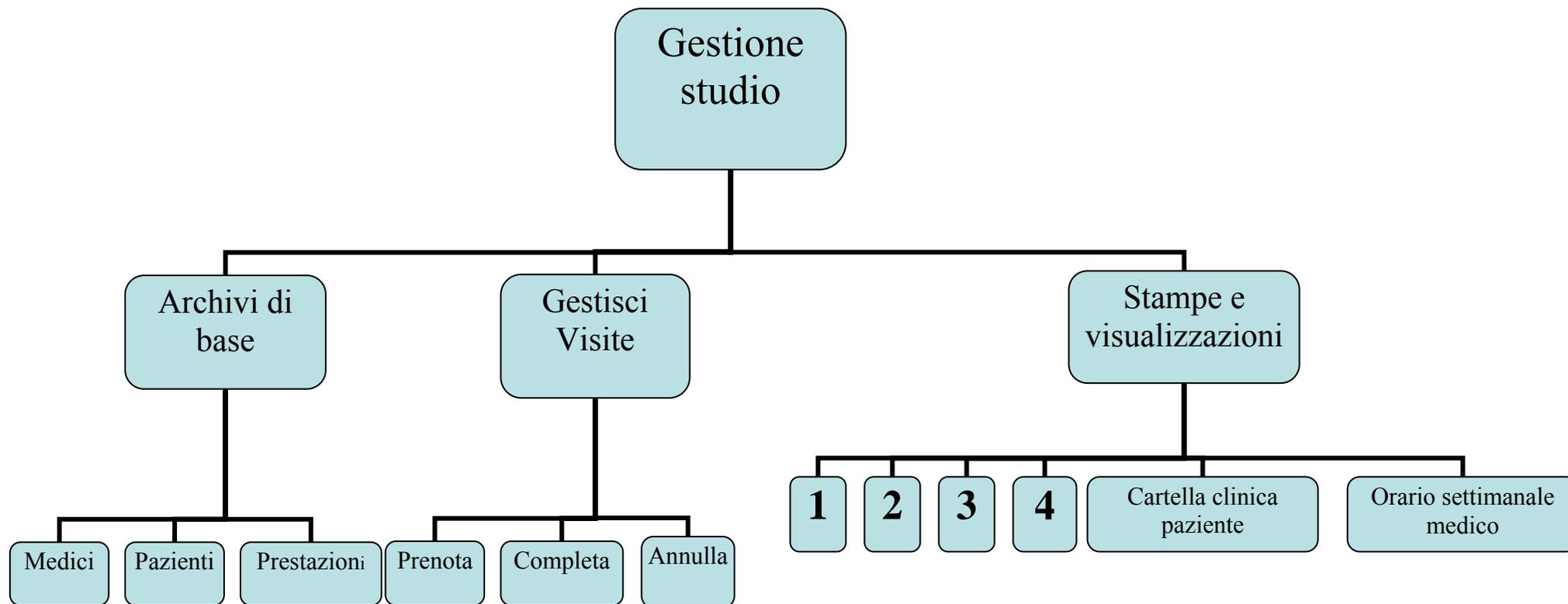
Bisogna solo chiarire i campi presenti in questa tabella e nella tabella Pazienti, come, per esempio, PressioneMin e PressioneMax; anche in questo caso non si tratta di ridondanza; nella tabella Pazienti sono valori generici mentre per la tabella Visite sono i valori effettivamente misurati nel giorno in cui si effettua la visita. Anche l'altezza può variare ed essere importante, nel caso in cui si tratti, per esempio, di visite pediatriche. Per visite dietologiche è importante il peso, per malattie cardiovascolari i valori di pressione e così via; per questo si sono utilizzati tutti questi campi dei quali solo alcuni o, al limite, tutti o nessuno saranno valorizzati dal medico durante la visita.

Si noti anche la presenza della chiave esterna Codice Prestazione che, di fatti, realizza una associazione uno a molti tra Prestazioni e Visite.



Questo il layout delle relazioni in MS ACCESS che, naturalmente, non potrà essere riportato nel compito svolto durante l'esame di stato.

Sulla base di dati appena descritta posso essere eseguite numerosissime operazioni tra le quali, oltre a quelle descritte dalla traccia e che si esauriscono con una semplice query in SQL, la prenotazione di una visita, la stampa o la visualizzazione della cartella clinica di un paziente, completa o parziale, ovvero riferita ad un solo medico e/o ad una sola prestazione; e l'elenco potrebbe essere molto lungo per cui concludo con il funzionigramma che rappresenta la gerarchia delle funzioni che l'applicativo dovrà realizzare:



Le funzioni 1-4 sono quelle richieste dalla traccia.

La gestione degli archivi di base consente le normali operazioni di inserimento, aggiornamento e cancellazione dei dati presenti nelle tabelle; la gestione delle visite consente di prenotare una visita, eventualmente anche on line attraverso il sito, di annullarla e di completarla, da parte del medico, dopo che essa è stata effettuata.

Le operazioni di stampa e visualizzazione sono sempre molto numerose; sono state riportate quelle richieste dalla traccia e la stampa della cartella clinica di un paziente, ovvero l'elenco delle sue visite, tutte o eventualmente solo quelle relative ad un medico e/o una prestazione.

L'ultima funzione consente di stampare o visualizzare l'orario settimanale di un medico di un medico.

Con riferimento agli algoritmi richiesti, fornisco le query SQL.

1. elenco giornaliero delle visite prenotate per ogni singolo medico:

```
SELECT Visite.*
FROM Visite
WHERE Visite.Effettuata=False AND Visite.Medico=[Immettere codice
medico] AND Day([visite].[data])=[immettere data];
```

2. elenco giornaliero delle visite prenotate e non effettuate:

```
SELECT Visite.*
FROM Visite
WHERE effettuata=False;
```

3. elenco settimanale contenente gli appuntamenti di ciascun medico suddivisi per giorno e per ora:

```
SELECT medici.codice, medici.cognome, medici.dibase, Visite.codice,
visite.paziente, visite.medico, visite.data, visite.codiceprestazione
FROM Visite
WHERE effettuata=False And day(visite.data) Between [immettere data da]
And [immettere data a];
```

4. elenco cronologico delle visite usufruite da ciascun paziente.

```
SELECT pazienti.*, Visite.*
FROM visite INNER JOIN pazienti ON pazienti.codicefiscale=visite.paziente
WHERE visite.effettuata=True
ORDER BY pazienti.cognome, visite.data DESC;
```

Per l'operazione riporto anche il diagramma a blocchi realizzato in VB.NET.

Diagramma

L'ultima parte della traccia richiede la realizzazione della pagina WEB per pubblicizzare lo studio medico associato. Ritengo opportuno prima indicare i

contenuti del sito WEB. Il sito WEB dovrà essere fortemente dinamico, per cui le informazioni che in esso saranno pubblicate sono quelle contenute nel data base. E' necessario, pertanto, dividere il sito in una parte ad accesso libero, dove vengono riportati informazioni riguardanti lo studio, i medici e, soprattutto, le prestazioni che lo studio offre, ed una ad accesso riservato per medici e pazienti. E' evidente che i pazienti potranno accedere soltanto ai propri dati, agli orari delle visite e, eventualmente, procedere ad una prenotazione on line; ricordo che i dati presenti nel data base sono non solo identificativi di persone fisiche ma addirittura sensibili, per cui vanno assolutamente protetti da accessi non autorizzati adottando le misure esposte in precedenza.

La pagina iniziale potrebbe essere la seguente:



User name

Password

Aggiungere altri oggetti

Il codice HTML è il seguente:

Codice HTML

P.S.: Il database access "Studio" è disponibile nella solita cartella condivisa sul server.



Capitolo 17.

Il sistema operativo Windows

17.1.1. Utilità del sistema

Un sistema operativo mette a disposizione dell'utilizzatore una serie di **programmi di utilità**, che svolgono le cosiddette **operazioni ausiliarie all'elaborazione dei dati**, quali copie di sicurezza, ottimizzazione delle risorse, individuazione di errori, aggiornamenti, ecc.

Il sistema operativo Windows non si sottrae a questa regola, anzi fornisce all'utilizzatore anche alcuni **semplici programmi applicativi** quali il browser **Internet Explorer** per la navigazione in Internet, il gestore della posta elettronica **outlook express**, **Windows Media Player** quale lettore multimediale per file audio, CD, video, ecc., il programma di word processing **WordPad**, la **calcolatrice**, ecc.; di questi ultimi ritengo, ovviamente, di fare a meno di scrivere anche un sol rigo; se vi interessa, potete imparare da soli il loro funzionamento utilizzando la guida in linea.

Sui programmi di utilità, invece, ritengo opportuno scrivere qualcosa, anche se potete tranquillamente utilizzare la loro guida in linea; per alcuni di essi, infatti, ritengo opportuno dare tutte quelle informazioni, in genere non comprese nella guida, che ne rendono più semplice la comprensione ed una conoscenza che non si ferma al mero utilizzo del software.

17.1.2. Scandisk



Il programma di utilità **scandisk** consente di individuare e correggere errori **logici** e **fisici** sui supporti di memoria quali hard disk, floppy disk, ecc.; questa utility si avvia accedendo, mediante il pulsante destro del mouse, al menù contestuale del volume che si vuole controllare, quindi <proprietà/strumenti>:



Scandisk individua e corregge errori logici causati da spegnimenti irregolari del sistema, cadute di alimentazione, errori del sistema operativo, ecc. Due dei problemi individuati e corretti da scandisk sono i **riferimenti incrociati**, ovvero due file allocati su uno o più cluster comuni, ed i **cluster persi**, ovvero che risultano occupati nella FAT ma nessun file punta ad essi. Nel primo caso la parte comune viene duplicata ed assegnata in maniera indipendente ai due file mentre nel secondo caso scandisk ripristina i cluster come spazio libero oppure crea dei file con nomi **FILEn.chk**, dove n parte da 0001 e si incrementa per ogni file creato; tali file vengono, in genere, registrati nella radice del volume controllato ed è a cura dell'utilizzatore verificarne il contenuto ed eventualmente procedere ad una loro cancellazione oppure ridenominazione. L'utente può decidere, di volta in volta, a quale correzione procedere oppure lasciare la decisione al sistema.

Lo **scandisk approfondito** consente di verificare la bontà della superficie dei dischi magnetici ed è avviato solo se richiesto dall'utente; si tratta di un'operazione che richiede un tempo piuttosto alto e, in caso di rilevamento di errori fisici su uno o più settori, il sistema tenta un recupero dei dati allocati su di essi ed eventualmente li copia in una zona non difettosa; i settori risultati danneggiati vengono "marcati" come guasti e non più utilizzati dal sistema.

Vi consiglio di utilizzare raramente il controllo approfondito e, comunque, sempre nei casi in cui si ricevano informazioni su errori di I/O, lettura/scrittura, su disco; nel caso in cui vengano individuati uno o più settori difettosi, vi consiglio di ripetere l'operazione dopo qualche giorno e, se il sistema continua a trovare nuovi settori difettosi, è opportuno procedere alla sostituzione del disco.

Il sistema operativo, in genere, esegue uno **scandisk automatico** all'avvio, quando rileva errori logici in lettura/scrittura o quando lo spegnimento non avviene in modo corretto. E' ormai chiara l'importanza che rivestono le memorie di massa nell'ambito di un sistema di elaborazione; un guasto al disco rigido comporta, quasi sempre, la perdita dei dati; è possibile recuperare i dati, o parte di essi, da un disco danneggiato rivolgendosi ad aziende specializzate, ma il costo di tale operazione è, in genere, molto alto e a volte proibitivo per le economie di chi ha perso i dati.

Da alcuni anni la tecnologia **SMART, Self-Monitoring, Analysis, Reporting Technology**, consente di predire un possibile guasto al disco rigido e di avvisare l'utilizzatore che, così, può procedere alla sua sostituzione, prima che si guasti definitivamente, con conseguente perdita dei dati. La tecnologia SMART coinvolge, ovviamente, i dischi rigidi, il BIOS del sistema ed il sistema operativo; quando le prestazioni del disco degradano, perché le operazioni di lettura/scrittura avvengono con tempi più alti o comunque con difficoltà, il firmware del disco rigido avverte il BIOS di un possibile deterioramento delle superfici del disco e, quindi, un messaggio viene inviato all'utente che, ovviamente, provvede alla sostituzione del disco.

17.1.3. Defrag



La frammentazione è l'allocazione non contigua dei file sulle memorie di massa; la frammentazione, aumentando gli spostamenti della testina sui dischi rigidi e, quindi, il tempo di posizionamento durante la lettura dei dati da un file frammentato, *rallenta le prestazioni del sistema*. *Defrag elimina la frammentazione* riscrivendo, in pratica, l'intero disco in modo tale che tutti i file siano allocati in maniera fisicamente contigua e che lo spazio libero sia tutto alla fine del volume.

Prima di avviare l'operazione, è possibile analizzare il volume per verificare l'opportunità di procedere alla deframmentazione, che richiede molto tempo ed anche qualche raro rischio di perdita dati dovuta ad eventuali interruzioni accidentali; per questo vi consiglio di eseguire questa operazione raramente e solo in casi di necessità.

L'operazione di deframmentazione può essere avviata come già descritto per lo scandisk oppure da <Start/Tutti i programmi/Accessori/Utilità del sistema/Utilità di deframmentazione dischi>.

17.1.4. Backup



Ormai abbiamo già perfettamente capito la grande importanza dei *dati*, per cui essi *vanno protetti da qualsiasi causa che ne possa determinare la perdita totale o parziale*. Anche i PC di casa contengono dati importanti: si pensi alle foto o alla musica oltre che ai documenti; ebbene *sempre* è opportuno copiare, a scopi precauzionali, i dati su supporti esterni, dai quali eventualmente ripristinarli in caso di necessità.

Potete copiare i dati semplicemente utilizzando esplora risorse e le operazioni di Copia/Incolla, ma le copie vanno effettuate con cadenza giornaliera e non è possibile procedere ogni giorno alla copia completa del disco rigido perché richiederebbe troppo tempo. Per poter effettuare copie giornaliere senza impiegare un tempo eccessivo, è necessario utilizzare un programma specifico opportunamente *pianificato*, ovvero in grado di procedere alla copia ad *intervalli di tempo regolari e senza intervento dell'operatore*. Anche se disponiamo di un RAID è opportuno procedere alla copia su supporti esterni per evitare i rischi derivanti, per esempio, da furti o incendi.

Per poter comprendere la pianificazione del backup, occorre parlare degli attributi dei file; i file possono avere uno o più attributi o proprietà; in effetti, con riferimento al backup, l'unico attributo che interessa è quello di archivio, ma ritengo opportuno riportare tutti gli attributi dei file:

- **A - Archive:** l'attributo di archivio viene assegnato automaticamente ad un file quando subisce una modifica;
- **H – Hidden:** un file con l'attributo di nascosto non viene riportato nell'elenco del contenuto di una cartella; è possibile modificare le impostazioni in modo da poterli vedere da gestione risorse in
<Strumenti/Opzioni/Visualizzazione/visualizza file e cartelle nascosti>;
- **S - System:** un file con l'attributo di sistema è nascosto e non può essere cancellato; è possibile modificare le impostazioni in modo da poterli vedere da gestione risorse in
<Strumenti/Opzioni/Visualizzazione/nascondi i file protetti di sistema>;
- **R – Read:** i file con attributo di sola lettura non possono essere modificati e cancellati solo mediante apposita conferma.

L'operazione di backup consente di selezionare i file da copiare: è possibile selezionare da un singolo file ad un intero volume o, addirittura, più volumi, anche su sistemi accessibili attraverso i servizi di rete. Una volta selezionati i file da copiare, è possibile scegliere tra i seguenti tipi di backup:

- **Completo o normale:** vengono copiati tutti i file selezionati, quelli con e senza l'attributo di archivio; l'attributo di archivio viene tolto;
- **Differenziale:** vengono copiati solo i file selezionati che hanno l'attributo di archivio; l'attributo di archivio non viene tolto, per cui questi file saranno di nuovo copiati dal successivo differenziale;
- **Incrementale:** vengono copiati solo i file selezionati che hanno l'attributo di archivio; l'attributo di archivio viene tolto, per cui questi file non saranno di nuovo copiati dal successivo incrementale;
- **Giornaliero:** vengono copiati tutti i file selezionati e modificati nel giorno in cui avviene la copia; si tratta di un backup che comporta qualche rischio se non viene eseguito tutti i giorni;
- **Copia:** vengono copiati tutti i file selezionati, quelli con e senza l'attributo di archivio; l'attributo di archivio non viene tolto; si tratta di una copia extra pianificazione perché non interferisce con quelli pianificati.

Una pianificazione delle copie di backup deve prevedere un backup completo, per esempio mensile, e un differenziale giornaliero oppure un backup completo ed un incrementale giornaliero.

L'operazione inversa, che, a causa di una perdita accidentale, riporta i dati dai supporti esterni, sui quali è avvenuta a suo tempo la copia, al disco rigido, è detta restore o ripristino. Con il differenziale è possibile ripristinare i dati, in caso di necessità, disponendo del completo e dell'ultimo differenziale, mentre con l'incrementale ho bisogno, come è facile intuire, di tutti gli incrementali e, ovviamente, del completo; a questo svantaggio, l'incrementale offre il vantaggio di

essere più veloce perché copia i file copiati in precedenza solo se questi hanno subito una modifica; il differenziale, invece, aumenta di giorno in giorno il numero di file copiati, perché copia tutti i precedenti più quelli che subiscono modifiche. Il restore corretto, disponendo di un completo e più incrementali, deve procedere ripristinando il completo seguito da tutti gli incrementali nell'ordine con il quale, a suo tempo, sono stati realizzati; per questo motivo è necessario numerare i supporti esterni multipli utilizzati per le copie incrementali.

Il differenziale e tutti gli incrementali non sono più necessari quando si procede ad una nuova copia completa.

Capitolo 9.	5
Data Base	5
9.1. Introduzione	5
9.2. Modelli	7
9.2.1. Modello E/R	8
9.2.2. Modelli logici	12
9.3. Modello relazionale	13
9.3.1. Operazioni su tabelle	16
9.3.2. La normalizzazione	20
9.4. DBMS	22
9.5. Esercizi	24
Capitolo 10. Il linguaggio SQL	25
10.1. Il linguaggio SQL	25
Capitolo 11. Software	41
11.1. Introduzione e non solo	41
11.2. Sistema operativo	45
11.2.1. Nucleo	47
11.2.2. Gestione della memoria	51
11.2.3. Gestione delle periferiche	55
11.2.4. File system	57
11.2.5. Shell	60
11.2.6. Programmi di utilità e software applicativo	61
11.2.7. Accounting e controllo degli accessi	61
11.2.8. Avviamento del sistema	62
11.3. Produzione del software applicativo	63
11.4. La qualità del software	66
Capitolo 12. Reti Informatiche	68
12.1. Trasmissione dell'informazione	68
12.2. Reti informatiche	72
12.2.1. Telematica: Hardware	78
12.2.2. Telematica: Software	80
12.2.2.1. Il protocollo TCP/IP	81
12.2.2.2. Altri protocolli	85
12.2.3. Il modello OSI	87
12.3. Internet	87
Capitolo 13. Sistemi informativi	90
Capitolo 14.	95
Informatica e diritto, sicurezza	95
14.1. Diritto d'autore	95
Diritti di utilizzazione economica	98
Cosa è possibile utilizzare liberamente?	98
Estinzione del diritto economico	99
Diritti connessi all'esercizio del diritto d'autore: diritti relativi al ritratto	99
14.2. Criminalità informatica	101
14.3. Sicurezza informatica	103
Tipi di sicurezza e di attacchi	103
Proprio sulla base di queste osservazioni, quando si parla di "sicurezza informatica" spesso si distinguono i concetti di <i>sicurezza passiva</i> e di <i>sicurezza attiva</i> .	103

Sicurezza passiva	103
Sicurezza attiva	103
Sicurezza nelle aziende	103
Sicurezza dei programmi	104
Principali tecniche di difesa:	104
14.4. Virus informatici	105
Cosa è un virus, dove si trova e come funziona	105
Tipologie di virus	106
14.5. Dialer	107
<i>Dialer legittimi</i>	107
Dialer illegali	108
Danni causati dai dialer	108
14.6. LEGGE SULLA PRIVACY	109
Capitolo 15. HTML.....	114
15.1. Documenti HTML	114
Cos'è un documento HTML	114
Tags	114
Un documento HTML minimo	114
Markup Tags	115
HTML	115
INTESTAZIONE (HEAD)	115
BODY	115
Formato dei caratteri	121
Escape Sequences	121
15.2. Riferimenti ipertestuali (Linking)	122
Riferimenti relativi e assoluti	122
Riferimenti a sezioni specifiche	123
Mailto	124
15.3. Inserimento di immagini	124
Dimensionamento	124
Allineamento	125
Background	125
Background Color	125
15.4. Tabelle	126
Elementi della tabella	126
Formato generale di una tabella	126
15.5. I FOGLI CSS	127
Impostare il linguaggio predefinito per i fogli di stile	128
Informazioni di stile in riga	128
Informazioni di stile nell'intestazione: l'elemento STYLE	129
Fogli di stile esterni	131
Fogli di stile preferenziali ed alternativi	131
Specificare fogli di stile esterni	132
CSS esterni e interni	132
Fogli collegati	133
Usare @import	133
Fogli incorporati	134
Fogli in linea	134
Capitolo 16. Esempio svolgimento traccia esame di stato	136
Capitolo 17.	150

Il sistema operativo Windows	150
17.1.1. Utilità del sistema.....	150
17.1.2. Scandisk	150
17.1.3. Defrag.....	152
17.1.4. Backup	152